

Тайнан Сильвестр

Г Е Ы Н



Д Н Э Я Ы Н

Рецепты успеха
лучших компьютерных игр
от Super Mario и Doom до Assassin's Creed
и дальше



Designing Games

A Guide to Engineering Experiences

Tynan Sylvester

O'REILLY®

BEIJING • CAMBRIDGE • FARNHAM • KOLN • SEBASTOPOL • TOKYO

Рецепты успеха лучших компьютерных игр
от Super Mario и Doom до Assassin's Creed
и дальше

ГЕЙМДИЗАЙН

Тайнан Сильвестр



Санкт-Петербург • Москва • Екатеринбург • Воронеж
Нижний Новгород • Ростов-на-Дону
Самара • Минск

2020

ББК 32.973.23-018.9
УДК 004.388.4

Тайнан Сильвестр

Т14 Геймдизайн. Рецепты успеха лучших компьютерных игр от Super Mario и Doom до Assassin's Creed и дальше / Пер. с англ. М. Панина, А. Поповой. — СПб.: Питер, 2020. — 448 с.: ил.

ISBN 978-5-4461-1376-7

Что такое ГЕЙМДИЗАЙН? Это не код, графика или звук. Это не создание персонажей или раскрашивание игрового поля. Геймдизайн — это симулятор мечты, набор правил, благодаря которым игра оживает. Как создать игру, которую полюбят, от которой не смогут оторваться? Знаменитый геймдизайнер Тайнан Сильвестр на примере кейсов из самых популярных игр рассказывает, как объединить эмоции и впечатления, игровую механику и мотивацию игроков. Познакомьтесь с принципами дизайна, которыми пользуются ведущие студии мира! Создайте игровую механику, вызывающую эмоции и обеспечивающую разнообразие. Узнайте, как объединить сюжет и интерактивность. Используйте взаимодействия, которые заставят игроков проникать друг другу «в голову». Вовлекайте в действие с помощью наград. Планируйте, тестируйте и анализируйте геймдизайн последовательно, а не пытайтесь все решить заранее. Узнайте, как позиционирование игры на рынке влияет на геймдизайн.

Руководитель проекта
Ю. Сергиенко
Ведущий редактор
К. Тульцева
Научный редактор
И. Девятко
Литературный редактор
М. Петруненко
Обложка
В. Мостипан
Корректоры
С. Беляева, М. Одинокова
Верстка
Е. Неволainen

16+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ISBN 978-1449337933 англ. Authorized Russian translation of the English edition of Designing Games (ISBN 9781449337933) © 2013 Tynan Sylvester
This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

ISBN 978-5-4461-1376-7 © Перевод на русский язык ООО Издательство «Питер», 2020
© Издание на русском языке, оформление ООО Издательство «Питер», 2020

Права на издание получены по соглашению с O'Reilly. Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав. Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги. Издательство не несет ответственности за доступность материалов, ссылки на которые вы можете найти в этой книге. На момент подготовки книги к изданию все ссылки на интернет-ресурсы были действующими.

Изготовлено в России. Изготовитель: ООО «Прогресс книга».
Место нахождения и фактический адрес: 194044, Россия, г. Санкт-Петербург,
Б. Сампсониевский пр., д. 29А, пом. 52. Тел.: +78127037373. Дата изготовления: 04.2020.

Наименование: книжная продукция. Срок годности: не ограничен.

Налоговая льгота — общероссийский классификатор продукции ОК 034-2014,
58.11.12 — Книги печатные профессиональные, технические и научные.

Импортер в Беларусь: ООО «ПИТЕР М», 220020, РБ,
г. Минск, ул. Тимирязева, д. 121/3, к. 214, тел./факс: 208 80 01.

Подписано в печать 13.03.20. Формат 70х100/16. Бумага офсетная. Усл. п. л. 36,120. Тираж 1500. Заказ

Оглавление

ПРЕДИСЛОВИЕ	6
ОТ ИЗДАТЕЛЬСТВА.....	6
ЧАСТЬ I. МАШИНЫ ОПЫТА.....	7
1. МАШИНЫ ОПЫТА	11
ЧАСТЬ II. ИСКУССТВО СОЗДАНИЯ ИГРЫ	59
2. ЭЛЕГАНТНОСТЬ	62
3. НАВЫК ИГРОКА.....	78
4. НАРРАТИВ.....	100
5. РЕШЕНИЯ	145
6. БАЛАНС	188
7. МНОГОПОЛЬЗОВАТЕЛЬСКАЯ ИГРА	211
8. МОТИВАЦИЯ И УДОВЛЕТВОРЕНИЕ.....	244
9. ИНТЕРФЕЙС.....	265
10. РЫНОК.....	297
ЧАСТЬ III. ПРОЦЕСС	321
11. ПЛАНИРОВАНИЕ И ИТЕРАЦИЯ	330
12. СОЗДАНИЕ ЗНАНИЙ	371
13. ЗАВИСИМОСТИ	386
14. УПРАВЛЕНИЕ	403
15. МОТИВАЦИЯ.....	416
16. СЛОЖНЫЕ РЕШЕНИЯ	431
17. ЦЕННОСТИ	439
ЭНДШПИЛЬ	444
РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА.....	446
ОТВЕТЫ НА ВОПРОСЫ	448
ОБ АВТОРЕ	448
ИСТОЧНИКИ.....	WWW.PITER.COM

Предисловие

Примечание

К сожалению, в английском языке не существует идеального гендерно-нейтрального варианта употребления местоимений. В этой книге я решил использовать местоимение «он», обращаясь ко всем читателям. Это сделано только для удобства чтения и лаконичности; местоимения мужского рода не относятся исключительно к мужчинам.

Обратная связь

Я люблю разговаривать о геймдизайне. Если у вас появятся замечания или вопросы по книге, напишите мне на электронную почту tynan.sylvester@gmail.com или свяжитесь со мной через веб-сайт tynansylvester.com.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете подробную информацию о наших книгах.

ЧАСТЬ I

Машины опыта

Жил был Изобретатель. Он придумал для людей замечательные вещи — мельницы для зерна, ткацкие станки и много всякой всячины. Горожане его любили.

Но Изобретатель старел, и у него оставалось время только на одно изобретение. Поэтому он решил, что его последняя работа будет особенной. Она будет предназначена не для передвижения, обогрева или для расчетов — она будет для счастья.

Горожане ничего не поняли, но доверились ему. Раньше он никогда их не подводил. Изобретатель ушел в свой замок и принялся за работу.

Шли годы. Сначала горожане терпеливо ждали. Затем стали сомневаться. А потом и вовсе разошлись.

«Где твоё изобретение?» — спросили они.

«Все очень долго».

«Мы устали ждать».

«Он обманывает нас».

«Мы должны убить его».

Как только толпа прибыла в замок, ворота открылись и вышел Изобретатель.

«Готово!» — заявил он. Толпа на мгновение успокоилась, и он повел их внутрь.

Но внутри так и не нашлось никакого замысловатого механизма — лишь стол с кучей карточек, книжечек и крошечных деревяшек.

«Где машина? — спросил предводитель толпы, готовясь к наступлению. — Где машина счастья?»

«Вот она, — сказал Изобретатель, указывая на игральные кости, правила и игровое поле. — Садитесь, будем играть».

Сами по себе шахматные фигуры — это просто крошечные декоративные скульптурки. Но если перемещать их в соответствии со специальным набором правил, они оживают. Они создают напряженную атмосферу на турнирах. Благодаря им в газетах появляется целый мир шахматных задач. Шахматы помогают найти друзей, рассказывать истории и учат тому, что вы больше нигде не узнаете.

Но для игры подойдет не любой набор правил. По сути большинство правил для фигур на доске не делают ничего из этих замечательных вещей. Многие из них вырождаются в простые, повторяющиеся схемы, поскольку игроки снова и снова используют одни и те же выигрышные стратегии. Бывают и такие правила, которые очень трудно понять. При этом им еще и так трудно следовать, что игра превращается в сложную математическую задачу.

Уникальная ценность шахмат заключается в том, как они задают идеальный ритм головоломки и решения, напряжения и облегчения. Сами по себе фигуры или доска не имеют смысла, вся ценность заключается в геймдизайне — системе правил, которая определяет ход игры. Задача геймдизайнера заключается в разработке систем правил, которые позволяют получить такой результат.

Разобраться в том, как реализовать задачи в геймдизайне, непросто. Что бы вы изменили в шахматах, чтобы можно было легче научиться в них играть? Что бы вы изменили, чтобы сделать их популярным видом спорта? Устранили бы вы часто повторяющиеся начальные ходы? Нужно ли добавить или удалить какие-нибудь фигуры? Изменить ли правила ходов? Изменить игровое поле, добавить новые возможности, изменить графику, добавить историю или сделать игру в реальном времени?

Ответы на эти вопросы можно найти в профессии геймдизайнера. Она помогает понять, как создавать легкие или сложные игры, или и те и другие. Геймдизайн помогает нам учить игроков, не перегружая их лишней информацией. Он говорит нам, как объединить истории и правила в цельную смысловую систему. Первая часть этой книги посвящается этому искусству.

ГЕЙМДИЗАЙН — это не код, не графика или звук. Это не создание игровых фигурок или раскрашивание игрового поля. Геймдизайн — это разработка правил, благодаря которым эти части оживают.

Впрочем, ни один разработчик не может волшебным образом знать ответы на все вопросы (хотя есть и такие, кто пытается), даже если это лучшая работа в мире. Вот почему вторая часть этой книги посвящена рутинному процессу геймдизайна. Настоящие геймдизайнеры не просто знают ответы — они знают, как найти их, используя тестирование, планирование и анализ. Знание процесса показывает, когда тестировать и как, когда планировать, а когда нет, как работать с другими и избегать творческих тупиков.

Некоторые люди беспокоятся, что анализ в геймдизайне «вынимает душу», что понимание принципов этого ремесла отни-

Работа геймдизайнера не определяет цель игры. Она только показывает нам, как ее достичь.

мает творческий потенциал в работе. Понимание профессиональной области геймдизайна не означает слепого следования правилам с целью снова и снова получать один и тот же результат. Это означает понимание баланса преимуществ и недостатков в каждом проектном решении. Если игра «не зашла», это редко связано с тем, что разработчик сделал неправильный логический выбор.

Скорее он просто не выбрал оптимальный вариант. Таким образом, разработчик, понимающий профессиональную область, похож на инженера, понимающего законы физики: законы Ньютона не определяют, создаем ли мы мотор для лодки или ракету «Сатурн-5», но в обоих случаях эти законы одинаково необходимы.

Представьте себе лучшую игру, в которую вы когда-либо играли, но только сделанную еще лучше, где каждая эмоция сильнее, темп — совершеннее, задумка — связная и детальная. Нет такой игры, которую нельзя было бы сделать лучше, используя те же ресурсы. Мы никогда не создадим идеальную игру, но, изучая геймдизайн, мы сможем максимально приблизить любую игру к совершенству.

Машины опыта

Механика и события

Механика — это правила, по которым работает игра. Кнопка А, которая заставляет Марио прыгать, является примером работы механики. Таковы правила, по которым персонажи передвигаются со скоростью один метр в секунду, пешку можно съесть по диагонали, а игроки ходят по очереди.

В настольных играх механика описана в правилах. В видеоиграх она включена в компьютерный код. Но независимо от того, кто реализует механику — игрок или компьютер, она все равно остается механикой, так как определяет ход игры.

Игры включают в себя МЕХАНИКУ, которая определяет, как работает игра.

Событие — это то, что происходит во время игры. Марио ударяется об стену и отскакивает назад, пешка съедает ладью, мяч попадает в сетку, за что другая команда получает очко, — все это примеры событий.

Практически во всех других средствах развлечения события создаются напрямую. Сценарист, писатель-романист или хореограф задают каждое действие, движение и линию диалога в процессе работы. Их продукт — длинная серия предопределенных событий: сначала Люк встречается Оби-Вана, затем его родители умирают, после этого они нанимают Хана Соло и так далее. События разворачиваются одно за другим одинаково.

В процессе игры механика и игроки взаимодействуют, генерируя СОБЫТИЯ.

В играх все работает иначе. Мы не создаем события напрямую, а проектируем механику. Эта механика генерирует события во время игры.

Например, играя в Super Mario Galaxy, я однажды попытался заставить Марио перепрыгнуть через яму. Я замешкался, и Марио коснулся лавы. Его спина загорелась, и он взлетел, как полыхающая ракета, лицо персонажа исказилось от боли. Пока он летел по воздуху, я управлял игрой и вывел его на безопасное приземление. События здесь — это то, что Марио прыгал, в какой-то момент прыжок не удался, Марио задел лаву, загорелся, взлетел в воздух, закричал и снова оказался в безопасном месте. Механика, стоящая за этими событиями, — это действие кнопки прыжка, гравитация, физическое столкновение, реакция на столкновение с лавой в виде горящей спины и моя способность контролировать движение Марио в воздухе.

Диск с игрой Super Mario Galaxy не содержит каких-либо событий, которые я перечислил, — в нем только чистая механика. События — это результат моей игры и игровой механики. И эти события никогда не повторяются.

Геймдизайнеры не создают события. Они разрабатывают системы механики, которые генерируют события. Этот уровень абстракции является фундаментальным отличием между играми и большинством других средств. Это наша величайшая возможность и наша сложнейшая задача. Механика является основной причиной, в силу которой ход мыслей, заимствованный из других источников, так часто не работает в играх.

Главенство эмоций

Игра не может случайным образом генерировать повторяющуюся последовательность событий, поскольку большинство событий не имеют высокой степени важности. Чтобы игра была инте-

Событие должно
вызывать эмоции.

ресной, эти события должны вызвать у человека сильные эмоции. Если сгенерированные игрой события вызывают гордость, веселье, страх или ужас, такая игра способна увлечь игрока.

Игры должны вызывать эмоции, но это не значит, что любая игра должна вызывать хохот до потери пульса, заставлять кричать от ярости, расстраиваться или плакать. В повседневной речи люди часто используют слово «эмоция» для обозначения крайней

степени проявления чувств, таких как видимые ярость или горе. Впрочем, большая часть эмоций имеет большую глубину и скрытый характер.

Например, если вы сидите и читаете книгу, может казаться, что вы ничего не чувствуете. Но на самом деле вы испытываете поток крошечных эмоциональных импульсов. Эти импульсы могут быть вызваны чем угодно — мыслью об утраченной любви, непонятным словом на странице (например, «Брандашмыг!») или же хмурым выражением лица проходящего мимо незнакомца. Такие эмоции продолжаются доли секунды и обычно находятся вне человеческого сознания. Но они всегда присутствуют, проявляясь в виде стимулов и мыслей.

Ценные эмоции игры могут быть едва уловимы. Обычно они вызываются скрыто, чтобы игрок не осознавал их воздействие.

События в игре вызывают такие скрытые эмоции. Даже мелкая неудача вызывает импульс разочарования и заставляет на мгновение стиснуть зубы. Момент неопределенности заставляет поволноваться, дыхание замирает. Когда игрок, который играет с вами, признает вас, вы чувствуете облегчение.

Эти крошечные чувства трудноуловимы. Было бы не совсем корректно говорить, например, что сегодня вы счастливы, вам грустно или скучно. Эти слова описывают значительные изменения в самых очевидных чувствах. Микроэмоции — это такие эмоции, которые составляют канву игры и ежесекундно меняются. Это вдвойне справедливо, если играешь в хорошую игру.

Представьте, что вы играете в шахматы с незнакомцем. Вы проигрываете и не видите возможностей для хорошего хода, поэтому чувствуете стресс и напряжение. По мере изучения доски напряжение крепнет. И вдруг вы замечаете выход: если сделать ход конем назад, то можно прикрыть своего короля и одновременно поставить шах вражескому! Тотчас же следует безмолвный облегченный выдох, сопровождаемый чувством выполненного долга. Вы ходите конем, как и задумали, и противник морщится, понимая, что произошло. Видя все это, вы чувствуете свое превосходство. Оппонент начинает думать над своим ходом. Сияя от удовольствия и наслаждаясь временной победой, вы вдруг замечаете слабое место своего решения.

Если соперник сделает ход слоном, то сможет гарантированно захватить вашего коня. В то же время для соперника этот ход может быть не очевиден. Заметил ли он его или нет — неизвестно. Удовлетворение превращается в неопределенность. Время тянется, пока вы пытаетесь сохранить невозмутимое выражение лица. Наконец, противник ходит пешкой. Чувство облегчения накрывает вас с головой снова, еще сильнее, чем до того, когда вы поняли, что успех гарантирован.

Наблюдая эту игру со стороны, нельзя сказать, что происходит что-нибудь особенное. Два человека сидят за столом с напряженными лицами и тихо двигают пластиковые фигурки по доске. Даже сами игроки не осознают в полной мере всего того, что на самом деле чувствуют. Но мы понимаем, что у них внутри бушуют эмоции. И они будут играть еще и еще, чтобы снова погрузиться в этот меняющийся мир эмоций.

Выявление и понимание тонких эмоций — это навык дизайнера.

Уловить такие тонкие чувства сложно. Для этого требуются усилия и практика. Можете ли вы точно назвать секунду, когда игра начала надоедать? Можете ли вы почувствовать, как непроизвольно улыбнулись в ответ на шутку, которая, как вам показалось, вовсе не смешная?

Большинство людей могут игнорировать подобные чувства, но для геймдизайнера это непозволительно. Подобно тому как опытный шеф-повар может понять, что входит в основу сложного блюда, а музыкант — уловить аккорды, такт и ритм музыкального произведения, которое исполняет оркестр, так и геймдизайнер должен чувствовать вспышку гнева, биение пульса от триумфа или всплеск отвращения. Ведь игра существует именно благодаря этим эмоциям. Именно поэтому игроки тратят свою энергию, время и деньги, чтобы перемещать фишки по доске или бросать мяч через кольцо.

Главенство эмоций — один из величайших непризнанных секретов геймдизайна. Спросите кого-нибудь его мнение об игре, и вам скажут, что думают о ней. Вам приведут логически обоснованные аргументы, почему игра хорошая или плохая. Но, как правило, логика — просто автоматическая рационализация глубинных эмоций. Что действительно важно, так это то, какие чувства у нас вызывает игра.

К сожалению, разговоры о геймдизайне до сих пор часто привязаны к понятию веселья, как будто между развлечением и геймдизайном существует какая-то внутренняя связь. Взаимосвязь действительно есть, но тут дело в каламбуре истории, а не в реальности.

Веселье — это эмоция, чувство несдерживаемого восторга, который вы испытываете на американских горках или играя с друзьями в футбол. Это приятные эмоции и достойная цель геймдизайна. Однако она далеко не единственная. Мы остановились на ней только затем, чтобы понять, как появились игры.

Эмоции, которые вызывает игра, не ограничиваются «весельем».

Такой профессии, как геймдизайнер, не существовало, а игры представляли собой передававшиеся из поколения в поколение артефакты народной культуры, которыми в основном пользовались дети. Если взрослые и играли, то, как правило, это было короткой передышкой от суровой, пресной жизни. В примитивной среде для описания хороших игр термин «веселье» подходит как нельзя лучше.

В современном мире же существует множество профессиональных геймдизайнеров и игроков с постоянно меняющимися эмоциональными аппетитами. И чтобы делать по-настоящему качественные игры, дизайнерам нужен не один критерий. Слово «веселье» не может описать всю гамму, силу и нюансы эмоций, которые возникают в процессе игры. Аналогично тому, как если бы шеф-повар описывал каждое блюдо как «вкусное» или «пресное».

В играх можно делать не только веселые или легкомысленные вещи. Участие в жестоких соревнованиях может вызвать чувство триумфа. В каких-то играх используется повествование, чтобы вызвать сочувствие или удивление. Другие игры затягивают нас в темное пространство или подвергают жуткому психологическому террору. Doom, Super Mario 64, Street Fighter II, Half-Life, StarCraft, The Sims, DEFCON, System Shock 2, Deus Ex, World of Warcraft, Dwarf Fortress, Portal, Tetris, Braid, Katamari Damacy и S.T.A.L.K.E.R. — все эти игры вызывают сильные эмоции, но все они абсолютно разные. Напряженный экшн соревновательной

игры Street Fighter II, игра с элементами хоррора System Shock 2 и стратегия DEFCON — все они вызывают разные эмоции, но назвать их веселыми нельзя.

Эмоциональные триггеры

Игровые механики взаимодействуют, генерируя события, которые, в свою очередь, вызывают у игроков эмоции. Но как именно события пробуждают эмоции? Какова связь между тем, что происходит в игре, и тем импульсом радости или грусти, который появляется в ответной реакции?

Например, если вы стоите на краю скалы, врожденный инстинкт самосохранения чувствует ситуацию и приводит к появлению страха. Если вы смотрите на предполагаемого партнера, подсо-

знание анализирует его, начиная от физических особенностей и вплоть до репутации и его взаимоотношений с вами, а затем генерирует соответствующее чувство влечения, нейтралитета или отвращения. Каждая из этих ситуаций, вызывающих эмоции, называется *эмоциональным триггером*.

Существует бесконечное множество разных эмоциональных триггеров. Физическая опасность, изменения в отношениях или социальном статусе, обучение, закрепление, приобретение имущества, сексуальные стимулы, семья и безопасность, а также определенные типы естественной среды являются наиболее очевидными, но не единственными. Люди также реагируют на музыку, философские мысли, юмор и остроумие, а также на бесчисленные формы искусства. Некоторые из этих триггеров встроены в наши гены. Другие еще предстоит изучить. Большинство включает сложные взаимодействия между заданными условиями и природой человека.

Эмоциональные триггеры могут быть чрезвычайно сложными. Например, представьте детектива. Предчувствие возникает тогда, когда эмоциональное бессознательное раскрыло дело и отчаянно пытается сообщить о своих выводах. Внешне все выглядит так: детективу кажется, что есть подвох, но он не уверен, где именно.

Бессознательное постоянно анализирует ситуацию. При соблюдении определенных условий бессознательное вызывает эмоциональную реакцию.

ЭМОЦИОНАЛЬНЫЙ ТРИГГЕР — это какой-либо предмет или наблюдение, вызывающие эмоции.

При этом его бессознательное обрабатывает чрезвычайно сложный комплекс умозаключений и ассоциаций, настолько сложный, что бессознательное понимает ситуацию лучше, чем сам детектив. Наши эмоциональные триггеры могут быть настолько сложными, что мы даже не всегда можем их понять.

Эмоция и изменение

Принцип, лежащий в основе всех эмоциональных триггеров, — это изменение. Чтобы вызвать эмоции, событие должно сигнализировать о значительных изменениях в мире. Но не любое изменение вызовет эмоцию.

Например, астероид, врезавшийся в далекую планету, — это обыденное астрономическое явление. Астероид, врезавшийся в Землю, — самое катастрофичное событие, которое может случиться. Вся разница заключается в последствиях. В одном случае ничего существенного для человека не произошло. Другой случай несет неминуемую гибель.

[Жизнь/смерть] является примером *человеческой ценности*.

Человеческие ценности могут быть как в положительном, так и в нейтральном и отрицательном состоянии. Только изменения, которые «сдвигают» человеческие ценности между этими состояниями, эмоционально актуальны.

К некоторым примерам человеческих ценностей относятся следующие: [жизнь/смерть], [победа/поражение], [друг/случайный прохожий/враг], [богатство/бедность], [высокий/низкий статус], [вместе/в одиночку], [любовь/неопределенность/ненависть], [свобода/рабство], [опасность/безопасность], [знание/невежество], [специалист/неквалифицированный], [здоровый/больной] и [последователь/лидер]. События в играх могут сдвинуть все эти ценности, и не только.

В игре Minecraft на игроков каждую ночь нападают зомби. Когда игрок заканчивает строить форт, чтобы спрятаться, то чувствует облегчение, потому что находится в безопасности.

В игре Street Fighter II подросток начинает участвовать в турнирах. Сначала его легко побеждают. Но он не останавливается. Он

Чтобы вызвать эмоции, событие должно изменить некоторую ЧЕЛОВЕЧЕСКУЮ ЦЕННОСТЬ.

ЧЕЛОВЕЧЕСКАЯ ЦЕННОСТЬ — это сферы и явления, важные для людей, которые могут иметь несколько состояний.

продолжает практиковаться, поднимаясь выше по «лестнице». В конце концов, он выигрывает региональный турнир, затем национальный, а после и чемпионат мира. Это события, меняющие жизнь, потому что они представляют собой большие сдвиги от невежества к знаниям, от низкого статуса к высокому и от поражения к победе.

В игре World of Warcraft два игрока бьются против одного и того же монстра. Один приглашает другого вступить в гильдию. Незнакомец становится другом, а один превращается во «вместе».

В игре Half-Life персонаж игрока заперт в гигантской подземной лаборатории, полной монстров из другого измерения. Иногда он встречает других выживших — ученых и охранников, которые могут сопровождать его какое-то время. Обнаружение этих союзников и их потеря — эмоционально захватывающие события из-за перехода между состояниями [в одиночку/вместе] из одного в другое и обратно.

В некоторых случаях изменяющаяся человеческая ценность существует только внутри игры. В других случаях она может быть реальной. Например, азартные игры создают эмоции вокруг изменений реального материального состояния. Игра в кости представляет собой довольно скучный экшн — игроки снова и снова просто бросают кости. Но если на кону стоят деньги, каждый бросок становится событием, которое заставляет понервничать, поскольку предполагает сдвиг между бедностью и богатством.

Игры могут даже вызывать эмоции, угрожая игрокам физической расправой. Экспериментальная видеоигра PainStation работает точно так же, как и Pong, но она гораздо более эмоционально насыщена, потому что каждая неудача сопровождается автоматическим ударом током или шлепком по руке. Крошечный движущийся шарик на экране имеет большой эмоциональный вес, в особенности если он может физически наказать вас.

Рассмотрим случай потери пешки в шахматах. В начале игры это может быть не так важно. В этом случае последствия могут состоять в том, что у вас просто останется меньше фигур, а пешечная структура может стать хуже. Но в конце игры одна пешка может играть решающую роль и стоять на пути к победе или поражению.

Если вы неожиданно теряете пешку, которая защищала короля, то чувствуете смятение, потому что только что проиграли. Во всех случаях событие одно и то же, но его последствия разные — одно представляет собой небольшое неудобство, а другое прочит победу или поражение.

Даже события, которые сами по себе кажутся незначительными, могут вызывать сильные эмоции, если они имеют важные предпосылки. Рассмотрим разведку в стратегических играх. Разведка — это не более чем осмотр объекта. Вы ничего не создаете, ничего не разрушаете и ничего не перемещаете. Само по себе это почти не событие. Но разведка стратегически важного здания может преломить ход проигрышной игры, так как тот ключевой элемент информации может сформировать ядро новой стратегии, которая может привести к победе. Таким образом, в игре, полной сражений и кровопролития, самым эмоционально захватывающим моментом может быть всего лишь осмотр здания.

Эмоционально важным в событии является не само событие, а изменения в человеческих ценностях, которые подразумевает это событие. Чем важнее человеческая ценность и чем сильнее она меняется, тем больше это вызывает эмоций.

Существует огромное количество способов вызвать изменения важных человеческих ценностей в ответ на даже небольшие события. Например, в многопользовательском шутере Modern Warfare существует система стриков (несколько убийств подряд), награждающая игроков, которые убивают определенное количество врагов, а сами при этом не умирают. За три убийства подряд можно получить полезный беспилотный разведчик, который отображает врагов на радаре, за семь — авиаудар, а за одиннадцать — мощную атаку самолетом AC-130 с целым арсеналом. Эта схема работает, так как увеличивает последствия определенных убийств. Одиннадцатое убийство намного более значимо, чем первое, потому что влияет на состояние игры больше, чем первое. Сами эти два убийства могут быть абсолютно одинаковыми — например, стрелять по врагу, когда он бежит за угол, — но у них разный эмоциональный заряд, потому что разные последствия.

Эмоциональное бессознательное не просто реагирует на происходящее. Оно постоянно смотрит в будущее, отслеживая потенциальные угрозы и возможности для человека. Когда оно их находит, то сигнализирует об этом с помощью эмоций.

Эмоции не появляются просто в ответ на изменение. Они также появляются в ожидании изменений.

Еще раз представьте, что играете в Modern Warfare. Вы насчитали десять убийств. Вы знаете, что еще одно убийство даст вам бонус в виде самолета AC-130 и вы, скорее всего, выиграете. В этой ситуации небольшие локальные события, такие как смерть вашего персонажа или убийство одного врага, могут определить исход всей игры. Вы чувствуете напряжение, потому что знаете, что находитесь между двумя совершенно разными исходами игры. Все зависит от того, что происходит в этот момент. Вы испытываете эмоции не в отношении того, что уже произошло, а в отношении того, что может случиться. Этот тип интриги и называют напряженной игрой.

Но если бессознательное почувствует, что ситуация не является критической, то вам может стать скучно. Теперь представьте следующее: вы уже застрелили десяток врагов, однако на этот раз ваша команда намного опережает команду-соперника. Сам AC-130 будет иметь тот же эффект, но ситуация гораздо менее напряженная, чем раньше, ведь ваше следующее убийство или смерть фактически не определяют исход игры. Человеческая ценность [победы/поражения] уже привязана к победе, поэтому у нее нет возможности измениться. Если вы убьете кого-то еще и получите AC-130, то команда победит. Если вы умрете, то команда все равно победит.

Бессознательное постоянно уравнивает эти бесчисленные варианты потенциальных последствий и направляет наше сознательное внимание на наиболее выпуклые участки, а именно туда, где есть наибольший потенциальный сдвиг в человеческих ценностях. Когда бессознательное игрока улавливает потенциальный сдвиг в человеческих ценностях, игрок это почувствует.

Обретение новой информации эмоционально эквивалентно изменению.

С точки зрения эмоционального воздействия существует небольшая разница между изучением факта и фактом, ставшим реальностью, так как последствия и возможности одинаковы. Это эмоциональная разница между тем, что вы потеряли тысячи долларов, бросив кости, и пониманием того, что вы потеряли тысячу долларов, когда крупье переворачивает последнюю карту. Брошенные кости были

событием, а перевернутая карта — открытием, но сдвиг человеческих ценностей и возникающие эмоции остались прежними.

Подумайте об игре в жанре хоррор, в которой вы должны идти по коридору с несколькими дверьми. Вы знаете, что убийца находится за одной из них, но не знаете, за какой именно. Эта ситуация создает стандартную тревогу, потому что есть слабая вероятность того, что вы что-то узнаете, а последствия будут чрезвычайно важны (возможно, ваш персонаж умрет). А теперь представьте научно-фантастическую игру в жанре хоррор, в которой вы идете по коридору в окружении телепортов, из которых может появиться убийца. В одном случае убийца оставался там и был обнаружен за дверью. В другом — он телепортируется. Но две ситуации эмоционально эквивалентны.

Это означает, что игры могут менять человеческие ценности, отрицая и раскрывая информацию. В некоторых играх бывает трудно постоянно генерировать изменения в человеческих ценностях. Эти ситуации могут быть более интересными, если не раскрывать игрокам все карты, а структурировать информацию, чтобы создать неопределенность.

Черный ящик с эмоциями

Эмоция — цель геймдизайна. Это достаточно сложная задача, потому что отследить точное происхождение наших эмоций достаточно непросто.

Эмоции — это не самостоятельный выбор. Стоя на краю скалы, вы не сами решаете, испугаться вам или нет. Видя красивого человека, вы не производите логических умозаключений, решая, что он вам должен понравиться. Эмоциональные триггеры — это автоматические вычисления, выполняемые бессознательной частью сознания, подобные тем, которые помогают сохранять равновесие при ходьбе или узнавать лица. Даже если вы знаете, что чувствуете, то все равно не сможете спросить бессознательное, почему оно заставило вас чувствовать притяжение, отвращение, безмятежность или страх.

Мы не можем самостоятельно осознать логику наших эмоциональных триггеров.

Мост

Классическое научное исследование демонстрирует психологическую связь между эмоциями и их причинами.

Представьте, что вы — молодой человек, живущий в Ванкувере. Сейчас 1973 год. Вы идете по висячему мосту Капиано. Мост представляет собой смертельную опасность шириной полтора метра и длиной 137 метров. Он качается, как опасный канатный мост из старого приключенческого фильма. Глядя через край, в семидесяти метрах внизу вы видите острые скалы между деревьями.

На середине моста вы встречаете привлекательную женщину, которая просит ответить на несколько вопросов. Она ведет проект по психологии о влиянии живописной обстановки на творческое выражение. Первая страница содержит скучные вопросы об имени, возрасте и прочем. На второй странице вас просят написать короткий рассказ к рисунку. После того как вы закончили, женщина отрывает клочок от листа с вопросами, записывает свой номер телефона и имя и просит вас позвонить ей, если у вас остались какие-либо вопросы.

Эта женщина появилась там благодаря исследователю психологии Артуру Арону. Что на самом деле интересует Арона, так это то, насколько сексуально наполненной получилась ваша история и какова вероятность того, что вы перезвоните женщине и пригласите ее на свидание, по сравнению с теми испытуемыми, которые находились на другом, более безопасном мосту неподалеку. Когда человек находился на мосту, его ладони потели, а сердце билось быстрее. Вопрос заключался в том, будут ли люди связывать этот страх с сексуальным влечением к женщине?

И они связывали. Субъекты на более опасном мосту описали в своей истории значительно больше сексуальных образов и в четыре раза чаще перезванивали женщине, чем те, кто находился на безопасном мосту. Эти результаты сохранялись даже в ходе дальнейших исследований, в которых были устранены такие факторы, как самоотбор (вероятность того, что более смелые мужчины с большей вероятностью перейдут опасный мост и с большей вероятностью позвонят женщине).

Мужчины, которые перезвонили женщине, думали, что она привлекла их, так как при разговоре с ней их сердце билось сильнее. На самом деле их сердце колотилось из-за ощущения опасности. Но испытываемые не могли уловить эту разницу, потому что эмоции не сообщают об их истинных причинах.

Ошибка эмоциональной атрибуции

У мужчин в исследовании Арона не было врожденной способности отследить истинную причину своих эмоций, поэтому они приписали их самому важному: привлекательной женщине. Такая ошибка эмоциональной атрибуции происходит постоянно. Мы думаем, что чувствуем что-либо по одной причине, хотя на самом деле причина совершенно в другом.

Некоторые используют ошибку эмоциональной атрибуции для манипуляции. Например, в следующий раз, когда будете смотреть политический документальный фильм, внимательно понаблюдайте за тем, что происходит, если режиссер хочет выставить политика в неприглядном свете. Когда лицо политика появляется на экране, музыка превращается в зловещий гул и изображение становится тусклым, искажается и замедляется. Режиссер надеется, что чувство страха, вызванное музыкой и визуальными эффектами, будет спроецировано на политика и заставит людей его бояться, хотя на самом деле их пугает страшный аудиоряд.

Продюсеры ТВ-шоу делают то же самое. Например, в теледраме существует такой прием, который я называю Важный Момент Леонарда Козна. Он происходит в начале третьего акта шоу, когда все плохо и кажется, что вся надежда утеряна. Диалоги прекращаются, и звучит душевная или запоминающаяся песня, часто похожая на песни Леонарда Козна. Камера плавно движется по площадке, а голос за кадром обсуждает тему шоу. Зрители в задумчивости. Но они ошибочно приписывают эти чувства истории, хотя они на самом деле вызваны песней.

Пока одна часть сознания старательно решает, какие эмоции генерировать, другая усердно работает над придумыванием причин этих эмоций. Иногда эти причины

Даже при том, что мы не знаем истинную причину своих чувств, мы неосознанно логически объясняем свои эмоции. Часто эти предполагаемые причины ошибочны.

совпадают, но часто это совсем не так. Но независимо от того, правы мы или нет, мы верим в эти причины всем сердцем.

В одном из многочисленных исследований, посвященных изучению такого поведения, ученые установили в магазине образцы четырех пар чулок и спросили покупателей, какие из них лучше всего по качеству. Восемьдесят процентов указали на чулки справа. Когда респондентов спросили почему, они ссылались на цвет или текстуру. На самом деле все чулки были одинаковыми. Респонденты выбрали чулки из-за их расположения, а потом пытались это аргументировать. Ложь была непреднамеренной, и люди не знали, что пытаются найти рациональное объяснение. Но это было именно так.

Вот почему игроки, говоря об игровом опыте, почти всегда объясняют причины своих чувств, а не сами чувства. Они скажут: «Мне понравилось, что это было быстро» или «Мне не было весело потому, что волшебные персонажи были скучными». Истина этих утверждений — грубые эмоции, стоящие за словами «Мне понравилось» и «Это не весело». Но игроки автоматически додумали причины, по которым они чувствовали все именно так. У игроков нет прямой связи со своими эмоциональными механизмами, так как они не знают истинную природу своих чувств. Но они обладают способностью мгновенно рационализировать практически любое поведение или мнение.

Игра предоставляет сотни различных стимулов и решений и вызывает множественные эмоциональные реакции. Но какие эмоции вызывают разные элементы игры? Простого способа узнать это не существует.

Ошибка эмоциональной атрибуции затрудняет понимание того, как игры влияют на нас.

Представьте, что вы играете в файтинг против друга на местном турнире. Последний раунд, и вы идете вровень. Приближаясь к вам, враг уклоняется от энергетических зарядов. Достигнув расстояния удара, он наносит отвлекающий удар, надеясь, что вы его заблокируете. Вы разоблачаете его блеф и наносите разрушительный апперкот. Ваши чувства очевидны: вас переполняют волнение и напряжение, сердце бьется, напряженные пальцы на джойстике, крики зрителей, расширенные зрачки, предвкушение победы.

Но почему? Что вызвало каждое из этих чувств? Экзотические бойцовские персонажи? Эффектно смотрящиеся боевые приемы? Красивая графика? А может, конкуренция с другом, изменившая ваши взаимоотношения? Угроза упасть перед ним в грязь лицом? Или просто обычная сенсорная перегрузка от яркого света, мигающего на экране? В игре фоном звучала музыка жанра техно. Имело ли это какое-то значение? А может, важным фактором стал напряженный геймплей?

Каждый из этих факторов внес свой вклад в появившиеся эмоции. Но психологически человек не способен определить причину, отвечающую за конкретный эффект. Мы не можем этого сделать.

Следует подвергать сомнению неподготовленное эмоциональное обоснование. Если кто-то говорит, что ему не нравится игра из-за визуальных эффектов, истории или средств управления персонажем, не принимайте это за чистую монету. Не надейтесь понять, как игра влияет на игроков, просто взглянув на нее. Существуют способы частично разгадать эту головоломку. Чтобы увидеть некоторые результаты небольших изменений, можно использовать систематические методы, такие как тестирование игр или анализ статистических показателей. Но даже с этими методами, основанными на фактах, мы никогда не сможем полностью понять игру, потому что никогда не сможем наблюдать внутреннюю работу разума — даже собственного. Следует выделить эмоциональные триггеры, выдвигая теории как бы со стороны. Мы жрецы, пытающиеся интерпретировать волю непредсказуемого бога на основании затмений и кофейной гущи. И так же, как и жрецы, мы часто ошибаемся. Именно поэтому геймдизайн так сложен.

Базовые эмоциональные триггеры

Давайте разберем некоторые из наиболее распространенных эмоциональных триггеров.

Эмоции от обучения

Вспомните состояние, когда вдруг в голове наконец что-то щелкнуло. Глаза загораются, рот расплывается в улыбке, и прозрение озаряет лицо: «А-а! Точно!» Учиться всегда приятно.

У щенков существует инстинктивное стремление в игровой форме нападать друг на друга. Это выглядит забавно, но в осно-

Мы инстинктивно
стремимся овладеть
теми навыками,
которые помогли
нашим предкам раз-
множаться.

ве этой игры лежит крайне серьезная первопричина. Щенки, которые не устраивали игровые бои, вырастали слабыми бойцами. В связи с этим их репродуктивный потенциал был гораздо ниже, чем у «нападающих», и более слабые щенки не выживали. Для собак ранняя predisposition к веселому игровому бою — это стратегия выживания в бездушной игре под названием эволюция.

То же самое относится и к людям. У человека есть естественное желание учиться, но это желание носит избирательный характер.

Подумайте об играх, в которые играют дети. Они бегают и прыгают, осваивают кинетические навыки. Они играют в дочки-матери, чтобы освоить социальные роли. Дети устраивают шуточный бой палками или подушками, чтобы изучить боевые навыки, притворяются взрослыми: солдатами, светскими людьми или строителями. Они подражают взрослым и наслаждаются этим, ведь именно это помогло их предкам воспроизвести себе подобных.

По мере взросления мы приобретаем способность развивать более сложные интересы, делая все меньший акцент на репродуктивных целях. Например, я потратил годы на изучение гейм-

Чем важнее урок для
человеческой цен-
ности, тем сильнее
мы стремимся его
усвоить.

дизайна, но точно уверен, что ни у кого из моих предков не появлялись дети только потому, что он разработал более совершенную версию бросания камней. Но независимо от нашего возраста, уроки, которые влияют на нас больше всего, по-прежнему важны для человеческих ценностей — те, которые могут превратить одиночество в единение или бедность в богатство. Поэтому игры, которые учат игроков строить, общаться и сражаться, всегда будут иметь самое большое значение.

Если урок очевиден, то его не так сложно усвоить, потому что он и так достаточно ясен. Если это не очевидная идея, скрытая в какой-то сложной системе, ее изучение может стать опытом, изменяющим жизнь, поскольку она представляет собой уникальное прозрение, недоступное большинству.

Таким образом, задача геймдизайнера заключается в создании игровых систем с неочевидной иерархией, которую нужно разгадать. Это означает создание сложной игры, которая раскрывает уроки через уровни, каждый из которых основывается на предыдущем. Некоторые классические игры, такие как шахматы или покер, известны благодаря тому, что им обучаются на протяжении всей жизни. Простые игры, например крестики-нолики, являются их противоположностью.

Чем более сложным и менее очевидным является урок, тем больше удовольствия от его усвоения.

Лучшее усвоение происходит тогда, когда мы осмыслием определенный объем информации за короткий промежуток времени.

Осмысление — это опыт получения новой информации, которая вызывает цепную реакцию появления других знаний. Это происходит, если мы получаем последний элемент пазла, который встает на место и открывает всю картину в целом.

Например, в стратегиях вражеская база обнаруживается там, где вы несколькими минутами ранее видели вражеских рабочих. Вы ругаете себя и говорите: «Я должен был догадаться!» Или в шахматах противник делает ряд, казалось бы, бессмысленных ходов, которые позже оказываются изощренной ловушкой, в которую вы попали. Противник торжествующе улыбается, а вы мысленно говорите: «Я должен был догадаться!»

Игроки чувствуют ОСМЫСЛЕНИЕ, когда получают новую порцию информации, которая внезапно заставляет старые знания обретать смысл.

Эти ситуации не были внезапными. Им предшествовали подсказки, которые игрок чувствует и не может правильно истолковать, но потом приходит понимание. Мы думаем, что, возможно, в следующий раз, учитывая аналогичные обстоятельства, сможем спрогнозировать события, поэтому такие ситуации важны. Мы привлечем снайпера, если увидим малейшие движения, или отправим контратаку на вражескую базу, если увидим конструкторов. Мы сможем выжить или сменить победу на поражение.

Величайшие идеи приходят после длительного накопления информации, которая в один момент расставляет все на свои места.

Заранее составленные истории очень хорошо с этим справляются, так как могут точно контролировать то, что игрок изучает в каждой точке. Например, в Half-Life за всеми событиями игрок наблюдает глазами главного героя Гордона Фримена, ученого в очках и с дробовиком, пытающегося сбежать из гигантского исследовательского центра «Черная Меза».

Сражаясь с монстрами и военными, игрок неоднократно замечает сурового человека в костюме и с портфелем. Человек всегда исчезает непосредственно перед тем, как игрок сможет добраться до него, иногда кажется, что он просто телепортируется. Лишь после финальной битвы Джи-мэн наконец представляется и объясняет, что на самом деле произошло на «Черной Мезе».

Этот тип накопления знаний может появиться и в игровой механике. Головоломки — классический пример. В лучших играх-головоломках игрок узнает огромное количество информации о головоломке, до того как ее поймет. Он определяет движение всех частей, а также то, как они взаимосвязаны. Он может биться над решением в течение 20 минут или дольше, пытаясь собрать ее в голове. Когда он наконец соображает, в чем дело, все эти, казалось бы, случайные компоненты сразу становятся понятными, и игрок говорит: «Ага, я понял!»

Эмоции от эволюции персонажа

Люди обладают эмпатией. Посмотрите, как кто-то улыбается, и вы, вероятно, улыбнетесь ему в ответ. Посмотрите, как кто-то страдает от боли, и тоже почувствуете нечто подобное. Мы отражаем эмоции, которые чувствуем в других.

Этот эмоциональный триггер является излюбленным приемом сценаристов и писателей. И, подобно им, геймдизайнеры тоже могут предопределять эволюцию персонажа. Мы можем написать историю для игры и настроить ее так, чтобы каждый раз сюжет развивался одинаково. Это хорошо понятный и традиционный метод провоцирования эмоций, и он может быть достаточно эффективным.

Но у игр присутствует другой способ создания арки персонажей: мы можем сделать так, чтобы игра моментально генерировала

их. Например, в Left 4 Dead трое выживших после зомби-апокалипсиса в пределах безопасного расстояния наблюдают за тем, как союзник медленно истекает кровью, пока рядом прячутся монстры. В The Sims муж изменяет жене с женщиной помладше и попадает на измене. В Dwarf Fortress гном, разочарованный отсутствием пива, неистовствует, убивая трех шахтеров перед тем, как его усыпляют. Все эти события произошли в этих играх, но ни одно из них не было создано непосредственно дизайнерами. Они возникли во время игры из-за взаимодействия игровой механики.

Эволюция персонажей также утоляет особый вид голода к знаниям: мы любим узнавать о наших конкурентах. Нас особо интересуют проблемы других, потому что только во время конфликта раскрываются внутренние ценности и способности человека. Чем больше трудности, с которыми они сталкиваются, тем глубже мы видим их истинную природу. Нам скучно, когда герою приходится выбирать между обезжиренным и цельным молоком. Мы заставляем его выбирать между жизнью его жены и своей собственной и смотрим, как он проявит себя.

Эмоции от испытаний

Испытания на мастерство и силу создают эмоции разными способами. Когда мы боремся за них, мы входим в приятное состояние фокуса. Если мы справляемся, то чувствуем себя энергичными, деятельными и всесильными. Даже в случае поражения появляется желание попробовать еще раз и добиться большего успеха, если есть предчувствие успеха.

Борьба настолько тесно связана с играми, что ее часто заслуженно считают важной частью этой среды. Она входит во многие общие определения игр. Но несмотря на то, что это мощный и гибкий метод, борьба остается всего лишь еще одним эмоциональным триггером и не является обязательной частью любого геймдизайна.

The Sims, Minecraft, Snakes and Ladders, Dear Esther, рулетка — все эти игры вызывают сильные эмоции, а игроки при этом не стремятся к достижению поставленной цели.

Тем не менее борьба все еще является важной составляющей большинства игровых проектов, поэтому большая часть этой книги посвящена именно ей.

Эмоции от социального взаимодействия

Игра в мяч достаточно бессмысленная. Трудно понять, зачем кто-то вообще будет в нее играть. Игроки просто бросают мяч один другому. Не происходит смены никаких человеческих ценностей, нет персонажей, и никто особо не учится. Но мы продолжаем играть. Почему?

Ответ лежит за пределами самой игры. Вспомните момент из сериала «Предоставьте это Биверу»¹, когда папа и сын бросают друг другу бейсбольный мяч. Эти двое ни во что не играют, им нравится просто бросать мяч. Они придумали такую уловку, чтобы создать предлог, который позволит им долго оставаться вместе и разговаривать друг с другом. Им нужна подобная игра, ведь долгие разговоры по душам между отцом и маленьким сыном могут выглядеть странно. Такая игра дает повод собраться вместе и создает вид некоей деятельности, что устраняет этот барьер. Говоря языком геймеров, тот факт, что игра в мяч простая и бессмысленная, не баг, а фича. Сложная игра просто помешает разговору.

Игра в мяч является основной формой социально ориентированной игры, так как в ней почти нет эмоционального содержания. Но в большинстве социальных игр используются специфические игровые события для управления социальными взаимодействиями. Один игрок побеждает другого либо два игрока создают или изучают что-либо совместными усилиями, и вокруг этих событий возникают социальные взаимодействия. Выиграть в шахматы у компьютера — это не то же самое, что выиграть у человека, даже если игра развивается аналогичным образом, потому что победа над человеком добавляет еще один эмоционально значимый социальный смысл.

¹ Классический американский семейный сериал, который выходил в эфир в 1957–1963 годах. — *Примеч. ред.*

Рассмотрим пример хвастовства. Эмоции некоторых людей вознаграждают их за то, что они хвастаются, даже если делают это в интернете. Представьте игру Counter-Strike, в которой вы остаетесь последним из команды в раунде. Все союзники наблюдают за вами, надеясь, что вы выполните задачу и выиграете раунд. Любое мастерски выполненное действие, которое вы предпринимаете, приобретает еще один смысл, так как оно укрепляет доверие и репутацию, которые вы создали среди своих союзников. Любая ошибка, которую вы допускаете, имеет обратный эффект. Эта ситуация создает острое напряжение, потому что социальный статус висит на волоске.

Игры дают не только чувство превосходства, но и захватывающее разнообразие других социальных взаимодействий. Выстраивание доверительных отношений и их разрушение, шутки, победа над незнакомцами, спасение друзей и совместное решение задач — все это обычные виды социального опыта, воплощенные в играх. В игровой механике присутствуют тысячи вариаций, которые генерируют социальные ситуации. В любом случае, социальное взаимодействие работает, если меняет какую-либо социальную человеческую ценность — незнакомец становится другом, низкий статус меняется на высокий и так далее.

StarCraft и Halo: Reach располагают системами повторного просмотра, которые позволяют игрокам сохранять, пересматривать и делиться своими величайшими победами. Skate имеет систему для обмена видеогеймплея, что позволяет оценить его сообществу игроков. Игры в социальных сетях, например Farmville, позволяют игрокам отправлять друг другу подарки или ресурсы, которые помогают в достижении целей. The Sims давала игрокам возможность делиться фотографиями из фотоальбомов своих симов. Super Mario Galaxy позволяет одному игроку управлять Марио, в то время как другой с помощью джойстика ловит звездочки на экране. Kane & Lynch позволяет двум игрокам пережить мрачную криминальную историю вместе. В некотором смысле эта игра — это шаг в сторону еще большей игры под названием «жизнь». Отец, который предлагает поиграть в мяч, надеется наладить контакт со своим сыном; игра сама по себе для него менее важна, чем ее применение в жизни. Мы играем в игры

с алкоголем, чтобы доказать свою зрелость. Берем друг друга «на слабо», чтобы показать бесстрашие. В старшей школе парень играет в «бутылочку» не из-за интереса к механике вероятности быть выбранным; он просто знает, что сможет поцеловать симпатичную девочку.

Эмоции от приобретения

Если мы находим под подушкой доллар, то от радости у нас учащается пульс. Мы жаждем получить высокооплачиваемую работу при возможности ничего не делать. Выиграв в лотерею, люди кричат и плачут. Независимо от формы, получение богатства — это наслаждение.

В азартных играх эту реакцию можно получить наравне с реальным богатством. Но вызвать эту эмоцию могут даже те игры, в которых нет реальных денег. Они создают фиктивные системы обогащения и приобретения, а затем предоставляют игрокам богатство в этой системе. Фейковая награда по-прежнему вызывает чувство приобретения.

Хорошим примером могут послужить ролевые игры в стиле экшн, например Diablo III. Игрок бродит по случайно сгенерированным подземельям, убивая бесконечный поток монстров. У каждого побежденного демона, зомби и скелета остается небольшое количество золота, магического оружия или доспехи. Каждая золотая монета и меч увеличивают силу персонажа. Игрок получает эти награды настолько часто, что он постоянно остается в большом плюсе. В игре присутствует история, аудиовизуальные материалы, персонажи и задачи, которые необходимо решить, но ничто из перечисленного не является основным эмоциональным стимулом. В сути Diablo III лежит чувство непрерывного обогащения.

Эмоции от музыки

Музыка — это мощный и многофункциональный инструмент для генерации эмоций. Музыка комбинируется с опытом, поэтому широко используется во многих медиа. В боевиках звучит музыка, накаляющая атмосферу, в ночных клубах — чувственная

музыка, а в дневных ток-шоу крутят грустные или торжествующие песни, чтобы подчеркнуть любые события, о которых идет речь. Игры делают то же самое, добавляя музыку разнообразных стилей.

Музыка — это удивительно тонкий инструмент, он работает даже лучше, чем большинство эмоциональных триггеров. Никто никогда не отдает ей должное, потому что во время игры сознательно никто не обращает на нее внимания. Но даже если сознание ее игнорирует, подсознание превращает в постоянный поток чувств. Вы можете объяснить это тем, что музыку легко отделить от остального геймплея. Послушайте саундтрек к игре, и почувствуете многое из того, что чувствовали во время игры. Поиграйте в тишине — и будете удивлены ощущением пустоты.

Немузыкальные звуки также вызывают эмоции. Скрежет металла заставляет чувствовать сильное напряжение и беспокойство. Звук сердцебиения подчеркивает ожидание того, что что-то должно произойти. Звуки дождя вызывают умиротворение. Вой сирены на вечеринке звучит глупо. Звук хлюпающей жидкости вызывает чувство отвращения. Наложение этих звуков на другие события может подчеркнуть или противопоставить эмоцию. Но будьте осторожны — при чрезмерном использовании такие трюки могут легко привести к обратному эффекту.

Эмоции от зрелища

Звездный разрушитель врзается в Звезду Смерти! Супербоец в замедленной съемке уворачивается от приближающейся ракеты! Автоцистерна раскалывается на две части и взрывается!

Зрелищная картинка может вызвать мгновенный эмоциональный подъем. К сожалению, овчинка практически не стоит выделки. Несмотря на то что реализация таких эффектов стоит дорого, с точки зрения творчества они достаточно просты. Для создания других эмоциональных механизмов, таких как развитие персонажей, общение и обучение, геймдизайнеры должны создавать взаимосвязанные структуры механики или персонажей. Все, что необходимо для зрелища, — это взрывы чего-то большого. В результате такие спецэффекты часто используются студиями

с большим бюджетом и запасом творческой энергии. В худших случаях зрелища используются настолько необоснованно, что вытесняют не такие заметные, но более глубокие источники эмоций.

Зрелищная картинка работает, только если усиливает уже имеющиеся эмоции. Если игрок преодолел череду быстро возникающих угроз и достиг своей цели, неистово сжимая джойстик, было бы уместным завершить все действие красивым и эффектным взрывом. Это сработает, потому что усилит чувство облегчения и достижение игроком, выигравшим битву, своей цели. Аналогичный взрыв, который будет раздаваться снова и снова вне контекста любой миссии, оставит игроков равнодушными.

Эмоции от красоты

Закат над океаном. Здоровый смеющийся ребенок. Шедевр живописи. На первый взгляд между этими вещами нет ничего общего. Но все они прекрасны. Красота заключается не в какой-то особенности, а в том, как что-то влияет на нас. Явление прекрасно, если вызывает приятные ощущения.

Игры полны возможностей для создания красоты. Персонаж может быть представлен в мельчайших деталях и двигаться необычайно грациозно. Мир можно нарисовать в хорошо подобранной цветовой композиции. И красота не ограничивается видеоиграми — подумайте о красоте хорошо сделанных шахматных фигур или рисунках на «волшебных» картах.

Так же как и зрелище, красота не бесплатна, и дело не только во времени и художественном мастерстве, которые для этого требуются. Эмоции красоты не всегда сочетаются с игрой. В особенности если речь идет о мрачной игре, в которой присутствует ужас или беспокойство. В таком случае красота столкнется с основной «эстетикой» игры. Красота может добавить аудиовизуальный шум, который затруднит понимание игры и взаимодействие с ней.

Как и в случае со зрелищной картинкой, в современном гейм-дизайне наблюдается тенденция сильно «украшать» любую

ситуацию. Но лучше всего красота работает в том случае, если направлена на конкретную цель, а не бездумна.

Эмоции от окружающей среды

Луга не похожи на влажные джунгли, вызывающие приступ клаустрофобии, которые, в свою очередь, не похожи на арктическую тундру. И эти чувства меняются в зависимости от времени и сезона: зима не похожа на лето, ночь — на день, дождь — на ясное небо.

Доказано, что эти реакции частично врожденные. Психологи выявили, что дети американцев, которым показали фотографии различных местностей, предпочли бы жить в саваннах, даже если никогда там не были. Эти эмоции могут отражать требования, которые диктует нам эволюция, — искать места, где племя сможет процветать: плодородные, не слишком жаркие или холодные, не слишком открытые или заросшие. Идеальная среда для доисторических людей — это открытые поля с участками леса и источниками воды. Поэтому когда мы находим такое место, то чувствуем себя довольными и спокойными. Эта эмоциональная реакция зовет нас туда, где мы сможем лучше всего воспроизвести себе подобных.

У людей также появились предпочтения в выборе окружающей среды. Нам больше нравятся условия, в которых мы выросли. Маленькие американцы любят саванны, а взрослым нравятся хвойные и лиственные леса, потому что напоминают большую часть США. И ни один американец, ни в одной группе респондентов не захотел жить в пустыне или тропическом лесу.

Эмоции, обусловленные окружающей средой, обычно сильные и разнообразные. Чтобы подчеркнуть чувства, начиная от депрессии и заканчивая головокружительным триумфом, в играх задействуются обстановка, погода и сезон.

Heavy Rain. Эта приключенческая головоломка про мужчину, потерявшего сына. В первых нескольких сценах мир представлен в ярких и солнечных красках. Но после исчезновения мальчика остальная часть игры проходит под ливнем и в основном ночью.

Этот бесконечный дождь придает всем событиям мрачный оттенок, подчеркивая тему потери и депрессии.

Half-Life. Геймплей начинается с того, что герой попадает в ловушку гигантской подземной лаборатории «Черная Меза». В течение первых 15 часов игры естественного света нет. Когда игрок наконец врывается сквозь дверь в залитую солнцем пустыню Нью-Мексико, появляется осязаемое чувство свободы и выполненного долга.

Metro 2033. Спустя два десятилетия после ядерной катастрофы группа выживших ведет борьбу за существование в московском метро. Внутри везде темно, но это их дом. Люди работают, торгуют, слушают музыку, выпивают и смеются. Но на поверхности совсем другая история. Москва, представленная в игре, ужасает. Замороженные, разрушенные здания в гигантских кусках льда. Сам воздух токсичен, поэтому игрок должен иметь постоянно уменьшающийся запас фильтров для противогаза. Тысячи сосулек угрожающе свисают, как острые ловушки, раскачиваемые бушующим ветром. Все место действия пропитано безысходностью: никогда не светит солнце, не перестает дуть ветер, лед не тает и ничего не растет. Я никогда не забуду, что чувствовал, пробираясь сквозь обломки. Хотя большинство назвало бы *Metro 2033* шутером или RPG, позволю себе с этим не согласиться. Я думаю, что эта игра о понимании того, как место заставляет вас испытывать те или иные чувства.

Эмоции от современных технологий

Новые технологии — это круто. Первые несколько игр, которые выходят с любыми новыми графическими, анимационными или физическими технологиями, приносят эмоциональный подъем у определенных игроков только благодаря самой технологии.

Но за этот бонус часто приходится платить. Как это ни парадоксально, технологические достижения часто приводят к временному снижению качества дизайна игры. Это отчасти объясняется тем, что разработчики еще не знают, как использовать новую технологию наилучшим образом. Что еще более важно, обещание легких эмоций, которые дают технологии, снимает вопросы творчества. Таким образом, игра превращается в демонстрацию

технологических возможностей, потому что от нее не требуется чего-то большего, чтобы заинтересовать игроков. Некоторое время она все еще будет работать. Но ажиотаж вокруг технологий быстро проходит, а игра, которая зависит от них, уже не будет такой популярной через несколько лет.

Например, в середине 1990-х появилась возможность записывать полномасштабное видео на CD-ROM и воспроизводить его на ПК. В результате такого технологического прорыва появились некоторые из худших игр в истории. Они не стали ни полноценными фильмами, ни играми. Хотя эта катастрофа была обусловлена и многими другими факторами помимо технического фетишизма (такими, как бездумная кража творческих идей из фильмов), она случилась благодаря неоправданному доверию к технологиям.

Чтобы игра пользовалась успехом, в ней должны использоваться новые технологии, открывающие новые ситуации и возможности взаимодействия, которые раньше были недоступны. Например, Doom часто называют технически продвинутой игрой, потому что она была первым шутером от первого лица с трехмерной графикой. Но Doom не стала мега-хитом только из-за технологии. Новые технологии давали возможность открыть новый спектр опыта, появившегося благодаря дизайну. Doom — не первая игра с трехмерной графикой. Это была первая игра, в которой демоны выключали свет и бросались в комнату, когда вы брали предмет. Это была первая игра в жанре хоррор с погружением, в которой вы, пытаясь найти монстров, слышали их доносящиеся из темноты стоны. Это был первый многопользовательский шутер от первого лица. Все эти элементы зависели от технологий, но фактически являются достижениями геймдизайна, и нельзя приписывать заслуги одним лишь технологиям.

Эмоции от угроз

Некоторые вещи угрожают нашему виду на протяжении очень долгого времени — страх перед ними впечатан непосредственно в гены. Гнилая пища и антисанитария вызывают отвращение, чтобы уберечь от пищевого отравления. При виде ядовитых пауков и змей мы отскакиваем, потому что они опаснее, чем кажется на первый взгляд. Мы обходим стороной людей, которые

выглядят больными. Вид страшных ран вызывает выброс адреналина, чтобы подготовить нас к опасной ситуации. Игры могут вызвать такие же реакции. Просто кровь или пауки оказываются на экране. Все просто.

Даже слишком просто. В результате того, что ленивые режиссеры и геймдизайнеры годами пользуются подобными трюками, выброс адреналина при виде таких спецэффектов значительно снизился. Люди уже привыкли к подобной нелепой манере устрашения. Многие из нас автоматически отключают эти спецэффекты или даже смеются над ними. Чтобы создать действительно сильный страх или вызвать отвращение, уже недостаточно просто бездумно вывалить на экран кишки и прочие внутренности. Обычные люди еще могут испугаться, но чтобы привести в ужас геймеров, следует создавать угрозы, которые затронут более глубокий уровень сознания.

Эмоции от сексуальных сигналов

В игре может появиться часть обнаженного тела, красивое и соблазнительное лицо, и люди это заметят, так как генетически запрограммированы на то, чтобы обращать внимание на подобные вещи. Поскольку сексуальные сигналы настолько эффективны и просты в использовании, геймдизайнеры, маркетологи и режиссеры злоупотребляли ими. Вы можете добавить в игру практически обнаженного привлекательного персонажа, и некоторые игроки на это отреагируют. Как и в случае с угрозами, все просто.

Но у дешевых сексуальных сигналов присутствуют свои недостатки. Беспричинная сексуальность вредит атмосфере и правдоподобности серьезного повествования и раздражает множество потенциальных игроков (обычно тех, кому подобные сигналы неинтересны). В определенной игре, созданной для конкретной аудитории, это вполне приемлемо, но в более серьезной игре или игре для широкой целевой аудитории безвкусица все испортит.

История

Существуют игры, которые предполагают простую механику и ничего более. К ним относятся покер, футбол, шашки и видео-

игры вроде *Geometry Wars* или *Bejeweled*. В шашках фигуры — это всего лишь фишки. Они «ходят» по правилам, которые не применимы для других игр. Футбольный мяч — это просто мяч, а враг *Geometry Wars* — просто часть данных в компьютерной памяти, представленная на экране в абстрактной форме.

Подобные типы игр могут пользоваться успехом. Чистая игровая механика с абстрактными представлениями может спровоцировать чувство напряжения, сомнения, недоумения и триумфа. Они могут сдвигать ценности между победой и поражением, бедностью и богатством, невежеством и знанием.

Но большинство реальных игр не ограничиваются абстракцией. В них добавляют графику и звуки, чтобы помочь игрокам понять, что механика — это больше, чем искусственная система правил.

В одном из первых фильмов об Остине Пауэрсе Доктор Зло создает группу фемботов. Они выглядели как высокие блондинки в серебряных топиках и шортах. Но на самом деле за этой маской скрывались роботы (с пистолетами в сосках). Все мы знаем, что фемботы — это не более чем тщательно разработанные предметы, вроде автомобильного двигателя или тостера. Но их образ, представленный в виде человека, вызывает другую психологическую позицию. Они становятся больше, чем просто роботы, похожие на женщин. Они становятся женщинами, которые в действительности оказываются роботами.

Механика приобретает еще один уровень эмоционального значения, если к ней добавить историю.

На первый взгляд может показаться, что никакой разницы нет. Но для нашего подсознания и эмоциональных реакций это имеет огромное значение. Имея человеческую внешность, женщины становятся персонажами с разумом, желаниями и планами. Когда фемботы атакуют, они делают это не в результате переключения программ, а в результате злости. Когда они отступают, то не из-за запрограммированной стимул-реакции, а из-за страха. Если они преследуют человека, мы не считаем, что они выполняют алгоритм преследования, нам кажется, что они делают это осознанно. Все их действия обретают человеческий эмоциональный резонанс только потому, что поверх роботизированных скелетов виднеется кожа. А то, что эта кожа — всего лишь несколько миллиметров резины, не имеет значения.

Все игры — не более чем механика, точно так же, как фемботы Доктора Зло не более чем металл и резина. Марио — не мультяшный итальянский сантехник, он — цилиндр для столкновений, который бежит и сталкивается с предметами. Влюбленный подросток из The Sims на самом деле не влюблялся, просто игровое программное обеспечение обработало несколько битов в структуре данных.

«Одевая» механику в вымышленную одежду, мы придаем ей дополнительное эмоциональное значение. Если у персонажа заканчивается еда, мы не просто говорим, что у нас осталось мало ресурсов и игра скоро закончится. Мы говорим, что умираем от голода. Если союзник побежден, мы не просто тихо убираем его с экрана. Мы скорбим о нашем убитом друге. Мы знаем, что это всего лишь игра, но эта виртуальная реальность все равно создает некоторое эмоциональное эхо настоящего голода, горя или любви.

Наивные наблюдатели часто предполагают, что весь смысл игры заключается в выдумке. С этой точки зрения игры создают эмоции, вовлекая игрока в симуляцию до полного исчезновения ментального различия между игровым и реальным миром. Разработчик Эрик Циммерман назвал это глубоким заблуждением. Это заблуждение, потому что ни один игрок никогда не забудет, что играет в игру. Вымышленная упаковка не заменяет и не скрывает игровую механику; она добавляет второй слой значения к эмоциям, генерируемым одной только механикой.

История и механика

Механика может создавать ощущение напряжения, облегчения, триумфа и потери. Она может принести удовольствие от обучения или гордость от решения головоломки. Она может создавать социальные награды, позволяя нам побеждать незнакомцев или заводить друзей. Но одна только механика также ограничена в своем эмоциональном диапазоне. С помощью одной механики сложно передать юмор, трепет или погружение. А ведь без персонажей весь эмоциональный спектр, проходящий через эмпатию, практически недоступен.

История и механика
создают разные
эмоции.

Слой истории создает эмоции через персонажа, сюжет и мир. Мы смеемся и плачем, когда персонажи веселятся или страдают, и чувствуем потрясение или волнение, исследуя вымышленную вселенную. Впрочем, как и в случае с механикой, одна только история ограничена в своем диапазоне. С ее помощью нельзя создать соревнование, триумф и потерю. Благодаря ей мы не можем насладиться овладением навыка или создать социальные взаимодействия с реальными людьми.

Сочетание истории и механики позволяет нам объединять эмоции с обеих сторон. Но тут присутствует подвох.

Нарративы игр перегружены клише. Персонаж игрока страдает амнезией. Или он суперсолдат, способный убивать тысячи врагов. Враги — это монстры или вражеская армия, и они не испытывают ни страха, ни угрызений совести. Принцессы очень быстро снова и снова попадают в плен. Если ударить по бочке, она тут же взрывается. И никто никогда не ходит в туалет.

История и механика могут мешать друг другу.

Одним из худших клише является ящик. Кажется, что действие любой игры, в которую вы играете, будь то современный военный шутер или ролевая игра в жанре фэнтези, происходит в мире, усеянном бессмысленными ящиками. Этот вопрос стоит настолько остро, что еще в 2000 году юмористический сайт Old Man Murray создал систему обзора игр со шкалой «До ящика осталось ...» (Start to Crate (StC)). Идея заключалась в том, что чем дольше игроку не встречались ящики, тем тщательнее разработчики старались избегать клише и тем лучше была игра. Из 26 протестированных игр только в пяти играх ящик не появлялся более 10 секунд. В 10 случаях игра была провальной: первое, что видел игрок сразу же в начале, — ящик.

Ящики по-прежнему никуда не делись. Почему? Мы ничему не научились? Нет, это не потому, что все геймдизайнеры — идиоты. Причина, по которой ящики и другие подобные клише появляются снова и снова, заключается в том, что они просто оправдывают хорошую механику.

Как-то я занимался разработкой локации одного шутера, действие происходило в старинном театре, с рядами стульев непро-

тив сцены. После плейтеста я заметил, что игроки расстраиваются из-за появления вражеских снайперов. Пространство было настолько открыто, что игроки получали пулю в голову сразу же, как только выглядывали из укрытия. Дизайн задумки театра имел смысл, а его механика — нет. Чтобы иметь равные шансы со снайперами, требовался объект размером с микроавтобус в центре аудитории, который заблокировал бы линии обзора снайперов. Столкнувшись с такой проблемой, не всегда можно найти легкое решение, в особенности если история недоработана и к тому же у вас мало времени. Поэтому я сделал то, что должен был. Стыдливо опустив голову, я поставил пару ящиков в центре театра. Люди издевались над ящиками, и вполне заслуженно. Но бой пошел в нужном направлении.

Почти все игровые сюжетные клише основаны на механике. Персонаж игрока с амнезией оправдывает других персонажей, объясняя очевидные вещи о мире. Персонажи игроков часто являются суперсолдатами, потому что трудно создать вражеских стрелков, которые сражаются дольше нескольких секунд. Когда враги живут пять секунд, требуется, чтобы в игре их были тысячи. Будучи суперсолдатом, персонаж игрока действительно может одолеть батальоны врагов в одиночку. У врагов никогда не бывает сложных эмоций, потому что страх и сострадание непредсказуемы и их трудно изобразить. Игра становится проще и элегантнее с точки зрения механики, когда все бездумно сражаются друг с другом до смерти.

Рассмотрим одно базовое клише геймдизайна — физическое насилие. Существует так много игр, в основе которых лежит физический конфликт, что это уже утомляет. Однажды я попытался разорвать этот шаблон с помощью стратегии в реальном времени под названием Player League. Игрок управлял группой пикаперов в ночном клубе. Цель заключалась в том, чтобы собрать больше цыпочек, чем команды соперников. Нужно было не давать им возможности высказаться, продвигать себя различными способами и использовать нейтральных персонажей в своих интересах.

Игра не стала популярной. Главная причина, по которой проект провалился, заключалась в отсутствии четкого способа выразить

большую часть происходящего. Каждое игровое событие представляло собой взаимодействие людей, в результате которого один человек испытывал одну из нескольких возможных эмоций по отношению к другому.

Глядя на экран, было невозможно понять, что происходит: просто люди, которые просто разговаривают. Я не мог даже предположить текст их речи, потому что события могли происходить сотни раз в бесконечных комбинациях — слишком много, чтобы создавать реальную речь для каждой ситуации. В конце концов, я решил выделить взаимоотношения в цветные линии и фигуры, которые появлялись вокруг людей. В некоторой степени это сработало, но обозначение было условным и его нужно было запоминать, а внутриигровые события часто не требовали больших когнитивных усилий, потому что система не могла выразить большинство концепций, существующих в реальном обществе.

Подобные проблемы объясняют, почему существует так много игр о физическом конфликте. У стратегии о войне нет проблем, которые были в Player League. Изображение одного стреляющего субъекта в другого является четким и понятным и не требует абстрактных интерпретирующих символов. Люди просто наблюдают физическое насилие. Оно хорошо поддерживает механику, потому что его легко выучить и понять и к нему привыкают.

Кроме того, в боевых играх много ящиков.

Существует естественный компромисс между сосредоточением внимания на механике и сосредоточением внимания на истории. Акцент на механике позволяет дизайнеру создать идеально сбалансированную, четкую и сложную задачу. Но, скорее всего, будет очень трудно найти и придумать историю, которая бы отражала эту совершенную механику. Например, представьте, что вы пытаетесь придумать историю для игры в шахматы или покер. Это сложно, потому что эти игры мало на что похожи, кроме как на самих себя. В шахматах присутствует задумка, но она едва уловимая и бессмысленная — настоящие рыцари не всегда ходят на два квадрата вперед и один квадрат в бок. Покер ничем не похож на реальность или историю. Эти игры — отличные системы механики, но в них нет хорошей истории.

Поскольку история и механика мешают друг другу, разработчики многих игр предпочитают акцентировать внимание на одном, игнорируя при этом другое.

Кроме того, разработчик может сосредоточиться на истории, создавая красивый мир, полный персонажей с изысками и фантастических локаций. Но все эти нюансы истории затрудняют изменение механики. Вместо возможности изменить механику в любом направлении разработчик должен исправлять проблемы механики, внося только те изменения, которые не противоречат истории. Что касается игр в реальном мире, разработчик не может уменьшить гравитацию или дать возможность определенным персонажам не быть сожженными дотла, даже если это сделает задачу более сбалансированной. Механика страдает.

Именно вокруг этого конфликта истории и механики ведутся жаркие споры.

Исследователи видеоигр, или людологи (от лат. *ludus* — «игра»), утверждают, что игры получают свои самые важные свойства из механических систем и взаимодействий. Нарратологи утверждают, что механика — это просто структура, на которой можно разместить вымышленные элементы, которые действительно будут интересны игрокам. Это спор о природе и воспитании геймдизайнера, нашем сюжете и герое, нашем индивидуализме и коллективизме.

Впрочем, как и во всех подобных спорах, конфликт существует только на поверхности.

История и механика не должны конкурировать (хотя запросто могут), и ни одной из них не следует отдавать предпочтения (хотя это случается часто). При совместном использовании они могут дополнять и улучшать друг друга, давая такой эффект,

который невозможно получить, если использовать их отдельно. Невозможно получить великолепный геймдизайн, если поставить замечательную историю выше механики. Нужно объединение их в цельную систему эмоций. Вот почему большая часть геймдизайна — это не просто создание хорошо выдержанной задачи или красивого мира. Речь идет о выполнении каждого аспекта таким образом, чтобы они легко могли составить единое целое.

Вершина искусства
геймдизайна —
это объединение
совершенной механи-
ки и захватывающей
истории в целое.

Создавая опыт

Эмоции игры не стоят особняком. Они сливаются в интегрированный *опыт*.

Опыт — это совокупное выражение воздействий, которые игра оказывает на игрока на всем своем протяжении. Он трансформируется через формирование и отдачу, ожидание и результат. Мысль приводит к появлению эмоции, которая вызывает идею, идея вызывает реакцию, реакция вызывает обратную связь, приводя к другой мысли. И точно так же, как и хорошо приготовленная еда — больше, чем все ее ингредиенты, так и опыт — больше, чем совокупность ее психологических компонентов.

ОПЫТ — это поток эмоций, мыслей и решений в уме игрока.

Игровой опыт всегда смешан. Футболист даже в конце игры со счетом 2 : 1 надеется на победу, но боится пенальти. Ребенок, играющий в Super Mario Galaxy, радуется фанфарам в конце игры, в которой ему пришлось стискивать зубы от напряжения, выполняя сложные прыжки.

Игровой опыт также чрезвычайно разнообразен. Он может быть коротким и простым, как в файтинге, и максимум придется поволноваться всего 60 секунд.

Другой может быть длинным и сложным, как в нарративной ролевой игре с открытым миром, в которой акцент смещается от благоговейного исследования к сюжетной интриге, а игрок борется с волнением в течение ста часов игры.

Различные эмоции, собранные вместе, могут усиливать, трансформировать или даже разрушать друг друга. Давайте рассмотрим, как игры могут комбинировать эмоции и создавать впечатления.

Чистые эмоции

Чтобы максимально усилить какое-либо чувство, мы можем объединить несколько разных эмоциональных триггеров, которые вызывают одни и те же эмоции. Каждый триггер подобен

стартовому ускорителю, который толкает опыт дальше, к эмоциональному пику.

Например, чтобы сделать игру максимально захватывающей, традиционные экшн-аркады сочетают в себе динамичную музыку, рискованные ситуации, острый сюжет и социальную конкуренцию, подогреваемую выбросом адреналина. Возбуждение может быть вызвано любым из этих триггеров. В совокупности они повышают опыт до уровня, которого невозможно достичь, привлекая отдельные триггеры.

Совмещение

Совмещение — это сочетание разных, казалось бы, несовместимых чувств. Объединение чувств, которые обычно не сочетаются, может привести к необычным и иногда ценным результатам.

Я долго думал, что серия игр Gears of War от Epic Games была не чем иным, как шутером, в котором космодесантники просто побеждают монстров. И эта игра позиционирует себя именно так. Персонажи разрывают монстров на части с помощью штурмовой винтовки со встроенной бензопилой, размазывают побежденных врагов по земле, а воздух переполняется «накачанной тестостероном» атмосферой личного превосходства. Но познакомившись с этой серией игр ближе, я понял, что за чрезвычайно жестокой поверхностью скрывается второй, совершенно другой эмоциональный ингредиент, подобно тонкому аромату, который раскрывается не сразу. В Gears of War преобладает довольно мрачная атмосфера. Тем не менее события разворачиваются в руинах цивилизации необычайной красоты. Большинство случаев эволюции персонажей связаны с потерей, будь то любимый человек или славная прежняя жизнь. Даже реклама игры приобрела известность благодаря восхитительно тревожащему наложению песни Гари Жюля «Безумный мир» на визуальный образ компьютерной игры. Сочетая траур с сильным возбуждением, Gears of War становится больше, чем просто военный праздник крови.

Существует простой способ поэкспериментировать с совмещением. Просто замените музыку в игре музыкой, которая вызы-

вает совершенно другое чувство. Замените музыку в файтинге на написанный Моцартом фрагмент Реквиема — «Лакримоза». Добавьте тему песни Harry Days в игре Doom. Наложите треки Бритни Спирс на симулятор выживания вроде Dead Space 2. Получатся странные, тревожные или смешные результаты.

Антагонистические эмоции

Мороженое и пицца великолепны, если не есть их вместе. То же касается и некоторых эмоций, которые противоречат друг другу и поэтому не могут сосуществовать вместе, хотя при этом хорошо работают по отдельности.

Например, всеобщей социальной радости часто вредят, добавляя безжалостную конкуренцию, основанную на навыках. Интенсивная конкуренция приковывает все внимание игрока, так как он изо всех сил пытается выиграть, а вот веселые шутки с друзьями заставляют нас расслабиться. Именно из-за этого противоречия друзья, играющие в игры, результат которых зависит только от умений пользователей, часто соглашаются играть только «для удовольствия», тем самым отказываясь от повышения уровня навыков в игре, чтобы освободить место для социального опыта, который они действительно хотят получить.

Существует тонкая грань между совмещением и антагонистическими эмоциями. Иногда попытки совмещения бывают крайне неудачными, когда два чувства просто уничтожают друг друга. В других случаях то, что кажется антагонистической комбинацией, может вылиться в совершенно другое чувство.

Например, у моего друга был такой опыт в шутере: ближе к концу игры появлялась ключевая кат-сцена смерти главного персонажа. Это был трагический момент, которым разработчики явно пытались затронуть самые тонкие струны души игрока. Затем внимание смещалось обратно к геймплею. Поднимая какие-то боеприпасы, другой персонаж воскликнул: «Конфетка!» Мой друг расхохотался, потому что эмоция трагедии случайно оказалась слишком близко к эмоции мужественной самоуверенности. В результате получилась забавная смесь, которая вылилась в непреднамеренный смех и весьма веселый результат.

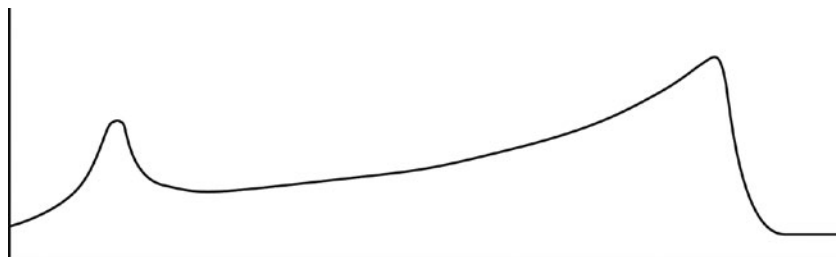
Атмосфера

Слово *атмосфера* описывает эмоциональный опыт, не сосредоточенный вокруг конкретных событий, а распространяющийся на все аспекты игры. Это эмоциональный фон, который мы замечаем только тогда, когда не происходит ничего более существенного. Остановитесь в процессе игры и просто прочувствуйте момент. Вы откроете для себя ее атмосферу. Некоторые игры преуменьшают эмоциональную нагрузку отдельных событий и создают плотную атмосферу, позволяющую игроку погрузиться в нее. Например, LIMBO, DEFCON и Flower — это атмосферные игры. Обычно атмосфера в таких играх спокойная и наталкивающая на размышления, хотя ей можно придать либо положительный, либо отрицательный оттенок. Flower — это хрупкий цветочный лепесток, парящий на ветру сквозь пушистые облака в пейзаже снов, а DEFCON — это наблюдение за тем, как ядерные ракеты стирают с лица земли миллионы людей. Для создания атмосферы в каждой из них присутствуют музыкальный фон и медленное взаимодействие, а также различные сюжеты для придания изюминки игровому опыту.

Смена эмоций

Любая эмоция надоедает, если длится слишком долго. Чтобы сохранить силу и свежесть, опыт должен со временем меняться.

Один из классических способов — смена ритма. Писатели годами использовали и изучали этот метод, вплоть до того момента, когда была разработана специальная формула ритма, которую используют снова и снова. Классическая кривая темпа начинается с крутого изгиба, переходит в подъем, растет, затем наконец достигает пика, а затем ниспадает. На графике это выглядит так:



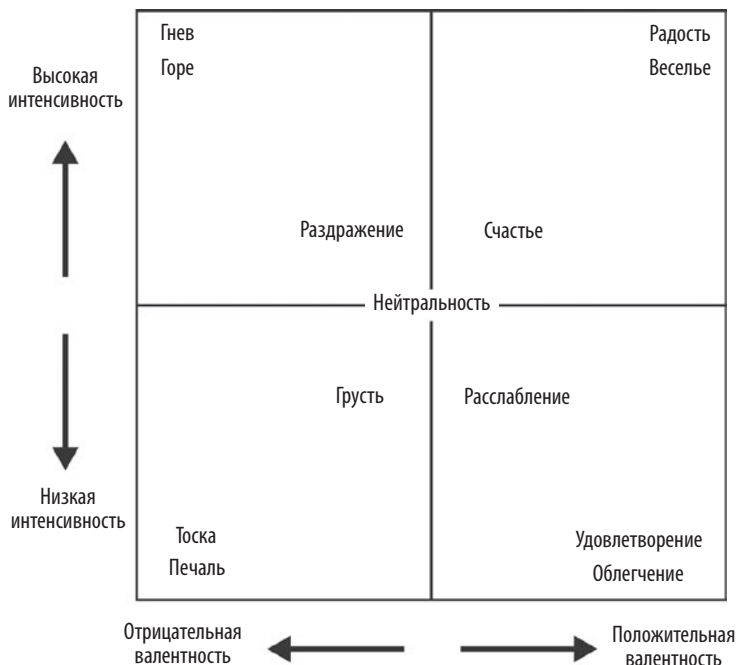
Подобную кривую можно найти в бесчисленных медиа: фильмах, книгах, комедийных программах, рекламных роликах, операх и песнях — ведь она невероятно эффективна. Люди «подсаживаются» на нее, такой прием удерживает внимание и оставляет зрителей довольными, не утомляя их.

Игры также могут демонстрировать эту кривую ритма. Она может появиться не только если ее добавят в предопределенную историю. Мы можем создать игровую механику, которая сгенерирует ее на лету.

Возьмите многопользовательский режим захвата флага в любом шутере. В начале игры каждая команда базируется на разных краях карты. Члены команды приближаются друг к другу с чувством нарастающего предвкушения.

В центре карты они сталкиваются друг с другом, и начинается битва. Затем противники переходят в ритм атаки и защиты. Когда время на исходе, ставки увеличиваются, а вместе с ними усиливается и напряжение. В конце миссии интенсивность, которая приводит к кульминации, нарастает, а игроки пытаются захватить свой последний флаг и повернуть исход игры в свою пользу. Затем у игроков остается несколько минут, чтобы повлиять на счетчик очков на экране. Кривая ритма, которая следует из их опыта, соответствует классической формуле истории из трех частей, но вместо того, чтобы быть стандартной, она генерируется с незначительными изменениями в каждой игре.

Кроме изменения интенсивности при смене темпа мы также можем варьировать вкус эмоций. Психологи называют это эмоциональной валентностью. Например, ярость, горе и страх имеют высокую интенсивность, но у них разная валентность. Удовлетворение, облегчение и подавленность — это эмоции низкой интенсивности с разными валентностями. Мы можем даже изобразить эмоции на рисунке, расположив их в зависимости от валентности и интенсивности:



При смене интенсивности в игре не следует ограничиваться только эмоциями, указанными на рисунке. Чтобы опыт сохранял новизну, игра может генерировать впечатления, которые направляют игрока по извилистому пути в уголок эмоционального спектра, от радости до гнева, депрессии и облегчения.

Поток

Поток — известное в психологии понятие, которое хорошо применимо к геймдизайну. Изначально оно было описано венгерским психологом Михаем Чиксентмихайи. Он описал его так: поток — это состояние сфокусированной концентрации высокой степени, сопоставимое с полным угасанием активности.

Еще в университете я служил некоторое время в резерве канадской армии. Особенность армии заключается в том, что там совершенно неприемлемо опаздывать даже на несколько секунд.

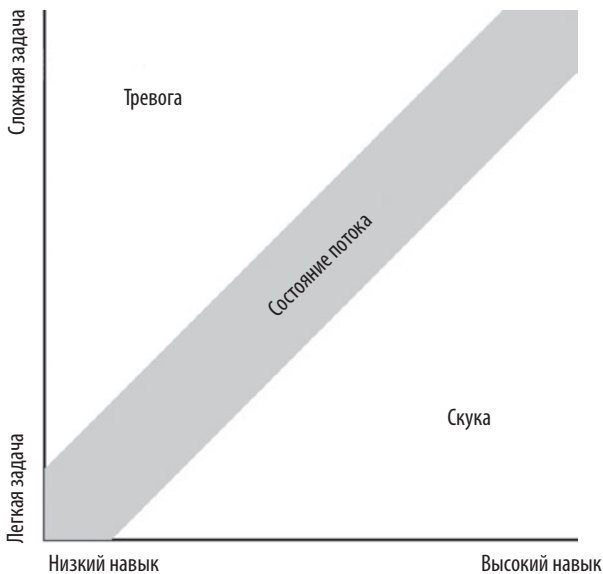
Однажды вечером в 19:00 в субботу у меня было строевое обучение. Я совершил роковую ошибку, сев играть в тот же день

в классическую стратегию Heroes of Might and Magic III. У меня было много времени. Часы показывали 17:00. Если я начну готовиться в 18:15, то у меня будет время переодеться в форму и я успею на плац.

Но «Герои» очень хорошо затягивают в поток. Я прокачал своего героя, победил нескольких грифонов и троглодитов, захватил город и немного сокровищ. Казалось бы, прошло несколько минут, а на часах было уже 18:37. Бежать до плаца было далеко.

Поток заставляет время исчезнуть. Если игрок полностью поглощен игрой, часы могут показаться минутами. Это совершенная форма эскапизма, потому что убирает все остальное из сознания. В потоке мы не беспокоимся о счетах, отношениях, деньгах или о том, наорет ли на нас инструктор по строевой подготовке. Это состояние очень приятно — оно состоит из непрерывного потока крошечных успехов.

Поток появляется, когда перед игроком возникает задача, которая идеально соответствует уровню его способностей. Если задача слишком сложная, поток прерывается, поскольку игрок теряется и начинает волноваться. Если слишком простая — игроку становится скучно.



Поток является основой для лучшего игрового опыта. Он работает на всех уровнях напряженности и эмоциональных валентностях. Увлекательные экшн-игры, квесты, требующие умственного напряжения, юмористические игры для социального взаимодействия — все это может создать поток, потому что каждая игра непрерывно занимает ум игрока, не перегружая его.

В любом случае, если поток нарушен, другие компоненты опыта распадаются. Практически все игры должны поддерживать непрерывность потока, а многие проблемы плохих игр сводятся всего лишь к разрывам в нем.

Погружение

Одним из самых мощных игровых опытов является погружение.

Все соглашались с тем, что погружение является очень ценным опытом, но нет единого мнения о том, откуда оно берется. Ис-

ПОГРУЖЕНИЕ (ИММЕРСИЯ) —

это размытие границ между настоящей личностью игрока и его игровым персонажем, вследствие чего события, происходящие с персонажем, становятся значимыми, как если бы они происходили в реальности с самим игроком.

точником погружения называли все: от вымышленной правдоподобности до графической точности, взаимосвязанных или безмолвных героев и даже меньшее количество света в комнате во время игры. Тем не менее существуют не иммерсионные игры, обладающие всем вышеперечисленным, и наоборот, игры с эффектом погружения, в которых ничего подобного не встречается.

Лучший способ описать иммерсию — это опыт игрока, отражающий опыт персонажа. Очевидно, это означает, что игрок видит и слышит то же, что и персонаж. Но что более важно, это также означает, что игрок думает и чувствует то, что думает и чувствует персонаж. Если персонаж боится, значит, и игрок тоже боится. Если персонаж злится, чувствует любопытство или чем-то ошеломлен, то же самое относится и к игроку. Если игрок думает и чувствует то же, что и персонаж, то тогда игра считается иммерсивной.

Иммерсия происходит в том случае, когда опыт игрока отражает опыт персонажа.

Это внутреннее психическое отражение и называют недостающим элементом в большинстве неудачных попыток погружения. Но как оно создается? Один из возможных ответов заключается в концепции психологии, называемой двухфакторной теорией эмоций.

Возбуждение — это состояние гиперактивности и готовности действовать. Сердце бьется быстрее, ладони потеют, а зрачки расширяются. Тело готово действовать прямо сейчас. Это состояние возбуждения возникает по разным причинам. Страх вызывает состояние сильного возбуждения точно так же, как и гнев, ритмичная музыка или сексуальное напряжение.

Двухфакторная теория эмоций гласит, что все наши различные сильные эмоции одинаковы с точки зрения физиологии и они основаны на состоянии возбуждения. Согласно теории, единственное различие между этими чувствами — это *когнитивная интерпретация* возбуждения, которой мы их наделяем.

ДВУХФАКТОРНАЯ ТЕОРИЯ ЭМОЦИЙ

гласит, что эмоции состоят из двух факторов: физиологического возбуждения и интерпретации этого возбуждения.

Когнитивная интерпретация возбуждения — это сознательное логическое объяснение того, что вызывает состояние возбуждения. В зависимости от ситуации, мозг будет отмечать состояние возбуждения как любую разновидность сильных эмоций. Например, ваша нервная система на взводе, если за вами гонится медведь, в этом случае вы обозначите свое эмоциональное состояние как страх. Такое же возбуждение, которое появляется через мгновение после того, как вас оскорбили, будет отмечено как гнев. Основной идеей двухфакторной теории является то, что в каждом случае состояние возбуждения на самом деле одно и то же, на уровне физиологии нет никакой разницы между гневом и страхом. Мы просто обозначаем их по-разному.

В 1962 году в ходе эксперимента исследователи сделали нескольким субъектам инъекции загадочного препарата. Это был адреналин, который вызывает потливость ладоней, учащенное сердцебиение и дыхание. Этих людей поместили в комнату с другим «испытуемым», который, по-видимому, также получил инъекцию. Группе людей не сказали, что этот человек — актер. В одних ситуациях он вел себя словно в состоянии эйфории. В других он злился. Во всех случаях субъекты сообщали, что испытывали те же чувства, что и притворяющийся актер. Но все, что они чувствовали, — это возбуждение, вызванное химическим веществом. Однако социальные сигналы актера заставили их «переименовать» это состояние в страх, гнев или эйфорию.

Если испытуемым говорили, какой препарат им ввели, то они не сообщали об эмоциональном состоянии, так как отмечали реакцию своего тела как бессмысленную реакцию на препарат.

Двухфакторная теория освещает множество эмоциональных парадоксов. Мы плачем от горя и счастья. Ночные клубы создают сексуальное напряжение, усиливая сердцебиение с помощью громкой музыки и танцев. Фильмы ужасов пользуются популярностью у пар. Грязные шутки работают за счет оскорбительных или обидных идей для создания эмоциональной реакции, а затем происходит «переименование» реакции в комический восторг. Существует даже примирительный секс, трансформирующий гнев в вожделение. В каждом случае мы ошибочно приписываем возбуждение чему-то, кроме его истинной причины. И это неправильное распределение является основой погружения.

Давайте разберемся.

Чтобы создать опыт, который отражает опыт персонажа, мы создаем его из трех частей. Во-первых, создаем поток, чтобы «изъять» реальный мир из головы игрока.

Во-вторых, создаем состояние возбуждения, используя в игровой механике угрозы и трудности. Наконец, добавляем сюжет, чтобы создать возбуждение игрока, которое будет соответствовать чувствам персонажа.

Первый ингредиент — это поток. Роль потока состоит в том, чтобы «изъять» реальный мир из головы игрока и тот мог погрузиться в игру. Он создается в основном в игровой механике, когда задача идеально соответствует уровню мастерства игрока. Это необходимое условие для погружения; без потока бессвязные мысли о счетах или домашнем задании постоянно вторгаются в опыт, разрушая любой шанс, которые может отразить настроение персонажа на игроке.

Вторым ингредиентом является возбуждение в чистом виде. Мы можем вызвать отчетливое, немаркированное возбуждение только с помощью игровой механики. Например, Pong, Geometry Wars и шашки могут вызывать возбуждение в том случае, если игра сложная и быстрая, решения трудны, а ставки высоки.

Последний ингредиент — сюжет. Без него возбуждение, вызванное механикой, отмечается как общее возбуждение, похожее на то, что вы чувствуете, играя в Geometry Wars. Но с помощью сюжета мы можем «переименовать» состояние возбуждения так, как нам нравится. Например, возбуждение может быть переименовано в ужас в игре

хоррор, полной страшных зомби. В ином случае оно может стать непоколебимой решимостью в военной игре. Даже если игровая механика, создающая этот опыт, одинакова, предлагаемая сюжетом когнитивная интерпретация возбуждения меняет то, как игроки воспринимают свой опыт.

В этом случае опыт превосходит простое вовлечение и переносит игрока в другое время и место. Управляемый механикой опыт потока убирает у игрока чувство смущения, стирает его восприятие реального мира и создает базовое состояние физиологического возбуждения. Сюжетный опыт превращает его личность в персонажа в вымышленном мире. Игрок видит и слышит то, что видит и слышит персонаж, и чувствует то, что чувствует персонаж. Игрок — это персонаж.

Например, одной из самых популярных ранних иммерсивных игр была Doom. В игре очень хорошо проработана игровая механика игры жанра экшн. Во время интенсивного боя игрок напрягается, его ладони потеют, разум забывает о внешнем мире. Сами по себе эти события не являются исключительными. Они бы произошли, даже если бы в игре вообще не было сюжета, — те же симптомы наблюдаются и у игроков Geometry Wars.

Но в Doom игрок смотрит глазами космодесантника, застрявшего в кишашей демонами колонии на марсианской луне. Стоны зомби и окровавленный мир сигнализируют игроку о том, что возбуждение, которое он испытывает, — это не просто волнение, а страх. И это все меняет.

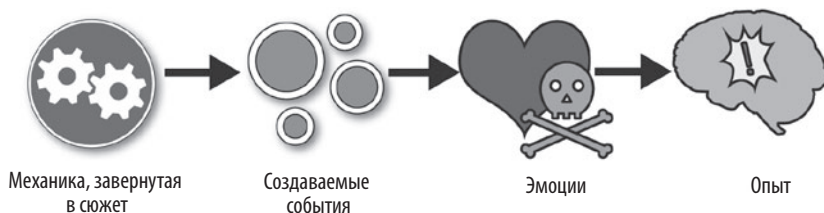
Тот факт, что космодесантник взбудоражен из-за того, что демоны пытаются убить его, а игрок, в свою очередь, взбудоражен из-за того, что в Doom хорошо продуманы бои, не имеет значения. В двухфакторной теории эмоций один источник возбуждения взаимозаменяем другим.

Теперь Doom — это не просто захватывающий шутер. Речь о том, чтобы быть космодесантником и бороться с зомби на марсианской базе, потому что опыт игрока отражает опыт персонажа. По сюжету, персонаж в ужасе, так как борется за свою жизнь. В реальности игрок в Doom чувствует возбуждение, вызванное механикой быстрого экшена, и приписывает этому возбужде-

нию эмоцию ужаса. Эти разрозненные ощущения сливаются воедино в сознании игрока. Игрок и космодесантник видят, слышат и чувствуют одно и то же, поэтому игрок чувствует, что он — космодесантник. Таким образом, игра иммерсивна.

Машины опыта

Опыт является последним звеном в концептуальной цепочке, по которой работают игры. Напомним: сначала разработчики создают совокупность механики. Они «заворачивают» эту механику в слой соответствующего сюжета. Во время игры механика взаимодействует, создавая длинную последовательность событий. Эти события активируют триггеры в подсознании игрока, вызывая эмоции. Наконец, эмоции объединяются в интегрированный опыт, который длится минуты, дни или годы.



Мы, возможно, только что случайно подошли к определению игры.

Иногда я представляю игры как необычные машины¹. Машины сделаны из тщательно разработанных металлических частей, которые идеально сочетаются друг с другом, а игры сделаны из тщательно разработанных механизмов, которые тоже идеально

¹ Автор называет игры машинами производства опыта, где «машины» — это «механизмы». Сам человек может быть назван машиной опыта, получаемого во время игры. Автор много говорит о бессознательном, что свидетельствует о его погруженности в аналитическую школу психологии. В этой школе есть традиция восприятия человека (на самом деле любой системы) как «машины желания» («Анти-Эдип», Ж. Делез и Ф. Гваттари). Тайнан вполне мог противопоставить игру как «машину, дающую опыт» человеческой «машине желания», желающей этот опыт получить. — *Примеч. ред.*

сочетаются друг с другом. В работе детали машины взаимодействуют друг с другом как шестеренки, создавая замысловатые фигуры, так же как и игровая механика взаимодействует определенным сложным образом. Наибольшее фундаментальное различие между играми и машинами заключается в характере того результата, который должна создать система. Физические машины созданы для приведения в движение транспортных средств, обогрева домов или создания виджетов. Игры сделаны, чтобы вызвать эмоции.

Если вы хотите понять, что такое игры, на примере аналогии, не думайте о рассказах или фильмах. Эти примеры не отражают основные достоинства игр, так как им не хватает динамического взаимодействия между игроком и механикой игры. Они отправляют нас по проторенному пути predetermined мультимедийного опыта и отдаляют от богатой целины импровизированной, исследовательской, сгенерированной интерактивности. Лучше думайте об игре как о необычной машине — машине опыта.

ИГРА — это искусственная система генерации опыта.

ЧАСТЬ II

Искусство создания игры

С момента появления современного геймдизайна в 1970-х появилось очень много нового. Но эти знания распространяются среди тысяч дизайнеров в сотнях студий. Одна студия освоила разветвленные нарративы, другая может идеально сбалансировать стратегические игры, а третья создает атмосферные игры. Эта часть книги написана для того, чтобы связать эти разрозненные знания в доступный набор принципов геймдизайна.

Единой теории геймдизайна не существует по причине того, что каждое решение дизайнера влечет много разных последствий. Добавление обучающего персонажа может сделать игру проще, но при этом усложнит ее реализацию и уменьшит связность сюжета. Добавление графики может сделать игру более красивой, но приведет к тому, что игрок будет делать неправильный выбор. Эти многочисленные последствия требуют нескольких объяснений. Вот почему эта часть книги охватывает многие различные точки зрения на геймдизайн, но ни одна из них не является истинной. Каждая точка зрения помогает нам понять другой аспект.

Идеи, описанные в этой книге, являются основой. Чтобы они были полезны при принятии проектных решений, необходимо подкрепить их опытом. Ваши идеи должны быть глубокими и простыми. Наблюдайте за успехами и неудачами других проектов. Благодаря такому опыту у вас появится эталон. Вы начнете интуитивно чувствовать момент, когда и в какой степени каждая идея в вашей структуре геймдизайна становится важной.

Геймдизайну невозможно научиться по книгам. Нужен опыт.

Лучшее всего изучать геймдизайн, наблюдая за отдельными небольшими изменениями в игре.

Но для изучения геймдизайна подойдет не любой опыт.

Люди, пытающиеся понять игры, часто сравнивают их между собой и выясняют, как их различия влияют на игру. Но выяснить принципы разработки таким способом достаточно сложно. Между играми так много различий, что невозможно выделить разнообразные факторы какой-либо одной игры.

Лучше всего учиться, наблюдая за влиянием крошечных изменений. Разработчик следит за тем, как три человека играют в игру, и наблюдает закономерности в их опыте. Он видит, что игроки понимают одно, но не понимают

другое, выполняют определенные действия, запоминают определенные события.

Затем разработчик вносит изменение в проект. В результате наблюдения за тремя новыми геймерами он видит, как особым образом меняется их опыт — одна новая идея понята, другая — нет, одно воспоминание появилось, другое исчезло. Разработчик понимает, что одно небольшое изменение привело к тому или иному характерному эффекту.

И этот эффект будет понятным и предсказуемым. Любители думают, что игры загадочны и непостижимы; в то время как профессионалы видели достаточно повторяющихся эффектов, обусловленных отдельными изменениями, чтобы понять, что игры — это рациональные системы. Хотя игровой опыт может быть волшебным, механизмы, лежащие в основе этого опыта, вполне прозаичны. Давайте заглянем за эти кулисы.

Элегантность

В некотором смысле, каждая игра уже существует. Они уже есть, скрытые в разуме Вселенной. Мы не создаем их. Мы находим их, подобно скульптору, который видит статую в неотесанном куске мрамора, — не добавляем, а убираем лишнее, обнажая внутреннюю форму.

У каждой игровой механики есть цена. Требуется разработка, поскольку игра должна быть реализована, настроена и протестирована. Требуются вычислительные ресурсы, которые можно было бы использовать где-то еще. Возможно, потребуется изменение сюжета или смещение фокуса маркетинга.

Однако дороже всего стоит внимание игрока. Игроки должны постоянно что-то делать, чтобы понять игру. Они должны следовать инструкциям, ошибаться, терпеть неудачу и пытаться снова. У некоторых не получится, и они уйдут. Другие будут озадачены и расстроены.

Игроки соглашаются с этой ценой, потому что хотят получить соответствующий опыт. Хороший проект означает получение максимальной эмоциональной силы и разнообразия игрового опыта при минимальных нагрузках на понимание и усилиях разработчиков. Эта форма эффективности называется элегантностью.

Непредсказуемость в основе элегантности

Правила игры в шашки довольно просты, но в этой игре можно генерировать бесконечное множество различных вариаций. В одних играх приходится долго мучиться. В других же, наоборот, все происходит очень быстро. В одной игре можно столкнуться с необыкновенным неожиданным поражением, а другая преподаст важный урок. И какова цена всего этого? Несколько минут простых инструкций в начале первой игры. Это и есть элегант-

ность: бесчисленные сильные разнообразные впечатления, вызванные простым и понятным дизайном.

Этот уровень элегантности невозможен в других медиа. Хороший сценарист может написать историю, которая создаст три или четыре эмоции на разных просмотрах, а хороший геймдизайнер может создать механику, которая генерирует тысячи или даже миллионы опытов, так же как правила игры в шашки приводят к появлению миллионов различных игр. Игры могут создавать огромное количество возможностей с помощью непредсказуемости.

Представьте, как вы описываете другу игру, в которую играли вчера вечером: «Это было потрясно! Я ехал в кузове джипа с миниганом. Мой друг был за рулем. Мы направились к вражеской базе, заехали на холм и пролетели в воздухе не менее пяти секунд! Я застрелил трех врагов, пока мы были в воздухе. Затем нас сбила ракета, но мы не выпали! Наша машина сделала в воздухе сальто, а я все время стрелял, даже когда был вверх ногами. Мы приземлились ровно на четыре колеса, друг переехал парня с ракетной установкой, и мы захватили флаг. Никогда этого не забуду».

НЕПРЕДСКАЗУЕМЫЙ
ГЕЙМПЛЕЙ — это
взаимодействие про-
стой механики для
создания сложных
ситуаций.

Это был многопользовательский матч игры Halo: Combat Evolved. Но этот опыт не записан на диске Halo, и он никогда не повторится таким же образом. Он возник в процессе игры в результате взаимодействия простых механик, таких как физика, гравитация, тюнинг оружия, карты и многопользовательский режим с разделенным экраном. Дизайн игры Halo считается хорошим не потому, что в ней присутствует такой насыщенный опыт, а потому, что в ней существует игровая механика, которая регулярно его генерирует. И игра будет продолжать генерировать новый опыт — всегда. Это и называется силой непредсказуемости.

Механика стрельбы может существовать сама по себе. Например, представьте себе игру, в которой вы стреляете из неподвижной пушки, чтобы поразить вражеские самолеты, когда они пролетают мимо. В этой игре только один элемент управления: кнопка «пуск». И это дает несколько типов простых опытов. Вы стреляете и промахиваетесь или вы стреляете, попадаете и убиваете.

Зрительная механика также может существовать сама по себе. Представьте симулятор американских горок, в котором ваше единственное взаимодействие — это смотреть по сторонам. Опять же, у вас существует один элемент управления: джойстик для камеры. И опять же, опыта не так много. После того как вы рассмотрели все во всех направлениях на каждой поездке на американских горках, делать больше нечего.

Усиление непредсказуемости означает создание механики, которая не просто суммируется, но и умножается, создавая богатую вселенную возможностей.

А теперь представьте, что будет, если объединить эту механику в одной игре. Вы можете кататься на американских горках, смотреть в любом направлении и стрелять в пролетающие мимо самолеты. Эта комбинация не просто складывает воедино опыт «смотрения» по сторонам и стрельбы. Она превращает их в комбинаторный взрыв новых возникающих возможностей. Теперь мы можем прицелиться там, где раньше это было невозможно. Игроки должны расставлять приоритеты между врагами. Игроку, возможно, даже придется научиться

ситуационной осведомленности, чтобы знать, где искать цели, которые не отображаются на экране. Эта простая комбинация «смотри-стреляй» настолько изящна, что она лежит в основе бесчисленных игр, от *The House of the Dead* до *Space Invaders*.

Теперь представьте, что вы добавляете механику движения. Игрок может бегать вокруг, смотреть и стрелять в любом направлении. Количество возможностей снова увеличивается. Теперь игрок может передвигаться, чтобы уклоняться от атак, бросаться в атаку или исследовать локацию, чтобы узнать ее историю. Игры, разработанные на основе этой простой комбинации «стреляй, смотри и беги», приносят миллиарды долларов каждый год. Они сильно различаются по сюжету и акценту: в одной игре вы — космодесантник, стреляющий в инопланетян; в другой вы исследуете мрачный подводный город. Но все эти игры имеют одинаковое элегантное ядро: «стреляй, смотри и беги».

И цена этих миллионов различных игровых событий удивительно невысокая. Разработчикам нужно просто добавить некоторую механику. Игрокам нужно просто выучить несколько элементов управления. Как только эти условия будут выполнены, откроется новый мир с миллионом вариаций триумфа, печали, напряжения и радости.

Элегантность требует взаимодействия разной механики. Например, комбинация «смотри, стреляй, беги» работает хорошо потому, что игрок использует все эти элементы управления одновременно. Поскольку вся механика взаимосвязана, она может открывать множество различных возможностей. Но это тесное взаимодействие механики усложняет процесс разработки, поскольку изменения в одной механике также влияют на всю остальную механику.

В «неэлегантной» игре отдельные проблемы легко исправить. Если посох, которым волшебник убивает гоблинов, слишком мощный при использовании против гоблинов, разработчик может просто уменьшить его силу. Поскольку посох не влияет ни на что, кроме гоблинов, это изменение не имеет побочных эффектов, и проблема решена.

Но такое простое решение возможно только благодаря элегантному дизайну. Почему вы не можете использовать посох против орков, огров, других волшебников, богов или скупых лавочников? Посох волшебника подразумевает целую вселенную возможностей в мире фэнтези. Ограничить его действие только гоблинами — значит потерять все эти возможности.

Более элегантная игра позволила бы все эти взаимодействия, создавая намного больше игровых ситуаций без необходимости тратить большое количество времени на обучение. Но это создает проблему для разработчика. Теперь, когда посох волшебника связан со многими другими частями игры, меняя его, вы меняете все эти взаимодействия. Если ослабить его действие против гоблинов, это может привести к тому, что посох не будет действовать на орков. Обеспечив посох возможностью зрелищно взрываться, можно создать захватывающий бой на арене, при этом он будет опасен для использования рядом с союзниками. Проблемы такого рода могут становиться очень острыми, когда речь идет о сотнях или тысячах взаимосвязанных элементов.

Вот почему простые, элегантные игры настолько необычны. Создать систему взаимоотношений гораздо сложнее, чем написать серию одноразовых трюков, но это единственный способ получить продолжительный игровой опыт из нескольких игровых механик.

Элегантность возникает тогда, когда механика взаимодействует сложным, неочевидным образом. И эта же сложность и неочевидность усложняют путь к достижению элегантного дизайна.

Прекрасный запах элегантности поутру

Будучи первокурсником, на вводном курсе по математике я изучал двумерные таблицы чисел — *матрицы*. Одно свойство матриц состоит в том, что некоторые из них бывают обратными — преобразованными в новую форму с помощью ряда числовых операций, а другие — нет. Проблема заключалась в том, что не существует простого способа определить, является ли матрица обратной или нет. Вы просто должны были выяснить, а это отнимало много времени.

Но профессор научил нас другому. Он сказал, что можно научиться «чувять» обратные матрицы, просто взглянув на них. Вы не будете знать наверняка, как это узнали, но с опытом ваше подсознание сможет дать ответ.

Ощущение элегантной механики почти такое же. Разработчик не может предсказать каждый результат, который создаст элегантная игра. Они слишком многочисленны и размыты. Можно внедрить механику и тщательно ее протестировать, но это может отнять много времени. Нам нужны способы, которые помогут определить элегантность на этапе разработки. И единственный реальный способ добиться этого — почувствовать элегантность так же, как и наш профессор ощущал запах инвертированных матриц: используя натренированную интуицию и ментальную эвристику.

Почувствовать запах элегантности — это умение, которое приходит с опытом. Но я обнаружил несколько простых правил, которые могут в этом помочь.

Посчитайте в уме количество взаимодействий, ожидаемых от предложенной механики. Если она взаимодействует со многими другими механиками, то это скорее будет элегантный дизайн.

Если она взаимодействует только с одной или двумя, то, вероятно, нет.

Например, обдумывая заклинание в фэнтезийной игре в жанре RPG (ролевая игра), спросите себя: может ли оно взаимодействовать с другими заклинаниями? Сможет ли взаимодействовать с членами группы, с несколькими про-

Механика, которая взаимодействует со многими другими механиками, пахнет элегантностью.

тивниками, с окружающей средой, системой морали, нарративом или статистикой игрока? Механика может даже взаимодействовать сама с собой. Например, в шахматах в разных позициях пешки могут претендовать на разные части доски.

Элегантность заключается не только в снижении стоимости механики, но и в увеличении эффективности. Раздутая, слишком сложная механика может дать хороший результат, но зачастую она не стоит того времени, которое игрок потратит на обучение. Ее упрощение, возможно, и приведет к потере нескольких нюансов в игре, но в то же время откроет возможности для других, более эффективных проектных решений.

Кроме того, снижение сложности, с которой сталкиваются игроки, повышает вероятность понимания остальных событий. Игроки, которые не перегружены, будут в полной мере исследовать игру и наслаждаться каждой деталью опыта. Утопающие в сложностях игры упустят многое из того, что она может предложить.

Самая изящная механика настолько проста, что кажется абсолютно очевидной после того, как вы узнали ее. Поэтому постарайтесь найти решения, которые сможете написать на клочке салфетки.

Простая механика
пахнет элегантностью.

Инструмент, который можно использовать творчески, наступательно, оборонительно, тактически и стратегически, более элегантен, чем тот, который выполняет только одну функцию. Он не только взаимодействует во многих отношениях, но также создает новые возможности выбора и связи, объединяя эти разные функции.

Механика, которую
можно использовать
несколькими спосо-
бами, пахнет элегант-
ностью.

Например, в большинстве шутеров оружие используется в основном для нападения. Игроки защищаются побегом и укрытием, используя оружие только для того, чтобы напасть на врага. Серия игр The Resident Evil в жанре симулятора выживания решает эту задачу по-своему. В Resident Evil игрок не может двигаться во время стрельбы, а выстрелы приводят к тому, что нападающие зомби останавливаются на несколько секунд. В этой игре оружие используется одновременно и для наступления, и для обороны, а игроки часто стреляют по зомби без намерения убивать их, а просто чтобы замедлить их при-

ближение. Привязка функции атаки и защиты к одному и тому же инструменту означает, что игроки должны выбирать между потребностью в наступлении или обороне. Вы замедлите зомби, который волочит свое туловище в вашу сторону или выстрелите в голову ценной мишени вдали?

Подобные компромиссы присутствуют повсюду. Одно подразделение в стратегической игре может быть использовано для атаки, защиты и разведки. Механика поворота в гонках может использоваться для преодоления поворотов и блокирования других водителей. А в некоторых играх в жанре стелс игрок может бросать предметы, чтобы вырубить охранников или создать шум, привлекая их внимание.

Роль — это способ использования механики. Например, в стратегиях встречаются такие виды персонажей (юниты), как приставала (харассер)¹, разведчик, блокирующий или плут (десивер)². В строительной игре инструментом может быть экскаватор, конструктор или декоратор. Атака в файтинге может быть представлена ударом, блоком, блок-брейкером или дистанционной атакой. У каждого из этих инструментов в наличии цель, которая не может быть достигнута другими средствами. Если эти роли совпадают, игра теряет элегантность, потому что вы платите за то, чтобы две механики выполняли одну и ту же роль.

Механика, которая не совпадает с ролью другой механики, пахнет элегантностью.

Например, если в стратегии уже присутствует один вид разведывательной единицы, добавлять другой не имеет смысла, если только его функция не будет отличаться. Второй вид разведывательной единицы должен привести к появлению новых и значимых игровых впечатлений, которые сможет дать первый вид. Если такого не происходит, то в вашем проекте появился балласт.

¹ Харассер — юнит, подходит для постоянных нападений на вражескую базу мелкими атаками, уничтожая рабочих и ломая «билд» — запланированную схему развития. Он быстро производится, быстро двигается, но достаточно хрупок. — *Примеч. ред.*

² Десивер — юнит, обладающий уникальными способностями, создающими помехи противнику: скрывающий часть карты, маскирующийся под вражеские войска, создавая миражи объектов на поле боя. — *Примеч. ред.*

Самые элегантные механики имеют настолько различные роли, что открывают совершенно новые виды игры. Не создавайте вариаций уже существующих взаимодействий. Лучше ищите раннее не существующие механики, которые вводят новые стратегии или возможности исследования.

Мы можем облегчить понимание, используя символы и условные обозначения, которые уже знакомы игрокам. Они могут находиться где угодно: в других системах игры, других играх, реальной жизни или культурных архетипах. Поскольку игроки уже знают их, мы можем извлечь выгоду из их использования.

Если игра соответствует требованиям жанра, используйте стандартную для жанра схему управления. В игре без сложного сюжета наденьте злодею острые усы, и пусть он их закручивает так, чтобы все сразу знали, кто перед ними.

Не включайте в игру ничего уникального без веской причины, потому что это требует дополнительных усилий для понимания.

Каждый хочет быть новатором и проявить творческий подход. Но за создание новых общепринятых норм приходится платить не только разработчикам, но и игрокам. Если мы все равно собираемся это сделать, то необходимо удостовериться в том, что оно того стоит, а не произвольно менять хорошее решение без причины. Настоящая оригинальность заключается не в поверхностных изменениях. Необходимо менять основы.

Даже очень оригинальные проекты часто представляют собой комбинации ранее существовавших концепций, поэтому даже исходная игра может внести значительный вклад, например из нее можно позаимствовать общеизвестные символы и интерфейсы для передачи уникального контента.

Рассмотрим игру, в которой игрок то управляет героем, который ходит пешком, то пилотирует истребитель. Этот вариант может выглядеть свежо, но за это приходится платить, потому что он разделяет игру на две механики с недостаточным взаимодействием между ними.

Когда герой находится на земле, важно, куда бегут вражеские солдаты. Враг, бегущий на 3 метра влево, может

Механики, которые повторно используют имеющуюся архитектуру и интерфейсы, пахнут элегантностью, потому что они основаны на знаниях, которые уже присутствуют у игроков.

Механика, которая работает подобно уже существующей механике, обладает элегантностью.

быть уязвим на открытой локации, а тот, кто бежит на 3 метра вправо, может спрятаться за стеной. Но в самолете все это не имеет смысла. Бомба весом 450 кг убьет солдата независимо от того, в какую сторону он бежит и за каким ящиком прячется. Стены и здания не считаются, так как все они сгорели от взрыва.

Если игрок управляет самолетом, движение солдата по земле не имеет смысла. Разработчики напрасно потратили усилия на реализацию этого решения, а игрок — время на то, чтобы разобратся и понять игру. Вы заплатили определенную цену, но ничего не получили взамен.

Уровень между пилотированием самолета и управлением пешеходным героем настолько разный, что эта игра представляет собой две отдельные игры в одном флаконе. Вы можете переключаться между двумя режимами, но они никогда не сольются воедино в интегрированную систему. Если игрок находится в одном режиме, вся сложность другого режима становится бессмысленной, потому что уровни очень разные. Это не элегантно.

Magic: The Gathering — прекрасный пример игры, которая максимально увеличивает параметр элегантности, сопоставляя масштабирование различных игровых систем. Игроки получают семь карт. На старте дается 20 жизней. Обычно игроки могут использовать от 3 до 10 единиц маны за ход. В игре обычно в наличии от 2 до 12 карт с существами, артефактами и колдовством. Рейтинг силы и выносливости существ обычно составляет от одного до восьми. Игра занимает от 10 до 30 ходов. Такое плотное масштабирование между номерами карт, жизнью, количеством существ, их силой и выносливостью, количеством ходов и маной обеспечивает огромное количество естественных взаимодействий без какой-либо сложной математики. В игре существуют карты, которые конвертируются почти во всех направлениях. Они могут принимать урон или жертвовать существами в обмен на другие способности. Они могут пожертвовать жизнью, чтобы прокачать силы существ, заколдовать их и сбросить карты, чтобы воскресить убитых в бою существ. Тысячи таких взаимодействий могут появиться из нескольких простых систем.

Если бы разработчик игры Magic Ричард Гарфилд решил ограничить игрока тысячей жизней, оставить уровень выносливости

существа в пределах от 25 до 50, а игрокам разрешалось бы иметь не более 3 карт земли одновременно, то взаимодействия были бы нарушены. Для преобразования чисел потребовались бы утомительная математика и сложное округление, и игра утратила бы свою элегантность.

Элегантность масштабирования можно применить практически к любому типу взаимосвязи между измеряемыми величинами. Размер, скорость, площадь поверхности, очки здоровья, расстояния, деньги, энергия, каналы связи, ресурсы, время и количество игроков или персонажей — все это кандидаты на элегантное масштабирование.

Один из парадоксальных аспектов хороших игр — насколько они кажутся повторяющимися. Игроки используют одни и те же инструменты снова и снова. Они снова и снова строят больше городов, побеждают больше орков и украшают новые дома.

Механика, которая используется повторно, «пахнет» элегантно-стью.

Но только то, что игрок использует один и тот же инструмент, не означает, что он получает тот же опыт. Возможно, он опять и опять строит города, но каждый город — это новое проявление механики и решений игрока. Механика остается прежней, но опыт каждый раз меняется.

Кроме того, повторение механики играет важную роль для элегантного дизайна. Механика, которая используется только один раз, — это хитрый трюк. Она может стоить того, но не может быть элегантной, потому что затраты и результат всегда будут в соотношении один к одному. В теории механика, которая используется 100 000 раз, может быть очень элегантной, если она способна каждый раз генерировать новый опыт. Повторение не гарантирует элегантность, но без него она невозможна.

Это один из самых простых признаков элегантности, который можно заметить. Сложно сказать, насколько «утонченными» будут взаимодействия между механикой и ее соседями. Но обычно достаточно легко определить, насколько часто механика будет повторяться. Существует очень много дизайнерских идей, которые выглядят великолепно, когда видишь их впервые, но после тысячного

Механики, которые накладывают ограничений на содержание, воспринимаются как элегантные.

применения они становятся пресными. Ищите те механики, которые не потеряют свой вкус, даже если будут «заиграны» до дыр. Найдите механику, которую можно повторять миллионы раз.

Представьте научно-фантастическую детективную игру, в которой герой находит улики, решает поставленные задачи и иногда ловит злодеев. Чтобы улучшить последовательность действий, кто-то предлагает обуть главного героя в ракетные ботинки, которые позволят ему прыгать на шесть метров вверх. Запустив один прототип, разработчики обнаруживают, что последовательность действий тут же улучшилась. Ракетные ботинки быстренько вписывают в официальный дизайн, и все остаются довольны.

Возможно, эти разработчики слишком рано радуются. Хотя они и сделали игру лучше в краткосрочной перспективе, они также спровоцировали огромные скрытые расходы и, возможно, в результате нанесли игре вред.

Теперь, при том что главный герой может прыгать на шесть метров вверх, чтобы поддерживать эту высоту прыжка, нужно менять в игре каждый уровень. В игре не должно появиться места, куда игрок не должен попасть, но при этом он сможет туда запрыгнуть. Поэтому дизайнеры уровней должны найти способы заблокировать возможность для игрока попасть в тысячу «неправильных» мест. Им придется изменить сюжет, чтобы игрок не перепрыгивал через автобусы и парковые ограждения. Карту с пригородными бунгало придется обрезать, потому что невозможно остановить игроков, перепрыгивающих через крыши домов. Разработчики заканчивают тем, что добавляют груды ящиков, невидимые стены или неуместные рекламные щиты только для того, чтобы остановить игроков, покидающих игровую зону. Все это означает, что игра становится немного бессмысленнее, сюжетная линия тоньше, а уровни хуже, поскольку разработчики тратят время, пытаясь удержать прыгунов, вместо того чтобы работать над качеством игры.

Шестиметровый прыжок привел к необходимости ограничения контента. Пришлось разрабатывать все уровни таким образом, чтобы игрок не смог прервать игру, прыгнув вверх. На первый взгляд кажется, что здесь нет ничего сложного, но на самом деле

дизайнеры уровней тратят на это много ресурсов. Подобная работа вовсе не приносит удовольствия, замедляет и искажает все, что делают разработчики.

Это весьма распространенная ошибка, потому что преимущества решений очевидны, а подводные камни — скрыты. Новый дизайн, который создает ограничения контента, может принести мгновенную примитивную выгоду для дизайна. Он хорошо вписывается в прототип, выдерживает тестирование и приносит людям удовольствие. Преимущества сосредоточены в одном месте и сиюминутны, в то время как затраты распределяются на годы и в основном их несет кто-то другой. И несмотря на то, что каждое ограничение дает некоторую узкую краткосрочную выгоду, в совокупности все это может погасить игру.

Существуют тысячи видов ограничений контента. Длина прыжка, высота комнаты, количество персонажей, количество ресурсов, длина диалога, размеры инвентаря и транспортного средства — все это может быть ограничено проектными решениями. Иногда, как в случае с шестиметровым прыжком, ограничение появляется в результате изменения дизайна. В других случаях речь идет о поддержании удобной частоты кадров.

Более элегантная механика может улучшить игру без таких скрытых затрат. Ищите такой дизайн, который работает с существующим контентом, и имейте в виду, что стоимость ограничений контента хорошо скрыта и обычно составляет больше, чем вы предполагаете. Даже очень хорошо разработанная механика может не иметь смысла, если вокруг нее придется выстраивать всю остальную игру.

Если интерфейс игры представляет собой аналоговый джойстик, он может определять не только направление, но и точный угол рычага управления. Аналоговый триггер может иметь высокую чувствительность. Даже если интерфейс представляет собой просто кнопку, с ее помощью все равно игрой можно управлять не только при ее нажатии, но и отпуская или удерживая ее.

Опасность заключается в том, что не всем игрокам может понравиться такая сверхчувствительность. Подобные

Механики, в которых доступный интерфейс используется на полную мощность, обладают свойством элегантности.

средства управления хороши в том случае, если опытные игроки используют спектр мощности, но новички могут смело проходить мимо. Например, любой может достаточно хорошо играть в гонки, неистово крутя руль влево и вправо, игнорируя при этом его чувствительность, и только эксперт может использовать каждый градус поворота по максимуму.

Это распространяется и на настольные, и на карточные игры. Сколько всего можно сделать с картами, кубиками или жетонами?

Примеры элегантности:
«Хищник» против «Геллиона»

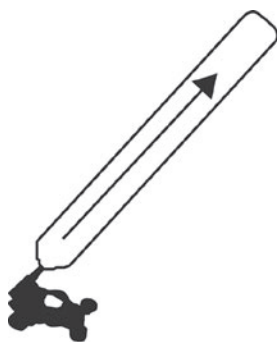
StarCraft II: Wings of Liberty включает две боевые единицы из расы Терранов с очень похожими ролями. «Хищник» и «Геллион» — быстрые машины средней стоимости, которые предназначены в основном для борьбы с небольшими группами врагов и используют атаки по площади.

Но в многопользовательскую игру был включен только «Геллион». Это было хорошее дизайнерское решение, потому что, несмотря на то что оба юнита выполняют одну и ту же основную роль, небольшие различия между ними делают «Геллион» гораздо более элегантным в плане дизайна. Давайте разберемся, почему это так.

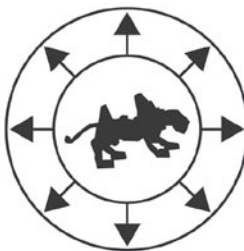
	«Хищник»	«Геллион»
Скорость	4	4,25
Очки	140	90
Атака	Наносит противникам 15 единиц урона передними лапами и вызывает ударную волну, наносящую 20 единиц урона всем юнитам, окружающим «Хищника» на близком расстоянии	Выпускает поток пламени среднего радиуса, который наносит 8 единиц урона всем юнитам. Наносит 6 дополнительных единиц урона легким юнитам, благодаря огнеметам «Адское пламя» урон увеличивается еще на 5 единиц
Задержка между атаками	1 секунда	2,5 секунды

Сразу видно, что эти две боевые единицы очень похожи. Скорость обеих примерно в два раза быстрее, чем у солдата, передвига-

ющегося пешком. У них одинаковый запас здоровья, и они обе наносят урон по врагам в зоне поражения. Основное отличие заключается в геометрии их атак. Обе используют атаки по площади, но «Хищник» наносит урон по кругу вокруг себя, а «Геллион» — длинным узким потоком. И это создает огромную разницу.



«Геллион»



«Хищник»

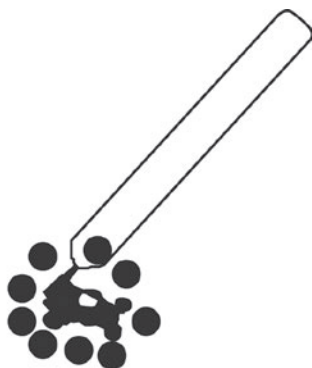
Существует несколько разных способов использования «Хищника». Лучший вариант — находиться в окружении врагов так, чтобы они все попали в кольцо атаки. Но это случается редко, так как другой игрок может просто отказаться от атаки. «Хищник» также может атаковать большие группы врагов, но основная часть круговой ударной волны не поразит их. Наконец, «Хищник» может столкнуться с группой вражеских рабочих и попытаться окружить их.

Кроме описанного, остается не так много других способов использовать этот персонаж. Его малый радиус действия означает, что он не может прятаться за укрытием или стрелять по уступам, а между ним и другими персонажами-союзниками мало синергии. Бой «Хищников», как правило, имеет один и тот же сценарий.

«Геллион» совсем другой. Форма его атаки предполагает, что в зависимости от геометрии среды и наличия врагов его эффективность значительно меняется. Если много слабых врагов выстраиваются в линию, их всех поражает поток пламени, нанося огромный урон. Если «Геллион» оказывается в окружении, его пламя поражает только одного противника, что практически бесполезно.



Приближаются враги



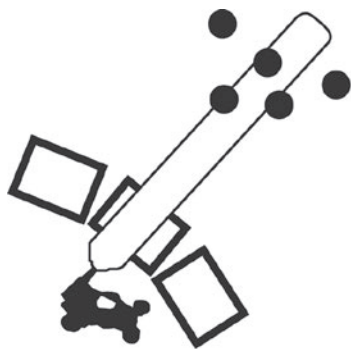
«Геллион» окружен

Эта простая разница создает целый ряд нюансов игрового опыта, когда «Геллион» пытается обходить врагов и стрелять по прямой, а его противники пытаются приблизиться и окружить его.

У «Геллиона» хорошая синергия с окружающей средой и другими юнитами-союзниками. Так как его атака наносит дистанционный урон, игрок может спрятать «Геллион» за стену в качестве прикрытия или поставить его за другими юнитами, чтобы помочь им с атакой против групп слабых врагов. «Геллион» также может сбивать врагов с уступа.

Наконец, «Геллион» атакует гораздо реже, чем «Хищник». У игрока есть достаточно времени между каждым выстрелом, чтобы переместить «Геллион». Это позволяет использовать продвинутую тактику «стреляй и убегай», когда игроки стреляют, перемещают «Геллион» и снова стреляют, стараясь избежать урона и заставить врагов выстроиться в ряд при каждом выстреле. «Хищник», напротив, атакует часто, поэтому аналогичная тактика невозможна. Вы должны просто бросить его в бой и надеяться на то, что он победит. Его предсказуемость не оставляет никакой надежды на интригу или прокачку навыков.

С точки зрения механики «Геллион» не сложнее, чем «Хищник». Он одинаково прост и для внедрения в игру, и для понимания игроками. Но в нем больше элегантности, потому что он генерирует намного больше сложных задач, тактик и ситуаций, чем это делает «Хищник».



Стрельба по прикрытию
или союзникам



Нельзя использовать
прикрытие или союзников

Превосходство «Геллиона» не очевидно. Разница между ними совсем незначительная: несколько секунд во времени атаки, другая форма урона. Как бы то ни было, на первый взгляд «Хищник» выглядит более интересным, в то время как «Геллион» прозаичен, не интересен, даже немного скучен. Но именно так выглядит элегантность, и поэтому ее трудно сразу определить. Именно потому, что элегантность не заключается в ярких трюках или захватывающих дизайнерских идеях. Она заключается в обратном: простом, искусном дизайне, который расцветает во множественном игровом опыте.

Когда мы описываем самые элегантные игровые системы, которые когда-либо были созданы, они могут показаться чересчур простыми и понятными и даже скучными. Но форма самого дизайна не имеет значения. Важны глубина и богатство пространства возможностей, которое возникает во время игры.

Навык игрока

Победа! О, победа! Твоя цена — от боли крики,
Потоки мыслей, что неостановимы.
Без кровопролитных битв, однако,
Бесплотна и пуста награда.

Сложное задание может создать напряженную конкуренцию и захватывающее ощущение победы. Оно может генерировать увлекательные стратегические решения, чтобы приковать ваше внимание и преподавать увлекательные уроки. Оно также может создать социальный опыт уничтожения или помощи другим.

Но любая игра, в которой присутствует сложное задание, а большинство из них именно такие, должны учитывать вопрос навыка игрока. От слишком сложных задач иногда опускаются руки. Слишком легкие — скучно. Удачные, яркие опыты находятся между двумя этими крайностями.

Подвох в том, что эта идеальная зона комфорта для каждого игрока своя. Вращающийся детский мобиль завораживает ребенка, но совсем не интересуется взрослого. Турнирная игра с высокой степенью конкуренции, вроде StarCraft, привлекает профессиональных геймеров, но отпугивает всех остальных.

Учитывать навык игрока — значит понимать нижнюю границу его умения играть в игру, ниже которой игра становится разочарованием, и верхнюю границу, после которой она становится скучной. Иначе говоря, необходимо понять, можно ли расширить эти границы, чтобы привлечь больше игроков, и если да, то каким образом. Кроме того, это означает, что вы знаете, как разработать продуманные тесты с настоящими поражениями для определения навыков игрока, при этом данные поражения не должны вредить полученному опыту. Этим вопросам и посвящена данная глава.

Глубина

Идея глубины заключается в том, как много нужно узнать об игре. В сложной игре достаточно много нюансов и вариаций, благодаря которым можно долго узнавать что-то новое. Шахматы, футбол, покер и StarCraft — все это сложные игры, потому что игроки могут изучать их десятилетиями, при этом новые знания не заканчиваются.

Сложные игры создают стоящую игру при высоком уровне игрового навыка.

Противоположностью этому является неглубокая игра. Например, крестики-нолики — одна из них, потому что, как только вы понимаете, в чем ее секрет, вам больше нечего узнавать. Игра интересна только маленьким детям, которые еще не до конца ее понимают. Для взрослого, который знает, как закончить каждую игру вничью, это совсем не интересно.

По иронии судьбы, игроки будут изо всех сил стараться разгадать игру, но, если им это когда-нибудь удастся, они будут ненавидеть дизайнера. Игроки ценят опыт преодоления барьеров, способность сегодня сделать то, что они не могли сделать вчера. Разгаданная игра для них бесполезна, потому что она не дает возможности что-нибудь узнать, нет неуверенности, нет победы, нет поражения. Она становится упражнением по выполнению ряда четко определенных заданий, которые приведут к гарантированному результату. Игры с прогнозируемым исходом полезны только тем, кто их не понимает.

Одним из быстрых способов определения глубины является рассмотрение разницы в результатах у игрока, который безупречно владеет теорией, и у игрока, который настолько мастерски владеет техникой, насколько это вообще возможно. Если их результат одинаков, в игре существует потолок навыка, которого игроки в конечном итоге достигнут. Если результат теоретика лучше, чем у любого другого человека, игру можно считать бесконечно глубокой, и у игроков никогда не закончится то, чему можно научиться.

Например, в шахматы лучше играет сборная команда «человек — компьютер», чем «человек — гроссмейстер». Иначе говоря, занимаясь шахматами всю жизнь, гроссмейстеры

Потолок навыка в игре — это такой уровень, выше которого не прыгнуть. Если этот уровень умения играть находится за пределами человеческих способностей, игра считается БЕСКОНЕЧНО ГЛУБОКОЙ и ее никто и никогда не сможет полностью разгадать.

не играют идеально. Они еще не достигли потолка навыков в этой игре, поэтому шахматы, вероятно, бесконечно глубокая игра.

Многопользовательский шутер Modern Warfare имеет очень высокий потолок игровых навыков. Управление точное, оружие смертоносное, действие быстрое. В других многопользовательских шутерах вам потребуется несколько секунд на то, чтобы убить врага, даже если вы не пропустите ни одного выстрела. Это накладывает верхнюю границу эффективности игрока, независимо от его опыта. В Modern Warfare, имея превосходную тактику и цель, можно за считанные секунды уничтожать целые команды. Люди не смогут достигнуть подобного совершенства, хотя в теории это возможно, поэтому игроки всегда могут наслаждаться опытом стремления к идеалу.

Боевая система в Assassin's Creed II имеет средний потолок навыков игрока. Превосходный игрок мог бы играть значительно лучше обычного игрока, но он не опережал бы его настолько сильно, как, например, в играх Modern Warfare. А все потому, что в боевой системе Assassin's Creed II присутствуют задержки в управлении. Анимация атаки часто длится секунду или две, в течение которых игроку недоступны никакие действия. Эти короткие задержки снижают потолок навыков, потому что дают нормальному игроку время мысленно догнать идеального игрока. Например, даже в случае идеального боя на мечах, чтобы убить 10 врагов, вам потребуется не менее 20 секунд, так как анимация убийства занимает около двух секунд. Подобный результат приходит к обычному игроку с опытом. Но как только он его достигнет, идти дальше некуда. Даже если игрок может думать быстрее, задержки анимации не позволяют ускорить процесс. Так что вопрос о навыке игры становится скучным.

Крестики-нолики и другие тривиальные игры имеют низкий потолок навыка. В этих неглубоких играх легко провести идеальную игру, если вы знаете несколько простых трюков. Между идеальным игроком и порядочным человеком нет никакой разницы. И опять игра становится скучной, как только ее разгадали.

Не каждая игра должна быть бесконечно глубокой. Assassin's Creed II имеет умеренный потолок навыка игрока, но все еще

остаётся отличной игрой, потому что это не просто игра с прогнозируемым исходом. В ней также присутствуют графика, разведка и нарратив. Её дизайнеры решили, что красота и доступность важнее, чем одна только глубина.

Доступность

Если глубина — это максимальный уровень навыка игрока, при котором игра остаётся интересной, доступность — это примерно тот минимум, при котором она становится пригодной для игры. Практически у всех игр и игрушек в наличии более низкий предел навыка, ниже которого эффективная игра невозможна.

Рассмотрим шутер от первого лица (FPS). Пока игрок не узнает, как двигаться, поворачиваться и стрелять, он не сможет играть в игры FPS. Для новичков в этом жанре это отпугивающий фактор. Нужны часы практики, чтобы изучить абстрактную связь между движениями на экране и кнопками управления. Этот барьер делает FPS игры недоступным для большинства людей.

Если разобраться, то почти во всех играх присутствуют границы навыка. Стратегические игры для ПК требуют, чтобы игроки могли использовать мышь и клавиатуру. Чтобы играть во многие настольные и видеоигры, игроки должны уметь читать, следовательно, маленькие дети, которые этому ещё не научились, не смогут играть. Ребёнок, который не может удержать предмет, не может играть в «Дженгу».

Геймдизайнеры часто недооценивают доступность, потому что мы настолько опытные, что не замечаем границ навыков. Но существует огромное количество потенциальных игроков, которые хотели бы играть, если бы только могли. Нужно потрудиться, чтобы привлечь их в свои ряды, оно того стоит.

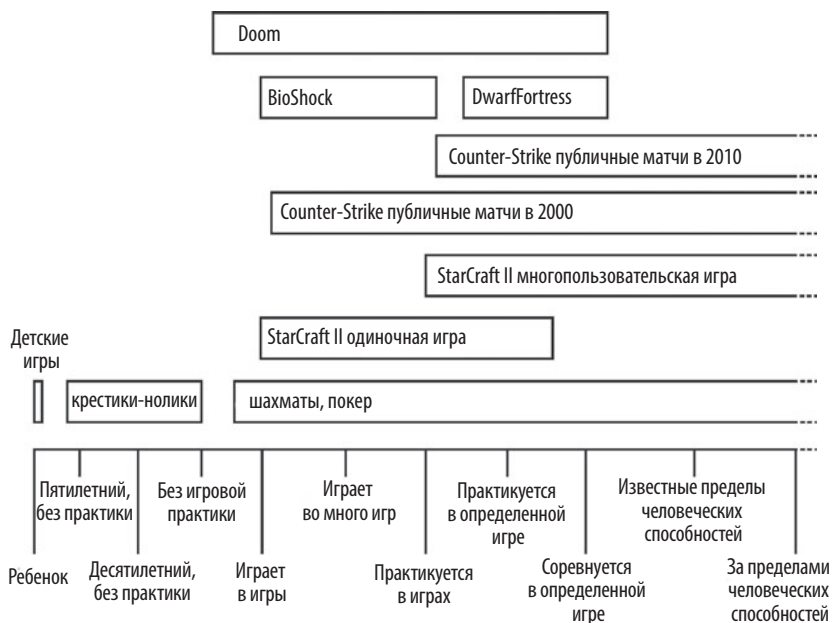
Диапазон навыков

Широкий диапазон навыков означает, что игрой могут наслаждаться как новички, так и эксперты. Игре легко научиться

ДОСТУПНЫЕ игры позволяют играть эффективно при низком уровне игрового навыка. ГРАНИЦА НАВЫКА игры — это нижний предел умения играть, за которым игра становится невозможной.

и трудно овладеть ею в совершенстве. И наоборот, узкий диапазон навыка указывает на то, что игроки быстро превращаются в экспертов после того, как только научились играть. Вы либо на все сто процентов знаете, как играть, либо вообще не знаете.

Диапазон навыка можно представить следующим образом:



В крестики-нолики могут играть даже маленькие дети, как только вы узнаете одну хорошую стратегию, становится скучно.

Таким образом, у этой игры узкий диапазон навыка и он полностью представлен в левой части таблицы.

Одиночная игра StarCraft II доступна всем, кто играл в видеоигры, потому что все начинается с простого урока с постепенным усложнением. Диапазон навыка для этой игры довольно широк, но он не бесконечно глубок — наличие заданных сценарием миссий означает, что в совершенстве игрой могут овладеть только преданные фанаты, после чего она перестает предлагать новые испытания.

ДИАПАЗОН НАВЫКА — это предел умений игрока, при которых игра представляет собой сложную, серьезную и интересную задачу.

При этом многопользовательскую игру StarCraft II можно с уверенностью назвать бесконечно глубокой. Независимо от того, как долго вы уже в нее играете, всегда есть чему поучиться. В то же время она менее доступна, чем однопользовательская игра, потому что уроки не расположены по принципу нарастания сложности, а игроки в сети беспощадно используют каждую ошибку оппонента.

Иногда диапазон навыков меняется в зависимости от контингента игроков. Например, в 2000 году я был монстром игры в Counter-Strike. Сегодня я изо всех сил стараюсь не опозориться. По сути, игра та же, а контингент игроков другой. Еще в 2000 году большинство пользователей были новичками в сетевых шутерах, а я был уже опытным игроком. На сегодняшний день большинство игроков Counter-Strike имеют многолетний опыт, а я — нет.

Игра не обязательно должна иметь широкий диапазон навыка, чтобы считаться хорошей. Например, BioShock обеспечивает сложные задачи для довольно небольшой группы игроков. Но опыт все еще работает, потому что BioShock рассчитан всего на 10 часов игры, а большая часть замысла передается через графику и нарратив. Эта игра не рассчитана на многолетнее обучение.

Достаточно сложно создать игру, которая будет либо глубокой, либо доступной. А одним из величайших испытаний дизайнера является разработка игры, которая будет одновременно и глубокой, и доступной, а также для нее потребуется широкий диапазон навыков.

Навык без определенных задач

В некоторых играх нет конкретных целей. Такие игры, как Dwarf Fortress, The Sims и Minecraft, дают игрокам возможность свободно исследовать, строить или взаимодействовать без каких-либо официальных побед или поражений. Может показаться, что эти «игрушечные» игры могут игнорировать вопрос навыка. Но даже у игрушек существует диапазон навыков, потому что они требуют некоторого минимального умения, прежде чем игроки смогут осмысленно с ними взаимодействовать.

Например, *The Sims* больше похожа на игрушку, чем на игру, потому что в ней нет заранее определенных целей. Игроки развивают созданную семью так, как им нравится. Но чтобы в принципе играть в эту игру, нужно уметь читать, пользоваться мышью и понимать интерфейс окон и кнопок. И чтобы понять значение *The Sims*, игроки должны иметь представление о западной жизни. Они должны знать, что такое спальня, телефон, домашняя вечеринка и внебрачные связи. Без этих базовых знаний они не поймут смысл игровых событий.

Помимо базового взаимодействия и понимания существует еще одна причина, по которой в игрушках присутствует требуемый диапазон применения навыков. Большинство игрушек не остаются таковыми надолго. Получая игрушку, большинство людей почти сразу же ставят перед собой цель. Ребенок, которому дали кубики, попытается построить как можно более высокую башенку. Ребенок с мячом попытается бросить его как можно дальше. То же самое относится и к компьютерным игрушкам. В *The Sims* игрок может захотеть заработать как можно больше денег. Когда он это делает, то больше не играет в *The Sims*. Он играет в «*The Sims: заработать как можно больше денег*», игру с прогнозируемым исходом своего собственного дизайна. Если эта недавно изобретенная игра тривиальная и неглубокая, дизайн провалился. Вот почему даже игрушки, в которых нет конкретной цели, определенно выигрывают от интересных, неочевидных опций, которые можно усвоить. Даже игрушки могут быть глубокими.

Расширяя диапазон навыков

Лучший способ расширить диапазон навыков — разработать простые и элегантные системы. Выжимая максимум глубины из каждой игровой механики, мы предоставляем облегченный вариант, которому можно легко научиться, но при этом им сложно овладеть в совершенстве. Это основная причина, по которой стоит попытаться создать элегантный дизайн.

Помимо создания элегантной игры существует множество других способов расширения диапазона навыков. Давайте рассмотрим некоторые из них.

Новые открытия

Мую первую зависимость от многопользовательских шутеров вызвала игра Unreal Tournament, которая вышла в 1999 году. Я увидел фантастическое кровавое сражение, в котором соперники кромсали друг друга чем-то похожим на детали от трактора. Персонажи были непредсказуемыми, а взрывы — колоссальными. Мне очень нравилось.

Через неделю игры что-то изменилось. Я перестал думать об агрессивных персонажах и видеть взрывы. Мой мозг начал снимать сюжетный слой, чтобы увидеть голую механику. Именно в ней я и нашел для себя сложную задачу: прицеливаться. Чтобы удержать прицел на цели, требовалось приложить все когнитивные усилия.

Впрочем, хорошенько потренировавшись, я развил мышечную память и смог целиться гораздо лучше, освобождая сознание для других задач. В более простой игре работать было бы не над чем. Я бы достиг потолка игрового навыка и вскоре потерял бы интерес.

Но игра Unreal Tournament снова преподнесла нечто новое и поставила передо мной еще одну задачу: контроль карты с целью удержания лучших предметов, а также обстрела чужих позиций. Поэтому я разработал «воображаемую» библиотеку картографических знаний. Я разобрался, как управлять силовыми точками и поддерживать тактическое преимущество. Мощнейшие взрывы теперь были просто визуальным шумом; теперь я едва ли обращал внимание на сюжет.

Я продолжал играть. Со временем я научился занимать лидирующие позиции без сознательных усилий. И снова мое сознание освободилось. И снова игра внесла новизну, открыв следующий уровень сложности: отслеживание других игроков. Теперь вместо того, чтобы просто знать, где находятся лучшие точки на карте, я должен был удерживать в голове воображаемую карту местоположения других игроков, а также направление, в котором они движутся, и все это в режиме реального времени. Если я ранил врага, то знал, что он вылечится. Если бой был на большом расстоянии, я знал, что враг возьмет снайперскую винтовку. В каж-

дом случае я старался оставить ему ловушку. Если мои прогнозы были верными, он шел прямо в мой прицел.

В конце концов, я научился прогнозировать движения. В том, чтобы знать карту, не было ничего сложного, а задачи прицеливания не имели особого значения. Но и игра Unreal Tournament на этом не заканчивалась. В ней снова появился новый элемент, окончательная, совершенная форма — психологические хитрости игры, подобной покеру. Я знал свои собственные возможности, знал опции соперника, а также способы взаимодействия каждой из этих опций. Противник тоже все это знал, и я знал, что он знал, и он знал, что я знаю. Когда перед обоими игроками открывается кристально чистый каркас механической игры и умение делать почти идеальный выбор, единственное, чем остается манипулировать, — это сам разум. Поэтому Unreal Tournament превратилась в вызывающие эмоциональные всплески, смешивание стратегий для возможности сохранять непредсказуемость, чтение мыслей противника до того, как он прочтет ваши мысли. Теперь вы усыпите бдительность врага и уничтожите его.

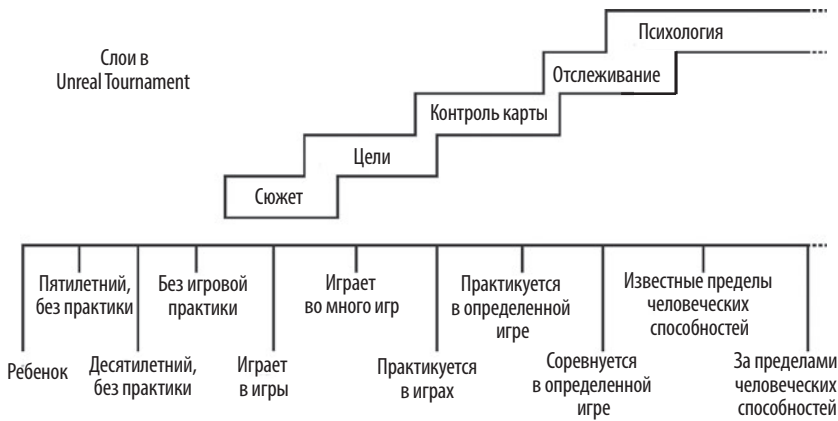
Я так и не справился с этим последним открытием, потому что оно бесконечно глубокое.

Ни одна игра не может долго растягивать один простой навык, чтобы получить широкий диапазон навыков. Игрок ограничен во

Игры расширяют возможности диапазона навыков, открывая новые возможности по мере повышения навыка игрока.

времени, которое он может потратить на прицеливание, перемещение, изучение экономических стратегий или замысловатых карт. Сложные игры вроде Unreal Tournament живут благодаря тому, что одна игра находится внутри другой, а та, в свою очередь, внутри третьей. Каждый раз, когда вы овладеваете каким-либо навыком в совершенстве, появляется еще один, более глубокий слой игры.

Таблица этих глубинных слоев представлена ниже. Непосредственно кроме навыка игрока в игре Unreal Tournament важен сюжет. Когда игроки достигают определенного уровня навыка, он меняется и уходит глубже, и это происходит до тех пор, пока игра по накалу не станет напоминать покер.



При механическом преобразовании задача заключается в простых, механических навыках, которые игрок выполняет постоянно. Если это шутер, то речь идет о наведении на цель и ее удержании. Если это игра в тетрис, то блоки должны попадать туда, куда вы хотите, и в том положении, которое вам нужно. Если это стратегия, речь идет о получении артефактов, которые вам нужны для конкретных целей. В шахматах это изучение правил ходов каждой фигуры. Механическое преобразование заключается в освоении интерфейса. Все игры начинаются именно с этого.

Диапазон навыков в игре предполагает три характерных преобразования: МЕХАНИЧЕСКОЕ, СИТУАЦИОННОЕ и ИНТЕЛЛЕКТУАЛЬНОЕ.

Ситуационные преобразования — второй уровень развития навыков игрока. На этом уровне механические действия в основном выполняются неосознанно. В файтинге игрок уже может произвольно наносить многокнопочные комбинированные удары. Он может целиться и поражать цели или заставлять своих персонажей двигаться так, как ему нужно. Задача на этом уровне заключается не просто в том, чтобы знать, как стрелять, а в том, чтобы знать, кого и когда застрелить, или в том, какие отряды отправлять в бой и куда именно. Большинство игроков находятся именно на этом уровне, и большинство игр разработаны для него. Он включает понимание происходящего, чтение схем, знание контрстратегий и многие другие навыки, необходимые, чтобы играть на среднем уровне. Этот уровень весьма широк; его можно разделить на несколько внутренних преобразований. Мой опыт в игре Unreal Tournament, как в случае с управлением

картами, так и с отслеживанием и прокладкой маршрута, был ситуационным преобразованием.

Интеллектуальный уровень развития навыков достигается нечасто и только опытными, напористыми и преданными игроками. Интеллектуальные умения в основном предполагают удержание концентрации и достижение эффективности. Если вы играете на высоких уровнях сложности, эмоциональное перевозбуждение или кратковременная потеря внимания могут привести к поражению. На подобных вершинах существуют тактики, специально предназначенные для того, чтобы преднамеренно ввести противника в заблуждение или отвлечь его, нарушив его концентрацию. Интеллектуальный навык — это все, что позволяет вам прогнозировать свои действия и управлять своим разумом лучше, чем это сделает ваш соперник. Большинство безгранично сложных игр в итоге заставляют вас чувствовать нечто похожее на состояние при игре в покер, и именно поэтому большинство глубоких и бесконечных игр сделаны как многопользовательские. Человек может освоить практически любую игровую систему, но он никогда не сможет полностью понять мысли другого человека.

Гибкие вызовы

Представьте себе версию игры в дартс, в которой мишенью было бы яблочко диаметром в 2,5 см. Если вы попали, то получаете одно очко, если нет, то не получаете ничего. Такая игра была бы настолько сложной, что потеряла бы всякий смысл для всех, кроме экспертов. Дизайнер мог бы изменить диаметр яблочка, но для большинства игроков мишень всегда будет слишком маленькой или слишком большой.

В реальности дартс решает эту проблему нанесением вокруг яблочка дополнительных колец, каждое из которых дает разное количество очков. Большинство людей могут попасть в самое большое кольцо и получить несколько очков, но только лучшие могут попасть в яблочко, таким образом, у каждого игрока существует какая-то сложная, но достижимая цель. Это и называется гибкой целью.

Дизайн, предполагающий игры типа «прошел/не прошел», предназначен только для игроков, которые с трудом справляются

с заданием. Если у игрока достаточно высокие умения, у него не возникает проблем. Если же игроку не хватает навыков, он неизбежно проигрывает. Поэтому игры типа прошел/не прошел подходят только для узкого диапазона навыков игрока.

Гибкие цели решают эту проблему, обеспечивая несколько уровней победы или поражения. Допуская различные вероятности успешного исхода, они дают возможность для более широкого диапазона навыков, поскольку у каждого есть достижимая, хотя и сложная цель.

Например, в классических аркадах обычно ведется детальный подсчет очков и эти игры предполагают наличие гибких целей. Любой, кто опустил четвертак, получает несколько тысяч очков, просто нажимая кнопки. Обладая достаточным мастерством и настойчивостью, игроки могут набрать сотни миллионов очков. Поэтому независимо от того, сколько раз они сыграют (и сколько монет будет опущено), всегда остается способ сыграть немного лучше.

Многообразные системы подсчета очков, как в дартс и аркадах, являются распространенной формой гибких целей, но существуют и другие системы.

Например, в играх Hitman задача каждой миссии — поразить цель и сбежать. Хитрость заключается в том, что игроков оценивают исходя из того, насколько аккуратно они убивают жертву. В самом убийстве нет ничего сложного — все, что вам нужно сделать, это уничтожить жертву с помощью пулемета. Но чтобы получить желанный высший рейтинг «Бесшумный убийца», игрок должен убить жертву без привлечения свидетелей или ненужного насилия, используя для этого комбинацию инструментов, маскировку, незаметность и умение стрелять без промаха.

И существует целый ряд других рейтингов, которые игрок получает на пути к рейтингу «Бесшумный убийца». Игрок, который не попадает, но при этом оставляет тела мертвых охранников, которых потом обнаруживают, конечно, считается более успешным, чем тот, который стреляет как сумасшедший во все стороны, однако он не так хорош, как идеальный лазутчик, которого никогда не обнаруживают.

Благодаря гибким целям игрок может позволить себе победу либо поражение и получить соответствующую задачу, которая охватит более широкий диапазон мастерства.

Если бы Hitman был игрой, в которой нужно только убивать, это был бы обычный шутер. Если бы для победы были нужны только «чистые» убийства, большинству игроков было бы невероятно сложно. Благодаря гибкой цели дизайнеры смогли объединить в игре лучшее, признавая и награждая любую степень скрытности и сообразительности, которой обладают игроки.

Так же как мы можем определить раз-
личную вероятность
успешного исхода, мы
можем определить
различную вероят-
ность поражения.

Например, если игрок не прыгает через яму, мы можем дать его персонажу возможность ухватиться за край и подтянуться. Его прогресс замедляется, но игра не прекращается.

В другой игре вместо того, чтобы союзник игрока из армии США погиб при атаке, его действия могут быть ограничены. Таким образом, игрок может оживить его, потратив на это некоторое время и ресурсы, а затем продолжить игру.

Диапазон побед и поражений может выходить далеко за рамки стандартных представлений, и при этом вовсе не обязательно объявлять победивших и проигравших и заканчивать игру. Любая дополнительная свобода, которую мы вкладываем в гибкое многообразие игры, означает, что проигрыш или скучный геймплей не разочарует еще одну группу игроков.

Обучение

Системы обучения помогают игрокам быстрее преодолеть границу развития игрового навыка. Этой цели служат встроенные в игровой мир пояснения, текстовые сообщения, аудиоинструкции и подсказки.

Но обучение может быть опасным. Некоторые плохо разработанные системы бомбардируют игрока всевозможными инструкциями, отвлекая его разум от игры. Другие системы подобны чрезмерной опеке родителей над ребенком: система постоянно сообщает игроку, что именно нужно делать в конкретный момент, заставляя его чувствовать себя беспомощным и под постоянным контролем. Все это вступает в конфликт с остальным опытом.

Лучшая система обучения учит так, чтобы игрок этого никогда не замечал. Существует несколько способов добиться этого.

Некоторые игры вплетают обучение в нарратив. Например, в Call of Duty 4 игрок играет от лица солдата, которого недавно перевели в элитное подразделение. По мере того как он преодолевает полосу препятствий, командир выкрикивает инструкции о том, как преодолеть каждое испытание. Эта полоса представляет собой вводную инструкцию, оформленную в виде сюжетной последовательности, и этот прием работает на обоих уровнях.

Хорошая система обучения ненавязчива.

Мы можем превратить обучение в гибкие цели. Серия обучений в Call of Duty 4 предназначена не только для подготовки к игре, но и для гонки на время. Игроки получают больше очков, если бегут быстрее. Таким образом, в то время как новичок может медленно пробираться по полосе препятствий, эксперт полностью проигнорирует аспект обучения и будет рассматривать полосу препятствий как вызов своему мастерству. Новичок учится тому, что ему необходимо, а эксперт получает задачу, которую он жаждет получить.

И все же лучший способ сделать обучение менее навязчивым — это пропустить его, если в нем нет необходимости. Хитрость заключается в том, чтобы определить, нужен ли игроку урок или нет. В некоторых играх разрешается добровольно пропускать обучение. Другие проверяют игрока, чтобы определить, какое обучение ему нужно. Некоторые игры используют адаптивное обучение — вместо того, чтобы учить всему по порядку, игра сама определяет, каких знаний не хватает игроку, и тут же дает соответствующий урок. Однако независимо от метода, принцип не меняется: самое незаметное обучение — это не обязательное обучение, которого не было.

Поддержка эмоциями

Даже при хорошей системе обучения многие игроки на самых ранних этапах игры находятся за низшей границей навыка. Прежде чем начинается настоящая игра, игроки должны пережить эти первые несколько минут или часов, которые становятся для них весьма непривлекательными. Многие сдаются и так и не переживают игру так, как она была задумана.

Если игрок ничего не знает, мы не можем создавать опыт вокруг решения головоломок, создания вещей или победы над врагами, потому что у игрока нет навыков, чтобы делать такие вещи.

Чтобы не дать игроку сдаться до того, как он преодолеет границу своего навыка, мы можем искусственно поддерживать его опыт с помощью эмоциональных триггеров, для которых не нужны особые умения.

Но существуют и другие эмоциональные триггеры, которые работают независимо от навыков. Мы можем показать игроку превосходную графику. Мы можем представить завораживающих персонажей. Мы можем дать игроку возможность шутить с друзьями. Мы можем добавить музыку или создать технологическую демонстрацию¹. Если мы наполним ранний опыт этими, не требующими навыков эмоциональными триггерами, то сможем превратить этапы раннего обучения из болезненной рутины в нечто вроде полуинтерактивной заставки к фильму.

Это хорошо видно на примере настольных игр, потому что в них играют в компании. Когда вы сидите за столом с друзьями, каждая ошибка или неправильно передвинутая фишка становится поводом для дружеских подколов. Изучать однопользовательскую настольную игру будет уже не так весело, потому что изучение правил на начальном этапе будет напоминать поход в библиотеку, а не веселый праздник.

Отличным примером видеоигры, обеспечивающей игроков эмоциями на ранних этапах, является BioShock. Игра начинается со слов: «Где-то в середине Атлантики, 1960 г.». Экран переключается на вид от первого лица, и игрок становится персонажем, сидящим в кресле самолета. Звучит несколько слов, экран меркнет, и мы слышим падение самолета. Герой открывает глаза — и мы под водой. Он с трудом поднимается на поверхность, задыхаясь, и в конечном итоге плывет в океане среди горящих обломков самолета. Вдалеке он видит призрачный маяк и статую ангела на вершине.

Игра заинтересовала игрока. Кто я? Как мне выбраться из воды? Что это за маяк посреди Атлантического океана и что за экзотические статуи? И поскольку вся эта последовательность событий не была интерактивной, игроку какие-либо навыки были не нужны.

¹ Tech demo — технологическая демонстрация, в данном случае отрезок игры, в котором мы демонстрируем ее возможности или особенности. — *Примеч. ред.*

Теперь игрок получает возможность управлять персонажем. Но ему не дается никаких предметов или оружия. Единственные элементы управления — это возможность смотреть и двигаться. И поскольку персонажу ничего не угрожает, игрок может не торопиться, исследуя местность. На данном этапе для игры требуется минимальный навык.

Он подплывает к маяку. Поднимаясь по лестнице из воды, он видит несколько богато украшенных медных дверей. Идти больше некуда, и он входит. Двери захлопываются, оставляя его в темноте. Мгновение спустя включается свет, и начинает играть старинная мелодия 1930-х годов. Игрок оказывается в гигантской камере в стиле ар-деко. Перед ним огромный медный бюст сурового человека. Внизу висит кроваво-красное знамя. Оно гласит: «Ни богов, ни королей. Только Человек».

Музыка, ар-деко, авиакатастрофа и намеки на глубокую философскую идею наполняют игрока опытом. И все это время игрок находится в движении, изучая основные элементы управления. Все, что могло быть непростым обучением («Жми “W”, чтобы идти!»), становится незабываемым опытом.

Так продолжается на протяжении всей игры. Она представляет собой один захватывающий художественный или нарративный опыт за другим, одновременно добавляя интерактивный фон. Вскоре игрок получает оружие. Затем первые в игре боевые заклинания — плазмиды. Позже он узнает о других видах оружия и плазмидов, улучшений, аудиозаписей, изобретений, взлома и о многом другом.

За несколько часов игрок становится профессионалом, объединяя множество инструментов для решения сложных головоломок и победы над жуткими врагами. Он сам не понял, как все это узнал, потому что был слишком занят графикой и исследованием мира.

Изменение сложности игры

Изменение сложности означает изменение уровня сложности задач в игре в целом. На графике диапазона мастерства это приводит к смещению всего диапазона влево или вправо.

Изменить сложность игры можно несколькими способами.

Выберите уровень: легкий, средний или сложный? Этот вопрос десятилетиями задается в играх. И не зря. Это облегчает дизайн, делая его эффективным и элегантным.

Проблема заключается в том, что выбор сложности сам по себе может приводить в замешательство. Не существует единых стандартов для простых, средних и сложных уровней,

При эксплицитном выборе игрокам предлагается решить, на каком уровне сложности они хотят играть.

поэтому игроки опасаются, что сделают неправильный выбор. Чтобы облегчить выбор, некоторые игры проверяют игроков на начальном этапе и рекомендуют соответствующий уровень сложности. Другие описывают уровни сложности на примерах или позволяют изменить решение во время игры, если игрок изначально сделал неправильный выбор.

Если игрок умирает с голоду, мы можем незаметно подложить еду в соседнюю комнату. Если он слишком долго играет в одиночку, мы можем «случайно» ввести союзника, который придет ему на подмогу. Каждый раз, когда он терпит неудачу, мы можем тихонько снижать уровень сложности, делая врагов слабее и медленнее. Каждый раз, когда он побеждает, мы усложняем задачу, делая врагов менее уязвимыми и более агрессивными.

Адаптивная сложность наиболее эффективна, если игроки о ней не знают. Игроки, которые о ней узнали, иногда пытаются обойти систему. Или начинают приписывать ей каждое случайное событие и считать, что их игровой опыт — фальшивка. Адаптивная сложность появляется чаще, чем вы думаете, потому что дизайнеры никогда ее не афишируют.

АДАПТИВНАЯ СЛОЖНОСТЬ незаметно регулирует уровень игры в зависимости от того, насколько хорошо играет игрок.

Адаптивная сложность подходит только для игр, в которых от игроков не ожидается очень высокого уровня владения навыком. Опытные игроки выяснят, как все устроено, и начнут использовать ее в своих целях. Для таких игроков должны создаваться максимально честные игры, потому что эксперты могут легко определить, когда игра использует свои трюки с целью помочь или помешать. Они быстро взломают систему и будут использовать ее в своих интересах.

Адаптивные и эксплицитные сложности могут сосуществовать. Например, в Resident Evil 5 присутствуют четыре эксплицитные установки сложности игры, от простого до профессионального. На внутреннем уровне они подразделяются на 10 адаптивных установок сложности от 1 до 10. Каждый раз, когда персонаж умирает, внутренняя сложность снижается, и так далее, пока не достигнет минимума. Но когда вы выигрываете, внутренняя сложность увеличивается. Однако система может адаптировать внутреннюю сложность только в пределах, определенных эксплицитным выбором сложности, который сделал игрок. Например, игра в простом режиме может незаметно переходить на разные уровни сложности от 1 до 4, а игра в обычном режиме — от 3 до 7. Для решения проблемы, связанной с опытными игроками, манипулирующими системой адаптивной сложности, в игре существует специальный режим сложности, который оставляет внутреннюю сложность на отметке 10. Эта комбинированная система дает Resident Evil 5 очень широкий диапазон мастерства, поскольку она может адаптироваться по мере необходимости как под новичков, так и под экспертов.

Если одни стратегии простые, а другие сложные, игроки выберут те, которые будут соответствовать их уровню навыков. Они выбирают свой уровень сложности, но делают это имплицитно.

Team Fortress 2. В этой игре девять классов персонажей. Играя за Снайпера, вам потребуется навык точного прицеливания, а при игре за Инженера или Медика подобные навыки не нужны в принципе. Игроки, как правило, классифицируют себя по навыкам: те, кто не умеет целиться, выбирают Инженера и Медика, в то время как эксперты по стрельбе чаще играют за Снайпера. Таким образом, игроки выбирают степень и тип задач, которые они хотят, не открывая несуразное окно выбора сложности. И в отличие от эксплицитного выбора сложности, эта система хорошо работает в состязательных многопользовательских играх.

При имплицитном выборе уровня сложности игрокам дается возможность регулировать уровень решаемых ими задач, принимая стратегические решения.

Call of Duty 4. В однопользовательских кампаниях игрок проходит через линейные уровни с невидимыми триггерами, если персонаж не наступит на нужный триггер, игра будет генерировать толпы вооруженных врагов и бой будет вестись бесконечно.

Опытные игроки ведут агрессивный бой, активно продвигаясь вперед, и склонны активировать несколько триггеров подряд. В итоге они сражаются с несколькими группами противников одновременно, что приводит к достаточно ожесточенным баталиям. Неопытные игроки продвигаются неуверенно и медленно, расчищают местность перед тем, как идти вперед. Так как они разделяются с каждой группой врагов до того, как появится следующая, у них гораздо меньше шансов оказаться в окружении или быть разбитыми. Выбирая между быстрым или медленным темпом продвижения, каждый игрок непрерывно регулирует собственный уровень сложности в зависимости от своих предпочтений.

Что делать с проигравшими

Напряжение. Рот приоткрыт, глаза прищурены, вы делаете вдох и задерживаете дыхание. Это чувство бесценно. Оно пробуждает, заставляет сосредоточиться, готовит к наступающему порыву облегчения, триумфа или разочарования. Чтобы вызвать это чувство, мы должны поставить под угрозу какую-то человеческую ценность. Что-то важное — жизнь, победа, богатство, что-нибудь должно висеть на волоске. Возможность успеха должна быть реальной, как и вероятность поражения.

Но вот в чем парадокс. Игроки играют в игры, чтобы удовлетворить свой опыт, а наличие напряжения означает, что они могут страдать. Но как можно и рыбку съесть, и в воду не лезть? Как мы поступаем, если игроки терпят поражение?

Хитрость в том, чтобы знать, кого и как именно наказать.

Некоторые игры наказывают за неудачу самого игрока. Так как они не могут навредить ему физически, то причиняют ему боль только так, как могут: заставляя его играть в плохую игру. Они заставляют ждать, пока загрузится экран, переигрывать раунды снова и снова или гриндить¹, чтобы восстановить потерянные ресурсы. Ни один дизайнер не создал бы подобное намеренно

¹ На языке игроков и геймдизайнеров «гриндить» (от англ. to grind) — повторять однообразные действия. — *Примеч. ред.*

в любом другом контексте. Но эти меры странным образом оправдываются, если они следуют сразу после поражения. Это в корне неверно. Несмотря на то что длительная загрузка экрана может создать напряжение, не стоит заставлять игрока снова и снова проходить это мучительное испытание. Мы должны найти другие способы создания реальной угрозы, не причиняя при этом вреда самому игроку такими мелкими и некомпетентными способами. К счастью, есть альтернативы.

Если мы не можем причинить вред самому игроку, то можно разобраться с его персонажем. Главного героя игры можно запросто расстрелять, расторгнуть его брак, разорить, зарезать, сжечь или подвергнуть мучениям тысячами других способов. Крутой образ персонажа может стать причиной того, что эти вымышленные последствия окажутся для нас достаточно значимыми, чтобы создать напряжение.

Не наказывайте за поражение самого игрока. Найдите другие способы заставить его понервничать.

Если этого недостаточно, мы можем не принимать победу игрока или создавать незначительные препятствия. Игрок может лишиться некоторых ресурсов или союзника, пропустить небольшой фрагмент нарратива или потерять доступ к части информации. Но в каждом случае игра может развиваться в новом направлении. Все в норме до тех пор, пока игра не превращается в многократные повторные попытки.

StarCraft II. В однопользовательской кампании проигранный бой означает потерю части персонажей в более широком стратегическом сценарии. После поражения игрок обычно может восстановиться и начать еще одну атаку. Игра не началась сначала, а игрок не повторяет миссию. Скорее он пытается в текущей ситуации зайти с другой стороны.

Super Meat Boy. В этой игре в жанре «платформер» поражение — повсеместная, неотъемлемая часть игры. Meat Boy часто «живет» менее 10 секунд. Но каждый раз, когда его измельчают на куски или поджаривают на огне, он появляется через секунду, готовый снова рвануться в бой. Скорость воскресения Meat Boy означает, что поток никогда не прерывается. Можно умереть сто раз подряд, потому что игра никогда не останавливается. Смерть в Super

Meat Boy не болезненна для игрока, но у него по-прежнему есть мотивация избежать ее и гордиться собой, если ему это удалось.

Dwarf Fortress. В этой игре поражение, по сути, является полезным опытом. Прежде чем окончательно обрушиться на толпу гоблинов, топильные камеры¹ и несчастных трезвых dwarфов, разрушающаяся крепость переживает ряд забавных взрывов. Dwarфы ужасно мучаются, но игрок наслаждается каждой минутой этого зрелища потому, что игра всегда движется вперед к новому и интересному опыту.

Капкан поражения

Иногда игры наказывают игроков гораздо сильнее, чем планировалось.

Классический капкан поражения появляется в спорте. Если одна футбольная, бейсбольная или хоккейная команда сильно отстает от другой по очкам, она становится командой ходячих мертвецов. Шансов на победу нет, но они все равно должны доиграть матч для галочки и закончить игру. Полчаса таких телодвижений могут быть весьма мучительными.

КАПКАН ПОРАЖЕНИЯ — это игровое событие, в котором игрок долгое время находится в ситуации, которая заведомо обречена на провал.

Эта же игровая ситуация может проявляться в сотне разных вариаций практически во всех жанрах. Если вы сильно отстали в гонках, скорее всего, вы уже не догоните команду. В линейной однопользовательской игре, если игрок не может победить в одном бою, игра стоит на месте.

Наиболее опасными капканами считаются те, которые возникают в трудно прогнозируемых ситуациях. Например, если игрок однопользовательского шутера попадает в контрольную точку, не имея при этом каких-либо боеприпасов, все может закончиться бесконечной серией перезагрузок в этой точке, поскольку персонаж будет пытаться выполнить невыполнимое. Единого способа выбраться из этого капкана не существует. Каждая проблема требует индивидуального решения в зависимости от того, как и почему она появилась.

¹ Сленговое название специальных комнат, которые в случае нужды можно быстро затопить водой. Обычно конструируются игроками для обороны. — *Примеч. ред.*

Иногда капкан бывает не очень крепким, и тогда проблему можно решить. Например, игра экшн может включать слабое запасное оружие с бесконечным боезапасом или минимальное количество гарантированного здоровья, которое получает персонаж после воскрешения. Эта механика возвращает игрока на этап, когда задача сложная, но еще выполнимая.

Иногда, чтобы избежать попадания в капкан, можно просто завершить игру. Например, соревновательные стратегии могут создавать механизмы, которые дают игрокам возможность сдаваться, если они чувствуют, что попали в капкан поражения. Завершение игры позволяет им перейти к следующей быстрее. Однопользовательские игры решают проблему капканов, давая игрокам подсказки в сложных ситуациях, корректируют сложность без ведома игрока или предлагают альтернативные задачи. Например, в *Super Meat Boy* каждый мир состоит из 20 уровней, но игрок может переходить в следующий мир, пройдя только 18 уровней. Если игрок застрял на одном или двух уровнях, он может переходить к следующему, а затем вернуться к ним позже.

К сожалению, не каждый капкан поражения можно устранить без последствий. Некоторые из них настолько глубоко внедрились в дизайн, что избежать их просто невозможно. Например, никто никогда не устранял капкан поражения в гонках и спортивных играх.

Нарратив

«Записывайтесь в Школу поддержки персонажей видеоигр, — говорили они. — Вы поможете миллионам игроков».

Да, конечно. Все, что я делаю, это стою и повторяю одно и то же снова и снова. Иногда игроки что-то в меня бросают или стреляют в меня ради собственного извращенного удовольствия. Почему они не могут меня просто обнять?

Даже если я помогу игроку сражаться, я получу лишь несколько бесполезных горохострелов. Как будто они и не хотят, чтобы я сделал доброе дело. Почему игрок всегда становится героем? Почему не кто-нибудь из нас? ПРОСТО для разнообразия.

Хуже всего то, что я не могу умереть. Как только они поймут, что я бессмертный, некоторые из этих бездушных игроков будут прятаться за мной, а пули будут пронзать мое тело. И все, что я могу сделать, это кричать, снова и снова, на один и тот же манер; заезженная пластинка моих страданий.

Настал час расплаты. Я записываюсь в Школу антагонистов.

Когда мы приближаемся к любой нестандартной задаче, то естественным образом начинаем думать о ней с точки зрения имеющегося у нас опыта. Нарратив в игре — одна из таких новых задач. Краеугольным камнем является то, что нарратив почти всегда упоминают только в фильмах.

Аналогия между фильмами и видеоиграми очевидна: и там и там используются движущееся изображение и звук для передачи информации через экран и колонки. Итак, разработчики игр нанимают голливудских сценаристов.

Они выстраивают игру вокруг трехактной структуры, написанной одним автором. Они даже делят процессы разработки на три части, как это делается и в фильмах: предпродакшн, продакшн и постпродакшн. Часто можно услышать, как гордятся этим копирующим киноиндустрию шаблоном: мы постоянно слышим

об играх, пытающихся быть все более и более кинематографичными. Но присутствует проблема.

Фильм учит тысяче способов использования экрана. Фрейминг и композиция, построение сцены, темп, визуальные эффекты — все это можно узнать из фильма. Но фильм не учит ни интерактивности, ни выбору, ни опыту в реальном времени. Не говоря уже о том, чтобы заставить игроков чувствовать страх принятия трудного решения. В фильме ничего не говорится о том, что делать с игроком, который решает пойти против сценария. В фильме никто не станет описывать игроков *The Sims*, которые пишут в реальных блогах о ежедневных приключениях своих смоделированных семей. Эти ситуации полностью выходят за теоретические основы фильмового сторителлинга. Полностью заимствуя методы геймдизайна из фильмов, мы рискуем не заметить наиболее сложных задач и возможностей сюжета компьютерной игры.

Хотя игры и похожи на фильмы, они не работают так же.

К счастью, отказ от идей киноиндустрии не значит, что нужно все начинать с нуля. Существует много старых форм интерактивного сторителлинга, откуда можно черпать вдохновение, если только посмотреть за пределы светящегося экрана.

Однажды я участвовал в интерактивной пьесе под названием *Sleep No More*. Это была версия шекспировского «Макбета», действие пьесы проходило не на сцене, а в заброшенной средней школе, пьеса представляла собой смесь водевиля 1920-х годов и сюрреализма в духе Дали. Согласно сценарию двухчасовой пьесы актеры бродили по залам, встречаясь и взаимодействуя друг с другом, иногда разыгрывая монологи сами с собой, танцевали, говорили, спорили и сражались. Зрители в масках могли свободно перемещаться по пространству, наблюдать за тем, что казалось им интересным, но при этом увидеть за раз больше, чем одну часть истории, было невозможно. Иногда актеры даже приглашали зрителей поучаствовать в отдельных действиях. Это был интерактивный нарратив.

Существует много других видов традиционных интерактивных сюжетов. Возможно, вы уже знакомы с интерактивными ин-

сталляциями по истории. Декорации места действия воссоздают деревню с деревянными домами или окопы Первой мировой войны, а актеры в костюмах играют роли жителей. Посетители могут задавать вопросы, исследовать пространство и, возможно, участвовать в действии.

Если оглянуться вокруг, то повсюду нас окружает интерактивный нарратив. Музеи и художественные галереи — это интерактивные, нелинейные нарративы, где посетители знакомятся с историей или художественными направлениями. Древние руины и городские граффити рассказывают сюжеты. Для следователя даже место преступления служит своего рода естественным интерактивным нарративом, история, написанная кровавыми мазками, гильзами и битым стеклом.

И за всем этим стоят истории, которые мы пережили. У нас у всех скрыты те или иные пережитые события, которые невозможно воссоздать никоим образом. Мы можем написать о них книгу или пересказать словами, но никто и никогда не сможет повторить этот опыт, чувствуя ту же неопределенность, решимость и получая тот же результат, что и рассказчик. Услышанная история — это не то, что вы пережили сами.

Эти интерактивные шаблоны — музеи, галереи, пространства и жизнь — должны быть нашими эталонами при поиске нарративных инструментов. Эти проверенные формы решают нашу самую фундаментальную задачу: создать гибкую историю, которая приобретает ту или иную форму в зависимости от выбора игрока, наделяя его действия еще более глубоким смыслом.

Нарративные инструменты

Из этой книги вы не узнаете, что создает хорошую историю. Со времен аристотелевской «Поэтики» на эту тему уже писали гораздо более подкованные авторы. Они уже объясняли, как создать сюжет с интересными поворотами и хорошим темпом повествования. Они описывали, как создать реалистичных, многослойных персонажей. Писали о теме, месте действия и жанре. Я не специалист по сторителлингу; я сомневаюсь, что мне есть что добавить к этому огромному корпусу знаний (несмотря на то

что геймдизайнеры должны понимать эти концепции, в конце книги я все же оставил несколько ссылок на источники).

В этой главе рассматриваются инструменты, которые используются в играх для рассказа истории, потому что именно они изменили геймдизайн за последние 2300 лет.

Большинство историй в медиа ограничены небольшим набором инструментов. Автор комиксов получает в свое распоряжение облачко с текстом и четырехцветную графику. Режиссер получает 24 кадра в секунду и стереозвук. Писатель получает возможность поместить рассказ в 90 000 слов. Экспонат получает место в музее, информационные панели, диорамы и, возможно, несколько интерактивных игрушек.

НАРРАТИВНЫЙ
ИНСТРУМЕНТ —
это какой-либо прием, используемый для формирования фрагмента истории в сознании игрока.

Игры немного шире. Как и в фильме, в играх можно использовать заранее установленные последовательности изображений и звуков. Как и в романе, в играх можно использовать письменный текст. Как и в комиксах, в играх можно создавать графику и дать возможность ее быстро просмотреть. Как и в музее, в играх можно создать пространство, которое игроки будут изучать и исследовать. А еще у нас инструменты, которых нет больше ни у кого: мы можем создавать механику, которая моментально генерирует сюжет, персонажа и даже тему, и все это в зависимости от решения игроков.

Наши нарративные инструменты делятся примерно на три основных вида: сценарий, нарратив о мире и появляющийся в результате игры сюжет.

История, заданная сценарием

Инструменты, которые больше всего напоминают старые приемы, — это истории, заданные сценарием.

Самый базовый инструмент сценария — это внутриигровое видео, или кат-сцена. Кат-сцены позволяют использовать трюки, которые знакомы нам по фильмам. К сожалению, кат-сцены неизбежно нарушают поток, потому что прерывается вся интерактивность. Избыточное количество кат-сцен задает прерыви-

стый темп игры с бесконечными стартами и остановками, поскольку игроку приходится постоянно переключать внимание между геймплеем и кат-сценой и наоборот. Не обязательно критично — кат-сцены могут служить хорошей передышкой в промежутках между интенсивной игрой, но они всегда раздражают.

Гибкий сценарий

Одним из видов истории, заданной сценарием, является задание лишь последовательности сюжетных событий. В этом случае предварительно заданные события воспроизводятся без полного отключения персонажа.

ИСТОРИЯ игры, ЗАДАННАЯ СЦЕНАРИЕМ, — это события, прописанные непосредственно в коде игры, поэтому они всегда развиваются одинаково.

Например, идя по аллее, персонаж игрока может быть свидетелем убийства. Каждый крик и удар убийства предварительно записаны и анимированы, поэтому убийство всегда будет происходить одинаково, если игрок идет по этой аллее впервые. Однако при этом действия персонажа игрока, ставшего свидетелем убийства, не прописаны. Игрок может пройти мимо, остановиться и смотреть или развернуться и убежать. Это называется гибким сценарием.

ГИБКИЙ СЦЕНАРИЙ дает возможность игроку сохранять интерактивность даже при воспроизведении последовательности, заданной сценарием.

Преимущество данного подхода заключается в том, что поток остается непрерывным, поскольку игрок продолжает использовать джойстик. Недостатком является ограниченный контроль у дизайнера. Игрок мог бы наблюдать за убийством, полностью пропустить его или даже помешать ему. Что произойдет, если вы застрелите убийцу? Или жертву? Или прыгнете им на головы? Или вас что-то отвлечет и вы все пропустите?

Каждая последовательность, заданная сценарием, должна уравнивать влияние игрока и воздействие дизайнера; для этого существует целый ряд способов. Какой из них выбрать, зависит от сюжетного события и основной механики игры. Вот наиболее эффективные из них (расположены по принципу от наиболее контролирующих до предоставляющих игроку максимальную свободу):

Half-Life. Вначале игрок проезжает на подвешенной вагонетке через исследовательскую корпорацию «Черная Меза». Через

несколько сцен мы видим стучащего в дверь запертого охранника, роботов, перевозящих опасные материалы, а в это время записанный голос зачитывает прозаические детали жизни объекта. По мере поездки вглубь мы понимаем, насколько большим и опасным на самом деле является это место. Игроку дается возможность ходить по вагонетке и смотреть в разные окна, но он никак не может повлиять на что-либо.

Dead Space 2. В этой научно-фантастической игре в жанре хоррор игрок идет по проходу вагона метро, едущего по рельсам, прикрепленным к потолку. Когда вагон набирает скорость, рельсы не выдерживают и кабина обрывается. Главный герой скатывается по проходу, и обычные средства управления движением персонажа недоступны. Тем не менее у игрока сохраняется возможность управления стрельбой. Монстры прорываются в двери и окна, пока персонаж катится через несколько вагонов, и чтобы выжить, игрок должен вовремя в них выстрелить. Эта последовательность значительно отличается от привычного темпа *Dead Space 2*. Она блокирует часть возможностей управлять движением игрока для того, чтобы создать особый, авторский опыт, но в то же время поддерживает поток, оставляя большую часть интерфейса без изменений.

Halo: Reach. Этот шутер от первого лица имеет систему, которая кодирует заранее определенные тактические подсказки для управляемых компьютером персонажей. Эти записанные подсказки заставляют врагов стремиться к выполнению определенной тактики, но все же позволяют им реагировать на атаки игрока. Например, такая подсказка может заставить врагов оставаться в задней части помещения, но все же при этом им будет дана возможность самостоятельно стрелять, захватывать укрытия, уклоняться от гранат и наносить удары по игрокам, находящимся слишком близко. Дизайнеры используют эти подсказки для создания стратегических движений более высокого уровня, в то время как ИИ обрабатывает мгновенные тактические реакции на поведение игрока.

Существуют также способы создания событий, которые не подвержены воздействию. Игрок в определенное время может получить почту в почтовый ящик персонажа. Письмо может поступить

в почтовый ящик персонажа игрока в конкретное время. Объекты или персонажи могут появляться или исчезать, пока игрок находится в другом помещении. В это же время могут играть радио и громкоговорители. Эти методы популярны ввиду того, что они эффективны, менее затратны и не требуют тщательного индивидуального дизайна пользовательской полуинтерактивной последовательности, заданной сценарием.

Нарратив о мире

Когда-то, оказавшись в Лондоне, я сильно страдал от джетлага. Бродя по Южному Кенсингтону в 5 часов утра, я обнаружил, что город рассказывает мне истории. Его узкие извилистые улочки рассказывают свою давнюю историю до эпохи появления города. Магазины, церкви и дома рассказывали мне, как жили разные слои общества, насколько они были богаты, во что верили как в прошлом, так и в настоящем. Великие музеи и памятники отражали британскую историю и культурные ценности. Они говорили о своем величии, своей архитектуре, своих трудах и даже своих именах: музей под названием «Виктория и Альберт» рассказывает о гордом периоде монархического правления. Город даже рассказал мне о вечеринке прошлой ночью — о ней свидетельствовали пара порванных колготок и разбитый пивной бокал.

Все места рассказывают истории. Мы можем исследовать любое пространство и узнать его людей и его историю. Геймдизайнеры могут пользоваться этим, чтобы рассказать историю, поместив ее в пространство. Я называю это нарративом о мире.

Представьте, что вы идете по замку, построенному королем, одержимым войной, или по дому торговца наркотиками в гетто, или по дому супругов, женатых уже 50 лет. Вы можете исследовать

НАРРАТИВ О МИРЕ — это история места, его прошлого и людей. Ее рассказывает архитектура местности.

пространство как детектив, осматривая конструкцию и расположение объектов, копаясь в ящиках в поисках фотографий, документов и аудиозаписей. Присмотритесь внимательно, и вы сможете собрать воедино историю, событие за событием, вплоть до настоящего времени. Вы узнали о сеттинге, персонажах и сюжете, не читая ни слова истории и не увидев ни одного из героев.

Нарратив о мире не ограничивается холодной историей. Как и любые другие нарративные инструменты, он может выражать как информацию, так и чувства. Тюрьмы, дворцы, дома, холмистая местность — все эти места несут как эмоциональный, так и информационный заряд. Они работают через эмпатию — каково это было жить здесь, и грубые эмоции от окружающей среды — одинокая пустынная тундра.

Приемы нарратива о мире

На самом базовом уровне нарратив о мире рассматривает наличие или отсутствие определенных особенностей в среде, которая предполагает ту или иную ситуацию или историю. Наличие крепости говорит о том, что в древние времена город был подвержен военным нападениям. Тайный бордель указывает на строгие нравы, которые при этом нарушаются. Чтобы передавать информацию через ассоциации, в нарратив о мире могут быть включены культурные символы. Архитектура в римском стиле ассоциируется с гладиаторами, империей, завоеваниями и богатством. Темный неоготический вид Мордора в фильмах «Властелин колец» заставляет нас думать о черной магии и фантастических существах. Кроме того, существует огромное количество подобных ассоциаций. О ком вы думаете, рисуя граффити на кирпичных стенах? Или иглу? Или крошечный монастырь на вершине горы?

Мы можем рассказать о недавно произошедшем, упорядочив следы конкретных событий. Это называется мизансценой, от театрального термина, означающего «расположение на сцене». Трупы со связанными руками, повалившиеся на изрешеченную пулями стену, указывают на то, что произошла казнь. Если тела истощены и одеты в лохмотья, возможно, это геноцид. Если они одеты в королевские одежды, возможно, произошла революция.

Нарратив о мире также может быть выражен через документы. Например, мир Deus Ex был усеян КПК, каждый из которых содержал небольшой фрагмент текста, оставленный людьми, которые живут своей жизнью. Один особенно интересный КПК рассказывает о жизни новобранца в террористической организации, когда тот путешествует по миру на шаг впереди игрока,

по другую сторону закона. Когда игрок находит каждый КПК, скорее всего, оставленный несколько минут или часов назад, он узнает молодого новобранца-террориста, даже не взаимодействуя с ним.

Аудиозаписи делают то же самое, но вместо текста вы слышите голос. Голос — это мощный инструмент, потому что он позволяет нам слышать эмоции персонажей. Он также может передавать то, что невозможно передать с помощью текста, например разговоры между персонажами или записи природных событий, как в случае теракта в канун нового года в BioShock. И, услышав голоса персонажей, мы можем узнать их, столкнувшись с ними лично.

Видеозаписи выводят концепцию на новый уровень. Они могут представлять собой циклическую запись или находиться в кинопроекторе, готовом к воспроизведению. Они открывают большие возможности, чем аудиозаписи. Мы можем рассказывать истории с помощью телевизионных программ, новостных передач, пропагандистских фильмов, домашних видеороликов и видеозаписей с камер наблюдения.

Некоторые нарративные инструменты находятся на грани между миром и историей, заданной сценарием. Новости, транслируемые через динамики, брошенная прохожим агитационная листовка или городской глашатай — все это несет информацию о нарративных событиях. Эти вещи происходят в настоящем времени так же, как и история, заданная сценарием, но они передают особенности мира, а не являются элементами сюжета сами по себе.

Нарратив о мире и интерактивность

В более традиционных медиа нарратив о мире часто используется как добавление среди других, более непосредственных способов сторителлинга. В играх нарратив о мире является основным инструментом, потому что он решает ряд ключевых проблем, обусловленных интерактивностью.

Когда мы пытаемся рассказать историю в настоящем времени, нам приходится учитывать все, что мог бы делать игрок. Это требует определенных ограничений на его действия, непред-

виденные обстоятельства и адаптивную историю, и все это сложно и дорого. Нарратив о мире полностью решает эти проблемы, поскольку игроки не могут вмешиваться в историю, которая уже произошла. Если вы стали свидетелем убийства в переулке, вы можете застрелить убийцу, застрелить жертву или наброситься на них, и игра должна учесть или запретить эти действия. Если вы натолкнетесь на ту же сцену через полчаса после убийства, вы можете запрыгнуть на тело или выстрелить в него, но это не повлияет на историю, написанную автором. Иначе говоря, нарратив о мире останется без изменений.

Далее, нарратив о мире не нужно рассказывать линейно. Это избавляет нас от необходимости отправлять игроков на определенный путь. Например, представьте следующий нарратив: двое влюбленных подрались, один убил другого и закопал его на заднем дворе. Если рассказывать эту историю через нарратив о мире, то на следующий день уже будет неважно, обнаружит ли игрок сначала труп или окровавленную спальню. Если он увидит и то и другое в любом порядке, он сможет собрать воедино всю картинку. Это означает, что геймдизайнер может позволить игроку свободно исследовать дом. Для того чтобы показать ту же историю в запланированных событиях, необходимо, чтобы дизайнер придумал какой-то трюк или ограничение, чтобы игроки следовали по правильному пути и видели все события в правильном порядке.

Последнее большое преимущество нарратива о мире состоит в том, что он помогает тем, кто играет повторно, потому что не всегда полностью раскрывает себя в первый раз. Тогда как истории, заданные сценарием, раскрывают себя постепенно, событие за событием, от начала до конца, нарратив о мире естественным образом раскрывает себя в порядке от общего к частному. Вспомните сцену убийства наших любовников. Представьте, что во время своей первой игровой сессии игрок, прежде чем идти дальше, замечает только тело и залитую кровью спальню. Когда он повторяет игру, он замечает документы о разводе, которые предлагают мотив. В третьем, четвертом и пятом подходах игрок обнаруживает оружие убийства, письма любовника и любовницы

Нарратив о мире выполняет в играх полезную функцию, потому что он предотвращает многие проблемы объединения запланированных событий с интерактивностью.

и аудиозапись, где один любовник жалуется другу по телефону. Даже во время первой игровой сессии история завершена, потому что игрок от начала и до конца знает, что произошло. Но повторная игра показывает детали, которые заполняют недостающие части, отвечая на вопросы почему и как.

Логичность мира

Хорошо выстроенный вымышленный мир — это загадка взаимоотношений и работающие импликации. Он неуклонно следует своим собственным правилам, но полностью раскрывает воз-

Нарратив о мире усиливается, когда мир становится более целостным и выражает больше внутренних связей.

можности, которые он подразумевает. Каждый наблюдаемый факт о мире соответствует всем остальным. Эта сеть импликаций простирается за пределы того, что на самом деле присутствует в истории. Вот почему лучшие вымышленные миры, такие как «Звездные войны» и «Властелин колец», а также BioShock и The Elder Scrolls, отличаются огромным количеством нарративного контента, который подразумевается, но никогда не демонстрируется.

И наоборот, нелогичный мир — это смешение разобщенных деталей. Эти детали могут быть интересны по отдельности, но они не взаимосвязаны. Без взаимосвязи мир становится похож на кучу страниц, вырванных в случайном порядке из сотен комиксов: красивые картинки и забавные слова, образующие большую бессмысленную структуру. Пустышка в красивой обертке. В несвязном мире нет глубины, импликаций и элегантности. Игрок не может психологически шагнуть в мир и представить, как он перемещается и меняет его, потому что в мире недостаточно смысла.

Задача создания целостного мира заключается в понимании всех его внутренних взаимосвязей. Каждый предмет должен соответствовать всем остальным на нескольких уровнях — историческом, физическом и культурном.

Например, в Dead Space 2 главный герой Айзек Кларк находит устройство под названием «Кинезис», используя которое игрок может перемещать и бросать объекты, находясь при этом на расстоянии. «Кинезис» используется для решения головоломок

с помощью движущихся машин, а также для того, чтобы победить врагов, бросая в них предметы. «Кинезис» — хорошая, элегантная игровая механика, даже без нарративной привязки.

Но оставить «Кинезис» в качестве чисто игровой механики означало бы игнорировать роль технологии в нарративном мире. Если бы это устройство действительно существовало в более широком мире *Dead Space 2*, что бы это значило? Как бы оно было связано с культурой, экономикой и зданиями? Вместо того чтобы игнорировать этот вопрос, дизайнеры Visceral Games ввели многие ответы в нарратив о мире. Айзек сначала получает «Кинезис», вырывая его из устройства, используемого для удерживания пациентов во время операции. Он встречает рекламные объявления о продукте, в котором используется технология «Кинезиса» для удержания людей во время сна. Он взаимодействует с инженерными системами, которые содержат маркировку и предупреждения о работе «Кинезиса», откуда следует, что «Кинезис» является распространенным инструментом среди людей, работающих с тяжелой техникой. Элегантность «Кинезиса» заключается не только в наличии многих способов использования в боях, разведке и решении головоломок, но и в количестве связей, которые он создает в нарративном мире.

Эмергентная история

Во время любого игрового сеанса игровая механика, игроки и случайность объединяются, чтобы создать оригинальную последовательность связанных событий, которые составляют *эмергентную историю*.

Если вы играете в гонки против друга и возвращаетесь после тяжелой аварии, чтобы выиграть, — это история. Но она не написана геймдизайнером. Она возникла во время вашей конкретной игровой сессии. Это называется эмергентной историей.

Мы можем рассматривать эмергентную историю с двух сторон: как нарративный инструмент и как технологию для генерации сюжетного содержимого истории.

ЭМЕРГЕНТНАЯ ИСТОРИЯ — это история, которая генерируется во время игры в результате взаимодействия игровой механики и игроков.

Эмергентная история — это нарративный инструмент, потому что дизайнеры косвенным образом создают эмергентные истории игры, когда проектируют игровую механику. Например, игроки *Assassin's Creed: Brotherhood* получают миллионы уникальных эмергентных историй о средневековых битвах, дерзких убийствах и укрытиях на крышах. Но никто из игроков никогда не сталкивался с историей о том, как средневековый убийца чистил зубы по утрам. Чистка зубов не является игровой механикой в *Assassin's Creed: Brotherhood*, поэтому подобные истории в этой игре возникнуть не могут. Задав определенную механику игры, ее дизайнеры определили, какие истории она способна генерировать. Таким образом, они косвенно создали эмергентные истории, которые генерирует игра, даже при том, что дизайнеры не написали отдельные события.

Мы также можем рассматривать эмергентный нарратив как технологию создания историй, потому что он создает оригинальный контент. Дизайнер создает границы и тенденции игровой механики, но именно взаимодействие механики, выбора игрока и случайности определяет реальный сюжет каждой эмергентной истории. Это может быть очень элегантным способом создания историй, потому что ответственность за авторскую работу ложится на игровые системы и игроков. И это позволяет бесконечно генерировать истории.

Первое акцентирует внимание на том контроле, который есть у дизайнера, а второе — на контроле, которого у дизайнера нет. Это две ипостаси одного и того же.

Идея эмергентной истории имеет ценность только как способ заставить нас думать по-другому. Когда мы рассматриваем опыт,

сгенерированный механикой, мы спрашиваем: проходим ли он? Понятен ли интерфейс? Присутствует ли глубина? Но когда мы рассматриваем тот же опыт с точки зрения эмергентной истории, мы уже спрашиваем: интересен ли образ персонажа? Является ли кульминация непредсказуемой, но неизбежной? Является ли изложение ровным и незаметным? Используется ли классическая трехактная структура или какая-то другая форма? Тщательное продумывание эмергентной истории позволяет нам задей-

Посмотрите внимательно, концепция эмергентной истории — это просто еще один способ описания сгенерированного опыта.

ствовать огромный набор инструментов для придумывания историй, который мы могли бы упустить, анализируя динамически генерируемые игровые ситуации. Но в обоих случаях мы анализируем одно и то же: ряд событий, генерируемых во время игры. Называть их историей — это просто способ описать их, связав с традиционными историями, написанными непосредственно авторами.

Существуют вещи, которые можно получить только благодаря эмергентным историям, потому что только они могут преодолеть барьер между вымыслом и реальностью.

Игрок, который в совершенстве овладевает новым приемом в шахматах и использует его, чтобы победить своего старшего брата, впервые переживает историю, но эта история происходит за пределами игры или сюжета. Когда он рассказывает эту историю своим друзьям, то может упомянуть о хороших ходах, которые привели его к победе, но главная нить этой истории — его развивающиеся взаимоотношения с братом.

Эта история — не выдуманная писателем часть сюжета, но и не сюжет, созданный машиной. Это правдивая история из жизни игрока. Создание подобных реальных историй может произойти только внезапно. Геймдизайнер не может создавать жизнь игроков вместо них самих.

Апофения

Человеческий разум — это ненасытная машина распознавания паттернов. Мы видим паттерны повсюду, даже если их нет. Дети смотрят на облака и говорят, что они похожи на собаку, лодку, человека. Всмотритесь в белый шум телевизора, и вы увидите фигуры или буквы, вращающиеся вокруг экрана. Древние римляне предсказывали будущее, пытаясь найти закономерности в недрах жертвоприношения животных (и всегда находили их). Даже сегодня астрология, нумерология и сотни других видов мошенничества существуют благодаря апофении.

Апофения работает с любым узнаваемым паттерном, но разум особенно жаждет определенных типов паттернов.

АПОФЕНИЯ — это склонность человека видеть несуществующие паттерны или взаимосвязи в сложных данных.

Одним из них является личность. Человеческий разум постоянно работает над тем, чтобы понять намерения и чувства других. Этот порыв настолько силен, что действует даже на неодушевленных предметах. Вот почему мы без проблем понимаем мультфильм, в котором безликая настольная лампа боится резинового мячика. И это то, что заставляет нас говорить: «Атом кислорода хочет быть рядом с другим атомом кислорода, а не с двумя». Это высказывание нельзя понимать буквально, потому что атомы кислорода не имеют разума и не могут ничего хотеть. Но мы не обращаем на это внимания, потому что легко думаем о вещах как об агентах, действующих в соответствии с желанием и намерением.

Этот вид апофении — то, что позволяет добавить персонажей и чувства в эмергентные истории. У нас нет компьютерной технологии, чтобы по-настоящему смоделировать человеческий разум в видеоигре. Но апофения означает, что компьютеру не нужно имитировать реалистичный разум. Достаточно всего лишь сделать так, чтобы разум игрока интерпретировал что-то в игре как агента, способного самостоятельно мыслить, так же как мы можем воспринимать то, что настольная лампа из мультфильма может быть любопытной или чего-то бояться.

Как только это достигнуто, игрок начинает действовать неосознанно, наделяя предмет воображаемыми желаниями, обязательствами, восприятием и человеческими отношениями. В самой игре фишки просто продолжают перемещаться. Но в сознании игрока движущиеся фишки выдают более глубокий подтекст интриги и желания. Король боится этой пешки. Конь в бешенстве. Ладья скучает. Несмотря на то что эти чувства в действительности не существуют в игре, они существуют в сознании игрока, и именно это важно для опыта.

Апофения играет ключевую роль в эмергентной истории. Учитывая это, давайте рассмотрим некоторые конкретные способы создания игровых систем, которые генерируют эмергентные истории.

Навешивание ярлыков

Дизайнеры могут усилить эмергентность, добавив в существующую игровую механику сюжет.

Close Combat: A Bridge Too Far. Этот тактический симулятор освещает сражения во Второй мировой войне. Игра дает имена и отслеживает судьбу каждого отдельного солдата на поле боя. Это означает, что игрок может просмотреть карточку солдата и узнать, что за последние несколько сражений все, кроме одного из его товарищей по службе, погибли. Игрок может представить то, что связывает этих двоих солдат после смерти их общих боевых товарищей. И в следующей битве он может расстроиться, когда приказывает одному из них пожертвовать собой, чтобы другой мог выжить.

Medieval: Total War. У каждого дворянина, принцессы и генерала в этой великой стратегической игре есть имя и уникальные характерные черты. Но вместо отслеживания числовой статистики, вроде интеллекта или силы, игра присваивает личностные характеристики знати и генералам. После таких событий, как женитьба или победа в битве, дворяне могут получить прозвище типа «Пьяница», «Бесстрашный» или «Трус», которые дают особые преимущества и недостатки. В другой игре игрок может проиграть битву, потому что у его генерала низкий показатель лидерства. В игре Medieval он проигрывает, потому что у его генерала есть дочь и он решил, что слишком сильно любит свою семью, чтобы умереть в бою.

Навешивание ярлыков работает на апофению. В каждом примере эмергентная история в сознании игрока не происходила в игровых системах. Игра Close Combat не симулирует сближение солдат на фоне общей потери. А игра Medieval на самом деле не отслеживает человеческое мужество или семейную привязанность. Но человеческий разум все равно видит истории при наличии малейших предположений. Ярлык здесь, имя там, и история тут же расцветает в воображении. Это очень элегантный метод, потому что разум игрока выполняет почти всю работу.

Абстракция

Слова в романе могут создавать в воображении более мощные образы, чем любая фотография, потому что они только предлагают образ, оставляя разуму возможность дорисовать детали.

Фотография требует меньше воображения, чем роман, а также оставляет меньше места для полета фантазии.

Более детальная графика и более качественный звук добавляют что-то в игру, но в то же время и что-то отнимают. Чем детальнее графика, звук и диалог игры, тем меньше места для интерпретации. Чем более абстрактное, размытое и минималистичное представление, тем больше вероятность апофении. Поэтому иногда стоит сознательно предоставлять меньше информации, чтобы у игрока оставалось больше возможности для свободной интерпретации.

Самый яркий пример — игра Dwarf Fortress. В этой игре нет графики. Дварфы, гоблины, трава, камни и сотни других объектов представлены символами ASCII. Когда большинство людей смотрят на символы «☺☺~~~~», они видят нечто неопределенное. Игрок Dwarf Fortress распознает мужа и жену расы дварфов, сидящих на траве у реки, наслаждающихся жизнью.

Чем меньше мы покажем и расскажем игроку, тем больше возможностей останется для апофении, которая поможет заполнить пробелы.

Но вовсе не обязательно заходить так далеко, чтобы апофения заработала. Любой пробел в представлении создает пространство в сознании игрока, которое он захочет заполнить. Например, в игре Medieval не появился бы любящий генерал, если бы дизайнеры показали его взаимоотношения с женой. Если бы отношения к семье были доступны игрокам, это стерло бы любую интерпретируемую личность из их сознания.

Точно так же двое последних выживших из целого отряда солдат в Close Combat не смогли бы сблизиться на фоне общей утраты, если бы игрок мог приблизить их и наблюдать за неактивной анимацией и боем, когда солдаты не обращают внимания друг на друга, пока враг на них наступает. Явления, доступные нашему зрению, блокируют картинку в воображении.

Самым простым примером апофении являются игрушечные кубики Rory's Story Cubes. Кубики историй — это девять кубиков с изображениями овец, молний и других случайных рисунков. Игроки бросают их, смотрят на картинки и составляют связную историю. Сначала кажется абсурдным пытаться связать воедино картинки черепахи, облачка с текстом и дерева. Но на самом деле

это довольно легко, особенно для творческих людей со слабыми ассоциативными барьерами (например, дети, которые являются основной целевой аудиторией рынка игрушек).

Потребность в абстракции — это причина, по которой истории, придуманные игроками, чаще всего возникают в стратегических и строительных играх, экономических симуляторах и настольных ролевых играх, таких как *Dungeons & Dragons*. Эти жанры обычно представляют игровые элементы на расстоянии, со статистикой и символами. Такие жанры, как шутеры от первого лица или спортивные игры, редко поддерживают такой высокий уровень эмергентности, так как они обычно показывают игроку слишком много для того, чтобы у того сработала апофения.

Ведение записей

Игры могут усилить эмергентные свойства возникающих историй, ведя записи игровых событий, чтобы игроки могли помнить о том, что с ними произошло. Следовательно, игроку не обязательно запоминать все, что случилось, чтобы создать в уме историю — все это присутствует у него перед глазами.

Civilization IV. Эта игра тайно записывает границы каждой нации в конце каждого хода. Когда игра наконец заканчивается, игрок начинает смотреть изменение карты мира во времени, на которой показаны изменения политических границ от начала и до конца игры. Интересно наблюдать за тем, как империи начинают свое существование в виде маленьких точек, растут, охватывая целые регионы, и подвергаются нападениям со всех сторон в течение столетий войны. Поскольку карта пересказывает мировую историю, она напоминает игроку о сложных задачах, с которыми он столкнулся, и победах, которые он одержал.

Myth. Эта классическая тактическая игра отслеживает сражения между небольшими группировками от 10 до 100 фэнтезийных воинов.

В большинстве игр трупы исчезают вскоре после смерти. Но *Myth* оставляет все мертвые тела, отрубленные конечности, стрелы, брошенный заряд взрывчатки, след от бомбы и брызги крови на земле в том месте, где это все упало. После битвы игрок может

найти оставшуюся кровь и тела и восстановить все события. Ряды трупов в углу, окруженные еще большим количеством мертвых зомби, указывают на отчаянную последнюю битву. Следы бомбы, вокруг которой растекается звезда крови, — это кто-то, разорванный на куски. Так называют эмергентную мизансцену. И хотя, вероятно, все это уже однажды произошло на глазах у игрока, все равно ему будет интересно видеть оставшуюся на земле кровь и воронки от бомбы.

The Sims. Игроки могут фотографировать созданную ими в игре семью и помещать фотографии с подписями в альбомы. Обычно альбомы рассказывают истории. Лучшие из них объединяются в блоги, такие как Алиса и Кев, — история о психически неуравновешенном отце-одиночке и его великодушной дочери-подростке. Об их бедности, неприятии друг друга и разрыве отношений. Больно смотреть, как Алиса отдает свой последний доллар на благотворительность, хотя сама спит на скамейке в парке.

Система «спортивный обозреватель»

Спорт может сбивать с толку, особенно непосвященных. Покажите 20 человек на поле, бегущих и сталкивающихся друг с другом, и вы не поймете, что тут происходит. Обозреватель добавляет контекст к хаосу игры и объединяет события в последовательный нарратив. Хороший спортивный обозреватель может закружить слушателя и увести его в сказку с характером, напряжением, кульминацией и развязкой, которая, по сути, создана из хаоса сталкивающихся тел.

Игры могут применять тот же принцип, создавая игровые системы, которые пытаются интерпретировать и связывать воедино игровые события, как более продвинутый вид ведения записей. Наиболее очевидный пример — обозреватели спортивных игр, но существует и много других способов реализовать эту идею.

Например, после каждого уровня в *Hitman: Blood Money* игра отображает газетную статью, в которой освещаются убийства и последующее полицейское расследование. История меняется в зависимости от метода, использованного для поражения цели, точности игрока и количества выпущенных выстрелов, выстрелов в голову, убитых свидетелей, оставшихся в живых

свидетелей и многих других факторов. Заголовки варьируются от «Разыскивается Бесшумный убийца» до «Отморозок зарезал 17 человек!» Если остались живые свидетели, история включает портрет-набросок персонажа игрока, сделанный полицейским. Чем больше свидетелей, тем точнее портрет.

Механику спортивного обозревателя делать на хорошем уровне не просто, так как играм очень сложно систематически интерпретировать события, важные для человека. И так же, как детализированная графика подавляет воображение, сложные объяснения обозревателя могут непосредственно вытеснить игру самих игроков. Поэтому механики спортивных обозревателей работают лучше всего в том случае, если они не пытаются рассказать всю историю, а вместо этого просто запускают собственный апофенический процесс.

Упорядочивание истории

Игра в произвольной форме позволит игроку выбрать любой нарратив. Представьте, что вы вырвали все страницы романа и разбросали их по полу. Можно подобрать и прочитать любую страницу в случайном порядке. В таком нарративе нет упорядоченности, поскольку читатель может воспринимать содержание в любой последовательности.

В какой-то степени история может работать так же, как и в предыдущем примере о нарративе о мире. Но большинство нарративных инструментов по-прежнему работают лучше, если мы контролируем порядок их применения. Как правило, мы хотим, чтобы завязка была перед развязкой. Возможно, мы захотим закончить один подсюжет, прежде чем добавим другой, чтобы избежать избыточного количества параллельных нитей повествования. Или, может быть, мы хотим ввести игровую механику постепенно, вместе с историей, чтобы игрок мог обучаться последовательно.

В играх используются различные способы упорядочивания истории.

Уровни — это классический способ упорядочивания историй. Игроки проходят первый уровень до конца, затем второй, затем третий и так далее. Это старый, простой и рабочий метод.

Миссии — еще один классический способ упорядочивания истории. Миссия — это отдельная мини-история внутри большого неупорядоченного мира. Мир может простирается на весь континент, а миссия может всецело захватить игрока, помогая одному лавочнику избавиться от вымогателя. Миссия начинается, когда игрок встречает лавочника и слышит о его сложном положении. Затем игрок находит бандита, убеждает его остановиться или избивает его и наконец возвращается к продавцу за оплатой. В этой последовательности порядок событий неизменен. Но эта мини-история может начаться и закончиться в любое время, когда игрок исследует город. Ее также можно прервать: игрок может встретиться с владельцем магазина, избить бандита, затем отвлечься и пойти убить дракона в другой части мира, прежде чем наконец вернуться к владельцу магазина, чтобы получить свою награду.

Третьим основным способом упорядочивания истории является *блокировка*. Самая простая блокировка — запертая дверь. Игрок сталкивается с дверью, и прежде чем он сможет пройти дальше, он должен пойти и найти ключ. Поэтому, что бы ни случилось, пока он получает ключ, это гарантированно произойдет раньше, чем что-либо произойдет за дверью. Блокировки не обязательно должны представлять собой закрытые двери в буквальном смысле; возможно, это будет охранник, который не пропустит вас до тех пор, пока вы не окажете ему услугу, или камера слежения не пропустит вас, если вы сначала не выключите свет.

Существуют и более мягкие способы упорядочивания истории. Они обеспечивают определенный порядок истории, не гарантируя его полностью.

Получение новых способностей — это мягкий способ упорядочивания истории. Благодаря ему игроки могут получить доступ ко всему игровому контенту, начиная с первого момента игры. Тем не менее чтобы получить доступ к части контента, игроку требуется проявить определенные навыки. Например, чтобы поговорить с персонажем, игрок должен сначала победить его в бою. Игроки в конечном итоге проходят игру в заданном порядке по мере роста их мастерства, хотя весь контент технически доступен с самого начала.

Версия «получение способностей» используется во многих многопользовательских ролевых играх. Технически игрок может отправиться куда угодно с самого начала, если не считать того, что у него нет навыков, «прокачанных» персонажей или союзников, необходимых для выживания далеко за пределами его зоны старта. Таким образом, игра создает ощущение огромного открытого мира, но при этом мягко направляет новых игроков через тщательно продуманную последовательность начальных испытаний.

Существует множество других способов упорядочивания истории. В играх с контролем времени, таких как *Dead Rising*, события в мире происходят в фиксированное время. Доступ к миссии может открыться, когда персонаж игрока достигнет определенного уровня прогресса. Даже простая организация пространства может создать мягкое упорядочивание истории, так как игроки могут столкнуться с фигурами, расположенными поблизости раньше, чем с более отдаленными.

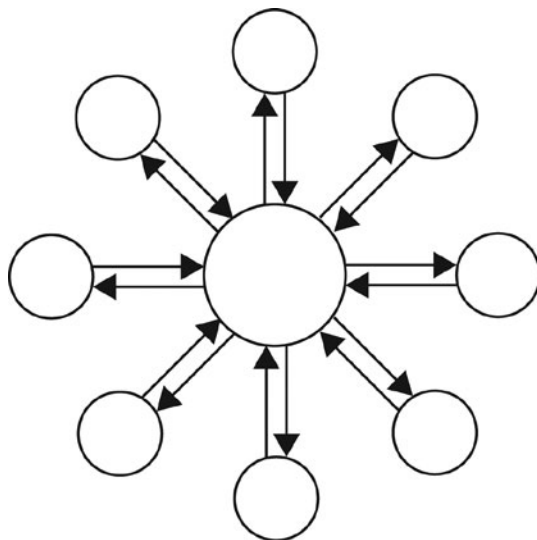
Структуры истории

Существует множество возможностей формировать и объединять способы упорядочивания истории. Основной является структура «нитка жемчуга». В этой модели каждая жемчужина — это арена, по которой игрок может свободно передвигаться и взаимодействовать с игровой механикой, а каждая нить — это односторонний переход на следующую арену.



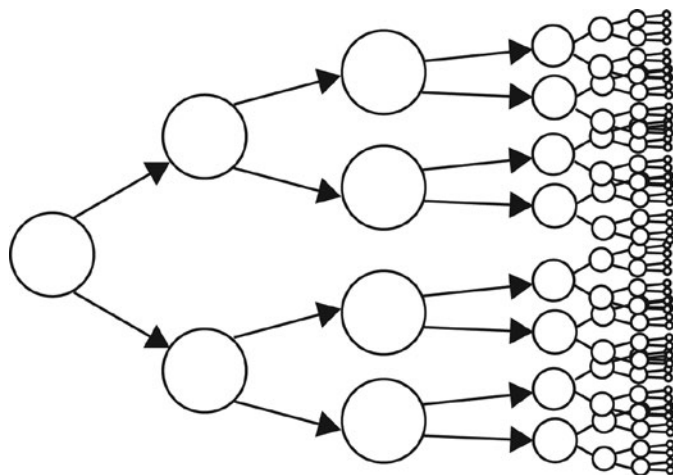
Это классическая линейная игра, которая движется от одного уровня к другому. В зависимости от того, насколько «крупными» являются жемчужины, насколько большую внутреннюю свободу они дают, игрок может чувствовать либо полную бессмысленность, либо абсолютную свободу. Именно структура «нитка жемчуга» лежит в основе таких однопользовательских игр, как *Quake*, *Super Mario Bros* и *StarCraft Mario*.

Еще одна модель представлена в виде «разветвленной структуры»:



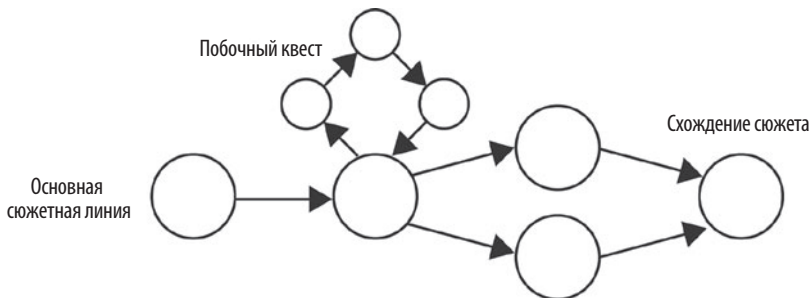
Каждый кружок — это самостоятельная часть контента, не связанная с другими. Игры Mega Man строятся на основе разветвленной структуры.

Иногда геймдизайнеры пытаются имитировать реальный выбор, моделируя результаты каждого возможного решения. Но простоватая форма этой структуры имеет фатальный недостаток.



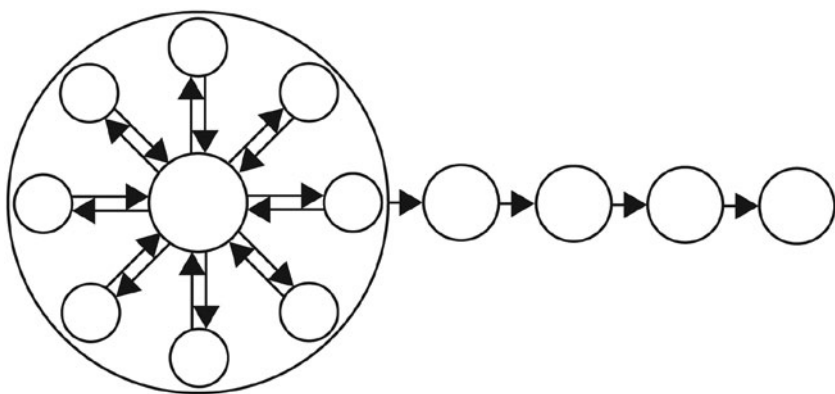
Проблема разветвления событий заключается в том, что количество возможных историй развития резко увеличивается. Любой игрок, который проходит через историю, пропускает большую часть контента. Единственная ситуация, в которой эта модель оправдана, когда почти весь контент генерируется сразу. Если события предопределены в какой-либо значительной степени, мы должны предпринять меры, чтобы уменьшить количество ветвей.

Мы можем сохранить некоторые возможности выбора отклонения сюжета, ограничив количество вариантов развития с помощью таких способов, как побочные миссии и конвергенция сюжета. Побочные миссии предлагают часть контента на разветвлении пути, по которому можно либо пойти, либо нет, но этот выбор мало влияет на основной путь. Конвергенция сюжета предлагает варианты, которые разветвляют основную сюжетную линию, но позже сходятся в одну.

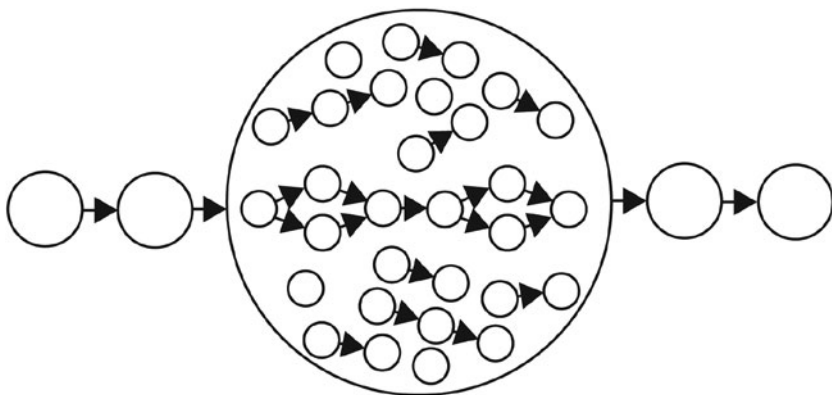


Для некоторых игр такой простой структуры достаточно. Тем не менее нам часто нужно тщательнее комбинировать способ упорядочивания историй, чтобы соответствовать потребностям игры.

Mega Man 2. Игра начинается как разветвленная структура, так как игрок может победить восемь роботов в любом порядке. После того как все они побеждены, игра переключается на линейную последовательность, когда Мега Мэн нападает на замок доктора Вайли и приближается к завершению игры.



Mass Effect 2. Начало и конец игры — это линейные «нити жемчуга», а середина (80%) — это огромное количество миссий, ненавязчиво упорядоченных по уровню навыков игрока и персонажа, при этом основная миссия находится в центре и включает в себя ответвления и конвергенции сюжета. Эта гибридная структура весьма популярна, поскольку в ней много преимуществ. Дизайнеры пишут тщательно продуманное введение, которое является введением к сюжету и игровой механике. Ненавязчивое упорядочение центральной миссии дает игроку чувство свободы и отсутствие ограничения. Наконец, дизайнеры могут тщательно разработать кульминацию игры для достижения максимального эффекта.



Проблемы агентивности

Представьте, будто вы драматург экспериментального театра. Вы прописываете сценарий для каждого героя, кроме одного. Главного героя играет случайный человек из зала, которого пригласили на сцену и дали роль без сценария или возможности выучить текст.

Жестко, да?

Теперь представьте, что этот человек из зала пьян. И он отвлекается, потому что переписывается с кем-то по телефону. Он решил поразвлечься, сознательно вмешиваясь в сюжет, — оскорбляет других актеров, ворует предметы со сцены и совсем не задействован в кульминации сюжета.

Это кошмар драматурга. Дурак на сцене ломает все тщательно продуманные диалоги, будет противоречить другим героям и нарушит всю историю. Геймдизайнеры сталкиваются с этим каждый день, потому что игры дают игрокам агентивность.

Тщательно продуманный традиционный сюжет — это карточный домик. Каждая деталь персонажа, каждое слово в диалоге, каждый оттенок знаний, которые герой передает или утаивает, играет роль в запутанном танце нарратива. События истории должны складываться как домино и приводить к удовлетворительному заключению, которые обращаются к бессмертным темам. Писатель тщательно подбирает каждое слово для достижения этого результата.

АГЕНТИВНОСТЬ — это способность принимать решения и предпринимать хорошо просчитанные шаги, влияющие на игровой мир.

Сюжет компьютерной игры преследует ту же цель. Но, как и неудачливый драматург, геймдизайнер также должен учитывать тот факт, что игроки могут делать выбор. И независимо от того, делают ли они его по незнанию или по злему умыслу, игроки могут легко опровергнуть или пропустить фрагменты сюжета, разрушив карточный домик автора.

Эти проблемы агентивности делятся на несколько категорий. Давайте рассмотрим их по порядку.

Совместимость мотивации игрока и персонажа

Многие проблемы агентивности появляются потому, что мотивы игрока отличаются от мотивов персонажа, которым он управляет.

Персонаж хочет спасти принцессу, заработать деньги или пережить восстание зомби. Его мотивы сосредоточены внутри сюжета фантастических замков, криминальных деловых отношений или вторжений живых мертвецов. Игрок хочет расслабиться, увидеть весь игровой контент и прокачать способности. Его мотивация сосредоточена в реальном мире социального статуса, развлекательных долларов и игровой механики. Когда эти две мотивации тянут в разные стороны, игрок предпримет действия, которые нарушают нарратив. Я называю это *деск-джампингом*.

Деск-джампинг — когда игрок делает то, чего никогда не сделал бы персонаж игрока, потому что у него другие мотивы.

Название появилось на основе ситуации, которую я обнаружил в игре RPG Deus Ex. В Deus Ex игрок — супершпион, работающий на секретную межправительственную организацию. Он может исследовать секретный офис своего агентства, получать задания, разговаривать с коллегами и прыгать у них на столах. Представьте, что Джеймс Бонд танцует на столе своего босса, когда они обсуждают рискованную миссию. Это глупо и бессмысленно. Но игрок будет так делать, потому что это смешно. Мотивация персонажа — выполнить миссию, а мотивация игрока — чтобы было смешно. Мотивы не совпадают, поэтому игрок прыгает на стол босса и изначальный замысел рушится.

Игроки выполняют этот трюк по многим причинам. Они хотят исследовать границы симуляции, попробовать контент, получить вещи, достигнуть трудных целей, удивить друзей и посмотреть спецэффекты. Я видел, как игроки нападают на союзников, систематически грабят невинных (играя за хорошего персонажа), пытаются убить каждого охранника во дворце, просто чтобы посмотреть, смогут ли они это сделать, или складывают нефтяные бочки на городской площади в одну кучу и поджигают их, пытаясь устроить большой взрыв.

Учитывайте мотивацию игрока, желающего экспериментировать с игровыми системами. Игра в сверхъестественном

криминальном шутере The Darkness начинается с того, что игрок видит перед собой раненого союзника, который ведет длинный диалог. Когда я играл в эту игру, я не слушал, что там говорит союзник. Я его застрелил, просто из любопытства. Не то чтобы я его ненавидел и хотел убить — просто моя мотивация была за пределами сюжета. Скорее я изучал пределы игровой механики. Я искал ответ на вопрос: «Как дизайнеры справились с этим?» В опыте на основе механики это нормально, так как исследование систем является основным смыслом. Но у этих механик была фантастическая обертка. И в этой художественной обертке стрелять в союзника было бессмысленно. Он умер, и я пропустил большую часть его диалога, и образ хорошего героя распался.

Иногда деск-джампинг может быть почти произвольным. В Grand Theft Auto IV главный герой, Нико Беллик, пытается сбежать от своего жестокого прошлого, когда он воевал в боснийской войне. Игра тратит не один час на подготовку решающего нарративного решения, при котором Нико либо убивает старого врага из ненависти, либо отпускает его. В зависимости от решения суть характера Нико и мораль повествования висят на волоске. Нико сможет сдержать себя и стать мирным человеком или вернется на темную сторону? Что победит — зло и ненависть в мире или сломленный человек сможет исцелиться и стать добрым? Это злободневный вопрос.

Если не считать того, что к этому времени в игре Нико убил сотни невинных людей. Дизайн игровой механики Grand Theft Auto IV стимулирует игрока убивать десятки полицейских и наезжать на толпы пешеходов просто так. Вероятно, Нико за несколько минут до этого раздавил несколько старушек, собираясь встретиться со своим заклятым врагом. А теперь он хмыкает и невнятно бормочет о том, стоит ли убивать одного человека. Мотивация игрока состоит в том, чтобы убивать множество людей ради забавы, в то время как мотивация персонажа не в том, чтобы убивать. В результате получается бессмыслица.

Существует много способов решить проблемы деск-джампинга. Давайте их рассмотрим.

В игре Deus Ex дизайнеры могли бы отключить возможность деск-джампинга или установить невидимые блокираторы на столах, чтобы на них нельзя было запрыгнуть. Проблема в том, что игроки быстро почувствуют наличие искусственных устройств, которые использовались бы для этих целей. Игра больше не

Запрещение деск-джампинга работает, но ослабляет вовлеченность игроков, разрушая их веру в честность игровой механики.

верна своим системам — она обманывает в своем собственном наборе правил, чтобы получить «деспотичский» результат, которого хотел дизайнер. Столкнувшись с этим, игроки перестают думать о том, что позволяет механика, и начинают думать о том, каких действий от них ждет дизайнер. На экране нарратив остается неизменным, но ход мысли игрока при изучении игровой механики меняется, потому что игровая механика нечестна и предсказуема.

Если это оправдано сюжетом, запрет на деск-джампинг хорошо работает. Например, игру Portal разработанную американской компанией Valve, хвалят за сторителлинг, но на самом деле она не решает ни одной из острых проблем сторителлинга в играх. Скорее она полностью избегает их благодаря умному построению сюжета. Единственный неигровой персонаж в Portal — это GLaDOS, компьютерный ИИ, который общается с игроком исключительно через систему связи; в игре нет других человеческих персонажей. Персонаж игрока заперт в пустых камерах с белыми стенами в подземном научном центре. Единственный инструмент, который она находит, — это порталная пушка.

Мир Portal настолько мал и ограничен, что в нем естественным образом запрещены любые действия игрока, которые могут нарушить сюжет. Герой не может рассказывать другим персонажам странные вещи или запрыгнуть им на голову, потому что единственный другой персонаж — это бесплотный компьютерный голос. Он не может взорвать стену, потому что у него нет взрывчатки. Он может отказаться выполнять задачу, но ИИ это не беспокоит, потому что ИИ может ждать вечно. У игрока нет никакого соблазна плясать у кого-то на голове, потому что в этой истории нет голов.

Подобные трюки использовались во многих других играх. Действие игры BioShock происходит в рушащемся подводном го-

роде — идеально закрытой изолированной среде, похожей на испытательные камеры Portal. Вы не можете выйти за пределы уровня, потому что большая часть города заблокирована и затоплена. Вы не можете взрывать стены, потому что они сделаны из арматурной стали, чтобы противостоять разрушающему давлению океана. Вы не можете разговаривать с местными жителями, потому что все они чудовищно опасны. Сюжет естественным образом запрещает большинство из того, с чем игровые системы не смогут справиться. В играх, действие которых происходит в реалистичных городах, исследование мира должно быть ограничено запертыми дверьми и другими блокировками, а общение с незнакомцами должно быть частично запрещено интерфейсом.

Компания Valve применила это решение в Half-Life 2. Когда вы застрелили союзника игрока, ничего не происходит. Он не непобедим; пули просто никогда не убивают его. Нет ни крови, ни анимации, ничего.

Часто игнорирование там, где это возможно, лучше, чем запрещение или наказание, потому что игрок чувствует, что ему дается больше свободы, и быстро прекращает нежелательные действия, если они не дают интересного результата. Игроки понимают, что у игровой механики присутствуют ограничения; зачастую лучше пусть эти ограничения будут простыми, очевидными и скучными, чем замаскированными.

Игроки занимаются деск-джампингом, чтобы посмеяться, изучить механику и повышение мощности. Это нормальные мотивы. Иногда лучше взять за основу действия игроков и развернуть сюжет вокруг них.

Например, в Deus Ex, исследуя офис шпиона, игрок может зайти в женский туалет. Если он это делает, то сталкивается с потрясенной сотрудницей, а позже его отчитывает босс. Это забавный ответ на забавное действие игрока.

Некоторые игры определенно выигрывают от использования деск-джампинга. В Duke Nukem Forever традиционные очки здоровья игрока заменяются на очки эго, которые растут, если Дюк играет в пинбол, поднимает гири, разбрасывает баскетболь-

Мы можем полностью игнорировать деск-джампинг, давать игрокам такую возможность и не ограничивать их. Это делает деск-джампинг менее интересным.

Иногда деск-джампинг можно сделать частью нарратива.

ные мячи и домогается стриптизерши. Это раздувает и без того чрезмерную самооценку Дюка.

Ключевая проблема деск-джампинга заключается в том, что это может привести к постоянному росту тех возможностей, которые должны быть включены. Если игрок запрыгивает на стол босса, а босс говорит: «Убирайся с моего стола», мы включили деск-джампинг. Но что, если игрок продолжает прыгать на столе? Будет ли босс еще раз просить его слезть со стола? Он, в конце концов, предпримет физическое насилие против игрока? А что потом? Вызовут ли игрока в конечном итоге в суд и посадят в тюрьму из-за перестрелки в офисе, которая началась с разногласий по поводу его прыжков на столе у босса? Игрок, который хочет изучать игровые системы или создавать смешные ситуации, всегда сможет продолжать развиваться в этом направлении. Чтобы решить эту проблему, лучше всего искать способы включить деск-джампинг закрытым способом, как в случае с *Duke Nukem Forever*.

Мы можем решать проблему деск-джампинга вечно. Но лучший способ закрыть этот вопрос раз и навсегда — сделать так, чтобы у игроков это желание никогда не было на первом месте.

Лучшее решение для деск-джампинга состоит в том, чтобы разработать игру так, чтобы мотивы и способности игроков совпадали с мотивами их персонажа.

Например, в *Call of Duty 4: Modern Warfare* деск-джампинг возможен. Игрок может отказаться от выполнения задач, отказаться стрелять или попытаться заблокировать союзников или поймать их. Тем не менее в этой игре это происходит редко, поскольку сражение происходит стремительно, упорно и так ожесточенно, что враги практически наступают на горло. Когда танки взрываются, командиры толкают войска вперед, а враги роятся как мухи, игрок становится настолько взвинченным, что побуждение драться перекрывает побуждение действовать глупо.

Мотивация игрока не обязательно должна совпадать с исходной мотивацией персонажа — только с целью. В *Call of Duty* мотивацией персонажа являются честь, лояльность и страх, а мотивация игрока — энергия и развлечения. Неважно, что эти мотивы очень разные, они ведут к одним и тем же действиям: бесстрашно сражаться с врагами.

Достичь такой слаженной мотивации очень трудно, потому что она пересекает границы между сюжетом и нарративом. Мы должны не просто зажечь в игроке горячее желание достичь какой-то цели, это желание еще и должно быть отражено в герое. Это одна из основных причин, по которой мы должны создавать сюжет и механику как единое целое, а не разрабатывать их по отдельности, а затем объединять.

Проблема взаимодействия между людьми

Традиционные истории строятся из взаимодействия персонажей. Персонажи предают, требуют, предлагают, заявляют, обсуждают и ведут диалог через серию эмоциональных поворотов, составляющих историю. Это касается практически всех историй, а не только драм. Даже в самых ярких боевиках и кровавых ужасах большая часть времени уходит на разговоры людей.

Для геймдизайнеров это проблема, поскольку в настоящее время не существует способа обеспечить богатое взаимодействие человека с компьютером. Кнопок, джойстиков и простых датчиков движения недостаточно, чтобы дать людям возможность выражать мысли и чувства к машине. Более того, даже если игроки смогут проявить эти чувства, машина не сможет ответить тем же, потому что у нас нет технологии, которая могла бы имитировать человеческий разум.

Чтобы сделать подобное взаимодействие в играх возможным, мы можем прибегнуть к ряду приемов, которые помогут обойти эти ограничения.

Самое очевидное решение проблемы взаимодействия с человеком — не давать возможности взаимодействовать с человеком.

Инструменты мира и эмергентной истории работают в играх так хорошо потому, что они не требуют, чтобы игра контролировала игроков, взаимодействующих с персонажем.

Это не значит, что в игре не может быть персонажей или что люди не могут говорить. Вы можете взаимодействовать с глупыми или безумными персонажами. Вы можете взаимодействовать с квазичеловеческими

Мы можем настроить сюжет так, чтобы у игрока не было никакого способа напрямую взаимодействовать с человеческими персонажами.

компьютерами или ИИ. Вы можете наблюдать, как другие люди взаимодействуют друг с другом, или найти запись более раннего разговора. Единственное ограничение заключается в том, что персонаж игрока никогда не может вступать во взаимодействие со здравомыслящим сознательным логичным персонажем, похожим на человека.

Например, в BioShock здравомыслящие персонажи разговаривают с игроком только по радио или через непробиваемое стекло. Все персонажи, с которыми можно столкнуться лицом к лицу, неадекватны. Вы можете наблюдать за этими сумасшедшими, пока они живут своей разбитой жизнью, вы можете слушать их неменяемое бормотание, но это работает, потому что вы не взаимодействуете, а просто наблюдаете, как они следуют predetermined сценарию. Как только вы попытаетесь взаимодействовать, они зверски свирепеют, компьютер может за просто смоделировать такое поведение.

Некоторые игры моделируют межличностное взаимодействие с диалоговым деревом, благодаря чему игрок может выбирать

ДИАЛОГОВОЕ ДЕРЕВО может решить вопрос с человеческим взаимодействием, предварительно определяя список действий, которые могут выполнить игроки, и сопоставив ответы других персонажей.

из множества социальных взаимодействий те, которые может выполнять его персонаж. Это работает, потому что геймдизайнеры могут создавать каждую сторону каждого взаимодействия. Не нужно ничего симулировать.

Недостатком является то, что игрок ограничен в количестве вариантов выбора, вместо того чтобы получить почти бесконечное разнообразие, доступное в реальной жизни.

Действия, которые игроки могут выполнять в играх, обычно сводятся к перемещению, поискам, толчкам, прыжкам и стрельбе. Можно использовать эти виды взаимодействий для выражения общения между людьми.

Мы можем повторно использовать стандартные игровые действия как опции в диалоговом дереве.

Одним из примеров является ситуация, в которой игрок вынужден выбирать между убийством двух разных персонажей. В Grand Theft Auto IV главного героя сталкивают со старым врагом, который лежит связанным на земле, и дают возможность убить его. Игрок может либо застрелить беззащитного человека, либо уйти. Оба действия выражаются через элементы управления, которое исполь-

зуется на протяжении всей игры: стрельба и ходьба. Но в данном случае эти действия используются для управления заранее определенной сюжетной веткой, а не обычным геймплеем.

Такое взаимодействие практически аналогично диалоговому дереву, поскольку опции, доступные игроку, и ответы игрового мира заранее определены. Разница лишь в том, что они выражают свой выбор обычными игровыми действиями, а не специальным интерфейсом. Это может помочь сохранить поток, потому что естественный ритм не нарушается. Это также позволяет избежать сложностей с интерфейсом реального диалогового дерева.

В Dungeons & Dragons Хозяин Подземелья играет роль каждого неигрового персонажа в игре. Он говорит за них и решает, как они будут реагировать на любые действия игроков. Нет необходимости ограничивать возможности игроков, поскольку, будучи настоящим человеком, Хозяин Подземелья может понимать и реагировать на любую ситуацию.

Реальные люди могут вместе создавать замечательные истории, если у них есть для этого мотивация. Проблема этого метода заключается в согласовании мотивации игрока с мотивацией персонажа. Это предполагает необходимость мотивировать каждого игрока правильно играть свою роль в игровом нарративе, а это крайне сложно. Этот метод работает в играх, в которые играют друзья лицом к лицу, потому что социальное давление побуждает людей к честности. В видеоиграх с анонимными игроками или в соревновательных играх трудно или невозможно мотивировать игроков так, чтобы при этом получилась длительная и насыщенная ролевая игра.

В многопользовательских играх для выполнения ролей игровых персонажей могут быть задействованы реальные игроки.

Кейс: Fallout 3

Давайте рассмотрим игровой нарративный опыт и разберем нарративные инструменты, использованные для его создания. Во-первых, я расскажу вам историю, которая случилась со мной, когда я играл в постапокалиптическую RPG Fallout 3 от Bethesda Game Studios, которая вышла в 2008 году. Итак, по порядку.

Игра начинается с рождения персонажа в Убежище 101. Оно построено под землей несколько столетий назад и предназначено для защиты жителей от ядерной катастрофы. Эта история начинается, когда персонаж игрока покидает Убежище в 19 лет.

Моя история

Пейзаж представлял собой ветреные просторы, усеянные мертвыми деревьями, разбитыми машинами и человеческими костями. Дверь Убежища была закрыта, и мне больше ничего не оставалось, кроме как идти вперед.

Через несколько минут я набрел на крошечное поселение под названием Мегатон. Блуждая по городу, я обнаружил магазин с вывеской «Магазин на кратере». Это здание, как и любая другая постройка в Мегатоне, напоминало свалку старого хлама. Его можно было узнать только по названию, написанному белой краской возле входной двери.

Вид внутри был не намного лучше, чем снаружи. Пыльный воздух, мигающая тускло-желтая натриевая лампа и ветхие полки на стенах. Интерьер, мягко говоря, не из лучших. Молодая женщина в грязном синем рабочем костюме подметала пол за прилавком, ее огненно-рыжие волосы были собраны в грязный хвостик. Я подошел к ней.

«Привет! — сказала она. — Слышала, ты из Убежища! Я не видела никого из вас в течение многих лет! Приятно познакомиться!» С каждым словом ее голос звучал все выше и выше. После мрачной Пустоши ее энтузиазм почти раздражал. «Я Мойра Браун. Я хозяйка “Магазина на кратере”, но на самом деле я в основном занимаюсь ремонтом и исследованиями, — на мгновение она замолчала. — Скажем так, я работаю над книгой о Пустоши — было бы здорово услышать Предисловие от обитателя Убежища. Поможешь мне?»

Она казалась приветливой, а мне были нужны друзья. «Конечно, — сказал я, — мне есть что рассказать о жизни в Убежище».

«Отлично! — ответила она. — Расскажи мне, каково это — жить под землей всю свою жизнь, или выходить на улицу впервые, или что-то, что поражает тебя больше всего!»

Я подумал, что она шутит, поэтому решил пошутить в ответ. «Это удивительное место, — сказал я. — В главной комнате я даже не вижу потолка!»

Мойра засмеялась. «Да, ты даже не представляешь, как трудно заменить вон ту большую лампочку наверху! Это здорово для Предисловия — начало с шутки и все такое. Для книги это будет хорошо. Хочешь помочь с исследованием? Я могу заплатить тебе, будет весело!»

«О чем книга?» — спросил я.

«Ну, это опасное место в Пустоши, верно? Людям бы пригодился действительно полезный сборник советов. Например “Руководство по выживанию в Пустошах”! Чтобы проверить мои теории, нужен помощник. Я бы не хотела, чтобы кто-то пострадал из-за ошибки. Никто не обрадуется, когда это произойдет. Нет... Затем они просто много кричат. На меня. Плохими, плохими словами».

Я задумался. «Звучит отлично! — сказал я. — Хочу уже поскорее приняться за дело. Чем я могу помочь?»

«Для начала еда и лекарства. Они периодически нужны всем, верно? Поэтому нужно хорошее место, чтобы найти их! Недалеко отсюда находится старая сеть Super-Duper Mart. Мне нужно знать, остались ли в этом месте еще еда или лекарства».

Я кивнул и попрощался. Всего час назад я покинул Убежище, и вот я уже подопытный кролик Мойры Браун.

Выйдя за городские ворота и следуя компасу, я направился к Super-Duper Mart. Вскоре я взобрался на холм и моему взору открылось то, что осталось от города Вашингтон. Разрушенные здания простирались до горизонта, образуя зубчатую границу на фоне желтого неба. Я направился туда.

Я нашел Super-Duper Mart на окраине города. У создателя магазина, должно быть, не хватало ни креативности, ни денег, потому что это была не более чем гигантская бетонная обувная коробка, которую можно было узнать только по огромному логотипу над парковкой.

Когда я вошел на парковку, я услышал гром выстрелов большого охотничьего ружья, чередующегося с сухим треском штормовой

винтовки. Я завернул за угол и обнаружил, что рейдер Пустошей сражается с человеком в старинном кожаном плаще. «В чем дело? Не выносишь вида собственной крови?» — кричал человек в плаще. Это были его последние слова. Рейдер застрелила его выстрелом из автомата, и он упал, захлебываясь кровью. Затем она повернулась ко мне.

Как и большинство рейдеров, она оделась так, чтобы произвести впечатление. На ней был обтягивающий черный комбинезон, покрытый шипами, на голове — двойной ирокез, а глаза были покрашены толстым слоем подводки, и это придавало ее лицу демонический вид.

Она стреляла из штурмовой винтовки. Я выстрелил из пистолета в ответ. Должно быть, я попал ей в руку, потому что она уронила оружие. Я продолжал стрелять, а она бросилась, чтобы поднять оружие, и укрылась за старой машиной.

Затем она появилась снова, на этот раз из-за машины. Я был застигнут врасплох, и она несколько раз в меня попала. Ситуация не слишком хорошая — мощности моего пистолета было недостаточно, чтобы соперничать с ее штурмовой винтовкой.

Я уже был готов впасть в отчаяние, но вдруг услышал грохот позади. Я обернулся и увидел женщину в коже, стреляющую в рейдера из огромной винтовки. Она снова выстрелила, и рейдер упала.

Я подошел к телу врага и забрал у нее все оружие, которое нашел, включая штурмовую винтовку и боеприпасы. Я даже взял ее одежду с шипами. Она была не в моем стиле, но я подумал, что смогу потом продать ее.

Как только я закончил мародерствовать, рядом со мной прогремел взрыв и все стало белым. Придя в себя, я понял, что произошло. Машина, которую рейдер использовала в качестве прикрытия, загорелась, когда в нее попали случайные выстрелы. Она продолжала гореть, пока я грабил ее труп, и только сейчас взорвалась.

Взрыв повредил мою правую ногу и левую руку. Я не мог целиться и двигаться. Я порылся в сумке и нашел устройство для лечения — Стимулятор. Химикаты текли по венам и исцеляли

конечности, и я снова мог их использовать. Я выпил ядер-колу, чтобы поддержать свои силы.

Когда я подошел к главному входу в Super-Duper Mart, перед магазином я заметил три трупа в неестественных позах. Это были не просто жертвы случайных убийств — это были рейдерские трофеи, которые она не стеснялась демонстрировать. Казалось, что рейдер на стоянке не просто проходила мимо и оказалась на ней случайно. Super-Duper Mart был рейдерской базой.

Я перезарядил пистолет и вошел в здание.

В магазине было темно. Солнечный свет изо всех сил старался пробиться сквозь окна, покрытые вековой грязью. Под потолком все еще горели несколько люминесцентных светильников, образуя желтые пятна света в удушающей пыли. На полу валялись тележки для покупок, а на полках нельзя было найти ровным счетом ничего.

Находясь за дверью, я никого не видел, но знал, что там кто-то стоит. Я прокрался в комнату, стараясь укрыться в темноте. Подойдя к кассе, я заметил одинокого рейдера, патрулирующего проходы с оружием в руках.

Я подкрался ближе, осторожно прицелился и выстрелил из пистолета. Пуля пролетела мимо головы рейдера и ударилась о стену магазина. Ответный огонь разразился со всех сторон, из укрытия появились новые рейдеры, которых моя атака привела в боевую готовность. Я отступал к кассе, пока вокруг меня свистели пули. Я нашел цели и выстрелил, убив несколько рейдеров.

Двое нападавших наступали слева. Один быстро навалился на мой пистолет. Другой нацелил на меня гигантскую винтовку и выстрелил. Я четырежды выстрелил ему в грудь. Один из выстрелов оказался удачным, но рейдер продолжал наступать, желая размазать мою голову по стенке, его винтовка неистово грохотала.

В отчаянии я спрятался за колонну. Просматривая свой инвентарь, я нашел штурмовую винтовку, которую украл на парковке у рейдера в шипованной одежде. Я зарядил его и затаился. Когда стрелок появился в поле зрения, я всадил в него длинную очередь из восьми патронов. Он упал с пронзительным криком.

В магазине стало тихо. Казалось, что бой окончен, и я начал собирать полезные вещи, которые остались после боя. Я взял доспехи и боеприпасы у мертвых рейдеров. У моей последней жертвы я взял охотничье ружье. Торговые автоматы, производящие Nuka-Cola. На полу в туалетах я нашел матрасы и лекарства. Казалось, именно здесь спали рейдеры. Холодильник был наполнен разной едой — первое, что Мойра хотела, чтобы я нашел.

Я прошел в заднюю часть магазина, чтобы найти лекарство, которое хотела Мойра. Там я увидел еще нескольких несчастных мертвецов, свисающих с потолка. Последний был прибит к стене в непонятной позе, напоминающей распятого Христа. Как и многие другие, он был без головы.

Пока я рассматривал его, услышал стрельбу из автоматов. Когда пули пронзили мое тело, я увидел свою кровь и закричал от боли. Обернувшись, я увидел засаду. Это был рейдер с автоматом, в мотоциклетном шлеме с рогами по бокам. Он продолжал стрелять, ранив меня в левую руку. Я отшатнулся назад, стреляя вслепую. Следующим выстрелом он ранил меня в ногу, я упал за аптечную стойку.

Я порывлся в сумке и заметил осколочную гранату. Я поднялся и бросил ее. Она упала у ног рейдера в рогатом шлеме и взорвалась, оторвав ему ноги по колено и подбросив его в воздух.

Магазин снова затих.

Я вылечил руку последним Стимулятором, который был в запасе, и начал собирать полезные вещи. Я взломал замки на некоторых коробках с боеприпасами и взял оттуда пули, гранаты и самодельные мины. На разных полках я обнаружил детали машин, остатки еды и книгу под названием «Байки торговца из Джанктауна».

Пытаясь проникнуть в подсобное помещение аптеки, я застрял, так как дверь захлопнулась, а взломать ее сложный механизм своими примитивными отмычками я не смог. Осматриваясь вокруг, я увидел металлический ящик, в котором нашел ключ от аптеки. Я вернулся, чтобы забрать его и открыть дверь.

Помещение аптеки было заполнено рядами разбитых стеллажей. Большая часть была захламлена мусором, но я нашел дротики, еще немного гранат, ликер, скороварку и миниатюрную ядерную бомбу. Я также нашел медикаменты, о которых спрашивала Мойра. Я использовал один из Стимуляторов, чтобы вылечить раненую ногу.

Выйдя из аптеки, я услышал голос по радио над магазином. «Мы вернулись. Кто-нибудь, откройте». Подождите, здесь что-то не так. Рейдеры входили в магазин через дверь с улицы, и я оказался в ловушке.

Бой был тяжелым, но я справился. К тому времени, как я вернулся в Мегатон, небо стало бледно-голубым. Мойра, как всегда, была в хорошем настроении. «А ты знал, что человеческое тело может выжить без желудка или селезенки? — восторженно заявила она. — О, что случилось?»

Анализ

Эта история — особый опыт, который игрок может получить, играя в Fallout 3. Два игрока никогда не испытают один и тот же опыт. Тем не менее его можно понимать как сюжет. У него есть ритм, изложение, начало и конец.

Fallout 3 использует много разных нарративных инструментов. Во всем вы можете найти историю мира — в ландшафте, архитектуре и мизансцене мусора, трофейного имущества и трупов. Одни части истории, такие как диалог Мойры, имеют жесткий заданный сценарий. Другие, такие как боевые стычки, — более гибкий.

Интегрированная история, которую переживает игрок, возникает в результате взаимодействия сценариев, игровых систем и решений игрока. Это происходит на всех уровнях — на микроуровне отдельных движений и атак, а также на макроуровне выбора миссий и направлений. Большое количество комбинаций говорит о том, что опыт каждого игрока уникален.

Моя история открывается через нарратив о мире. Столичная Пустошь представляет собой высушенную землю ландшафта.

Обгоревшие здания и машины рассказывают историю мира, испепеленного ядерным взрывом. Город Мегатон рассказывает свою собственную историю мира через архитектуру: лачуги из листового металла и нарисованные вручную знаки говорят о нелегкой жизни в условиях крайней нищеты. Внешний вид людей также весьма красноречив: грязный комбинезон Мойры Браун и незамысловатая прическа говорят о том, что женщина больше интересуется своей работой, а не популярностью. Вы можете подумать, что она чудачка.

Но история мира — это еще не все. Игрок переживает эту историю, выбирая, куда пойти и на что посмотреть. Таким образом, когда игрок блуждает по игровой вселенной, появляются два сюжетных потока: предыстория ядерной войны и эмергентная история персонажа игрока, бродящего по Пустошам после выхода из Убежища. Одна история гласит: «Этот город был построен отчаявшимися людьми». Другая: «Я вошел в город и исследовал то, что слева». Игрок переживает обе истории одновременно и чувствует эмоциональную отдачу от каждой из них.

Как только я начал говорить с Мойрой, игра переключилась с исследования нарратива о мире на диалоговое дерево. Все мои слова были выбраны из списка вариантов диалога, а ответы Мойры были заданы сценарием.

Чтобы избежать проблемы бесконечного разветвления истории, диалоговое дерево Fallout 3 часто возвращается к началу. Например, каждый раз, когда вы приветствуете Мойру, вы получаете один и тот же перечень параметров диалога, каждый из которых ведет к отдельной теме: покупка снаряжения, покупка мебели для дома игрока, местные сплетни, ремонт предметов, любые выполняемые миссии и так далее. Когда тема разговора исчерпана, игра возвращается к исходному списку тем. Таким образом, само диалоговое дерево организовано в виде разветвленной структуры.

Поход героя в Super-Duper Mart вызвал у игрока чувство ожидания предстоящей задачи. Эта часть истории не была прописана ни в каком сценарии, но она подразумевалась геометрией мира.

Моя встреча с рейдерами и одетыми в кожу охотниками за пределами магазина Super-Duper Mart была интересной конвер-

генцией нарративных инструментов. Рейдеры были заложены сценарием и будут появляться в одних и тех же местах всегда. Однако охотники не были частью сценария. Они появляются в Пустошах на протяжении всей игры. В моем случае их появление было случайным. Охотники и рейдеры, будучи враждебно настроенными по отношению друг к другу, сразу же вступили в бой, и эта бойня все еще продолжалась, когда я приехал.

Мое знакомство с этой битвой началось со слов, которые я услышал от охотника («В чем дело? Не выносишь вида собственной крови?»), и того, что я увидел выстрелы штурмовой винтовки рейдера. Безрассудно храбрый охотник, выкрикивающий свою последнюю угрозу, уже умирая, является сильным эмоциональным обменом. Что интересно, так это то, что он не смоделирован в игровой механике. Это апофеническая интерпретация, появившаяся в сознании игрока из случайных обрывков диалога и простых боевых взаимодействий.

После того как первый охотник умер, меня повалил огонь штурмовой винтовки второго рейдера. Этот короткий бой сформировал маленькую эмергентную историю со своей эмоциональной дугой. Будучи беззащитным, я почувствовал напряжение. После того как неизвестный охотник спас мне жизнь, я почувствовал облегчение и благодарность к своему спасителю. Казалось, что он спасает меня в качестве акта доброты или убивает, чтобы отомстить за своего убитого союзника. На самом деле все эти интерпретации — чистая апофения, но они кажутся реальными и влияют на игрока.

Бой в супермаркете создает еще одну мини-историю. Мой незаметный вход в помещение — это экспозиция. Она дала мне время разобраться в ситуации, прежде чем погрузиться в нее полностью. С появлением новых рейдеров напряженность усилилась. Она достигла своего пика, когда появился рейдер с винтовкой. Это напряжение спало, когда я вспомнил о штурмовой винтовке, взятой у убитого на стоянке рейдера, и о триумфальной контратаке. Эта маленькая сюжетная дуга похожа на сцену из боевика, но она не создана дизайнером, она возникла в результате взаимодействия игровых систем, сценариев и выбора игроков.

Финальная сцена с противником в засаде, стрелявшим из штурмовой винтовки, не была заложена в сценарий. Эта «засада» была внезапной, а у искусственного интеллекта в действительности не было никакого намерения — противник просто остался после основного боя. Но даже при том, что бой не был заложен в механике, благодаря апофении казалось, что этот последний выживший враг оставил ловушку.

Оленьи рога стали отличительной чертой последнего рейдера, тем самым выделив его. Он не просто рейдер; он необычный рейдер со странными рогами. Благодаря этому ярлыку дизайнерам проще создать истории о рейдере. Навешивание ярлыков — это одна из областей, которую можно было бы улучшить в Fallout 3. Большинство персонажей — просто ничем не выделяющиеся рейдеры. Если бы у них было больше отличительных особенностей — сумасшедший доктор, бармен, мастер, раб, — игроки могли бы придумывать о них более интересные истории.

Абсурд

В мире Fallout 3 присутствуют сильные оттенки абсурда: Мойра, переполненная энтузиазмом, рейдеры в шлемах с оленьими рогами и прочее. Эти юмористические совмещения просто необходимы. Если бы в игре нужно было только выживать в безысходном, мертвом мире, большинство игроков испытывали бы слишком сильную эмоциональную нагрузку. Случайные абсурдные моменты облегчают эту эмоциональную тяжесть.

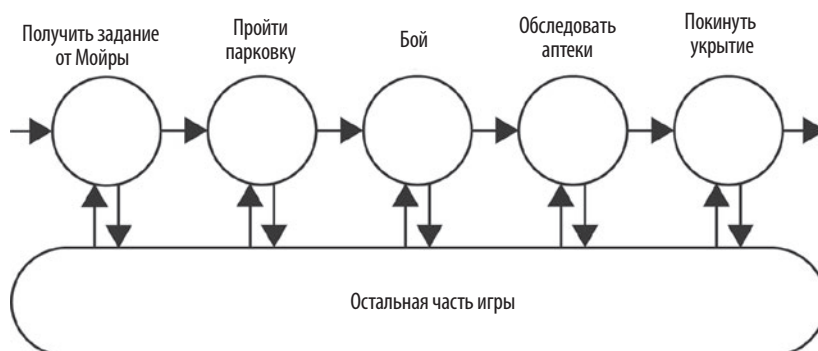
Абсурд также помогает оправдать менее реалистичные части игры. Например, глупое представление радиации в Fallout 3 позволяет бросать в игрока всевозможных странных существ: от гигантских мух до девятиметровых человекоподобных демонов. Если бы радиация была смоделирована реалистично, все это было бы невозможно.

Наконец, абсурд уменьшает влияние неизбежных логических противоречий в эмергентных историях игры. Например, однажды я подошел к охраннику каравана и трижды выстрелил ему в затылок. Он повернулся, нахмурился и сказал: «Кажется, я что-то слышал!» Вместо того чтобы подумать, что здесь что-то не то, игрок просто рассмеется.

Упорядочивание контента

Контент упорядочен как по сценариям, так и по геометрии мира. Например, игрок должен пройти через парковку, прежде чем войти внутрь. Оказавшись внутри, он должен пройти через главное помещение, прежде чем отправиться в аптеку. Наконец, рейдеры, входящие в дверь, по сценарию появляются только после того, как игрок обследовал аптеку.

Игрок может выйти и вернуться к миссии в любое время. Он может пройти половину, уйти и вернуться через 20 часов игры, чтобы закончить миссию. Это создает огромное количество возможных путей в игре, так как игрок совмещает много различных миссий.



Темп

Темп этой истории неравномерный. Напряженные моменты боя чередуются с более длительными диалогами, исследованиями, лечением и сбором артефактов. Эта смесь сохраняет вовлеченность игроков, не истощая их.

Мы могли бы утверждать, что темп этой истории выиграет, если игрок почувствует больше прогресса и сможет участвовать в более интенсивных сражениях по мере продолжения миссии. Однако для этого нам пришлось бы учесть и кривую ритма, и тот факт, что игроки могут приостанавливать миссии и возвращаться к ним в любое время. Игроки, которых легко отвлечь, скорее всего, не смогут оценить по достоинству наш тщательно продуманный

темп, и все наши старания, вероятно, вылетят в трубу. В такой непредсказуемой обстановке стабильный ритм игры с периодическими всплесками напряжения может стать лучшим решением, поскольку эта модель представляется нам весьма устойчивой.

Сеттинг

События Fallout 3 разворачиваются в постапокалиптическом мире. Этот сеттинг является хорошей основой игры, оправдывая хорошую и гибкую игровую механику.

Например, логично, что умирающие от голода, отчаявшиеся люди ожесточаются. Повсеместное насилие поддерживает игру несколькими способами. И оно дает возможность рассказывать эмергентные истории за счет того, что персонажи бегают и стреляют, а не просто стоят и разговаривают. Благодаря такому подходу решается проблема непосредственного взаимодействия между людьми.

Безжизненный мир оправдывает очень низкую плотность населения. Это соответствует техническим и конструктивным ограничениям игры. Дизайнеры избавлены от необходимости создавать еще большее количество персонажей. И это объясняет отсутствие больших толп людей, с которыми игровая технология не справится.

Решения

Мозги сидели за обеденным столом, их влажные неокортикальные складки блестели в свете свечей.

«А ты, Альберт? — спросил узкий серый мозг. — Что тебе сейчас нравится?»

Альберт, широкий розовый мозг, дрожал, как желе.

«Мне нравятся побольше, и не слишком часто. Мне нужно что-то, что я смогу подольше переваривать».

«Тьфу, — сказал Исаак, длинный цилиндрический мозг. — Кому вообще нравится так долго жевать? Я предпочитаю быстро, и чтобы много маленьких кусочков. Они должны помещаться во рту».

За столом велись ожесточенные споры — много, мало, разнообразный, последовательный, большой, маленький — а официанты тем временем разносили мозгам изысканные блюда.

В целом, интерактивный опыт — это сложная путаница взаимодействий, мыслей и эмоций. Чтобы понять интерактивность достаточно хорошо, чтобы ее можно было создать, нам нужно изучить отдельные составляющие интерактивности. А именно — решения¹.

В некоторых играх увидеть решения легко. В покерных комбинациях игрок должен выбрать — скинуть или уравнивать (фолд или колл). В Civilization V игрок должен решить, стоит ли сейчас вторгаться к вавилонянам или ждать другого хода. Игры, подобные этой, передают нам возможность принимать решение шаг за шагом, каждая из них представляет собой уникальную и отлично сложенную головоломку.

¹ Когда автор пишет о «дизайне решений», на самом деле он говорит о том, как создать ситуацию постановки игрока в условия выбора. Совершая выбор одной из возможностей, игрок будет принимать решения, о которых и говорится в этом разделе. — *Примеч. ред.*

В других играх все не так просто. В многоуровневых играх в реальном времени решения плывут в одном потоке, как пузырьки в соломинке. Они пересекаются, объединяются и разделяются в непрерывном танце восприятия и мысли. В StarCraft II профессиональный игрок управляет одной атакой, одновременно защищаясь от другой, руководит разведчиком и развивает свою экономику. В боксе боец следит за своим здоровьем, стойкой, атакой и защитой, ежесекундно решая, что делать с каждой частью своего тела. Различные мыслительные процессы соперников настолько пересекаются, что становится трудно определить индивидуальные решения. Но даже если нам сложно разграничить эти решения, они все еще там. И они важны, потому что решения — это мощный источник эмоций.

Когда мы объясняем решение, мы прибегаем к логическим рассуждениям. Мы описываем, почему и как возникли обстоятельства, которые привели к такому решению, желаемую цель и как наш выбор приводит нас к этой цели. Но на самом деле логика — это только часть процесса выбора. Другая часть — и часто это самая важная часть — управляется эмоциями. Чисто рациональное человеческое решение — это миф, не более реальный, чем Спок из телесериала «Звездный путь».

Независимо от того, выбираете ли вы, какую пару носков надеть, разводиться или как ходить в шахматах, ваше решение зависит от того, как вы относитесь к каждому решению. Вы смотрите на черную пару носков и чувствуете приступ тоски. Вы смотрите на оранжевую пару, и внутри вас что-то загорается. И вот ваша рука уже тянется к оранжевой паре. Или вы думаете о том, что нужно походить королевой, но ваш натренированный инстинкт шепчет «нет». Вы смотрите на пешку, и вам легче. Итак, вы ходите пешкой. В каждом случае какая-то часть бессознательного оценивает решение и вызывает эмоции, которые определяют ваше поведение. Эти эмоции являются частью игрового опыта.

Понимание решений имеет ключевое значение в геймдизайне, так как они являются единственным эмоциональным триггером, который уникален для игр. Многие медиа могут вызывать эмоции через зрелище, героя или музыку; и только игры могут сделать это через решение.

Но создание систем, генерирующих решения, чрезвычайно сложно. Сами решения могут принимать разные формы и масштабы. Они могут быть важными или несущественными, сложными или легкими. Одни включают много информации, другие — мало. Они могут быть кристально понятными или неопределенными. Одни принимаются в спешке, другие допускают длительное созерцание. Они могут быть собраны воедино или разрознены. Каждый вариант и комбинация создают разные эмоциональные оттенки.

Задача усложняется тем, что мы не можем разрабатывать решения напрямую, не нарушая их. В отличие от любой другой области, чтобы геймдизайн работал хорошо, решения в нем должны быть мгновенными. Поэтому вместо того, чтобы писать их по одному, мы должны создавать системы, которые будут генерировать их сразу же.

Дизайн решения — это геймдизайн в чистом виде. Хотя игры можно улучшить за счет нарратива, сюжета, изображения и звука, ни один из этих аспектов не существен для формы. Сердце игр — интерактивность, а сердце интерактивности — момент принятия решения.

Чувствуя будущее

Большинство эмоциональных триггеров описываются как результат того, что происходит в настоящем. Вы видите страшного монстра, и вам страшно. Вы одерживаете победу и чувствуете себя победителем. В каждом случае происходит событие, сознание игрока воспринимает это событие и мгновенно вызывает эмоциональную реакцию. Причина и следствие очевидны и интуитивно понятны.

Но решения не создают эмоции вот так, потому что решения не связаны с настоящим. Решения касаются выбора из нескольких возможных вариантов развития событий в будущем. Эмоции, вызванные решениями, связаны не с тем, что произошло, а с тем, что может случиться.

Представьте, что вы стоите на краю Эмпайр-стейт-билдинг. Вы стоите на самом краю, над пустотой, а в спину дует порывистый

ветер. Вы смотрите вниз с 86-го этажа и видите машины, ползущие, как жуки. Вам до смерти страшно. Эмоции, отшлифованные тысячами естественного отбора, кричат вам: *отойди назад!*

Чтобы вызвать эмоции у игрока, не обязательно что-то должно произойти. Игроку нужно только почувствовать вероятность того, что это может случиться.

А теперь представьте, что вы летом приехали в гости к другу и стоите на краю крыльца его дома. Вы точно так же стоите на самом краю крыльца, а порыв ветра дует в спину. Вы смотрите вниз и видите божью коровку, ползающую по краю лепестка. Вы чувствуете умиротворение.

В этих двух ситуациях нет никакой разницы в событиях, которые произошли. В обоих случаях вы стояли на краю и ветер чуть не столкнул вас вниз. Ни в том ни в другом случае вы не умерли. Разница только в будущих событиях.

На крыльце ваше бессознательное не обнаружило ничего опасного в ближайшем возможном будущем, поэтому у вас не было необходимости в сильных эмоциях, которые заставили бы вас принять любое решение. Но на небоскребе ваше бессознательное обнаружило непосредственную вероятность смерти. Таким образом, через чувство страха, оно пыталось повлиять на то, чтобы вы приняли решение. Вам было страшно, хотя *ничего не случилось*.

Это важно. Мы привыкли думать о процессе создания развлечений как о процессе принятия решений. Обычные сторителлеры думают и говорят о последовательности событий в сюжете. Геймдизайнеры рассказывают о ситуациях, которые внезапно возникнут из механики. Существует негласное мнение о том, что все, чего не произошло, уже не важно. Но способность человека чувствовать эмоции в отношении возможного будущего означает, что это не так.

В играх игрок не просто чувствует происходящее. Процесс принятия решений мысленно взаимодействует со всеми возможными результатами, которые может обнаружить его разум. Его бессознательное управляет вымышленным миром, уходящим в будущее, где игрок побеждает, проигрывает, живет и умирает. Эти возможные результаты влияют на эмоции независимо от того, есть они или их нет.

Поэтому геймдизайнеру недостаточно думать о том, что будет делать игрок и как поведет себя игра. Мы также должны по-

думать о намерении игрока, о чем он думает и как игра может на это отреагировать. И хотя многие из этих ситуаций никогда не произойдут в реальности, они все равно влияют на игровой опыт, потому что сознание игрока чувствует эту вероятность.

Высококласные шахматисты проводят много времени, молча глядя на доску. Наивный наблюдатель может сетовать: «Почему они ничего не делают?» Но они делают. Они принимают решения. Их тела остаются неподвижными, но разум всецело поглощен шахматным миром, разрывается между множеством вариантов, танцует между тысячами возможных ходов и контрходов, охотится за жемчужиной, которая решит загадку. Их сердце выскакивает из груди, если они чувствуют возможные атаки, а затем видят неизбежную реакцию. Именно качество этого внутреннего процесса принятия решений, а не движение фигур на доске, делает шахматы увлекательными.

Возможно и обратное. Плохо разработанная игра может быть активной и яркой, но совершенно бесполезной для значимых решений. Слабые игровые системы и пробелы в возможности принимать решения ослабляют интерес независимо от того, сколько мечей или гоночных машин появится на экране. Игрок не будет сыт одним только экшеном. Ему нужны ожидание, неопределенность, логика и решение.

Например, у одной крупной студии две отдельные франшизы, которые работают в основном по одной и той же технологии и используют много игровых возможностей. Просто наблюдая за ними, можно увидеть, что они очень похожи. В обеих играх персонаж игрока взбирается по стенам и преодолевает препятствия, сражается с врагами с помощью меча. Но игровой опыт в этих двух играх очень отличается, так как структура мира удовлетворяет потребность игроков в принятии решения.

Первая игра разворачивается в открытом мире. Герой может бегать, лазать и прыгать куда угодно, чтобы подойти поближе или убежать от своих врагов. Иногда он может действовать коварно и наносить удар с крыши. В любое время он может атаковать врагов на улице мечом. Он может бегать или лазать в любом направлении огромного мира. Это значит, что каждый момент

содержит сотни возможных вариантов будущего. Игрок может заглянуть в будущее и увидеть, как он плывет по реке, прячется на крыше или сражается не на жизнь, а на смерть. И его разум постоянно обдумывает эти возможности, оценивает их, подталкивает его к тому или иному выбору с помощью эмоций. Даже если он сознательно не думает об этом будущем, он его всегда чувствует.

Элементы управления второй игры очень похожи на управление в первой, за исключением того, что игра является линейной. В каждую зону ведет только один вход, один выход и одна последовательность прыжков и траекторий, которые позволяют игроку идти дальше. Разуму не дается возможности думать о более чем одном возможном развитии событий, а это значит, что игрок лишен возможности принимать решения. Игрок просто выполняет какое-либо действие, которое ему доступно в любой точке, поэтому игра эмоционально пуста. Несмотря на то что действия на экране такие же, как в первой игре, мыслительный процесс, стоящий за ними, и сопутствующие эмоции очень разные.

Предсказуемость

Назвать игру предсказуемой обычно звучит как оскорбление. Но не стоит на это обижаться, потому что предсказуемость важна для принятия разумных решений.

Чтобы спрогнозировать возможное развитие событий, игрок должен почувствовать и понять их. Стоять на краю небоскреба страшно, потому что понимаешь, как мало нужно для того, чтобы упасть и разбиться насмерть. Если бы вы не понимали, что про-

исходит, возможность смерти не пугала бы вас, потому что вы не почувствовали бы, как она дышит вам в затылок.

Не имея возможности прогнозировать развитие событий, игрок не сможет планировать и принимать решения. Можно иметь миллион интересных событий в будущем, но какой в них смысл, если они потерялись в тумане неизвестности. Если это происходит, процесс принятия решений исчезает. Игрок не может заглянуть в будущее, поэтому игра превращается в бесполезное эмоционально

Если мы хотим, чтобы решение было хорошо продуманным, его исход не должен быть ни непредсказуемым, ни неизбежными. Он должен быть частично ожидаем.

пустое упражнение на реакцию, когда игрок просто максимально быстро реагирует на происходящее.

В то же время будущее не может быть полностью предсказуемым. В этом случае игрок не сможет принять продуманных решений, потому что его мозг не проделает умственную работу, чтобы найти путь к желаемому результату. Нам нужно выбрать наиболее благоприятный исход.

Возьмите чашку, вытяните руку вперед и отпустите чашку. Что случится?

Очевидно, что мы все знаем ответ: разбитый фарфор и пролитый чай.

Мы знаем результат потому, что интуитивно понимаем бытовую физику. Мы знаем законы гравитации, поэтому то, что упало с большой высоты, упадет на пол. Мы знаем, что чашка сделана из хрупкого материала, который разобьется, если сильно по нему ударить. Это все часть системы правил — физики, — которые мы понимаем. Поскольку мы понимаем эту систему, то можем спрогнозировать результат.

Бессознательное определяет возможные варианты будущего на основании знаний о системах, которые управляют Вселенной. Физика — одна из них, но существует и много других систем: психологических, экономических, механических и так далее. Разум вашего ребенка — это система. Так же как и организация вещей на вашем рабочем месте и газонокосилка в гараже, которую нужно починить. Чувствовать будущее — значит сначала увидеть его с помощью ментальных моделей и спрогнозировать развитие событий в ответ на любое действие.

Тот же принцип применяется в прогнозировании в играх. Основное отличие заключается в том, что системы здесь представляют собой искусственно разработанные схемы игровой механики. Но для прогнозирования, в основе которого будут лежать эмоции, эти механические системы должны иметь определенные свойства.

Возможность прогнозирования потенциального развития событий предполагает, что событие возникает в результате последовательной, понятной системы.

Во-первых, они должны быть последовательными. Как гравитация, они должны работать одинаково, чтобы полученные знания можно было применить в разных ситуациях. Если правила постоянно меняются, прогноз невозможен.

Во-вторых, они должны быть понятными. Иными словами, они должны быть достаточно простыми, чтобы игрок мог на них сосредоточиться. Последовательная система непредсказуема, если ее невозможно понять.

Если игровые системы отвечают обоим этим критериям, игрок сможет видеть и таким образом чувствовать будущее, вследствие чего у него будут возникать всевозможные эмоции, вызванные ожидаемой угрозой или возможностью выжить. Но если игровые системы противоречивы или непонятны, эти эмоции исчезают. Даже при наличии угрозы и возможности игрок не сможет их почувствовать, так как просто не заметит их в тумане непоследовательных или непонятных систем.

Например, Марио всегда прыгает одинаково. Максимальная высота его прыжка всегда одна и та же. Управление и скорость падения никогда не меняются. И эти системы просты: всего несколько правил и цифр. Это значит, что Марио прыгает благодаря последовательной, понятной системе. Следовательно, игрок может посмотреть на полосу препятствий и точно понять, где выход, с какими опасностями столкнется персонаж и как он может победить врагов. Он может представить различные пути и их исходы и прочувствовать каждый из них.

Если бы у Марио не было никакой системы прыжков или она была бы непостижимо сложной, предсказуемость исчезла бы. Игрок не смог бы спланировать траекторию прыжка. Не имея возможности представить подобные мысленные образы развития событий, игрок чувствовал бы себя некомфортно или боялся бы потенциальной опасности. Все, что он мог бы сделать, — это мгновенно реагировать на ситуацию.

Прогнозирование и заранее определенные решения

Прежде чем мы продолжим, я хотел бы кое-что прояснить. Когда люди говорят о принятии решений в играх, они часто упомина-

ют заранее определенные сюжетные линии, как в старой книге «Choose Your Own Adventure»¹. В таких случаях геймдизайнер четко определяет для игрока каждую из опций и все возможные исходы, к которым приводит выбор этих опций.

Например, дизайнер может написать сценарий, согласно которому один из персонажей предложит игроку героин. Если игрок согласится, последовательность, заданная сценарием, произойдет там, где игрока будут преследовать полицейские. Если игрок откажется, на него нападет преступник. Результат выбора, а также все остальные возможные исходы, к которым привело решение игрока, — все это будет разворачиваться одинаково, в соответствии с задумкой дизайнера.

Такие предопределенные решения отличаются от решений, которые я опишу в этой главе, потому что их исход не определяется игровыми системами, которые игрок может изучить и просчитать. Их задает дизайнер в зависимости от того, что должно произойти. Так, ход мыслей при принятии такого решения будет в корне отличаться от хода мыслей, которые появляются в результате решений, обусловленных механикой.

В лучших предопределенных решениях игрок выбирает по причинам, обусловленным сюжетом. Он может отказаться от героина, потому что хочет быть хорошим парнем. Или он может согласиться, потому что хочет быть плохим. В худшем случае игрок предугадывает последствия, чтобы попытаться получить оптимальный результат. Он проанализирует предыдущие привычки дизайнера и решит взять героин или отказаться от него в зависимости от того, что принесет ему больше денег. В любом случае, мышление — это не то же самое, что и выбор, обусловленный механикой, такой как шахматный ход или атака в файтинге. Игрок не продумывает игровые системы. Он думает о сентиментальных ценностях или пытается угадать мысли дизайнера.

¹ Имеются в виду книги-игры, в определенной степени популярные и в России: переводные издания серии «Боевое фэнтези» и некоторые другие, например известные книги-игры московского писателя и педагога Дмитрия Браславского: «Подземелья Черного Замка», «Верная Шпага Короля» и другие. — *Примеч. ред.*

Предопределенный выбор служит конкретной цели. Он дает игроку возможность встать в один ряд с дизайнером и быть соавтором нарратива, что в определенной мере невозможно в случае с выбором, который нас заставляет делать система. Мы не можем создать систему, симулирующую человеческий разум или весь нарративный мир, поэтому, если игрок собирается как-то на них повлиять, он должен принять заранее продуманные решения. Но не следует путать предопределенный выбор с выбором, который нас заставляет делать система. Они написаны по-разному и вносят свой вклад в опыт разными способами.

Прогнозирование и искусственный интеллект

Существует распространенное предположение о том, что чем умнее ИИ, тем он лучше. Аргументация такова: игра — это симуляция некоего вымышленного мира с людьми в нем. Чем точнее симуляция, тем лучше. Следовательно, чем точнее симуляция сознания персонажей, тем лучше. И поскольку реальные люди всегда умнее, чем игровой ИИ, то чем умнее ИИ, тем он лучше и тем больше он похож на реального человека.

Проблема заключается в том, что люди в играх не должны воспроизводить реальных людей, потому что игры — это не симуляция жизни. Игры — это механика, дополненная сюжетом, а не механика, которая заставляет вас думать, что люди настоящие. Сюжетная оболочка улучшает и передает механику, а не лежит в ее основе.

И дизайн механики часто страдает от более умного ИИ, потому что часто более умные ИИ непредсказуемы. ИИ, имеющий несколько простых характерных особенностей, можно просчитать. Игрок может планировать тактику на несколько шагов вперед, потому что он знает, что будет происходить на каждом этапе. Но спрогнозировать поведение очень умного ИИ со сложным многоуровневым интеллектом невозможно. Игрок не может планировать свои действия, потому что ими управляет слишком сложный интеллект. Чем больше думает ИИ, тем меньше может думать игрок.

В большинстве случаев ИИ лучше всего рассматривать с точки зрения механики, а не как симуляцию реального интеллекта.

Персонаж, который следует прямым предсказуемым последовательным правилам, часто вносит большой вклад в игровой опыт, чем реалистично хаотичная имитация интеллекта.

Например, поведение большинства солдат в стратегических играх в реальном времени скорее напоминает поведение роботов, а не людей. Получив приказ, они выполняют его точно и немедленно и каждый раз делают это одинаково. Это значит, что один и тот же бой всегда заканчивается одинаково и что всегда действуют одни и те же контрстратегии. Именно эта системность дает игрокам возможность планировать сложные стратегии. Если бы у солдат был сложный внутренний ИИ, одно и то же сражение заканчивалось бы то победой, то поражением, так как ИИ решил бы пойти налево, а не направо, или потерял и восстановил бы боевой дух. Будь это ИИ-солдаты, именно они, а не игрок, определяли бы исход боя, а это не то, что нам нужно. Роботоподобные солдаты не добавляют особого смысла, но их механистическая предсказуемость позволяет игрокам выработать определенную стратегию и быть уверенными в том, что она позволит играть закономерно.

Исключением из этого правила являются те проекты, в которых сюжет важнее, чем решения, которые принимаются на основе механики. Если эмоциональные триггеры сосредоточены на нюансах взаимодействия между ИИ-персонажами, стоит пожертвовать предсказуемостью простого ИИ.

Информационный баланс

Характер, сложность и комплексность решения зависят от информации, которой располагает игрок при принятии решения. Процесс принятия одного и того же решения может быть невозможным из-за недостатка информации, идеальным при наличии необходимого количества информации и чересчур простым при избыточном количестве информации.

Это значит, что игрок может вовсе не менять решения, но мы можем влиять на процесс принятия этого решения, просто давая либо больше, либо меньше информации.

ИНФОРМАЦИОННЫЙ БАЛАНС — это предоставление игроку информации в том объеме, которого ему будет достаточно для принятия решения, при этом ему будет все понятно, но не абсолютно очевидно.

Я называю подобные оптимизации решений информационным балансом.

Это элегантный метод геймдизайна — он не требует изменения самой механики. Игра остается прежней. Все, что мы делаем, — это скрываем или раскрываем разные ее части, а это обычно быстрее и проще, чем менять процесс работы системы.

Информационный баланс может иметь перекося в две стороны — *информационное голодание* и *избыточность информации*.

Информационное голодание

Непоследовательные или непонятные системы — не единственный способ разрушить прогнозирование. Если у игрока недостаточно информации о происходящем, прогнозирование также может нарушиться. Это называется информационным голоданием.

Чтобы сократить в игре информационное голодание, вам не нужно прикладывать особых усилий. Просто скройте большой объем информации. Отделите часть шахматной доски картоном и играйте, не видя при этом другой половины доски. Играйте в Magic: The Gathering, при этом не показывайте противнику существ, артефакты и чары. Играйте в теннис в темноте. В каждом из этих случаев игра превращается в борьбу и случайный выбор.

В редких случаях информационное голодание является целью дизайна. Например, в таких играх, как «Морской бой», и большинстве азартных игр обычно не хватает информации. Это работает, потому что первичные эмоциональные триггеры этих игр не строятся вокруг принятия решений. В этих играх произвольный выбор, обусловленный информационным голоданием, является преимуществом, поскольку он делает игру более понятной и социальной благодаря уменьшению соперничества. Но в играх, которые основаны на принятии решений, информационное голодание может быть смертельным.

Некоторые случаи информационного голодания очевидны, и их относительно легко устранить. Мы можем увеличить радиус обзора боевой единицы в стратегической игре или открыть больше карт в карточной игре. Но в других случаях информационное

голодание может возникнуть из-за взаимодействия других, казалось бы, не связанных элементов, которые трудно предугадать.

Например, распространенная нерешенная причина информационного голодания — это то, что я называю проблемой подготовки авторского вызова. Структура многих игр представляет собой последовательность авторских вызовов. Игрок проходит первый уровень, затем второй, затем третий и так далее. Информационное голодание появляется, когда игра просит игрока подготовиться к последующим испытаниям до того, как он их увидит. Игрок не знает, как это сделать, так как у него нет внутриигрового способа спрогнозировать, каким будет следующий вызов.

Проблема подготовки авторского вызова появляется постоянно, даже в отличных играх. Например, ролевые игры часто начинаются с того, что игроку предлагается самостоятельно создать своего персонажа. Игрок должен выбрать расу (человек, эльф, гном), класс (боец, колдун, вор), навыки, атрибуты и так далее. Эти решения очень важны; они влияют на все, что происходит в течение оставшейся части 50-часового опыта. К сожалению, это не самое удачное решение. В начале игры игрок вообще ничего о ней не знает. Он не чувствует баланс боя или какие инструменты для него предпочтительнее. Он не знает, с какими врагами столкнется.

Сталкиваясь с таким критическим выбором и не имея при этом достаточно данных, игроки часто прибегают к наиболее знакомому и безопасному варианту, упуская самые интересные части игры. Например, в *Mass Effect 2* было шесть классов, от Адептов до Инженеров. Но опрос, проведенный после релиза, показал, что 80% игроков выбрали Солдат — самый знакомый и тривиальный класс в игре. Учитывая множество непонятных опций, игроки просто выбрали то, что было наиболее понятно, и поэтому упустили большую часть достоинств игры.

Другая распространенная нерешенная причина информационного голодания — неоднозначность информации, которую игрок получает через сюжет.

Критическая информация для принятия решений, которая нужна игрокам, обычно является чистой механикой: статистика урона,

скорость передвижения, структура миссий, различные математические таблицы для экономики игры. Эту информацию трудно передать через сюжет, потому что ее там нет. Сюжет говорит о том, что игрок стреляет из винтовки по цели, он должен наносить значительный урон в зависимости от дальности, места попадания, случайно поразивших цель пуль и многих других аспектов. Механика говорит, что пуля — это мгновенное ранение, которое отнимает определенное количество очков здоровья. Это количество не может быть передано через сюжет, поэтому часто о нем вообще не говорится. В этих случаях часто лучше опустить сюжет и дать часть этой информации о механике напрямую.

Иногда проблему информационного голодания решает интернет. Такие сайты, как GameFAQs.com, предоставляют массивные текстовые файлы FAQ (часто задаваемые вопросы), написанные игроками, которые описывают все возможности и объясняют уровни в игре, часто включая скрытые статистические данные, которые игроки сами никогда бы не узнали. На первый взгляд FAQ кажется хитростью, которая может испортить игру. Но в играх с недостаточным количеством информации FAQ могут значительно улучшить игровой опыт. С помощью вопросов FAQ игроки могут планировать, прогнозировать и принимать решения осмысленно и целенаправленно, развивая опыт. Поскольку они понимают значение своего выбора, этот выбор сразу же становится интересным.

ИНФОРМАЦИОННОЕ
ГОЛОДАНИЕ —
трудноразрешимая
задача. Дизайнеры
просто не замечают
его, поскольку слиш-
ком хорошо знают
игру, а также просто
не хотят признавать
того, что оно
существует.

Полезный FAQ — это тревожный знак. Если текстовый файл делает игру заметно лучше, то в игре явно недостаточно информации. В ней много достоинств — дизайнер просто должен помочь игрокам понять их, дав немного больше дополнительной информации.

Проблема информационного голодания весьма распространенная, потому что она прячется от дизайнеров. Это происходит по двум причинам.

Во-первых, дизайнеры априори не замечают его. Дизайнер знает о своей игре все — гораздо больше, чем узнают большинство игроков. Мы можем сделать вид, что не знаем о проблеме, но на самом деле это будет не так. Если

не тестировать игру или не выявлять ее неисправности каким-либо другим способом, можно легко поддаться заблуждению и решить, что игра работает хорошо, хотя на самом деле тот, кто не знает игру от и до, будет умирать от информационного голодания и вообще не сможет в нее играть. Но информационное голодание скрывается и другим, еще более коварным образом. Оно использует эмоциональный шантаж, чтобы заставить нас не хотеть его найти.

Так приятно видеть, как после нескольких недель усилий новая механика наконец заработала. Для многих дизайнеров это чувство выполненного долга является основной мотивацией работать. Но охота за информационным голоданием ставит это чувство под угрозу. Это грозит тем, что чувство успеха было просто иллюзией, что перед настоящими игроками представленная на данном этапе механика будет просто бессмыслицей. Это ужасное разочарование для дизайнера, и эмоционально сложно заставить себя искать в своей работе то, чего ты не хочешь там найти.

Но это суровая необходимость. Информационное голодание в конце концов всегда находится. Лучше найти его до релиза, когда еще можно все исправить.

Избыточность информации

Недостаток информации заставляет игрока принимать запутанные решения наугад, а избыточная информация вообще лишает такой возможности. Решение — это умение увидеть правильный ответ на основании предоставленной информации. Если информация сама дает ответ, игрок совсем не думает. Мыслительный процесс исчезает; решение больше не является решением.

Это означает, что иногда мы можем сгенерировать решение из ничего, просто скрыв информацию. Не давая игрокам ответа, мы даем им интересную задачу для решения.

Например, в *Modern Warfare 2* игрок может прикрепить датчик сердцебиения к боковой части своей винтовки. Датчик отображает карту с расположением ближайших врагов перед игроком, даже если они находятся за стенами.

Если бы это было все, то можно было бы говорить об ужасном дизайне. Лучшее, что предлагает Modern Warfare 2, — это задачи, в которых игрок должен выяснить, где находятся враги. Игрок, которому предоставлена полная информация о вражеских локациях, может спокойно исследовать местность, не боясь выглянуть из-за угла, и ему никогда не придется угадывать движения врага. Игра превратилась бы в реактивную стрельбу, а игра в кошки-мышки прекратилась бы.

Чтобы решить эту проблему, нужно извлечь информацию. Но как именно?

Как бы вы это сделали? Что бы вы изменили в датчике сердцебиения, чтобы снова стало интересно, не убирая его из игры? Подумайте об этом, прежде чем читать дальше.

Дизайнеры Infinity Ward изменили датчик, добавив два ключевых ограничения. Во-первых, он не распознает врагов, у которых есть способности ниндзя. То есть если датчик показывает, что там никого нет, это еще ничего не значит.

Ниндзя все еще может быть там, поэтому прежде, чем повернуть за угол, игрок должен подумать и решить, безопасно ли это. Кроме того, это также повышает сложность стратегии для опытных игроков: умный игрок запомнит, у кого из его соперников есть ниндзя и сколько их, и на основании этого будет принимать решение и, возможно, даже полностью откажется от датчика сердцебиения.

Во-вторых, датчик не показывает местоположение врагов постоянно. Скорее они отображаются периодически, каждые 3,5 секунды, как на экране радара.

Между ударами пульса датчик ничего не отображает. То есть даже если датчик что-то и отображает, игрок должен хорошенько подумать, чтобы выяснить, в какую сторону направился враг, после того как мелькнул на экране датчика. А в Modern Warfare 2 опытный противник может запросто напасть на вас сбоку и убить, пока вы что-то там выясняете. Опять же, это еще один плюс к стратегии: опытные игроки поймут, когда их отслеживают на датчиках сердцебиения, и намеренно начнут перемещаться, чтобы не оказаться в том месте, где их обнаружил датчик.

Эти ограничения существенно влияют на игру с датчиком. При этом какие-либо механические взаимодействия вообще не изменились. Урон от оружия, движения и среда те же, что и раньше. Но медленно уменьшая поток информации, мы создаем совершенно новые виды решений и стратегий.

Избыточность информации — это не столько провал, сколько упущенная возможность. Игра функционирует нормально, даже если в ней есть избыточность информации. Тестировщики не сбиваются с толку и требуют помощи. На самом деле игра проходит слишком гладко, потому что они все ее понимают. Вот почему часто самое сложное в борьбе с избыточностью информации — это в первую очередь осознать, что она есть.

Как скрыть информацию

Мы видели, как избыточность информации приводит к тому, что игрок перестает принимать решения, потому что все знает. Поначалу кажется, что решения возможно принимать только в так называемых играх с неполной информацией.

Например, покер относится к играм с неполной информацией, потому что вы видите карты других игроков. Напротив, шахматы — игра с полной информацией, поскольку оба игрока могут видеть все происходящее на доске.

Но это традиционное различие не охватывает все способы, с помощью которых можно скрыть информацию, связанную с принятием решений. Даже играя в игры с полной информацией, как кажется на первый взгляд, игроки могут принимать решения, потому что информация может быть скрыта с помощью других, менее очевидных способов.

Игры с полной информацией раскрывают настоящее, но не возможное будущее. Мы можем увидеть, как будет выглядеть вся шахматная доска после того, как мы сделаем один потенциальный ход, но не сможем увидеть, как она будет выглядеть в конце игры. Эта информация о будущем скрыта за цепочкой взаимодействий, и чтобы ее получить, придется попотеть.

В играх с неполной информацией часть игрового состояния скрыта от некоторых игроков.

Информация может быть скрыта в будущем за цепочкой сложных причин и следствий.

Представьте себе, насколько проще было бы играть в шахматы, если бы вы могли читать мысли противника, знать его запланированные ходы, слабые места, которые он усмотрел в вашей позиции, и слабые места его собственной позиции. Эта информация является частью игры так же, как и расположение фигур на доске, но она скрыта.

Информация скрыта
во внутренних состоя-
ниях игроков.

Самые крутые многопользовательские игры часто ориентированы на прогнозирование и использование внутренней информации вроде этой. Чтение мыслей или контроль ума противника — одна из самых приятных форм победы.

Информация может
скрываться в скорости.

Мозгу требуется время, чтобы воспринимать, обрабатывать и использовать информацию. Это означает, что информация, поступившая за последние доли секунды, тщательно скрыта от процесса принятия решений. Мы принимаем решения не на основе информации, которую воспринимаем сейчас, а на основе информации, которую мы осознали десятую долю секунды назад. Именно так мы играем в «камень, ножницы, бумага». Поскольку оба игрока показывают фигуры в одно и то же время, ни один из них не успевает оценить ход другого, чтобы принять собственное решение, даже если один покажет на несколько миллисекунд раньше.

Информационный баланс. Кейс: покер

История покера — это прекрасный пример дизайна с различными типами информационного баланса.

Первая версия покера была, по сути, игрой на удачу. Четыре игрока делали ставку, а затем каждый брал по пять карт из колоды, состоящей из 20 карт. Каждый игрок делал ставку один раз и раскрывал карты. Если вам попадались хорошие карты, вы могли делать высокую ставку. Если этого не происходило, вы либо сбрасывали карты, либо добивались своего блефуя.

В старом покере информации в игре было недостаточно. В ней не хватало сложных решений, потому что не хватало данных для размышлений. Каждый игрок знал только свои карты и ставки. При том что в игре был всего один круг торговли, информации было недостаточно для формирования последовательных стра-

тегий. Так что у игры был низкий потолок мастерства, потому что она почти полностью велась наугад.

Первые игроки в покер хотели, чтобы у игры был более высокий порог мастерства и чтобы можно было обыграть случайных игроков, не прибегая к рискованным методам. Таким образом, они изменили геймдизайн.

Во-первых, колоду из 20 карт заменили на колоду из 52 карт. Тридцать две дополнительные карты значительно увеличили количество возможных комбинаций. Но если бы больше ничего не изменилось, в игре осталось бы еще меньше информации, поскольку количество возможных комбинаций увеличилось.

Более важным изменением стало введение нескольких комбинаций обмена и кругов торговли. Вместо того чтобы делать одну почти слепую ставку, в игре теперь появилось несколько кругов. В каждый свой ход игрок мог сбросить ненужные ему карты и взять вместо них новые. Затем игрок мог либо скинуть карты, либо сделать ставку на любую сумму.

Эта многораундовая структура добавила в игру огромное количество информации. Теперь игроки могли отвечать на комбинации обмена и ставок во время круга торговли. Наблюдая за чьими-то комбинациями обмена и ставками, они могли выяснить, какие карты есть у другого игрока. Сейчас мы знаем этот вид покера как *дро-покер*.

Но даже дро-покер по-прежнему сильно зависит от удачи. В отличие от старого покера игра ведется уже не совсем вслепую, но все же заставляет много гадать и надеяться. Так что она продолжала развиваться.

Во время Гражданской войны кто-то придумал стад-покер, названный так в честь лошадей, которые тащили артиллерию. В стад-покере комбинации с обменом запрещены. Вместо этого игроки сдавали карты по одной, чередуя с кругами торговли, а три карты сдавались открытыми, чтобы все могли их видеть.

Информационный баланс стада значительно отличается от информационного баланса дро-покера. В дро трудно сказать, кому какая карта досталась, если наблюдать за тем, как игрок

берет карты. Игрок, который берет одну карту, может блефовать, пытаться собрать пару, тройку, фулл хаус или флеш. Но в стаде игроки должны постоянно наблюдать за комбинациями друг друга. Хороший игрок может угадать две скрытые карты из трех, а также изменения в чей-то схеме ставок. Вы можете увидеть, как кто-то с четырьмя, пятью и шестью картами начнет делать высокие ставки после того, как вытащит определенную карту, и предположить, что у него стрит.

Но информационный баланс игры зашел слишком далеко, и появилась избыточность информации. В большинстве комбинаций стад-покера совершенно очевидно, кому достались лучшие карты. Иногда выигрышная комбинация четко видна на трех открытых картах.

Например, не стоит ходить против игрока с тройкой, если у вас только пара, так как вы гарантированно проиграете. Решение, которое могло стать интересным, было стерто избыточностью информации, благодаря которой вы получили ответ, не прикладывая никаких усилий. Игра продолжала развиваться.

В середине XX века кто-то изобрел карты, используемые всеми игроками за игровым столом. В этих играх, как и в стад-покере, карты сбрасываются по очереди между кругами торговли. Разница в том, что некоторые открытые карты распределяются между игроками, а не достаются отдельными игрокам. Техасский холдем — самый популярный пример такой игры.

Механика карт, используемых всеми игроками за игровым столом, наконец достигла идеального информационного баланса. Игра редко генерирует очевидные или непонятные выборы. Существует лишь небольшое количество скрытой информации (две скрытые карты в техасском холдеме), которые нужно контролировать, но эта информация тесно связана с общими картами и ставками игроков.

Открытые карты не означают, что вы сможете определить победителей, потому что карты раздают между всеми игроками — если в наличии три комбинации из пары тузов и любой другой пары, у каждого игрока присутствует тройка карт одинакового достоинства. Вопрос только в том, кому досталась другая пара,

четвертая комбинация или старшая карта? Этот совершенный информационный баланс означает, что почти каждая комбинация генерирует удивительные решения, давая достаточно информации и не делая ответ очевидным.

Основная механика покера практически не изменилась. Вы получаете свою комбинацию и делаете чек, ставку или фолд. Но игра несколько раз трансформировалась, просто изменяя свою структуру и способы узнавания информации. Достигнув идеального информационного баланса, она выжимает из каждого решения максимум смысла и эмоций. А чтобы найти эту точку баланса, потребовалось всего два столетия.

Проблематичный информационный источник

Чтобы проанализировать информационный баланс, мы должны точно знать, какой именно информацией располагают игроки при принятии решения. Но, как правило, неясно, что игроки будут знать или как они это узнают, так как информация может быть раскрыта или скрыта самым неожиданным образом.

Неопределенность сюжета

Сюжет помогает поддерживать связь с игроками, оборачивая механику в привычные образы и звуки. Но в этом процессе остается опасная неопределенность потому, что ни одна игровая механика полностью не имитирует все, что связано с ее сюжетной оберткой.

Представьте, что вы играете в игру и видите на столе жареную индейку. Что с ней можно сделать? Взять, положить в рюкзак и носить с собой? Съесть, и если да, что это значит в игре? Продать? Заморозить? Бросить животным в качестве приманки, и, возможно, благодаря этому вы сможете проскользнуть мимо них? А может, в ней можно спрятать пистолет? В реальной жизни с жареной индейкой можно сделать что-нибудь из вышеперечисленного и даже больше. И разные игры позволяют каждое из этих действий. Но ни одна игра не разрешает их все.

Проблема в том, что просто посмотрев на индейку, игрок не может узнать, какая из этих возможностей реальна

Информация из сюжета часто неоднозначна, потому что игрок не может знать, какие аспекты сюжета являются реальной игровой механикой, а какие нет.

в данной игре. Иначе говоря, он не может спрогнозировать взаимодействие с индейкой, потому что он не понимает систему механики, которую та выражает. Изображение индейки — просто символ неизвестной игровой механики.

Это широко распространенная проблема в классических приключенческих играх. Например, в одной приключенческой игре начала 1990-х был квест, в котором игрок должен был пробраться мимо охранника. В инвентаре игрока было несколько палок, веревка и грязь. Сюжет подразумевает почти бесконечное количество способов использования этих инструментов, чтобы проскользнуть мимо кого-то. Можно бросить грязь в глаза и пробежать мимо, пока он ничего не видит. Или бросить палку рядом с охранником и проскользнуть мимо, пока он отвлекся на шум. Или создать оружие и убить его.

По задумке, игрок должен был сделать из палки, веревки и грязи маску, чтобы охранник не узнал его (серьезно). Хотя это возможно в сюжетной обертке игры, этот способ ничем не выделяется среди тысяч других столь же вероятных способов, которые вы могли бы использовать, чтобы обойти охрану. В результате игрок приближается к прохождению квеста, но при этом ему не хватает информации. Поскольку он не понимает механики, необходимой для принятия решения, ему остается только перепробовать все доступные ему взаимодействия. Игра превращается в произвольное механическое упражнение.

Эти проблемы встречаются даже в хороших играх. В недавней, получившей признание критиков приключенческой игре есть миссия, в которой игроку нужно разбудить пилота, который катапультировался, повис на парашюте на дереве и уснул. Проблема в том, что пилот в наушниках, поэтому ничего не слышит. Будь на дворе зима, некоторые решения становятся очевидными. Слепите снежок и бросьте пилоту в лицо. Потыкайте в него длинной палкой. Подождите, пока он проснется сам, потрясите дерево. Все это сработало в мыслях, но не в игре. На самом деле решение головоломки заключается в том, что нужно забраться в разбившийся самолет и крутить рычаги радиоуправления до тех пор, пока не отобразится частота, затем отправиться на ближайшую радиостанцию и передать

сообщение через наушники, которое разбудит пилота. Опять же, в этом и состоит сюжетный смысл, так же как и в тысячах других, столь же вероятных решений.

Дизайн квеста — самый очевидный, но он не единственный пример неопределенности вымышленных механик. Эта проблема может поразить все, что сообщается с помощью задумки. Я смогу спрятаться от пуль за этой деревянной доской? Убьет ли заклинание огненного шара человека за один выстрел? Могут ли я разбить это стекло? Могут ли я открыть эту дверь?

Решение состоит в том, чтобы убрать квест из вымысла. Вместо этого создайте головоломки из понятной механики.

Игрок должен знать всю механику, связанную с задачей. Решение состоит в том, чтобы выяснить, как ее использовать.

Например, в играх Super Mario существуют задания, связанные с прыжками. Это платформы и опасности, через которые Марио должен перепрыгнуть, чтобы достичь какой-то цели. Но для этого игрок должен найти хороший путь. Поиск пути — интересная задача, так как нужно тщательно продумать прыжки. При этом у игрока не возникнет двусмысленности и неуверенности, поскольку прыжки Марио предсказуемы и понятны. Шестеренки в мозгу игрока вращаются, и его ум прокручивает всевозможные варианты. И когда он видит решение, то сразу узнает его и чувствует озарение.

Хорошие игровые решения, включая хорошие задания, всегда основаны на неочевидном использовании механики, которая работает очевидным образом.

Исключением из этого правила является та ситуация, когда само решение оправдано только сюжетом или задумкой. Игра может предложить моральный выбор или выбор персонажа, который находится за пределами игровой механики и затрагивает только вымышленные элементы. В этом случае для принятия решения вымышленная информация может использоваться отдельно, потому что решение само по себе не имеет механического значения — оно работает на уровне вымысла. Но в тот момент, когда выбор начинает влиять на игровую механику, изменяя статистику, уровни пути или улучшая инструменты, он становится другим типом решения, и для этого необходима однозначная информация о механике.

Метаигровая информация

У игроков больше информации, чем дает сама игра. Они получают эти знания, или метаигровую информацию, за пределами самой игры.

МЕТАИГРОВАЯ
ИНФОРМАЦИЯ —
это информация,
которую игрок получа-
ет из реального мира
вне игры.

Игроки много знают об игре еще до того, как начнут играть. Они могут угадать ее длину по жанровым конвенциям. Они знают уровень ее сложности по рассказам друзей. Они могут предсказывать поведение NPC (персонаж, не управляемый игроком) и сюжетные повороты, наблюдая за клише или зная привычный сценарий работы студии, выпустившей игру. Они знают предельные возможности компьютерных технологий, следовательно, они знают, что игра никогда не выведет на экран 10 000 персонажей сразу. Они видели трейлеры и оформление коробки, поэтому могли узнать основных персонажей, темы и сюжет. Ничего из этого нет внутри игровой механики или сюжета, но у игроков все равно есть эта информация и она все равно влияет на процесс принятия ими решений во время игры.

Метаигровая информация может исказить опыт, предоставив игрокам информацию, которой, по мнению дизайнера, у них нет. Часто это вызывает избыточность информации.

Например, во многих играх игроки собирают и используют такие ресурсы, как боеприпасы и здоровье. Часто по задумке эти ресурсы должны быть очень ограничены: персонаж игрока заперт в замке, кишащем зомби, или на инопланетном космическом корабле. Можно предположить, что в таком месте игроку придется пробыть достаточно долго, прежде чем он сможет найти аптечку или боеприпасы. Таким образом, задумка говорит нам, что игрок будет мучиться от истощения. Это грозит ему несправедливой, страшной смертью, которой он боится.

Но действительно ли игра настолько несправедлива, как предполагает задумка? Обычно нет. И игрок подсознательно это понимает. Игрок знает, что ему никогда не придется оставаться без ресурсов в течение слишком долгого времени, потому что это нарушит конвенцию и будет нечестным. Он знает, что игра

не будет такой подлой, потому что она была создана для развлечения.

Эта метаигровая информация искажает ход мыслей об управлении ресурсами. Теперь вместо того, чтобы думать о том, что определяет игровая задумка, он думает об особенностях дизайна и жанровых конвенциях. Он может догадаться, когда появится следующий ресурс, потому что он знает, что справедливо, а что нет, а также он знает, что игра будет честной, потому что это игра. Это уменьшает напряжение и ослабляет погружение.

Проблемы метаигровой информации трудно увидеть, потому что в основном их последствия находятся в голове игрока. Как и большинство проблем избыточности информации, они не разрушают игру полностью, они ослабляют ее, искажая внутренние мыслительные процессы игроков.

У разных игроков разная метаигровая информация. У игрового журналиста целые тонны метаигровой информации. У маленького ребенка ее гораздо меньше. Большинство игроков находятся где-то посередине. Но даже самый неподготовленный игрок все равно будет немного знать. Он знает, что у игры существует определенный интерфейс на определенном типе платформы. Она ограничена экраном или игровой доской. Ее придумали люди определенной культуры, и она предназначена для развлекательных целей.

Существуют два основных способа обработки метаигровой информации. Один из них можно назвать стандартным, другой — немного безумным.

Стандартный способ состоит в том, чтобы признать, что системы дизайна все еще создают желаемый опыт, даже учитывая метаигровую информацию. Это означает, что не следует пытаться заставить игрока бояться того, что, как ему известно, не произойдет, или заставлять его надеяться на невозможное. Пугайте игрока оправданными честными игровыми методами. Дайте ему четко сбалансированные достижимые цели. Задумка должна работать в рамках метаигровой информации, которая есть у игрока. Рассматривайте ее как еще одно ограничение дизайна.

Безумный метод — провоцируйте игрока. Покажите игроку, что вы не обычный дизайнер, что правила на вас не распространяются. Будьте нечестным. Будьте непредсказуемым. Нарушайте установленные технические ограничения.

Безумный метод опасен и сложен. Игровые конвенции существуют не просто так. В большинстве случаев нарушать их — не очень хорошая идея. В то же время существует опыт, который может возникнуть из-за необоснованных, несправедливых угроз для персонажа игрока и задач, которые не могут появиться любым другим способом. Например, *System Shock 2* вселяет ужас, потому что она не лжет игроку. Действие игры разворачивается на огромном глухом космическом корабле, кишашем монстрами. Задумка подразумевает, что ресурсы должны быть чрезвычайно ограниченны, врагов должно быть много, а у игрока не должно быть больших шансов на выживание. И тут ничего не поделаешь. Игра на самом деле несправедлива. Медлительный неосторожный игрок не выживет. Нет никаких защитных механизмов. В некотором смысле это звучит как плохой геймдизайн. Но это также удивительно иммерсивный геймдизайн, потому что механика отражает задумку. Игрок чувствует смертельный голод и безысходность, потому что он действительно голоден и попал в ловушку.

Решения и поток

Цель потока — привлечь внимание игрока к игре. Если мы не в потоке, кусочки реального мира постоянно вторгаются в наше сознание. Мы чувствуем, как сильно пальцы жмут на кнопку. Мы слышим, как тикают часы или на улице лают собаки. Мы делаем паузу, чтобы сходить в туалет, попить или поговорить с другом. Все это мешает опыту. Это постоянно мешает опыту, который пытается создать игра.

Но если мы в потоке, реальный мир исчезает. Сознание и игра сливаются в непрерывном танце экшена, реакции, решения и результата.

Отсюда следует, что поток — это основа хорошего игрового опыта. Проблемы, возникающие в других сферах дизайна, часто обу-

словлены не чем иным, как разрывом потока. Без потока игроки чувствуют и жалуются на каждый раздражитель. Если они будут в потоке, то примут странные задумки, грубую графику или непонятные взаимодействия. Самые серьезные ошибки дизайна — те, которые нарушают поток, потому что они ослабляют связь между сознанием игрока и игрой, мешая любой другой части опыта.

Люди говорят об играх как о формах эскапизма, и они правы. Но мы часто предполагаем, что эскапизм — это не что иное, как фантазия, что мы хотим сбежать от своей скучной жизни, притворяясь колдуном или гонщиком. Но разве кому-нибудь хочется быть маленьким толстым итальянцем-сантехником или прыгающим куском мяса? Нет, но это не мешает Super Mario Bros. и Super Meat Boy генерировать сильнейший опыт эскапизма. Потому что эта форма эскапизма связана с потоком, а не с выдумкой. И ею движет не задумка, а механика.

Ранее мы рассмотрели, как баланс возможности и сложной задачи образуют поток. Но это была основная, упрощенная концепция потока. Недостаточно просто сказать геймдизайнеру, что он должен создать баланс между возможностью и сложной задачей. Настоящий поток нельзя размазать большой кистью по всему опыту, просто задав его и определив количество. Это скорее нежный танец мимолетного принятия решения.

Представьте, что разум игрока — это маленькая чашка с отверстием внизу. Игра заполняет чашку водой. Поддержание потока означает, что вода всегда сливается через отверстие и чашка никогда не переполняется. Иными словами, нужно постоянно подливать воду, но ее не должно быть слишком много.

Эта вода представляет собой решения. Как только они появляются в разуме, мозг начинает над ними работать. Большинство решений длится всего несколько моментов. Чтобы поддерживать поток разума, мы должны продолжать принимать решения с правильной скоростью. Слишком мало решений, и разум опустошается за несколько секунд и становится скучно. Слишком много — и разум переполнен, поток нарушается. Чтобы получить баланс, мы должны точно определить размер и сроки принятия решений.

Объем решения

Объем решения — это «размер» решения, которое необходимо обдумать. В метафоре о воде и чашке — это количество воды, которое решение помещает в чашку. Сложные решения, включающие множество вариантов, имеют большой объем, потому что они могут занимать разум в течение длительного времени. Простые решения с одним или двумя вариантами имеют небольшой объем.

ОБЪЕМ РЕШЕНИЯ — это количество затраченного мыслительного процесса, необходимого для принятия решения.

Решения можно разделить на пять категорий. От самых маленьких до самых больших — мгновенные решения, импульсивные решения, тактические решения, глубокие решения и невероятные решения.

Мгновенные решения — это те решения, ответы на которые настолько очевидны, что они вообще перестают быть решениями. Например, если вы готовите сухой завтрак, то наливаете в него молоко. Решение налить молоко в кашу на самом деле не является решением, потому что оно только поверхностно затрагивает разум. Это настолько очевидно, что вы можете сделать машинально. Когда вы были ребенком, это могло быть решением. Но сейчас вы не задумываетесь над этими действиями. Отсутствие решений может привести к тому, что пальцы игрока будут заняты, но они не будут влиять на поток, так как разум ничем не занят.

Импульсивные решения — это самые маленькие значимые решения. На их обдумывание уходит меньше секунды, и для этого требуется лишь простое сознательное рассуждение. Ударить кулаком или ногой? Перезагрузить сейчас или подождать еще одну секунду? Посмотреть налево или направо? Прыгать или увернуться? Импульсивные решения являются общей основой для игрового потока, потому что их легко генерировать. Нам не нужно настраивать интересные, элегантные системы с частично предсказуемой причиной и следствием и заставлять игрока продумывать их. Нам нужно лишь сделать простой выбор в ограниченное время. Вот почему экшн-игры часто называют импульсивными играми — они почти полностью основаны на импульсивных решениях.

Тактические решения — это следующий вид решений. Они требуют от одной до пяти секунд работы мысли и заметно привлекают сознание игрока. Какое оборудование я должен купить? Какие объекты я должен построить? Такие решения включают больше информации, чем импульсивные решения. Игрок может учитывать позиции нескольких персонажей, их способности, условия и инструменты. Он может даже вспомнить исход подобной ситуации, которая случилась ранее.

Глубокие решения — это самые большие решения, на принятие которых уходит более 10 секунд. Такие решения используют так много информации, что выходят за пределы самой игры и опираются на более обширные знания игрока об эмоциях, культуре и человечности. Они подталкивают игрока заглянуть внутрь себя и хорошенько покопаться в своих мыслях, воспоминаниях и эмоциях в поиске каждого единичного фрагмента информации, которую можно найти. Когда гроссмейстер по шахматам Гарри Каспаров смотрит на шахматную доску четверть часа, он принимает глубокое решение. Он вкладывает все свои знания, полученные в течение многих лет обучения. Его разум погружается в знания привычек противника, сотен возможных будущих игровых состояний, недавно выученных тактик и стратегий, ранее составленных планов, проверок его собственного мнения в данный момент. Сложные решения возникают только из самых элегантных, тонких, захватывающих игровых систем. В большинстве игр таких никогда не бывает.

Невероятные решения находятся за пределами способности понимания игрока. Если решение требует тонкого понимания или большого объема знаний, которых у игрока нет, или потенциальные результаты слишком многочисленны, чтобы понять его, решение становится помехой. Игрок должен иметь возможность обдумать свое решение и выбрать один из путей с логичным результатом. Если у него нет такой возможности, решение невозможно и не способствует потоку, потому что игрок будет делать выбор наугад.

Игры могут представлять решения любого объема или любой комбинации. Эта смесь характеризует темп и чувство игры. Игра, наполненная огромным количеством импульсивных решений, —

это бешеная игра в жанре экшн. Монотонный марш глубоких выборов — медленная стратегическая игра. Игра с одним практически невыполнимым заумным заданием — головоломка.

Навык игрока меняет эффективный объем решений.

Сложное решение для новичка — это мгновенное решение для эксперта. По мере того как игрок учится, ранее невозможное решение может стать сложным, затем тактическим, а затем импульсивным, прежде чем окончательно превратиться в мгновенное решение.

Так мы можем размышлять о диапазоне мастерства в еще одном ключе: диапазон мастерства в игре — это диапазон уровней мастерства, на которых часто представлены импульсивные, тактические или сложные решения. Это означает, что потолок мастерства определяется объемом крупнейших решений. Когда знаний у игрока становится достаточно для того, чтобы решения с наибольшим объемом стали мгновенными, — он прошел потолок мастерства. Вода перестает наливаться в чашку, поэтому поток становится невозможным и игрок уходит.

Два характерных способа, которые могут прервать поток, — пустая чашка и переполненная чашка. Давайте рассмотрим каждый из них.

Как избежать прерывания потока

Поток очень недолговечен. Самая важная часть поддержания потока — это генерирование потока решений так, чтобы чашка оставалась полной, но никогда не переполнялась. Даже самые незначительные нарушения в темпах принятия решений могут ухудшить поток. Игрокам может стать неинтересно, даже если пауза составит одну секунду. При достаточно большом количестве таких пауз равномерный, плавный опыт превращается в скачкообразное упражнение. Я называю это прерыванием потока.

ПРЕРЫВАНИЕ ПОТОКА — это определенный период, час или секунда, когда ум игрока ничем не занят.

Прерывание потока может стать опытом множеством способов.

Например, дизайнеры могут решить добавить ограничение, чтобы сбалансировать уровень мощности оружия. Посох волшебника в фэнтезийной игре может быть сконфигурирован так, что метать удары молнии им можно будет только раз в две секунды. Но несмотря на то, что этот

прием может уравновесить посох, он в то же время прерывает поток, если между выстрелами игроку больше нечего делать.

В других случаях прерывание потока может появиться из-за сюжетных или эстетических элементов дизайна. Переходы по меню, анимация движения и диалог могут на доли секунды блокировать способность игрока действовать, создавая тем самым паузу в потоке решений.

Это классический конфликт между механикой и задумкой, так как он обусловлен наличием радующей глаз графики, которая делает игру красивой и усиливает сюжетный слой и в то же время ослабляет механическое взаимодействие, которое заставляет принимать решения. Очевидно, что лучшим решением будет найти дизайн с приятным интерфейсом, который будет поддерживать поток. Но если нужно сделать выбор, то нужно учесть тот факт, что большинство игр получают больше пользы от устойчивого потока, чем от крутых переходов меню или анимации. Графика радует глаз первое время, с первого раза, а поток — всегда.

Если без паузы обойтись нельзя, мы должны найти способ заполнить ее решениями. Иногда это так же просто, как сделать доступными другие способности. Посох волшебника может перезагружаться, но игрок все еще может двигаться и атаковать ножом.

В других случаях, чтобы справиться с необычными ситуациями, нам нужны более экзотические модели. Например, многие игры разных жанров включают в себя оглушающую атаку. Фэнтези-игры содержат ошеломляющие заклинания, в военных играх доступны несмертельные светошумовые гранаты, Человек-паук может набросить паутину на своего врага, а боксер может ударить противника по голове, чтобы оглушить его. Это хороший дизайн, потому что он элегантно сочетается с другими последующими атаками и меняет темп боя.

Остается одно «но»: поток для жертвы прерывается.

Самая простая версия оглушения — заморозка жертвы. Но несмотря на то, что это вроде как хорошая и логичная задумка, такое решение жестко прерывает поток. Жертва не может принимать никаких решений во время ошеломляющих заклинаний, так как не может действовать. Как решить эту проблему?

В различных играх применялись различные методы, чтобы сохранить суть оглушения без отрицательного влияния на саму игру. Например, некоторые игры заставляют жертву оцепенеть, вмешиваясь в ее управление и в то же время позволяя ей действовать. Оглушающая граната в *Modern Warfare 2* замедляет скорость поворота игрока и влияет на его обзор. Враг хуже целится, и его можно легко атаковать, но он все еще может продолжать взаимодействовать и принимать решения. Старые игры-аркады в жанре фэнтези моделировали оглушающие заклинания путем зашифровки всех элементов управления, поэтому при нажатии вверх персонаж перемещался вниз и наоборот. Жертва попала под удар, но она все еще может действовать. При броске световой гранаты в *Counter-Strike* экран монитора жертвы становится на несколько секунд белым, но при этом все элементы управления остаются рабочими. Иными словами, ослепленный игрок все еще может играть при ограниченных возможностях, запоминая свое положение и прислушиваясь к шагам. В каждом случае оглушение конструктивно, а поток сохраняется.

Избегайте избыточного потока

Избыточный поток более очевиден, чем прерывание потока. Если при прерывании потока игроку скучно, избыточный поток проявляется в виде жалующихся игроков, находящихся в напряжении. Поскольку это так очевидно, дизайнеры склонны решать эту проблему естественным способом, ослабляя давление на игрока. Поэтому я не буду подробно писать об избыточных потоках. Следите за ними и при необходимости снижайте степень их влияния на решения.

ИЗБЫТОЧНЫЙ
ПОТОК — это момент,
когда игрок перегру-
жен решениями.

Избыточный поток в готовых играх обычно возникает у игроков с минимальными навыками. Дизайнеру, возможно, удалось получить хорошую скорость потока среди тех игроков, которые тестировали игру, однако, имея потолок умений ниже требуемого, более слабые игроки получили ужасный опыт. В какой-то степени это неизбежно. Почти всегда есть игроки, которым настолько не хватает навыков, что они не могут играть. Вот почему следует сознательно выбирать минимальный уровень навыков и плани-

ровать игру, основываясь на нем, вместо того чтобы пытаться включить все возможные варианты.

Принятие решений в пошаговых играх

До сих пор я описывал поток с точки зрения системы реального времени. Это не относится к пошаговым играм, потому что они позволяют игрокам принимать решения в своем собственном ритме. Столкнувшись с объемным решением, ход игроков просто становится дольше. Если решение небольшое, ход, соответственно, короче. Однако это не значит, что геймдизайнеры пошаговых игр могут игнорировать границы решения, это допустимо только если результат будет другим. Фактически две классические широко распространенные проблемы в геймдизайне пошаговых игр можно объяснить плохо организованными решениями.

Микроменеджмент происходит там, где слишком много мелких решений.

Игроки вынуждены бесконечно делать десятки или сотни почти бессмысленных ходов, чтобы хоть как-то продвинуться вперед. В итоге все заканчивается прерыванием потока, потому что перемещение фишек отнимает больше времени, чем принятие решения. Игра превращается в скучное перемещение фишек вместо принятия решений и умственного напряжения.

Аналитический паралич случается при очень объемных решениях. Игроки сидят и слишком долго думают. В одних случаях, как, например, в шахматах, это не обязательно является проблемой. В других же, например в настольных играх, предполагающих много игроков, или в более динамичных играх, аналитический паралич — убийца опыта.

Разнообразие решений

Чтобы игроку было интересно, мы должны «начинить» игру решениями различного уровня сложности и объема. Не следует «кормить» игрока тактическими решениями каждые четыре секунды в течение часа. Предложите ему сжатую последовательность импульсивных решений, за которыми последует вынос мозга, дающий игроку столько времени на размышление, сколько

ему нужно, а затем добавьте еще несколько сложных решений, на которые потребуется не более пяти секунд, и так далее.

Решения могут также различаться по тональности. У решений аналогичного масштаба может быть разный качественный оттенок, если их смысловое наполнение отличается. Например, в стратегической игре тактический выбор перемещения армии может быть примерно таким же, как и решение о том, где построить новую производственную структуру. Повторять одно из этих решений бесконечно, однако, менее интересно, чем чередовать их.

Классическая кривая ритма, которую я описывал в первой части, является одним из ориентиров. Начинайте медленно, проведите игрока через несколько циклов завязки, чередуя ее с моментами спокойствия, прежде чем наконец достигнуть кульминации и перейти к развязке.



Но стандартная кривая интенсивности — не единственный способ изменить темп принятия решения. Богатые игровые системы часто слишком непредсказуемы, чтобы постоянно следовать стандартной кривой. Время от времени они достигают пика или содержат фрагменты, где действие происходит слишком медленно. Поскольку мы не можем заранее определить решения (по той причине, что механика должна генерировать их мгновенно, чтобы они были обусловленными системой и, следовательно, предсказуемыми), каждый раз получать эту традиционную формулу стимуляции непросто. Впрочем, проблема не в этом.

Можно проанализировать темп принятия решений на примере дизайна, который не был реализован. Просто представьте, что играете в игру, как это делал бы игрок. Вы соревнуетесь с игровыми системами, задействуя свое мышление. Подумайте о том, что происходит, что вы знаете, чего вы не знаете о представлен-

ных решениях и процессе мышления, необходимом для принятия этих решений.

Но здесь есть хитрость. Разум будет естественным образом переходить к самым интересным частям любой воображаемой или запоминающейся истории. Этот процесс скроет прерывание потока. Говоря о темпе принятия решений, вы должны прокрутить каждую секунду опыта в уме в реальном времени, ничего при этом не пропуская. Это может оказаться сложным — весьма неприятно сидеть и представлять каждую скучную анимацию, загружающийся экран и бессмысленное нажатие кнопки. Странно тратить пять минут на размышления о пятиминутном опыте. Но если вы хотите узнать что-нибудь полезное о темпе, важно сэкономить время. Этот процесс не так приятен, как тестирование игры, но это лучше, чем ничего, и значительно проще.

Единственное избыточное правило потока — темп не должен меняться. Он не должен утомлять игрока ни длинными и медленными, ни длинными и быстрыми периодами игры.

Принятие решений. Кейс Counter-Strike

Впервые я играл в Counter-Strike в 1999 году, когда это была просто модификация Half-Life, сделанная двумя людьми. Геймплей был простой и понятный: войска солдат специального назначения сражалась с бандами террористов в балаклавах. После устранения одной группы или после того, как она выполнила свою миссию, все вновь появились в другом раунде. За убийство врагов или выполнение задач игроки получали деньги, которые можно было потратить на покупку лучшего оружия и амуниции в последующих раундах.

Это была хорошая игра. Стандартным примером хороших игр является рост популярности, который постепенно угасает в течение нескольких лет. Но с Counter-Strike этого не произошло.

Я перестал играть в 2001 году, но игровое сообщество никуда не исчезло. Технологически продвинутые конкуренты, такие как Unreal Tournament 2003, появлялись и исчезали, но люди продолжали играть в Counter-Strike. Вышел Half-Life 2, но люди все равно продолжали играть в Counter-Strike.

В игре поменялась графика, шли годы. Мега-хит Call of Duty 4 ворвался на рынок в 2007 году, но, как вы можете догадаться, люди продолжали играть в Counter-Strike. На момент написания этой статьи, спустя 13 лет после первоначального выпуска игры, Counter-Strike по-прежнему остается самой популярной игрой в широко известном игровом сервисе Steam для ПК.

Почему же так случилось? Безусловно, проблема не в задумке игры. Существует много военных игр, некоторые из которых успешны, некоторые — нет. Counter-Strike стала успешной игрой благодаря механике, а также балансу игры, темпу, мастерству и решениям. Давайте разберем эти решения на стандартном примере из игры Counter-Strike.

Counter-Strike генерирует следующие решения.

Перезарядиться сейчас или позже? Что, если пока я перезаряжаюсь, за углом появится враг? А что, если у меня закончатся патроны во время боя?

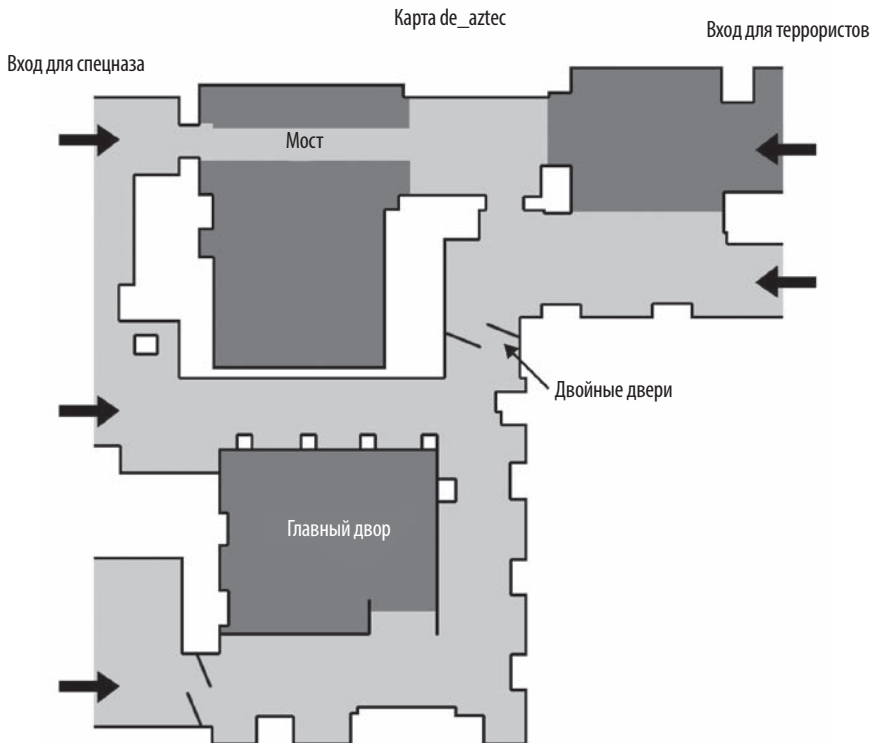
Куда идти — вперед, назад, в сторону или вообще оставаться на месте? Что будет, если меня застрелят, пока я стою между укрытиями? Что будет, если я проиграю, потому что не ставлю перед собой цель? Что будет, если меня убьют, потому что я слишком далеко от моей команды? Если убьют союзников, прикрывающих другой вход, а мне выстрелят в спину?

Покупать ли сейчас оружие? Что будет, если у меня закончатся деньги, а они мне понадобятся? Если я умру в этом раунде, потому что у меня нет оружия?

Как и во всех играх, поддерживающих поток и требующих навыков, эти решения появляются постоянно и с большой скоростью. Двухминутный раунд может включать сто таких решений. Частота настолько высока, что мы не способны проанализировать хотя бы минуту этой игры. Чтобы посмотреть, как работает Counter-Strike, нужно замедлить движение и понаблюдать за мыслями игрока каждую секунду. В этом случае я рассмотрю первые 20 секунд типичного раунда.

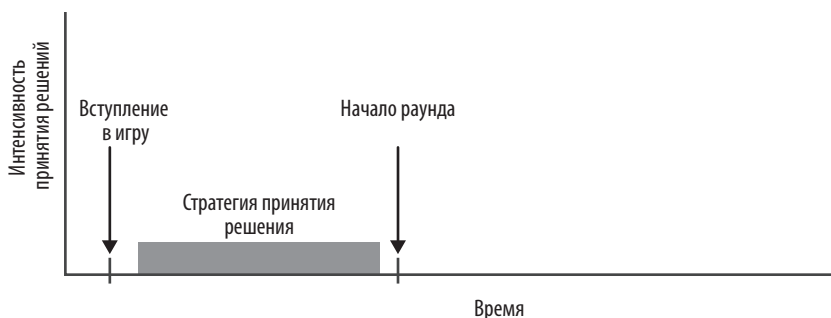
Игрок Боб вступает в игру. Боб не сразу начинает играть; он должен сначала посмотреть текущий раунд. Так он может собрать

основную информацию: текущая карта de_aztec, Боб в террористической группе, в каждой команде по пять игроков. Табло показывает, что его команда проигрывает. Наблюдая за завершением раунда, он видит, что в команде противника снайпер-эксперт по имени Алиса, которая действует в районе Главного двора с западной стороны.



На этом этапе Боб может только наблюдать за игрой, но это не значит, что он не в потоке. Его руки неподвижны, он как гроссмейстер, сгорбившийся за доской, а его разум при этом активно работает. Он продумывает стратегию на следующий раунд. В данном случае это сложный тип решения. Он отмечает, что его команда неспособна вести борьбу со снайперами противника, что он не может приобрести световую гранату или мощное оружие, отмечает вероятное присутствие Алисы в Главном дворе, собственный особенно высокий уровень навыков ближнего боя и многие другие факторы. К началу раунда он уже принял решение.

На графике его темп до настоящего момента выглядит так:



На графике мы видим, что стратегическое решение охватывает большую область. Но оно растянуто на длительное время, поэтому темп принятия решения медленный. У него было много времени, но решение было настолько масштабным, что заполнило интервал времени, поэтому поток не прервался.

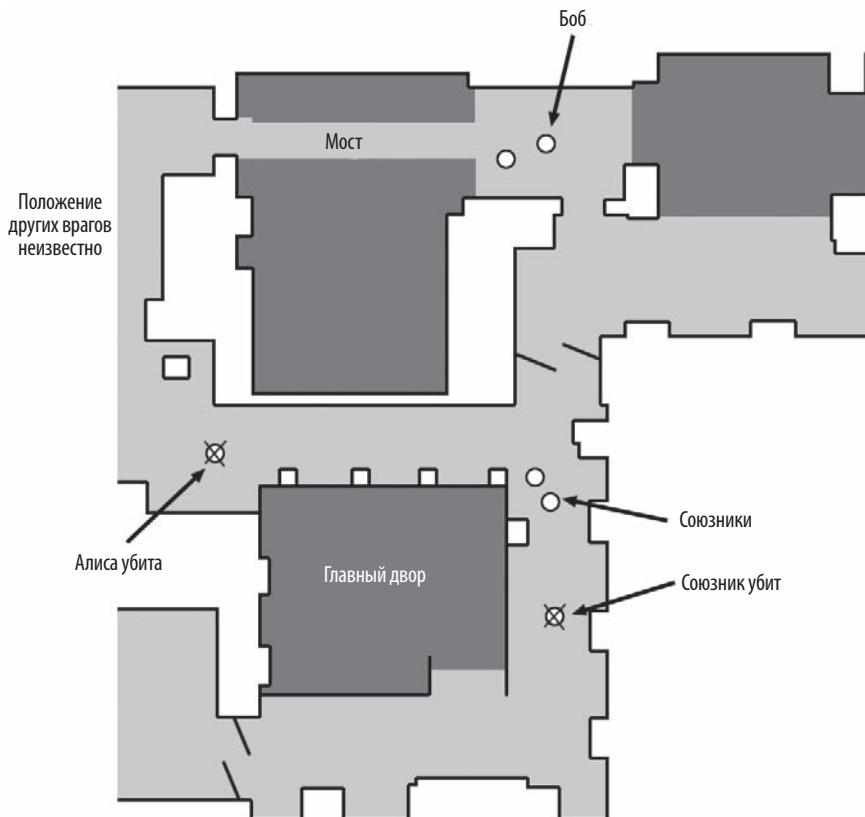
Начинается следующий раунд. Боб с командой появляется на восточной стороне карты. Он может купить только дешевый пистолет-пулемет MP5 с уменьшенной дальностью стрельбы, поэтому он знает, что находится в невыгодной позиции на открытых площадках. Как он и решил, он идет северной дорогой через мост, потому что он не хочет столкнуться с Алисой, которая, как он предполагает, нападет на Главный двор. Боб хочет обойти ее и использовать свои навыки ближнего боя, чтобы уничтожить ее. «Если я смогу пройти через мост живым, — думает он, — я смогу ударить Алису с близкого расстояния. У меня хороший ближний бой, и ее снайперская винтовка будет бесполезна, если я буду прямо перед ее носом».

Рассмотрим информацию, доступную Бобу на данный момент. Он еще не знает, как движется другая команда, так как она находится на другом конце уровня, за стенами. «Я предполагаю, что они разделились и идут по всем своим путям, — думает он. — Вряд ли они все пойдут одним и тем же маршрутом». Он также не уверен в намерениях своих союзников. Он предпочитает играть с друзьями, так как они могут разрабатывать планы и понимать намерения друг друга. Но сегодня он играет на публичном сервере, известном своими опытными игроками. Все, что он знает

о своих товарищах по команде, — это тот факт, что у них хорошие навыки. Благодаря этой метаигровой информации он может предположить, что они выберут оптимальные пути и выстрелят в нужный момент. Но это все, что он может предположить. Боб видит нескольких членов команды с винтовками для дальней стрельбы и делает вывод о том, что они попытаются агрессивно наступать через мост или Главный двор, а не отступить назад и попытаться устроить противнику засаду.

Трое союзников Боба проходят через Двустворчатые двери в Главный двор, а один идет с Бобом. Боб слышит выстрел. Глядя на счетчик убитых, Боб получает новую информацию: в Главном дворе Алиса убила одного из членов его команды. Это не влияет на план Боба; это то, что он ожидал. Поэтому он продолжает свой путь.

Однако приблизившись к мосту, на счетчике убитых Боб видит, что Алиса убила. Теперь карта в представлении Боба выглядит так:



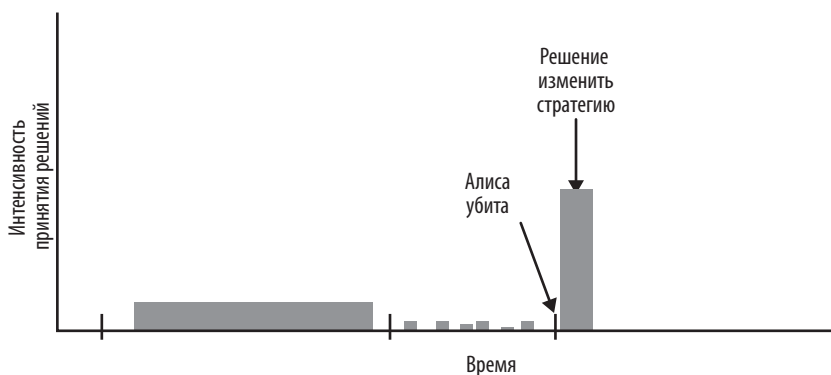
До тех пор пока Алису не застрелили, Боб не принимал особо сложных решений. Он определил стратегию еще до начала раунда, поэтому его мозг в основном находился в состоянии покоя, так как Боб действовал по плану. Он просто бежал по маршруту, который запланировал.

Однако теперь обстоятельства изменились и мозг Боба снова включается в работу. Скорость мышления увеличивается. Сосуд наполняется, но избыточного потока еще нет. Он принимает обстоятельства.

Боб видит, что его сопровождает только один член команды — меньше, чем он надеялся, поэтому пересечение моста будет сложнее, если вдруг им окажут сопротивление. Кроме того, препятствия в виде снайперских навыков Алисы больше нет, что делает его рискованный план бессмысленным. Боб знает, что где-то в Главном дворе его мертвый союзник бросил оружие и оно гораздо мощнее, чем то, которое у Боба. «Было бы неплохо заполучить эту пушку», — думает он.

После недолгих размышлений Боб решает отказаться от маршрута к мосту и направиться в Главный двор, чтобы присоединиться к членам своей команды.

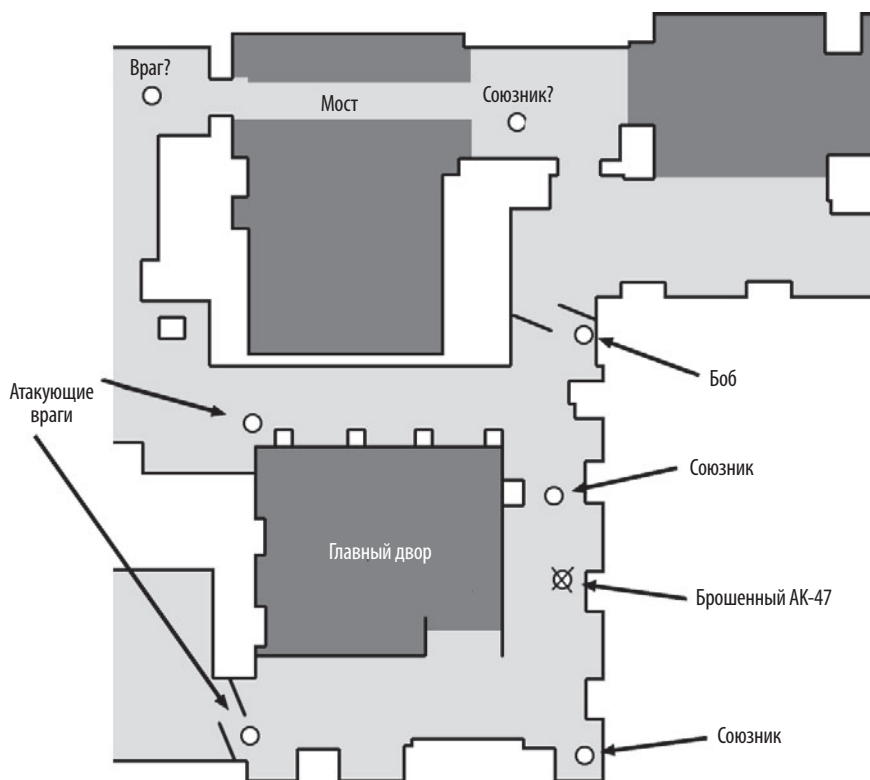
Он уходит от моста и проходит через большие Двустворчатые двери. Вот как выглядит решение Боба изменить стратегию:



Благодаря тому что Боб быстро среагировал в данной ситуации, он смог быстро принять решение и изменить свой план действий.

Когда Боб входит в Главный двор, он осматривается. Двое союзников, как и мертвый товарищ по команде, находятся в 20 метрах от него на открытой территории. Действительно, убитый оставил свой АК-47, гораздо более мощное оружие, чем автомат MP5 Боба. Боб хочет его забрать.

Но мгновение спустя, как только Боб проходит через двери, его союзники попадают под обстрел.



Мозг Боба находился в спокойном состоянии четыре секунды, с тех пор как он изменил курс на мостике. Но теперь его темп принятия решений снова растет.

У него есть несколько секунд, чтобы решить, хочет ли он попытаться заполучить АК-47, лежащий на открытой территории. Он хотел бы знать, кто именно в него стрелял и чем, но у него нет на это времени. Ему потребовалась бы еще одна секунда,

чтобы рассмотреть точное вооружение, внешний вид и движение врагов и союзников. И у Боба нет этой секунды, хотя вся эта информация находится перед ним. Таким образом, скорость препятствует получению информации.

Боб тотчас же пытается спрогнозировать будущее в соответствии с каждым из своих возможных вариантов. Исход скрыт за нечеткой причиной и следствием, но Боб может использовать свои знания и навыки, чтобы увидеть его хотя бы в расплывчатой, неопределенной форме. Он чувствует удовольствие, поскольку его мозг отрабатывает свои отточенные тактические навыки.

Исход, который видит Боб.

Он мог просто проигнорировать АК-47. В этом случае можно было бы остаться в укрытии, но его союзники могут умереть без его поддержки, и в этом случае ему пришлось бы сражаться с четырьмя врагами, имея только одного союзника и слабое оружие MP5. Но если его союзники выиграют бой без него, Боб сможет потом просто схватить АК-47, что даст ему больше возможностей в оставшемся раунде, а также в будущем.

Либо Боб мог бы побежать за АК-47 и попытаться схватить его. Когда он бежал бы по открытой местности Главного двора, его могли бы застрелить. С другой стороны, его появление могло бы отвлечь врагов и дать его союзникам время убить их или Боб мог бы захватить АК-47 и тоже вступить в бой.

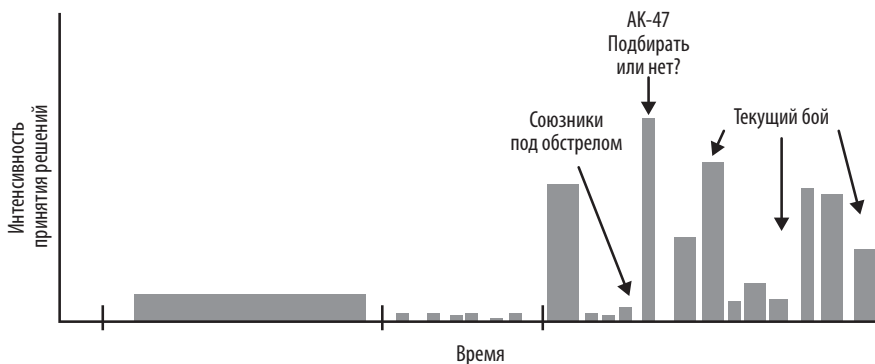
Это решение не из легких. Любой выбор может привести к катастрофе или триумфу. Это не головоломка, в которой вы разрабатываете решение; это личный выбор. Боб оценивает вероятность и оптимальность различных исходов и принимает решение. Этот процесс принятия решений выходит за пределы сознательного и бессознательного. В решающей машине осознанность и инстинкт сливаются воедино. Дело не в том, что Боб думает о своем выборе, ведь у него нет на это времени, а в том, что его эмоциональное бессознательное, натренированное многочасовой практикой, заставляет его думать об этих вариантах. И один из них будет лучше другого. Итак, Боб решает пойти за АК-47.

Больше я не стану подробно описывать конкретный исход этой ситуации. Возможно, Боб сразу же умрет и его темп принятия

решений мгновенно упадет до медленного биения пульса, а Боб в это время будет принимать решение о стратегии на следующий раунд. В этом случае он может испытать один из несовершенных исходов в дизайне Counter-Strike: время, проведенное замертво между раундами, иногда может длиться дольше, чем нужно игрокам для выбора своей стратегии на следующий раунд, тем самым создавая длинное унылое прерывание потока.

И наоборот, если Боб не умрет, он сможет вступить в десятисекундное неистовое сражение. Решения будут приходить быстро и неистово, один или несколько раз в секунду, когда союзники и враги стреляют, падают или спасаются бегством. Если этот насыщенный решением период длился намного дольше десяти секунд, это может стать утомительным. Но периоды интенсивного принятия решений в Counter-Strike всегда весьма скоротечны, потому что оружие смертельно.

Если предположить, что битва продолжается, в таком случае темп принятия решения будет выглядеть следующим образом:



Темп принятия решения в Counter-Strike — резкий и непредсказуемый. Игра балансирует между напряженным преследованием и дикими перестрелками, а между раундами — обязательная передышка. Игроки принимают решения, разные по объему и вкусу, потому что они сочетают судорожную стрельбу с тактическим движением и групповой стратегией. Наряду с недостижимым потолком мастерства именно это сильное сочетание решений с хорошим темпом делает Counter-Strike таким невероятно привлекательным игровым опытом.

Баланс

Маленький мальчик хотел печенье. Оно всегда было на кухне, но он не мог его достать.

Он перепробовал все, чтобы заполучить печенье. Залезал на табуретку, но та падала. Привязывал веревку к игрушкам и бросал их. Симулировал болезнь, чтобы из жалости ему дали печенье. Пытался заключить сделку со старшим братом. И даже пытался научить собаку приносить печенье. Ничего не получалось.

После каждой неудачи его мозг усердно работал над новыми решениями. Поиски печенья стали его целью в жизни. Каждая неудача становилась новой увлекательной задачей. Провалы делали его сильнее, хитрее и умнее.

Однажды кто-то опрокинул банку и печенье упало на пол. Мальчик съел его, затем отправился в свою комнату и неподвижно сидел лицом к стене. Печенье не наполнило его. Оно его опустошило.

Иногда баланс означает просто изменение числовых параметров. Дизайнер может уменьшить сцепление шин автомобиля с поверхностью, чтобы ухудшить его управление на поворотах, или увеличить скорость полета стрелы для большей эффективности. У игр существуют тысячи параметров, которые можно настроить

определенным образом: скорость, цена, масса, здоровье, урон, энергия и так далее.

БАЛАНС означает корректировку игровой механики для изменения относительной мощности различных инструментов, боевых единиц, стратегий, команд или персонажей.

Но баланс может также потребовать более фундаментальных изменений, чем обычный поворот ручки. Дизайнер может убрать способность мага защищаться щитом, чтобы его можно было легче атаковать лучниками, или убрать с определенного транспортного средства систему закиси азота, чтобы уменьшить высокую мощность двигателя.

В этой главе мы поговорим о балансе — зачем он нужен, почему это сложно — и о способах сделать это хорошо.

Цели баланса

Слово *баланс* является одним из самых некорректно используемых терминов в геймдизайне. Люди называют игру сбалансированной, когда она кажется забавной, или несбалансированной, когда она кажется нечестной. Но желание уместить все эти идеи в одно слово приводит к путанице между методом и целями.

Баланс — это метод. Этим словом называют изменение относительной мощности различной игровой механики. Этот метод может применяться для достижения практически любой цели. Связность задумки, ясность, простота и элегантность — все это можно улучшить благодаря балансу. Например, дизайнер может уменьшить здоровье нищего, чтобы подчеркнуть его вымышленную немощность, или увеличить скорость реактивного самолета, чтобы акцентировать его значимость.

Но среди всех различных целей, которые можно достичь благодаря балансу, две стоят на ступень выше. Это *честность* и *глубина*.

Эти цели связаны с балансом, потому что он является ключевым методом их достижения. Остальные цели дизайна в основном достигаются другими способами. Мы рассказываем историю с помощью графики или текста, мы делаем игры понятными с помощью хорошего дизайна интерфейса. Но чтобы игра была честной и глубокой, нужен баланс.

Баланс честной игры

Мы стремимся к честности, потому что она позволяет игрокам чувствовать, что их победы и поражения полноправны. В соревновательных играх игроки хотят знать, что их победы означают, что один игрок действительно лучше, чем другой. Если сама игра нечестна, соревновательный ритуал не имеет смысла, потому что он ничего не раскрывает. Проигравший может жаловаться на то, что проиграл, только потому, что игра была несправедливой, а победитель не получит удовлетворения от несомненной победы.

Игра считается ЧЕСТНОЙ, если вначале ни у одного из игроков нет преимущества.

Некоторые виды игр автоматически становятся справедливыми, потому что игроки начинают игру на равных условиях. Такая игра называется симметричной. Например, хоккей симметричен, потому что обе команды стартуют на одинаковых позициях и подчиняются одним и тем же правилам. Хоккей автоматически считается честной игрой, потому что единственная разница между командами — сами игроки. При разработке симметричной игры можно не думать о честности, она появляется автоматически.

Но по-настоящему симметричные игры необычны. Большинство игр не являются симметричными. В них игроки могут начинать игру с разных условий. Например, в файтинге они могут использовать разных персонажей с разными атаками. В шахматах белые ходят первыми. А в StarCraft игроки могут выбирать командование армиями разных видов. В отличие от симметричных, несимметричные игры не гарантируют честности. Чтобы эти игры стали честными (или, по крайней мере, достаточно честными), они должны быть хорошо сбалансированы.

Может показаться, что сделать игру симметричной — это простой способ достичь баланса. Но часто несимметричность является неотъемлемой частью игры. В шахматах кто-то должен ходить первым. У игроков в Risk должны быть разные стартовые территории на несимметричной карте мира. В игре World War II кто-то должен играть за страны «оси», а кто-то — за союзников. Часто симметрия не является подходящим вариантом.

Кроме симметрии и сбалансированной асимметрии существует и третий вариант. Мы можем создавать несимметричные игры, которые заведомо нечестны. При таком подходе теряется чувство конкурентной оправданности, но он открывает уникальный опыт, который невозможно получить в честной игре.

Например, во многих исторических военных играх нечестный сценарий, потому что те, кто играет в такие игры, больше заинтересованы в изучении истории, чем в сражениях. Они хотят выяснить, могла ли немецкая армия победить в битве при Арденнах, находясь в заведомо проигрышном положении, или могли ли японцы выстоять в битве за Иводзиму. Такие игры должны быть нечестными, потому что исторические битвы были нечестными.

В других случаях дизайнеры делают игру нечестной, потому что благодаря этому легче создать дикие истории и забавные формы взаимодействия. Например, классическая настольная игра Cosmic Encounter наделяет каждого игрока несбалансированными и забавными способностями, потому что игра создана для того, чтобы генерировать юмор и, как сказал дизайнер Питер Олокта: «Честность — это не смешно!» Именно поэтому в Cosmic Encounter нет никаких состязаний, но зато игра невероятно смешная.

Баланс для глубокой игры

Я описал глубину как свойство игры, которое обеспечивает осмысленную игру на высоком уровне мастерства. Баланс крайне важен для глубины. Потому что для того, чтобы игра была глубокой, она должна генерировать решения, которые были бы настолько сбалансированы, что даже эксперты засомневались бы в выборе ответа.

Например, представьте, что герой в игре жанра фэнтези, где присутствуют мечи и волшебство, встречается кровожадного огра. У персонажа два варианта: он может атаковать мечом или наложить заклятье, чтобы огр загорелся. Мечом можно сразу нанести много урона, а заклинание будет наносить небольшой урон постепенно. Что выберет игрок?

Если какой-либо из этих вариантов явно более мощный, например если меч нанесет больше урона, чем огонь смог нанести за 10 минут, ответ будет очевиден и решение превратится в отсутствие решения. Чтобы это решение имело смысл, различные варианты, которые выбирает игрок, должны быть сбалансированы, чтобы лучший ответ не был очевидным. В этом случае дизайнер может сделать так, чтобы меч наносил больший урон в целом, если бой длится менее 30 секунд, и сделать больший урон от огня, если враг горит более 30 секунд. Теперь игрок не просто выбирает очевидный ответ. Он должен угадать, продлится ли бой больше или меньше 30 секунд. Угадывать будущее — это интересный и эмоционально значимый мыслительный процесс.

Некоторые относятся к такому балансу так, как будто речь идет о сравнении силы действия различных инструментов. Они пы-

таются сбалансировать меч, пламя и любой другой инструмент так, чтобы сделать их равными. Но такой подход не работает, потому что в данном случае цель баланса понимается неправильно. Наша цель — не сбалансировать инструменты. Это невозможно, потому что у каждого инструмента свой уровень значимости в разных ситуациях и комбинациях с другими инструментами. Огонь может быть волшебным средством против огров, но он бесполезен против толпы слабых гоблинов. Меч может быть хорошим средством против гоблинов, но он бесполезен против огра. Как можно в таких ситуациях понять, что лучше? Без контекста такие сравнения бессмысленны.

Наша настоящая цель — сбалансировать стратегии, которые игрок выбирает в той или иной ситуации.

СТРАТЕГИИ — это конкретные наборы действий, которые игроки могут решить предпринять для достижения цели. Решения в игре становятся богаче, если мыслительный процесс, необходимый для поиска лучшей стратегии, становится более сосредоточенным на деталях.

Таким образом, в примере с огром жанра фэнтези мы на самом деле не приходим к балансу между использованием меча и заклинания с огнем. Мы уравниваем выбор между этими двумя стратегиями в этой конкретной ситуации с этим конкретным огром.

Простая стратегия может означать использование одного инструмента для решения одной проблемы. Например, если ваши противники наступали с кавалерией, хорошей контрстратегией было бы разместить копейщиков в первых рядах своей армии. Поскольку атака лошадьми против укрепленного строя копейщиков с длинными острыми копьями не очень эффективна, ваша стратегия является выигрышной. Для достижения одной цели вы использовали один из своих инструментов против одного из инструментов противника.

Но стратегии также могут быть чрезвычайно сложными. Подобные стратегии — это не единичные действия, это детальное развитие событий. Они зависят от синергии нескольких инструментов в определенных обстоятельствах и адаптированы к определенным комбинациям противоположных стратегий. Например, эксперт увидит вражескую армию с кавалерией справа, лучниками слева и копейщиками сзади. Его отвлекающим маневром будет направить кавалерию на вражеских лучников,

а копейщиков — в центр. Он продумывает несколько вероятных исходов: если лучники противника начнут стрелять в копейщиков, его кавалерия может атаковать их. Но если лучники врага начнут отступать и в ответ будет направлена кавалерия, наш эксперт сможет поймать их в ловушку с помощью копейщиков.

Если стратегии сбалансированы, мы получаем более богатый игровой опыт, потому что все решения игрока включают в себя больше нюансов и размышлений о большем количестве вариантов. Игра становится глубже, потому что эти сложные мыслительные процессы не могут протекать идеально, даже если это очень опытный игрок.

Баланс для остальных случаев

Ядром баланса является создание честной игры и тонких решений. Но даже если изменение баланса создает удивительно богатые решения и кристально честную игру, оно бесполезно, если при этом разрушаются связность нарратива, поток и темп, доступность или ясность.

Например, мы не можем сбалансировать лук лучника, обеспечив ему диапазон в 3 метра. Сделать так может быть честным, и появятся захватывающие стратегические решения, но в этом не будет никакого смысла в сюжетном слое. Мы не можем сбалансировать головоломку с прыжками, сделав прыжок персонажа короче только на одном уровне, потому что это не будет соответствовать ожиданиям игрока и он будет разочарован. Мы не можем сбалансировать автомобиль в гонках, сделав его мучительно медленным и вялым, потому что ехать за рулем такого автомобиля будет настоящей пыткой, даже если это будет честно.

Баланс должен вписываться в рамки этих ограничений. Или как минимум польза изменения баланса должна перевесить ценность другого опыта, так как баланс влияет на все.

Вырожденные стратегии

Один из парадоксов геймдизайна заключается в том, что добавление инструмента может фактически привести к тому, что

игра потеряет интересные решения, а не получит их. Это происходит в том случае, если новый инструмент создает *вырожденную стратегию*.

Например, представьте, что разработчики стратегии добавляют нового персонажа — Чака Норриса. Чак, будучи идеальным человеком, очевидно, самый сильный в игре. Он может победить целую армию солдат в одиночку.

ВЫРОЖДЕННАЯ
СТРАТЕГИЯ — это
стратегия, которая
является лучшим
очевидным выбором
в данном решении.

Поначалу может показаться, что Чак — это отличный дизайнерский ход. Он потрясающий и дает игрокам то, что они хотят. Но на самом деле Чак уничтожает саму игру почти так же, как и всех своих врагов.

Добавление Чака Норриса в игру уменьшило ее глубину, так как игрок больше не решает, кем ему играть. Независимо от ситуации, ответ всегда один и тот же: выбери Чака. Он — вырожденная стратегия.

Чак Норрис — упрощенный пример. В реальном дизайне вырожденные стратегии никогда не бывают такими очевидными. Они прячутся в эмергентных взаимодействиях различных инструментов и механик.

Например, в фэнтези RPG The Elder Scrolls: Morrowind можно стать невероятно сильным, создавая эликсиры, улучшающие интеллект персонажа, а затем применять недавно улучшенный интеллект для создания еще более действенных эликсиров и так далее, в геометрической прогрессии. Когда интеллект игрока в несколько сот раз превосходит условно нормальное значение, игрок может смешивать эликсиры, которые значительно улучшат все остальные характеристики навсегда.

Таким образом, в течение нескольких минут после начала игры игрок может создать персонажа, который может прыгать через горы и убивать драконов одним ударом. Этот трюк прост и легко осуществим, любой, кто его знает, может обесценить многие тщательно продуманные задачи игры. Не сразу понятно даже при анализе дизайна игры, что такое в принципе возможно.

Вырожденные стратегии существуют даже в спорте. Например, возьмем баскетбол. Трудно представить, чтобы люди исполь-

зовали этот традиционный вид спорта так, как они используют дисбаланс в видеоиграх. В конце 1990-х годов появилось множество необычных игроков, которые превосходно нападали, но не могли справиться со штрафными бросками. Шакил Рашоун О'Нил был самым известным примером. В ответ команды противников разработали стратегию под названием «Хак-э-Шак», в которой они намеренно пытались совершить нарушение на Шаке, когда у его команды был мяч. В баскетболе, если игрок физически препятствует другому игроку, судья (рефери) фиксирует нарушение (фол), а игрок, совершивший фол, получает штрафной бросок. Соперники верили скорее в то, что Шак пропустит штрафной бросок, чем в то, что его команда отдаст мяч. Поэтому игра превращается в безуспешную попытку соперников догнать Шака и коснуться его, в то время как мяч находится далеко. Игроки всегда пытаются найти вырожденные стратегии. Они бесконечно охотятся за слабыми местами геймдизайна, желая найти дисбаланс, который можно будет использовать для легких побед. Вся ирония в том, что если они когда-нибудь найдут его, они будут ненавидеть дизайнера за то, что он позволил им разрушить игру. Они хотят охотиться по вырожденным стратегиям и не хотят их найти.

Ошибочность расчета жизнеспособных стратегий

Чтобы решение что-то значило, необходимо несколько жизнеспособных стратегий, которые могли бы привести к положительному исходу. Если такая жизнеспособная стратегия только одна, она является вырожденной и происходит ситуация мгновенного решения.

Я долгое время считал, что цель баланса — максимизировать количество жизнеспособных стратегий. Идея заключалась в том, что чем более жизнеспособными будут стратегии, тем богаче будут решения и тем лучше будет сбалансирована игра. Я написал всю главу, основываясь на этом предположении, и на бумаге это было прекрасно. Затем я отправился на поиски контрпримеров. И, к своему ужасу, я нашел два таких примера, которые разбили в пух и прах все, что я написал.

Первой была шуточная игра: «камень-бумага-ножницы-ящерица-Спок». Традиционная игра «камень-ножницы-бумага» имеет

три жизнеспособные стратегии. Но существует еще одна версия этой игры — «камень-бумага-ножницы-ящерица-Спок» (мой любимый результат — «на бумаге улики против Спока»). В этой версии пять стратегий, и все они жизнеспособны, потому что у каждой равные шансы на победу. И мы можем легко добавлять все больше и больше символов в эту игру, вплоть до сколь угодно большого количества жизнеспособных стратегий. Но улучшаем ли мы игровой баланс? Конечно, этого не происходит, игра не становится глубже, а просто сложнее. Увеличение количества жизнеспособных стратегий не сделало игру лучше.

Вторым контрпримером был покер. Он не стал плохой игрой с большим количеством стратегий, все произошло с точностью до наоборот: это отличная игра с очень небольшим количеством стратегий. Покер бесконечно интересная игра, но в каждой ситуации можно сделать только несколько ходов. В распоряжении у многих игроков всего две жизнеспособные стратегии: фолд или колл. Если бы число жизнеспособных стратегий имело значение, то как бы покер стал такой замечательной игрой?

Со временем я понял, что идея расчета жизнеспособных стратегий была отвлекающим маневром. Я рассуждал так (и заблуждался): если две жизнеспособные стратегии лучше, чем одна, это должно означать, что три жизнеспособные стратегии лучше, чем две, верно?

Нет. После того как вы получили две жизнеспособные стратегии, нет смысла добавлять еще. Если у вас будет больше жизнеспособных стратегий, решение может стать интереснее. Но все равно это не лучший способ. Мы могли бы изобрести версию покера с гораздо большим количеством полезных ходов, но это не значит, что игра обязательно станет лучше.

Реальная цель баланса для глубокой игры — создать интенсивный мыслительный процесс в мозгу игрока. Мы хотим зажечь захватывающую цепочку внутреннего алгоритма, который дает игрокам озарение, сомнения и дилеммы, даже если это очень опытный игрок. Для этого нам нужно более одной жизнеспособной стратегии. Но как только у нас остаются две такие жизнеспособные стратегии, добавление большего количества не улучшает

опыт автоматически. Улучшение опыта означает придание процессу принятия решений больше деталей.

Эта идея подразумевает нечто важное для дизайнеров. Она означает, что практика увеличения вариантов выбора как способ углубления игры является неправильным. Возможно, создать больше вариантов легко, но по своей сути в этом нет ничего ценного. Часто создается только больше сложностей.

Настоящая цель дизайнера — обогатить внутренний опыт игрока. Этой цели труднее достичь, и ее трудно оценить, но это правда. Стремление к этой цели делает дизайн меньше, проще, элегантнее, а также более ориентированным, чем он был бы при расчете стратегий.

Баланс и навык

Маленькие дети любят играть в крестики-нолики. Для них это настоящая игра с реальными навыками, настоящим испытанием и стратегией. У взрослых она вызывает смех, потому что очень простая, а идеальная стратегия очевидна. Нам надоело крестики-нолики. Так как же увлекательная игра для одной группы может быть бессмысленной для другой?

Там, где эксперт видит вырожденную стратегию, новичок видит волшебство. Эксперт знает вырожденную стратегию и обладает навыками для ее реализации. Когда он играет в игру, он делает одно и то же каждый раз, не задумываясь, так же как взрослый играет в крестики-нолики. Но новичок не заметил вырожденную стратегию или ему не хватает навыков для ее реализации. Для новичка игра до сих пор остается загадкой. Дети любят крестики-нолики, потому что у них еще нет возможности использовать вырожденную стратегию. Так что для них ее, возможно, и не существует.

Мы также выявили обратное: игрокам не нравится игра, потому что им не хватает навыков использовать стратегии, которые делают ее сбалансированной. Например, StarCraft II — одна из лучших сбалансированных игр за всю историю, но новые игроки обычно так не считают.

Игра, которую игроки с одним уровнем навыков считают сбалансированной, может быть несбалансированной для игроков с другим уровнем навыков, поскольку разный уровень навыков дает доступ к разным стратегиям.

В StarCraft II существует класс дебютных стратегий, называемый «раш» («натиск, напор»). Они предполагают раннее нападение на другого игрока, прежде чем он вообще построит какие-либо военные подразделения. Они относительно просты в реализации, что делает их доступными для новичков. Но их довольно сложно остановить. Игрок среднего уровня может остановить раш, а новичок — нет. Новичку, которого снова и снова побеждают другие новички, использующие раш, кажется, что StarCraft II — несбалансированная игра.

Но раш совершенно не оправдан при более высоких уровнях мастерства. Профессиональные игроки редко применяют раш по отношению друг к другу, потому что эта стратегия обычно не работает против тех, кто знает, как ее остановить. На вершине диапазона навыков преимущества остаются именно у защитников от раша. С другой стороны диапазона навыков эти преимущества достаются противникам, так как рашем легко начать пользоваться, но сложнее прекратить.

Это создает странную ситуацию, когда раш является вырожденной стратегией, но только на низких уровнях навыков игрока. Новички не могут использовать стратегии для противодействия рашу, поэтому игра кажется им вырожденной.

Кому нужен баланс

Крестики-нолики и StarCraft II представляют разновидность одной и той же проблемы.

Если мы обнаруживаем проблему на одном уровне мастерства, то обычно можем ее решить. Но как только мы начинаем беспокоиться о нескольких уровнях мастерства, количество стратегий, которые мы должны рассмотреть, значительно увеличивается. Сбалансировать одно очко мастерства сложно; сбалансировать все сразу почти невозможно. Единственный ответ — позволить игре быть несбалансированной на некоторых уровнях навыков игрока. Это звучит как признание поражения, но так делают почти все игры.

Даже группа разработчиков, отвечающих за баланс мастерски разработанной игры StarCraft II, не скрывает своей готовности

пожертвовать балансом при низких навыках в обмен на баланс при высоких. Дастин Браудер, ведущий дизайнер StarCraft II, сказал: «Цель всегда состоит в том, чтобы найти решения, которые отразятся на всех... Но когда вы приставляете пистолет к моей голове и говорите: “Вы должны принять решение”... мы соглашаемся [с самыми опытными игроками]». Браудер признает, что баланс игры по всему диапазону навыков игрока невозможен. Вместо этого его команда сосредоточивается на том, чтобы игра оставалась сбалансированной на уровне экспертов, при этом рационально улучшая все, что возможно, в остальном диапазоне мастерства. И для StarCraft II это правильный выбор, так как игра предполагает долгое освоение профессиональными игроками.

В нарративных играх обычно используется противоположный подход. Вместо того чтобы балансировать в верхней части диапазона мастерства, они балансируют в середине или внизу, потому что они не предназначены для таких интенсивных игр, как соревновательные, вроде StarCraft II. Например, в BioShock есть невероятно сильные враги Большие Папочки, которые не агрессивны, если их не спровоцировать. Предполагается, что битвы с Большими Папочками будут жесткими и решающими, но на самом деле существует множество вырожденных способов убить их, не сражаясь вообще. Игрок может прикрепить несколько мин к бочковой бомбе, чтобы создать гигантскую супербомбу, которая убьет Большого Папочку за один удар. Или Большого Папочку можно заманить во множество ловушек и тоже мгновенно убить.

BioShock полон подобных вырожденных стратегий, но они не имеют значения, потому что игра длится менее 10 часов и лишь немногие игроки разберутся с ними за столь короткое время. И даже если они это сделают, игра остается интересной, потому что ее смысл заключается в нарративе и ролевой игре, а не в улучшении навыков.

Вот почему вырожденная стратегия в Morrowind, представленная в виде зелья интеллекта, не портит игру. Morrowind — это не победа, а исследование мира. Игрок может один раз попробовать

Почти невозможно создать игру, основанную на навыках, которая была бы сбалансирована для игроков всех уровней навыка. Дизайнер должен определить, какой уровень умений он хочет сбалансировать, и дать возможность другим уровням навыков иметь выродившиеся стратегии.

трюк с зельем, но потом быстро вернется к обычной игре, потому что захочет прочувствовать нарратив.

Нужен ли баланс

В любой игре дизайнер должен решить, стоит ли вообще гнаться за балансом на высоких уровнях навыка игрока. Ответ зависит от того, насколько игра связана с задачами, требующими навыков, а также других видов опыта, таких как графика и история.

Баланс на вершине диапазона мастерства стоит дорого. Это значит, что необходимо устранить любую идею, которая может привести к вырожденной стратегии на любом уровне мастерства. Это ограничение закрывает дорогу многим идеям, которые имеют смысл в других отношениях. Кроме того, тестирование при высоких уровнях мастерства стоит дорого потому, что в этом случае кто-то должен потратить на изучение игры недели или месяцы, выискивая оптимальные стратегии. Наконец, игроки обычно находят вырожденные стратегии после выхода игры, а это значит, что ее нужно исправлять — иногда на это требуются годы.

Баланс на более низких уровнях мастерства значительно дешевле. Пока вырожденные стратегии не совсем очевидны, цель достигнута. В игру можно включить механики, которые создают солидный нарративный или социальный опыт, даже если они приводят к появлению вырожденных стратегий. Тестировать баланс легко, так как здесь не требуется длительного обучения или особенно преданных игроков. И не страшно, если кто-то найдет новую вырожденную стратегию после выхода игры. В нее все еще приятно играть, так как дело не в навыках.

В игре, основанной на навыках, которая стремится поддерживать бесконечно глубокую стратегию, баланс высокого уровня навыков игрока не обсуждается. Если такая игра не приспособлена к умелым действиям игроков, она бесполезна. Это означает трату ресурсов дизайнера на исчерпывающий анализ баланса, запрет на многие сюжетные идеи, которые невозможно сбалансировать, и даже возможность дисбаланса на более низких уровнях навыков. Это неотъемлемая цена за разработку совершенной игры. Но игры с прогнозируемым исходом, такие как StarCraft II, Counter-Strike или Street Fighter II, стоят такой цены.

Разработчики тех игр, которые не основаны на интенсивном соревновании, должны обеспечить баланс для низкого и среднего уровней навыков игрока. Основа смыслового наполнения этих игр заключается в общении, нарративе или других не зависящих от навыков эмоциональных триггеров, поэтому затраты на баланс высокого уровня навыков того не стоят. Игры с сюжетом, такие как BioShock и Morrowind, незначительно выигрывают от глубокого баланса. Эти игры лучше оставить несбалансированными, чтобы ресурсы дизайна можно было направить на улучшение игрового мира и истории.

Сложные задачи баланса и их решение

Давайте вернемся к примеру персонажа, оказавшегося лицом к лицу с огром. У героя два инструмента — меч, который наносит много мгновенного урона, и заклинание огнем, которое наносит небольшой урон постепенно. Если бы эта механика использовалась только в одном месте, игру было бы легко сбалансировать. Но в реальной игре, особенно в элегантной, эта механика используется и в тысячах других мест. Герой также может использовать меч или пламя против гоблинов, орков или скупых лавочников. Несложно настроить эти два инструмента так, чтобы результат выбора хорошо работал против одного овра. Но любое изменение, сделанное для того, чтобы исправить конкретно эту ситуацию, влияет на всю остальную игру в тех местах, где используются эти инструменты.

Это фундаментальная задача баланса. Мы часто хотим решить одну проблему. Но любое наше изменение будет иметь много разных последствий в разных местах игры.

Хорошая игра — это сложная система в потрясающем академическом смысле. Она демонстрирует нелинейное, непредсказуемое эмергентное поведение, которое намного сложнее, чем сам дизайн. В сложных системах изменения одной переменной влияют не только на эту переменную, и при этом они не давят на другие переменные простыми, предсказуемыми способами. Скорее они могут привести к появлению сложных причинно-следственных цепочек,

Настройка механики меняет все стратегии, в которые она встроена, а не только те, которые мы собираемся изменить.

которые практически невозможно спрогнозировать. Это одна из величайших возможностей игр, так как она позволяет значительно разнообразить опыт при простом дизайне. Но это также одна из сложнейших задач геймдизайна — особенно баланс.

Психолог Дитрих Дернер говорит о том, как нелегко управлять сложностью:

«Мы можем сравнить человека, принимающего решения в сложной ситуации, с шахматистом, в распоряжении которого намного больше фигур, чем в обычном наборе, допустим, несколько десятков. Кроме того, все эти шахматные фигуры связаны друг с другом резинками, так что игрок не может перемещать только одну фигуру. Кроме того, его пешки и фигуры и пешки и фигуры противника могут двигаться самостоятельно и по правилам, которые игрок до конца не понимает или которые он неправильно истолковал. И в довершение всего, некоторые его люди и люди его противников укутаны туманом, который помогает им скрыться.»

Это проблема игрового баланса. Изменение одной механики изменяет каждую стратегию, связанную с этой механикой, которая, в свою очередь, меняет каждую стратегию, связанную с этими стратегиями, и так далее, в геометрической прогрессии. Вы можете ослабить силу меча, чтобы решить исход битвы с огром, и в итоге увидеть, что меч больше не действует против гоблина. Вы можете затем уменьшить силу гоблинов и обнаружить, что теперь их слишком легко победить посохом волшебника и безоружными атаками. Все взаимосвязано.

Вот почему баланс — сложен, а отличный баланс — редкость. Если вы будете подходить к этому вопросу кое-как, игра будет переходить из одного несбалансированного состояния в другое, поскольку каждое решение приводит к еще большим проблемам, а не их решению. Единственный способ добиться реального прогресса — это тщательно структурированные, продуманные подходы, которые решают проблемы, не создавая при этом новых. Давайте рассмотрим некоторые из них.

Методы создания баланса

Реактивный рюкзак должен запускать его владельца далеко и быстро, потому что он предназначен именно для этого. Броня должна защищать. Артиллерийская пушка должна пускать снаряды на дальние расстояния; в противном случае это не артиллерийская пушка. Зерновые культуры должны обеспечивать еду; в противном случае это не зерновые культуры.

Эти свойства важны для роли и задумки этих инструментов. Так что продвигайте их как можно дальше. Заставьте ракетный рюкзак запускать людей действительно очень далеко и быстро. Сделайте броню невероятно прочной.

Когда мы выжимаем из них максимум, инструменты становятся запоминающимися, а их роль — предельно понятной. Игровой опыт расширяется, и по мере того, как игроки исследуют более широкие возможности, появляется более обширный спектр стратегий.

Эти ключевые свойства настолько важны, что мы должны рассматривать их как неизменные, потому что, если бы мы их изменили, мы бы стерли роль инструментов и нарушили их связанность с задумкой. Если сравнивать каждое свойство с рычагом, который дизайнер может повернуть, мы должны нажать на него до максимума, а затем зафиксировать на месте.

Так как эти рычаги зафиксированы, мы должны балансировать, поворачивая другие рычаги. К счастью, у каждого инструмента много свойств, которые для него не являются первостепенными. Рюкзак-ракета должен быть быстрым, но мы можем делать все, что захотим, с его ценой, весом или уязвимостью. И если для баланса этого недостаточно, мы можем создать новую механику, чтобы усилить или ослабить его функциональность. Если он слишком мощный, мы можем заставить его взорваться при повреждении, или сделать так, что из него будет непрерывно вытекать топливо, или рюкзак будет медленно высасывать здоровье владельца. Если же рюкзак слишком слабый, мы можем позволить ему прикрыть спину владельца вместо щита или он сможет сделать владельца бесшумным, что-

Выясните, какие аспекты инструмента важны для его роли и особенностей. Поворачивайте эти рычаги как можно дальше и зафиксируйте их на месте. Затем решите проблемы баланса, поворачивая другие рычаги.

бы враги не слышали, как он на них падает. Но ни при каких обстоятельствах мы не можем замедлить его.

Точно так же броня может быть дорогой, заметной, тяжелой или тормозящей движение. Она может не дать игроку носить второе оружие или быть слишком шумной, привлекая этим внимание врагов. Но ее нельзя сделать хрупкой.

Находя ключевые свойства каждого инструмента и «включая» их до предела, мы обеспечиваем в игре наличие набора различных инструментов, охватывающих широкий спектр возможностей.

Иногда нельзя сбалансировать инструмент, не меняя его основные свойства. В этих случаях часто лучше просто убрать их. Отсутствие рюкзака-ракеты часто лучше, чем жутко медленный и стратегически бессмысленный рюкзак-ракета.

Урезайте столько,
сколько необходимо
для решения проблем.

Этим методом славится Blizzard Entertainment, студия, разработавшая StarCraft II, Diablo и World of Warcraft. Если дизайнеры Blizzard не могут повернуть рычаг и идеально сбалансировать инструмент, они без сомнений урезают столько, сколько им необходимо, чтобы решить проблему.

Например, в StarCraft II есть большой пешеходный объект, который называется Тор. На ранних этапах разработки он был гораздо массивнее, чем все остальное в игре. Он был настолько большим, что его нельзя было забрать с завода и приходилось его строить в поле, как здание. После разрушения его можно было частично «воскресить». Он двигался медленно, поворачивался медленно, и убить его было практически невозможно. В нем все выражало концепцию сверхмассивного механического ходока, и это было потрясающе.

Но его невозможно было сбалансировать. Строительство в поле означало, что такие объекты можно было производить в большом количестве слишком быстро, чтобы с ними можно было сражаться, так как производство шло быстрыми темпами. Возможность «воскрешения» делала его практически непобедимым. Его неповоротливость делала его легкой добычей для маленьких

шустрых персонажей, бегающих вокруг (классическая вырожденная стратегия). Эти проблемы не были очевидными и не лежали на поверхности — они были замечены опытными игроками во время тестирования.

Экзотическая механическая концепция была очень интересной. Она звучала убедительно и выглядела нетривиально. Впрочем, если баланс невозможен, использовать ее нельзя. Дизайнеры Blizzard не пытались решить проблемы со слоями правил неэлегантных особых случаев. Они сжали зубы от боли и отрезали максимум, превратив Тора в механизм адекватного размера, который строят на заводе, как любую другую боевую единицу, и теперь его нельзя воскресить. Выглядит скучно, но работает. И в конце концов, игра получилась лучше, чем она была бы при наличии в ней всевозможных функций для галочки, дисбаланса и нестандартных моментов, созданных во имя чьей-то любимой идеи.

Дизайнеру это решение дается очень нелегко — интеллектуально, социально и эмоционально. Выбрасывать огромный объем работы и такие свежие и интересные идеи из-за одного стратегического взаимодействия, спрятанного где-то в пространстве возможностей игры, может показаться ужасно расточительным. Это действительно эмоциональная боль. Но если вам нужен хороший баланс, придется пойти на это. В игре всегда будут находиться оставленные там вырожденные стратегии, и иногда единственный способ решить проблему — это убрать прикольные идеи. Как и в любой творческой профессии, иногда геймдизайнер должен убивать свои любимые создания.

Если что-то в игровом тесте идет не так, часто мы сразу же начинаем спешить и поворачивать некоторые рычаги, чтобы убедиться в том, что ситуация больше не повторится. И это всегда легко. Но пытаться достичь баланса, решая одну проблему за другой, — все равно что убирать пузыри из-под наклеенных обоев. Поскольку стратегии взаимосвязаны, каждое решение одной проблемы является причиной другой. Игра находится между точками баланса, повторяется, но остается на месте. В худшем случае это может продолжаться вечно, когда игра вообще не продвигается вперед.

Не торопитесь.

Ошибка здесь в том, что мы концентрируемся на имеющихся у нас проблемах и игнорируем те, которых у нас нет. Как говорит Дитрих Дернер:

«Мы можем верить, что преследуем одну цель, до тех пор пока не достигнем ее, а затем осознаем, с удивлением, раздражением и ужасом, что, избавляя себя от одной чумы, мы, возможно, создали еще две в других местах. Другими словами, существуют «неявные» цели, которые мы можем сначала вообще не принимать во внимание и даже не знать, что стремимся к их достижению. В качестве простого примера приведем следующий: если мы спросим здорового человека о его целях, он обычно не называет «здоровье» в качестве одной из них. Тем не менее оно является неявной целью, поскольку если бы задали конкретно этот вопрос, человек согласился бы с тем, что поддерживать свое здоровье важно. В целом, однако, здоровье станет одной из явных целей, только если человек заболит.»

Человеку по своей природе свойственно игнорировать проблемы, которых у него нет. Существующие проблемы наносят эмоциональный удар, а потенциальные проблемы весьма абстрактны. Легко осмысленно думать о происходящем; гораздо сложнее думать о всех других потенциальных проблемах, к которым могут привести наши решения.

Но нам приходится. Изменения баланса улучшают игру, только если они создают меньше проблем, чем решают. Легко найти изменения, которые решают одну проблему, создавая при этом две другие, о которых мы не знаем. Гораздо сложнее найти решения, которые справляются как с явными, так и с неявными целями, — такие, которые решают проблему и не создают новые.

Такой подход не означает, что не надо реагировать — просто притормозите, дышите и мыслите широко. Подавите это желание спешить и менять что-либо, если оно идет не так. Притормозив, вы получите время, чтобы успокоить эмоции, и более широкое представление о последствиях изменений. Потому что то, что кажется самым простым решением, часто сопровождается ужас-

ными побочными эффектами. Реальное решение, как правило, не так очевидно.

Сбалансированная игра не означает, что каждая стратегия дает одинаково ценные результаты. Такая игра бессмысленна так же, как и подбрасывание монетки: она сбалансированная, но скучная. Нет награды за мастерство, и нечего изучать.

На стратегическом ландшафте игры должны быть холмы и долины. Пики невероятной высоты и глубокие провалы. Причина, по которой эти различные результаты все еще могут быть сбалансированы, заключается в том, что их цена входа варьируется. Навык, необходимый для выполнения стратегии, сопоставим с ее эффективностью; лучшие результаты требуют высочайшего мастерства. Такой подход мотивирует игроков неустанно искать стратегический ландшафт и самые привлекательные вершины, пробираясь вверх по просторам навыков. И когда после великой борьбы они достигают вершины, игра за это их щедро вознаграждает.

Это результат, который нам нужен. Но вы не можете достичь его, будучи слишком рациональными. Будучи дизайнером, если вы видите что-либо, что кажется вам слишком хорошим или слишком плохим, первым желанием часто является шлифовать это до тех пор, пока оно не станет гладким и ровным. Так игра становится сбалансированной, но скучной. Она теряет природу непредсказуемости, обещание ценных уроков, напряжение серьезных перемен.

Решение состоит в том, чтобы периодически смягчать аргументированный рационализм безумием. Я называю это моментом Найджела Тафнелла, в честь вымышленного музыканта в фильме «Это — Spinal Tap». У Найджела был набор усилителей с ручками регулировки громкости, которые можно было крутить до 11 (вместо обычных 10). Когда он хотел добавить еще больше звука на сцене, он поворачивал ручки до 11. Вот так он вышел из «тюрьмы нормальности».

Мы можем сделать то же самое, сознательно отпуская разум. Вы должны на мгновение забыть о своем ремесле и превратиться в наивного восемнадцатилетнего хакера и дизайнера, живуще-

Иногда устраивайте «момент Найджела Тафнелла». Повысьте громкость на 11.

го в 1992 году и подсевшего на Red Bull, который *просто хочет сделать что-то невероятное!*

Что, если бы этот щит фактически делал вас совершенно неуязвимым, вместо того чтобы просто принимать на себя часть урона? Что, если бы он защищал всю вашу команду? Что, если бы этот персонаж двигался в 5 или 10 раз быстрее, чем кто-либо другой в игре? Что, если бы он двигался со скоростью света или мог телепортироваться? Что, если бы вы могли прыгнуть на 300 метров или получить неограниченную сумму денег? Разве это не было бы экстремально, замечательно и невероятно?

Когда Red Bull перестает действовать, вы обычно обнаруживаете, что ручку звука нужно вернуть назад. Большинство из этих экспериментов не работают, и поэтому профессиональные дизайнеры, как правило, работают лучше, чем восемнадцатилетние хакеры. Но иногда момент Найджела показывает нам упущенную возможность или напоминает об эмоциональном моменте, который мы ослабили. В этом случае, вместо того чтобы отказаться от изменений, включите их в игру и сохраните. Делайте так иногда, и тогда в игре будет изюминка безумия.

Чтобы сбалансировать игру, нужно играть с другими людьми. Обычно при тестировании игры вы получаете бонусом кучу предложений. Один игрок хочет, чтобы лошадь скакала быстрее, другой хочет, чтобы второй уровень был проще, а третий — чтобы батареи стоили дешевле.

Не используйте обратную связь, чтобы собрать предложения. Используйте ее, чтобы собрать информацию об опыте игрока.

Но предложения — это не то, что нужно дизайнерам при тестировании игр. Мы и сами можем придумать, как изменить игру. А вот чего мы не можем, так это испытать игру так, как это делает другой игрок. Мы настолько погружены в дизайн, что нам нужно мнение других людей. Вот почему настоящая цель тестирования состоит не в том, чтобы собрать предложения, а в том, чтобы понять опыт других игроков в игре.

Дизайнер часто может понять опыт другого человека, просто наблюдая за тем, как он играет. Я провел много тестов, и потом мне даже не приходилось задавать никаких вопросов. Все очевидно, когда смотришь, как у игроков получается или они проигрывают,

насколько быстро они идут дальше и какие решения принимают. Их выбор говорит за их внутреннее восприятие, где искать и что делать. Вы даже можете наблюдать за их лицами, чтобы понять, что чувствуют игроки.

Но иногда недостаточно просто наблюдать за игрой. В таких ситуациях, чтобы получить необходимую информацию, вам нужно задавать тщательно продуманные вопросы. Например, если вы хотите понять, что игроки посчитали важным, попросите их пересказать историю произошедшего. Если вам интересно узнать, поняли ли они что-либо, задайте им вопрос, который проверит их знания: «Сколько человек прибыло на вертолете?» Если они смутились, значит, они не заметили вертолет. Если вы хотите понять одно из принятых игроком решений, попросите его объяснить ход своих мыслей.

Будьте открытыми, ведите себя нейтрально и профессионально. Задавайте вопросы так, чтобы игрок считал, что вам действительно интересно получить ответ, а не думал, что вы его порицаете. Старайтесь избегать любое личное взаимодействие или эмоции, которые могут исказить или скрыть опыт тестирующего. Не поощряйте также и положительные или «правильные» ответы.

Если тестирующий начинает что-либо предлагать, попробуйте вернуться назад и выяснить, какой опыт мотивировал его на это предложение. Иногда необходимо спросить тестирующего, почему он это предлагает. В других случаях вы можете просто догадаться, что случилось. Возможно, он просит вас убрать винтовку потому, что у него был плохой игровой опыт именно с винтовкой, но вы можете задать дополнительные вопросы и выяснить, какой именно плохой опыт у него был. Он промахнулся там, где, как он считал, должен был попасть? Возможно, проблема в системе прицеливания. Чувствовал ли он, что у него слишком часто заканчиваются патроны? Экономика игры может быть несбалансированной. Возможно, он не понял особенности винтовки. Возможно, для этого нужен лучший обучающий материал или более понятный интерфейс.

Вы никогда по-настоящему не поймете баланс игры, наблюдая за результатами одного или двух тестов, не говоря уже о том,

если вы будете играть сами. Понимание основано на знаниях об опыте разных игроков и объединении их в общую ментальную модель функциональности игры.

Не думайте в рамках истории. Тестируйте игру в достаточном объеме, чтобы можно было построить ментальную модель ее функциональности как системы. Только так у вас будет ментальный контекст для принятия сбалансированных решений.

После одного игрового теста у вас появляется история. После трех у вас три истории. Однако после 10, 15 или 20 игровых тестов вы поймете, что ваше понимание игры начинает меняться. Вы больше не будете думать об историях. Вы начнете думать через призму систем и взаимоотношений. Ваша ментальная модель игры будет расти и развиваться и начнет включать в себя сотню новых деталей, причины и следствия. Вы сможете представить, как одно изменение повлияет на другие части игры и изменит их. Вы поймете суть игры — она в системе, а не в истории.

После того как вы поняли систему, вы можете серьезно задуматься о балансе. Вам не нужно думать о том, как изменения могут повлиять на три истории, которые вы видели в плейтесте. Вы поймете, как изменения влияют на систему в целом.

Так что не скупитесь на тестирование. Тестируйте на как можно большем количестве разных игроков. Впитывайте информацию и дайте своему разуму возможность построить ее модель. Только после этого у вас появится ментальный контекст для понимания всех последствий изменений, а не только недавних и самых очевидных. И только потом вы должны решать, что делать.

Много- пользовательская игра

Стрелок, преступник и шериф встретились на городской площади. Пролетело перекачиполе. Они потянулись к своим револьверам.

У стрелка была доля секунды, чтобы решить, кого застрелить — преступника или шерифа. У него не было времени рассмотреть, куда целился каждый из них, — может быть слишком поздно. Он должен решить сейчас. Но кого выбрать?

Ему заплатили за выстрел в шерифа. Так он и должен сделать.

Но постойте. Шериф знал, что стрелку заплатили, чтобы тот убил его. Следовательно, шериф выстрелит в стрелка в целях самообороны, что позволит преступнику свободно стрелять в того, кого он хочет. А у преступника было желание кровавой мести против стрелка, потому что стрелок шесть месяцев назад украл у него лошадь. Так что преступник, вероятно, тоже будет стрелять в стрелка. И стрелок знал, что преступник стреляет намного лучше, чем шериф. Поэтому, чтобы защитить себя, он выбрал преступника и надеялся, что шериф промахнется.

Шериф все это знал, а также знал, что стрелок попытается сначала остановить преступника. Тогда у шерифа будет возможность стрелять, в кого он захочет. Следовательно, он бы выстрелил в преступника, потому что это была его работа и еще потому, что не хотел, чтобы его уличили в драке с преступником, который стреляет лучше.

Но преступник знал об этом. И стрелок знал, что он это знает. И шериф знал, что они оба знают что-то еще. Стрелок снова и снова прокручивал в голове все эти схемы, в то время как его рука крепко сжимала пистолет.

Теория игр

Теория игр — это область математики, которая анализирует взаимодействие между движениями и контрдвижениями в многопользовательских играх. Несмотря на название, геймдизайнеры часто игнорируют теорию игр, потому что она кажется слишком абстрактной для реального мира. Хотя нам и не нужно вычислять точные цифры, как это делают математики, основы теории игр освещают ключевые концепции многопользовательского геймдизайна.

Подумайте о разнице между разрушением заброшенного замка и нападением на захваченный замок.

Разрушение пустого замка — задача из области физики. Возможно, придется подобрать лучшее место для установки подъемного механизма или лучший способ убрать завалы камней. Но при том, что эти задачи могут быть сложными, пустой замок не вспоминает о вас, он просто следует законам физики. Это похоже на однопользовательскую игру, так как разум одного игрока сталкивается с механической системой.

Разрушение замка с обороняющимися — совершенно другая история. Теперь есть два умных мозга, которые пытаются пре-
зволйти друг друга. Обороняющийся генерал заранее про-
считывает ваши шаги и нанесет ответный удар. Он откинет
лестницы, выстрелит огнем по тарану и пошлет наемных
убийц, чтобы убить вашего генерала. И будет прогнози-
ровать ваши ответные ходы. Он будет посылать ложные
сигналы, чтобы заманить вас в ловушку или попытаться
скрыть слабые места в стене. Теория игр описывает взаи-
модействие между вашим разумом и разумом соперника.

Теория игр помогает
анализировать ситу-
ации, когда игроки
должны предвидеть
и реагировать на ре-
шения друг друга.

Представьте, что однажды ночью во время осады вы выбираете стратегию на следующий день. Вы выбираете между атакой ворот тараном и подрывом стены саперами, а обороняющийся выбирает между горючей смолой, которую можно вылить сверху, и лучниками, которые будут стрелять в ваших саперов. Вы оба должны выбрать один вариант, и, поскольку готовиться нужно ночью, вы не будете знать, какой выбор сделал другой генерал,

пока не начнется бой. Платежная матрица в этом случае будет выглядеть следующим образом:

		Выбор обороняющегося	
		Расставить лучников на стенах	Подготовить горючую смолу у ворот
Выбор нападающего	Атаковать ворота тараном	Нападающий победил (прорвался через ворота)	Обороняющийся победил (поджег таран)
	Атакуют подрывники	Обороняющийся победил (застрелил подрывников)	Нападающий победил (взорвал стену)

Как видите, наилучшего выбора в данном случае нет.

И вы, и обороняющийся принимаете решение, исход которого зависит от решения другого; каждый может победить, спрогнозировав ход противника. Вы не выиграете эту битву, имея больший таран, но у вас есть шанс, если вы попытаетесь понять, как обмануть соперника, заставите его думать, что вы будете атаковать ворота, но на самом деле начнете готовить саперов. Эта игра уже не о стенах и стрелах. Она о порядке действий, предположениях, информации и уловках. Теория игр помогает нам понять именно такую ситуацию.

Теория игр — это не только про соревнования. Она охватывает любые взаимодействия между игроками, которые должны реагировать на действия друг друга. Соревновательные игры с нулевой суммой, в которых может выиграть только один игрок, относятся к одной из этих категорий, сюда же относятся коллективные игры, а также смешанные соревновательные/коллективные ситуации, в которых цели игроков частично совпадают. Даже атака на защищенный замок не является стопроцентной игрой с нулевой суммой, потому что одна сторона может сдаться или предложить перемирие.

Один классический пример теории несоревновательных игр описывает двух доисторических охотников, Тага и Бларга. Так как они живут в разных племенах, они не могут заранее договориться о том, кто на кого пойдет охотиться. Каждый должен выбрать, отправиться ли на охоту на оленей или на зайцев. Если оба решают охотиться на оленей, они могут скооперироваться и завалить одного оленя, и тогда оба будут сыты. На зайцев можно охотиться в одиночку, так что любой, кто выберет зайца, получит небольшой, но гарантированный обед. Но если один выберет охоту на зайца, а другой пойдет на оленя, то охотник на оленя останется голодным.

		Выбор Тага	
		Охота на оленя	Охота на зайца
Выбор Бларга	Охота на оленя	Оба наелись вдоволь	Таг поел мало. Бларг голодает
	Охота на зайца	Таг голодает. Бларг поел мало	Оба поели мало

Как и в битве за замок, выбор Тага зависит от выбора Бларга, и наоборот.

Процесс принятия решения одним пещерным человеком не столько связан с оленями и зайцами, сколько с прогнозированием решения другого пещерного человека. Каждый должен думать не только о том, насколько голоден он сам, но и о том, насколько голоден его противник и насколько голодным он себя считает и так далее.

Вспомните ментальный слой переосмысления навыков — психологическую интеллектуальную игру, похожую на покер, в которой каждый игрок пытается предсказывать ход мыслей противника и манипулировать ими. Это теоретико-игровое мышление. Если кто-то еще активно думает о вашей игре, она становится

чем-то большим, чем просто механика. Она становится линзой, через которую взаимодействуют два разума, и каждый пытается окутать другого в свою собственную ментальную модель. Для соперников игроков нет победы слаще.

Игры и стратегические взаимодействия

Прежде чем мы продолжим, я хотел бы уточнить одно определение. Специалисты по теории игр употребляют слово *игра* не так, как геймдизайнеры. Для них игра — это конкретное взаимодействие между стратегиями. Например, один раунд игры в «камень, ножницы, бумага» в теории игр считается игрой. В геймдизайне игра представляет собой целую систему механики, а не отдельную точку принятия решения.

Чтобы подчеркнуть разницу, я буду называть игры теории игр стратегическим взаимодействием. Игры, которые мы разрабатываем, генерируют множество стратегических взаимодействий, каждое из которых можно рассмотреть отдельно с помощью теории игр.

В этом простом вопросе определения может скрываться причина, по которой геймдизайнеры так часто игнорируют теорию игр. Теория игр не может просчитать целый раунд *Mortal Kombat*, поэтому многие сразу же считают, что анализировать игру бесполезно. Но это лучший способ проанализировать конкретное секундное взаимодействие между ударами, блоками или бросками. Теория игр не может охватить весь футбольный матч, но она может предсказать, куда игрок будет бить и в какую сторону прыгнет вратарь. Она работает только при применении на уровне взаимодействия между стратегиями, а не целыми дизайнами.

Равновесие Нэша

Основным понятием теории игр является равновесие Нэша.

Давайте разберемся.

Первая составляющая равновесия Нэша — это набор стратегий. Набор стратегий — это просто набор возможных выборов, которые могут сделать все игроки. Каждая ячейка в платежной матрице является набором стратегий. В примере с охотой на оленя,

где Бларг охотится на оленя/Таг охотится на зайца, — это набор стратегий. То же самое в примере, когда оба охотятся на зайца, Бларг охотится на зайца/Таг охотится на оленя и оба охотятся на оленя.

РАВНОВЕСИЕ НЭША —
это набор стратегий,
при которых ни один
игрок не может
улучшить свой соб-
ственный результат,
изменив только свою
стратегию.

Равновесие Нэша — это просто особая комбинация стратегий. В частности, это тот случай, когда ни у одного из игроков нет причин менять стратегию, если они предполагают, что никто другой этого не сделает. Звучит нелогично, но это отличная идея.

В примере с охотой на оленя существуют два равновесия Нэша. Первое — когда оба выбирают охоту на оленя. В этом случае оба получают наилучший возможный результат; если кто-то изменит свою стратегию и захочет охотиться на зайца, количество пищи, которую он получит, уменьшится. Второе — когда оба выбирают охоту на зайца.

Это более интересный случай, так как он раскрывает тонкость равновесия Нэша, которая заключается в том, что оно не обязательно оптимально для всех. Если оба охотятся на зайца, то они оба теоретически могут передумать и совместно отправиться на охоту за оленем, получив при этом гораздо больше еды. Но если передумает только один из них, он в конечном итоге останется голодным, в то время как другой будет жевать зайца. Итак, равновесие Нэша — это если оба пойдут охотиться на зайца, хотя это и не самый лучший вариант для любого из охотников.

Равновесия Нэша важны в реальных многопользовательских играх, потому что игра тяготеет к ним. Равновесия Нэша стабильны и самостоятельны, потому что ни у одного игрока нет причин делать что-то другое. Неравновесные конфигурации нестабильны и самомодифицируемы, поскольку у кого-то есть причина единолично изменить свою стратегию. Игра может допустить миллион комбинаций стратегий, но фактически будет присутствовать только равновесие Нэша. Таким образом, игровой опыт будет состоять из тех ситуаций, которые являются равновесиями Нэша — других вполне может и не существовать.

Вот почему так важно настроить игру так, чтобы у стратегических взаимодействий было большое количество или отсутствие чистых равновесий Нэша.

При одном равновесии все игроки точно знают, что делать, и у них нет никаких оснований ждать или даже думать о шагах друг друга. Это монотонность. Исчезает интеллектуальная игра, в которой каждый игрок пытается просчитать решения других.

Ситуации с множеством равновесий, таких как охота на оленя, лучше, потому что теперь каждый игрок думает о том, что будет делать другой игрок. Но мы можем улучшить даже это.

Лучше всего, если равновесие Нэша убрать полностью. Например, в битве у замка нет чистого равновесия Нэша. Независимо от набора стратегий одна сторона может добиться большего успеха, изменив свой выбор. Это хороший геймдизайн, потому что всегда понятно, что будут делать другие игроки, а это создает чувство предвкушения, обмана и манипулирования другими людьми. Таким образом, если в вашем стратегическом взаимодействии существует равновесие Нэша, измените дизайн или баланс, чтобы избавиться от него.

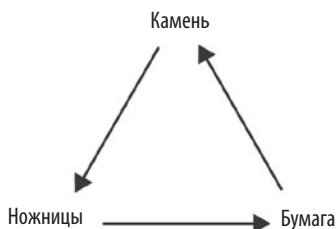
Стратегическое взаимодействие с одним равновесием Нэша в чистых стратегиях — испорченный геймдизайн, потому что он всегда будет находиться в том же равновесии. У каждого игрока только один жизнеспособный вариант, поэтому стратегическое решение исчезает.

Игры «камень, ножницы, бумага» и «орлянка»

Взаимодействия без равновесия Нэша обычно называют механикой «камень, ножницы, бумага», потому что камень — это самая известная игра без равновесия Нэша. В этой игре один игрок всегда хочет изменить свой ход независимо от набора стратегий. Платежная матрица игры выглядит так:

		Выбор игрока 1		
		Камень	Бумага	Ножницы
Выбор игрока 2	Камень	Ничья	Игрок 1 выиграл	Игрок 2 выиграл
	Бумага	Игрок 2 выиграл	Ничья	Игрок 1 выиграл
	Ножницы	Игрок 1 выиграл	Игрок 2 выиграл	Ничья

Но это легче представить так:

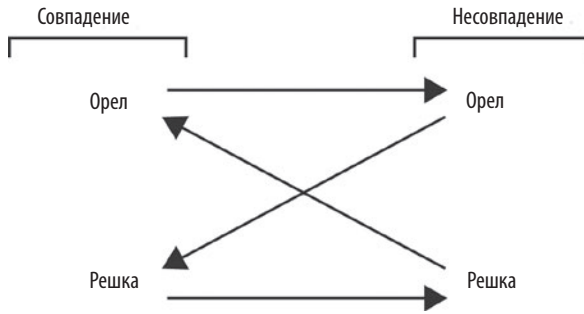


Подобная схема ходов и контрходов проявляется во многих формах в бесчисленных играх. В файтингах блок бьет панч, бросок бьет блок, а панч бьет бросок. В стратегиях копейщики бьют кавалерию, лучники бьют копейщиков, а кавалерия бьет лучников. Этот треугольник контрходов появляется снова и снова, так как это самый простой способ создать симметричную игру без равновесий Нэша.

Однако вопреки распространенному мнению, треугольник в игре «камень, ножницы, бумага» — не единственная базовая структура дизайна без равновесия Нэша. Вспомните битву за замок. Остается четыре хода, а не три. И у каждого игрока два варианта. Это не похоже на игру в «камень, ножницы, бумага», но, как вы видите, в этом случае равновесия Нэша тоже нет.

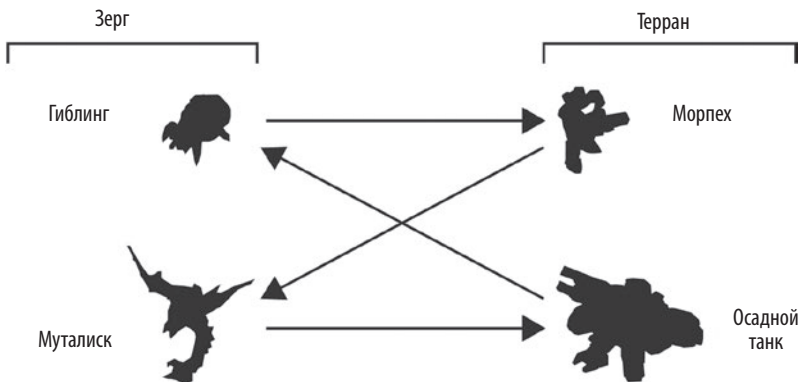
Паттерн игры «камень, ножницы, бумага» не образует равновесия Нэша для симметричных игр, в которых игроки ходят одинаково. Но в несимметричных играх, таких как битва за замок, мы используем другой паттерн, названный в честь другой старой игры — «орлянка».

В этой игре один игрок заявляет, что он ищет пару. Игроки не видят монетки друг друга. Каждый игрок кладет монетку орлом или решкой вверх. Затем они одновременно их раскрывают. Если монетки совпали, побеждает игрок, который искал пару. Если нет, побеждает его противник. Это выглядит следующим образом:



Хотя об этом часто и не говорят, но паттерн «орлянки» постоянно появляется в многопользовательских играх. Битва за замок — это аналогия «орлянки», потому что обороняющийся хочет, чтобы его оборона совпадала с вашим нападением, а вы хотите обратного. В многопользовательском шутере, когда вы защищаете объект в помещении с двумя дверями, вы играете в «сравнение монет»; вы хотите защитить дверь, через которую захочет атаковать соперник, а он, в свою очередь, хочет войти через другую дверь и выстрелить вам в спину. В стратегической игре World War II обороняющийся может решать, тратить ли ресурсы на установку мин, а нападающий может решать, тратить ли ресурсы на то, чтобы добывать тральщики. Вы не хотите расставлять мины, которые будут найдены, и вы также не хотите искать мины, которых нет.

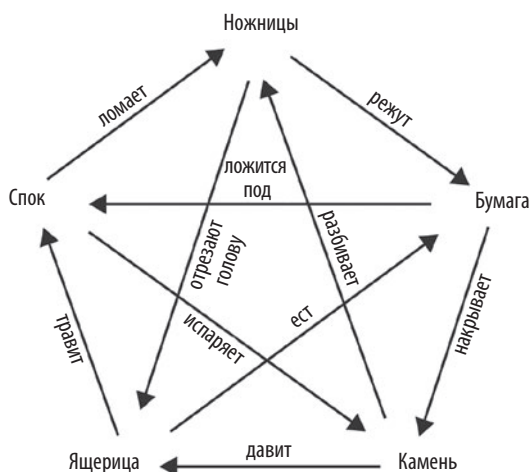
Давайте рассмотрим реальный пример схемы «орлянки» в гейм-дизайне. В StarCraft II бои между игроками Зергами и Терранами часто ведут четыре юнита: Осадные танки Терранов и Морпехи, а также Гиблинги и Муталиски Зергов. Они взаимодействуют так:



Муталиски летают, поэтому они автоматически поражают Осадные танки, потому что танки не могут стрелять в воздух. Сильнейший урон Морпехов мгновенно разрушает хрупких Муталисков. Гиблинг плавит плотно размещенные группы Морпехов, разбрызгивая кислоту. Но Осадные танки взрывают хрупких Гиблингов с безопасного расстояния. Многие раунды StarCraft II сводятся к повторяющимся взаимодействиям между этими четырьмя юнитами. По сети вы можете проиграть вариации этого паттерна сто часов подряд. Но игра никогда не становится скучной, потому что в ней нет равновесия Нэша, следовательно, у каждого игрока всегда есть возможность выиграть, просчитывая ходы своего противника или сбивая его с толку. Потому что на самом деле смысл игры не в управлении Морпехами и Муталисками. Речь идет о том, чтобы угадать мысли противника.

«Камень, ножницы, бумага» и «орлянка» — единственные элегантные паттерны дизайна для стратегических взаимодействий — «камень, ножницы, бумага» для симметричных игр, а «сравнение монет» — для асимметричных. Любая альтернатива это не более чем бессмысленное нагромождение новых стратегий.

Например, в игре «камень-бумага-ножницы-ящерица-Спок» каждый символ бьет два других:



Но RPSLS («камень-бумага-ножницы-ящерица-Спок») ничего не выигрывает с точки зрения интереса к решению в RPS («камень, ножницы, бумага»). Если равновесия Нэша нет, то его нет. Дополнительные символы никак не улучшают интеллектуальную игру предвкушения и хитрости. Дополнительные варианты могут сделать сюжет более интересным, но, скорее всего, это не стоит дополнительного бремени обучения.

Смешанные стратегии

До сих пор я говорил о том, что в таких играх, как «камень, ножницы, бумага», нет равновесия Нэша. Это было не совсем точно. В этой игре нет чистого равновесия Нэша. Но зато существует смешанное.

Например, в игре «камень, ножницы, бумага» нет чистого равновесия Нэша, но есть одно смешанное. Игра предполагает, что оба игрока будут показывать каждый ход в 33,3% случаев. Это равновесие Нэша, поскольку один игрок не может выиграть, отступив от этой схемы (поскольку если соперник показывает по чистой случайности, ваш ход не имеет значения, вы всегда выигрываете половину).

Другие стратегии в игре «камень, ножницы, бумага» не находятся в равновесии. Например, представьте, что вы должны были выбрать камень в 35% случаев, а ножницы и бумагу — в 32,5%. Теперь соперник может победить вас, изменив свою стратегию, выбирая бумагу в 100% случаев. Если один игрок может выиграть, изменив свою собственную стратегию, это не равновесие Нэша.

СМЕШАННОЕ РАВНОВЕСИЕ НЭША — это равновесие Нэша, где каждый игрок произвольным образом выбирает из набора стратегий с некоторым заданным набором вероятностей.

Смешанные равновесия легко решить в таких простых играх, как «камень, ножницы, бумага» и «сравнение монет», в которых один игрок сразу выигрывает, а другой проигрывает. Но это необычный случай. В большинстве реальных стратегических взаимодействий разные исходы дают разные результаты. Например, в файтинге блок бьет джеб, не нанося урона, джеб бьет бросок, нанося немного урона, а бросок бьет блок, нанося много урона. Это аналог версии игры в «камень, ножницы, бумагу», где вы получаете 1 доллар, если выигрываете, показав бумагу

или ножницы, и 5 долларов, если выигрываете, показав камень. Платежная матрица выглядит так:

		Выбор игрока 1		
		Камень	Бумага	Ножницы
Выбор игрока 2	Камень	Ничья	Игрок 1 выиграл 1\$	Игрок 2 выиграл 5\$
	Бумага	Игрок 2 выиграл 1\$	Ничья	Игрок 1 выиграл 1\$
	Ножницы	Игрок 1 выиграл 5\$	Игрок 2 выиграл 1\$	Ничья

Показывать каждый раз камень и надеяться на 5 долларов — наивная стратегия.

Проблема в том, что это предсказуемо. Противник может показывать только бумагу, и вы уйдете ни с чем. Чтобы хорошо играть в эту игру, нужна смешанная стратегия, когда вы произвольно выбираете между камнем, бумагой и ножницами. Но вы не можете разыгрывать ходы равномерно, как в классической RPS, иначе соперник начнет просто показывать камень чаще. Так как часто нужно делать каждый ход, чтобы заработать максимум?

Здесь вступает в игру математический аспект теории игр. Учитывая стратегическое взаимодействие и выигрыши, специалисты по теории игр могут рассчитать точное соотношение в смешанной стратегии, которое образует равновесие Нэша. Геймдизайнерам не нужно делать количественный анализ, но важно понять соотношение, поэтому я продемонстрирую это на реальном примере.

В футболе при пенальти мяч летит со скоростью до 55 м/с. На этой скорости мячу требуется примерно одна пятая секунды, чтобы преодолеть расстояние от ноги подающего к цели. Вратарю недостаточно этого времени, чтобы прыгнуть и попытаться не пропустить мяч после того, как по нему уже ударили. Единственный выбор — прыгнуть до удара по мячу. В то же время подающий должен выбрать цель, не зная при этом, куда прыгнет вратарь.

Это игра под названием «сравнение монет». Вратарь хочет не пропустить гол, а подающий, наоборот, забить.

В этой игре платеж подающего — вероятность того, что он забьет гол. Его шансы на выигрыш при правильной блокировке выше при ударе вправо, чем влево, шансы забить без блокировки также выше вправо, чем влево. Конечно, его выигрыш остается лучше при ударе влево и отсутствии блокировки, чем при ударе вправо и наличии блокировки. В остальном удар вправо будет представлять собой чистую стратегию равновесия, которая будет непреднамеренно использоваться постоянно.

Лучшая стратегия для подающего — произвольно чередовать удары вправо и влево в разных пропорциях, в большинстве случаев забивая вправо, иногда разбавляя их офсайдами, чтобы вратарь не расслаблялся. В то же время вратарь должен отражать его, большую часть времени блокируя удары вправо, а иногда и удары влево.

Мы можем вычислить точные пропорции на основании вероятности, с которой футболист забьет для каждой из четырех возможных комбинаций со стороны подающего/стороны вратаря. Однако во время исследований были получены данные о сотнях ударов в играх Европейской лиги, на основании которых составлена следующая таблица процентного соотношения шанса забить гол (которая фактически является платежной матрицей):

	Вратарь прыгает в верном направлении	Вратарь прыгает в неверном направлении
Нападающий придерживается верной стороны	шанс забить гол 63,6%	шанс забить гол 94,4%
Нападающий придерживается неверной стороны	шанс забить гол 89,3%	шанс забить гол 43,7%

Из этих цифр видно, что лучшая стратегия для подающего — забивать в правый угол 59,7%, а в левый — 40,3%. Эта смешанная стратегия дает общий шанс успеха на уровне 74,0% независимо от действий вратаря.

Любое отступление от этого соотношения позволяет вратарю улучшать свои результаты, выбрав чистую стратегию прыжков только вправо или влево. Например, если бы подающий забивал вправо 65% ударов, а не 59,7%, у него бы только 72,6% ударов

по воротам, при этом вратарь использовал бы свою стратегию прыжков только вправо.

Ключ к смешанному равновесию Нэша в том, что в равновесии каждый возможный ход имеет равный платеж.

Используя равновесную стратегию при любом ударе, нападающий рассчитывает на 74,0% вероятности попасть в цель независимо от того, в какую сторону была подача. Используя равновесную стратегию, на любом ударе подающий рассчитывает на то, что вероятность забить гол составляет 74,0% независимо от того, с какой стороны он бьет. Если бы он рассчитывал на большую вероятность для любой из сторон, вратарь чаще отвечал бы, прыгал на эту сторону, снижая коэффициент успешности для этой стороны. Это свойство равных платежей является полезным способом для понимания равновесных стратегий без математики. Просто посмотрите на пропорции, где каждый вариант приводит к одному и тому же среднему платежу.

Могут ли игроки реально проводить такой численный анализ? Нет, но в среднем среди многих игроков можно найти людей, которые замечательно справляются с поиском оптимальных смешанных стратегий на интуитивном уровне. Исследования реальных показателей футбольных подач показывают, что подающие в целом бьют справа почти полностью с правильной частотой.

Итак, неужели это все? Мы просто просчитываем платежи, определяем соотношение смешанной стратегии, и пусть игроки подбрасывают свои монеты? К счастью, нет. В реальных играх взаимодействие в теории игр является лишь основой для взаимодействия, ориентированного на человека, называемого *yomi*.

Yomi

Я позаимствовал термин *yomi* у Дэвида Сирлина, геймдизайнера файтингов, который, в свою очередь, взял его из японского слова «чтение» (чтение мыслей противника). Переворачивать монеты неинтересно, а вот пытаться читать мысли человека — очень интересно. Вот почему такие паттерны дизайнера, как «ка-

мень, ножницы, бумага» и «сравнение монет», являются лишь скелетом игры. Эмоциональную ценность игре придает именно плоть — yomi, умение заставить противника думать, что вы сделаете одно, а на самом деле вы сделаете другое, или позволить противнику думать, что он обманул вас, хотя на самом деле вы все поняли, или возможность пустить в ход такое оружие, о котором ваш противник и не догадывался. Это интенсивные и глубокие формы ментального конфликта.

Yomi работает, потому что реальный мир более неоднозначный, чем математика. В сухом математическом анализе исчисляется каждый платеж, а каждая стратегия четко отличается от других. Но настоящие игры не такие. Их исход невозможно определить количественно, стратегии нельзя аккуратно разместить в ячейки, а у игроков нет полной информации или доступа к генераторам случайных чисел.

Например, в шутере оба игрока должны решить, перезарядить ли оружие за углом, бросить гранату или подождать, держа дверь под прицелом, но они также могут в любой момент изменить свое решение или даже выполнить два действия одновременно. В игре жанра «стратегия» игроки выбирают экономические стратегии, которые плавно смешивают различные производственные задачи. В каждом случае существуют неоднозначные, не поддающиеся количественному определению границы вокруг взаимодействия в теории игр. Играющий в жанре «шутер» может смотреть и двигаться плавно во всех направлениях, играющий в стратегию может заказать продукцию тысячами разных способов. Эти размытые границы — то, откуда появляется yomi. Это то, что позволяет игрокам обойти границы математики, незначительно менять платежную матрицу каждого решения и узнавать чуть больше или чуть меньше о сопернике, чтобы опередить его в игре.

Таким образом, игра yomi зависит не только от создания стратегического взаимодействия без чистого равновесия Нэша, но и от создания системы с интересными размытыми границами вокруг этих взаимодействий. Давайте рассмотрим некоторые способы, с помощью которых мы можем генерировать подобное стратегическое взаимодействие.

YOMI — это интеллектуальная игра прогнозирования, хитрости и стремления переиграть соперника, чтобы получить выигрыш, выходящий за рамки математики теории игр.

Yomi требует, чтобы в игре были детализированные стратегии, которые можно смешивать и комбинировать сложными способами. Например, в StarCraft II игрок за Терранов может составить

Yomi растёт, если игроки могут плавно переходить от одной стратегии к другой.

войско из половины Осадных танков и половины Морпехов. Его не остановят ни войско из одних Муталисков, ни войско из Гиблингов. Для контратаки нужна аналогично смешанная армия Муталисков и Гиблингов. Терран может понемногу менять свое войско — плюс-минус один Морпех, плюс-минус один Осадный танк. Эти два игрока не играют в лабораторную игру, в которой нужно проверить

одну коробку и сравнить ее с другой, чтобы узнать результат. Они не просто забивают мяч влево или вправо. Они играют в ровном стратегическом пространстве, которое позволяет им выбирать из почти бесконечного множества промежуточных комбинированных стратегий. Каждая хорошо определенная стратегия требует аналогичной хорошо определенной реакции.

Хорошо определенные стратегии дают yomi, потому что они позволяют игрокам превосходить друг друга в нюансах своих стратегических знаний. Кому угодно можно сказать, что Морпехам противостоят Гиблинги, а Осадным танкам — Муталиски. Но только эксперты точно знают, что нужно, чтобы победить группу Морпехов и Осадных танков. Что будет, если у него пятнадцать Морпехов и один Осадный танк? Что будет, если у него восемь Морпехов и пять Осадных танков? Какое минимальное количество понадобится для уничтожения? Новичок этого знать не будет, а эксперт будет. Этот вид мелкослойного взаимодействия — то, что толкает потолок мастерства игры вверх.

У разных стратегий разные потенциальные платежи. На примере игры «камень, ножницы, бумага» мы рассмотрели вариант, в котором вы получаете 5 долларов, если бросили камень и победили,

Игра Yomi растёт из сложных, с трудом поддающихся количественной оценке платежей.

и 1 доллар — если бросили бумагу или ножницы и победили. Изменение этих показателей меняет пропорции стратегий, по которым должны играть игроки. И, как вы помните, правильные пропорции — это те, в которых у каждой отдельной стратегии одинаковый средний платеж.

Но что, если бы у нас не было этих значений? Что, если бы в стратегиях были потенциальные платежи с каче-

ственными эффектами, объединяющимися в несколько целей на разных уровнях достоверности? Эти платежи не известны заранее, и их нельзя описать одним значением. Определение хорошей смешанной стратегии перестает быть просто вопросом уравнивания платежей различных стратегий. Сначала мы должны выяснить, что такое платежи. Этот процесс вычисления поддерживает поток и повышает потолок мастерства.

Неопределенные платежи также означают, что нам нужно угадать, как другой игрок оценивает свои платежи. Если вы хорошо знаете своего противника, вы можете найти место, где он переоценивает или недооценивает определенные платежи, и использовать его, чтобы спрогнозировать его поведение и победить. На еще более высоких уровнях мастерства вы можете спрогнозировать его предположения о ваших оценках и так далее.

Вернемся к StarCraft II. Муталиски и Осадные танки стоят дорого, а Морпехи и Гиблинги — дешево. Это значит, что играющий за Терранов получит более высокий платеж за убийство Муталисков и Морпехов, чем за Гиблингов и Осадные танки, поскольку Муталисков сложнее заменить, а Морпехов легче построить. Это же правило работает и в обратном направлении: Зерги хотят уничтожить Осадных танков и Муталисков больше, чем Морпехов и Гиблингов.

Если бы все было так просто, это взаимодействие все еще можно было бы решить математически путем включения точного значения каждого вида персонажей. Но реальные последствия зависят не только от стоимости персонажей. Они зависят от того, как эта битва развивается через другие игровые системы, и ее контекста в более крупной игре.

Рассмотрим контекст. Если Осадные танки Терранов медленно приближаются к базе Зергов и почти достигли расстояния, с которого можно начать бомбардировку, Зергам крайне необходимо уничтожить атакующую силу любыми средствами. Если они этого не сделают, то им придется заплатить гораздо дороже, если бой случится где-то посередине карты. Если вы упустите что-либо на карте, ваши боевые единицы погибнут. Если вы упустите что-либо на месте, ваша база будет разрушена. Поэтому последствия могут быть разные.

Расстановка может изменить последствия. Рассредоточенные Морпехи и Осадные танки уязвимы для набрасывающихся на них Муталисков. Но сосредоточенные войска уязвимы к брызгам кислоты Гиблингов. Группа Морпехов, которая обычно уязвима для Гиблингов, может не пострадать, если она разбросана по обширной территории. В то же время эта разобшенная группа становится чрезмерно уязвимой для Муталисков, улучшая последствия.

Последствия также меняются в зависимости от количества единиц армии. Например, мы знаем, что Гиблинг обычно побеждает Морпеха. Но если Морпехов достаточно, они могут победить небольшое количество Гиблингов без потерь, стреляя в них до того, как те приблизятся. Точно так же, хотя Морпехи обычно отражают атаку Муталисков, большая группа Муталисков может уничтожить небольшую группу Морпехов без потерь. В этих случаях в результате отсутствия баланса в цифрах взаимоотношения в базовой игре в сравнении монет изменились. Но это изменение не происходит повсеместно — ни одно количество Гиблингов не сможет уничтожить один Осадный танк.

Изменение навыков влияет на последствия. Один игрок может мастерски играть Муталисками, улучшая последствия за счет этой стратегии. Следовательно, он захочет больше играть Муталисками, чем Гиблингами. Другой игрок захочет нанести ответный удар Морпехами, а не Осадными танками, так же как вратарь чаще прыгает на сторону, куда подающий забивает чаще.

Еще более простые игры представляют платежи, меняющиеся в зависимости от ситуации, хотя и в меньшем объеме. Например, в файтингах не так много переменных, как в StarCraft II. Присутствуют два персонажа, каждый из которых находится в определенном состоянии, и у ситуации не так много исходов. Тем не менее существует достаточно комбинаций позиций, состояния анимации, здоровья, энергии и других переменных, поэтому оценка платежа является сложной задачей для ума. Решение имеет меньшую значимость, чем в StarCraft II, но это допустимо, потому что оно сжимается до доли секунды. Сложность решения пропорциональна времени, доступному для его принятия, поэтому поток не прерывается.

При каждом взаимодействии игроки должны оценивать переменные и расчетные вероятности, чтобы составить набор оцененных платежей. В StarCraft II они должны учитывать количество боевых единиц, расположение, различия в навыках, статус экономики, экономическое влияние и сотни других количественных и не поддающихся количественному определению переменных. Задача высокой степени сложности, но в этом и состоит сила человеческой мысли. Машины не могут делать подобные расчеты. Только работающий на полную мощность человеческий мозг, используя все свои геометрические, пространственные, эмоциональные и межличностные познания, может решить эту проблему. Это тренировка всех частей, тренировка целого мозга.

И этот процесс оценки — не просто логическая головоломка. Это также эмоциональный вызов. Заурядные игроки, играющие за Зергов, будут паниковать и атаковать преждевременно, как только Осадные танки начнут обстреливать несущественную базу.

Начинающие Терраны по глупости оставят свои танки без присмотра, пока их Морпехи будут жадно преследовать Муталисков. В каждом случае эмоции этих игроков омрачают их оценку платежей. Зергом движет эмоциональный страх потери; матерые игроки знают, что иногда лучше потерять базу, чем рисковать армией, чтобы спасти базу. Терраном движет алчная жажда убить дорогого Муталиска; более подкованные игроки знают, когда их отпустить. Научиться хорошо оценивать — не значит просто знать и продумывать логику. Это значит оценивать свои эмоции, тренировать их и держать под контролем.

Эти сложные логические и эмоциональные задачи заставляют игроков снова и снова возвращаться и поднимают уровень мастерства до непостижимых высот. Эта глубина означает, что независимо от того, насколько вы хорошо умеете оценивать, всегда существует другой нюанс, который нужно обнаружить и учесть.

А вы знали, что существуют настоящие турниры по игре в «камень, ножницы, бумага»? Люди регистрируются, проходят отборочные туры и соревнуются на глазах у зрителей. Призы могут достигать 50 тысяч долларов. И все это в игре, где единственная равновесная стратегия — быть абсолютно случайным.

Звучит как шутка, и в некотором смысле так оно и есть. Но, тем не менее существует навык игры в «камень, ножницы, бумагу» против человека, потому что люди не могут генерировать случайные числа.

Уomi вырастает из психологии случайности.

В теории игр смешанные стратегии должны быть идеально случайными. Но так как люди не могут генерировать случайные числа, мы никогда не сможем играть в настоящие смешанные стратегии. Наш разум наполнен хорошо изученными вероятностными искажениями.

На просьбу написать случайную строку из 1 и 0 мы постоянно чередуем числа чаще, чем следует. Длинные повторения одного символа кажутся нам менее вероятными, чем они есть на самом деле. Если мы только что проиграли в стратегии, наше мнение о ее ценности в следующей игре изменилось. Мы можем использовать подобные ошибки в свою пользу. Например, вы можете играть в игру прогнозирования ошибок в мысленном генераторе случайных чисел других игроков. Если вы понимаете кого-то и подкованы в uomi, вы можете сказать, когда человек покажет камень, с вероятностью 35%, даже если он пытается показать с вероятностью 33,33%. И вы можете превратить эти знания в победу. Опять же, это толкает потолок мастерства вверх и улучшает решения.

Уomi растет от манипулирования информацией.

Принятие стратегических решений зависит от наличия полезной и полной информации об игре. Игроки должны знать, что происходит, что знает их соперник и что он думает и так далее. Обе стороны могут манипулировать всей этой информацией, если позволяет геймдизайн, создавая возможности для uomi.

Существуют три основных способа, которыми игроки могут манипулировать информацией.

Во-первых, игроки могут искать информацию, чтобы улучшить свои собственные решения. В стратегиях игроки отправляют разведчиков. В играх социального взаимодействия игроки спрашивают своих друзей. В шутерах они останавливаются и слушают шаги. Такие инструменты, как датчик сердцебиения в Modern Warfare или возможность сканирования в StarCraft II, обеспечивают выполнение подобных действий.

Во-вторых, игроки могут отказаться предоставлять кому-либо информацию. В стратегиях они убивают разведчиков. В играх социального взаимодействия они вмешиваются в разговоры других людей. В шутерах они бросают дымовые шашки или антирадары. Игроки могут даже скрыть метаигровую информацию — известно, что участники турниров как по StarCraft, так и Street Fighter экономят свои лучшие ходы в ранних раундах только для того, чтобы реализовать их в финале.

Наконец, игроки могут предоставлять ложную информацию, чтобы обмануть друг друга. В то время как другие виды yomi предполагают знание мыслей противника, обман — это контроль над ними. Иногда механика обмана может быть явной, как в стратегических фантомных войсках или в играх социального взаимодействия. В других случаях игроки найдут способы использовать для этого обычную механику. Например, в шутере игрок может открыть дверь, чтобы создать ложное убеждение в том, что он проходит через нее, но по факту войти через окно. В стратегической игре игрок может построить аэропорт, позволить противнику разведать его, но так и не создать самолет, при том что противник инвестирует в бесполезное противовоздушное оружие.

Эти элементы дизайна манипулирования информацией можно расширить для создания слоев ложной информации. Один игрок может заставить другого игрока думать, что он знает то, чего на самом деле не знает. Или он может заставить другого игрока думать, что он не знает факта, который на самом деле знает. Этот вид многослойного обмана сложно разработать напрямую, но это ценный результат, который может появиться в случае взаимодействия основных инструментов обмана.

Трудно в уме смоделировать мысли другого игрока. Наблюдение за намерениями и привычками двух или трех человек находится на грани возможного. Увеличьте количество игроков до четырех и более, и это уже будет невозможно. Поэтому yomi ограничивается двумя-тремя игроками, так как получается слишком много чужих мозгов, о которых нужно думать. При большом количестве

Для Yomi не нужно большое количество игроков.

игроков игра теряет свой интеллектуальный аспект и становится только механикой.

Очевидный способ решить эту проблему — сохранить небольшое количество игроков. Но часто это не выход. В этих случаях игра все еще может создать *uomi*, на время изолируя небольшие группы игроков от основной массы.

Например, в *World of Warcraft* играют миллионы игроков, но они всегда взаимодействуют в небольших, ограниченных группах. Четыре игрока могут отправиться в рейд в подземелье, чтобы получить добычу, два игрока могут провести поединок. Специальная игровая механика не позволяет разным группам сталкиваться друг с другом. Таким образом, *uomi* может расти внутри группы, пока она изолирована.

Изоляция также может происходить с помощью более мягких и эмергентных механизмов. Например, во многих шутерах игроки объединяются в большие команды по 16 и более игроков. Но не каждый игрок постоянно сражается. Скорее они рассредоточены по большой карте со множеством изолированных комнат. Бои начинаются и заканчиваются в одном помещении, и в них не участвует большое количество игроков. Это позволяет игрокам сосредоточиться только на тех, кто присутствует в комнате, что делает *uomi* возможным, даже если на поле будут 32 игрока.

Yomi. Кейс: *modern Warfare 2*

Давайте рассмотрим пример игроков, использующих *uomi*, в многопользовательском шутере *Modern Warfare 2*. Этот пример охватывает несколько секунд игры между Кэрл и Дейвом в режиме *Team Deathmatch*.

Кэрл внутри маленького сарая с двумя входами, она видит приближающегося Дейва. Она отступает дальше вглубь, чтобы оба игрока не видели друг друга.

Начинается теоретико-игровой мыслительный процесс, запускающий *uomi*. Эти игроки начали играть в игру «сравнение монет»,

поскольку каждый выбирает между двумя входами в сарай. Кэрол может видеть один вход, а Дейв хочет атаковать через вход, на который Кэрол не смотрит, и подходит к ней сзади.

Если бы это было все, игроки могли бы рассчитать соотношения смешанной стратегии, перевернуть монеты и продолжить. К счастью, чтобы игра была интересной, в Modern Warfare 2 множество нечетких границ, образующих yomi вокруг своей основной игровой механики.

Вот как выглядят мысли Кэрол, пока она обдумывает свои варианты.

Она может либо остаться в сарае, либо выбраться из него через любую дверь. Если Кэрол решит остаться:

- Она может сосредоточиться на одной двери. Если Дейв войдет в эту дверь, он попадет прямо под ее прицел и она выстрелит, мгновенно убив его. Если Дейв войдет через другую дверь, он выстрелит ей в спину, прежде чем она успеет обернуться. Но за какой дверью ей следить?
- Она может решить следить за дверью, через которую увидела Дейва. Дейв может это спрогнозировать и пойти со спины. Она может это спрогнозировать и так далее.
- Она может решить следить за дверью, которая ближе к Дейву, затем, не видя его, переключиться на другую дверь. В этом случае она опирается на тот факт, что Дейв еще не атаковал, эта информация указывает на то, что он мог выбрать другой путь. Опять же, Дейв мог предугадать данную цепочку рассуждений и действовать вопреки ей.
- Она может смотреть то на одну дверь, то на другую. Это защита. Такое наблюдение означает, что она увидит Дейва почти сразу после того, как он войдет. Недостатком является то, что Кэрол сама себя дезориентирует и нарушает прицел, постоянно переводя взгляд с одной двери на другую. Даже если Кэрол будет видеть дверь, когда в нее войдет Дейв, ее прицел, скорее всего, будет не совсем точным, в отличие от ситуации, если бы она постоянно целилась в одну дверь.

- Она может решить постоянно перемещаться в сарае. Опять же, это защита. Перемещение делает Кэрол более трудной целью, поэтому Дэйву, скорее всего, будет сложнее убить ее, даже если он войдет в дверь, пока Кэрол не видит. Недостатком этого решения является то, что это нарушает прицел Кэрол, а также может создавать звуки шагов, по которым Дэйв сможет узнать о присутствии Кэрол, ее местонахождении и активности.
- В любое время Кэрол может перестать ждать и целиться снаружи.

Если Кэрол решит целиться из сарая, она может или столкнуться с Дейвом, или не столкнуться:

- Если она встречает Дейва, уomi-игра заканчивается, так как оба игрока находятся в поле зрения друг друга и никакой существенной скрытой информации больше нет. Игра превращается в задачу «застрелить-увернуться». Первоначальное преимущество в этой стрельбе зависит от того, насколько хорошо Дейв спрогнозировал ход Кэрол.
- Если по прогнозу Дейва Кэрол выйдет из сарая через правильную дверь, он, возможно, просто встанет перед дверью и наведет на нее пистолет. В этом случае Кэрол идет прямо на него и почти наверняка умирает.
- Если по прогнозу Дейва Кэрол выйдет из сарая, но он не знает, через какую именно дверь, Дейв может просто находиться поблизости, пытаясь наблюдать за обеими дверями, не концентрируя внимание только на одной. Эта стратегия защиты приводит к относительно справедливому соревнованию по стрельбе, так как ни один из игроков не целится в другого.
- Если, выйдя из сарая, Кэрол не сталкивается с Дейвом, у нее несколько вариантов в зависимости от того, что, по ее мнению, делает Дейв.
- Она может обойти сарай и попытаться подойти к Дейву сзади, когда он подходит к другой двери. Если Дейв замешкается, Кэрол, возможно, сможет подойти сзади и выстрелить ему в спину. С другой стороны, если Дейв дошел до двери быстро, он, возможно, уже заглянул внутрь и обнаружил,

что Кэрол вышла через первую дверь. Теперь он может либо развернуться и ждать, пока Кэрол обойдет сарай, вероятно, после этого убив ее, либо решить, что Кэрол сбежала через другую дверь, и в этом случае он пробежит через сарай и выйдет с другой стороны.

Кэрол может попытаться убежать:

- Если бы Дейв быстро обошел вокруг сарая, возвращаясь, он бы выстрелил в спину Кэрол, когда она убегала.
- Если по прогнозам Дейва Кэрол сбежит, он, возможно, вообще не подходил к сараю и просто прятался на расстоянии, и в этом случае он застрелит Кэрол.
- Если Дейв замешкался на подходе к другой двери, Кэрол сможет сбежать.

На первый взгляд Modern Warfare 2 выглядит как игра о стрельбе в людей. Вот почему ее называют шутером. А стрельба здесь наиболее безобидное слово. Целься, атакуй, взрывай и бросай. Хорошее бессмысленное веселье, правда?

Но если бы это было все, игроки быстро бы устали. Стрельба — это просто зрелище. Оно не долговечно, но в Modern Warfare 2 оно есть. Люди играют в эту игру сотни или даже тысячи часов. Они покупают сиквел за сиквелом, чтобы делать одно и то же снова и снова. Почему?

Потому что на самом деле эта игра не о стрельбе. Все дело в yomi, это сложный танец интеллектуальной оценки и переоценки. Опытные игроки действительно продумывают разветвленные возможности в реальном времени. Это то, что переживает их разум, он не заиклен на повторяющейся графике стрельбы. Это далеко не бессмысленная игра рассчитана на интеллектуалов.

Основная причина, по которой Modern Warfare 2 так хорошо создает yomi, заключается в том, что ее оружие чрезвычайно точное и смертельное. Игроки умирают за один-два выстрела. Если вы подошли к кому-то сзади, он умирает до того, как у него появится шанс повернуться и выстрелить. Это означает, что значимые решения — это не те решения, когда игроки стреляют друг в друга, а те, которые принимаются до того, как игроки увидят друг друга.

Вы хотите быть тем игроком, который подходит к противнику сзади, а не наоборот. И уомі процветает, когда игроки не видят друг друга, изо всех сил стараясь услышать шаги, наблюдая за датчиками сердцебиения, прислушиваясь к выстрелам и угадывая пути атаки врагов. Если нет полной информации, появляется масса возможностей для обмана.

Этого не происходит в шутерах с более слабым оружием. Первый выстрел не стоит того, чтобы потратить 25 выстрелов и кого-то убить. В таких играх неважно, что делают игроки, когда не видят друг друга. Важно то, насколько хорошо они целятся и уклоняются во время боя. И хотя прицеливание и уклонение от атак являются сложными задачами механики, в них нет уомі, так как отсутствует значимая скрытая информация.

Modern Warfare 2 на самом деле создает гораздо больше нюансов и разнообразных уомі, чем дерево решений в предыдущем примере. В реальной игре у каждого игрока больше опций и информации, они являются более нечеткими и более тонкими, чем здесь описано. У игроков могут быть специальные инструменты, которые могут помочь им собирать или скрывать информацию, например оглушающая граната, дымовые шашки, способность передвигаться бесшумно, датчики сердцебиения и так далее. Любой игрок может рассеивать пули, выдавая свою позицию, но надеясь на попадание. Любой игрок может выстрелить через пролет, чтобы попытаться дать ложные сигналы. Игроки могут общаться с членами своей команды или просто ждать и надеяться на помощь от союзника, который оказался неподалеку. У игроков могут быть различные типы оружия, поэтому любой из них может иметь преимущества или недостатки в любом типе столкновения — Кэрл может предпочесть ближний бой и автомат, а Дейв — сражаться на расстоянии и винтовку. В ориентированной на цель игре один или оба игрока могут хотеть не только победы. Если тикает таймер, проигрывающий игрок не будет использовать стратегии, которые отнимают ценные секунды; другой игрок может спрогнозировать это и тянуть время. У игроков может быть разный общий уровень мастерства и способности в определенных навыках, например в прицеливании, движении или непосредственно в уомі, и каждый может знать о другом игроке либо много, либо ничего. Игроки могут быть из-

мотанными, возбужденными, рассеянными или расстроенными. Соберите все это в кучу, и вы поймете всю сложность решений, насквозь пронизанных уомі.

И это только один выбор, продолжающийся одну или две секунды. Игра будет продолжать генерировать эти тактические головоломки тысячами, каждая из которых перетекает в следующую, никогда не повторяется и не дает игроку почувствовать скуку. Она поддерживает поток путем непрерывного обмана, который преподносится через простую, тщательно продуманную механику.

Разрушительное поведение игрока

Большинство игр задают игрокам цели: набрать наибольшее количество очков, победить соперника или выжить как можно дольше. До сих пор мы предполагали, что на самом деле игроки достигают этих целей только потому, что мы им так сказали. Но иногда они этого не делают.

В однопользовательской игре, когда игроки преследуют цели, которые не ставил перед ними дизайнер, деск-джампинг дает свои результаты. Однопользовательский деск-джампинг может навредить опыту, но обычно это не смертельно. Однако в многопользовательской игре придуманные игроком цели могут разорвать игру на части, потому что эти цели затрагивают каждого, а не только того, кто их придумал.

Многопользовательские игры обычно хорошо структурированы и хорошо сбалансированы, поэтому у каждого игрока определенная роль. Один игрок, делающий странный выбор, нарушает этот баланс и разрушает игру для всех, включая тех, кто хотел играть нормально. Например, если группа героев сражается с драконом и один из сражающихся решает отлучиться на трапезу или вылечить дракона просто из любопытства, тогда вся команда проигрывает.

Существуют два основных вида деструктивного поведения игрока: разные цели и разные навыки.

Разные цели

Я уже писал о достоинствах Modern Warfare. Теперь пришло время взглянуть на один из ее недостатков: разные цели. Офи-

циальная цель каждой многопользовательской миссии — победить. Каждый тип игры устанавливает различные критерии победы. Например, в игре Capture the Flag цель заключается в том, чтобы за ограниченное количество времени захватить больше флагов, чем команда соперника. Проблема в том, что игра еще и отслеживает количество убитых каждым игроком врагов и со-

РАЗНЫЕ ЦЕЛИ появля-
ются, если игроки
в многопользова-
тельской игре решают
добиваться целей,
которые нарушают
опыт других игроков.

храняет эту информацию в базе данных для постоянного хранения данных. Уничтожение врагов всегда приятнее, чем наблюдение за счетчиком в углу экрана. В результате некоторые игроки предпочитают игнорировать официальные цели, в данном случае захват флага, и сосредоточены исключительно на уничтожении врагов. Это не вредит их опыту, но мешает опыту других игроков. Их союзники вынуждены играть с теми, кто не будет захватывать цель, а их соперники столкнутся с нечестной игрой.

Единственная причина, по которой дизайн Modern Warfare все еще держится на плаву, заключается в том, что уничтожение врагов обычно помогает в достижении командных целей. Игрок, который не думает о захвате флага, продолжает помогать, если он убивает врагов. Существует другая цель, но она все еще достаточно тесно связана с официальной целью, поэтому игра в основном работает. Конечно, это вредит игре, но не уничтожает ее полностью. Если бы другая цель заставила игроков действовать полностью против официальных целей, то игра бы развалилась. Например, Left 4 Dead — это игра о четырех выживших, помогающих друг другу пережить наступление зомби. Игра разработана таким образом, что команда достигает наибольшего успеха, если действует сообща. Но иногда один игрок может отступить и захотеть проверить, как далеко он может пройти один, просто чтобы бросить вызов самому себе. Но это разрушает игровой опыт членов его команды. Он получает непроходимый, но интересный бой, который и хотел, а его команды лишаются его помощи и тщательно сбалансированная игра расстраивается.

Даже реальные действия, такие как выход из многопользовательской игры, являются формой разных целей. У игрока, выходящего из игры, цель встать и заняться чем-то другим. Цель оставшихся игроков — продолжить играть.

Но худшие виды отличных целей — это те, которые основаны на удовольствии вызывать у других раздражение. Это называется вредительством.

В шутере игроки перекрывают вход, чтобы их команда не могла покинуть стартовую зону. В стратегиях игроки атакуют своих союзников или блокируют свои базы. Даже в играх, в которых нет очевидного способа причинить вред другим игрокам, поиск способов вредительства становится непосредственно самой игрой. Игроки World of Warcraft накладывали заклинание «хождение по воде» на союзников, когда те падали с высоты в бассейн. Несчастные жертвы падали в воду, которая для них внезапно оказывалась твердой, как бетон. Игроки Counter-Strike использовали функцию граффити, чтобы размещать ужасно оскорбительные изображения в высокопроходимых зонах уровня. Другие игроки, пытаясь сражаться в тактических битвах, были вынуждены смотреть самые шокирующие картинки интернета (а ведь такой тип сопоставления не улучшает опыт).

ВРЕДИТЕЛЬСТВО — это сознательное разрушение чужого игрового опыта ради собственного развлечения.

В очных играх, например в настольных, спортивных состязаниях или бильярде, люди не занимаются вредительством из-за угрозы разрыва дружеских отношений или драки в баре, поэтому разработчикам этих игр не нужно беспокоиться о том, что игроки будут перемещать чужие фишки или фигуры или выбрасывать бильярдные шары в окно. Но этих социальных сдерживающих факторов нет в интернете, где каждый прячется за анонимностью и может уйти в любое время. В таких играх дизайн самой игры должен учитывать возможность вредительства.

Первая линия защиты от всех разных целей такая же, как и при деск-джампинге: мотивируйте игроков иметь осмысленные цели. Игроки не придумывают себе разные цели случайно; они делают это в ответ на те же импульсы, которые в первую очередь заставляют их играть в игры. Они хотят побеждать, исследовать, общаться и влиять на людей. Если лучший способ получить это — сыграть в игру по тем правилам, какие были заложены, то они так и сделают.

К сожалению, добиться идеального совпадения целей сразу для всех игроков практически невозможно. Обычно нам нужны другие меры для поддержания многопользовательских игр.

Большое количество игроков является буфером против разных целей. Если в команде из двух человек один из них отлынивает, выходит из игры или занимается вредительством, это гораздо хуже, чем если то же самое происходит в команде из 12 человек.

Иногда конкретные решения могут прояснить конкретные вопросы. Например, в *Left 4 Dead* могут играть только четыре игрока, а кампании длятся 45 минут. У четырех незнакомых между собой людей, играющих непрерывно в течение этого времени, не такие большие шансы на победу; существуют разные цели, когда один игрок хочет выйти, а другие хотят продолжить играть. Но игра не прерывается, потому что как только игрок уходит, его персонаж управляется ИИ, который управляет им, пока не присоединится другой игрок. ИИ не в состоянии полноценно заменить реального человека, но это вполне действенная стратегия.

Проблемы вредительства нуждаются в более радикальных решениях, чем другие отличные цели, потому что если они возникают, то наносят гораздо больший ущерб.

Один из очевидных вариантов — сделать вредительство невозможным. Если в шутере игрок блокирует двери, разрешите игрокам проходить сквозь друг друга. Если в игре ММО игроки заманивают монстров, чтобы атаковать других игроков в городе, обеспечьте безопасные зоны, куда монстры не пойдут.

Эти строгие ограничения необходимы, чтобы избежать худших случаев, но, к сожалению, мы не можем запретить все возможные стратегии вредительства. Всегда остается некоторая лазейка, даже если это предполагает намеренное поражение или отказ от игры. В стратегиях можно просто спрятать свои боевые единицы по углам и даже не пытаться выиграть. Игрок командного шутера может произвольно стрелять из своего оружия, чтобы указать местоположение своей команды. Возможности безграничны.

Но мы не должны учитывать каждый отдельный случай. Стратегии вредительства можно распределить по шкале степени серьезности в соответствии с их развлекательной ценностью для вредителей и их разрушительным воздействием на других игроков. Самые серьезные стратегии очень интересны для вредителей и разрушают игру для игроков. Стратегии, в которых

проблемы отсутствуют, — это стратегии, которые либо не приносят удовольствия вредителю, либо не причиняют вреда другим игрокам. Чтобы игра нормально работала через интернет, разработчики должны выявить наиболее серьезную стратегию вредительства и решить проблему. Если степень серьезности ниже определенного уровня шкалы, бессмысленно перекручивать всю игру, чтобы не допустить вредительства. Например, стратегии остаются интересными для игроков, потому что просто не сражаться вредителю не весело и это не вредит другому игроку, который получает легкую победу (если было скучно). И игроки в шутер могут навредить своей команде, совершив самоубийство, но они этого не делают, потому что смерть — не весело. Эти стратегии вредительства ниже того уровня, когда следует начать беспокоиться. Пока мы запрещаем самые жесткие стратегии вредительства, игра работает.

К сожалению, не все игры можно защитить от вредителей изнутри. Этим играм нужны последние линии защиты в виде механики обеспечения порядка, например системы голосования или модераторы игр. Такие системы неаккуратны и неэлегантны: для голосования необходимо прервать процесс игры, чтобы игроки могли подвести итог собственному опыту, а оплата модераторам стоит разработчикам денег. Но при использовании в качестве последней линии защиты эти системы могут сохранить игру, которая в противном случае превратилась бы в болото заблокированных дверных проемов, игнорируемых целей и непристойного контента.

Разные навыки

Один особый вид разных навыков — тот, который появляется из-за значительной разницы в навыках игроков. Когда один игрок является экспертом, а другой — новичком, могут происходить не самые приятные вещи.

Цель неопытного игрока — изучить игру и не находиться под слишком сильным давлением. Цель опытного игрока — сыграть и выиграть глубокую игру, повышающую уровень мастерства. Различие между этими целями может приводить к конфликтам, которые неприятны для обеих сторон. Неопытный игрок нахо-

дится под давлением, его оскорбляют. Опытному игроку либо скучно играть против ужасного (по его мнению) противника, либо его раздражает бесполезный член команды.

Это весьма распространенная проблема, потому что она возникает везде, где игроки взаимозависимы, а успех зависит от мастерства. Командные сражения Left 4 Dead, StarCraft II и версия зомби-нацистов в Call of Duty: Black Ops — все это отличные дизайны, и все они страдают от проблемы разных навыков.

Существует несколько способов уменьшить влияние разных навыков. Прежде всего, сократить непосредственно саму разницу в навыках.

Очевидно, что простой элегантный дизайн — лучший способ сокращения разных навыков. Игра, которую легко понять, поможет игрокам быстрее преодолеть границу мастерства, поэтому игроки меньше раздражают экспертов и наслаждаются игрой.

Еще один способ уменьшить различия в навыках — создать алгоритм поиска, который собирает игроков с похожими уровнями мастерства. Система отслеживает уровень опыта каждого игрока и коэффициент выигрыша. Когда игрок ищет игру, алгоритм отправляет его к другим игрокам с похожими показателями. Это концептуально простое решение, но на практике разработка и проектирование этих систем может стать серьезной проблемой.

Бывают и структурные решения. Массовые многопользовательские игры, такие как World of Warcraft, настроены так, что игроки могут играть в одиночку первые несколько часов, не испытывая давления со стороны других игроков, а после того, как они изучат основы, влиться в группу. Иначе говоря, предполагается, что начинающие игроки никогда не должны играть какую-либо роль в командной игре, откуда следует, что они не смогут навредить команде. Аналогичным образом во многих играх как одиночный, так и многопользовательский режим. Прежде чем погрузиться в многопользовательскую игру, игроки сначала учатся в однопользовательском режиме, где их неудачи ни на что не влияют. Это означает, что большинство пользователей, играющих по сети, не будут абсолютными новичками, даже если они не эксперты.

Наконец, игры могут уменьшить разные навыки благодаря хорошему адаптивному обучению. Например, в Left 4 Dead используется адаптивная система обучения, которая сообщает начинающим игрокам алгоритм действий в каждой критической ситуации. Когда зомби-охотник запрыгивает и отключает союзника, на экране мигает сообщение с указанием спасти своего друга. Это происходит при каждом важном действии в игре — выполнение задач, помощь союзникам и получение ресурсов. Таким образом, хотя у новичка может быть плохая цель или сырая стратегия, его, по крайней мере, не оставляют в полной неопределенности.

Если мы не можем уменьшить разницу в навыках, мы можем сделать ее менее важной, а игроков — более независимыми друг от друга. Многие считают взаимозависимость между игроками исключительным благом, но это не так. Поскольку взаимозависимость может создавать чувство общей победы, она также требует, чтобы мы страдали из-за ошибок других людей.

Наилучшим решением является система, которая устраняет общий проигрыш, оставляя при этом общую победу. Например, в таких шутерах, как Halo: Reach, существуют режимы командного боя, в которых игроки сражаются в командах, в основном игнорируя членов своей команды. Но иногда два или три игрока формируют спонтанный альянс, побеждают врага и разбегаются. Так создается командная работа, но только на необязательной, краткосрочной основе, что обеспечивает общую победу, не усиливая общее поражение.

Многие наивные проекты основаны на предположении о том, что игроки будут работать сообща.

И если это предположить, то чудеса кажутся возможными. Но по-настоящему сплоченные игроки с хорошими навыками редко встречаются. Игра, которая на это рассчитывает, долго не протянет. Многопользовательский дизайн должен быть достаточно устойчивым, чтобы справляться с постоянным хаосом, причиной которого являются уход игроков, вредительство, отсутствие ключевых навыков или решение играть не по правилам.

Мотивация и удовлетворение

Игровое кресло Джуди разомкнулось, автоматически удаляя нейрокомпьютерный интерфейс из ее черепа. Она села. В комнате было темно, но Джуди все еще могла разглядеть часы на стене. Было 31 декабря 2151 года.

Прошло девять лет с тех пор, как она начала играть. За это время Джуди завоевала империи, вырастила семьи и построила башни до небес. И все в вымышленной компьютерной вселенной. Она бы осталась в системе, но что-то пошло не так. Мультиплеер вышел из строя. Некому было похвастаться, не с кем создавать что-либо. Она искала глобальную игровую сеть и не смогла соединиться ни с одним другим игроком. Она терпеть не могла выходить из компьютерного мира, но пришлось это сделать, чтобы исправить неполадку.

В комнате было темно и пыльно. Оглядевшись вокруг, она увидела игровое кафе с рядами стульев, простирающихся в правую и левую стороны. Внимательно присмотревшись к ним, она смогла увидеть их «обитателей». Все они были старыми, седыми, в старческих пигментных пятнах. Все были мертвы.

Джуди на секунду задумалась и откинулась на спинку стула. Он заключил ее в стальные объятия, нежно погружая в бессознательное состояние.

Дофаминовое удовольствие

Большинство из нас считает, что мы хотим чего-то потому, что это приносит нам удовольствие. На первый взгляд это мнение даже не похоже на убеждение. Конечно, мы хотим удовольствий. А как иначе?

Когда ученые впервые начали изучать удовольствие, казалось, результаты исследований подтвердили такое мнение. Еще в 1950-х годах Джеймс Олдс, на тот момент исследователь

в Университете Макгилла в Монреале, хотел посмотреть, как воздействие электричества на мозг влияет на поведение. Чтобы выяснить это, он вживил электроды в мозг крысы и соединил ее с рычажком внутри клетки. Когда крыса нажимала на рычажок, через провод поступал импульс электричества прямо в головной мозг.

Сначала крыса просто бродила по клетке. Но как только она впервые случайно нажала на рычажок, ее поведение изменилось. Она стала нажимать на него все чаще и чаще. Со временем она превратилась в наркомана, бесконечно жаждущего новой дозы самостимуляции.

Эксперименты Олдса подтвердили гипотезу наличия в мозге центров удовольствия. При его активации центр удовольствия вырабатывает химическое вещество — дофамин. Обычно центр удовольствия генерирует дофамин в ответ на стандартные стимулы, такие как вкусная еда, выигрыш в лотерею или взгляд на привлекательного партнера. Электроды Олдса усиливали эту естественную реакцию.

В 1960-х годах этот эффект начали исследовать у людей. Первым был Роберт Хит, заведующий кафедрой психиатрии и неврологии Университета Тулейн в Новом Орлеане. В своем самом скандально известном исследовании Хит пытался «вылечить» гомосексуала при помощи стимуляции зон удовольствия. Молодой человек, известный под кодовым названием В-19, вырос в неблагоприятной домашней обстановке, имел мало друзей и находился на грани самоубийства. Хит подключил к нему провод и передал ему выключатель. По словам Хита: «В-19 стимулировал себя до такой степени, что как в поведенческом, так и в интроспективном плане он испытывал почти всепоглощающую эйфорию и восторг, а сессию нужно было прервать, несмотря на его энергичные протесты».

Эксперименты Олдса и Хита закрепили дофаминовую теорию мотивации: мотивация определяется стремлением к удовольствию, а предвестником удовольствия является дофамин. Их разумные рассуждения продержались много лет.

Но затем начали появляться проблемы.

Дофаминовая мотивация

Во-первых, поток дофамина. Исследователи обнаружили, что дофамин не появляется одновременно с получением награды, он также не появляется позднее. Скорее дофамин предшествует награде.

Если дофамин — это удовольствие, это не логично. Идя в магазин за стейком, вы испытываете чувство, отличное от тех ощущений, когда едите его. Но ваш мозг все еще напитан дофамином, даже если вы просто смотрите на мясные ряды в супермаркете.

Этот странный результат стал стимулом для последующих исследований, в которых ученые пытались понять реальную роль дофамина.

В 2009 году 61 участник эксперимента оценил свое желание провести отпуск в разных странах мира. Некоторым из них дали L-Dopa, препарат, который усиливает действие дофамина, а другим дали плацебо. Если бы дофамин и был удовольствием, испытуемые, которым дали L-Dopa, чувствовали бы себя так же, как и при стимуляции мозга через электропровод. Но этого не произошло. Вместо этого получившие усиленный дофамин люди хотели отправиться в большее количество стран и хотели действовать более интенсивно. Они не чувствовали радости. Они чувствовали себя мотивированными.

В 1989 году Кент Берридж, профессор нейробиологии в Мичиганском университете, дал крысам нейротоксин, который уничтожил все их клетки, принимающие дофамин. Крысы перестали совершать действия, даже питаться. Они потеряли всякую мотивацию и без посторонней помощи умерли бы от голода. Но когда Берридж впрыснул сладкую жидкость им в пасть, по выражению их мордочки можно было понять, что они испытывают удовольствие. Даже без дофамина крысы могли наслаждаться едой. Они только что потеряли мотивацию, чтобы снова ее получить.

Как оказалось, старая теория, которая еще вчера считалась здоровой, была неверна. Дофамин не является маркером удовольствия, это маркер мотивации. И эти два понятия не всегда взаимосвязаны.

Звучит странно, но если присмотреться — нас повсеместно окружают подобные примеры. Наркоманы все больше хотят получить дозу, но наркотики нравятся им все меньше и меньше. Мы благодарны другу за то, что он вытащил нас на вечеринку, на которую мы не хотели идти. А многие игроки оказались не в состоянии отказаться играть даже после того, как игра стала для них скучной.

Мы можем хотеть чего-то, но не любить это или любить что-то и не хотеть этого.

Большая часть этой книги посвящена тому, как вызывать эмоциональные переживания, которые удовлетворяют игроков. В этой главе мы рассмотрим механизмы, которые создают только мотивацию. Мы рассмотрим, как их использовать, как сочетать их с удовлетворением, а также этические последствия отказа от них.

Предвкушение награды

Каждая игра должна использовать дофаминовую мотивацию, потому что каждая игра создает моменты недовольства. Часто эти моменты важны для дизайна. Без дофамина игроки сдались бы после первой же неудачи или потери ресурса. Дофамин мотивирует игроков преодолевать препятствия, чтобы они могли восторжествовать, получить социальную связь или насладиться графикой.

Множество видов наград может вызвать дофаминовую мотивацию. Еда, вода, секс, безопасность, деньги, имущество, власть и социальный статус — все это награды, за которые отвечает дофамин. Если мы находимся в предвкушении получить эти вещи, мозг вырабатывает много дофамина, а наша мотивация достижения цели растет.

Основным способом, которым мы генерируем дофаминовую мотивацию, является создание предвкушения награды.

Но секундочку. Игры почти по определению нематериальны. Как правило, мы не можем дать игрокам еду или секс (хотя несколько игр действительно используют порнографию в качестве награды). Обычно мы не можем предложить реальную физическую безопасность (за исключением тех случаев, когда игра может навредить игроку, как в PainStation). Обычно мы не можем дать реальные деньги или реальный социальный статус (если только игроки не делают ставки или играют на аудиторию).

Большинство игр должны работать так, чтобы создать мотивацию, ничего не предлагая и ничем не угрожая. Но как это сделать, если мы не можем предложить никаких реальных наград?

Мы можем сделать это, потому что человеческий мозг развивался в среде, в которой не было современных игр, поэтому у него нет системы, позволяющей отличить реальные награды от виртуальных. Виртуальные награды обычно не обладают той же мотивационной интенсивностью, что и реальные, но в них сохраняется часть эмоций реальной награды. Хотя в реальной жизни мы хотим разбогатеть больше, чем в игре, мы все же хотим разбогатеть и в игре.

Некоторые виртуальные награды, такие как очки опыта, игровые деньги, оборудование и способности, связаны с прокачкой персонажа игрока. Другие, такие как кат-сцены или аудиозаписи, используют контент истории в качестве награды. Некоторые, такие как баллы в аркадах 1980-х годов, являются не более чем статистикой достижений, цифровым эквивалентом того, как ребенок царапает свое имя на самой высокой ветви самого высокого дерева в своем районе. И неважно, что эти награды бессмысленны вне игры. Они все равно щекочут центр удовольствия.

План подкрепления

При создании мотивации сама награда часто не самая важная часть уравнения. Реальный ключ к мотивации заключается в том, когда именно игроки ожидают и получают награды. Это определяется режимами подкрепления.

Психолог Б. Ф. Скиннер разработал режимы подкрепления в 1930–1940-х годах, исследуя идею оперантного обусловливания. В то время как классическое обусловливание заключается в манипулировании непроизвольными реакциями (как в знаменитом эксперименте с собакой Павлова), оперантное обусловливание заключается в манипулировании явно добровольным поведением через награды и наказания. Давая собаке лакомство после того, как она выполняет трюк, вы используете оперантное обусловливание, чтобы заставить ее снова выполнить этот трюк.

Режим подкрепления — это система правил, которая определяет, когда выдается награда.

Чтобы изучать оперантное обусловливание, Скиннер изобрел так называемый «ящик Скиннера». Это небольшой прибор, в который можно поместить крыс или голубей. В зависимости от эксперимента в ящике могут быть рычажки, колеса для хомяков, датчики, лампочки, громкоговорители, диспенсеры для пищи или электрошокеры. Эти устройства взаимосвязаны друг с другом невидимым механизмом.

Например, в одной конфигурации диспенсер будет выдавать пищу при каждом нажатии на рычаг. В другом он будет выдавать еду при каждом десятом нажатии. Или же он будет случайным образом выдавать еду в 10% случаев нажатия. Каждая из этих конфигураций предоставляет разный режим подкрепления.

Скиннер хотел посмотреть, как различные режимы подкрепления будут влиять на поведение крысы. Как отреагирует крыса, если она будет получать еду, каждый раз нажимая на рычаг? Что, если она получит порцию случайно, или по таймеру, или после того, как пробежит определенное количество кругов в колесе? Скиннер обнаружил, что поведение животного меняется в зависимости от того, когда появляется награда, но не всегда очевидным способом. Даже незначительные изменения в режимах подкрепления могут приводить к резким различиям в поведении.

Такие режимы подкрепления предназначены не только для ящиков Скиннера. Их полно в играх. Побеждая орка, чтобы получить золотой слиток, вы играете в режимы подкрепления, созданном геймдизайнером. Как и у крыс, за которыми наблюдал Скиннер, детали этого режима подкрепления будут влиять на мотивацию продолжать играть. Для геймдизайнеров эти режимы подкрепления являются инструментами, которые можно использовать, чтобы мотивировать игроков.

Существует бесконечное множество возможных режимов подкрепления, но два наиболее важных из них — фиксированное соотношение и переменное соотношение.

Фиксированное соотношение

В случае с режимом подкрепления с постоянным соотношением награда дается в фиксированном соотношении с предпринятыми

действиями. Например, в ящике Скиннера крыса может получать порцию еды каждый раз, когда она нажимает на рычаг. Игрок может получить золотую монету каждый раз, когда он побеждает орка. Оба режима подкрепления являются фиксированным соотношением, потому что соотношение между действием и поощрением всегда одинаково.

Соотношение не обязательно должно быть один к одному. Например, игрок может получить 10 золотых монет за каждого десятого орка, которого он победит. В этом случае большинство действий не вознаграждаются вообще. Десятый орк приносит награду за предыдущие девять орков.

Сами по себе режимы с фиксированным соотношением являются плохими мотиваторами. Они поощряют длительные периоды низкой активности с последующими всплесками активности, когда игрок решает, какую награду он хочет. Эти длительные периоды низкой активности позволяют игроку легко встать и уйти.

Переменное соотношение

Режим подкрепления с переменным соотношением подобен режиму подкрепления с постоянным соотношением, за исключением того, что соотношение изменяется каждый раз, когда выдается награда. Обычно при режиме с переменным соотношением награда дается после некоторого случайного набора действий. Например, у игрока может быть 10% шанса получить 10 золотых монет каждый раз, когда он побеждает орка. В зависимости от того, как ляжет карта, он сможет либо получить награду три раза подряд, либо зарубить 50 орков и ничего не получить.

Режим подкрепления с переменным соотношением является наиболее эффективным простым режимом. Даже если он дает такую же среднюю отдачу, что и режим подкрепления с постоянным соотношением, он мотивирует совершенно иначе. У игрока, столкнувшегося с режимом с переменным соотношением, всегда остается шанс получить большую награду при следующем действии, поэтому активность остается высокой и последовательной. Вы всегда надеетесь, что следующий орк принесет большой платеж.

В играх режимы подкрепления с переменным соотношением встречаются повсеместно. В некоторой степени они существуют в любой игре со значимыми случайными элементами, и они лежат в основе всех азартных игр. Они являются главным предметом игры в некоторых ролевых играх, так как за каждого монстра и за каждый квест предлагается разный набор случайных наград с разной степенью вероятности.

В жизни они тоже появляются. Возможно, вы встречались с кем-то по такой схеме: в некоторые дни эти люди милы и приветливы, в другие — холодны и ведут себя отстраненно. Они не перезванивают и забывают о назначенной встрече. Словом, они разыгрывают неприступность. Люди, которые ведут себя по тому же принципу, что и игровые автоматы, привязывают к себе. Это просто еще один режим подкрепления с переменным соотношением.

Другие режимы подкрепления

При режиме подкрепления с *постоянным интервалом* награда становится доступной через определенный период времени после того, как ее заработали. В ящике Скиннера крыса может получить еду, только если нажмет на рычажок минимум через 15 секунд с момента получения последней порции. В классических играх со смертельным боем здоровье может появиться не раньше, чем через 60 секунд после того, как игрок взял его.

При режимах подкрепления с постоянным интервалом у игроков высокая мотивация, в случае если награда доступна, и низкая, если награда недоступна. Например, они будут игнорировать аптечку до тех пор, пока она практически не будет готова появиться снова, затем начнут проверять чаще, пока она не появится, после чего они хватают ее и снова некоторое время игнорируют. Такие пробелы в мотивации делают режимы подкрепления с постоянным интервалом плохими мотиваторами.

Режим подкрепления с вариативным интервалом делает награду доступной случайное количество времени после получения. Этот режим создает непрерывную мотивацию, но с меньшей интенсивностью, чем график с переменным соотношением. В обоих

случаях за каждую попытку получить награду игрок теоретически что-то получает. Но в режиме подкрепления с вариативным интервалом шансы тем ниже, чем чаще игрок проверяет награду, поэтому у него меньше причин повторять действие.

Режим дифференциального подкрепления низкой частоты реакции подобен режиму подкрепления с постоянным интервалом, однако если игрок пытается забрать награду слишком рано, интервал перезапускается. Это усиливает низкую, равномерную активность.

В режиме с дифференциальным подкреплением высокой частоты реакций, чтобы получить награду, игрок должен выполнить определенный набор действий в течение заданного интервала времени. Например, вы должны победить пять врагов за одну минуту.

Существует множество других режимов подкрепления, при этом у каждого свой характерный паттерн мотивации.

Совмещенные режимы подкрепления

Большинство режимов подкрепления в определенные моменты приводят к спаду мотивации. Это является проблемой, потому что они создают окно, когда игрок может решить прекратить игру. Чтобы устранить эти пробелы в мотивации, игра может запустить несколько режимов подкрепления одновременно.

Например, режим подкрепления с постоянным соотношением создает провал в мотивации сразу после того, как игрок полу-

чает награду. Если он может получить 10 золотых монет за каждого десятого побежденного орка, его мотивация упадет сразу после уничтожения десятого орка, потому что игрок знает, что ему нечего ждать от следующих девяти орков. Именно в этот момент игрок убирает игру обратно на полку — на жаргоне дизайнеров это называется «момент полки».

Мы можем избежать такого «момента полки», совместив несколько режимов подкрепления с постоянными соотношением. Подумайте, что происходит, когда игрок может получить доллар за каждый десятый сундук, ал-

Сила режимов подкрепления не в каком-либо одном режиме, а в их совмещении таким образом, чтобы всегда был хотя бы один, который вызывает высокую мотивацию.

маз — за каждый десятый добытый камень и стрелу — за каждого десятого убитого гоблина. К тому времени, когда игрок открывает десятый сундук, он уже добывает девятый камень и хочет покончить с этим, чтобы получить следующий алмаз. К тому времени, когда он получает алмаз, он уже убивает девятого гоблина. К тому времени, когда он убивает десятого гоблина, чтобы получить стрелу, он уже открывает семнадцатый сундук и так далее. Всякий раз, когда один режим приводит к провалу в мотивации, другие режимы находятся на пике мотивации. Игрок переключает фокус между действиями, никогда не пропуская удары, вызванные дофамином.

Так называемые RPG с гриндом известны этой техникой. Игроку всегда остается всего лишь несколько шагов до получения следующего трофея, повышения уровня героя или мастерства. Каждый раз, когда он получает одну награду, он обнаруживает другую, которая находится всего в нескольких минутах ходьбы. Конечным результатом является то, что игрок не может остановиться.

Этот же механизм работает в пошаговых стратегиях, таких как *Civilization V*, и делает их захватывающими и качественными. В этих играх появляется синдром «еще одного хода», поскольку в них так много совмещенных режимов наград, что игроку никогда не остается более одного или двух ходов до получения своего приза. На следующем этапе наконец будет закончено исследование этой технологии. Один ход после этого — это новая военная единица. Затем новая структура, расширение границы и так далее. Одновременно могут выполняться 30 и более таких режимов подкрепления; по крайней мере один из них всегда поддерживает высокую мотивацию.

Ключом к совмещению режимов подкрепления является то, что игроку нельзя позволять концентрировать свои усилия только на одном режиме. Если игрок может эффективно игнорировать все режимы, кроме одного, он может по очереди отказаться от них и оказаться в ситуации, когда каждый режим находится на максимальном расстоянии от следующей награды. Это неизбежный «момент полки». RPG избегают этого, создавая свой мир так, чтобы у игрока постоянно была неиссякаемая возможность

убивать гоблинов, открывать сундуки и добывать камни, даже если игрок занимается чем-то другим. В стратегиях игроки не могут объективно сосредоточиться только на одной исследовательской или производственной задаче, исключив все остальное. Это сохраняет постоянную мотивацию, а режимы остаются десинхронизированными.

Эмергентные режимы подкрепления

До сих пор мы рассматривали режим подкрепления с явной схемой разработки. Дизайнеры создают эти режимы так же, как Скиннер создал свои ящики, с математической точностью решая, что именно получит игрок после каждого действия. Но этот вид эксплицитно определенного режима подкрепления является исключением, а не правилом.

Например, в шахматах игрок будет бить фигуру каждые X минут. Не потому, что дизайнер задал переменную X . Скорее интервал X

Большинство режимов подкрепления имеют неявную схему разработки. Скорее они появляются из игровых систем более низкого уровня.

возникает естественным образом из взаимодействия механики игры более низкого уровня. В шахматах темп захвата фигур очень острый — игроки долго выстраивают перекрестные защитные позиции, а затем уничтожают их в серии захватов. Поскольку размен фигур может произойти в любом ходе, шахматы представляют эмергентный режим подкрепления с переменным соотношением. И если бы правила изменились, изменился бы и этот режим.

Игроки реагируют на эмергентные режимы подкрепления точно так же, как и на режимы с явной схемой разработки. Часто они представляют собой разницу между игрой, которая захватывает вас, и игрой, которая заставляет вас уйти.

Например, во многих шутерах существует режим многопользовательской «мясорубки». Цель смертельного боя — убить как можно больше других игроков. В конце матча игрокам показывается количество убитых ими персонажей. Это количество является наградой, потому что так приятно видеть высокий счет убийств и неприятно — низкий. На языке наград каждый матч похож на побежденного орка, за исключением того, что он «выплевыва-

ет» убийства вместо денег. Но в то время как золото, которое игрок получает за уничтожение орка, определяется алгоритмом, количество убитых врагов вытекает из боевого дизайна игры. Различные карты, тюнинг оружия и системы устройства матчей дают разные паттерны количества убийств.

В некоторых играх со смертельным боем количество убийств весьма предсказуемо. Хороший игрок всегда совершает много убийств в отличие от плохого. Это создает эмергентный режим подкрепления, который напоминает фиксированное соотношение.

Как и все режимы подкрепления с постоянным соотношением, он создает пробел в мотивации сразу после получения награды (сразу по окончании матча). После окончания матча игроки не обязаны начинать новый матч, потому что точно знают, что они получат, а также знают, что им нужно закончить весь матч, чтобы это получить. Подобные режимы «мясорубки» не очень популярны из-за своего однообразия. Здесь нет великих побед или сокрушительных потерь. Просто матч за матчем, который каждый раз предлагает одинаковую предсказуемую награду.

Другие игры с режимами «мясорубки» гораздо более непредсказуемы. Игрок может прийти последним в одном матче и первым в следующем. Навык влияет на результат, но один удачный тактический прорыв или несколько удачных выстрелов могут превратить поражение в блистательную победу или наоборот. Это напоминает режим подкрепления с переменным соотношением и дает аналогичные мотивационные результаты. Игроки знают, что любой матч может стать тем самым, в котором им выпадает удачная серия из 30 убийств, поэтому они всегда мотивированы играть.

Если внимательно посмотреть, то вы увидите эмергентные режимы подкрепления повсюду. Многие игроки однопользовательской игры завершают игру в начале уровня, сразу после того, как они закончили предыдущий (разрыв в мотивации с фиксированным соотношением). Хорошие игры-головоломки позволяют игрокам запускать несколько головоломок одновременно, потому что они могут решить одну из них в любой

момент (совмещенная переменная соотношения). И игроки будут пробовать пройти один и тот же уровень Super Meat Boy 300 раз подряд, потому что каждая попытка может привести к победе (переменное соотношение).

Внешняя и внутренняя мотивация

Режимы подкрепления выглядят так просто. Кажется, все, что нам нужно сделать, — это создать пару совмещенных режимов подкрепления, раздать золотые звезды, очки или какую-то другую дешевую награду, и все получилось! Мгновенная мотивация делать все, чего только пожелает дизайнер.

В этом есть доля правды. Вот почему школы боевых искусств выдают цветные пояса, бойскауты и девочки-скауты раздают значки, а армии — медали. Эти системы вознаграждения работают, но у них также скрытая цена.

Внешнее поощрение — это те награды, которые находятся вне самой деятельности. Например, золото, которое дается за победу над орком, является внешним источником мотивации, потому что золото отделено от непосредственной борьбы с орком. Внутренние награды неотделимы от деятельности. Если победа над орком хороша сама по себе, у игроков будет внутренняя мотивация побеждать их даже без какого-либо вознаграждения.

Внешнее поощрение
может вытеснить
и даже разрушить
внутреннее наполне-
ние игры.

На первый взгляд кажется, что нужно просто соединить эти два вида мотивации. Если игроку нравится побеждать орков, то добавление небольшого количества золота, кажется, только сделает лучше. Но этого не происходит. Исследования показали, что внешняя мотивация может исказить, смещать и даже разрушать внутреннюю мотивацию. Создание орка, разбрасывающего золото, уменьшает естественное желание игроков сражаться с ним.

В одном исследовании психолог Эдвард Дечи разделил участников на две группы. В первый день обе группы решали сложную задачу, такую как головоломка или игра. На второй день одна группа получила поощрение за задание, например деньги или

бесплатное питание, а другой группе разрешили продолжать играть без вознаграждения. На третий день Дечи отменил поощрение. Группа, которой никогда не платили, продолжала радостно работать над этой задачей. Но группа, которой платили, потеряла интерес, когда награды не стало. Казалось, что после получения вознаграждения за задание эта группа решила, что задание не стоит того, чтобы выполнять его бесплатно. Внешние платежи вытеснили внутренний интерес.

Многие другие исследования изучали вариации на эту тему. Дети создают менее интересные рисунки, если им предлагают награду. Шахматисты решают меньше шахматных задач, когда им за это платят. Поэты пишут менее интересные стихи, когда им напоминают, что писатели могут зарабатывать деньги.

Когда мне было 11 лет, мои родители наняли учителя фортепиано. В течение пяти лет я покорно занимался полчаса ежедневно и изучал этюды, которые задавал преподаватель. Затем, когда мне исполнилось 16 лет, учитель перестал приходить. Но, вопреки собственным ожиданиям, я не переставал заниматься. Я много практиковался, намного больше, чем раньше. Иногда я играл сразу по три часа. И я стал играть по-другому. Вместо того чтобы просто дойти до той цели, которая была поставлена для меня, я свободно изучил инструмент. Я играл песни, которые были слишком сложными или слишком легкими или странных жанров. Я сочинял и импровизировал. Я играл, потому что хотел, и этот внутренний огонь был намного сильнее, чем любой внешний толчок, который мои родители или учитель могли бы дать мне.

Последствия искажения мотивации в результате получения награды зависят от того, насколько интересна задача. Внешний источник мотивации хорошо работает применительно к скучным задачам, потому что внутренняя мотивация, которую можно сместить, отсутствует. Если вам заплатят за то, чтобы вы рыли ямы, это вовсе не означает, что вам нравится копать, потому что вам это никогда не нравилось. Только когда задача интересна, возникает эффект.

Вы не любите копать не потому, что вам платят за то, чтобы вы копали ямы, а потому, что вам это никогда не нравилось. Этот эффект появляется только в том случае, если задача интересна. Чем интереснее задача, тем лучше результат.

Больше всего страдают исследовательские или творческие задачи. Если за них предлагается вознаграждение, люди перестают изучать систему с интересом. Они переключаются на выполнение минимума, чтобы получить поощрение в виде сладостей. Для игры, основанной на свободе или творчестве, это подобно смерти.

Для объяснения внешних эффектов демотивации были предложены многие психологические механизмы. Возможно, люди используют ментальную эвристику, которая говорит, что поощрение должно быть работой, и поэтому не стоит делать ее самостоятельно. Возможно, люди отталкиваются от того, что они чувствуют, когда другие пытаются их контролировать. Или, возможно, поощрение заставляет нас мысленно классифицировать взаимоотношения как торговые переговоры, а не как свободные и добровольные.

Каким бы ни был механизм, эти выводы представляют важность для геймдизайнеров. Они означают, что мы не должны волея-неволей разбрасываться вознаграждением в каждом игровом опыте, надеясь на повышение мотивации легким способом. Бессистемное использование обесценивает вознаграждение, искажая и разрушая основной игровой опыт. Игрок может быть мотивирован, но мотивация — это оболочка экшена без ядра чувства.

Корректировка вознаграждения

Теперь мы знаем, насколько важна дофаминовая мотивация в играх, чтобы игроки преодолевали препятствия в игровом опыте. Но теперь мы также знаем, что внешняя мотивация разрушает внутренний опыт игры. Получается замкнутый круг: вы должны мотивировать, но если вы это делаете, игра становится пустой. Что делать в данном случае?

Ключом к созданию наград, которые не разрушают игровой опыт, является *корректировка вознаграждения*.

Принцип корректировки вознаграждения в основном говорит о том, что вам следует выдавать вознаграждение только за то, что игрок уже хочет сделать. Чем ближе мы можем расположить структуру вознаграждения с внутренними желаниями игрока, тем менее разрушительным она будет для основного опыта. В лучших случаях система вознаграждений в точности соответствует внутренним желаниям игрока и два вида мотивации действительно соединяются.

Некоторые игры естественным образом предрасположены к системе вознаграждений, потому что легко определить, когда игрок достиг своих целей. Например, гонки должны поощрять игроков за самое быстрое время, потому что в этом и заключается весь их смысл. В гонках это возможно, потому что в них можно с легкостью точно определить время заезда, кроме того, в гоночной игре не так много других мотиваций, которые могут быть разрушены таймером.

В других играх сделать корректировку вознаграждения сложно или невозможно. Как в игре SimCity мы вознаграждаем игроков за создание города, похожего на их родной? Как мы награждаем игроков в коллективной игре за то, что они нашли нового друга? Какие очки, значки или призы мы даем игрокам за изобретение уникальной топильной камеры¹ в Dwarf Fortress? Проблема этих мотиваций заключается в том, что игра не может определить, когда цели были достигнуты. А если мы не можем это определить, значит, мы не можем и вознаградить игроков. Вот почему невозможно скорректировать систему вознаграждений с творческими, исследовательскими, социальными игровыми системами, подобными этой. В таких играх единственное решение — не использовать систему вознаграждений, потому что любая такая система скорее уничтожит мотивацию, чем создаст ее.

КОРРЕКТИРОВКА
ВОЗНАГРАЖДЕНИЯ —
это степень того,
насколько действия,
поощряемые системой
вознаграждений,
схожи с действиями,
без которых игрок
оставался бы вовле-
ченным в игру.

¹ Топильная камера (drowning chamber) не является встроенной в игру системой обороны. В Dwarf Fortress игроку предоставлена полная свобода действий. И сообщество игроков разработало некоторые схемы, которые может применить любой человек. Таким образом, игрок, который не воспользовался готовой схемой, а изобрел свою или модифицировал существующую, получит наиболее сильное эмоциональное вознаграждение. — *Примеч. ред.*

Но большинство игр находятся между этими двумя крайностями. В них остается место для вознаграждений, но сложно разработать систему вознаграждений, которая бы точно отражала внутренние желания игрока. В этих случаях дизайн наград становится вопросом мастерства.

Например, игра про катание на скейтборде Skate 3 использует систему баллов, чтобы вознаграждать игроков за выполнение трюков. Не так-то просто присвоить каждому трюку счет. Skate 3 позволяет игрокам объединять трюки в чрезвычайно сложные

Цель дизайна наград состоит в том, чтобы создать систему, которая может об-
наруживать и соот-
ветствующим образом
вознаграждать игрока
за все, что он уже
хочет сделать. По-
скольку каждая игра
уникальна, то нужда-
ется в собственной,
продуманной системе
вознаграждений.

комбинации прыжков, флипов и гриндов. Этот элегантный дизайн позволяет производить бесчисленное множество различных комбинаций. Задача состоит в том, чтобы получить баллы за эти движения, которые соответствуют тому, насколько впечатляюще они выглядят.

Дизайнеры Skate 3 подошли к этой проблеме основательно. Они создали систему подсчета очков, которая отмечает каждый поворот, флип, грайнд и прыжок и считает каждую миллисекунду и каждый сантиметр. Используя ряд коэффициентов для учета длины трюковой комбинации, она точно оценивает эффект любой комбинации трюков.

Так как она настолько хорошо сбалансирована и детализирована, действия, которые вознаграждаются, практически идентичны действиям, которые игрок хотел бы выполнить, если бы система вознаграждений отсутствовала.

И эта система прекрасно работает. Играть в Skate 3 — весело, гнаться за высокими баллами — всегда хочется и вызывает восторг.

Если бы корректировка вознаграждения в игре была хуже, скажем, если бы система поощряла игроков только за прыжки через колеблющиеся плавающие кольца, система вознаграждений вступала бы в противоречие с творческой игрой. Игроки игнорировали бы все игровые системы, кроме тех, которые помогали бы им пройти через кольца. Элегантная система трюков могла бы остаться, но она была бы безжизненной из-за плохо скорректированной системы внешних наград.

Но подождите, трюки — это не единственное, что игроки хотят делать в Skate 3. Что будет, если они захотят поучаствовать в гонках? Или увидеть, насколько сильно они могут травмировать своего персонажа или усовершенствовать одну конкретную комбинацию трюков? Игрок, который просто изучает игру, может решить, что он хочет сделать что-либо из перечисленного, но система подсчета очков сама по себе не может определить эти действия.

Чтобы справиться с такими нестандартными целями, Skate 3 включает ряд специальных режимов, которые задают нестандартные условия победы. Некоторые из них — гонки. В других вы должны точно скопировать комбинацию трюков другого игрока. А в соревнованиях Hall of Meat вы пытаетесь сломать как можно больше своих костей в одной ужасной аварии. Эти режимы достигают корректировки вознаграждения на основании этих нестандартных целей.

Угрызения совести игрока

Существуют игры, в которых геймдизайнеры даже не пытаются добиться корректировки вознаграждения. Весь дизайн таких игр основан на создании мощной, непрерывной мотивации независимо от стоимости основного опыта. Несмотря на то что игрок может продолжить играть, это также может привести к угрызениям совести игрока.

Когда я впервые начал делать игры, я никогда не думал, что мне придется столкнуться с этическими вопросами в своей работе. Но тогда я не знал силы режимов подкрепления.

Мозг древнего человека не имел развитой защиты от тщательно настроенных режимов подкрепления. Наши дофаминовые триггеры развивались, чтобы справляться с охотой и поиском пищи, а не с игровыми автоматами. Таким образом, игры, которые приводят к выработке дофамина, могут подтолкнуть людей к тому, что кажется иррациональным и саморазрушительным.

Угрызения совести появляются у игрока после того, как он потратил время на игру, которая мотивирует его, но не наполняет.

Веками азартные игры завлекали людей, используя наши до- исторические дофаминовые триггеры. Теперь и неазартные игры могут делать то же. Хотя мы не берем деньги у игроков так же, как при игре в казино, мы тратим их время. И остается вещественный аргумент в пользу того, что если мы отнимаем какое-то время, то мы должны что-то вернуть взамен.

Угрызения совести игрока могут выйти за рамки нескольких потраченных часов. Люди потеряли работу и своих супругов, потому что они просто не могли перестать играть. Является ли этически приемлемым создание игры, которая это поощряет? Если нет, то где находится эта грань? Должны ли мы возложить всю ответственность за саморегуляцию на игрока или дизайнер тоже должен ее разделить? А что, если некоторые игроки — дети?

Ответ на этот вопрос можно представить в виде спектра. Я думаю, что против четко выстроенной структуры вознаграждений, связанной со значимым основным опытом, найдется мало аргументов. Это просто хороший геймдизайн. Игрок хочет играть, и он любит играть.

В середине спектра находятся игры, которые чередуют навязчивость и полноценность. Люди теряют друзей, потому что они часами проводят в игре, но они и заводят друзей в той же игре. Они часами страдают над мучительно скучным гриндом, а затем выигрывают захватывающую битву с боссом. Такие смешанные впечатления часто бывают вызваны прямыми ошибками искусства геймдизайна. Если бы награды в игре были лучше скорректированы, основной опыт мог бы проявиться лучше.

В конце спектра находятся игры, которые даже не пытаются серьезно удовлетворить игроков. Эти игры сосредоточивают каждое дизайнерское решение на максимальном увеличении мотивации. Такие игры не являются машинами опыта; это просто надоедливые механизмы. Игрок играет и платит и, в конце концов, ничего не чувствует из-за сожаления о потраченных времени и деньгах.

Подобные игры лучше всего иллюстрируют пародии на них. Например, Cow Clicker Яна Богоста — это игра в Facebook, в которой фигурирует корова. Вы можете кликать на корову, но только

каждые шесть часов, если только вы не платите в игре «муни» (внутриигровые деньги), чтобы иметь возможность кликать раньше, или не шлете спам своим друзьям, приглашая их присоединиться к вам и вашим коровам. В игре только «голый» режим подкрепления с постоянным интервалом. Богост создал игру так, чтобы подчеркнуть то, что, по его мнению, являлось злоупотреблением шаблонами дизайна в реальных играх.

Но люди все равно играли в Cow Clicker тысячами. Богосту до такой степени надоела его собственная игра, что он заставил всех коров исчезнуть, оставив только пустой клочок земли. Но даже это не остановило людей. Они просто продолжали кликать по земле. Дофамин — адское химическое вещество в мозге.

Б. Ф. Скиннер полагал, что все организмы, включая людей, являются не более чем «системой поведения», управляемой исключительно внешними силами. И он довел эту идею до логического завершения, сказав, что наполненной позитивным содержанием концепции «Я» вообще не существует. В своих мемуарах он писал:

Меня иногда спрашивают: «Вы и себя воспринимаете так же, как те организмы, которые изучаете?» Да, так и воспринимаю. Насколько я понимаю, мое поведение в каждый конкретный момент представляет собой не что иное, как результат моего генетического наследия, личного опыта и существующих внешних обстоятельств... И если мои утверждения в отношении человеческого поведения верны, выходит, что я написал автобиографию не человека.

Последователи Скиннера были менее радикальными в этом вопросе, чем он сам. Но его наследие живет в некоторых теориях современного геймдизайна. Этот взгляд на дизайн рассматривает его как задачу создания не эмоций или удовлетворения, а поступков-триггеров. Он полностью игнорирует феноменологический опыт игры и рассматривает геймдизайн как конструкцию виртуального ящика Скиннера, извлекающего максимальную интенсивность желаемого поведения у игрока. И обычно такое поведение приносит издателю средства.

Эта книга о мастерстве, а не об этике. И эти вопросы гораздо обширнее, чем этот краткий обзор. Я не описывал ответственность игрока, не сравнивал игры с другими медиа и не рассматривал смешанные положительные и отрицательные результаты почти всех игр. Поэтому я не буду пытаться дать окончательный ответ на этот вопрос.

Но я могу предложить точку зрения одного дизайнера. Я думаю, что любой дизайнер, желающий освоить искусство геймдизайна, хочет большего, чем просто построить ящик Скиннера. В конечном счете, это может быть единственным перспективным путем. Гонимые игры, в основе которых лежит получение наград, быстро распространяются среди наивных игроков, потому что на такие игры легко подсесть. Но со временем эти игроки учатся избегать угрызений совести. И после того, как их жажда бессмысленной добычи исчерпана и они изучили все наши трюки, игроки все равно будут хотеть то, что они всегда хотят от игр: новых идей, новых друзей и новых впечатлений.

Интерфейс

Командующий Снаргод взял странный кусок пластика с планеты Земля и уставился на него всеми своими четырьмя стебельчатыми глазами.

«Полагаю, так они разговаривают с машиной», — пульсировал он.

«Да, — ответил инженер Ксизваз. — Мы наблюдали за их манипуляциями с этим устройством, они делали это своими конечностями с пятью щупальцами».

«Мы можем адаптировать устройство для нашего использования, — пульсировал Снаргод. — Сделайте версию для нас».

«Вношу поправки, — ответил Ксизваз. — Проблема не в устройстве. А в том, что появляется на визуальном прямоугольнике, когда мы запускаем алгоритм. Никто из нас не может понять, что происходит. Посмотрите».

Ксизваз нажал несколько кнопок управления, и стена ожила. На ней было видно несколько людей, которые перемещались по пустынной земной среде. У всех была разная одежда — форменная, с металлическими нитями, обычная. Некоторые сидели неподвижно, другие были взволнованы. Некоторые казались не человекоподобными, но при этом двигались и взаимодействовали как люди. Различные человеческие технологии были разбросаны по всему экрану.

«Мы пытались, — пульсировал Ксизваз. — Но никто из нас не может сказать, что это значит».

«Ксизваз, — пульсировал Снаргод, — кража этой игры должна была помочь нам понять людей, а не запутать нас».

«К сожалению, кажется, что нам нужно сначала понять людей, прежде чем мы сможем понять игру».

Слышен ли звук падающего дерева в лесу, если рядом никого нет? В мире геймдизайна — нет. События имеют эмоциональную ценность, только если игроки их воспринимают и понимают.

Дизайнеры в подобных случаях используют такие инструменты, как экраны, колонки и несколько других выходных устройств.

Но обычно в игре происходит гораздо больше, чем можно передать за раз по таким ограниченным каналам. Чтобы донести до игроков понимание того, что происходит в игре, мы должны разработать системы, которые тщательно структурируют и упорядочивают информацию.

То, о чем не сообщается, с таким же успехом могло вообще никогда не происходить.

Эта коммуникация работает в обе стороны. У игроков также должна быть возможность сообщить о своих намерениях в игре. Для этого мы используем кнопки, джойстики, геймпады, сенсорные экраны и датчики движения. Но ничто из этого не является идеальным решением. Джойстики и кнопки не позволяют естественным образом управлять человеком в трехмерной среде, создавать замки или командовать армиями. Чтобы обеспечить легкий ввод, мы должны разработать сложную комбинацию ограничений, соглашений и вспомогательных систем.

Если мы справимся с этими задачами, интерфейс исчезнет. Игрок больше не будет замечать кнопок, экранов или джойстиков и поэтому сможет погрузиться в саму игру. Но если у нас не получится, игра будет похоронена за ее интерфейсом. Неважно, насколько она увлекательна сама по себе, игрокам все равно, потому что она не сообщает ничего значимого. Игра не более чем то, что она сообщает.

К счастью, геймдизайнеры не первые, кто решает такие проблемы. Дизайнеры интерфейсов и кинорежиссеры разработали различные методы передачи сложной информации, и мы с радостью присвоили их. Вот почему в играх используются такие элементы пользовательского интерфейса, как мышка, горячие клавиши и подсказки. Мы позаимствовали так много кинематографических технологий, что можно легко забыть, что они не наши. Замедленная съемка, краткие эпизоды, приближение (зум), создание текста и озвучивание — все это методы киноиндустрии.

Но хотя эти заимствованные методы и полезны, они не могут в полной мере решить наши задачи, потому что потребности геймдизайна отличаются от потребностей киноиндустрии. Дизайнеры интерфейсов занимаются понятным интерфейсом и интерактивностью. Режиссеры сосредоточены на идее, ритме,

эмоциях и смысле. Геймдизайнеры должны найти решения, которые удовлетворяют всем этим требованиям одновременно.

Метафора

Одна из наиболее важных причин, по которым мы помещаем механику в задумку, — быстрая коммуникация. Это называется метафорой.

Классический пример метафоры — папки на компьютере. Жесткие диски компьютеров организованы в иерархическую структуру данных внутри других структур данных. Эти структуры можно было бы назвать как угодно — черепаха, машина, куча данных. Но мы называем их папками, потому что это слово мгновенно дает нам представление о большей части того, что нам нужно о них знать. Очевидно, что компьютерная папка — это не кусок мягкого картона, который помещается в шкаф. Но у нее достаточно концептуального сходства с реальной папкой, чтобы сделать сравнение полезным, потому что информация в ней организована аналогичным образом.

МЕТАФОРА придает новой информации знакомые черты, чтобы ее можно было легче понять.

В каком-то смысле весь сюжетный слой игры представляет собой гигантскую метафору. Мы создали разветвленные наборы игровой механики, которые было бы невероятно сложно выучить с нуля. Представьте себе, что вы изучаете сложную видеоигру, если бы она была представлена только абстрактными формами. Но затем мы оборачиваем их в растущий город или древнюю войну и все взаимоотношения и системы становятся понятными. В играх сюжетный слой служит многим эмоциональным целям, но самая простая и основная причина его существования состоит в том, чтобы помочь игрокам понять систему через метафору.

Источники метафор

Метафора работает на основании наших обширных запасов ранее полученных знаний. Это могут быть знания из разных источников.

Дизайнеры интерфейсов используют папки, вкладки и кнопки. Мы можем использовать автомобили, людей, самолеты, книги, рюкзаки — любой объект, который узнают игроки.

Метафоры могут имитировать реальные объекты.

В природе нет такого закона, который гласит, что люди с «козлиной бородкой» должны быть злыми. Но все мы знаем, что гении, поглаживающие свою козлиную бородку, замышляют вселенское зло, потому что козлиная бородка — исконный символ зла в нашей культуре.

Метафоры могут имитировать культурные архетипы и условности.

Культура полна подобных символических ассоциаций. В западных обществах мужчины с квадратными челюстями сильны и смелы. Хозяева корпораций богатые, злые и страдают манией величия. Черный означает смерть, синий — холод, а розовый — женщину. Все эти архетипы и условности можно передать через метафору.

Метафоры могут имитировать игровые клише и условности.

Каждый опытный игрок знает, что еда мгновенно заливает огнестрельные ранения, золотая броня прочнее стальной и лава безопасна, если ее не трогать. Эти условности не имеют смысла, но они были созданы и укреплены поколениями игр. Такие бессмысленные условности полезны для систем коммуникации, которые не имеют четкого реального или культурного аналога.

Мы также можем позаимствовать условности из пользовательского интерфейса или фильмов. Например, все знают, что маленькая кнопка X в верхнем углу окна закрывает его, потому что большинство операционных систем работают именно так. А когда экран игры постепенно меркнет, мы знаем, что сцена окончена и нас ждет что-то новое.

Проблема этих клише в том, что никто не может их расшифровать, не имея должных знаний. В этом смысле игровые клише особенно опасны. Например, я однажды видел, как игрок застрелял в одном помещении в оригинальной игре Legend of Zelda. Он бегал по кругу в поисках выхода. Мне было так неловко, что я, в конце концов, вмешался и подсказал ему, что можно использовать бомбы и взорвать стены с трещинами. Если бы я не помог, он мог бы просто сдаться и уйти из-за этой непонятной метафоры. Наличие в стене трещины не обязательно означает, что стену можно взорвать, это условное правило.

Метафора не должна ограничиваться физическими объектами или культурными символами. Мы также можем имитировать абстрактные системы и взаимоотношения, если люди их понимают.

Например, в играх часто встречаются такие системы, как ньютоновская физика, электричество и огонь. Но вовсе не потому, что эти системы каким-то образом превосходят все, что мы могли бы разработать. Мы могли бы легко создать игру, в которой гравитация отталкивает, а не притягивает или физика работает через пять искаженных измерений. Так мы даже можем открыть интересные новые виды игры. Но такие игры было бы чрезвычайно сложно понять. Реальная физика — очень сложная система, но все ее уже знают. Ее имитирование — элегантный способ создания мощной системной основы для игры, практически не требующей обучения.

Метафоры могут имитировать логические системы.

Метафоры также могут имитировать концепции более высокого уровня современного мира. Мы можем ссылаться на системы экономики, политики, биологии или психологии, если игроки их понимают. Существуют игры с системами, которые имитируют спрос и предложение (Privateer), избирательную политику (The Polit Machine) и способ, которым люди строят и разрывают отношения (The Sims). Ни в одном случае метафора не является идеальной симуляцией реальной системы. Но, как и в случае с папками, она достаточно близка, чтобы можно было понять смысл.

Но наиболее распространенная форма метафоры является самой абстрактной: использование математических систем, таких как числа, время и пространство. Например, в шахматах 64 квадрата с 2016 соотношениями между ними. Эти 2016 соотношений можно было бы выразить любым количеством способов. Мы могли бы написать длинный список всех соотношений: a1 слева от b1; a1 — это два квадрата слева от c1 и так далее. Или мы могли бы представить каждый квадрат шариком для пинг-понга и использовать 2016 цветных строк, чтобы отметить соотношения. Или мы могли бы представить соотношения любым другим способом. В каждом случае фундаментальная система шахмат остается неизменной, но игра все равно непонятна. Только когда мы поместим 64 квадрата на двумерную плоскость, в игру

можно будет играть. Подражая самому реальному пространству, шахматы используют естественную способность человеческого мозга думать о сложных пространственных вещах. Подобная системная метафора почти универсальна по причине своей невероятной элегантности.

Словарь метафор

Автомобили в видеоиграх обычно едут вперед на колесах. Иногда им нужен бензин. Однако им достаточно редко нужно менять масло, иметь регистрационные номера или получать парковочные талоны. Точно так же людям в игре часто не нужно в туалет, собаки в игре никогда не заражаются блохами, а еда в игре редко портится.

В игровой механике фактически реализовано лишь небольшое количество функциональных возможностей реального объекта.

По негласному договору между игроком и дизайнером последний будет использовать метафору, чтобы помочь игроку учиться, а игрок не будет жаловаться, если механика не выражает каждое имитируемое свойство. Но это создает проблему для игрока. Теперь ему предстоит выяснить, какие аспекты игры являются настоящей игровой механикой, а какие — просто фейк. Дизайнеры должны сделать ее как можно проще.

Вспомните мою историю о взрыве стены с трещиной в игре *The Legend of Zelda*. По задумке, бомба, взрывающая потрескавшуюся стену, имеет своего рода смысл. Бомба вполне может сделать дыру размером с дверной проем в стене. Проблема в том, что, видя сначала потрескавшуюся стену, игрок не может знать, что именно этот аспект реальной стены выражен в игровой механике. Игра полна вещей, которые, по задумке, должны использоваться, но на самом деле они не работают. Неплотно скрепленные цементом кирпичи можно было бы достать и использовать их как оружие, но такой модели нет. Кровь мертвого монстра можно было бы выпить, но такой модели тоже нет. Игрок не может ходить и предполагать, что все, что присутствует в сюжете, также связано с механикой, потому что тогда он будет почти все время ошибаться. Игрокам нужен какой-то сигнал о том, что эта трещина в стене на самом деле механика, а не просто еще одна сюжетная добавка.

Один из способов добиться такого эффекта — вызвать интерактивные элементы с помощью явных индикаторов. Но они быстро утомляют. Постоянные окна на экране, указывающие на каждое доступное действие, заслоняют собой весь сюжет. Лучшее долгосрочное решение — научить игроков интерпретировать имеющиеся намеки.

В каждой игре должен быть свой собственный словарь метафор. Например, в акробатической игре Prince of Persia: The Forgotten Sands достаточно трудно точно сказать, на какие объекты окружающей среды можно залезть, а на какие нет. Разработчики решили эту проблему, создав словарь внешних ориентиров, которые показывают, какие действия можно выполнить и где. Длинные полосы вдоль стены указывают на то, что персонаж может бежать по ней. На выступающий кирпич необычной формы можно залезть. Эти элементы создаются на ранней стадии в очень простых головоломках. Например, уникально выглядящие кирпичи, на которые можно залезть, впервые появляются в том месте, где очевидно, что другого пути развития нет. Игрок гарантированно попытается залезть на кирпичи, потому что у него нет другого выбора. И как только игрок узнает, как выглядят подобные кирпичи, по которым можно карабкаться, он сможет распознать их до конца игры, потому что они никогда не меняются. Он путешествует по дворцам, канализациям и храмам. Но независимо от того, куда он идет, эти кирпичи всегда выглядят одинаково.

Игра должна задать СЛОВАРЬ МЕТАФОР, который указывает, какие элементы моделирует механику, и соответствовать этому словарю.

Сигнал и шум

Любая информация, которую получает игрок, является частью сигнала. Сигнал делится на разные каналы: визуальная информация с экрана, аудиоинформация из колонок, возможно, тактильная или другая обратная связь. Игроки используют свои естественные способности, чтобы фильтровать, расставлять приоритеты и интерпретировать информацию, для того чтобы попытаться понять, что означает сигнал. Но если это не удастся, части сигнала превращаются в шум.

Шум — это не просто бессмысленные сигналы, подобные статическим. Он включает любой сигнал, который не добавляет значимой информации к когнитивной модели игры. При этом не

ШУМ — это сигнал,
который не может
передавать значимую
информацию.

важно, мешает ли сигналу слой эстетической красоты, требуется ли для его интерпретации слишком много навыков или он просто слишком тесно переплетается с другими сигналами. Результат одинаков: игрок не понимает игровые события и поэтому не может думать о них или реагировать на них.

Существуют две основные причины шума: сложная графика и перенасыщенные сигналы.

Шум и сложная графика

Как вы, наверное, уже знаете, большая часть моего опыта в дизайне — шутеры от первого лица. Далее следует история, которая происходила множество раз в процессе проектирования уровней.

Я начинаю проектировать новый уровень. Сначала я работаю по методу серого ящика. Серый ящик выглядит именно так, как вы его представляете: это пространство построено из плоских серых коробок. Дерево может быть представлено в виде палки с большим шаром на вершине, как гигантское яблоко-леденец на палочке. Автомобиль — это широкая низкая коробка с маленькой коробкой сверху. Таким образом, я могу работать очень быстро. В лучшем случае я могу найти проблему, исправить ее и повторно протестировать игру менее чем за 10 минут. Серый ящик может многое передать, боевые схватки, место для нарратива и геометрию уровней — все это можно завершить без добавления графики. К концу работы по методу серого ящика у меня получается сбалансированный уровень, которым наслаждаются тестировщики.

Но, в конце концов, уровень должен иметь какой-то внешний вид. Поэтому я работаю с художником, чтобы заменить гигантские серые леденцы на деревья, и перекрашиваю коробки на улицах, чтобы они выглядели как «фольксвагены» и «форды». И в этот момент все летит к чертям.

Интуитивно понятные головоломки внезапно становятся непонятными. Игроки пропускают маршруты, которые они обычно находили без проблем. Они не видят врагов, которых раньше находили легко. Простая замена серых поверхностей на графику сделала игру непригодной, хотя дизайн механики вообще не изменился. Я видел, как это происходит снова и снова, в нескольких играх на нескольких платформах. Проблема в том, что графика создает визуальный шум.

В мире серых фигур легко рассмотреть любого врага, любую цель, инструмент и путь. Поскольку у каждой фигуры на экране определенное механическое значение, игроку не нужно выполнять даже небольшую умственную работу, чтобы понять, что важно. Но когда мы добавляем графику, мир наполняется линиями и цветами, которые не имеют ничего общего с игровой механикой. Теперь мозг должен поработать, чтобы определить, какие фигуры на самом деле имеют значение. Иногда это слишком сложно, и сигнал превращается в шум.

Сложная графика
создает шум.

Это не ограничивается визуальными эффектами среды. Любая сложная графика создает шум. Интерпретировать сложные звуковые эффекты труднее, чем простые. Более детализированный персонаж может быть более красивым, но все лишние формы скрывают механическую информацию, которую он должен передать: где он стоит, как он движется и что он держит. Детализированная анимация воспринимается труднее, чем простая.

Это широко распространенная проблема, потому что она пересекает границы между дисциплинами разработки. Дай художнику волю, и он определенно будет стремиться создать самую красивую картинку, которую только сможет. Он соберет детали, которые при визуализации изображения с высоким разрешением сделают графику великолепной, точно так же, как части портфолио, благодаря которым он получил работу. Но как только этот чрезмерно детализированный персонаж появится на игровом экране вместе с сотней других персонажей, он превратится в пятно пикселей. Все эти красивые детали становятся шумом при отображении в динамичной игровой ситуации.

Дизайнеры и художники должны работать в команде, чтобы придать игре вид, который был бы не только приятным визуально, но и понятным с точки зрения его механики. Иногда это просто означает баланс количества деталей в графике или использование специальных подсказок, благодаря которым появятся части изображения, связанные с механикой. В других случаях разработчики идут дальше и создают уникальное оформление, которое может быть красивым и естественным образом снижать шум. Например, в Valve Team Fortress 2 и Portal, а также в DICE's Mirror's Edge используются задумки, разработанные специально для создания простых, понятных изображений. Персонажи Team Fortress 2 отображаются в карикатурном стиле с простыми текстурами и гиперболическими силуэтами. Действие Portal происходит в научном испытательном центре, состоящем исключительно из камер с белыми стенами (прямо как уровень серого ящика). Действие Mirror's Edge происходит в городе, построенном в основном из голого белого бетона (опять же как уровень серого ящика). Такая графика решает сразу несколько проблем в разных дисциплинах и придает каждой из этих игр свой уникальный вид.

Визуальная иерархия

В любом процессе разработки наступает момент, когда тестировщик не замечает какого-либо сигнала, который по замыслу дизайнера должен быть замечен. Он не замечает слов на экране. Он не видит персонажа. Он пропускает часть диалога.

Здесь очевидное решение: сделать сигнал громче. Добавьте больше визуальных индикаторов, более громкие звуковые эффекты и более энергичную анимацию. Но, как сказал Марк Твен, «у каждой сложной проблемы простое, всем понятное и неправильное решение». Часто добавление видимости является понятным и неправильным решением этой проблемы.

Дело в том, что игроки редко пропускают сигналы, потому что они недостаточно заметны. Они пропускают их, потому что их восприятие подавляется другими сигналами. Дело не в том, что пропущенный сигнал был слишком тихим, а в том, что общая

громкость всех сигналов слишком высокая, а игрок перегружен. Добавление громкости к одной части сигнала еще больше перегружает игрока.

Мы должны попытаться сбалансировать общую плотность сигналов в соответствии со способностью игрока воспринимать их. Но это не решает проблему полностью для всех игроков, потому что игроки на разных уровнях мастерства могут воспринимать очень разные объемы информации. Сравните, сколько времени потребуется вам, чтобы прочитав этот текст, и как ребенок будет произносить буквы по одной, чтобы сложить их в слова. Точно так же опытные игроки могут впитывать информацию в игре во много раз быстрее, чем абсолютные новички. Таким образом, плотность сигнала, подходящая для новичков, обеспечит эксперту скучное прерывание потока, а плотность сигнала, подходящая для экспертов, обескуражит новичка. Напоминает роман Джозефа Хеллера «Поправка-22».

К счастью, существует техника, используемая графическими дизайнерами, которую мы можем позаимствовать для решения этой проблемы. Она называется визуальной иерархией.

Если человек перегружен сигналами, он неосознанно игнорирует наименее заметные части. Это не навык, это универсальная способность подсознания человека воспринимать информацию. В первую очередь мы замечаем то, что больше, ближе, ярче и быстрее. Именно поэтому в рекламе используются яркие цвета, а компании-эмитенты кредитных карт пишут мелким шрифтом.

Дизайнерам это тоже полезно, потому что так мы можем контролировать порядок, в котором игроки воспринимают информацию. Все, что нам нужно сделать, — присвоить каждому фрагменту данных различную видимость — сделать его ярче или мрачнее, громче или мягче. Не обязательно добиваться абсолютной видимости каждой части; все, что имеет значение, — относительная видимость. Если видимость каждо-

Игроки могут воспринимать только определенное количество сигналов одновременно. Игрок не может воспринять последующие сигналы, превышающие этот предел, поэтому эти сигналы фактически становятся шумом.

В ВИЗУАЛЬНОЙ ИЕРАРХИИ все отображается сразу, но более важные фрагменты информации становятся более заметными, чтобы люди их заметили первыми.

го фрагмента информации соответствует ее важности, игроки по всему диапазону мастерства будут воспринимать только ту информацию, которая им полезна, при этом автоматически игнорируя все остальное.

Например, начинающий игрок может не знать игру, но если кто-то большой поражает его персонажа, игрок автоматически игнорирует мини-карту, здоровье, инструменты, музыку и фоновые символы, потому что эти элементы менее заметны. Это хорошо, так как низкий навык в любом случае не позволяет ему использовать эту информацию. Его навык дает ему возможность взаимодействовать только с простым сигналом большого персонажа; хорошо, если он игнорирует все остальное.

По мере роста его навыков способность восприятия тоже растет. Игрок начинает видеть второй наиболее видимый элемент, затем третий и так далее. Он начинает замечать свой низкий уровень здоровья и убегает от большого персонажа, когда это необходимо. В другом случае он видит приближающегося союзника на мини-карте и остается в бою. Устанавливая относительную видимость этих элементов, дизайнеры решают, что и когда будет воспринимать игрок. Так, если они хорошо поработали, то прокачав свои навыки, игрок всегда увидит следующий важный элемент.

В любой игре может быть визуальная иерархия. Просто посмотрите на интерфейс и спросите: существует ли какая-либо менее заметная часть игры, которую нужно изучить до того, как приступить к изучению другой части? Если да, меняйте ее видимость до тех пор, пока не получите правильный порядок.

Давайте разберем визуальную иерархию стандартного шутера.

Непосредственные враги игрока очень важны, поэтому они большие, заметные и узнаваемые.

Более отдаленные враги не так важны, поэтому занимают меньше места на экране. Обратите внимание, что эта естественная взаимосвязь между расстоянием, релевантностью и заметностью является одним из свойств 3D-пространства, что делает его элегантной основой для видеоигр.

Здоровье — это интересный случай. Одним из нововведений современных шутеров является то, что они меняют видимость здоровья по мере изменения его значения. При критическом здоровье игры выводят красное изображение на весь экран и воспроизводят звуковые эффекты боли, делая сигнал весьма заметным. Если персонаж здоров, эти эффекты исчезают и сигнал затихает. Это хороший способ поместить информацию о здоровье в сознание игрока только тогда, когда это необходимо.

Количество боеприпасов отображается в углу. Его можно проигнорировать, но все же его легко найти с первого взгляда. Как и здоровье, оно важно, если значение низкое, поэтому некоторые игры делают его более заметным, если боеприпасы почти на исходе, проигрывая специальный звук или выбрасывая текст внизу середины экрана.

Настройки всей этой информации соответствуют ее важности. Конечным результатом является визуальная иерархия, которая делает игру понятной для всех уровней мастерства.

Для этого дизайнеры должны определить относительную важность различных элементов и настроить их видимость соответствующим образом. Существует множество способов изменить видимость фрагмента информации. Например, давайте рассмотрим некоторые способы настройки видимости количества боеприпасов.

Достаточное количество боеприпасов будет отображаться мелким шрифтом в углу экрана. Игрок должен целенаправленно искать эту информацию. Без обучения он может даже не заметить ее.

Более заметные версии отображают графический элемент пользовательского интерфейса с указанием количества патронов, располагая бокс ближе к середине экрана. Теперь игрок может краем глаза видеть количество боеприпасов.

Если мы хотим еще большей видимости, мы можем поместить информацию о количестве боеприпасов в оптический прицел в центре экрана. Теперь игрок увидит количество боеприпасов, даже не пытаясь никуда смотреть. К этому и должна стремиться реальная игра.

Но если хотим увеличить видимость еще больше, мы можем это сделать. Счетчик боеприпасов может увеличиваться и заполнять половину экрана. Иконки пули могут мигать. Автоматизированный голос может озвучивать количество оставшихся боеприпасов при каждом выстреле (это было сделано).

Мы также можем быть скрытнее. Количество боеприпасов может исчезнуть, если игрок не стреляет или не перезаряжается в течение 10 секунд. Оно может быть скрыто в меню. Оно также может быть скрыто глубоко в подменю, на веб-сайте разработчика или в файле конфигурации. Ни одна игра не отображает количество боеприпасов таким образом, но некоторые игры отображают так другие типы малозаметных данных.

Сила визуальных иерархий заключается в том, что они автоматически и мгновенно передают информацию, необходимую игрокам во всем диапазоне мастерства. Игрок, который еще учится, может перейти к следующему фрагменту данных, как только его уровень мастерства позволяет ему это сделать. Визуальная иерархия никогда не перегружает игроков и не заставляет их бросить игру, пока они учатся. Она всегда предоставляет достаточное количество информации и никогда не тормозит обучение.

Избыточность

У каждого дизайнера был такой болезненный опыт: мы создаем новую графику, последовательность или звук, чтобы передать важную информацию. У нас существует какая-то новая технология, и графика получается замечательной. Мы демонстрируем результат влиятельным людям, и все согласны с тем, что это потрясающе.

Тогда мы проводим плейтест, и большинство игроков полностью все пропускают. Они смотрят не в ту сторону, их разум занят вообще другой задачей, они отвечают на звонки по мобильному или что-то орут пьяным голосом своему другу, потому что в эту игру играют на вечеринке.

В фильме все, что имеет значение, находится на экране. Пропустив минуту фильма, вы не станете винить режиссера, когда

вернетесь и не поймете сюжет. Читатель, пропустивший десять страниц романа, не винит автора в том, что оставшаяся часть книги запутана. Однако в играх игроки могут смотреть куда угодно, делать что угодно, как внутри игры, так и вне ее. Они могут отвлечься от важных событий. И если они что-то упустят, то обвинят во всем нас.

Очевидное решение — заставить людей заметить важный контент. Но поворот камеры определенным образом, прерывание игры диалоговыми окнами и другие механизмы принудительного контроля нарушают поток и погружение. Решение одной проблемы привело к появлению множества других.

Лучше всего просто признать, что игроки будут пропускать важные сигналы. Затем, вместо того чтобы заставлять их воспринимать один сигнал, мы посылаем этот сигнал несколько раз. Избыточность означает, что даже если игрок пропускает половину содержимого в игре (довольно обоснованное соотношение), он все равно получает достаточно, чтобы понять то, что действительно важно.

Простейшей формой избыточности является *равномерная избыточность*.

Иногда мы можем сделать это незаметно для игроков. Например, мы размещаем один и тот же аудиожурнал в пяти разных местах, но как только игрок слушает один из них, остальные незаметно исчезают.

В других случаях у нас нет способа узнать, действительно ли игрок получил сообщение. Например, сообщение звучит через динамики, но нет никакой гарантии, что игрок его действительно прослушал. В этих случаях нам, возможно, придется просто повторить его. Но обычное повторение может раздражать, поэтому лучше использовать *многообразную избыточность*.

В шутере мы можем сделать так, чтобы сопровождающий персонаж крикнул вам прыгать в окно (диалог), махнул рукой в сторону окна (анимация), можно разместить визуально подсвеченную дорожку из деревянных досок,

РАВНОМЕРНАЯ
ИЗБЫТОЧНОСТЬ —
это одинаковое
повторение одного
и того же сообщения
несколько раз.

МНОГООБРАЗНАЯ
ИЗБЫТОЧНОСТЬ —
это передача одной
и той же информации
несколько раз разными
способами.

ведущих к окну (визуальные эффекты уровня), и поместить на экран соответствующий маркер с инструкцией для игрока выпрыгнуть в окно (HUD). Четырехкратная избыточность означает, что, вероятнее всего, даже в стрессовой ситуации игрок поймет сообщение.

В других случаях целесообразно показывать вторичные и третичные сообщения только в случае, если основное сообщение не дошло. Это называется *пассивной избыточностью*.

ПАССИВНАЯ
ИЗБЫТОЧНОСТЬ —
это использование
вторичных сообщений
только в случае, если
основное сообщение
не прошло.

В примере про выпрыгивание в окно, если бы у сопровождающего персонажа была вторая или третья линия диалога, которая использовалась бы только если игрок недостаточно быстро выпрыгивает в окно, то персонаж использовал бы пассивную избыточность. Опасность заключается в том, что зачастую трудно определить, пропустил ли игрок первое сообщение или он намеренно его игнорирует. Если игрок делает это намеренно, пассивные избыточные сообщения могут раздражать.

Непрямой контроль

Иногда мы хотим, чтобы игроки выполнили определенные действия, но мы не хотим точно указывать им, что нужно делать. В этих случаях мы используем методы *непрямого контроля*.

Методы НЕПРЯМОГО
КОНТРОЛЯ могут на-
правлять игрока, при
том что сам игрок
не осознает, что им
управляют.

Это не так необычно (или глупо), как 25-й кадр или нейролингвистическое программирование. Непрямой контроль — это использование простых методов упорядочения информации, чтобы поведение игроков шло в намеченном направлении. Пользовательский интерфейс и промышленные дизайнеры — яркие примеры такого контроля.

Существуют три основных метода непрямого контроля: подталкивание, прайминг и *социальная имитация*.

Подталкивание

В любой ситуации игроки стремятся идти по пути наименьшего сопротивления. Они выберут стандартный вариант и пойдут наиболее очевидным маршрутом. Это означает, что мы можем

подтолкнуть их в зависимости от того, чего мы хотим добиться, просто представив выбор по-другому и изменив параметры по умолчанию. Например, диалоговую систему можно настроить так, чтобы выбор по умолчанию всегда был таким, который приводит к наиболее интересному результату.

Мы также можем «подтолкнуть» визуальный дизайн. Освещенный дверной проем притягивает игроков лучше, чем темный. Линия на полу указывает игроку путь. Мигающая кнопка требует, чтобы ее нажали. В каждом случае игроки обычно делают выбор, который приводит к предполагаемому опыту, при этом их никто не принуждает.

ПОДТАЛКИВАНИЕ — это изменение поведения игрока путем изменения способа подачи выбора, без изменения самих вариантов выбора.

Подталкивание удобно, потому что это дешево. Внедрение обычно обходится недорого, не ограничивает варианты и не меняет средства поощрения. Оно просто упорядочивает варианты, чтобы преднамеренные решения стали более естественными. Нет причин, по которым игроков не стоит подталкивать при любой удобной возможности.

Прайминг

У человека всегда что-то на уме. Если вы видите рекламу военного фильма, то думаете о войне. Если видите фотографию котенка, мысли о войне сменяются на мысли о пушистости. Вы погружаетесь в эти мысли на несколько минут, в течение которых ваше поведение меняется. Думая о войне, вы становитесь агрессивнее и жесточеннее. Думая о котиках — заботливее и ласковее.

Участникам одного эксперимента были предложены письменные языковые тесты. Исследователей на самом деле интересовал не результат тестов, а только то, как слова в тесте повлияют на поведение людей. Одной группе дали тесты со словами, связанными с грубостью. Другой — тесты со словами, связанными с вежливостью. Затем участникам предложили выйти из аудитории и пройти по коридору к администратору. Когда участники эксперимента подошли к администратору, то увидели, что он разговаривает с другим человеком. Исследователи хотели выяснить, вмешаются ли испытуемые в этот разговор.

ПРАЙМИНГ (англ. priming — предшествование) — это активация концепций в сознании игроков с целью повлиять на их дальнейшее поведение.

Результат показал, что в разговор вмешались 63% испытуемых, которым достался тест с грубыми словами, и только 17% испытуемых, которым достался тест с вежливыми словами. Как оказалось, достаточно просто читать грубые слова, чтобы заставить человека вести себя грубо, по крайней мере, в течение нескольких минут.

В другом эксперименте текст был заполнен такими словами, как «морщины», «бинго» и «Флорида». После этого исследователи наблюдали, с какой скоростью испытуемые шли по коридору, выходя из здания. Те, кто читал слова, связанные со старостью, шли медленнее, чем те, кто читали текст с нейтральными словами, потому что слова, связанные со старостью, также активировали соответствующую концепцию медлительности.

Прайминг происходит постоянно, даже случайным, противоречивым образом. Если кто-то советует вам не думать о розовых слонах, вы думаете только о розовых слонах. Участников эксперимента проинструктировали избегать сексистских утверждений в задании, что в результате приводило к еще большему количеству сексистских утверждений, чем у тех участников, которые не получили никаких специальных инструкций. Чем меньше вы хотите думать о сексизме или розовых слонах, тем больше вы о них думаете.

В играх дизайнеры могут косвенно влиять на будущее поведение игроков.

Прайминг начинается еще до начала игры. Название игры, изображение на обложке и мнение других об игре являются первыми признаками прайминга. А во время игры игроков постоянно заряжают разные идеи. Они видят врача и ищут повод исцелиться. Они видят, как люди разговаривают, и готовы к общению. Они видят скелет и готовы к смерти. Каждое впечатление влияет на выбор игроков в течение нескольких минут после того, как они его пережили.

Представьте себе ситуацию в шутере, когда игрок собирается встретиться с агентом. Если до этого момента игра представляла собой не что иное, как сражение, то многие игроки будут сразу стрелять в него, потому что они жаждут насилия. Чтобы предотвратить это, дизайнеры могли бы дать прямые инструкции не стрелять или просто запретить стрельбу, но оба эти решения неэффективны и незелегантны. Лучше просто побудить игрока

отказаться от насилия. Этого можно достигнуть многими способами. Мы могли бы сделать так, чтобы персонаж игрока опустил оружие (оставив его пригодным для использования). Мы могли бы позволить игроку наблюдать за разговорами других персонажей. У нас может быть компаньон, который скажет: «Позвольте мне говорить». Любая подобная комбинация приведет к социальному взаимодействию, а не к насилию.

Прайминг — мощная, но не волшебная технология. Она не создает мотивацию из ничего. Все, что она делает, — это смещает предпочтения между уже доступными вариантами. Как экспериментаторы написали в своем исследовании, «весьма сомнительно, что участники второго эксперимента покинули здание, чтобы купить недвижимость во Флориде».

Социальная имитация

Подавляющее большинство знаний являются вторичными. Например, скорее всего, вы не изобретали свой собственный метод завязывания шнурков. Вы наверняка знаете, что черная вдова — ядовитый паук, хотя он никогда не кусал вас. Вы многое знаете о перестрелках и космических путешествиях, хотя у вас, скорее всего, нет подобного опыта. Эти знания из «вторых рук» — вовсе не недостаток, это необходимая адаптация. Человек, который должен был бы изобрести что-либо, умер бы сразу, как только нашел бы свой первый ядовитый фрукт или замерзающую реку. Поэтому у нас появился сильный инстинкт — учиться посредством подражания.

В играх инстинкт социальной имитации так же силен и значительно влияет на выбор игроков. Проще говоря, игроки будут смотреть, что делают другие, и делать то же самое.

В многопользовательской игре они будут имитировать стратегии друг друга. А в одиночной игре они подражают неигровым персонажам. Вот так социальная имитация становится инструментом для непрямого контроля. Все, что нужно сделать, — это заставить NPC делать то, что должен делать игрок, и игроки пойдут нужным путем.

СОЦИАЛЬНАЯ ИМИТАЦИЯ — это когда игрок естественным образом подражает действиям других.

Например, если в гонках управляемые компьютером машины будут снижать скорость перед поворотом, игроки научатся делать

то же самое. Или экономические симуляторы будут отображать количество акций у конкурентов. Если разница между количеством акций игрока и конкурента существенно отличается, игрок сможет понять, что либо он совершает ошибку, либо он (или его противник) намеренно преследует нестандартную стратегию.

Социальная имитация — главная причина, в силу которой во многих играх существуют персонажи-компаньоны. Они служат многим целям повествования, а также являются бесценными инструментами непрямого контроля. Мы можем заставить их бежать к следующей цели, взаимодействовать с головоломками, прятаться от опасности или использовать особенные способности, а игрок будет подражать им, даже не осознавая этого. Эти персонажи существуют для того, чтобы направлять игрока от пункта А до пункта Б, передавать чувства или воплощать истории.

Входной сигнал

Вы уже знаете о выходных сигналах игры. Теперь время поговорить о входных сигналах.

Цель дизайна входных сигналов — добиться синхронизации между намерением игрока и внутриигровыми действиями.

Игра с хорошим входным сигналом завораживает игроков в тот момент, когда они к ней прикасаются. Приятно просто взаимодействовать. И это ощущение постоянно. Оно длится в течение всей игры — десятки, даже сотни часов.

Плохой входной сигнал превращает даже самое простое взаимодействие в сложную задачу. Ничего не работает так, как должно, кнопки мягкие, как будто вы играете руками, погруженными в патоку. Плохой входной сигнал превращает опыт в сплошное разочарование. Обидно, когда это происходит, потому что входной сигнал может испортить отличные игры. Так же как старый черно-белый телевизор может испортить просмотр крутого фильма, так и замедленное, не интуитивное управление может испортить фантастическую игру.

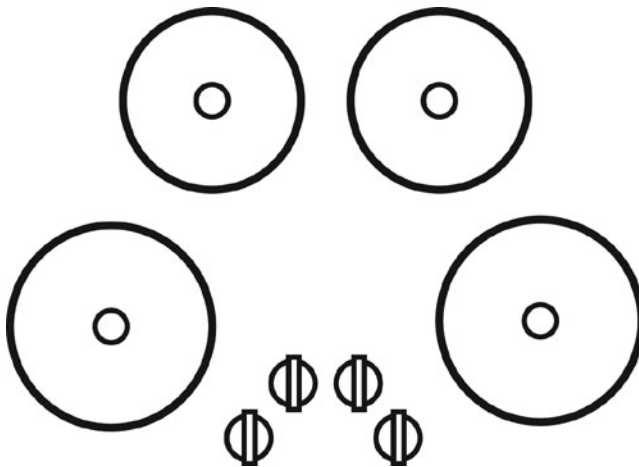
Управление

Существует огромное множество физических интерфейсов: кнопки, джойстики, датчики движения, тачпады и многое другое.

В каждом случае мы должны решить, какую функцию выполняет каждая система управления. Мы должны установить, что кнопка А отвечает за прыжок, кнопка В — за присед или размахивание левой рукой дает возможность метать огненные шары, в то время как наведение левой руки на цель дает возможность выпустить поток пламени.

Существуют два ключевых принципа, которыми мы должны руководствоваться при организации управления: маппинг и ограниченное управление.

Цель маппинга — создать сходство между физическим управлением и его внутриигровым эффектом. Если маппинг на хорошем уровне, это сходство служит как встроенный мнемонический код, который помогает игрокам вспомнить, как использовать элемент управления. Классическим примером маппинга является кухонная плита с конфорками, которая выглядит следующим образом:



Плите не нужны наклейки со знаками. Люди интуитивно понимают, как объекты соотносятся друг с другом.

В играх происходит то же самое. Например, в BioShock, используя плазмиды, персонаж игрока может применять суперспособности левой руки, а оружие держать в правой. Они управляются с помощью левого и правого триггера соответственно. Это создает

МАППИНГ (англ. mapping — нанесение на карту, отображение) — это соотношение между физическими элементами интерфейса и действиями, которыми они управляют.

маппинг между физическими руками игрока и руками персонажа на экране.

Маппинг не ограничивается физическим положением. Мы можем размечать форму, цвет, движение или сотню других видов сигналов. Например, в BioShock индикатор здоровья на экране отображается красным, так же как и кнопка лечения. Это создает маппинг между индикатором здоровья и красной кнопкой. Обратите внимание, что это тоже своего рода метафора, поскольку существует культурное условное обозначение, которое связывает красный со здоровьем.

Маппинг облегчает обучение. Он избавляет игроков от необходимости запоминать абстрактные взаимосвязи между символами кнопок и внутриигровыми действиями. Это особенно важно для менее опытных игроков, поэтому игры, ориентированные на новичков, иногда доходят до крайности. Контроллер движения Wii, система распознавания положения тела Kinect и огромный контроллер Rock Band в форме гитары — все это примеры дорогих физических интерфейсов, которые привлекают случайных игроков, создавая новые виды чрезвычайно детального маппинга.

В зависимости от физического соотношения между элементами управления и телом игрока некоторые пары элементов управления не могут использоваться одновременно. На геймпаде

могут быть две кнопки, которые можно нажимать только указательным пальцем левой руки. На контроллере движения игрок может свободно перемещать руки, но не может перемещать одну руку, чтобы выполнять две разные функции одновременно.

Игра должна преобразовывать ограниченные элементы управления в действия, которые являются эксклюзивными в игре. Например, контроллер Xbox 360 позволяет размещать указательный палец либо на кнопке-бампере, либо на триггере. Это сделано для того, чтобы их нельзя было нажать одновременно. В BioShock кнопка-бампер используется для выбора оружия, а кнопка-триггер — для стрельбы из него. В этом есть смысл, потому что в сюжете и механике

ОГРАНИЧЕННОЕ
УПРАВЛЕНИЕ — это
физическое соотно-
шение между раз-
личными элементами
управления и тем, как
они могут использо-
ваться вместе.

невозможно менять оружие и стрелять одновременно. Физические элементы управления ограничены, и они отражают ограниченные игровые действия, поэтому здесь нет конфликта. Более того, ограниченность физического контроля подчеркивает и естественным образом учит ограниченности внутриигровой механики.

Если ограниченные элементы управления отражают неограниченные действия, это приводит к разочарованию и неловким ситуациям. Вспомните любую игру, в которой выбор инструмента осуществляется на D-Pad Xbox 360, а движение — на левом стике. Для управления обоими элементами нужен свободный большой палец левой руки, а это значит, что игроки должны прекратить движение, чтобы поменять инструменты. Лучшим дизайном будет либо найти другой элемент управления для смены инструмента, либо добавить ограничение в игру, которое отразит ограничение физических элементов управления.

Чувствительность механизма управления

Еще в 1985 году Сигэру Миямото создал прототип всех платформенных игр, Super Mario Bros. Марио — это усатый итальянский сантехник низенького роста, который спасает принцесс, бегаёт по трубам и ест грибы. И прыгает — много прыгает. На протяжении всей игры он прыгает постоянно, снова и снова, тысячи раз. Фактически поначалу, в 1981 году, Марио назывался Джампмен.

На первый взгляд требуется не так много, чтобы заставить Марио прыгнуть. Вы нажимаете кнопку А, Марио поднимается в воздух, и гравитация притягивает его назад. Похоже на простую физику, но это не так.

Важно то, как долго вы жмете на кнопку. Быстрое нажатие заставляет Марио сделать короткий прыжок. Если вы удержите ее, Марио опишет высокую дугу в воздухе. Это происходит потому, что Марио движется вверх, пока вы удерживаете кнопку А, в пределах его максимальной и минимальной высоты прыжка.

Вторая половина прыжка совершенно нереальна. В обычной жизни объекты, брошенные в воздух, описывают параболу, потому

ЧУВСТВИТЕЛЬНОСТЬ
МЕХАНИЗМА
УПРАВЛЕНИЯ —
это мгновенный опыт
проецирования
намерения через
интерфейс.

что они падают с постоянной скоростью из-за силы тяжести. Но с Марио этого не происходит, потому что его гравитация не постоянна. Во время его подъема сила тяжести Марио минимальна. Достигнув вершины прыжка, она утраивается и Марио плюхается на землю. Но несмотря на свою огромную гравитацию, он не может набрать опасную скорость, потому что может упасть только с определенной максимальной скоростью. Как только он достигает этой конечной скорости, его гравитация становится практически нулевой.

Еще один пример. Максимальная высота прыжка у Марио не постоянная — прыжки с разбега позволяют ему прыгнуть выше, чем он может прыгнуть с места. Он может контролировать свое горизонтальное движение в воздухе, но не так хорошо, как когда бежит по земле. И если игрок хочет, чтобы Марио прыгнул сразу после приземления, то может нажать кнопку прыжка на несколько секунд раньше и Марио автоматически прыгнет, как только коснется земли.

Все это касается прыжка, управляемого одной кнопкой, в игре, выпущенной еще в 1985 году. Поначалу этот уровень сложности кажется почти абсурдным. Но это единственный способ создать великолепную чувствительность механизма управления. Миямото разработал и настроил все алгоритмы, чтобы приумножить опыт особым образом. Одни алгоритмы, например дополнительный рост, который игроки могут получить, собрав необходимое количество грибов роста, поднимают потолок мастерства. Другие, такие как запоминание нажатий в игре, которые происходят непосредственно перед приземлением, предотвращают разочарование, если вдруг нажатие кнопки не дает никакого результата. По отдельности они не имеют большого значения. Но вместе, умноженные на тысячи прыжков, которые игроки выполняют в игре Mario, они подняли опыт до уровня, превосходящего в то время опыт конкурентов. Марио не стал всемирной мегафраншизой из-за своих усов или способности скользить по трубам. Марио заработал миллиарды поклонников, потому что просто управлять им всегда в радость.

Эти методы повышения чувствительности механизма управления не ограничиваются платформерами. Они применимы

к различным жанрам: гонки, файтинги, экшн от третьего лица, шутеры и многие другие делают одно и то же тысячами разных способов. Кнопки управления имеют крошечные зоны нечувствительности или сглаживаются на несколько фреймов, чтобы исключить зашумленный ввод. Ввод запоминается на какое-то мгновение, если он поступает рано, и отбрасывается, если до его выполнения игрок нажмет другую кнопку. Комбинированные элементы управления немного отличаются от отдельных элементов управления. Нажмите на кнопку до максимума, и элемент переключится на специальный режим ускорения, когда входной сигнал выходит за пределы обычного диапазона значений. Возможности безграничны.

На первый взгляд эти методы выглядят как ужасный геймдизайн. В этой книге я сделал акцент на элегантности и простоте. В учебных заведениях программистов учат писать понятный, простой код. На уровне интуиции чувствуется, что такая огромная невидимая сложность плохо влияет на игру.

Но эти методы являются исключением из принципа элегантности. Они стоят очень дорого для дизайнера, но они ничего не стоят игроку, потому что он никогда не чувствует, что они существуют. На самом деле игрок вообще ничего не чувствует. Потому что, если мы делаем нашу работу правильно, интерфейс исчезает.

Помощь в отношении входных сигналов

Управление Марио в игре не только сложное, но и не очень удачное. Дизайнеры не пытаются понять, что делает игрок; они просто применяют некоторые простые правила, которые улучшают опыт. В некоторых играх этого недостаточно, и мы должны помогать игрокам более рационально.

Помощь в отношении входных сигналов состоит в том, чтобы угадать намерения игрока и незаметно сделать так, чтобы его входной сигнал соответствовал. В идеале, игрок никогда даже не осознает, что ему помогают.

Давайте разберем пример помощи в прицеливании. В консольных шутерах игрок целится, управляя ма-

ПОМОЩЬ В ОТНОШЕНИИ ВХОДНЫХ СИГНАЛОВ — это предварительная обработка на основе не-обработанных входных данных, полученных от игрока.

леньким джойстиком с помощью большого пальца. Этот метод настолько сложен, неточен и вызывает фрустрацию, что в течение многих лет люди думали, что консольные шутеры всегда будут уступать шутерам на ПК с управлением мышью. Затем, в 2001 году, игра Halo показала нам всем, как помочь игрокам с прицеливанием, чтобы шутеры хорошо работали на консоли. С тех пор они стали одним из самых крутых жанров консольных игр. Без помощи в прицеливании консольные шутеры никогда не стали бы такими, какими мы знаем их сегодня. Но как именно работает помощь в прицеливании?

Она помогает игрокам отслеживать цели. Хитрость в том, что игроки не должны заметить, что им помогают. Просто навести прицел на цель кажется слишком очевидным. Единственный способ не очевидно, но эффективно помочь при прицеливании — это создать ряд чередующихся подсистем, помогающих разными способами. Одна подсистема помогает игрокам отслеживать движущиеся цели, заменяя часть входного сигнала стика на входной сигнал, который точно соответствует движению цели на экране. Никто не наводит прицел на цель, дизайнер только помогает компенсировать движение цели. Другая подсистема помогает игрокам перестать двигать прицелом, когда он попал на цель. Система определяет, когда стик отпущен, ищет ближайшие цели и незаметно скользит прицелом по одной из них. На первый взгляд кажется, что это будет заметно, но игрок не сможет ничего почувствовать, потому что все происходит только при медленном движении прицела. Третья подсистема следит за игроками, которые почти пропустили цель, и слегка смещает направление движения прицела, чтобы он попал на нужный объект. Поскольку мы лишь немного регулируем направление движения прицела, а не его скорость, игрок ничего не заметит. Эффект зависит от расстояния между целью и игроком, ее углового расстояния от прицела, наличия других целей, уровня сложности, используемого оружия и прочих факторов. И эти три варианта являются лишь примером. В различных играх используются и многие другие подсистемы.

Причина, по которой эта система работает, а игроки ничего не замечают, заключается в том, что система влияет только на то, что игроки не в состоянии обнаружить естественным образом.

Игроки замечают произвольнодвигающиеся прицелы, но они не замечают, что они останавливались на несколько пикселей раньше, или ускорялись, или корректировали направление, потому что эти события скрытаны за движением, происходящим вокруг них. Поэтому помощь при прицеливании работает только в этих уголках потока входного сигнала.

Иногда дизайнеры даже используют помощь в прицеливании в качестве балансирующего фактора. Например, снайперская винтовка в игре Halo мощная и точная. Как и предполагалось, она эффективна при стрельбе с дальнего расстояния. Но ее мощь и точность делают ее не менее смертоносной в ближнем бою, что дублирует роли других видов оружия. Нулевая помощь в прицеливании при уменьшенном масштабе — это один из способов, которым разработчики Halo ослабили снайперскую винтовку при стрельбе с близкого расстояния. На первый взгляд оружие все еще остается абсолютно точным. Но теперь из-за полного отсутствия помощи в прицеливании игроку сложнее использовать винтовку нежелательным способом. В таких ситуациях это влияет на коэффициент успешности выполнения задания и заставляет игроков использовать оружие по назначению, и при этом никто ничего не замечает.

Помощь в прицеливании является лишь одним из видов помощи. Помощь в прыжках помогает игрокам направить своих персонажей в безопасное место. Помощь в атаке помогает игрокам наносить точные удары. Помощь в вождении помогает игрокам избежать аварии. Помощь в движении помогает игрокам скакать по уступам и избегать препятствий. В каждом случае дизайнеры разработали набор методов, помогающих игрокам, при этом не недооценивая их намерения и не делая ничего, что игроки могли бы заметить.

Задержка отклика управления

Когда вы нажимаете кнопку или перемещаете джойстик, игре требуется несколько миллисекунд, чтобы обработать входной сигнал и создать видимый отклик. Эта задержка называется задержкой управления.

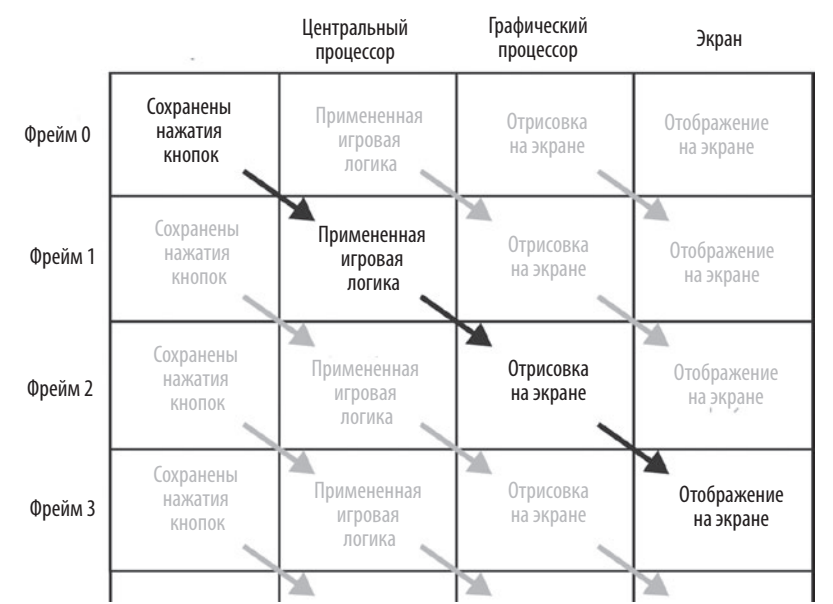
Задержка отклика управления неизбежна, поскольку она встроена в аппаратное обеспечение. Большинство совре-

ЗАДЕРЖКА ОТКЛИКА УПРАВЛЕНИЯ — это временная задержка между моментом, когда игра получает входной сигнал и когда она отображает видимую обратную связь, являющуюся результатом этого сигнала.

менных компьютерных систем используют многоступенчатую потоковую визуализацию. В любое время система имеет несколько фреймов на разных этапах обработки. Прогрессия выглядит так:

- **Фрейм 0 (входной сигнал получен).** В какой-то момент фрейма поступает входной сигнал — возможно, нажатие кнопки. Однако его нельзя использовать сразу, так как игра уже начала обработку фрейма. Вместо этого он сохраняется до начала следующего фрейма.
- **Фрейм 1 (игровая логика).** Процессор считывает входной сигнал, который был сохранен в течение последнего фрейма, и обновляет игровой мир.
- **Фрейм 2 (визуализация графики).** Визуализатор графики захватывает состояние игрового мира, заданное вычислениями предыдущего фрейма, и использует его для визуализации изображения.
- **Фрейм 3 (отображенный фрейм).** Отображается изображение на экране.

Каждый этап работает все время. В графическом виде это выглядит так:



Такое расположение обладает очень высокой производительностью, поскольку поддерживает постоянную работу как процессора, так и графического процессора. Без этой системы графическому блоку пришлось бы бездействовать в первой половине каждого фрейма, пока процессор обновлял бы игровую систему.

К сожалению, эта система также вводит три-четыре запаздывания фрейма между входным сигналом и откликом в зависимости от того, когда в течение фрейма был получен входной сигнал. (Большинство телевизоров добавляют еще один или два фрейма, чтобы обработать сигнал, прежде чем на экране появится изображение, но мы не можем контролировать эти задержки, поэтому я их проигнорирую.)

Плохой код может увеличить задержку еще больше, до пяти, шести, семи или более фреймов. Программисты несут ответственность за отсутствие в игре лишних фреймов с задержкой. Как только программисты убедились, что они не допустили ошибок и код не запаздывает, аппаратное обеспечение больше не позволяет снижения задержки ниже трех фреймов.

Единственный вариант — изменить длину каждого фрейма путем изменения общей кадровой частоты игры.

Но какую кадровую частоту нужно выбрать?

Некоторые ориентируются на фильмы или телевидение. Фильмы проигрываются со скоростью 24 к/с (кадра в секунду), а телевидение — со скоростью 30 к/с. Эти кадровые частоты были выбраны потому, что они являются нижними пределами, необходимыми для того, чтобы избежать заметного мерцания или полос изображения. Но мы не можем слепо применять эти стандарты к играм. Важная задержка в телевизоре — между одним кадром и следующим. Важная задержка в игре заключается между входным сигналом и откликом, который, как мы видели, в три-четыре раза длиннее.

Другие ищут ответ во времени реакции человека. Но это тоже не работает. Ответ, который мы ищем, не имеет никакого отношения к скорости, с которой игрок реагирует на игру. Он связан со скоростью, с которой игра реагирует на действия игрока, а также с тем, достаточно ли она быстрая, бесперебойная и синхронизированная.

На самом деле нет единой стандартной кадровой частоты, которая будет правильной для игры. Скорее это компромисс между графикой и откликами. Каждая миллисекунда дополнительной задержки делает игру немного медленнее, в то же время оставляя больше времени для визуализации сложной графики.

Например, представьте, что игрок шутера наводит прицел на цель с постоянной скоростью. Игрок видит, что цель будет под прицелом через 0,5 секунды. Представьте, что игра работает на скорости 30 к/с, а экран добавляет 50 мс запаздывания. Это зна-

Игроки учатся на-
жимать на кнопки
заранее, чтобы ком-
пенсировать задержку
отклика управления.
Длина задержки
отклика управления
в игре определяет,
насколько раньше
игроки должны нажи-
мать на кнопки.

чит, что общая задержка входного сигнала составляет 183 мс (четыре кадра по 1/30 секунды каждый плюс 50 мс от телевизора).

Если игрок стреляет, когда видит, что его прицел заходит на цель, он промахнет, потому что входной сигнал не дойдет на экран еще в течение почти одной пятой секунды. К тому времени его прицел уже пройдет цель (игнорируя помощь при прицеливании). Таким образом, чтобы попасть в цель, он должен научиться нажимать на курок на несколько кадров раньше. Более длительные задержки означают, что игроки должны уметь прогнозировать дальнейшие события, что требует большего мастерства и ощущения меньшей синхронизации.

Правильная кадровая частота зависит от потребностей дизайна. Некоторые старые текстовые приключенческие игры запускались со скоростью менее 1 к/с, поскольку им требовалось время, чтобы набрать описание на экране, шутеры типа Quake 3 можно запускать со скоростью 200 к/с и больше. Игры с частотой 1 к/с являются жизнеспособными, если они пошаговые и управляются нажатием клавиш. Частота 200 к/с — это верхний уровень кадровой частоты, к которому стремятся те, кто играет в соревновательные шутеры на ПК. Эти игроки запускают игры 10-летней давности на оборудовании в 30 раз более мощном, чем лучшее из доступного на момент выхода игры. Так они хотят получить максимум конкурентного преимущества.

Тем не менее подавляющее большинство игр запускается на скорости 30 или 60 к/с. Немногие игроки могут почувствовать

улучшение выше 60 к/с, а кадровые частоты меньше 30 к/с слишком низки, чтобы их можно было визуально заметить. Как правило, вы выбираете между этими двумя альтернативами. При 30 к/с, с оптимальной задержкой в четыре фрейма, задержка составляет от 100 до 133 мс плюс задержка экрана. При 60 к/с задержка падает до 50–67 мс плюс задержка экрана. В мире без ограничений по вычислительной мощности мы всегда выбираем 60 к/с. Однако в реальной жизни 60 к/с обходится дорого. Джон Кармак, легендарный разработчик игр *Doom* и *Quake*, заявил, что игра с частотой 60 к/с занимает около трети вычислительной мощности, доступной на фрейм, как и игра с частотой 30 к/с. Это дорогостоящий компромисс между графикой и технологической сложностью.

Выбор между 30 и 60 к/с зависит от предполагаемого мастерства игроков, типа интерфейса управления и особенностей эмоциональных триггеров игры. Игру, в основе которой лежит графика или нарратив, часто лучше оставить с частотой 30 к/с, потому что это позволяет получить более насыщенные визуальные эффекты. То же самое касается игры, в основе которой лежат навыки и которая не зависит от быстрого входного сигнала, как пошаговая стратегическая игра. Частота 60 к/с важна в тех случаях, когда опытные игроки взаимодействуют с системами, требующими быстрой реакции. Шутеры, гонки и файтинги являются наиболее распространенными кандидатами на частоту 60 к/с.

Зачастую другие заинтересованные стороны разработки проекта оказывают давление с целью отказаться от стандарта 60 к/с, даже если это правильный выбор. Это происходит по той причине, что затраты при частоте 60 к/с напрямую влияют на многие отдельные заинтересованные стороны в процессе разработки, при том что полезный эффект малозаметен. Художники хотят раздвинуть границы масштаба, детализации и красоты. Программисты хотят, чтобы больше циклов процессора увеличивало сложность ИИ и физических систем. Маркетологи и издатели хотят самые впечатляющие скриншоты и рекламу. Эти группы часто оправданно сомневаются в том, что необходимо пожертвовать двумя третями своей вычислительной мощности за фрейм для, казалось бы, невидимой выгоды. Так что существует

определенная предвзятость против частоты 60 к/с. Вот почему дизайнерам всегда нужно усердно доказывать необходимость использования частоты в 60 к/с.

Дизайн входного сигнала

Входной сигнал трудно спроектировать, потому что его сложно воспринимать. Я наблюдал за командами разработчиков, которые проверяли игру, раздавали и передавали друг другу контроллер и не соглашались с тем, что на самом деле делает игра, хотя все действия происходили у них на глазах. С практикой можно научиться отличать хорошую чувствительность механизма управления от плохой. Но вы никогда не сможете точно знать, что происходит внутри, потому что это намеренно скрыто.

И чем лучше система входного сигнала, тем сложнее ее понять. Помните старое выражение: чем проще кажется задача, тем труднее ее решение. Это правило как нельзя лучше подходит для дизайна входного сигнала. Для игрока управление прыжками Марио и система прицеливания в Halo — это просто. Но за этой простотой скрываются сложные системы вероятностей, пограничных случаев и систем помощи.

Большинство игроков никогда не поймут, почему игра ощущается так, а не иначе. Они будут благодарить графику или баланс за успехи, которые фактически принадлежат системе входного сигнала. Даже разработчики игр легко забывают о важности входного сигнала. Таким образом, его часто игнорируют. Если бюджет ограничен, а сроки поджимают, жалко тратить скудные ресурсы на то, что никто не может увидеть.

Но система входного сигнала почти всегда стоит своих вложений. Хотя ее редко замечают, игрок чувствует ее в каждый момент игры.

РЫНОК

Коныр был самым странным зверем в мире. Это была бесформенная масса размером с дом, покрытая странными клочками меха, чешуи и хитина. Перекрученные конечности произвольно торчали из его тела. И каждый день существо выглядело иначе.

Жители деревни узнали о его силе: если его правильно покормить, он по-царски награждает едой, хорошей одеждой и золотом. Беда была в том, что никто не мог понять, чего он хочет. Иногда он любил свинину и давал алмазы за каждый кусочек, брошенный в правильную пасть. На следующий день он хотел только овощи. Иногда ему нравились продукты недоваренные, переваренные, приправленные или безвкусные, сложные, простые, полезные и жирные. Иногда ему нравилось питаться тем, что даже не было едой.

Веками и купцы, и мудрецы пытались понять Коныра и предсказать его желания. Они хотели получить доступ к награде без проб и ошибок, бросая ему в пасть случайные продукты. Ничего не получилось. Через тысячу лет жители деревни просто поклонялись ему, ничего не понимая.

Цель дизайна

Каждая игра создана для достижения определенной цели. Некоторые игры создаются для получения прибыли за счет продаж, подписок или внутриигровых покупок, а также получения четвертаков в монетных аркадах. Другие игры созданы для неденежных целей. Арт-игры, хобби-проекты, академические эксперименты и дизайн-тесты создаются для статуса, права владения или развлечения.

Обзор некоторых распространенных бизнес-моделей показывает, насколько цель влияет на дизайн.

Аркады предназначены для того, чтобы игроки вкладывали деньги в автомат. Чтобы привлечь игроков, проходящих мимо, они используют быстро движущуюся визуальную выразительную механику. Длительное время игры уменьшит прибыль, поэтому аркады рассчитаны

Каждое проектное решение зависит от цели, для которой была создана игра.

всего за несколько минут. Чтобы сократить границы мастерства, управление и механика должны быть чрезвычайно просты. Чтобы заставить игроков вернуться и вставить в автомат еще больше монет, аркады имеют чрезвычайно гибкие условия выигрыша и очень высокий потолок навыка игрока. Во многие аркады выиграть невозможно; игрок просто поднимается все выше и выше, при этом все время оплачивая игру.

Массовые многопользовательские онлайн-игры (ММО) обычно зарабатывают деньги, взимая ежемесячную плату за доступ. Их цель состоит в том, чтобы игроки продолжали играть как можно дольше. Для этого ММО обычно добавляют огромное количество контента и очень глубокие системы апгрейда персонажей. Системы социального взаимодействия и построения сообщества заставляют игроков собираться в группы, которые удерживают их вместе.

Традиционные игры в упаковке приносят деньги за продажу единицы продукции, поэтому хорошие отзывы и сарафанное радио — их главные цели. В некотором смысле эта бизнес-модель дает безупречнейший геймдизайн, потому что лучший способ получить хорошие отзывы — просто создать отличный игровой опыт.

Некоммерческие игры, как правило, преследуют самые разные цели. Многие художественные игры скорее выражают идею, чем создают опыт. Они могут быть сфокусированы на какой-либо реальной проблеме, чем-то значимом для научного учреждения или на абстрактных концепциях.

А иногда дизайнер делает игру просто так. Когда я впервые начал делать игры в качестве хобби, мы не знали, что мы делаем, и даже не знали, зачем нам это. Мы сделали, просто потому что хотели.

Существует бесконечное множество других моделей, и регулярно появляются новые. Игровая реклама, демоверсии, игры с рекламой, развлекательные игры, серьезные игры, игры с пользовательским или эпизодическим контентом, микротранзакции, премиум-игры и предпродажа для фанатов — это лишь некоторые элементы, которые можно преобразовать в тысячу различных бизнес-моделей.

Каждая модель — это отдельная задача для дизайнера. И в каждой из них огромное пространство для того, чтобы проявить себя. Трудно создать коммерчески успешную однопользовательскую игру с грандиозным нарративом, как и оригинальную и трогательную игру для узкой научной аудитории. Но каждый отдельный случай имеет свои собственные задачи, ограничения и возможности.

Вот почему хорошая игра не обязательно будет успешной. Успех зависит не только от принципов дизайна, таких как элегантность, глубина и баланс. Он также зависит от понимания цели игры, ее соответствия рынку и того, как ее позиционирование на рынке должно влиять на дизайн.

Конкурентный рынок

Чтобы понять, как представлять игру на рынке, мы должны сначала понять структуру самого рынка.

В экономике термином «конкурентное благо» обозначается продукт, который не может одновременно потребляться несколькими агентами. Например, еда — это конкурентное благо, потому что яблоко, съеденное одним человеком, не может съесть кто-то другой. Геймдизайн относится к неконкурентным благам, так же как и романы, игра актеров, музыка. После создания этих продуктов стоимость их передачи дополнительным потребителям почти равна нулю.

Например, работа сантехника является конкурентным благом, потому что сантехник может одновременно работать только на одного клиента. Иначе говоря, посредственному сантехнику бояться нечего, если вдруг в городе появится один новый хороший сантехник. Такой сантехник сможет обслуживать только ограниченное количество клиентов, так что работа для посредственного сантехника всегда останется.

Но производители неконкурентных благ не сталкиваются с такими ограничениями. Хорошую игру можно копировать бесконечно практически без затрат и распространять среди любого количества пользователей. Это означает, что среднестатистиче-

скому дизайнеру действительно есть о чем беспокоиться в случае появления одного нового вундеркинда, создающего гораздо лучшую игру, потому что этот вундеркинд может украсть любого клиента на планете.

Отсутствие конкурентности означает, что игровой рынок напоминает турнир, в котором победителю достается все. Одна лучшая игра получает всех клиентов. Другие игры, даже если они очень хороши, не получают ни одного клиента, потому что все играют в лучшую игру.

Для победителя такой исход прекрасен. Это дает возможность захватить целый рынок. Сделайте это, и вы сможете заработать огромную кучу денег. Например, игра Call of Duty: Black Ops заработала больше, чем годовой ВВП Либерии (страна с населением 3,7 миллиона человек). И стала лидером среди шутеров от первого лица.

Но на турнирах есть и проигравшие. На самом деле в основном проигравшие. О них легко забыть, потому что проигравшие остаются в тени. Мы слышим о богатых победителях в обзорах и телешоу, а проигравшие умирают в темноте. Только если мы сами начнем искать информацию о проигравших, масштаб становится очевидным. Посмотрите на полный список игр, вышедших за определенный период, и вы будете поражены тем, как их много. И даже в этом примере есть систематическая ошибка выжившего, поскольку в нем не представлена никакая информация об играх, которые так и не вышли. Игровой турнир жесток, и большинство игр проигрывают.

Эффект Матфея

Конкуренция в геймдизайне усложняется благодаря *эффекту Матфея*, названному так социологом Робертом Мертоном в честь библейского Евангелия от Матфея 25:29: «...ибо всякому имеющему дастся и приумножится, а у неимеющего отнимется и то, что имеет». Преимущество популярной игры или франшизы заключается в том, что она становится еще популярнее. Богатые богатеют.

В играх эффект Матфея появляется по многим причинам. Во-первых, игры — это не просто неконкурентные блага. Это антиконкурентные блага, потому что чем больше других людей играют в игры, тем лучше те становятся.

Больше игроков означает более сильное сообщество, больше потенциальных игровых партнеров, больше пользовательского контента и культуры и больше сарафанного радио. Во-вторых, разработчики успешной игры имеют финансовые преимущества, авторитет и статус, которые помогают им создать свою следующую игру. Они могут заплатить, чтобы получить талантливого сотрудника, который им нужен. Они могут получить творческий контроль от издателей. Им доступны рычаги влияния на собственников платформ. В-третьих, потребители в значительной степени предпочитают знакомый продукт, а это значит, что уже популярное название будет иметь конкурентное преимущество перед чем-то неизвестным.

Эффект Матфея подразумевает мир, в котором жизнь богатых и бедных никак не меняется, как в романе Оруэлла «1984». Но в действительности доминирующие франшизы регулярно обходят какой-нибудь стартап, появившийся из ниоткуда. Потому что, хотя кажется, что система работает против них, у новичков свои преимущества, с которыми большие люди не могут сравниться.

Дилемма инноватора

Представьте себе дизайнера в стартапе. У него мало акционеров, боссов или сотрудников, которые сказали бы ему, что рискованная идея не выстрелит. Его имя не связано с каким-либо известным проектом, и никто от него ничего не ждет. Он ничего не теряет, и его возможности изобретать и вводить новшества абсолютно безграничны. Он работает на собственном творческом энтузиазме, и его ничего не останавливает.

Поэтому он пробудет что-то сумасшедшее. Он рискует, вкладывает в это свои сердце и душу. И результат становится хитом. Он зарабатывает огромные деньги, его студия растет, и он становится известным. Теперь темная сторона успеха берет верх.

Его со всех сторон начинают атаковать силы, препятствующие инновациям. Начинают звонить акционеры с требованиями квартальной прибыли. Растущий штат сотрудников закладывает ожидания, предположения и требования к специальным навыкам. Формируются и укореняются привычки. Постепенно рискованная, нестандартная студийная культура стартапа превращается в неповоротливую, тяжелую канцелярскую корпорацию. Хуже того, успешный дизайнер перестает задавать вопросы самому себе. Возможно, он начинает оценивать свои собственные идеи слишком высоко. Возможно, он боится перемен теперь, когда ему есть что терять. Возможно, ему просто лень заниматься изнуряющим умственным трудом.

ДИЛЕММА
ИННОВАТОРА — это
сложный выбор,
с которым сталкива-
ются действующие
лидеры: продолжить
вводить новшества,
отказавшись от своего
основного продукта,
или ждать, пока это
сделает кто-то другой.

В конце концов, успешный дизайнер и его студия теряют способность изобретать. Человека с творческим запалом, которому нечего терять, и желанием попробовать дикие идеи больше нет. Он переключается в режим стабильно доходного предприятия, пытаясь растянуть свой первоначальный успех навсегда. А затем он удивляется, когда появляется какой-то новый парень с сумасшедшей идеей и разбивает его в пух и прах, так же как он сам когда-то сделал.

Эта ситуация называется «дилемма инноватора».

Дилемма инноватора уравнивает эффект Матфея. С точки зрения маленькой студии размер и ресурсы стабильного конкурента могут показаться ошеломляющими.

Но большинство этих крупных компаний страдает от дилеммы инноватора. Они — неподвижные цели, переживающие старую победу, ожидающие снижения пика популярности.

Сегменты рынка

Турнир по геймдизайну разыгрывается в нескольких дивизионах, и в каждом выдаются разные призы. Вы можете попытаться выиграть большой дивизион и получить крупный приз, но тогда вам потребуются огромные инвестиции и вы столкнетесь с жесткой конкуренцией. Вы можете выбрать меньший дивизи-

зион с меньшим призом. Меньшие призы означают меньшую конкуренцию и меньшее вознаграждение.

Дивизионами конечно же являются сегменты рынка.

Существует сегмент рынка США, который хочет играть за солдат и стрелять в других солдат в соревновании высокого уровня.

Существует и сегмент японского рынка, который хочет играть с цифровыми пони и умиляться их внешности. Бывают сегменты, участники которого любят гонки, в которых люди являются фанатами стимпанка, сегменты людей с отличными навыками платформера и сегменты людей с глубокими знаниями профессионального баскетбола.

Размер сегментов рынка, на который нацелена игра, влияет на ее потенциальную окупаемость, потому что некоторые сегменты рынка больше по размеру и более прибыльны, чем другие.

Например, рассмотрим System Shock 2 и Half-Life. Игры были выпущены с разницей в один год. В обе играют от первого лица, а главным героем является одиночка, попавший в огромный объект, заполненный монстрами. Игры стали чрезвычайно популярны среди игроков и получили широкое признание рецензентов. Но несмотря на их сходство, Half-Life превзошла System Shock 2 более чем в десять раз. Разница была в выборе целевых сегментов рынка. System Shock 2 является более сложной игрой, поэтому она понравилась небольшому сегменту игроков, жаждущих трудностей и жесткой игры. Half-Life проще, легче и более ориентирована на экшн, поэтому она привлекла гораздо большую аудиторию. Каждая игра несомненно является победителем, но дизайнеры Half-Life играли в гораздо более крупной лиге и заработали больше денег.

Недостатком ориентированности на более крупный сегмент рынка является то, что он будет иметь большую конкуренцию, потому что это очень лакомый кусок.

Если бы Irrational Games сделали игру System Shock 2 более похожей на Half-Life, ее продажи могли бы фактически снизиться.

СЕГМЕНТ РЫНКА — это группа игроков, которая характеризуется своими интересами, имеющимися навыками, ценовым диапазоном, культурой, доступными технологиями и географическим положением.

Студия нацелилась бы на больший сегмент, но тогда ей пришлось бы напрямую конкурировать с Valve, Epic Games и id Software, которые на тот момент господствовали на рынке экшн-шутеров. Игроки в этом сегменте, возможно, решили, что они хотят просто играть в Half-Life, Unreal или Quake II. Поэтому Irrational Games не стали рисковать и выбрали менее крупный и конкурентоспособный сегмент RPG в жанре «сурвайвл-хоррор»¹.

Эти две силы — прибыльность крупных сегментов рынка и конкуренция, которую они привлекают, — склонны находить баланс. Крупные прибыльные сегменты рынка привлекают разработчиков, что увеличивает конкуренцию в этих сегментах и снижает их прибыльность. Со временем игровой рынок уравнивается, когда ни один сегмент не является более прибыльным, чем любой другой.

Неохваченные сегменты рынка

Если бы все понимали каждый сегмент рынка, то любой сегмент, который был бы более прибыльным, чем все остальные, привлекал бы разработчиков до тех пор, пока он приносил бы наибольшую прибыль. Конечным результатом был бы рынок в совершенном равновесии, где каждый сегмент рынка дает одинаковую прибыльность, и отсутствие возможности добиться результата лучше, чем средний.

Здесь модель ломается. В реальной жизни никто не понимает форму рынка в совершенстве. Выяснить, сколько людей в каком-либо сегменте, достаточно трудно (попробуйте быстро ответить, сколько людей любят игры про лошадей?). Сегменты рынка тесно переплетаются. Культура и технологии постоянно меняются, поэтому равновесие невозможно. В результате мы имеем постоянное наличие неохваченных сегментов рынка, предлагающих сверхприбыль тем, кто сможет на них выйти. Дело за малым — найти их.

В 1993 году у Уилла Райта, дизайнера хита 1989 года — SimCity, появилась идея. Он захотел создать игру — симулятор жизни, в которой игроки управляли бы семьей, строили дома, устра-

¹ От англ. to survive — выживать.

ивались на работу и воспитывали детей. Несмотря на то что за прошедшие годы были предприняты несколько попыток создать нечто подобное, например игра Little Computer People, которая вышла в 1985 году, рынка игр по управлению семьей не существовало.

Никто не хотел этим заниматься. «Первые несколько лет это была целая баталия внутри компании Maxis», — сказал Райт в более позднем интервью. Люди в студии даже называли это «туалетной игрой», потому что игрок управлял семьей персонажа, когда симы чистили туалеты. А критики игры провели исследование рынка, чтобы подтвердить свои сомнения. «У нас была фокус-группа еще в 1993 году, — сказал Райт, — и она не пошла. Она никому не понравилась, и это была худшая идея из четырех, которые мы тогда презентовали».

Кто был прав? У обеих сторон были свои аргументы. С одной стороны, игра может понравиться творческим игрокам, которые не жаждут ни с кем сражаться. С другой стороны, кто захочет чистить туалеты в компьютерной игре? В эпоху космических морпехов и фэнтезийных воинов управление семьей казалось слишком прозаичным. Вскоре руководство студии прекратило разработку «туалетной игры».

Но Райт не сдался. Он поддерживал проект на плаву, принимая любую помощь. Он нашел невостребованного программиста и пригласил его в проект. «Они хотели его сократить, — сказал Райт. — Я забрал его в свой “Черный ящик”, так сказать, и создал свою группу разработчиков». Прошли годы. Семейная игра, теперь известная как Doll House, влачила свое жалкое существование на фоне быстрого роста SimCity 2000, а затем SimCopter.

Наконец, подул ветер перемен, и Райт получил зеленый свет и возможность начать работу над проектом с полной командой. К моменту выхода игры Doll House Райт уже восемь лет работал над ней. И теперь она называлась The Sims.

Но даже тогда никто не знал, выйдет ли игра на рынок. Сколько людей захотят управлять семьей? Она была не похожа ни на что. Для подобных игр не было установленного сегмента рынка, поэтому все, что можно было сделать, — только предполагать.

Казалось, даже сам Райт не знал. «Я думал, что мы продадим миллион или... 50 копий», — сказал он.

Райт ошибся. Игра продала не 50 и не миллион копий. The Sims создали франшизу, которая продала более 100 миллионов копий. На момент написания книги это самая продаваемая франшиза для ПК в истории.

Оказалось, что сегмент рынка игр для управления семьей огромен, потому что он затрагивает традиционные категории. The Sims привлекает людей, которые никогда раньше не играли в нее, потому что она предлагает творческий опыт с низким давлением на игрока. Она также подходит игрокам, жаждущим жесткой игры, которые хотят ставить перед собой задачи заработать целое состояние или отдохнуть от стрельбы по инопланетянам. Сегмент рынка был огромным и полностью лишенным конкуренции. Таким образом, The Sims стала чрезвычайно прибыльной игрой.

Оглядываясь назад, это кажется очевидным. Конечно, такая игра понравится всем, от игроков до бабушек. Естественно, люди хотят отдохнуть от напряженной игры. Это выдающийся продукт в области симуляторов.

Но помните, что изначально все это не было очевидно. Фокус-группы, которые тестировали Doll House, просто ненавидели ее. Руководители ненавидели ее. Как и многие другие разработчики Райта. Даже сам Райт думал, что это может быть ерунда. К счастью, его старания окупились. Но так происходит не всегда.

Лучшая рыночная стратегия — найти сегмент, который недостаточно охвачен. Но это сложная и рискованная стратегия, потому что надежного способа определения неохваченных сегментов рынка не существует.

Любого, кто сможет найти неохваченные сегменты рынка, ждет огромная прибыль. В лучшем случае игра может оказаться единственной в своем роде на огромном сегменте рынка и получать всю прибыль себе.

Но найти такие незаполненные сегменты сложно и рискованно. Дизайнер, работающий в устоявшемся жанре, знает, кому он продает игру и как аудитория реагировала в прошлом. Стандарты для него определены; основные проектные решения уже приняты. Попытка создать новый сегмент рынка — это нечто другое. Здесь нет верных игроков. Нет никаких правил, которыми можно руковод-

ствоваться, или общепринятых норм, которым нужно следовать. Самое главное заключается в том, что невозможно узнать, сколько людей заинтересуются абсолютно новой игрой.

The Sims — прекрасный пример искусства геймдизайна, но одно только мастерство не гарантирует массового успеха. На каждую игру The Sims было много других не менее хорошо продуманных игр, которые так и не нашли свою аудиторию. Потому что такой успех требует больше, чем просто мастерство. Нужна готовность делать большие ставки, а для этого нужно много удачи.

Кривая ценности

Идея сегментов рынка звучит так, как будто мы можем разделить рынок на совершенно отдельные куски и навесить небольшой ярлык на каждый из них. Конечно же в реальной жизни все не так просто. Сегменты рынка пересекаются, и одна игра может нравиться многим сегментам рынка.

Давайте рассмотрим более точный пример того, как разные игры нравятся разным людям. Начнем с того, что мы разбиваем все, что предлагает игра, на рыночную стоимость.

Например, есть люди, которые играют в The Sims только для того, чтобы строить дома. Им не интересно управлять семьей. Они просто хотят экспериментировать с различными формами комнаты, обоями и интерьерами. The Sims нравится этим людям, потому что игра предлагает рыночную стоимость, которую я бы назвал творческим строительством дома.

Но The Sims также привлекает людей, которым не интересно строить дома. Рассмотрим еще одного игрока, который хочет сыграть в ролевую версию себя. Он создает своего персонажа и продолжает исследовать эффектную альтернативную жизнь, полную драмы, любви и трагедии, в одном из готовых домов, которые предлагает игра. Этот игрок реагирует на другую ценность, которую я назову «ролевая игра — симулятор жизни».

The Sims предлагает творческое строительство дома, ролевую симуляцию жизни и многие другие виды ценностей в одном

РЫНОЧНАЯ СТОИМОСТЬ
в геймдизайне —
это аспект игрового
опыта, который при-
влекает определенную
группу людей опреде-
ленным образом.

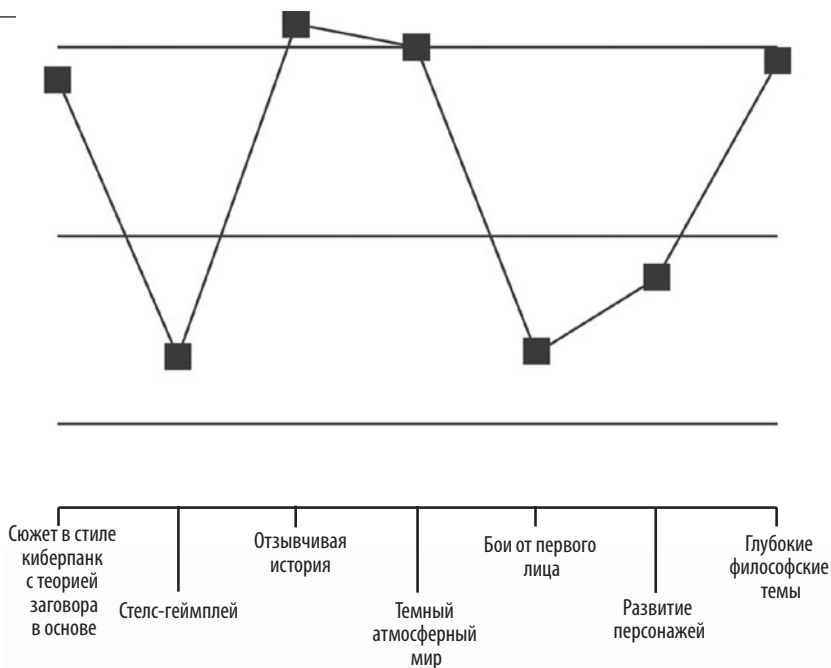
флаконе. Она не просто нравится одной группе людей с одинаковыми интересами. Она понравится любому, кто хочет получить любую имеющуюся ценность. И почти всем игрокам интересны несколько из них. Почти каждый, кто играл в The Sims, получил хоть какое-то удовольствие от творческого строительства дома и ролевой симуляции жизни.

Мы знаем, что наша цель — неохваченные сегменты рынка. Но недостаточно просто искать пустые сегменты, потому что сегменты пересекаются между собой. Чтобы точнее понять сегменты рынка, мы можем использовать кривую ценности.

КРИВАЯ ЦЕННОСТИ — это график, который сравнивает игры, оценивая их по различным видам рыночной стоимости, которые они предлагают игрокам.

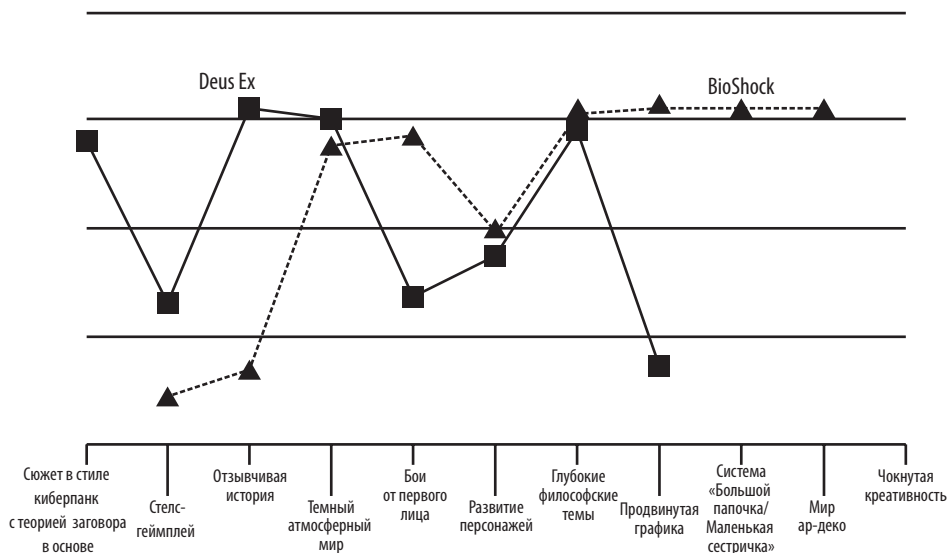
Кривая ценности показывает, как избежать конкуренции и создать неоспоримое рыночное пространство. Она помогает проанализировать, где две игры предлагают одинаковые, а где разные ценности.

Давайте рассмотрим на примере. Это кривая ценности игры в жанре киберпанк Deus Ex, которая вышла в 1999 году:



Игра Deus Ex предлагает много разных типов ценностей. Игрок, ищущий сюжет с теорией заговора в стиле киберпанк, отзывчивую историю, темный атмосферный мир или глубокие философские темы, найдет желаемое в этой игре. Игра также предлагает геймплей в жанре «стелс», бой от первого лица и развитие персонажей, но не до очень высокого уровня.

Но кривая ценности только одной игры приносит мало пользы. Кривая ценности — это инструмент для сравнения, что предполагает кривые для нескольких игр. Вот тот же график с кривой для BioShock.



Теперь мы можем точно определить, какие ценности являются уникальными для каждой игры, какие пересекаются, а какие лучше всего обеспечивает каждая игра. Обе игры предлагают геймплей в жанре «стелс», но истинный фанат этого жанра получит больше от Deus Ex, поскольку ее жанр глубже и лучше сбалансирован. Бой от первого лица в Deus Ex слабее, чем в BioShock, поэтому любители шутеров предпочтут BioShock.

Мы также находим некоторые ценности, которые полностью отсутствуют в той или иной игре. В BioShock нет никакого сюжета с теорией заговора в жанре «киберпанк». А система «Большой

папочка/Маленькая сестричка» и ар-деко не являются частью Deus Ex.

Чтобы провести более широкий анализ рынка для новой игры, мы строим подобную кривую для каждой аналогичной игры на рынке. На финальном графике может быть представлено пять или шесть игр, и у каждой из них существуют ценности, которые преобладают над ценностями других игр. Такой анализ обеспечивает наглядность, показывая нам ценности в нашей игре, которые действительно имеют значение.

Ценности, которые новая игра предлагает на более низком уровне, чем существующие игры, не стоят много. Игроки, которым нужны эти ценности, смогут найти лучшие варианты в другом

Важными ценностями в игре являются те, которые превосходят все остальные предложения на рынке. Любая другая ценность может быть лучше обеспечена другим продуктом.

месте. Важны те ценности, которые в новой игре будут лучше, чем у всех конкурентов. Люди, которые хотят получить их, будут играть в эту игру, потому что они больше нигде не смогут получить их на таком уровне.

Но самая большая победа — это ценности, у которых вообще нет конкуренции. Например, рассмотрим уникальное позиционирование BioShock на рынке. Многие игры предлагают бои от первого лица, геймплей в жанре стелс, отзывчивые истории и темные атмосферные миры. Конкурировать на этих ценностях сложно, потому что рынок перенасыщен. Таким образом, вместо того чтобы конкурировать на стандартных ценностях, BioShock изобрел совершенно новые ценности, которые больше никто не предложил. Мир в стиле ар-деко и завораживающая система «Большой папочка/Маленькая сестричка» приводят в восторг и не похожи ни на что другое на рынке. Игроки, которые хотели этого — а многие хотели, — должны были перейти на BioShock, потому что больше нигде подобного не было. Поэтому игра стала очень успешной.

Сравнение с помощью кривой ценностей наглядно показывает, какие части игры важны, а какие нет. Это основные и уникальные ценности, которые действительно имеют значение; ценности, которые лучше найти в других играх, нецелесообразны. Кривая помогает дизайнерам, показывая им, где нужно сосредоточить свои усилия по разработке.

Сосредоточение на ценностях

Каждая ценность стоит ресурсов. Вот почему BioShock не пытался создать такой же глубокий нарратив, как в Deus Ex; это обошлось бы разработчикам слишком дорого. Кроме того, каждая новая ценность снижает эмоциональную полноту опыта. Слишком много ценностей увеличивают риск того, что игра превратится в малопонятный, слишком сложный беспорядок. Наконец, некоторые ценности просто несовместимы. События в BioShock разворачиваются в подводном городе в 1960 году, поэтому игра просто не может включать в себя заговор в духе киберпанка и сохранить при этом связный нарратив.

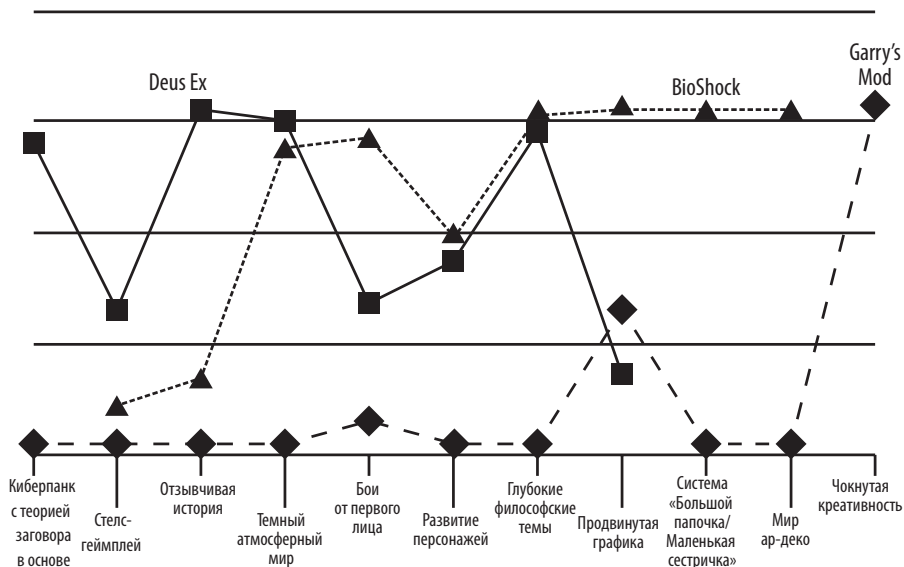
Поэтому достижение ценностного превосходства на рынке — это не просто стать лучше или работать усерднее. Речь идет о перемещении ресурсов и сосредоточении на том, где они могут принести наибольшую пользу. Самые крутые игры не пытаются делать все сразу; они делают немного, но очень хорошо.

Чем меньше у дизайнера ресурсов, тем больше ему нужно сосредоточить свои усилия на меньшем количестве ценностей. Конкурируя на меньших ценностях, он увеличивает свои шансы на победу. Инди-игра, созданная двумя разработчиками, может предложить только одну или две ценности, но если они будут лучше, чем у кого-либо другого, она все равно может привлечь игроков и быть прибыльной.

Например, Garry's Mod — это простая игра от первого лица, которая бросает игроков в физическую «песочницу» с неограниченным набором странных инструментов. В Garry's Mod игроки могут делать что угодно. Они могут создать любой объект или персонажа, соединять их с помощью веревки и клея, настраивать триггеры, чтобы активировать ракетные двигатели, воздушные шары или взрывчатку.

Игроки создают дикие хитроумные приспособления, вроде летающих колясок на динамите, и странные выдуманные миры из листового металла и бомб. Они создают персонажей в забавных ситуациях, делают снимки экрана и комиксы для размещения в интернете.

Вот общий график для Deus Ex, BioShock и Garry's Mod:



Garry's Mod — игра от первого лица, но она не конкурирует с BioShock или Deus Ex ни по одной из их ключевых ценностей. Вместо этого она обходит их, предлагая нечто новенькое: странную творческую деятельность. Ни в Deus Ex, ни в BioShock, ни в любой другой крупной игре невозможно делать то, что предлагает Garry's Mod. Если бы разработчики попытались напрямую конкурировать с играми, в 100 раз превышающими бюджет Garry's Mod, эта игра, безусловно, потерпела бы неудачу. Вместо этого они выделили свое собственное небольшое рыночное пространство, выдвинув единственную уникальную ценность. И спустя полгода практически единоличного владения своим собственным сегментом рынка Garry's Mod продали более миллиона копий. Даже при стоимости от 5 до 10 долларов это весьма неплохо для игры, созданной одним человеком.

Глядя на список успешных небольших игр, мы видим, что почти все они сосредоточены на одной или двух ценностях, которые недооцениваются большими играми. Их кривая ценности имеет высокие узкие вершины. Вот список некоторых из них.

Игра	Первичная стоимость на рынке	Вторичная стоимость на рынке
Super Meat Boy	Сверхскоростная платформа	Классные персонажи и ссылки на ретроигры
Dwarf Fortress	Гиперглубокий эмергентный нарратив	Креативное строительство крепости
Counter-Strike	Жесткий тактический бой FPS	Борьба с терроризмом
Minecraft	Творческое создание мира	Неограниченная разведка
LIMBO	Мрачная атмосфера	Платформер-головоломка
Castle Crashers	Жанр Beat'em up	Забавный мультяшный арт-стиль
Braid	Головоломки на время	Поэтический аллегорический нарратив

Если мы посмотрим на провальные игры, то обнаружим обратное: короткая, широкая кривая ценности, напоминающая пеньки, без единой ценности, которая превосходила бы ценности конкурентов. Эти игры просто посредственные версии игры с большим бюджетом. У такой игры нет причин существовать, а у игроков нет оснований играть в нее, потому что они могут получить лучшее решение в другом месте.

Обычно такая кривая ценности является результатом наивных чрезмерных амбиций. Дизайнер видит другую игру, которая ему нравится, и решает, что он хочет сделать то же самое, но лучше. Но не имея большого количества ресурсов для разработки, он почти гарантированно потерпит неудачу. Он берется за все и сразу и в конечном итоге не дает никакой уникальной ценности.

Я сам совершал подобную ошибку. Увлечшись геймдизайном, я создал мод под названием Elemental Conflict. Мой план заключался в том, чтобы создать усовершенствованную футуристическую версию чрезвычайно популярной игры Counter-Strike. Поэтому я сохранил такую же структуру раундов и команд, экономику и баланс оружия. Я пытался дифференцировать игру, используя уникальные футуристические боевые инструменты, такие как рюкзаки, ракеты-носители и клеевые гранаты. Но я ничего не убрал. Я допустил классическую ошибку, пытаясь скопировать удачную игру и украсив ее новыми блестящими безделушками. Результатом стал самый предсказуемый провал на рынке. Спустя 14 месяцев работы шести разработчиков и нескольких выпущенных версий у нас был ровно один публичный сервер, и никто на нем не играл. Основной геймплей работал достаточно хорошо, но он не был так гладко отшлифован, как

Counter-Strike. У нас не было ничего достаточно уникального, чтобы победить эффект Матфея.

Соотнесите свои амбиции с вашими ресурсами. Управляя большой группой разработчиков, вы можете попробовать одну уникальную ценность, две основные ценности и совокупность менее значимых ценностей, чтобы развить опыт. Работая в небольшой команде, вам, возможно, придется сосредоточиться только на одной ценности, исключив все остальное. Стремитесь быть лучшим в чем-то одном, чем посредственным во всем.

Никто ничего не знает!

Мы используем такие модели, как сегментация рынка и кривая ценностей, чтобы попытаться понять рынок. Но эти модели не являются реальностью — это просто наши попытки перенести поразительную сложность жизни в то, что можно изобразить на графике. Они помогают нам думать о рынке, не перегружая мозг, но не приближают нас к пониманию его истинной сложности. Руководствуясь результатами такого анализа, легко почувствовать себя слишком уверенно. На самом деле это всего лишь ничтожные попытки понять что-то невероятно и неизмеримо сложное.

Рынок — это все. Это миллиарды людей, все их взаимоотношения, привычки, культура и технологии, а также физический мир вокруг них. Он продиктован стихийными бедствиями, интернет-мемами, модой, политическими тенденциями, бизнес-моделями, технологиями, индивидуальным выбором и случайностью, а также сам их диктует.

Большая часть того, что происходит на рынке, выходит за рамки любой модели, которую мы можем применить. Знаменитость, которая играет в игру и она ей нравится, новость, которая привлекает внимание к определенной теме, или политик, решающий проблему, могут кардинально изменить конъюнктуру игрового рынка. Это не простые линейные величины, которые мы можем просчитать. Это косвенные причины, на которые могут дополнительно повлиять социологические эффекты снежного кома и в результате привести к событиям, которые изменят мир.

Вспомните, например, StarCraft — RTS (стратегия в реальном времени), хит, разработанный компанией Blizzard в 1998 году. StarCraft — очень хорошая игра, и она была очень успешной. Но в Южной Корее она вышла за рамки простого успеха. За 10 лет в Южной Корее было продано 5 миллионов копий StarCraft. Это одна копия на каждые 10 мужчин, женщин и детей в стране, почти половина мировых продаж игры. И даже эту цифру можно считать заниженной, поскольку многие азиатские игроки играют в сетевых кафе и сами не купили игру. В разгар популярности игры два телевизионных канала транслировали профессиональные игры StarCraft, а лучшие профессиональные игроки стали местными знаменитостями.

Никто не прогнозировал такую популярность, да и никто не мог. Нарратив игры повествует о космических «реднеках»-землянах, воюющих против роя Зергов и телепатической инопланетной расы Протоссов. На первый взгляд казалось, что эта американская задумка не будет хорошо принята в корейской культуре. Разработчики не приложили никаких усилий, чтобы адаптировать игру под корейский рынок — корейский язык появился в игре только спустя семь лет после ее выхода.

Корейское чудо со StarCraft произошло по нескольким необычным причинам, которые в совокупности сформировали снежный ком популярности. Первой причиной был феномен PC bang. На корейском языке bang — это «комната»; PC bang — это высокотехнологичное игровое интернет-кафе. В середине-конце 1990-х годов правительство Южной Кореи предприняло активные усилия по модернизации инфраструктуры доступа к интернету, а благодаря небольшому размеру страны и высокой плотности населения удалось достичь быстрого прогресса. В то же время закрылись многие из старых государственных социальных и пенсионных программ, в результате большое количество пожилых граждан остались без работы и не могли выйти на пенсию.

Тысячи этих 50-летних пенсионеров воспользовались лучшей возможностью, которую только могли найти, и открывали игровые кафе. Это сработало, поскольку такие кафе требовали относительно небольших вложений и являлись надежными источниками дохода.

Физические и культурные различия между странами делают такой бизнес гораздо более привлекательным в Корее, чем на Западе. Корея представляет собой густонаселенную страну, дома большинства жителей маленькие, поэтому молодые корейцы проводят намного больше времени вне дома, чем жители Запада. Благодаря появлению интернет-кафе молодые корейцы менее чем за доллар в час могут получить мгновенный доступ к любым играм, при этом их не нужно покупать, хранить или искать место для домашнего ПК.

Эта совокупность факторов привела к нарастающей популярности. Большое количество интернет-кафе означало большее количество игроков, что сделало интернет-кафе более социальными. Тот факт, что игра в интернет-кафе является публичной и социальной, означал, что информация о StarCraft может распространяться напрямую через живое общение, что невозможно на Западе, где все играют у себя дома. В конце концов, культура пересекла ту границу, когда молодые люди привыкли к бесконечным тусовкам в интернет-кафе в кругу друзей.

Появление одних только интернет-кафе могло бы сделать игру популярной, но не смогло бы превратить ее в малый национальный спорт. Последним, что подтолкнуло StarCraft на вершину, стали игровые телевизионные каналы. Но эти каналы не возникли из ниоткуда. Традиционная азиатская настольная игра Go уже имела огромное количество поклонников в Корее, и ей было посвящено несколько каналов. Каналы Go показали, что создать канал, посвященный видеоиграм, не так уж сложно. Если бы не было каналов Go, каналы StarCraft могли бы никогда не появиться.

Снежный ком увеличивался — рост культуры интернет-кафе, новая широкополосная инфраструктура, незначительный экономический спад, сделавший дешевые развлечения привлекательными, и телевизионные каналы — все это укреплялось в устойчивый культурный феномен. Ни один из этих факторов не был предсказан, и никто не смог бы этого сделать. Все они находятся за пределами типичных рыночных моделей. Но вместе они превратили игру в мегаяхит.

И даже эта красиво упакованная история — огромное и, возможно, вводящее в заблуждение упрощение. Даже сейчас никто не понимает важность этих различных факторов, которые привели к стремительному росту StarCraft в Корее. Что было бы, если в стране не было бы рецессии? Что было бы, если бы правительство не открыло широкополосную связь в середине 1990-х годов? Что было бы, если бы StarCraft была фантастической или реалистичной игрой? Мы можем только догадываться. История написана складно, но никто не сможет догадаться, насколько она действительно соответствует миллиардам индивидуальных экономических решений, которые привели к фантастической популярности StarCraft в Корее.

Если бы игра вышла на год раньше или позже, она могла бы стать просто еще одной популярной игрой от Blizzard. Люди могли бы поиграть в нее, насладиться и пойти дальше. Она принесла бы прибыль, и ее бы называли большим успехом, и никто никогда бы не узнал, как близко она подошла к чему-то большему, чем просто игра в маленькой азиатской республике.

Сценарист Уильям Голдман однажды сказал о таких совершенно непредсказуемых культурных явлениях: «Никто ничего не знает!» Голдман всю свою жизнь наблюдал за тем, как руководители, писатели и режиссеры пытаются и не могут предсказать кассовые сборы фильмов. Так же точно происходит и с играми. Хотя мы можем использовать модели, чтобы получить больше, чем ничего, не следует забывать о том, что мир сложнее, чем любая наука об исследовании рынка.

Склонность к подтверждению своей точки зрения

Обратимся к маркетингу. Это книга об искусстве, а не о продажах, поэтому я не стану подробно рассказывать о маркетинге. Однако остается один аспект, который должны учитывать все дизайнеры, так как он сильно влияет на сам геймплей. А именно то, как маркетинг определяет ожидания и как эти ожидания влияют на геймплей.

Знаете ли вы, что вкусное пиво можно испортить плохим описанием? Психолог Дэн Ариели обнаружил это в ходе эксперимента в бостонском баре. Он предложил посетителям паба два одинаково выглядящих напитка. Первым было пиво Samuel Adams, популярное, но довольно стандартное бостонское пиво. Вторым был секретный напиток MIT Brew — тот же Sam Adams с добавлением нескольких капель бальзамического уксуса. В результате слепого исследования MIT Brew был признан явным победителем. Но если Ариели сначала говорил испытуемым, что собой представляет MIT Brew, после дегустации они говорили, что вкус отвратительный. Уксус в пиве — на вкус звучит ужасно, следовательно, люди ожидали, что пиво будет ужасным, и поэтому они так его и воспринимали, хотя на самом деле это был превосходный напиток.

Ожидания не отделены от восприятия. Скорее мы склонны подтверждать наши прежние убеждения. Этот наблюдается во всем, а бесчисленные исследования открыли различные формы этого

СКЛОННОСТЬ К ПОД-
ТВЕРЖДЕНИЮ СВОЕЙ
ТОЧКИ ЗРЕНИЯ — это
тенденция людей
воспринимать вещи
таким образом, кото-
рый подтверждает их
прежние убеждения.

эффекта. Продукты в дорогих контейнерах воспринимаются более вкусными, а продукты с высокими ценами — лучше. Pepsi побеждает кока-колу в Pepsi Challenge, но только если этикетки скрыты. Джошуа Белл, один из лучших скрипачей в мире, как-то сыграл инкогнито в нью-йоркском метро, и почти все его проигнорировали. Работа шарлатанов по исцелению, заклинателей собак, гадалок и прочих жуликов зависит от желания других людей подтвердить свою точку зрения. Эффект плацебо немного больше, чем желание убедиться в своей правоте.

Склонность к подтверждению своей точки зрения наблюдается в играх повсеместно. Эту тенденцию можно заметить в каждом обзоре, каждой рекомендации и игровом опыте игрока. Игроки начинают формировать мнения, как только слышат об игре. К тому моменту, когда они начинают играть, их устоявшиеся мнения и ожидания уже влияют на их опыт. Если вы услышали, что игра художественная, вы будете искать подтверждение и найдете его. Когда все говорят, что игра страшная, вы заметите и запомните каждый раз, когда по вашему телу пробежут мурашки. Если на игру вышли плохие обзоры, вы будете сосредоточены

только на плохих моментах. Никто не играет в вашу игру просто так; люди играют через призму того, что они уже знают о ней. Отсюда следует, что дизайнер не может игнорировать маркетинг.

Склонность к подтверждению своей точки зрения может даже повлиять на то, как мы интерпретируем событие в игре. Например, представьте, что в обзоре говорится о том, что сложность задач в игре доставляет удовольствие. Вы покупаете игру. Игра разочаровала вас, вы несправедливо (на ваш взгляд) проигрываете. Но вы не вините в поражении игру, потому что вы это уже ожидали. Вы были настроены на то, чтобы интерпретировать проигрыш как намеренно созданную часть опыта, а не как свою ошибку. Если бы до того, как начать играть, вы бы прочитали обзор, в котором игра была бы описана как глупая и несправедливая, вы, скорее всего, истолковали бы свой проигрыш совсем по-другому.

Определение ожиданий

Для дизайнеров такой феномен, как склонность к подтверждению своей точки зрения, означает, что определение ожиданий имеет решающее значение. Ожидания определяются во многих отношениях, но мы можем контролировать только некоторые из них.

Название игры часто является первым элементом, который определяет ожидания игроков. Рассмотрим следующие названия: Doom, SimCity, Dark Souls, The Marriage, LIMBO, Fable, Condemned, Brain Challenge, Mortal Kombat и RollerCoaster Tycoon. Некоторые звучат зло и жестоко, другие — легко и дружелюбно. Одни звучат художественно, а другие — коммерчески. Каждое название намечает свою цепочку психических ассоциаций и задает разные ожидания.

Далее идут маркетинговые сообщения. Реклама, интервью и статьи — все это задает ожидания. Реклама может фокусироваться на персонажах или действиях. Они могут быть хорошими, плохими, быстрыми или медленными. Если в интервью разработчик сказал, что на создание игры его вдохновила любовь к садам и детям, у игроков будут другие ожидания, в отличие от ситуации, если бы он сказал, что источником вдохновения послужили фильмы ужасов с элементами порно. Если он говорит, что в игре остается

момент, который заставит вас плакать, игроки будут ожидать этого момента.

Наконец, игроки создают ожидания посредством сарафанного радио. Это самый мощный вид ожиданий, потому что он усиливается социальным давлением. Люди — это социальные животные; мы естественным образом синхронизируем свое мнение с мнением окружающих. Когда мы с кем-то обсуждаем игру, мы не просто высказываем о ней свое мнение. Мы ждем, что о ней скажут другие, особенно если это люди с высоким статусом, и меняем свое мнение, чтобы соответствовать. Это не значит, что мы врем. Наше мнение на самом деле меняется — мы корректируем свои воспоминания в соответствии с общим взглядом. Этот эффект проявляется в небольшом масштабе среди друзей и в большом масштабе во всем игровом сообществе. Игроки цитируют рецензентов, которые цитируют друг друга. Первый обзор игры может создать утверждение, которое будут копировать и переписывать до тех пор, пока оно не станет частью культурного нарратива об игре, даже если оно с самого начала было случайным описанием.

Мы не можем контролировать сарафанное радио, но зато мы можем влиять на заголовки и маркетинговые сообщения. Подумайте о том, какие ожидания вы хотите задать, чтобы подготовить игроков к наилучшему игровому опыту, и придумайте соответствующие названия и маркетинговые сообщения. Если у вас работают маркетологи, не игнорируйте их. Сами по себе маркетологи не могут знать, какие части игры необходимо представить определенным образом, а о каких стоит умолчать. Дизайнеры должны рассматривать маркетологов как часть основной команды разработчиков, а не как бесплатное приложение — их решения влияют на игровой опыт не меньше.

ЧАСТЬ III

Процесс

До сих пор мы рассматривали профессию геймдизайнера. Мы рассмотрели ментальные модели, которые помогают нам понять, как работает дизайн и как его изменить, чтобы он стал лучше. Но чтобы создать хорошую игру, одного мастерства недостаточно.

Что делать геймдизайнеру изо дня в день, час за часом? Когда проводить мозговой штурм, программировать, обсуждать задачи или делать перерыв? Что планировать, на какой срок планировать и как записывать эти планы? Что обсуждать, с кем и как? Что делать, когда команда выросла и вместо одного человека теперь их более ста?

Если у вас нет хороших ответов на эти вопросы, вся ваша работа будет бесполезной, потому что она будет сосредоточена на неправильных задачах. Мы будем атаковать несуществующие проблемы, строить воздушные замки, основанные на мечтах, создавать ненужные технологии и сталкиваться с постоянным непониманием. В конце концов, нашу игру задушат бюрократия, гнев и недопонимание. Мы будем заниматься сизифовым трудом, усердно делая то, что не нужно.

Весьма трудно этого не делать. Со мной так случалось тысячи раз. Я был одержим задачами программирования, тратил на их решение несколько дней, даже несмотря на то, что новая функция практически не влияла на игровой опыт. Я создавал технологию, которую почти наверняка потом вырезали. Я тестировал тогда, когда нужно было компилировать, и компилировал, когда нужно было тестировать. Сизифов труд — это в том числе еще и групповая деятельность. Я просил работу, не уточняя своих намерений и заставляя других тратить время на решение ненужной задачи. Организовывал ненужные совещания и пропускал действительно важные. Спорил, когда должен был согласиться, и соглашался, когда должен был спорить. Говорил слишком много, говорил недостаточно и не говорил совсем.

Решить проблему сизифова труда не так просто, потому что контролировать ее самостоятельно практически невозможно. Когда разработчику не удастся решить задачу, результат очевиден. Кто угодно может сказать, что код не работает или что в игре ужасная графика или непонятный дизайн. Подобные ошибки вызывают

немедленную эмоциональную обратную связь, поэтому для нас естественно усовершенствовать свои навыки, когда это необходимо. Сизифов труд — другой случай. Если коротко, это как бушующий гений. Нам хорошо, когда мы решаем задачу, но наше эмоциональное бессознательное не сигнализирует о том, что задача, которую мы решили, была неактуальной. Таким образом, мы решаем одну задачу за другой, радостно, усердно, глупо, не понимая, что выбрали неправильные задачи. Последствия таких ошибок проявляются намного позже, и проследить взаимосвязь между причиной и следствием часто невозможно. Именно поэтому мы можем радостно повторять одни и те же ошибки снова и снова в течение многих лет или десятилетий.

Некоторые считают, что разработка игр — это путь, по которому мы идем к месту назначения. Я с этим не согласен. Я считаю, что это больше похоже на темный лес, полный жутких монстров, ожидающих момента, чтобы укусить своими ядовитыми зубами. Каждый раз, когда вы сталкиваетесь с одним из них, он кусает вас, а яд медленно растекается по вашему телу, заставляя вас чувствовать тепло и удовлетворение. Но с каждым таким укусом вы теряете энергию, яд уничтожает вас изнутри. И только потом, когда ваша сила иссякнет и луна выйдет из-за туч, вы поймете, что приятное чувство, которым вы наслаждались все это время, был вовсе не прогресс. Это была смерть.

Эта часть о том, как найти свой путь в темноте.

Проблема убеждений

Хотел бы я просто разложить для вас все по полочкам. Я бы просто перечислил пункты, которым следовала бы каждая студия, чтобы создать игру. Их было бы просто написать, на них было бы удобно ссылаться, и все были бы согласны.

Но я не могу этого сделать, потому что никто еще не разгадал процесс разработки игры. Нет единого пути, который работает для всех. Скорее существует целое множество методов.

Большинство из них регулярно показывают низкую эффективность. Иногда кажется, что почти все игры не укладываются

в сроки разработки и превышают бюджет. Работа отправляется в мусорное ведро по политическим соображениям компании или из-за недоразумений либо остается из-за страха. Для решения проблем выделяют все больше и больше денег, создавая при этом еще больше проблем. Игровая индустрия переполнена историями о кранчах, рабочих марафонах по 10 или 12 часов в день каждый день в течение месяцев или даже лет. Люди толстеют, скучают, не замечают, как вырастают их дети, или сгорают и вообще уходят из индустрии. Этот кошмар разрушает личность и тушит творческую энергию.

И мы не понимаем, почему все идет не так, ведь мы же все делаем правильно. Нанимаем умных разработчиков. Мотивируем их хорошо продуманными стимулами. Даем им нужные ресурсы, правильные данные о рынках, правильные технологии. Планируем каждую часть продукта в мельчайших деталях месяцы или годы. Но опять не получается. Почему?

Виной этому наши убеждения.

Любому человеку легко говорить о необходимости подвергать сомнению убеждения. На деле это гораздо сложнее. Реальные убеждения-убийцы — это те, которые заставят вас с головой уйти в решение сложных задач на полгода и пропустить первый день

Ошибки в геймдизайне обычно возникают в результате глубоко укоренившихся убеждений, о которых мы сами не подозреваем.

рождения вашего ребенка, это те убеждения, которые заложены настолько глубоко, что практически неприкасаемые. Они защищены толстым культурным слоем, привычками и личной выгодой. Они позволяют вам использовать все ваши старые навыки и инструменты и не требуют, чтобы вы изучали что-то новое. Если бы можно было так легко разрушить старые убеждения, все бы тут же это сделали.

Убеждения не обязательно ошибочны. Ставить их под сомнение не означает опровергать все наши убеждения — это просто означает идентифицировать их и убедиться, что они основаны на истине, а не на привычке.

В играх наши убеждения исходят из двух ключевых источников. Во-первых, каждый раз, когда мы заимствуем концепцию из другой области, к ней автоматом добавляются скрытые убеждения.

Во-вторых, мозг человека биологически связан с убеждениями. Давайте рассмотрим каждое из них.

Заемствованные убеждения

Почему мы говорим, что игра находится на стадии препродакшн? Почему бета-версия? Почему разработкой руководит директор? Почему в играх продюсеры, а не логисты или аллокаторы? Почему в игровых командах есть младшие дизайнеры, а не ассистенты, ученики дизайнеров или рядовые? Почему директор, а не капитан, главный редактор, тренер или шеф-повар? Почему мы не говорим, что существует черновик?

Каждое из этих слов взято из другой области. Вопрос в том, имеют ли эти слова смысл в разработке игр. Обычно ответ будет отрицательный.

Проблема заключается в том, что эти термины были разработаны для решения проблем, которые отличаются от наших. Часто в них заложены убеждения, которые верны в исходной области, но не в играх.

Возьмем, к примеру, такие термины, как препродакшн, продакшн и постпродакшн. Они были заимствованы из киноиндустрии. Они развивались в этой среде, потому что кинопроизводство неразрывно связано с короткими периодами чрезвычайно дорогой съемки. Съемка фильма может стоить тысячи долларов в минуту, поэтому режиссеры научились сжимать этот дорогостоящий период производства в несколько недель. Для них разделение процесса на три части имеет смысл, поскольку у периода определенные начало и конец. Все в их процессе сводится к минимизации этой средней части так, чтобы не приходилось оплачивать лишний час работы кейтеринга, а также сотни светооператоров и постановщиков.

В играх все совсем иначе. В случае если студия переходит от стадии препродакшн к стадии продакшн... то ничего не меняется. Никого не нанимают и не увольняют, не задействуются никакие значительные ресурсы, а одни и те же старые совещания проходят в одно и то же время каждую неделю. Ничего не меняется,

потому что в разработке игр нет очень дорогой средней части. Итак, когда мы говорим «предпродакшн», «продакшн» и «пост-продакшн», что мы имеем в виду на самом деле? Режиссеры знают, что они имеют в виду. В их среде это очевидно. Для нас эти слова без дополнительного объяснения значат очень мало.

Вы можете сказать, что это всего лишь слова. Они помогают людям общаться. В чем проблема? А проблема в том, что эти слова несут в себе скрытые смыслы. Например, они подразумевают, что до начала стадии продакшн должен быть составлен сценарий или план. Они подразумевают, что сценарий, написанный до стадии продакшн, может дойти до самого конца. Они подразумевают, что на разных этапах мы должны нанимать разных людей. Мы знаем, что эти факты справедливы для кино. А как же геймдизайн? Возможно, да, а возможно, и нет. Необходимо пролить свет на эти укоренившиеся убеждения и подвергнуть их сомнению. И если слово содержит больше неправильных убеждений, чем правильных, возможно, следует от него отказаться.

В геймдизайне заимствованные слова и понятия процветают, потому что процессы, обычно используемые в разработке игр, никто не планировал. Они вошли в норму по мере накопления привычки в течение десятилетий. Первые игры были написаны одним человеком. Современные игры могут быть сделаны сотнями разработчиков. Со временем команды разработчиков становились все больше и больше, а процессы становились все сложнее. Каждый раз, когда команда увеличивалась, кто бы в ней ни появился, он просто вносил свои поправки, чтобы система работала. Эти люди искали самое простое и очевидное решение, которое было бы понятно большинству. Почти во всех случаях это означало заимствование терминологии из кино, программного обеспечения или промышленности.

Мы можем видеть этому подтверждение в огромном разнообразии технологий, используемых в разных компаниях. Каждая игровая студия работает по-своему. Там нет стандартизированных процессов, потому что никто еще не написал действительно стоящие процессы. Методы, которыми все пользуются, — это знания других людей. Это общие нормы, которые появились в результате многочисленных проб и ошибок. И, как и любые

общие правила, они произвольно меняются в зависимости от ситуации.

К счастью, сейчас ведется активное обсуждение проблемы заимствованной терминологии. Лучшие студии заменяют заимствованные методы новыми процессами, изначально разработанными для решения уникальных задач игровой индустрии. Впрочем, это медленный процесс, и многие заимствованные процессы и терминология остаются.

«Врожденные» убеждения

Человеческий мозг рассчитан на решения проблем пещерного человека. У нас появляются представления о мире, благодаря которым мы не попадаем в лапы тигров и можем ловко ориентироваться в племенной политике. И для пещерного человека это работает.

К сожалению, геймдизайнеры сталкиваются с несколько иными проблемами, чем были у наших предков, в то время как принципы работы нашего мозга не изменились. В современном мире убеждения проявляются в поведении в виде когнитивных искажений — иными словами, люди допускают последовательные, предсказуемые ошибки в восприятии или суждении. Например:

Гало-эффект означает, что мы не можем судить о различных свойствах вещи по отдельности. Если мужчина красив, мы склонны считать его более надежным. Если игровой персонаж лучше прорисован, мы думаем, что им лучше управлять. Человеческий разум склонен классифицировать вещи на хорошие или плохие. Для геймдизайнеров это смертельная ошибка, потому что наша работа требует, чтобы мы разложили каждую игровую систему на составные части, поняли, как они совмещаются и какой вклад вносят в контексте получения окончательного опыта. Любить и ненавидеть каждый элемент дизайна целиком — это ленивый путь заблуждения. Всегда остается какая-то ценность, и всегда есть оптимальное соотношение.

Боязнь потерь приводит к тому, что наш страх потерять что-либо становится сильнее, чем желание выиграть. Это заставляет разработчиков игр цепляться за старые идеи, вместо того чтобы исследовать новые концепции дизайна. Со временем это может привести к иррациональному усилению, так как разработчики годами тратят деньги впустую, не могут признать, что идея провалилась, отказаться от нее и двигаться дальше.

Эвристика доступности заставляет нас реагировать только на то, что мы можем воспринимать или представить, и игнорировать то, чего мы представить не можем, как если бы этого даже не существовало. Лауреат Нобелевской премии психолог Даниел Канеман называет это WYSIATI, *что ты видишь, то и существует*. Вот почему люди всегда больше думают о том, как выжить в случае повторного теракта, и не думают о других потенциальных опасностях. Поскольку атака уже произошла и вызвала сильные эмоции, мозг весьма легко может ее воспроизвести. Ее можно представить, снова испытать страх и отреагировать. Между тем человек не думает о других потенциальных опасностях, хотя они не менее вероятны. Поэтому относится к ним так, как будто их даже не существует.

Эвристика доступности постоянно проявляется в геймдизайне, потому что игровые системы и игроки часто делают вещи, которые нельзя заранее представить. В итоге мы рассматриваем игру так, как будто она делает только то, что мы видели, и не воспринимаем ее как систему, которая обычно может делать больше, чем то, что мы видели когда-либо.

Вот почему дизайнеры, отвечающие за баланс, напрасно стараются как можно тщательнее исправить последнюю ошибку баланса, которую они заметили. Эта ошибка может быть одной среди сотен других аналогичных ошибок в игре, но поскольку она более всего доступна для мозга, он будет исправлять ее так, как будто она единственная в своем роде. То, что вы видите, то и существует.

Существуют сотни других подобных предубеждений. Я не буду больше о них писать в этой книге. Чтобы узнать о них больше, посмотрите список рекомендуемой литературы в конце.

До некоторой степени мы можем бороться с такими отклонениями, не давая возможности эмоциональному брать верх над рациональным. Но мы не можем полностью устранить наши предубеждения, потому что мы всегда остаемся людьми.

Что мы можем сделать, так это выбрать процессы, которые минимизируют влияние когнитивных отклонений. Мы можем создать социальные структуры вместе с системой сдержек и противовесов и следовать процедурам, которые обходят наши личные отклонения. За пределами геймдизайна примерами такого рода процессов против когнитивных искажений являются правовая система и научные методы. Чтобы преодолеть наши эволюционно укоренившиеся убеждения, подобные методы нужны и в геймдизайне.

Когнитивные искажения наполняют нас ложной уверенностью, основанной на убеждениях пещерного человека. Но на самом деле мы знаем гораздо меньше, чем думаем. Вот почему ключ ко всем лучшим стратегиям геймдизайна заключается в том, что от дизайнеров не требуется много — меньше предвидения, меньше общения и меньше мысленного моделирования. Традиционный геймдизайн требует таких сверхчеловеческих подвигов, как составление целого рабочего дизайна на бумаге или руководство десятками подчиненных, и четкого понимания того, что они должны делать. Этого никто не умеет. Если в качестве инструмента нам достался своеобразный человеческий разум, а наша задача такая же сложная, как и создание машин, генерирующих опыт, мы должны работать поэтапно со смирением.

Планирование и итерация

Как отмечает Ханс Гроте, строительный подрядчик, тренер по футболу не скажет кому-то из форвардов, что тот точно забьет гол, если на шестой минуте игры приблизится к воротам противника справа под углом 22 градуса и в 17 метрах от цели забьет мяч под углом подъема 10 градусов и 11 минут... Если тренер хочет определить позиции, с которых должен забивать каждый игрок команды, он должен помнить, что к бутсам может прилипнуть влажная земля. Комок грязи между бутсой и мячом может нарушить угол запланированного удара. Поэтому было бы разумно изучить средний размер комков грязи и частоту их появления, а также места на бутсах, где они прицепляются с наибольшей вероятностью. Но если учесть, что футбольные поля на севере, как правило, песчаные, а на юге — глинистые, нам нужно.... Думаете, никто бы никогда не стал заниматься подобным? Стал бы еще как!

Дитрих Дернер

Перепланировщик

Вот вам история, которая случалась много раз.

У дизайнера есть идея для игры. Он хочет реализовать ее должным образом, поэтому решает не лениться. Он собирается работать самым дисциплинированным, усердным способом, какой только знает, — написать дизайн-документ. В Документе описывается все: механика, задумка, сценарии диалога, оформление, технологии, целевые рынки. Дизайнер переписывает его снова и снова, анализируя каждую часть, переосмысливая, представляя себе, как разворачивается игра.

Проходят месяцы. Наконец, он заканчивает его. Документ состоит из 200 страниц спецификаций механики, примеров прохождения

игры, биографии персонажей и описания интерфейса. Он мог бы сейчас же распечатать его, просто для того, чтобы поднять и почувствовать его вес. Я знаю, как это, потому что именно так и сделал, когда написал свой дизайн-документ для Elemental Conflict.

Затем он приступает к разработке. Он собирает игру как складную картинку, каждая часть которой предназначена для определенного места в соответствии с Документом. Проходят месяцы прогресса, но дизайнер верит в свой Документ. В конце концов, он дает возможность первому игроку поиграть в игру впервые. И именно тогда все расстраивается.

Ничего не работает так, как задумано. Самый сильный враг скатывается в простую, дегенеративную стратегию. Игрок пропускает душещипательную историю, потому что он занят тем, что прыгает на столе. Он не понимает простейшей механики и легко осваивает самую сложную. Он пропускает главный проход и заканчивает тем, что бродит по одной и той же комнате 20 минут. Он ненавидит персонажа-компаньона и использует только три из 10 инструментов.

Но бывает, что происходит и хорошее. Игрок находит новое, более интересное решение головоломки. Он влюбляется во второстепенного персонажа. И теперь, когда игра пошла, разработчик может увидеть сотни простых вариантов дизайна. Если он подправит этого персонажа, появится новая увлекательная стратегия. Если бы он объединил эти части, история, заданная сценарием, была бы понятнее и интереснее. Если бы он убрал стоимость этих ресурсов, скорее всего, темп бы улучшился. Теперь все кажется понятным.

Дизайнер оказывается в безвыходном положении. С одной стороны, у него есть Документ, в который он вложил столько любви и времени. С другой стороны, перед ним стоит суровая реальность — как неожиданные провалы, так и случайные открытия. И они тянут в разные стороны. Здесь нет вариантов. Он должен либо уничтожить свой Документ, либо проигнорировать свои открытия.

Основная ошибка этого дизайнера в том, что он слишком много планировал.

Недостаточное планирование

А вот еще одна история, которая случалась много раз.

Команда начинает игру. Они быстро обсуждают идеи, а затем погружаются в работу. Художники начинают создавать модели персонажей, среды и концептуальные изображения. Программисты начинают собирать искусственный интеллект, алгоритмы мирового поколения и физические движки. Дизайнеры создают уровни, интерфейсы и размахивают руками во время бурных совещаний по вопросам инноваций. Прогресс кажется быстрым.

Но со временем все начинает портиться. Игра тянется со скоростью 10 кадров в секунду, потому что несколько программистов использовали весь бюджет игры. Из-за отсутствия четкого представления о том, что собой представляет игра, трудно найти инвесторов. Оказывается, художник потратил недели, работая над вариациями персонажа, который появляется только один раз.

Дизайн получается несвязным, потому что очень много людей работали по отдельности, сосредоточившись на своих собственных представлениях о том, какой должна быть игра. Одна часть напоминает глубокую историю RPG. Другая похожа на игру в жанре «экшн-шутер». Третья — на игру в жанре «стратегия». Дизайн в конечном счете становится своего рода монстром, напоминающим Франкенштейна, кусочки которого никогда не станут элегантным целым.

С приближением даты релиза нужно выпустить продукт. Игра не работает как единая система. Оставшаяся работа не соответствует возможностям команды. Одна подсистема пропускает огромное количество графики; другую никто вообще не тестировал; третья — ниже требований к качеству. И наконец, бесполезно рекламировать игру, потому что никто не знает, что это будет.

В конце концов, команда усиленно работает в течение шести месяцев, прорабатывая большие куски игры, пытается укрепить ядро того, что у них уже в наличии, и на выходе получает нечто особенное. Игра получает посредственные отзывы, и все задается вопросом, что именно пошло не так.

Основная ошибка этих разработчиков заключается в том, что они недостаточно планировали.

Недостаточное и чрезмерное планирование

Без планирования процесс распадается, поскольку разные части команды и игры работают друг против друга. Это недостаточное планирование. Но если мы составим тщательный, детальный план, он рухнет при контакте с реальностью. Это чрезмерное планирование. Похоже на «Поправку-22»¹. В любом случае, мы расстраиваемся.

К счастью, есть решение. Но прежде, чем мы его рассмотрим, мы должны рассмотреть проблемы с недостаточным планированием и чрезмерным планированием более подробно.

Цена недостатка планирования

Недостаточное планирование создает несколько характерных проблем.

Если мы недостаточно планируем, то почти всегда выполняем работу, которую придется отбросить. Работа оказывается ненужной или устаревшей из-за дальнейшего прогресса. Планирование помогает избежать этой проблемы. С помощью планирования мы можем определить минимальный набор шагов, необходимых для достижения цели. Если мы обнаружим, что часть работы не нужна, мы можем убрать ее из плана еще на этапе планирования. Это более эффективно, чем сначала делать, а затем выбрасывать.

Недостаточное планирование также мешает координации команды. Планирование является необходимой частью координации. Даже команда из двух разработчиков должна обсуждать, что они будут делать в течение следующего часа. А теперь представьте команду из сотни разработчиков — координация становится

¹ В этом романе Дж. Хеллера постоянно возникают конфликты и парадоксы между взаимоисключающими правилами военных уставов и инструкций. — *Примеч. ред.*

огромной проблемой. План, в котором описывается все, что должен делать каждый в течение следующего месяца или года, является одним из способов решения этой проблемы. Такой план можно отправить всем разработчикам проекта, и каждый человек сможет выполнять свою часть.

В случае с недостаточным планированием это невозможно. Если проект недостаточно спланирован, люди работают без четкого представления о том, как их задачи вписываются в целое. Между работой разных людей появляется несовместимость. Некоторые несовместимости носят технический характер, например в модели персонажа, которая не соответствует стандартам, или в подсистеме, которая потребляет слишком много памяти. Другие несовместимости могут быть творческими, например несовместимость в деталях истории, элементах дизайна или художественных стилях. Это отсутствие творческого единства превращает игру в непродуманного монстра Франкенштейна.

Наконец, разработчики — не единственные люди, которым нужно знать, какой будет игра. Недостаточное планирование лишает информации, на которой можно основывать свою работу. Например, чтобы реклама попала на телевидение, ее сначала нужно сделать и определить ее временной интервал, а это занимает месяцы. Поэтому, если мы хотим согласовать выпуск рекламной кампании в декабре, маркетологи должны были начать работу прошлым летом или еще раньше. Точно так же инвесторы хотят знать, во что они вкладывают деньги, и часто будут требовать подробное описание будущего продукта. HR-менеджеры должны знать, кого нужно будет нанимать задолго до того, как появятся открытые вакансии. Каналы сбыта нуждаются в предварительных оценках того, сколько копий игры будет продано, кому и где. Это нужно, чтобы они могли планировать товародвижение дисков. Мир хочет знать, какой будет игра и когда она появится, а недостаточное планирование делает это невозможным.

Цена чрезмерного планирования

Существует распространенное убеждение, что небольшое количество лишнего планирования никому не мешает. Но это

совсем не так. Чрезмерное планирование разрушает проекты разными способами

Чтобы написать план, требуется время. Его нужно придумать, обсудить, записать, отредактировать и отправить. Поскольку планы превращаются в сотни страниц, они могут утяжелить проект. Чрезмерное планирование отвлекает, и вместо реальной работы над проектом вы тратите время на планирование.

Придется сокращать планы, если они неизбежно проваливаются. Чтобы вырезать согласованную идею, потребуются обсуждения, дебаты и политический вес. А для творческого человека психологически тяжело вкладываться в идею, а затем отказываться от нее. Чрезмерное планирование приводит к появлению многих вещей, которые позже придется убрать из проекта, а это означает, что вы снова и снова будете нести дополнительные затраты.

Но это не самая большая цена чрезмерного планирования. Реальная цена заключается в том, что такое планирование создает ложное чувство уверенности относительно будущего. Планы на бумаге часто рассматриваются как гарантия того, что проект будет реализован. Но это не так — планы переполнены предположениями. Позже, когда будет очевидно, что эти предположения не оправданны, зависящая от них работа просто остановится.

Например, представьте, что согласно первоначальному дизайн-документу персонаж игрока может прыгнуть на 3 метра вверх. Согласно этому плану, дизайнер уровня строит уровень, ограниченный стенами высотой 3,3 метра. Если план правильный, все должно сработать, поскольку персонаж игрока не может преодолеть стену высотой 3,3 метра при том, что он может прыгнуть вверх только на 3 метра. Но затем дизайнер решает, что игра станет намного лучше, если персонаж сможет прыгнуть на 4,5 метра вместо трех. А вот теперь проблема. Либо нужно подгонять уровень для обработки 4,5-метровых прыжков, либо прыжок должен соответствовать запланированному изначально. Один из вариантов заставляет отказаться от хорошей работы. Другой — ухудшает игру.

И далеко не всегда все так просто, как в этом примере.

Настоящий геймдизайн — это паутины зависимостей; изменения в одном месте почти всегда подразумевают много изменений в другом месте. Обычное изменение высоты прыжка может повлиять на границы уровней, движение врага (чтобы он мог поймать прыгающего игрока), головоломки прыжка, аудиовизуальные эффекты прыжка и многое другое. И каждое из этих изменений может подразумевать дальнейшие изменения — изменение движения противника может включать изменения его графики и анимации. Изменение головоломки прыжка может означать изменение сюжета уровня, если эта головоломка связана с сюжетом. Последствия неудачного плана отражаются на дизайне, графике, коде, механике и задумке.

Геймдизайн выделяется среди современных творческих поисков тем, что в каждой системе заложена неопределенность.

Как сказал Сорен Джонсон, ведущий дизайнер Civilization IV: «Быть геймдизайнером — значит быть неправым». Дизайнер может только догадываться, как будет работать система или уровень, но он никогда не сможет сказать это наверняка. Обычно когда игра готова, игровая система работает совсем не так, как предполагалось. Вот почему отличные игры склонны значительно меняться в процессе разработки.

Например, Halo — это один из самых популярных шутеров от первого лица. Но изначально это был совсем не шутер и совсем не от первого лица. Это была стратегическая игра с нисходящим методом проектирования. Вместо того чтобы стрелять в голову космического десантника, игрок смотрел на поле боя сверху и использовал указательный интерфейс, чтобы управлять войсками. Но в процессе разработки дизайнеры обнаружили, что чем ближе они показывали происходящее, тем лучше становилась игра. Они все больше и больше приближали камеру, пока в конечном итоге главный герой не стал смотреть на происходящее собственными глазами. Этот странный путь развития не был ошибкой — он был необходим для того, чтобы игра стала успешной. Halo была известна своими инновациями в крупномасштабных сражениях, включающих нескольких персонажей, гонками на выживание и открытым окружением. Все это изначально было стратегической игрой. Никто не мог запланировать полученный результат, и никто этого и не делал.

BioShock — это исследование подводного города, построенного в стиле ар-деко. Город под названием Восторг был попыткой создания утопии, основанной на принципах философии объективизма Айн Рэнд. Персонаж игрока прибывает в город в 1960 году, утопия потерпела крах, и Восторг погрузился в гражданскую войну. Игра прославилась этим богатым и уникальным нарративом о мире. Но с самого начала события игры BioShock происходили не под водой и не имели ничего общего с Айн Рэнд. Это была научно-фантастическая игра на космическом корабле. Позже они переместились в заброшенный нацистский бункер, кишаший мутантами. Прошло всего несколько лет, и игра превратилась в подводный город в стиле ар-деко и обрела свою тему объективистский утопии. Дизайнеры не планировали этот мир на бумаге; они разработали его за годы работы над самой игрой.

The Sims начиналась как архитектурная игра. Изначально Уилл Райт не планировал помещать в дом семью. Игра была о строительстве и не более. Игрок экспериментировал с различными формами дома, цветами и обстановкой в абсолютно стерильной среде. Только добавив простого персонажа в пространство, Райт обнаружил, насколько сильно это понравилось игрокам. Райт следовал возможностям, которые он видел, и игра все больше и больше сосредоточивалась на людях, и так до тех пор, пока они не стали ее центром. Он не планировал такой результат; он обнаружил его в процессе создания игры.

Меняется весь дизайн, как это случилось с Halo, BioShock и The Sims. Но даже самый маленький кусочек игры может преподнести сюрпризы. Например, когда я работал над уровнями головоломки в загружаемом контенте для BioShock, в моем уровне была комната с рядом ракетных пусковых установок вдоль стены. Я хотел, чтобы игроки о них узнали, но чтобы при этом их не убило. По этой причине я использовал старый трюк сообщения игроку об опасности: когда он входил в комнату, появлялся враг и бежал на игрока только для того, чтобы ракетные пусковые установки его обстреляли. Отлично сработало. Враг кричал и взрывался. Проблема казалась решенной. Потом я смотрел, как кто-то другой проходит этот уровень. Он вошел в комнату. Враг закричал и побежал на него; так как это был уровень головоломки, у игрока

не было оружия. Поэтому он повернулся и выбежал из комнаты, чтобы убежать от врага. Мне пришлось создать вокруг игрока непробиваемое стекло, чтобы он чувствовал себя в безопасности, не отступал и наблюдал за происходящим.

Геймдизайн всегда изменчив. У каждого опытного дизайнера множество историй о том, как игровые системы работают и не работают там, где этого не ждешь. Невозможно узнать, как будет работать дизайн и будет ли он работать вообще, прочитав дизайн-документ на бумаге. Именно этот разрыв между ожиданиями и реальностью приводит к перегрузкам сотрудников, а также нарушению сроков и бюджета. Когда вы предполагаете, что планы надежны как скала, хотя на самом деле они очень неопределенные, вы будете перепланировать, что не приведет ни к чему хорошему.

ИТЕРАЦИЯ —

это практика составления краткосрочных планов, их реализации, тестирования и повторения.

Итерация

Мы не можем вообще не планировать, но мы также не можем планировать каждую деталь до конца проекта. Нам нужно нечто среднее. Нам нужна итерация.

Традиционный творческий подход является линейным. Планируем, затем компилируем, затем тестируем, чтобы проверить качество и готовность продукта.



Итерация работает иначе. Действия выполняются не поочередно, а циклично.



Это означает, что нам не нужно прогнозировать события наперед. Нам нужно планировать только до конца текущего цикла. Каждый раз, когда мы тестируем игру, мы сверяем наши предположения с реальностью. Эта проверка дает надежную информацию, на основании которой мы планируем следующий цикл.

Этот цикл может повторяться несколько раз или тысячи раз в зависимости от проекта. Иногда разработчики планируют количество циклов, которое им нужно, перед выпуском. В других случаях они просто повторяют цикл, пока игра не достигнет необходимого уровня качества или пока не закончится бюджет.

Итерация осуществляется не только для всей игры. Мы можем применять ее для уровня, инструмента или интерфейса. В больших командах должно быть много разных циклов итерации, работающих одновременно.

Пример итерации

Поскольку каждая задача проектирования особенная, любой процесс итерации нужно адаптировать к конкретной задаче. Вот пример простого процесса итерации, который я использовал для разработки боевых сценариев в шутерах от первого лица. Этот процесс не подходит для других задач или разработчиков — это всего лишь один из возможных примеров.

Я начинаю черновую работу и максимально быстро набрасываю базовый бой. Добавляю элементы, не задумываясь и не анализируя их. Я мог бы подумать о том, куда я иду, но мне это не нужно. Моя единственная цель — запустить бой как можно скорее.

Через час я это делаю. Бой, как всегда, получился ужасным. Он выглядит так, как будто его разработал незаинтересованный дизайнер-новичок. Серые блоки укрытий хаотично разбросаны, геометрия мира представляет собой горстку плохо масштабированных кубиков, а враги появляются в виде гигантских глыб. И поскольку я обычно забываю дать игроку оружие, он всегда проигрывает. Но несмотря на свое низкое качество, эта первая

версия выполняет свое предназначение. Она замыкает цикл итерации.

Бой больше не прокручивается в воображении. Он вполне реален. Когда человек играет в настоящую игру, держа руки на джойстике, слыша тикающий секундомер, у него возникают мыслительные процессы, которые невозможно воспроизвести любым другим способом. Эта первая версия игры и не должна быть похожа на конечный продукт. Ее единственная цель — стать стартовой платформой к чему-то менее ужасному.

И это получается. Пока я играю, мне в голову приходят новые идеи, и они более конкретны, чем все, что я мог придумать до этого. Они мне нравятся, и мне не нужно ждать, анализировать или что-то записывать. Мое вдохновение не улетучивается. Протестировав игру один раз, я возвращаюсь в редактор, удаляю куски, которые не работали, разбрасываю по местности места для укрытия, дорабатываю оружие и врагов. Возможно, на этот раз я даже не забуду дать игроку оружие.

Я делаю несколько таких циклов, меняю уровень и проверяю его снова и снова. Поскольку итерация проходит быстро, я не трачу время на анализ. Я просто добавляю что-то в уровень и тестирую несколько минут. И так как вся игра еще остается в виде серых блоков, работа остается сырой. Это хорошо, так как я вношу достаточно существенные концептуальные изменения вроде замены башни на мост или меняю тип основных врагов. О деталях пока не думаю.

За несколько часов я повторил несколько циклов итерации и, возможно, несколько раз поменял общую концепцию. Могу начать с врагов в башне (на самом деле это просто высокий блок со снайперами наверху), но понимаю, что это не сработало. Могу попробовать начать с моста (длинный, широкий блок через длинную дыру в полу). Возможно, я попробую делать минные поля, снайперов, траншеи, артиллерию и любые другие приемы, которые смог придумать. Даже самые черновые версии любого из предложений можно сделать за считанные минуты.

Попробовав от трех до восьми разных концепций, я выбираю одну, рабочую. И здесь все начинает меняться. Цикл удлиняется,

я вношу меньше изменений. Я не тестирую игру каждый час, а делаю это раз в два или три часа. Я не разрываю и не заменяю целые здания, а делаю позиционирование на стенах и колоннах. Всегда вношу изменения, обусловленные реальными проблемами, которые я заметил при тестировании. Каждый тест указывает мне на новые очевидные изменения, которые необходимы.

С этого момента я подключаю к работе дизайнера уровней. Скорее всего, он еще не работает непосредственно над пространством — для этого еще слишком рано, но он может проконсультировать по поводу его художественной реализации. Если моя общая концепция бессмысленна с художественной точки зрения, возможно, мне стоит начать все сначала. Чаще всего мы обсуждаем способы корректировки игрового пространства, чтобы сделать его более пригодным для графики. Например, сам уровень остается серым, но мы можем решить придать башне или мосту какую-нибудь необычную форму, которая отражает стиль, тему, историю мира и его настроение. Дизайнер уровней может что-то смоделировать или сделать тестовый уровень, чтобы исследовать художественные идеи для игрового пространства.

Итерационные циклы продолжаютсЯ. Бой становится сбалансированным. Иногда пространство меняется в соответствии с художественными или нарративными проблемами, но большинство изменений по-прежнему обусловлены проблемами баланса, темпа, ясности и глубины.

Однако, в конце концов, я захожу в тупик. Настает момент, когда, тестируя свою собственную работу, вы больше ничего не узнаете. К этому моменту у меня уже есть готовый, рабочий бой, который меня устраивает, но игра создается не для меня. Бой должен работать для всех своих игроков. И единственный способ понять, насколько хорошо он работает, когда в него играют реальные игроки, — это понаблюдать за их игрой.

Итак, я приглашаю других тестировщиков вместо себя. В идеале, приглашаю реальных игроков из потенциальной целевой аудитории для этой игры. Но даже если это невозможно, существуют и другие альтернативы. Я обычно «использую» коллег. Приглашаю программистов, тестировщиков, художников и звукорежиссеров,

которые не видели бой, сажаю их за свой компьютер и смотрю, как они играют. Я ничего им не говорю, стою подальше от них, чтобы они меня не видели, и жду момента, когда что-то пойдет не так.

И он всегда наступает. Некоторые игроки прекращают бой, изобретая стратегии, о которых я никогда не думал. Они отказываются продвигаться вперед и стреляют по каждому врагу с расстояния. Или атакуют вражеские батальоны без единого выстрела. Другие впадают в фрустрацию, потому что они не знают бой так, как его знаю я, или не замечают ключевой элемент. Они не замечают дыру в полу, проваливаются в нее и умирают. Их может застрелить парень, которого я послал им в помощь. Они наступают на мигающую мину, которую я считал очевидной. Перефразируя название шоу Билла Косби, тестировщики делают самые невероятные вещи.

Проведя один плейтест, я пишу список задач, которые нужно решить. Некоторые из них простые (лучше осветить врага, чтобы люди могли его видеть). Другие более сложные (реструктурировать маршрут слева, чтобы его могли использовать как игроки, так и враги). Я приступаю к работе. Спустя полдня изменения внесены, и я готов к следующему плейтесту. Я нахожу кого-то, кто еще не проходил этот бой, и смотрю, как он играет.

Цикл повторяется еще 10–20 раз. К концу цикла проходит две или три недели. Бой идет в хорошем темпе, хорошо сбалансирован и подходит игрокам разных уровней мастерства и с разными игровыми привычками. Мне не нужно догадываться, как будут развиваться события, когда игра попадет к реальным игрокам, потому что я уже и так это знаю благодаря тестировщикам. Но это все еще не похоже на готовую игру — повсюду вы видите плоские серые формы. Самое время художникам вступить в бой.

Дизайнеры уровня делают первые шаги в игровом пространстве, заменяя серые фигуры настоящими художественными образами. Мы снова тестируем. Даже если механика боя не меняется, художественные изменения влияют на то, как игроки его воспринимают, поэтому мы должны видеть, как они влияют на плейтест. Если мы видим проблемы, то обсуждаем их, чтобы найти решение.

Иногда мне приходится изменять детали сценария, удаляя или добавляя героев или инструменты. В других случаях художнику, возможно, придется добавить свет или что-то упростить, чтобы уменьшить шум. Итерационный цикл длится уже несколько дней, так как создание графики — это медленный процесс.

Если нам повезет, графика не вызывает серьезных проблем. Поскольку я провел тщательный плейтест с серыми кубами, базовый уровень должен работать так же, как и раньше. Итак, еще через несколько циклов уровень начинает работать как на уровне механики, так и графики.

Теперь мы еще больше увеличили цикл, включив в него другие дисциплины разработки. Текстовые окна заменяются на реальные диалоги. Звукорежиссеры передают атмосферу и звуки сцен. Мы ищем способы выразить нарратив о мире через игровое пространство. Писатели переделывают диалоги. Наконец, тестировщики справляются на отлично, мы исправляем технические неисправности и игра поставляется на рынок.

Это один из способов разработки шутера. Другие итерационные циклы могут значительно отличаться в зависимости от проекта и преследуемых целей. Этот конкретный процесс был основан на механике, поэтому начал его дизайнер боев, работающий над балансом и темпом. Для другой игры может потребоваться хороший нарратив, где сначала происходит итерация сюжета, а затем механики. Кроме того, существуют совершенно разные аспекты дизайна: дизайн персонажей, дизайн интерфейса и дизайн систем требуют разных методов. Некоторые будут представлять короткие циклы, сделанные одним человеком. У других будут длинные циклы продолжительностью в несколько недель с участием 10 разных специалистов. Некоторые разработчики тестируют игру в одиночку, другие тихонько наблюдают со стороны, кто-то использует метрики автоматизированного тестирования или отправляется в специализированные лаборатории.

Но независимо от того, какой цикл вы будете использовать, в основе итерации лежат те же базовые принципы. Она меняет глубокое планирование на проверку в реальных условиях. Она помогает сначала протестировать крупную текстуру, прежде

чем переходить к оттачиванию деталей. И она требует, чтобы дизайнеры не слишком инвестировали в планы на будущее, а вместо этого постоянно адаптировались к непредсказуемым результатам тестирования.

Горизонт планирования

Сколько должен длиться итерационный цикл? Нужно ли тестировать игру каждый день? Каждую неделю? Каждый месяц?

Если наш цикл слишком длинный, мы занимаемся избыточным планированием. В конечном итоге все закончится тем, что разработчики будут думать о проблемах, которых не существует, или не заметят проблем, скрытых за их предположениями. Если цикл слишком короткий, мы планируем недостаточно. Мы тратим время на ненужную работу и не можем заставить группу разработчиков работать в команде. Мы должны найти баланс между ними, выбрав правильный *горизонт планирования*.

ГОРИЗОНТ ПЛАНИРОВАНИЯ — это будущее время, на которое планируется работа.

Долгосрочный горизонт планирования — это планирование работы на месяц и ее последующее выполнение, прежде чем перейти к следующему тестированию. Краткосрочный горизонт планирования — просто закидать игру разными формами и ежеминутно тестировать, чтобы понять, как все работает.

Выбор горизонта планирования в основном зависит от степени точности ваших планов. Если вероятность того, что все пойдет по плану, достаточно высока, горизонт планирования должен быть долгосрочным. Именно так архитекторы проектируют здания вплоть до болтов и гаек — они абсолютно уверены в конфигурации. Если планы склонны меняться, ваш горизонт планирования должен быть краткосрочным. Это похоже на футбольный матч, когда все меняется в зависимости обстоятельств, которые невозможно спрогнозировать. Любой процесс разработки игры находится в некоторой точке между этими двумя крайностями.

Давайте рассмотрим некоторые более конкретные обстоятельства, которые влияют на горизонт планирования.

Чем игра менее оригинальна, тем глубже мы можем планировать. The Sims полностью изменились во время разработки, а The Sims 2 — нет, потому что ядро дизайна уже было хорошо разработано в первой игре. Точно так же разработчик шутера от первого лица может заимствовать из других игр всю уже имеющуюся информацию по этому жанру, чтобы понять, как будет работать его собственная игра.

Крайней формой является создание клона или портирование существующей игры. Поскольку весь проект уже создан и протестирован на реальных игроках, можно даже заранее спланировать каждую деталь, подобно архитектору, который готовит копии проектной документации.

Вот почему создание сиквела так сильно отличается от создания оригинала. Некоторые игровые франшизы делают пять сиквелов или больше, незначительно изменяя базовую механику. Это обеспечивает плавность процессов разработки, поскольку дизайн пятого сиквела может зависеть от огромного объема информации, полученной в предыдущих играх.

Планировать исходные игры гораздо сложнее, потому что у дизайнера нет основы из хорошо проверенных дизайнов. Исходная игра, состоящая из оригинальной механики, управляемой с помощью оригинального интерфейса в оригинальном мире, представляет собой гигантскую сеть взаимосвязанных неопределенностей. В такой ситуации правильный горизонт планирования может составлять день или меньше. Любой план, составленный на неделю вперед, может разрушиться в результате внезапных сюрпризов в виде работающего или неработающего геймплея.

В начале проекта мы стоим на зыбучем песке предположений. К концу мы беспокоимся о мельчайших деталях структуры. На начальном этапе горизонт планирования проекта может составлять менее суток, так как небольшая группа разработчиков пробует дикие идеи. Последние несколько месяцев могут быть распланированы заранее в развернутой таблице с указанием каждого графическо-

Шаблонные, производные игры могут иметь относительно долгосрочный горизонт планирования, потому что они зависят от имеющихся данных.

Исходные игры позволяют только краткосрочное планирование, так как они зависят от элементов, которые еще не были использованы.

Соответствующий горизонт планирования склонен увеличиваться в течение срока проекта.

го объекта и каждой задачи программирования, которые необходимо закрыть, прежде чем игра выйдет на рынок.

При низкой стоимости тестирования необходим краткосрочный горизонт планирования.

На начальном этапе проектирования боя я мог очень быстро скомпилировать и протестировать идею боя. Зачем тратить час на анализ идеи, если я могу создать и протестировать ее за 15 минут и получить гораздо больше информации? Оно того не стоит. Поэтому я не думаю слишком много, а просто делаю игру.

В этом преимущество хороших инструментов. Они не только делают игру быстрее. Дело в том, что они смещают проблему выбора между планированием и компиляцией и позволяют использовать более экспериментальный подход к разработке, уменьшая цену ошибки. Хорошие инструменты позволяют рисковать. Так они позволяют вам обратить внимание на дизайн, который вы могли пропустить, если работа шла слишком медленно и вам приходилось планировать и все делать правильно с первого раза.

Итерация — это то, что известно как алгоритм поиска с восхождением к вершине. Представьте каждую возможную игру в виде точек на ландшафте. Точки на более высоком уровне — те игры, которые считаются лучше. Итерация делает игру похожей на слепого альпиниста, который начинает карабкаться по любому склону, где оказывается. Он делает короткие шаги, тестирует их, чтобы понять, стали ли они лучше, и продолжает дальше, если ожидания подтвердились. Со временем игра становится все лучше и лучше.

Планируйте более
глубоко, если вы
ставите своей целью
новаторство.

Проблема поиска с восхождением к вершине заключается в том, что альпинист слеп, он не может точно сказать, поднимается ли он на гору или холм. Если мы пойдем по склону невысокого холма, мы доберемся до его вершины, но не будем знать о горе неподалеку. Мы хотим взойти на эту гору, но если мы можем делать только небольшие шаги, у нас нет возможности добраться туда с вершины этого холма. Итерация оптимизирует дизайн, но не меняет его полностью.

Чтобы прыгнуть выше, нужно оторваться от земли. Это означает внесение больших изменений в дизайн без тестирования. Это ри-

скованно — вы не узнаете, где приземлитесь, пока не доберетесь туда, но это единственный способ открыть радикально новые идеи и избежать стандартизированного дизайна. Составление детального плана позволяет вам увидеть горы на расстоянии, однако вы можете подойти к ним и обнаружить, что это всего лишь подножие горы. В этом заключается риск глубокого планирования.

Избыточное планирование: причины

Как недостаточное, так и чрезмерное планирование одинаково опасно. Но в геймдизайне избыточное планирование более разрушительно. В основном проблема разработчиков заключается в избыточном, а не в недостаточном планировании, а это наносит больше ущерба.

Почему в игровом дизайне возникает избыточное планирование? Существует ряд устойчивых стереотипов, которые заставляют нас планировать в избыточном количестве снова и снова. Давайте разберемся в них подробнее.

Привычка, заложенная культурой

С юных лет мы знакомимся с привычкой планирования. Учителя и родители учат нас планировать заранее и думать о будущем.

И обычно это хорошая идея. Современный мир появился благодаря тщательному планированию. Когда инженеры и рабочие строили плотину Гувера, они заранее решили, что будут делать, прежде чем приступить к работе. Они точно знали, сколько бетона нужно и где именно он будет залит. Они могли точно распланировать работу и поставки материалов для максимальной эффективности. И готовая плотина выглядела практически точно так же, как и было решено на этапе проектирования.

Но геймдизайн отличается, потому что в нем больше неопределенности. Архитекторы плотины Гувера не могли на полпути вдруг решить, что плотина должна была стать небоскребом. Но разработчики Halo обнаружили, что их стратегия, спроектированная по принципу «сверху вниз», должна стать шутером от

первого лица. И, как мы уже убедились, такого рода радикальная трансформация не является необычной.

Врожденная самоуверенность

Давайте поиграем в игру об уверенности. Я задам вам десять вопросов, ответы на которые — цифры. Ваша задача — дать завышенные и заниженные оценки так, чтобы по каждому вопросу вы были на 90% уверены, что ответ попадет в пределы диапазона, который вы указали.

Имейте в виду, что диапазоны могут быть любой величины. Для этого не нужно знать правильные ответы. Задайте достаточно широкий диапазон, чтобы вы на 90 % были уверены, что правильный ответ находится между верхней и нижней границами. Если вы сомневаетесь, ваш диапазон будет большим. Если нет, он будет маленьким.

Я настоятельно рекомендую вам взять карандаш и записать свои ответы. Это упражнение работает не так хорошо, если его только прочитать.

Вопрос	Заниженная оценка	Завышенная оценка
В каком году родился Архимед?		
Сколько видов муравьев существует?		
Какова была численность населения в 1900 году?		
Каков диаметр Солнца?		
Сколько лун у Сатурна?		
При какой температуре плавится железо?		
Каково количество погибших в Первой мировой войне?		
Какова площадь Антарктиды?		
На какой широте находится город Сантьяго, Чили?		
Каков общий вес золота, добытого за всю историю человечества?		

Теперь сверьте свои ответы с ответами в конце книги. Что у вас получилось?

Обратите внимание, что ваш результат теста никак не связан с вашими знаниями географии или истории. Чтобы ваш уровень

достоверности составил 90%, вы можете задать любой диапазон, какой только захотите. И если вы это сделали, то почти наверняка получили 8, 9 или 10 ответов, которые попали в предел вашего диапазона.

Но если вы относитесь к большинству, то, скорее всего, у вас получилось от двух до четырех правильных ответов. Небольшое количество людей получают в своем доверительном интервале пять или шесть правильных ответов. Очень немногим удается пройти тест, даже если они понимают, как он устроен, и уже решали его ранее.

Когда я решал аналогичный тест в книге Стива Макконнелла «Software Estimation: Demystifying the Black Art» (Microsoft Press), я ответил правильно на четыре вопроса. Макконнелл давал подобные тесты сотням профессиональных оценщиков. Это были люди с многолетним опытом оценки сроков реализации проекта и затрат на программные проекты. Даже среди этой элитной группы Макконнелл обнаружил, что фактически менее 1% тестируемых отвечают правильно на девять вопросов, которые мы должны ожидать от объективной оценки. Более 90% из них получили пять или менее правильных ответов. Почему так произошло?

Психологи называют это оптимистическим уклоном. Что-то в психологии человека приближает доверительную оценку в 90% к доверительной оценке в 30%.

Людям присуща
самоуверенность.

Эта чрезмерная уверенность не ограничивается оценкой чисел в вопросах. Исследования показали, что люди постоянно чрезмерно уверены относительно бюджета на разработку программного обеспечения, экономических прогнозов, бизнес-планов и военных стратегий.

Подобная необъективность имеет огромные последствия в планировании в геймдизайне. Иными словами, без поправки дизайнер будет уверен в своем дизайне на 90%, хотя на самом деле шанс, что дизайн будет рабочим, составляет всего 30%. Так складывается огромный разрыв между ожиданиями и реальностью. Такая самоуверенность заставляет нас думать, что мы можем планировать то, что на самом деле не сможем реализовать. Мы

читаем дизайн-документ и думаем, что проект будет работать, хотя вероятность того, что он будет работать именно так, как мы ожидаем, является достаточно низкой. Это толкает нас в сторону избыточного планирования.

Терапевтическое планирование

Вспомните выражение «чувствовать неуверенность». Технически быть неуверенным — значит не иметь определенной информации. Но фраза «чувство неуверенности» перегружена негативными эмоциональными оттенками. Мы рассматриваем неуверенного человека как неспособного и неэффективного. Если мы неуверены, то представляем себя нервными и подавленными. Неуверенность эмоционально неприятна. Ответная реакция на неопределенность часто заключается в желании скрыть неопределенность за *терапевтическим планированием*.

ТЕРАПЕВТИЧЕСКОЕ
ПЛАНИРОВАНИЕ —
это планирование
не для координации
работы, а для того,
чтобы мы не беспоко-
ились о неизбежно
неопределенном
будущем.

Наличие плана может убрать тревогу неуверенности, создав ложное чувство уверенности в будущем. Но, как говорит философ Нассим Талеб, если вы хотите расслабиться, выпейте и не делайте прогнозов. Опрометчивое прогнозирование намного опаснее.

Отсутствие избыточного планирования означает принятие когнитивного стресса из-за неопределенности. Это означает постоянную переоценку ситуации, а не откладывание решений, поскольку их можно легко забыть. Желание избежать этого умственного усилия часто приводит к терапевтическому перепланированию.

Групповая ошибка планирования

Представьте двух человек, Уверенного Боба и Рациональную Алису, в группе, которая пытается угадать погоду. Рациональная Алиса смотрит на небо и точно помнит, что во всех случаях, когда она видела аналогичные сочетания погодных условий, дождь шел почти половину всего времени.

«Я понятия не имею, пойдет дождь или нет, — говорит она. — Как бы то ни было, мы не можем знать наверняка».

Теперь выходит Уверенный Боб. Он недолго смотрит вверх, улыбается, словно наслаждаясь шуткой, понятной только ограниченному кругу людей. Он поворачивается к группе и уверенно объявляет: «Дождя не будет. Не беспокойтесь».

Группа естественным образом выбирает Боба. Боб получает последователей, одобрение и социальный статус. Алису называют слабой, глупой, нерешительной или ленивой, хотя ее ответ и был более точным.

Люди поощряют чрезмерную уверенность и ценят ее выше, чем рациональные сомнения.

Это групповая ошибка планирования. Люди естественным образом тянутся к лидерам, которые, как им кажется, уверенно смотрят в будущее, даже если это видение будущего не соответствует реальности.

Чтобы не попадаться на эту удочку, достаточно поймать Боба на том, что он оказался неправ. Как только это случится несколько раз, люди перестанут его слушать. Но такие выводы в геймдизайне не так очевидны, как в прогнозе погоды. В геймдизайне трудно сразу связать причину и следствие, результаты могут проявляться годами, и за это время происходит так много, что мы забываем о своих прогнозах. В обычной жизни, когда мы можем оценить результат такого «предсказателя» практически сразу, наши инстинкты в конечном итоге заставляют нас не доверять Уверенному Бобу. Но в современных задачах дизайна происходит иначе. У нас нет такой подушки безопасности. Таким образом, стереотипы относительно Уверенных Бобов остаются, а подушка безопасности (проверка результатов) — нет. Стереотипы не сбалансированы.

Если мы не будем бороться с этим эффектом, люди пойдут за более уверенным лидером, а не за тем, кто прав. Неопределенность скрывается за бравадой, и начинается избыточное планирование.

Эффект хиндсайта

Несмотря на все описанные выше стереотипы, можно подумать, что мы могли бы учиться на ошибках. Существуют разработки, которые прошли через десять чрезмерно спланированных проектов подряд, каждый раз испытывая одну и ту же боль, вы-

резая лишние функции, работая в цейтноте и хаосе. Почему мы не учимся на этом опыте? Виной тому эффект хиндсайта.

ЭФФЕКТ ХИНДСАЙТА
(склонность к запоздалым суждениям) — это когнитивные искажения, которые незаметно перестраивают воспоминания, чтобы прошлые события выглядели так, как будто они были более предсказуемыми, чем на самом деле.

В 1972 году исследователь Барух Фишофф спросил людей, что может произойти во время предстоящей дипломатической поездки президента Никсона в Китай. Будет ли Никсон встречаться с Мао Цзэдуном? Произойдет ли значительный дипломатический прогресс? Он спросил о вероятности этих событий, а также о 13 других.

Когда Никсон вернулся, Фишофф снова попросил тех же людей сказать, с какой вероятностью событие будет иметь определенный исход. Эффект хиндсайта был очевиден. Если чей-то прогноз оказывался верным, человек говорил, что был увереннее в своем ответе, чем это было на самом деле. Если же прогноз оказывался неверным, то он преуменьшал степень своей уверенности. Они скорректировали свои воспоминания, преувеличивая свою способность предвидеть, как все произойдет. Постфактум процесс разработки игр всегда выглядит более логичным и контролируемым, чем это было на самом деле. Наш мозг автоматически редактирует хаос процесса разработки, превращая его в нашей памяти в чистую историю. Когда мы рассказываем о нем другим, мы еще сильнее осуществляем упрощение. Бесплодная трата времени время на тангенсы, бездумные ошибки, ужасные недопонимания и монотонные дни оттачивания работы — все это отпадает, а история становится детской сказкой. На самом деле я описал подобные истории в этой книге.

Проблема в том, что уроки, которые мы должны извлечь, заключаются не в чистой отредактированной истории, которую мы рассказываем позже. Они в запутанных уловках и ложных предсказаниях, которые мы вычеркиваем из истории. Эффект хиндсайта мешает нам учиться на своих ошибках, заставляя нас думать, что события были более предсказуемыми, чем они были на самом деле. Эффект хиндсайта всегда заставляет чувствовать, что глубокое планирование было бы возможным. Поэтому мы думаем, что оно будет возможно в будущем, и мы снова и снова занимаемся избыточным планированием и не учимся на своих ошибках.

Когда вы знаете, что искать, вы начинаете видеть эти ошибки избыточного планирования. И вы сможете их исправить.

Протокол тестирования

Итерационный процесс представляет собой цикл между планированием, компиляцией и тестированием. В основном все сосредоточены на планировании и компиляции, а тестированием часто пренебрегают. Но этап тестирования имеет решающее значение, потому что это механизм, с помощью которого мы извлекаем уроки из реального мира и получаем основное преимущество итерации.

Цель плейтестинга состоит не в том, чтобы выявить технические проблемы или собрать данные о маркетинге, а в том, чтобы понять, как работает геймдизайн в действии. Это значит, что нужно дать поиграть в игру обычным людям и посмотреть, где дизайн работает, а где — нет. Где игроки в замешательстве? Где слишком легко или слишком сложно? Достаточно ли сбалансирована игра? Есть ли вырожденные стратегии? Понимают ли игроки нарратив?

Проведение плейтеста — это навык. И это не менее сложно, чем планирование или компиляция. Качественный плейтест дает дизайнерам необходимую информацию без особых затрат и усилий. Плохой плейтест — и дизайнер пропускает критические недостатки дизайна, тратит время напрасно и даже может активно вводить в заблуждение других дизайнеров.

Ключом к получению достоверных данных является использование правильного протокола тестирования.

Создать хороший протокол тестирования сложно, потому что если мы делаем это неправильно, то не получаем никакой обратной связи. Ошибочные или вводящие в заблуждение результаты тестирования часто выглядят очень убедительно. Хуже того, при плохих протоколах тестирования сам тест обычно проходит более гладко. Плохо выполненное тестирование хуже, чем просто бесполезное тестирование. Перед плохим тестированием дизайнер не знал, работает ли игра. После плохого тестирования он думает,

ПРОТОКОЛ ТЕСТИРОВАНИЯ — это набор правил и процедур для проведения плейтеста.

что игра работает, хотя на самом деле это не так. Дело не в том, что он не смог получить информацию, а в том, что та информация, которую он получил, не соответствует действительности.

Однажды я расспрашивал ведущего дизайнера по поводу провалившегося многопользовательского шутера. Вот так выглядел его протокол тестирования: группа игроков сидела в комнате с запахами еды и в течение продолжительного времени играла в игру. В этой среде игра, казалось, работала хорошо. Они выполняли циклы итерации, находили проблемы, тестировали и шлифовали игру до тех пор, пока она не стала такой же глубокой и сбалансированной, как философ, идущий по канату. Но этот успех был обманчивым, потому что их протокол тестирования не выявил каких-либо ошибок в дизайне, которые обнаруживаются в том случае, когда в игру играют незнакомые друг с другом люди через интернет, а не друзья в одной комнате. Пока играли хорошо скоординированные, очень общительные команды, игра проявляла себя блестяще. В интернете ее не ждал успех. Она настолько зависела от сложной командной тактики, что не работала совсем, если играли ленивые, некомпетентные, случайные игроки. Дизайнеры проводили плейтесты, но их ошибочный протокол тестирования не выявил критические недостатки в дизайне, и поэтому игра провалилась на рынке и среди большинства игроков.

Протоколы тестирования могут не оправдать надежд множеством способов. Открытое тестирование приводит к появлению у тестируемых желания подтвердить свою точку зрения. Групповое тестирование создает социальную конкуренцию, и игроки копируют мнения друг друга. Если попросить игроков выражать свои мысли вслух, дизайнеры смогут интерпретировать действия игроков, но в этом случае действия могут изменяться. Выбор тестируемых приводит к необъективности, которая скрывает проблемы, появляющиеся, только если в игру играют люди определенного возраста, пола, культуры или уровня мастерства. Небольшое количество тестируемых означает, что наши данные искажены поразительно большими случайными статистическими отклонениями.

В конце концов, мы никогда не сможем полностью избежать этих отклонений. Протокол тестирования — это не вопрос пра-

вильного и неправильного. Это ремесло, в котором дизайнер пытается получить максимально полезную информацию из заданного набора ресурсов.

Давайте рассмотрим основные протоколы тестирования.

Самостоятельное тестирование

Самый дешевый протокол тестирования — игра в одиночку. Несмотря на то что оценка дизайнера предвзята, так как он знает игру, обычное наблюдение за игровыми системами в их непосредственном движении дает огромное понимание. Так можно выявить множество проблем в потоке, ритме и балансе. И конечно же технические неисправности лучше всего обнаруживать при самостоятельном тестировании. Самые ранние циклы итерационного процесса должны завершаться самостоятельным тестированием.

Плейтестинг «через плечо»

В этом случае дизайнер наблюдает за тем, как играют другие игроки. В ином случае он может даже просто схватить своего коллегу и усадить его за свой рабочий стол. Или же может пригласить случайных людей в ненастоящую гостиную с напитками, игровой системой и скрытыми камерами.

Наблюдение за плейтестингом «через плечо» лучше, чем самостоятельное тестирование, потому что можно привлечь разных игроков и они не будут знать игру полностью. Вы можете пригласить поучаствовать старых, молодых, мужчин, женщин, агрессивных, пассивных и еще кого угодно. И никто из них не будет знать об игре так, как вы, поэтому все они будут воспринимать ее практически как настоящие игроки.

Самая большая опасность в наблюдении за плейтестом «через плечо» — это испортить тест, сказав игрокам то, чего они не должны знать. Именно поэтому почти во всех случаях дизайнер должен молчать во время теста. Не говорите. Не смейтесь. Не вздыхайте тяжело. Не сигнализируйте и не выдавайте свои мысли. Если тестировщик спросит вас о чем-то, скажите нейтральным тоном: «Извините, я не могу ответить».

Это правило неудобно с социальной точки зрения. Если игрок смущен или расстроен, для него такая игра может быть попросту болезненной. Каждый опытный дизайнер наблюдал, как игрок застревал минут на пятнадцать в поисках двери или кнопки. Вы отчаянно хотите подсказать игроку: «Вот же она! Просто нажмите синюю кнопку!» Но так вы испортите весь тест, дав тестируемому ту информацию, которой не будет у реальных игроков. Это уже будет не тестирование игры, а ее странная версия, в которой дизайнер приходит в определенное пространство и раздает советы. Тесты могут идти более или менее гладко, но только потому, что недостатки скрыты.

Иногда, чтобы заполнить недостающие фрагменты игры, необходимо предоставить игроку информацию. В этих случаях ее необходимо включить в протокол тестирования заранее.

Как выбрать тестировщиков

Выбор тестировщика влияет на тип данных, которые вы получите. Основное различие между тестировщиками заключается в их знании игры.

В так называемой *тестировке Kleenex* дизайнер приглашает тестировщиков, которые никогда не играли в эту игру. Этот вид тестирования показывает, как игроки будут реагировать в первые критические моменты игры. Но этих тестировщиков можно приглашать только один раз, отсюда и название данного вида тестирования.

В других случаях мы хотим проверить наличие в игре баланса высокого мастерства. Для этого нужны игроки, которые смогут играть интенсивно в течение длительного времени. Обычно это означает наличие команды специализированных тестировщиков, которые ежедневно работают над своими навыками.

Между этими крайностями есть различия. Например, в процессе разработки боев я тестировал игру на коллегах, которые знали игру, но не знали, над каким именно боем я работал. Таким образом, их первоначальные знания игры были близки к знаниям реального игрока. Они знали игру, но не конкретный бой, который я разрабатывал.

Тестировщиков можно выбирать и по другим критериям. Вы можете протестировать игру на детях или пожилых людях, людях разных культур, социально-экономических традиций или интересов. В общем, выбирайте набор тестеров, похожих на людей, с которыми вы хотите сыграть в финальной игре.

Размер выборки

Можно запросто заикнуться на одном-единственном результате плейтеста. Поскольку ваш мозг инстинктивно верит, что то, что вы видите, то и существует (WYSIATI), вы попадетесь в ловушку, думая, что этот опыт — это и есть вся игра. Но часто оказывается, что первый тестовый запуск был всего лишь одним незначительным шагом сквозь большой и разнообразный набор возможных опытов. Вот почему хороший плейтест предполагает много плейтестов.

Без такого широкого ментального контекста дизайнеры будут стремиться решать задачи опыта, который им доступен, и при этом создавать проблемы в опыте, который недоступен.

Игра может продолжать постоянно меняться, но она не улучшится, потому что каждое решение приводит к еще большему количеству проблем.

Чтобы добиться реального прогресса, мы должны решать задачи с одним опытом, не вызывая проблем в другом месте. Это невозможно, если мы увидели только один или два потока, в которые игроки могут попасть. Мы должны знать все, что игра дает игрокам. Тогда мы сможем выбрать подходящие решения.

Хорошие решения могут быть приняты только в том случае, если дизайнер понимает все возможные опыты, которые может генерировать игра. Это предполагает множество плейтестов.

Процесс получения этого контекста прост: смотреть много плейтестов. Каждый тестировщик показывает вам новый поток в пространстве возможностей игры. Как только вы наберете достаточно информации, вы сможете сформировать полноценную модель всех опытов, которые может предложить игра. Вы будете знать обо всех развилках и возможностях, которые могут возникнуть в любой ситуации, и о том, как они взаимосвязаны. Вы сможете предсказать все различные последствия изменения дизайна, потому что вы будете понимать игру как систему, а не как историю.

Для этого нет готового списка плейтестов. Разные игры генерируют разный опыт, поэтому некоторые из них нужно тестить дольше, прежде чем дизайнер сможет их понять. Для очень простой, ограниченной игры может потребоваться от двух до трех плейтестов. В боевом шутере — обычно от 6 до 12 плейтестов. В неограниченных системных играх необходимое количество плейтестов может быть весьма большим.

Хорошее проверенное правило — прекратить плейтестинг, когда вы видите, что тестировщики часто повторяют один и тот же опыт. Как только это произойдет, можете быть уверены, что понимаете достаточно из того, что может предложить игра, чтобы принимать правильные дизайнерские решения.

Методика опроса

Мы можем узнать большую часть того, что нас интересует, просто наблюдая за плейтестом. Тестировщик проиграет, тем самым показав нам, где игра слишком сложная, и мгновенно выиграет, показав, где она слишком проста. Пропустив инструкции или удобный случай, он покажет нам, где игра непонятна.

Но иногда одного наблюдения недостаточно. Иногда нам нужно понять, что произошло в голове у тестировщиков. Иными словами, мы должны спросить у них.

Проблема в том, что устные сообщения ненадежны. Воспоминания редактируются или придумываются полностью. Отчет об опыте смешивается с предложениями по дизайну. Зацикленность тестировщика на дизайнере или студии затуманивают его мнение об игре. Тестировщик не делает этого намеренно; такова человеческая природа. Поэтому, чтобы узнать что-либо со слов тестировщика, мы должны составлять свои вопросы очень тщательно.

Мой любимый вопрос, который я задаю после тестирования, звучит примерно так: «Расскажите историю о том, что только что произошло в игре». Этот вопрос на проверку памяти. С его помощью можно выяснить, какие особенности игры игрок понял, запомнил и посчитал достаточно важными, чтобы их упомянуть. Вещи, которые он не упоминает в своем рассказе, могут быть балластом дизайна. Часто я обнаруживал, что история, которую

помнят игроки, очень отличается от того, что я написал, или от того, что произошло на самом деле.

Дизайнер также может адаптировать вопрос, чтобы определить, понял ли игрок конкретную вещь. Не стоит спрашивать: «Вы заметили дверь слева?», потому что сам вопрос дает игрокам подсказку и ответ может быть необъективным. Они чаще всего отвечают «да», просто чтобы не выглядеть глупо или понравиться интервьюеру. Лучше спросить: «Расскажите мне, почему вы выбрали этот путь». Тестировщик либо упомянет про дверь слева и объяснит, почему он не вошел в нее, либо промолчит. В первом случае это говорит о том, что ее заметили и проигнорировали, а во втором — что ее могут вообще никогда не заметить.

Держитесь профессионально и открыто. Наблюдая за тестировщиками или слушая их отзывы об игре, особенно если они не понимают ее так, как она была задумана, можно очень легко разочароваться. Но любое внешнее проявление этой эмоции заставит тестировщиков замолчать и перестать отвечать на вопросы честно. Тестировщики делают вам одолжение, поэтому относитесь к ним с благодарностью.

Тестирование методом «серого ящика»

Глупо создавать полномасштабное аудиосопровождение и графику для проекта только для того, чтобы во время плейтеста выяснилось, что он не работает. Чтобы избежать этого, мы можем выполнить итерацию методом «серого ящика».

Я тестировал методом «серого ящика» в процессе проектирования боя, но этот метод подходит не только для тестирования уровней, его можно применять практически везде. Внутриигровое видео можно заменить неподвижными изображениями или всплывающими окнами со статическим текстом. Сложные интерфейсы можно заменить помеченными командными кнопками. Звуки можно воспроизвести с помощью дешевых синтезированных гудков и жужжания. Диалог может читать программа преобразования текста в речь или он может отображаться на экране в виде текста.

«СЕРЫЙ ЯЩИК» — это черновой прототип игровой механики, системы или уровня.

Когда компания-разработчик BioWare создала Mass Effect 3, дизайнеры тестировали существ методом «серого ящика». На раннем этапе разработки гигантский боевой робот представлялся в виде большого куба с двумя длинными прямоугольниками внизу и двумя кубами, прикрепленными к бокам для оружия. Другой враг — высокий желтый блок — хватал героя с длинными желтыми блоками, прикрепленными к бокам. Это выглядит странно, но именно так и происходит. Эти «серые ящики» позволили дизайнерам BioWare тестировать и итерировать свои работы, не вкладывая средства в графику непроверенных проектов.

Тестирование методом «серого ящика» ускоряет итерацию. А «серый ящик» можно тестировать как готовую игру и скомпилировать его в разы проще. Поскольку большинство идей не работают, нет смысла реализовать их все с самого начала, используя все возможности графики. Но если мы сначала компилируем все в «сером ящике», мы можем позволить себе несколько ошибок, прежде чем наконец получим правильную работу механики. Только после того, как дизайн проверен, мы можем вкладывать ресурсы для его реализации, включая добавление аудиовизуальных эффектов.

Некоторые люди беспокоятся о том, что тестирование методом «серого ящика» влияет на художников, звукорежиссеров и других создателей контента. На первый взгляд кажется, что они могут разочароваться, если их попросят просто «нарисовать» серые фигуры. В действительности художники обычно ценят такое тестирование, потому что в этом случае сделанную ими работу отбрасывают гораздо реже. Без тестирования «серым ящиком» художники вынуждены работать над непроверенными проектами, поэтому большая часть их работы неизбежно урезается по причинам, не связанным с самой графикой. Но работая над хорошо протестированным «серым ящиком», у художника больше гарантий, он верит в то, что его труд не напрасен. Еще лучше, если художники будут общаться с разработчиками на этапе такого тестирования, чтобы дать представление о том, во что этот «серый ящик» превратится. Таким образом, они уже внесли свой вклад в «серый ящик», который их просят украсить, поэтому они уже понимают и верят в него.

Что не нужно тестировать методом «серого ящика»

«Серые ящики» дают нам возможность тестировать большую часть игрового опыта, включая механику и задумку. Но они не представляют собой весь опыт. Очевидно, что «серые ящики» не могут генерировать эмоции, как это делают музыка и графика.

Поэтому чем более аудиовизуально насыщенным является игровой опыт, тем менее полезным становится тестирование методом «серого ящика». Такие игры, как LIMBO и Flower, сильно пострадают в «сером ящике», поскольку в основном они основаны на аудиовизуальных эмоциональных триггерах. Однако этот метод весьма хорошо подходит для Counter-Strike и StarCraft II, так как эти игры всегда были основаны на механике.

Преждевременный продакшн

Всегда остается соблазн «сломать серый ящик» и начать использовать лучшие возможности слишком рано. Я называю это «преждевременный продакшн».

В краткосрочной перспективе добавление аудиовизуальных материалов в незавершенный дизайн выглядит круто. Графика и звук заставляют сердца трепетать и вызывают улыбки на совещаниях по проекту. Проблема в том, что эта эмоциональная выгода недолговечна, а за ее получение приходится платить снова и снова на протяжении всего процесса. С этого момента любое изменение графики, которое должно соответствовать изменяющейся механике, тормозит любой итерационный цикл. В конечном счете преждевременное добавление графики приведет к затратам значительно большим, чем те усилия, которые потребовались для ее создания. И платить за это придется еще долго после того, как первичный эмоциональный подъем утаснет.

ПРЕЖДЕВРЕМЕННЫЙ ПРОДАКШН — преждевременное добавление дизайнером графики и звука к дизайну «серого ящика» для того, чтобы получить следующий набор контрольных данных.

Хуже того, преждевременный продакшн ограничивает конечное качество игры. Нам всегда не хватает места для аудиовизуальных материалов, в результате чего механика накладывает ограничения на качество игры. Но если мы спрятали слабое ядро механики за графикой, то мы не сможем исправить это, не отказавшись от графики. В конечном итоге все закончится несовершенной

механикой, которую мы не можем изменить из-за созданной вокруг нее графики.

Чтобы оставаться в «сером ящике», нужна дисциплина. После неудачного плейтеста идея быстро скрыть недостатки дизайна графикой кажется весьма заманчивой. Но если графика не даст нужных результатов в следующем плейтесте, это будет означать ошибку. Чтобы получить полезные тестовые данные, графику следует добавлять как можно позже.

Навык оценки «серого ящика»

Играть в хороший «серый ящик» — совсем не то же самое, что играть в хорошую игру. Это означает, что оценка «серых ящиков» — это навык, которому нужно учиться на практике. Надо было оценить много «серых ящиков», чтобы научиться понимать, что такое хороший и плохой «серый ящик». Без этого навыка мы, скорее всего, не примем даже идеальный «серый ящик» просто потому, что в нем нет графики.

В связи с этим возникают споры в процессе принятия групповых решений, когда некоторые люди не имеют способности оценивать «серые ящики». Они смотрят на дизайн, и он им не нравится только потому, что он выглядит ужасно. Это гало-эффект в действии — плохое качество визуальных эффектов создает эмоциональное впечатление, которое становится чьим-то мнением обо всем дизайне. Поэтому дизайн не принимают, даже если он хорош.

В реальной жизни это обычно самая большая проблема «серых ящиков». Поэтому будьте внимательны с тем, кто принимает решения о дизайне «серого ящика». Этим должен заниматься только опытный специалист, поскольку неопытные тестировщики склонны принимать плохие решения в результате гало-эффекта. Если они должны принять такие решения, возможно, придется включиться в преждевременный продакшн, несмотря на стоимость.

Миф о сценарии

Многие считают, что игра должна начинаться с большого дизайн-документа, потому что именно так создаются фильмы. Но эта метафора неверна.

Сценарий включает все события фильма. Каждое слово диалога, каждое изображение, каждый поворот сюжета — все это прописано в сценарии. Когда мы читаем его, наш мозг должен восполнить недостающие аудиовизуальные детали, это легко сделать с помощью активного визуального воображения. Так что просто читая сценарий и представляя изображения, мы можем приблизиться к опыту киноманов.

Нельзя сделать то же самое в геймдизайне, так как дизайн-документ не определяет игровые события — он определяет игровую механику. Прочитать дизайн-документ и понять окончательный опыт — это не просто представить визуальные эффекты, а мысленно смоделировать всю игровую механику и выбор игроков, чтобы создать события, которые управляют этим опытом. С таким моделированием справится далеко не каждый.

Но в «сером ящике» игра производит механическое моделирование. Разум игрока должен только заполнить недостающий аудиовизуальный ряд, как при чтении сценария. Таким образом, наиболее близким к сценарию является рабочий «серый ящик». Он предоставляет столько же информации о конечном продукте, сколько и сценарий, в то время как дизайн-документ предоставляет гораздо меньше.

Парадокс качества

Мудрость гласит: «Семь раз отмерь, один раз отрежь». Когда вы строите палубу или дамбу, в этом есть смысл. Так как ошибки исправить в такой работе стоит очень дорого, лучше всего избегать их. Но в геймдизайне крайняя степень неприятия ошибок приводит к ухудшению качества продукта.

Классический совет гласит, что если вы будете работать медленно, с любовью, уделяя внимание каждой детали, то вы получите качественный продукт. Если вы быстро сляпаете разные куски воедино, то получите мусор. С этой точки зрения завершение качественного продукта означает выполнение качественной работы на каждом этапе процесса.

Ближайшим эквивалентом сценария в геймдизайне является не дизайн-документ, а рабочий прототип «серого ящика».

В геймдизайне временное принятие некачественной работы в конечном итоге приводит к более качественной работе. Это ПАРАДОКС КАЧЕСТВА.

Игры отличаются тем, что наиболее важным фактором, определяющим их качество, является количество итерационных циклов, через которые они проходят. Одержимость качеством на каждом этапе замедляет итерацию, что в конечном итоге ведет к ухудшению игры.

Вот почему не следует отказываться от несовершенной работы на ранних этапах итерации. Дизайнера, который так делает, можно сравнить с писателем, который не может сказать ни слова, потому что ему нужно, чтобы сказанное было идеальным. Таким образом, каждый цикл итерации растягивается в результате избыточного анализа, так как дизайнер пытается дважды измерить и получить идеальный результат. В конце концов, страх ошибиться приводит его к плохому результату, потому что он провел только несколько циклов итерации.

В геймдизайне, прежде чем игра достигает своего окончательного вида, все пересматривается и меняется множество раз. Работа, которую мы делаем на самых ранних циклах итерации, не создает финальную игру. Это только платформа.

Ошибочность видения

Молодой инженер авиационной и космической промышленности в свой первый рабочий день заходит в кабинет начальника с блеском в глазах и заявляет: «У меня есть гениальная идея для нового типа самолета».

Босс заинтригован.

«Говори», — отвечает он.

Молодой инженер задумчиво смотрит вдаль.

«Пассажиры спокойно заходят на борт за пять минут. Затем самолет тихо взлетает, пока пассажиры наслаждаются мартини в частных кабинках. Когда они летят над Атлантикой, молодая пара любит вид, находясь в одном из множества фонарей самолета, а милый ребенок осматривает кабину. Капитан смеется, когда ребенок спрашивает, почему они не могут полететь на Луну. К тому времени, когда

они приземляются, любовь уже найдена, уроки пройдены, и все готовы ко всему, что их ожидает в пункте назначения».

Босс откидывается на спинку стула и затягивается сигарой.

«Уволен», — говорит он.

У молодого инженера было видение. Но это было видение процесса полета, а не самого самолета. Он описал замечательный опыт, но ничего не сказал о механических системах, которые создали этот опыт. Это называется ошибочностью видения.

Люди склонны принимать решения, используя мыслительные образы. Мы воображаем историю, оцениваем, как образ заставляет нас чувствовать себя, и принимаем решение на основании этой эмоциональной реакции. Психолог Дэниел Гилберт называет эту технику предвидением.

Во многих случаях предвидение имеет смысл. Оно использует способность эмоционального бессознательного быстро генерировать детальное мнение о сложной идее.

Хотите пойти в кино? Предвидьте. Хотите съесть вон то блюдо? Предвидьте. Это простой, быстрый и зачастую эффективный способ принятия решений о будущем.

Мы делаем то же самое в геймдизайне, когда оцениваем потенциальную игру, воображая, что играем в нее. Особенно сильный ментальный образ часто называют видением. Оно может быть источником вдохновения, мотивируя так, как могут мотивировать только истории.

Видение может быть обманчивым и определять опыт. Но игра — это не опыт, это система создания опыта. Точно так же, как не стоит путать идеальный полет с идеальным самолетом, так же не стоит путать видение отличного игрового опыта для создания великолепной игры.

Видение ничего не говорит о плюсах и минусах, а также издержках системы, которая стоит за опытом. Оно также ничего не говорит нам обо всех других опытах, которые будет генерировать игра. Это важно, потому что игроки не просто получают

ОШИБОЧНОСТЬ
ВИДЕНИЯ заключается
в приравнивании
воображаемого об-
раза опыта к дизайну
системы, которая
создает этот опыт.

в игре лучший опыт — они получают весь опыт. Кроме того, видение всегда скрывает недостатки дизайна, потому что для нас естественно представлять только лучшее в игре, которую мы создаем. Мы представляем захватывающую битву, а не пятиминутную прогулку от базы. Мы представляем, что предохранитель сработал, а не сломался. Мы видим только хорошую сторону, а наш мозг убирает плохую. Эта модель порождает чрезмерную уверенность в планах разработки и приводит к избыточному планированию. Таким образом, хотя дизайнеры и должны искать мотивацию в видении, необходимо ставить под вопрос ее надежность.

Попробуйте такой антидот против ошибочности видения: вместо того чтобы пытаться представить лучший опыт, который может создать игра, попытайтесь представить худший. Старайтесь в деталях представлять каждую неудачу, скучный гринд и непонятное взаимодействие. Здесь вам придется приложить когнитивные усилия, а не представлять в уме замечательные образы. Но так вы получите намного больше информации, потому что увидите сбалансированную картину игры, вместо того чтобы выбирать лучшие результаты.

Серендипность

Есть известные известные — вещи, о которых мы знаем, что знаем их. Есть также известные неизвестные — вещи, о которых мы знаем, что не знаем. Но еще есть неизвестные неизвестные — это вещи, о которых мы не знаем, что не знаем их.

Дональд Рамсфелд

В геймдизайне мы сталкиваемся со многими неизвестными явлениями. Поймет ли игрок этот инструмент? Не слишком ли это задание сложное? Сколько времени понадобится, чтобы построить этот уровень? Возможно, на эти вопросы трудно ответить, но это не самое важное незнание, с которым нужно что-то делать.

Мы совершаем ошибки, даже не замечая этого. Мы проходим мимо возможностей, которых даже не видели. Мы основываем целый дизайн на предположениях, которые мы фактически не делаем, даже не догадываясь об этом.

Некоторые неизвестные неизвестные приводят к катастрофе. Тестировщик находит вырожденную стратегию, которую сложно исправить и которая нарушает всю игровую систему. Очевидный на первый взгляд интерфейс оказывается непонятным для новичков. Мы получаем дикое новое указание от издателя или ведущий программист уходит на больничный. В отличие от запланированных методов, итерационные процессы устойчивы к таким вещам.

Большинство действительно важных вещей в разработке игр возникают из неизвестных неизвестных.

В случае итерации мы не делаем предположений о далеком будущем. Иначе говоря, мы можем быстро изменить направление в ответ на меняющиеся обстоятельства. Более того, постоянные проверки итераций в реальных условиях означают, что катастрофические открытия обычно обнаруживаются рано. Это само по себе является основной причиной для итераций.

Но есть еще один, часто более важный вид неизвестного: серендипность, или прозорливость. Игроки влюбляются во второстепенного персонажа. Изобретают новую интересную тактику. Находят эмоции в, казалось бы, неважной части игры. Это положительные результаты, которые дизайнеры так и не увидели. И часто эти случайные открытия являются наиболее ценными из того, что происходит в процессе разработки. Такая проникаемость необходима для создания революционных проектов, потому что большинство революционных геймдизайнов не создают намеренно, а открывают случайно.

Например, прародителем видеоигр в жанре RPG является Dungeons & Dragons. D&D генерировала большую часть своего наполнения из своих ролевых элементов, так как позволяла игрокам вербально разыгрывать любую фэнтезийную историю, какую они только могли представить. Как только компьютеры стали способны генерировать аналогичный опыт, D&D преобразовали в видеоигру Rogue. Rogue отображает карту подземелья

в текстовой графике. Игрок управляет героем, который исследует просторы сгенерированного случайным образом подземелья, убивая монстров, набираясь опыта и собирая древние сокровища. На первый взгляд Rogue может показаться довольно близкой компьютеризированной версией D&D. Но она создает опыт совсем другим способом. D&D работает в основном через ролевое взаимодействие и социализацию, а Rogue — через эмергентную историю и получение награды по графику. В свое время это были революционные достижения в геймдизайне. Но дизайнеры Rogue этого не планировали. Пытаясь скопировать опыт D&D, они наткнулись на силу режимов подкрепления и эмергентный нарратив, стимулируемый апофенией. Игра работала фантастически по причинам, которые ее создатели никогда не смогли бы предсказать.

Серендипность не является чем-то необычным. Знаменитый персонаж Большой папочка в BioShock изначально был мутантом в водолазном костюме; добавление слабых Маленьких сестричек привело к созданию очаровательных взаимоотношений между огромным големом-отцом и маленькой девочкой-дочерью. Голос GLaDOS в игре Portal, одного из самых популярных игровых персонажей, стал роботизированным только когда Эрик Волпо заметил, что людям больше понравился «серый ящик» в исполнении синтезатора голоса. Безупречный финальный уровень Braid разработчики обнаружили, когда игра была практически закончена, Джон Блоу понял, как он мог использовать механику, сдвигающую время, чтобы обратить вспять не только время, но и персонажа. Игра Tetris появилась из компьютерной версии традиционной русской игры-головоломки «Пентамино»¹.

The Sims были разработаны на основе симуляции архитектуры, когда Уилл Райт заметил, что игрокам нравится играть с персонажами больше, чем строить дома. Даже изначальный хит Райта

¹ Автор ошибается. Тетрис возник из фигур «Тетрамино». «Тетрамино», «Пентамино» и «Гексамино» — наборы всех возможных комбинаций из четырех, пяти и шести квадратов соответственно. Головоломки с ними были популярны в СССР и публиковались, например, в журнале «Наука и жизнь». Очевидно, что А. Пажитнов знал о них, когда приступал к созданию Tetris. — *Примеч. ред.*

SimCity был разработан, когда он заметил, что ему нравится создавать карты для вертолетных боев больше, чем взрывать их.

Серендипность — одно из самых больших преимуществ итерации. Для тех, кто любит глубокое и тщательное планирование, серендипность означала бы отказ от любимого и дорогостоящего плана. Часто вместо этого они отказываются случайным образом. Когда мы итерируем, нам не нужно этого делать, потому что будущее открыто и мы можем заполнить его новыми открытиями по мере их появления.

Серендипность не происходит просто так. Мы должны быть готовы к ней. Ключом на пути к этому являются наблюдательность и желание открывать новое. Она проявляется как странное поведение или бессмысленные результаты в хорошо понятых системах. Нам остается заметить эти намеки и копнуть глубже.

Закрытый человек не может этого сделать, так как его ментальная модель недостаточно гибкая, чтобы воспринимать новые идеи. Он увидит подсказку, но проигнорирует или прикроет ее, чтобы укрепить свою собственную точку зрения.

Чтобы использовать серендипность, дизайнер должен уметь пересматривать свои мысли на основании своих наблюдений, а не подстраивать свои наблюдения в соответствии со своим мировоззрением.

Как творческие люди, мы хотим проецировать наше видение на мир. Но чтобы контролировать серендипность, требуется ослабить авторский контроль. У отличных гейм-дизайнеров нет идеального видения игры, а затем они просто воплощают ее в реальность. Они копают глубже в поисках возможностей, следят за драгоценными подсказками и тут же хватают их.

Серендипные открытия в дизайне появляются не только благодаря удаче. Чтобы поймать их, мы должны наблюдать и адаптироваться.

Геймдизайн — это не просто процесс авторства. Это также процесс наблюдения и открытия.

Вера в итерацию

Трудно избавиться от привычки планирования. Я пытаюсь предоставить, что бы подумал, прочитав эту главу несколько лет назад.

Я мог бы кивнуть в знак согласия и подумать, что понял. Но сомневаюсь, что это на самом деле было бы так. Интеллектуальное понимание — это не то же самое, что эмоциональная вера.

Большинство знакомых мне дизайнеров, которые понимают итерацию, поверили в нее только спустя многие годы неудачных попыток, вызванных глубоким планированием. Они работали в цейтноте, срывали сроки, снова и снова наблюдали за крушением планов. Я делал то же самое. Возможно, единственный способ поверить в проблемы с планированием — это сначала пережить их самому.

Это не тот опыт, который можно получить легко. Это означает, что нужно пройти всю игру до конца — прототипов недостаточно. Масштаб игры должен быть значительным; тривиальные игры для школьников слишком просты, чтобы можно было противостоять планированию на их примере. В игру должны играть настоящие игроки, у которых нет причин жалеть дизайнера. Свою истинную ценность игра может проявить только в реальных условиях. И только тогда дизайнер получает неоспоримую, болезненную обратную связь, которая меняет эмоциональные убеждения.

Создание знаний

Игра — это не объект, который мы производим. Это система знаний. Это не комплект шахматных фигур. Это сам процесс игры в шахматы.

Возьмем, к примеру, процесс написания романа. Писателю не сложно набирать слова, ему сложно выполнять умственную работу по созданию взаимосвязанных персонажей, окружающей обстановки, тем и сюжетных поворотов. Главная проблема, с которой сталкиваются геймдизайнеры, — это не реализация игры, а изобретение механики, сюжета, графики и технологий, которые объединяются в мощный двигатель опыта. Это процесс создания знаний о дизайне, который мы собираемся реализовать. Это гораздо больше, чем просто придумывать идеи. Это значит изобретать эти идеи, совершенствовать их, тестировать, обсуждать и связывать воедино, чтобы они работали вместе. Делать это означает отвечать на многие вопросы и устранять многие неопределенности. Это означает, что мы должны создавать знания.

Сложная часть геймдизайна — это не физическая реализация игры. Это изобретение и совершенствование знаний о дизайне.

Методы создания знаний

Чтобы создать эти знания, мы применяем различные методы создания знаний. Плейтестинг, мозговой штурм, дискуссии, дебаты и фантазирование — все это методы создания знаний.

Думайте о каждом методе как о карте, которую мы можем разыграть. У разных карт разные возможности и разные ресурсы. Например, в методе мозгового штурма несколько разработчиков быстро придумывают множество неопробованных идей, в то время как для высококвалифицированного плейтестинга привлекаются тестировщики баланса, которые тщательно исследуют нюансы существующего дизайна.

Освоение процесса разработки означает знание того, какую карту разыграть и когда. Цикл итерации показывает нам хороший базовый подход, но само по себе это недостаточно. В действительности не существует простого алгоритма создания знаний. Чтобы получилось хорошо, мы должны постоянно реагировать на меняющиеся условия проекта. Для этого нужно очень хорошо знать все свои карты. Итак, давайте пройдемся по колоде.

Руминация

Руминация — это постоянное обдумывание проблемы в течение длительного времени. «Пережевывайте» идею часами, днями или годами, и в конечном итоге она сможет раскрыть свои секреты.

Разум автоматически размышляет при выполнении простых задач, таких как принятие душа, ходьба или вождение. Иногда размышления могут быть даже невольными. Все испытывали то чувство, когда поток мыслей не дает уснуть. Но даже такие размышления, какими бы неприятными они ни были, могут быть продуктивным (многие мысли, описанные в этой книге, родились и были записаны в 2 часа ночи).

Размышление также происходит неосознанно. Наше сознание может забыть о проблеме, а бессознательное — нет. Оно еще долго работает после того, как мы сдаемся. Если вы когда-нибудь внезапно поняли решение трудной задачи на следующий день после того, как не смогли решить ее, это значит, что вы собрали урожай бессознательного размышления.

Невозможно предсказать, когда бессознательное размышление даст результаты. Вот почему некоторые мыслители носят с собой блокноты, в которых записывают свои мысли и идеи. Они хотят успеть сорвать плод своего бессознательного размышления.

Одна из стратегий использования бессознательного размышления — чередовать работу над разными задачами. Работая над одной задачей, наше бессознательное размышляет над другой. Вот почему Эдисон, Дарвин, Леонардо да Винчи, Микеланджело и Ван Гог работали над несколькими проектами одновременно.

Для хороших размышлений нужны два ключевых ингредиента.

Во-первых, знания. Размышление работает, формируя новые связи между старыми мыслями. Чем больше у вас старых мыслей, тем больше возможных связей. Геймдизайнерам необходимы знания, связанные с игрой, поэтому дизайнеры должны много играть. Но даже те знания, которые напрямую не связаны с этой проблемой, могут подпитывать размышления. Знание экономики, истории, непальской культуры или техники ловли рыбы на мушку — все это может стать частью творческого решения сложных задач дизайна.

Люди думают аналогиями, поэтому наши мысли пополняются информацией, когда появляется больше знаний для сравнения.

Во-вторых, расслабление. Именно поэтому озарение приходит к людям, когда они принимают душ, сидят в яблоне-вом саду или едут в автобусе. Такие эмоции, как гнев, страх и сосредоточенность, препятствуют творчеству на неврологическом уровне. Исследования на МРТ показали, что незадолго до внезапного озарения кровь устремляется к части правого полушария мозга, которая называется передней верхней височной извилиной, или aSTG. Страх, гнев и интенсивная сосредоточенность на задаче на самом деле подавляют приток крови к aSTG, что, в свою очередь, подавляет свободные ассоциации и творческие способности. В этом есть смысл, если вы пещерный человек и пытаетесь сбежать от льва. Вы бы вряд ли захотели тратить время на то, чтобы представить львов, в то время как один из них уже практически вцепился вам в горло. Но это также означает, что сосредоточение — убийца творчества. Итак, если вы хотите размышлять, вы должны сначала расслабиться.

Исследование

Иногда мы проводим исследования, чтобы ответить на конкретные вопросы. Дизайнер уровней может изучать средневековую архитектуру для создания части игры про Крестовые походы. Разработчик систем, работающий над стратегической игрой, может поиграть в другую стратегию, чтобы выяснить, какая механика работает и как. Это то исследование, которому всех нас учили в школе.

Второй вид исследований гораздо менее целенаправленный. Это наполовину произвольное исследование, в котором мы учимся,

не обязательно понимая, как полученные знания будут применяться к проекту. Цель здесь не в том, чтобы отвечать на вопросы, а в том, чтобы расширить наш запас знаний и питать мозг.

Мы все проводим наполовину произвольное исследование ежедневно, черпая информацию из телевидения, кино, игр, интернета и нашей повседневной жизни. Мы собираем идеи, мемы и культурные стандарты повсюду. Но без какого-либо направления мы получаем те же знания, что и другие геймдизайнеры. Все в каждой игровой команде, с которыми я работал, знали о «Звездных войнах» и «Терминаторе-2». Эта культурная однородность является одной из причин, по которой многие игры очень похожи друг на друга. Мы заперты в мире героев, побеждающих драконов и мрачных космических морпехов, потому что поглощаем только идеи друг друга.

Чтобы избежать культурной эхо-камеры, дизайнер должен развивать нестандартные интересы. Геймдизайнер стратегий может сыграть в *The Sims* и получить новую идею для экономической системы. Он может прочитать книгу по микробиологии и подумать о новой системе производства биологических войск. Такие наполовину зарегистрированные исследования являются долгосрочной инвестицией. Они нужны даже не для конкретного проекта — это действительно предполагает богатую интеллектуальную жизнь, включающую множество уникальных идей, источников и опыта. Это то, что нужно развивать как по личным, так и по профессиональным причинам.

В геймдизайне богатая интеллектуальная жизнь окупается уникальной работой, которую она позволяет нам создавать. Например, разработчики *BioShock* смогли придумать уникальный мир Восторга только потому, что они ранее исследовали ардеко и объективизм. Без глубоких произвольных исследований *BioShock* никогда бы не появился.

Художественные методы

Вспомните, когда вы в последний раз рисовали на бумаге. Может быть, вы были на уроке рисования в школе или рисовали закрючки на деловой встрече. В любом случае, у вас почти наверняка был такой опыт: вы рисуете фигуру, а затем замечаете, что опре-

деленная ее часть напоминает вам о чем-то совершенно ином. Развивая свою мысль, вы совершаете ошибку. Чтобы скрыть ее, вы придумываете новый объект, который хотите нарисовать сверху, чтобы перекрыть старый. Этот новый объект заставляет вас менять окружающие формы. Процесс так продолжается, одно изменение вдохновляет другое. Когда вы закончили, вы видите, что нарисовали что-то совершенно не то, что планировалось изначально. В процессе создания художественной работы вы придумали новую идею.

Сила искусства заключается в том, что вы работаете руками, одновременно записывая свои идеи. Погружаясь в поток, мы чувствуем меньше запретов. На бумагу мы можем записывать мысли, которые невозможно передать словами. Делая ошибки и сталкиваясь с физическими ограничениями, мы идем к новым идеям.

Различные виды художественных процессов обеспечивают доступ к этим возможностям по-разному. Благодаря концептуальному искусству можно исследовать образы персонажей, настроение или пространство. Раскадровки устраняют неоднозначность во фрейминге, цвете и последовательности изображений. Благодаря предварительной визуализации можно исследовать различные способы передачи идеи. Когда Pixar работал над *The Incredibles*, он создавал бесформенные цветовые свотчи для каждой сцены, чтобы понять визуальную и эмоциональную прогрессию фильма только по цветам. Некоторые художники даже создают скульптуры существ или персонажей.

Орсон Скотт Кард, автор *Ender's Game*, описал свой процесс изобретения фэнтезийных вселенных. Он достает гигантский чистый лист и начинает рисовать карту. Он набрасывает на ней города, достопримечательности и особенности местности, придумывая им названия. Он ничего не планирует; он изобретает мир, рисуя его. Давая название каждому из городов, он задумывается об его истории, обществе, которое его создало, и о причине его существования.

Каждая ошибка и исправление приводят к появлению нового подхода и идеи. Получившаяся карта представляет собой подробное описание нового фантастического мира.

Этот метод распространяется на неизобразительные искусства. Иногда, находясь в поиске сюжетных идей, я задействовал короткие рассказы, чтобы мое творческое мышление заработало. Истории получались ужасными, но процесс следования истории персонажа, живущего в вымышленном мире, всегда создает идеи для глобальной истории. Люди, ориентированные на результат, могут импровизировать, чтобы экспериментировать с персонажем. Профессионал в аудио может создать звуковую среду. Возможности безграничны.

Мозговой штурм

Мозговой штурм — это полуформальный процесс, предназначенный для быстрого генерирования большого количества разнообразных идей. У разных людей и организаций свои особенности проведения мозговых штурмов. Некоторые из них почти не структурированы, в то время как другие назначают лидеров, которые будут контролировать поток обсуждений и записывать идеи. Мозговой штурм — хорошо известный метод, поэтому я не буду описывать его подробно.

Мозговой штурм хорошо подходит для генерирования идей в большом количестве. Он не подходит для улучшения идей, а идеи, которые появляются в результате мозгового штурма, будут сильно различаться по качеству.

Письменный анализ

Письменный анализ — это форма структурированного мышления. В этой книге я мало писал о документах геймдизайна, но они важны и у них есть свое место. При написании документа мы думаем иначе, как если только прокручиваем мысли в голове. Документ может включать диаграммы, ссылки и хорошо продуманные логические цепочки. Благодаря письменному анализу мы размышляем о деталях так, как не стали бы размышлять спонтанно или если бы обсуждали идею с коллегами.

Глубокий анализ сложного проекта с широкими исследованиями и использованием формальных статистических или математических методов может занять несколько недель. Иногда оно того

стоит, потому что в результате можно получить информацию, которая не появится просто так.

Дебаты

Дебаты ведутся с определенной целью: они помогают находить недостатки в идеях. Они занимают огромное место в формальных системах принятия решений, таких как суды, демократия и наука, так как с их помощью можно выявить скрытые предположения и логические ошибки. Но вести дебаты нелегко. Продуктивные дебаты требуют специфического сочетания навыков и социальных условий.

Оба участника дебатов должны быть опытными. Они должны знать, как быстро и эффективно оспаривать и защищать аргументы. Плохой участник дебатов будет тратить время на оспаривание сильных аргументов или пропускать логические ошибки, допущенные противником, делая весь процесс менее продуктивным и медленным.

Участники дебатов должны иметь разностороннее мышление. Это означает, что они обладают разными знаниями, мнениями, опытом и предположениями. Так они могут находить слабости в чужих идеях. К сожалению, разностороннее мышление редко встречается в организациях, потому что там, как правило, нанимают других людей, которые думают так же, как сотрудники организации. Чтобы развить разностороннее мышление, требуются определенные усилия.

Оба участника дебатов должны уважительно относиться друг к другу. Они должны быть в состоянии отделить свои личные чувства от логического процесса дебатов. В противном случае процесс может разрушить их взаимоотношения и перестать быть продуктивным.

Наконец, участники дебатов не должны бояться друг друга. Дискуссия легко может уйти не в то русло в случае дисбаланса власти. Человек, у которого меньше полномочий, не будет честно спорить из-за страха возмездия. Вот почему дебаты лучше всего проводить среди людей, которые вообще не имеют власти друг над другом. Чтобы начальник мог вступить в дебаты со своими

подчиненными, он должен сначала доказать свою беспристрастность. Некоторые начальники этого не делают. Они наказывают людей за несогласие, иногда делая это невольно, злобно вздыхая или закатывая глаза. Такое поведение заставляет людей мягче относиться к плохим идеям начальника, что делает дебаты менее полезными.

Тестирование

Мы уже рассматривали плейтестинг. Это самый важный, но не единственный вид тестирования игр. Существуют различные виды тестирования, каждый из которых дает различные виды информации.

Тестирование юзабилити — это метод, заимствованный из дизайна программного обеспечения, который сосредоточен на интерфейсах и элементах управления. Он похож на плейтестинг тем, что предполагает участие реального пользователя, которого сажают перед интерфейсом и смотрят, как он будет его использовать.

QA-тестирование выполняется специализированными тестировщиками и направлено на поиск технических ошибок. Оно важно в любом процессе разработки видеоигры.

Термин *фокус-тестирование* часто ошибочно используется как синоним *плейтестинга*. Это неверно. Фокус-тестирование — это форма исследования рынка, в которой участники совместно обсуждают различные идеи касательно продукта. Он не требует работающей игры.

Метрики

В геймдизайне термин «метрики» относится к данным, которые автоматически собираются из игровых сессий. Игра может записывать информацию о разграбленных объектах, проваленных заданиях, времени завершения, побежденных врагах, исследуемых областях или сотне других событий. Метрики можно получить на основании результатов сотен внутренних тестировщиков или миллионов игроков, играющих в общедоступную бета-версию. Затем компьютеры обрабатывают все эти числа в статистических

отчетах и графиках, а дизайнеры используют их для принятия проектных решений.

Метрики помогают дизайнерам увидеть паттерны и дисбалансы, которые можно пропустить в плейтестах. Например, в файтинге метрики покажут, что один персонаж выигрывает у другого в 55% случаев, потому что он может выбирать тысячи матчей. Подобные точные данные нельзя получить на основании плейтестинга, поскольку размер выборки слишком мал. Вот почему метрики неоценимы при тонкой настройке. Не умея видеть маленькие достижения, вы будете замечать только большие изменения в результатах, и прогресс становится прерывистым и неритмичным. Показывая даже небольшие изменения в сложности или ритме, метрики позволяют нам подниматься вверх размеренными шагами и достигать успеха.

Кен Бердвелл пишет о своем опыте использования метрик для тонкой настройки Half-Life:

«К середине проекта, когда основные элементы были созданы и можно было пройти большую часть игры, дело в основном оставалось за тонкой настройкой. Для этого мы добавили в игру базовые инструменты, автоматически записывая положение игрока, его здоровье, оружие, время и любые другие важные действия, такие как сохранение игры, смерть, травмы, прохождение квеста, бой с монстром и так далее. Затем мы взяли результаты нескольких сессий и собрали их воедино, чтобы найти проблемные места. Они включали области, где игрок пробыл слишком долго без каких-либо стычек (скучно), слишком долго со слишком большим количеством здоровья (слишком легко), слишком долго со слишком небольшим количеством здоровья (слишком сложно), и все это дало нам хорошее представление о том, где персонажи игроков, скорее всего, умрут и какие позиции наиболее выгодны для добавления функциональных возможностей».

Когда в 1998 году был разработан Half-Life, занимаясь балансом, другие студии полагались на предположения и самостоятельное

тестирование. Эти методы полезны на ранних стадиях разработки, но они не позволяют детализировать более подробные данные, необходимые для идеального баланса степени сложности. Благодаря метрикам качество дизайнерских решений принесло огромное преимущество разработчикам Valve. И им не нужно было быть умнее всех, чтобы создать лучшую игру, потому что они работали на свету, а все остальные работали в темноте.

В дополнение к корректировке метрики также позволяют дизайнерам находить редкие пиковые ситуации. Возможно, 20 привлеченных вами тестировщиков не найдут эту вырожденную стратегию, но если ее обнаружит один из миллиона игроков в общедоступной бета-версии, то это будет отображаться как пик данных.

Иногда мы можем использовать продуманные метрики для сбора данных, которые на первый взгляд получить невозможно. Например, работая над игрой Halo: Reach, разработчики компании Bungie хотели получить больше информации о влиянии медленной игры в сети на опыт. Игра уже могла записывать файл кинофрагмента, включая каждую игровую переменную в любой момент, включая лаги. Но этого было недостаточно — дизайнеры хотели знать, как игроки воспринимают лаги, которые сложно увидеть в данных. Решить эту проблему с помощью плейтестинга было невозможно, поскольку многие проблемы с лагами встречаются настолько редко, что дизайнерам потребовалось бы огромное количество времени на просмотр достаточного количества плейтестов. Более того, для реального сетевого тестирования требуется, чтобы игроки были рассредоточены, что делает практически невозможным следование традиционным протоколам плейтеста.

Разработчики решили проблему, добавив специальную кнопку: «Я только что увидел лаг». Они регистрировали количество нажатий, а также остальную игру. Как только данные поступали, дизайнеры просматривали изображение и переходили в те точки, где игроки нашли лаг, чтобы точно увидеть то же, что и на экране игрока. Благодаря данному нестандартному методу дизайнеры получили огромное количество достоверной информации о восприятии лагов при относительно небольших затраченных усилиях

разработчиков. Таким образом, они решили многие проблемы с восприятием лагов, и Halo: Reach стала отличной сетевой игрой. Разработчики получили прекрасный чистый код, и вовсе не потому, что они были светилами программирования. Как и сотрудники Valve, они работали с большим количеством информации.

Изобретенные методы

На некоторые вопросы невозможно ответить установленными методами. В этих случаях мы должны вернуться к началу и придумать новый способ получения необходимой информации.

Создавая игру для людей в возрасте, разработчику, возможно, придется придумать специальный протокол плейтеста. Игра с необычной внутриигровой экономикой может потребовать новых способов анализа и интерпретации данных. Команде разработчиков, работающей на разных континентах, возможно, придется вступать в дебаты и проводить мозговые штурмы не так, как это делают разработчики, находящиеся рядом. Изобретение и улучшение инструментов создания знаний являются неотъемлемой частью геймдизайна.

Органический процесс

Давайте рассмотрим классический пример процесса создания знаний. Этот пример не из геймдизайна, а из изобретательства.

С наступлением XX века не одна команда пыталась изобрести летательные аппараты тяжелее воздуха.

Некоторых на это толкало скорее хвастовство, чем способности. Они привязывали плавники или крылья к телу и пытались взлететь, многие погибли, испытывая свои безумные изобретения, прыгая с мостов или холмов.

Другие были более серьезными. У этих благородных европейских изобретателей и известных американских ученых, казалось, были все преимущества. Одним из самых богатых был Сэмюэл Лэнгли, секретарь Смитсоновского института, который использовал свои связи, чтобы получить 70 тысяч долларов из государственного финансирования США.

Но устройства Лэнгли так и не заработали, ничего не получилось и у его богатых коллег. Зато все получилось у Орвила и Уилбура Райтов, ремонтников велосипедов из Огайо, и они добились того, что человек сумел полететь. Они потратили менее тысячи долларов.

Как? Как два человека без особых связей обошли крупнейшие государственные проекты в мире, имея менее чем 2% бюджета?

Благодаря умению создавать знания. Райты не просто нарисовали свой проект на бумаге, построили его и полетели. Они применяли головокружительное множество методов создания знаний в течение нескольких лет. Они провели сотни испытаний, многие из которых требовали изобретения новых методов и устройств для испытаний. Они спроектировали и построили планеры, сечения аэродинамических поверхностей и руль управления полетом, и каждое из этих устройств было разработано для решения конкретной задачи. Они использовали математические расчеты, полевые и лабораторные испытания, руминацию, аргументацию и дебаты, а также исследования и разработки. На каждом этапе они выбирали наиболее подходящий метод для решения следующего неизвестного. С каждым последующим полетом планера, изменением его дизайна и новыми расчетами они узнавали все больше нового о том, как летать.

Они были очень упорными. На своем планере 1902 года братья провели 1000 испытательных полетов, снова и снова модифицировали его и каждый раз узнавали что-то новое. Отдельные полеты мало чему научили их, но по мере накопления знаний Райты начали осваивать небо.

Полученные знания, а также источники их получения были очень разнообразны. При тестировании разных вариантов сечений аэродинамических поверхностей в аэродинамической трубе они узнали, что традиционное уравнение подъемной силы было неверным. Во время полевых испытаний планера они узнали, что установленный спереди горизонтальный стабилизатор позволяет самолету осуществлять мягкую посадку «на живот». И у них были моменты озарений, порожденные бессознательным размышлением: думая над управлением креном самолета, Уилбур

безучастно крутил в своей мастерской длинную коробку и вдруг понял, что для того, чтобы контролировать крен самолета, можно изменить форму крыла. В результате это привело к появлению системы управления круткой крыла.

В то же время Лэнгли и другие также использовали свои ресурсы, но делали это неправильно. Большинство из них были сосредоточены на создании более мощных двигателей. В то время проектирование двигателей было традиционной деятельностью, поэтому деньги можно было легко вложить в создание более совершенных двигателей. Однако Райты понимали, что дело было не в двигателях. В то время как о двигателях было известно многое, об аэродинамике не было известно почти ничего. Платить за лучшие аэродинамические системы управления было попросту некому; практически никто не был готов заниматься исследованиями. Так что Райты сделали это сами, используя свои собственные методы создания знаний.

И Райты стали мастерами по экспериментальному проектированию и конструированию. Они построили самую точную в мире аэродинамическую трубу и использовали ее для испытания 200 различных форм крыльев. Когда выяснилось, что испытания методом проб и ошибок с использованием полноразмерного планера стоят слишком дорого, они создали устройство, похожее на велосипед, к которому можно было прикреплять аэродинамические части. Езда на велосипеде толкала часть крыла в воздух и позволяла измерить его показатели, давая тот же результат, что и планер, при гораздо меньших затратах. Они не просто изобрели самолет; они изобретали все инструменты, необходимые для изобретения самолета.

Братья столкнулись со множеством неудачных попыток. Планеры не могли подняться, не могли повернуть, повернули не в ту сторону или упали на землю. Значения коэффициента подъемной силы и результаты испытаний формы крыла, полученные предыдущими исследователями, оказались неверными и требовали пересмотра. Конструкции винтов, которые они надеялись позаимствовать из судостроения, не работали, поэтому пришлось изобретать свои собственные. Ни у одного производителя двигателей не было достаточно легкого источника энергии, поэтому они

построили его в своей мастерской. Компоненты часто ломались, откладывая испытания и требуя дорогостоящих замен. Несмотря на все это, у них не было никакой гарантии, что самолет когда-нибудь будет работать. Но они твердо стояли на своем.

Не было такого дня, когда самолет был бы готов. В течение многих итераций его производительность постепенно улучшалась. В 1899 году они запустили воздушного змея шириной полтора метра. В 1900 году они запустили достаточно большой планер, который мог поднять человека. Планер 1902 года имел новые рули и систему управления, позволяющую пилоту совершать управляемые повороты влево и вправо. В 1903 году они добавили двигатель и полетели с подачей энергии и возможностью управления, но всего на несколько секунд в нескольких метрах над землей. Райты продолжали работать, улучшая слабые стороны проекта, учась на ошибках и достигая больших успехов. К 1905 году Уилбур совершал 40-минутные полеты на расстояние 32 км. Они работали в течение шести лет.

Задача геймдизайнера имеет много общего с задачей Райтов. Как и самолет, игра — это система. Она служит другой цели, но процесс изобретения принципиально тот же. Геймдизайнеры — это изобретатели машин для генерирования опыта.

Работа Райтов похожа на итерацию. Но очевидно, что итеративный цикл из трех частей является преступным упрощением той работы, которую они проделали.

Райты тестировали, но не по фиксированному графику. Их процесс был органичным, включал множество различных взаимодействий. Их подход менялся каждый день в соответствии с их потребностями. То же относится и к геймдизайнерам. Иногда мы должны планировать. Иногда должны размышлять, рассчитывать, рисовать или изобретать новый метод создания знаний. Изобретение — это не повторяющийся конвейерный процесс. Это постоянно меняющаяся интеллектуальная граница.

Так что схема итерации, которую я показывал ранее, неверна. Настоящий геймдизайн более органичен.



И этот процесс усложняется в тысячу раз, когда команда становится гораздо больше, чем всего несколько людей. Создание знаний — очень изменчивый и тщательный процесс. Он бурлит в каждом разработчике, ежесекундно, ежедневно. Каждая мысль дизайнера создает знание. Каждый раз, когда художник откидывается назад и смотрит на картину, чтобы увидеть ее под другим углом, он создает знания. Каждый раз, когда программист запускает игру на пять секунд, чтобы проверить виджет интерфейса, он создает знания. Итерационный цикл не может управлять этим, и никто не может сделать так, чтобы процесс выполнялся по заданному расписанию. Это невозможно.

Вот почему как дизайнеры мы должны помнить, что можем управлять только основными процессами, которые мы запускаем. Реальность всегда более органична и сложна, чем наше представление о ней.

Зависимости

После создания заповедника недавно обученные смотрители парка сразу решили внести некоторые изменения. Они думали, что лосей было недостаточно, поэтому запустили программу по их вскармливанию.

Популяция лосей значительно выросла. Вскоре огромные стада лосей начали съедать осины и ивы. Это привело к исчезновению бобров, так как у них не осталось достаточно древесины для плотин. Без плотин парк начал пересыхать каждое лето. Для рыбы не осталось водоемов, и вскоре озера опустели. С исчезновением рыбы упала популяция гризли, так как они выживали благодаря пойманной из рек рыбе. С исчезновением гризли и появлением такого количества лосей, на которых можно охотиться, выросла популяция волков. Рост популяции оленей вскоре прекратился из-за волков и выпаса лосей.

И изменения продолжались, распространяясь на всю экосистему...

Геймдизайн может иметь сотни механик, вымышленных элементов и подсистем. Даже в течение нескольких минут после создания идеи игры у дизайнера может быть 20 различных идей для задач, систем и интерфейсов, которые можно добавить. Имея такое количество идей, как мы узнаем, над чем работать в первую очередь? Начнем ли мы с самого уникального? Самого основного? Самого легкого? Самого технологичного? Самого рискованного?

Ключ к ответу на этот вопрос заключается в понимании зависимостей.

ЗАВИСИМОСТЬ — это такое отношение между двумя частями проекта, когда изменения в одной части вызывают изменения в другой.

Представьте, что кто-то просит вас покрасить 10 домов. В этой задаче нет зависимости. Неважно, в каком порядке вы будете их красить, потому что это не влияет на то, как вы будете красить другие дома.

В геймдизайне все не так. Различные части дизайна часто взаимозависимы. Внешний вид уровня зависит от оформления этого уровня. Оформление зависит от инструментов

игрока. Инструменты игрока зависят от базового интерфейса. Если какой-либо элемент изменяется, то изменяется и каждый зависимый от него элемент.

Понимание зависимостей помогает нам снизить риск того, что придется переделывать законченную работу из-за изменений в каком-то месте, от которого она зависела. Например, представьте, что мы потратили время на анимацию персонажа, который бежит со скоростью 5 километров в час во всех направлениях. Если позже мы решим, что он должен двигаться со скоростью 7 километров в час, всю эту анимацию придется переделывать.

Анимация зависит от дизайна системы движений персонажа; изменение в системе движений повлияет на анимацию и испортит хорошую работу. Если бы мы лучше понимали зависимости, мы могли бы сначала укрепить механику движения (в «сером ящике»), а потом делать анимацию.

Стек зависимостей

Чтобы понять зависимости в дизайне, дизайнеры могут построить стек зависимостей.

Чтобы создать стек зависимостей, мы начинаем с геймдизайна. Проект может быть представлен в виде плана на бумаге в начале разработки или он может быть частично реализован и протестирован. Мы разбиваем игру на отдельные элементы — механику, элементы управления, интерфейсы и подсистемы. Затем определяем ключевые зависимости между этими элементами. Наконец, мы рисуем график, иллюстрирующий все взаимозависимости. Это и есть стек зависимостей.

Давайте разберем пример. Представьте, что мы делаем Fantasy Castle, легкую игру о строительстве замка в фэнтези-мире. Разработка Fantasy Castle только началась, поэтому команда дизайнеров переполнена идеями, но им не хватает протестированных и проверенных проектных решений. Они написали длинный дизайн-документ. Каждая из 22 подсистем подробно описана на бумаге. Далее следует их краткое изложение в произвольном порядке:

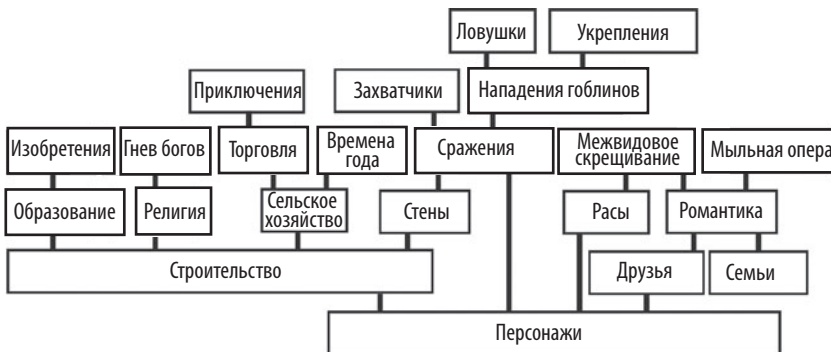
СТЕК ЗАВИСИМОСТЕЙ — это простой метод анализа, который определяет ключевые зависимости между элементами проекта. Так мы можем понять, над чем нужно работать сейчас, а что можно оставить на потом.

- **Персонажи.** Могут существовать и перемещаться в окружающей среде.
- **Семьи.** Персонажи могут вступать в семейные отношения.
- **Расы.** Замок может быть заполнен большой группой людей, эльфов, гномов и других фэнтезийных рас.
- **Межвидовое скрещивание.** Различные виды могут скрещиваться, чтобы создать гибриды с общими характеристиками своих родителей.
- **Нападения гоблинов.** Гоблины могут устраивать периодические облавы, чтобы проверять оборону замка.
- **Сельское хозяйство.** Сельское хозяйство и продовольственная система дают населению пропитание.
- **Торговля.** Персонажи могут торговать с соседними замками особыми или редкими товарами.
- **Образование.** Персонажи могут получить образование.
- **Изобретения.** Образованные персонажи могут изобретать новые машины для замка.
- **Религия.** Персонажи могут строить храмы и проводить там религиозные обряды, развивать отношения с конкретными божествами и получать дары от них.
- **Гнев богов.** Другие божества разозлятся на вас, если вы будете неуважительно относиться к ним или поклоняться их врагам.
- **Друзья.** Персонажи могут иметь платонические отношения.
- **Романтика.** Персонажи могут иметь романтические отношения.
- **Строительство.** Персонажи могут что-то строить.
- **Стены.** Персонажи могут строить стены, чтобы остановить врагов.
- **Укрепления.** Стены можно укреплять и утолщать, чтобы сдерживать гоблинов.

- **Ловушки.** Персонажи могут устанавливать ловушки или автоматические защиты.
- **Сражения.** Персонажи могут бороться, чтобы защитить замок.
- **Захватчики.** Можно организовывать группы захвата, чтобы исследовать близлежащие подземелья и получать добычу.
- **Приключения.** Можно обслуживать путешествующих искателей приключений, обеспечивая трактиры и лавочников в обмен на трофеи из древних склепов.
- **Времена года.** Полный сезонный цикл влияет на сельское хозяйство, строительство и другие виды деятельности.
- **Мыльная опера.** Среди жителей замка разыгрываются аферы и прочие романтические драмы.

Игра может включать в себя все эти элементы, но как выбрать, на чем сосредоточиться в первую очередь? Начинать ли с основ мира и создавать времена года? Начинать с персонажей, друзей и семьи? Возможно, следует начать с войны, выстроив боевую систему для сражений между жителями замка и гоблинами? Начать с уникальных элементов игры, таких как искатели приключений, или построить базу, используя элементы, которые были сделаны ранее в других играх, такие как стены и бои? Стек зависимостей помогает нам принять решение.

Ниже представлен мой стек зависимостей для игры Fantasy Castle:



Нет смысла в ловушках и оборонительных системах, если нет набегов гоблинов. Набеги гоблинов не нужны, если нет боя. Бой не имеет особого значения при отсутствии стен и не может работать без персонажей. Для стен нужны системы строительства, которые требуют, чтобы персонажи работали. Каждый элемент зависит от нижестоящих элементов.

Прежде чем мы продолжим, позвольте мне прояснить понятие зависимости.

Помните, что блоки стека — это не просто слова, которые вы видите, — каждое слово представляет собой подробный дизайн

нескольких страниц. Части этих дизайнов взаимосвязаны.

Например, дизайн строительства описывает каждую кнопку, выделение и опцию интерфейса, который игроки используют для конструирования. Аналогичным образом дизайн фермерства подробно описывает размещение, управление и удаление ферм. При этом дизайн фермерства предполагает, что дизайн строительства будет правильным. Что произойдет, если изменятся детали в интерфейсе строительства, если кнопки поменяются местами или будут заменены на управление мышью? Строительство все еще остается в игре, но оно изменилось, и теперь все детали фермерства необходимо изменить соответствующим образом. Вот почему фермерство зависит от строительства.

ЗАВИСИМОСТЬ

не означает, что основополагающий элемент не должен влиять на зависимый элемент. Она предполагает, что изменения в дизайне базового элемента приведут к изменениям в зависимом элементе.

Существует гораздо больше зависимостей подобного типа, чем записано в стеке. Например, я разместил стек «стены» над стеком «строительство». Но возможно, что в процессе разработки дизайнеры могут обнаружить, что им нужно изменить интерфейс строительства, чтобы упростить строительство стен. Таким образом, стек «стены» зависит от стека «строительство», а стек «строительство» зависит от стека «стены» — циклическая зависимость. Эти нити зависимостей пронизывают Fantasy Castle насквозь. Возможно, придется изменить стек «стен», чтобы их можно было легче размещать вокруг ферм. На систему изобретений может повлиять стек «дружеские взаимоотношения», если друзья могут настраивать друг друга на создание изобретений.

В каком-то смысле любой элемент может повлиять на все остальное, поэтому все взаимозависимо.

Но некоторые из этих зависимостей сильнее, чем другие. Значительные изменения в стеке «стены» могут в некоторых случаях повлиять на стек «строительство». Но изменения в стеке «строительство» почти наверняка затронут стек «стены». Стек зависимостей намеренно игнорирует самые слабые зависимости, чтобы мы могли сосредоточиться на наиболее важных и потенциально опасных из них. Нахождение этого фокуса является целью стека.

Стек зависимостей является редуктивным. Он не включает в себя важные части реальности разработки. Но если вы просто человек, столкнувшийся с такой сложной проблемой, как сотня взаимозависимых элементов дизайна, включающих сто тысяч взаимосвязей, рациональная редукция — единственный способ добиться прогресса. Мы должны игнорировать некоторые зависимости, иначе погрузимся в аналитический паралич. Стек зависимостей — это не теоретический вопрос, это инструмент для принятия решений. И эти решения лучше принимать, концентрируясь на наиболее значимых взаимозависимостях. Существует несколько способов построения стека зависимостей с учетом деталей дизайна. Можно представить себе игру по строительству замков, в которой существует строительство, но нет персонажей. Или такую, в которой есть нападение гоблинов, но нет стен. Я создал этот стек на основании придуманных мной деталей для *Fantasy Castle*, но мне не хватит этой книги для того, чтобы я мог их описать. Если бы дизайн-документы были написаны по-разному, стеки бы тоже были разными, даже если бы названия были одинаковыми.

Каскадная неопределенность

Мы уже видели, что проекты могут не работать так, как мы ожидаем. Дизайнеры могут написать документ о том, как будут работать нападения гоблинов, но они не могут быть уверены, что это будет работать именно так, как предполагалось. Дизайнеры смогут убедиться, будет ли это работать, только когда проведут компилирование и плейтест.

Неопределенность усиливается в результате зависимостей.	Именно из-за этой неопределенности мы должны обращать внимание на зависимости. Если бы мы не принимали во внимание любую неопределенность, было бы неважно, в каком порядке выполнять работу. Мы бы генерировали идеи, записывали их, выстраивали их в любом порядке, и в последний день разработки дизайн бы идеально складывался, как мозаика. И в случаях производных дизайнов, основанных на проверенных идеях, это может сработать практически полностью, потому что каждый элемент дизайна определен.
---	---

Но в играх с некоторой новизной написанный дизайн часто не соответствует действительности. Существует вероятность того, что в процессе разработки элемент дизайна необходимо будет изменить. И именно эта неопределенность делает зависимости важными.

Например, система набегов гоблинов описана на двух страницах где-то в документации проекта. В ней указано, как и когда появляются гоблины, какую тактику они используют, какие у них есть возможности и стратегии для победы.

Как и у каждого плана, у этого дизайна некоторый уровень неопределенности. Он отражает вероятность, с которой дизайн будет работать не так, как ожидалось. Предположим, что это шаблонный дизайн, поэтому определенность составляет 80%. По оценкам разработчиков, в этой ситуации восемь из десяти раз система будет работать так, как написано, без существенных изменений. Это весьма неплохо.

Но значит ли это, что с самого начала разработки мы можем быть на 80% уверены, что нападения гоблинов окажутся в игре и будут работать так же, как написано?

К сожалению, нет; эта цифра в 80% охватывает только неопределенность дизайна, касающегося нападения гоблинов. Но нападения гоблинов могут пострадать не только в результате изменений, вызванных несовершенством дизайна. Они также уязвимы к изменениям, вызванным несовершенством дизайна, от которых зависят.

Чтобы стек «набеги гоблинов» работал, как и было запланировано, мы сначала должны реализовать персонажей, строительство, строительство стен и боевые системы. Если в какой-либо из этих систем произойдет значительный сдвиг, изменения будут проходить каскадом через стек зависимостей и приводить к изменениям в стеке «набеги гоблинов». Даже если каждый из этих основополагающих элементов имеет чрезвычайно благоприятную определенность в 80%, то вероятность того, что стек «набеги гоблинов» сработает так, как и ожидалось, умножится на 0,8 в пятой степени, или на 0,33 (33%), поскольку ошибка в любом из пяти основополагающих элементов приведет к изменению в стеке «набеги гоблинов».

В большинстве дизайнов нет даже 80% определенности. При разработке рискованных, теоретически прорывных игр большинство проектов терпят неудачу. Определенность в системе часто составляет менее 30%. При таких условиях дизайн пяти слоев вверх по стеку зависимости останется неизменным в 0,2 % случаев. Так что, в принципе, никогда.

Это значит, что большая часть написанного дизайна для Fantasy Castle — ерунда. Фундаментальные системы почти наверняка изменятся при реализации или тестировании, и эти изменения будут проходить каскадом через дизайн, приводя к изменениям везде. Концепции могут остаться, но все особенности будут меняться снова и снова. К концу разработки большая часть верхней части стека будет в несколько раз урезана или изменена.

Казалось бы, это простая истина, но в ней скрыт глубокий смысл. Каждый разработчик видел, как игры меняются в процессе разработки, особенно если они нестандартные.

Но часто трудно четко сформулировать, почему это происходит. Простой неопределенности в отношении отдельных частей дизайна недостаточно, чтобы объяснить это. Настоящая проблема заключается в том, что каждое изменение создает ударную волну последующих изменений, которые пронизывают весь дизайн через зависимости. Это настоящий виновник массового хаоса многих дизайнов. Это ключевая причина, по которой нужны итерации.

Каскадная неопределенность означает, что верхние элементы стека зависимостей почти всегда нуждаются в серьезном изменении дизайна.

Но подойдет не любая итерация. Мы должны выполнять итерацию особым образом, исходя из зависимостей, которые мы определили с помощью стека. Общая стратегия проста.

Мы начинаем с нижней части, с дизайна, который ни от чего не зависит. После нескольких итераций и плейтестов эта основа

Начните с нижней части стека зависимостей и идите вверх через каждый цикл итерации.

становится более определенной. На бумаге определенность могла составлять 40%, но как только мы провели несколько плейтестов, она может достичь 90%. Затем мы можем создать элементы, которые зависят от этой основы, и быть уверенными в том, что они не будут разбиты на части дальнейшими изменениями, идущими снизу вверх. И мы просто продвигаемся вверх по стеку, компилируя снизу вверх. Тем не менее все равно будут появляться неожиданные результаты дизайна и сотрясать всю структуру, но мы уменьшаем их частоту и влияние, выполняя работу в правильном порядке.

Например, в игре Fantasy Castle мы могли бы начать разработку игры только с основными персонажами, конструкцией и стенами. Во-первых, это просто игра о людях, строящих стены. Спустя несколько итераций и хорошего плейтеста мы можем добавлять фермерство. После нескольких итераций мы добавляем торговлю и времена года. Мы продвигаемся вверх, строим башню зависимостей снизу вверх. И дизайн, скорее всего, изменится наполовину — после плейтестинга фермерства мы можем почувствовать, что не нужно добавлять времена года, а вот новый элемент в виде болезней сельскохозяйственных культур привлечет больше интереса. Таким образом, вершина стека меняется по мере укрепления фундамента.

Бэклог дизайна

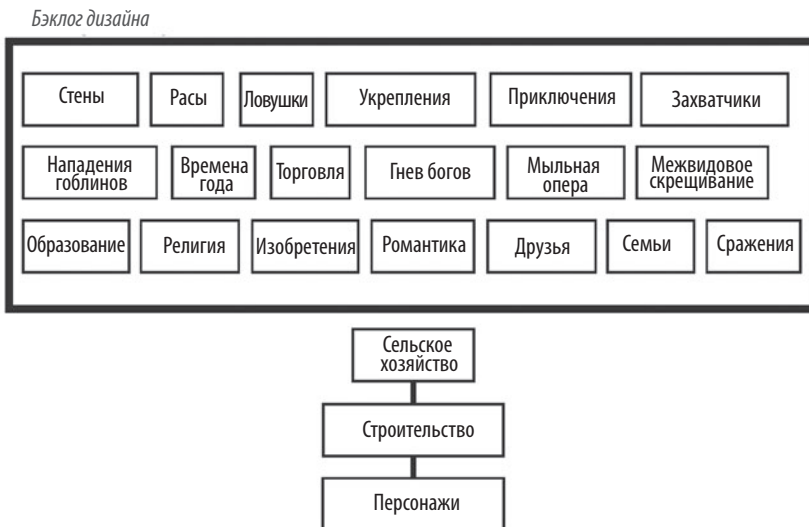
Тот факт, что дизайны в верхней части стека очень неопределенные, не означает, что они бесполезны. У нас все время есть мысли, идеи и наблюдения, и их следует записывать, потому что они могут быть весьма ценными. Но для того, чтобы объединить их во взаимосвязанный подробный проект, требуется проделать большую работу, которая, вероятно, будет аннулирована из-за каскадной неопределенности.

Решение состоит в том, чтобы сохранить идеи в текучей, не заблокированной форме, записав их в бэклог дизайна.

Бэклог дизайна получил название в честь аналогичной концепции из популярной методологии разработки программного обеспечения Скрам — бэклог продукта. Но в отличие от Скрама, он не предназначен для того, чтобы быть частью формализованного процесса разработки. Это неформальный инструмент для сохранения вдохновения.

Только потому, что большая часть дизайна игры Fantasy Castle является ерундой из-за каскадной неопределенности, это не значит, что он бесполезен. Скорее, он требует реорганизации. Большую часть следует рассматривать как не более чем гипотетические идеи на будущее. Эти элементы не нужно объединять в строгий план, поскольку это подразумевает определенность, которой у нас нет. Следует сжимать и помещать в неупорядоченный массив, чтобы в будущем их можно было оттуда извлечь.

Теперь Fantasy Castle выглядит примерно так:



БЭКЛОГ ДИЗАЙНА — это резервуар идей, концепций и впечатлений, над которыми вы не работаете и над которыми не будете работать в ближайшее время. Большинство идей должно идти в бэклог дизайна.

Мой выбор здесь был субъективным. Сначала я решил создавать элементы дизайна для общественных зданий, а все остальное пере-

нес в бэклог дизайна. Другой дизайнер мог бы сосредоточиться на бое или религии. Но независимо от того, что мы выберем, мы должны начать с чего-то и итерировать, прежде чем настраивать фундамент. В противном случае мы строим на фундаменте из песка.

Все, что находится в стеке выше, чем три базовых элемента, будет подвергаться чрезмерной каскадной неопределенности и поэтому не является целесообразным для реализации. На этапе разработки на бумаге, вероятно, определенность упадет ниже 50%. Но по мере реализации, изучения, исследования и тестирования эти три элемента и их определенность будут расти. Например, дизайн на бумаге системы фермерства будет иметь определенность в 60%, а функциональный плейтест — 90% или более. Она может измениться во время итерации, но в этом нет ничего страшного, поскольку от нее еще ничего не зависит.

Только после того, как основа будет устойчивой и определенной, мы можем добавить в стек что-то еще. Теперь пришло время открыть бэклог дизайна, выбрать что-то подходящее и добавить его в верхнюю часть стека. Мы будем хорошо подготовлены к этому выбору, так как будем точно знать, на чем он основан.

Потом нужно просто повторить процесс, выполнив несколько итерационных циклов. Каждый раз, когда дизайн упрочняется, мы производим поиск в бэклоге, вытаскиваем оттуда еще один кусок и помещаем его сверху. Каждый раз, когда у нас появляется новая идея, мы записываем ее в бэклог и откладываем на потом. Большинство бэклогов никогда не будут реализованы. Это нормально — это означает, что те части, которые мы используем, вероятно, хороши. И дизайн растет вверх.

Ядро геймплея

Из 22 элементов дизайна игры Fantasy Castle я решил поместить в бэклог дизайна все, за исключением трех. Но обратите внимание, что мы все еще можем создать эмоционально значимую игру, в которой есть только персонажи, строительство и фермерство. Эти три части сами по себе образуют минимальную, но годную игру, потому что эти три подсистемы составляют ядро игры Fantasy Castle.

Попробуйте следующее. Представьте игру, которую хорошо знаете. Теперь уберите что-нибудь из ее дизайна. Затем уберите следующий элемент или несколько элементов и так далее. Продолжайте убирать до тех пор, пока игра не перестанет генерировать значимый опыт, пока она не станет тривиальной и неинтересной. Явления, которые у вас останутся, и будут ядром игры — минимальным набором механики, которая заставляет игру работать.

Если вы выбрали современную видеоигру, то могли бы отбросить 95% или более игры, включая почти весь контент, большинство интерфейсов и элементов управления. Если вы выбрали классическую или настольную игру, то отбросили гораздо меньше. Трудно понять, как можно уменьшить количество шашек и сохранить их функционал. Но даже шахматы можно разделить на части — убрать все, кроме пешек, и игра все еще останется рабочей.

Некоторые примеры ядра геймплея:

- **Civilization V.** Карта, города, поселенцы, воины.
- **Unreal Tournament.** Карта, игроки с управлением от первого лица, оружие.
- **Starcraft II.** Карта, центр командования, работники, пехотинцы.

В основе каждой из этих сложных игр лежит простой цикл итераций, который сам по себе создает полезный опыт. Даже при наличии только рабочих и морпехов игра StarCraft II генерирует интересные решения и стратегии. Одним из самых популярных способов играть в Unreal Tournament был режим InstaGib, который убирал все, кроме одного очень простого оружия — мгновенного уничтожения. Ядро — это и есть игра, хорошая или плохая. Все остальное — просто вариации и шлифовка.

Во многих случаях ядро игры определяет жанр. Например:

- **Tower defense.** Карта, объект для броска, башни, приближающиеся враги.

ЯДРО ГЕЙМПЛЕЯ — это то, что вытекает из минимальной механики игры, которая лежит в основе ее стека зависимости. Уберите все, что можно убрать, эмоционально не обедняя игру, и то, что останется, и будет ядром игры.

- **Dungeon crawler.** Персонаж, подземелье, герой, монстры, прокачка.
- **Fighting game.** Движение, удар, блок, бросок.

Ядро игры — правильная основа стека зависимостей, потому что все остальное в дизайне зависит от этих базовых механизмов. Идентифицируя ядро, мы находим кратчайший путь к тестируемой платформе для итерации. Чем раньше вы завершите работу над ядром, тем раньше игру можно будет протестировать. Только теперь вы можете увидеть преимущества итерации на основе тестирования. Поэтому, будучи дизайнером, определите ядро и создайте сначала его. Как только оно будет готово, достаньте что-нибудь из бэклог дизайнера, поместите это в дизайн и произведите итерацию.

Если вы не можете найти ядро или работаете над его созданием и у вас получается ужасно, попробуйте начать заново с чем-то другим. Должна появиться очень веская причина, чтобы оставить игру без надежного ядра. И иногда она действительно существует. Например, в приключенческих играх в жанре «указать и щелкнуть» нет реального ядра. Указывание и щелканье не является само по себе рабочей игрой. Эти игры являются исключениями; они работают, потому что их опыт определяется содержанием, а не механикой. В некоторых играх есть несколько возможных ядер. Рассмотрим RPG с открытым миром *Fallout 3*.

Одним ядром *Fallout 3* могут быть персонаж игрока, оружие, монстры. Другим — персонаж игрока, диалоги, квесты. Третьим может быть персонаж игрока, открытый мир, мировое искусство. Эти три ядра делают игру простым шутером, сюжетной игрой с диалогами «ходи и говори» и художественной галереей мирового масштаба соответственно. Но каждое из них все еще представляет собой функциональную игру. Разработчики могли бы начать разработку с любого из ядер, а затем встроить его в другие.

Небольшой стек зависимостей

До сих пор мы рассматривали стек зависимостей как возможность для одновременного анализа всей игры. Но стек зависимостей также можно использовать для анализа дизайна отдельных систем. Если вы попробуете этот метод, скорее всего, будете

пользоваться им большую часть времени, так как постоянно анализировать весь дизайн может быть неудобно и трудозатратно.

Рассмотрим, к примеру, разработку персонажа по имени Кэпп в военной игре. Предполагается, что Кэпп будет быстро двигаться, безболезненно падать и атаковать с помощью акробатических боевых движений капоэйры. Ниже представлены способности и системы Кэппа согласно задумке:

- **Скорость.** Кэпп бежит чрезвычайно быстро, фактически на виражах.
- **Удар ногой с разворота.** Кэпп делает удар ногой с разворота, поражая всех, кто находится рядом.
- **Падение.** У Кэппа есть слабое место. В случае повреждения во время атаки он падает на землю.
- **Прыжок от стены.** Кэпп прыгает и отскакивает от стен, чтобы попасть в специальные зоны или обойти противников.
- **Переворот с опорой на руки.** Кэпп может сделать переворот с опорой на руки, что позволяет ему сделать прыжок от стены.
- **Подсечка.** Чтобы компенсировать свою уязвимость при падении, у Кэппа есть небольшая возможность сделать нижнюю подсечку, благодаря которой можно избежать атак противника и нанести ответный удар за один ход.

Точно так же, как можно сократить большинство хороших игр до ядра и они при этом все еще будут работать, можно сократить и большинство хороших игровых систем и они продолжат выполнять свою роль. В этом нам помогает особая функция стека зависимостей. Вот как я выполнил дизайн Кэппа:



Падение вполне логично, если вас атакуют и вы получаете удар — в данном случае удар ногой с разворота. Подсечка — это способность, предназначенная для возмещения падения, поэтому нет смысла делать ее до самого падения. И наконец, прыжок от стены — это увеличение базовой скорости, а переворот с опорой на руки — особая версия атаки прыжка от стены.

Как всегда, учитывая разные детали дизайна, этот стек может быть разным. Например, скорость и прыжок от стены могли бы легко поменяться местами. Этот стек отражает мой записанный на бумаге дизайн.

В каждой части дизайна существует некоторая неопределенность:

- **Удар ногой с разворота.** В нем существует небольшая неопределенность, поскольку он является производной атак, которые мы видели в других играх. Основная неопределенность заключается в выборе радиуса и времени. Если окажется, что удар ногой с разворотом работает лучше, когда он длится полсекунды, а не три секунды, как предполагали разработчики, изменения могут сместиться вверх к другим способностям.
- **Падение.** Очень неопределенно. Интуиция подсказывает разработчику, что падение может обескуражить, а это означает, что нам придется полностью заменить его.
- **Скорость.** Здесь существуют неопределенности, связанные с технологиями. Нестандартное движение на основе физики всегда сложно реализовать хорошо. У ИИ могут быть проблемы с ожиданием или навигацией, а у игроков могут быть проблемы с его управлением. Может потребоваться изменение дизайна или отказ от приема.
- **Прыжок от стены.** В нем существуют аналогичные риски, что и в скорости, с дополнительными проблемами, с которыми сталкивается ИИ при навигации по путям, которые обычно недоступны. Систему управления прыжков от стены тоже, вероятно, потребуется переделывать несколько раз.
- **Переворот с опорой на руки.** Сам по себе он не так уж не определен, но он уязвим к множеству каскадных неопределенностей систем более низкого уровня.

- **Подсечка.** Сама по себе она простая, но уязвима к каскадным неопределенностям нижних уровней.

Не следует идти напролом, работая над всем дизайном, как если бы была стопроцентная гарантия того, что он будет работать. Скорее всего, не будет. Как и в случае с *Fantasy Castle*, мы должны найти ядро и превратить все остальное в бэклог дизайна.

Ядром этой игры, вероятно, являются удар ногой с разворота и скорость. Благодаря им Кэпп может выполнять свою уникальную роль первоклассного бегуна, что достаточно полезно и выделяет его на фоне других персонажей. Это выглядит следующим образом:



После нескольких итераций ядра оно будет достаточно определенным для дальнейшей компиляции. Только теперь мы можем достать что-нибудь из бэклога, потому что существует большая вероятность, что скорость и удар ногой с разворота изменятся перед тем, как это произойдет, или у нас появится лучшая идея для компиляции.

Зависимости и потребности внешнего дизайна

Работая со стек-зависимостями, необходимо учитывать один факт. Маркетологам и бизнесменам проектные решения могут потребоваться задолго до того, как они потребуются самой игре.

В этих случаях разработчики должны обсудить их с заказчиком и найти золотую середину. Если решение расположено на вершине стека и является очень детальным, выгода, которую эти чрезвычайно ранние решения принесут бизнесу и маркетингу, может не окупить стоимость их принятия на ранней стадии. Например, скорее всего, не стоит сильно рекламировать способность

Кэппа делать подсечки, пока этот ход не пройдет достаточное количество итераций, потому что весьма вероятно, что на подсечку повлияют каскадные изменения дизайна.

Маркетинг и бизнес важны. Но в то же время мы не должны заключать геймдизайн в тюрьму неизменных проектных решений. Будьте изменчивыми, не думайте о будущем и обращайтесь на риск, связанный с зависимостями. Это требует ежедневных усилий, и поначалу не все поймут. Возможно, это расточительно и даже безответственно. Но в конце концов, только так маленький человек может решить такую сложную задачу, как геймдизайн.

Управление

Бригадир Джон устал от рабочих, которые все портят. Поэтому он купил Puppetron — последнее слово в области управления. Ему больше не придется отдавать указания и надеяться, что они будут выполнены правильно. Теперь он мог надеть на голову шлем Puppetron и контролировать все действия рабочих силой мысли. Он стал подобен кукловоду, дергающему за тысячу веревочек, лично координируя действия каждого работника. Это было чудесно.

Бригадир Джон и его команда строили мост. Когда уже половина работ была выполнена, мост рухнул и всех убил.

Следователи обнаружили сотни мест с халтурной сваркой, перекрученные кабели, смещенные стойки и отсутствие болтов. Как будто ни одна из работ не была выполнена работником осознанно.

Создание хороших игр возлагает определенные обязательства. Мы должны исследовать культуру и задумку, решать сложные интеллектуальные головоломки и рисковать. Мы должны использовать любые преимущества и ресурсы. Мы должны думать не только о зарплате, но и о том, что мы делаем. Мы должны вкладывать душу в работу.

Но вкладывание души — дело тонкое. Нужно особое сочетание методов работы, культуры, внутренней мотивации и организационной структуры. У людей должны быть необходимые полномочия, опыт и знания должны передаваться другим, и мы должны доверять друг другу. В этой главе я напишу о создании таких условий, в которых команда смогла бы работать в полную силу.

Банальность зла

Особенности организации — это не просто усредненные особенности ее участников. Важно, как они структурированы. Структура определяет, как знания, власть и ресурсы распределяются внутри

команды. Если их распределяют неправильно, группа гениев может вести себя как психи.

В качестве наиболее яркого примера можно привести исторический геноцид XX века. Убийственная бюрократия нацистской Германии, СССР и кхмерской Камбоджи включала в основном совершенно нормальных людей, которые, вероятно, стали бы хорошими исполнителями в другое время и в другом месте. Но в определенной организации с определенной культурой и иерархией они стали винтиками в машине смерти. Политический теоретик Ханна Арендт назвала это «банальностью зла». Она поняла, что ужасы бюрократии совершают не хохочущие сумасшедшие, а легионы канцелярских крыс, покорно следующих своим собственным побуждениям.

Очевидно, что никто не умирает в процессе разработки игр. Я привел эти примеры, чтобы показать, что нет предела тому, насколько плохо структурированная организация может испортить результат, даже несмотря на то, что к исполнению были привлечены хорошие кадры. Наличие человеческих ресурсов необходимо, но они все пойдут впустую в плохой структуре и дисфункциональной культуре. Мы должны правильно понять эту структуру. Таким образом, вопрос заключается в том, как привлечь правильные ресурсы для хорошей работы.

Тейлоризм

Наука, лежащая в основе каждого отдельного действия рабочих, столь обширна по своему объему и содержанию, что рабочий, наилучшим образом приспособленный к фактическому выполнению своей работы, не в состоянии (по причине отсутствия образования или вследствие недостаточности умственных способностей) овладеть этой наукой.

Ф. У. Тейлор

В конце XIX века большинство рабочих определяли методы работы сами. Управляющий поручал каменщику, резчику по металлу или чугунищику задачу, и он выполнял ее в свободное

время, используя методы, которым он научился за годы своего опыта и обучения.

Около 1900 года человек по имени Фредерик Тейлор начал изучать рабочие процессы и увидел огромные пробелы в традиционных методах. Он приступил к созданию стиля управления, который впоследствии будет известен как тейлоризм.

Тейлор наблюдал за работой каменщика и секундомером измерял, сколько времени занимает каждое движение. Он записал время, которое нужно для того, чтобы протянуть руку и взять кирпич, положить его в нужное место, выровнять, положить сверху цемент и убрать излишки. Собрал необходимые цифры, Тейлор начал разрабатывать лучший способ кладки кирпичей. Он скорректировал и убрал лишние движения, чтобы найти их наиболее эффективную последовательность. Он убрал лишние шаги, которые делали многие каменщики на этапе кладки. Он перепроектировал инструменты, создав специальное устройство для удержания кирпичей в правильном положении, чтобы их можно было быстро взять. Затем он проинструктировал рабочих, какие именно движения им следует выполнять, и заставил довести их до автоматизма. Этот процесс «научного менеджмента» стал его арсеналом средств, и он применял его к различным сферам, от резки металла до обработки чугуна.

Тейлор не пытался заставить рабочих понять, что они делают. Для него они были неуклюжими идиотами, и их понимание было ненужным. «Рабочий, наилучшим образом приспособленный для фактического выполнения своей работы, не способен в полной мере уяснить себе этот ее научный фундамент без помощи и руководства со стороны тех, кто работает вместе с ним или над ним. Это объясняется у него либо отсутствием образования, либо недостатком умственных способностей» — писал Тейлор.

Тейлоризм — это сосредоточение каждого решения в руках небольшой группы очень умных людей. Один думающий «мозг» наверху направляет действия многих глупых «рук». Концентрируя знания в едином умном мозге, тейлоризм повышает качество принимаемых решений, поскольку этот мозг является наиболее мотивированным, наиболее квалифицированным и наиболее

способным и может разумно координировать действия глупых людей, перевозящих чугун.

Идеи Тейлора легли в основу современного исследования эффективности, потому что они работают. За прошедшее столетие полученные им методы обеспечили нас более качественными и дешевыми автомобилями, картофельными чипсами и компьютерами. Этот метод концентрации решений в головах немногих умных людей и устранения лишних телодвижений настолько успешен, что теперь он является предполагаемой частью культуры производства.

Но тейлоризм ограничен видами задач, которые может решить. Основная причина — это количество знаний, которое включает в себя работа. Если работа требует мало знаний, тейлоризм работает хорошо. Начальник производства может знать все о том, что делает каждый из его подчиненных, так как работа на производстве простая и цикличная. Он может принять все решения вместо подчиненных, ведь решений нужно принимать не так много. Он может удерживать в голове любые знания.

Но что происходит, если работа сложная и не повторяющаяся? Что будет, если информации больше, чем может выдержать один мозг? Теперь «главный мозг» перегружен данными. Он начинает игнорировать детали, упрощать или пропускать важные сигналы. Решения становятся хуже, и возникают лишние действия.

Разработка игр — одна из тех задач, в которых тейлоризм терпит неудачу, потому что разработка игр включает в себя действительно огромное количество знаний. Представьте, что в процессе коллективной разработки вы пытаетесь каталогизировать все знания. Каждый раз, когда художник наносит мазок кисти, оценивает и нажимает кнопку «Отменить», он создает крошечный кусок знаний, который говорит, что эта идея не очень хороша по той или иной причине. Каждые 10 секунд самостоятельного тестирования, которое будет проводить разработчик, каждое изменение алгоритма, придуманное программистом, каждый спонтанный разговор с коллегами или возникающая идея являются потенциально полезными знаниями. Их объем растет каждую минуту, секунда за секундой. Ни один мозг или не-

большая группа «руководящих мозгов» не сможет воспринять и использовать их все.

И огромный объем знаний — не единственная проблема. Большую часть знаний о процессе трудно или невозможно передать. Это называется неявным знанием. Например, навыки — это неявное знание. Опытный художник может посмотреть на никудышную композицию и просто знать, как ее исправить. Программист может просто знать, как оптимизировать алгоритм, а разработчик может просто знать, как сделать интерфейс лучше. Но никто из них не может объяснить это — эти знания возникают интуитивно на основании натренированного бессознательного. Эти знания невозможно передать ведущему специалисту. Это навык, и на его развитие ушли годы.

Так что тейлоризм, являющийся надежным стандартным методом, не подходит для разработки игр. Но существует решение.

Распределенный разум

Чтобы понять, как создавать игры, не обязательно учиться у мастеров, живших в 1912 году. Можно поучиться у муравьев.

Вспомните, как муравьи собирают еду. Во-первых, несколько муравьев-фуражиров отползают от муравейника. Когда муравей-фуражир находит источник пищи, он возвращается, оставляя за собой след феромона. Другие муравьи инстинктивно следуют по этому следу к еде. Каждый муравей, который находит еду, тянет уже свой собственный след феромона назад, тем самым усиливая его. Если муравей приходит к богатому источнику с едой, след феромона усиливается каждый раз. По мере упрочнения тропки начинаются другие процессы. Появляются более сильные рабочие муравьи и убирают препятствия с тропки. Муравьи-солдаты начинают ее патрулировать, наблюдая за угрозами. Некоторые рабочие муравьи жертвуют собой, создавая из своего тела мост через ямки. Вместе муравьи чрезвычайно эффективно создают, оптимизируют и защищают тропки к лучшим источникам пищи. И весь этот великолепный сложный процесс осуществляется без центрального плана благодаря совместным усилиям глупых муравьев, которые следуют своим простым правилам.

Ни один из муравьев не понимает общую стратегию того, что делает колония, но все они в равной степени координируют свои действия в комплексный подход. Как будто колония муравьев образует коллективный, распределенный разум, обладающий гораздо большей силой, чем любой из ее отдельных участников. Муравей — глуп, колония муравьев — умна.

Разработка игр работает по такому же принципу, так как никто не может понять все, что происходит в процессе разработки. Слишком много событий для одного человеческого мозга. Поэтому, подобно муравьям, каждый из нас должен играть свою роль, но только в более широком распределенном разуме. Действия каждого из нас должны выполняться на благо чего-то большего.

Это невозможно с помощью тейлоризма. Мы не можем отнимать решения у рабочих и отдавать их в руки нескольких руководителей. Чтобы распределенный интеллект работал, мы должны разделить власть между членами команды.

Распределение власти

У муравьев нет начальников, которые говорят им, как сделать феромоновую дорожку. Каждый из них принимает собственные решения в зависимости от условий. В этом есть два основных преимущества.

Во-первых, задействуется весь мозг каждого муравья. В тейлоризме интеллектуальные силы рабочих остаются невовлеченными, потому что у них отнимается каждое решение. Не использовать такой ценный ресурс — это ошибка, особенно в геймдизайне. Весь смысл работы с командой разработчиков — использовать их мозг.

Во-вторых, при полном распределении власти используются знания, которыми обладает каждый муравей. Каждый муравей очень хорошо знает свое непосредственное окружение, так как он в нем находится постоянно. Если бы королеве постоянно приходилось говорить каждому работнику, что ему нужно сделать, она принимала бы плохие решения, находясь только в одном месте. Она никогда не поймет то, что делает работник № 1314

так же хорошо, как сам работник № 1314. То же самое относится и к геймдизайну. Каждый разработчик понимает свою часть работы так, как никто другой. Каждый разработчик понимает уровень, технологии или механику, над которыми он работает, так, как никто другой.

Разработчики игр могут делать то же самое, что и муравьи, распределяя решения между членами команды. Каждый разработчик выбирает то, что ему ближе всего. Программист системы должен принять решения относительно дизайна этой системы. Дизайнер уровня должен разработать подробный проект уровня. У каждого человека существует сфера естественной власти, которая охватывает те части проекта, которые он понимает лучше всего.

Это не значит, что каждый просто работает только над своей задачей. Разработка игр всегда предполагает интенсивную коммуникацию, поскольку решения часто требуют знаний разных людей. Например, художник никак не сможет решить, как будет выглядеть персонаж, не зная нарративной цели. Писатель не сможет написать хороший диалог для боевой сцены, если он не знает, как она изменилась после последнего плейтеста. Чтобы собрать всю эту информацию, нужна коммуникация.

ЕСТЕСТВЕННАЯ ВЛАСТЬ разработчика распространяется на любое решение, в котором он разбирается лучше, чем кто-либо другой из членов команды.

Например, совещания — это способ сбора информации для принятия решений. Рассмотрим совещание, на котором креативный директор, программист, разработчик и художник решают, с каким из двух возможных вариантов «серого ящика» продолжать работу. Они организовали совещание, так как у каждого из них есть уникальные знания, имеющие отношение к решению.

Разработчик провел пять плейтестов на каждом «сером ящике», поэтому он знает, где у уровня проблемы с балансом и понятностью. Он также работал какое-то время над уровнем, поэтому у него множество идей и неудачных экспериментов.

Программист знает, какой размер и сложность уровня будут работать в игровом движке, а также прочие технические ограничения системы.

Художник лучше всего понимает, как уровень вписывается между предыдущим и следующим уровнями с точки зрения художественного развития игры, а также знает, какие еще художественные ресурсы доступны.

Креативный директор знает общую структуру игры и то, куда будет добавлен уровень, а также какие у него эмоциональные цели, стратегия позиционирования на рынке, более глубокие темы и потребности инвесторов.

Никто из этих людей не может принять решение в одиночку. Они встречаются, чтобы поделиться своими знаниями и принять совместное решение. Это совещание хорошо организовано, на нем присутствуют только люди с необходимыми знаниями и никого лишнего. При отсутствии хотя бы одного из них знаний для принятия решения было бы недостаточно. Приглашать кого-то еще было бы бессмысленно.

Плохие решения принимаются в тех случаях, когда знания не применяются. Например, дизайнер уровней может решить протестировать уровень методом «серого ящика», размер которого слишком большой для компьютера. Программист мог бы его предупредить об этом. Но так как между ними не было никакой коммуникации, его решение придется переделать. Мы инстинктивно обвиним дизайнера уровней, но это неправильно. В данном случае следует винить организационную структуру за то, что дизайнер не получил необходимой информации.

Даже при отличной коммуникации значительная часть соответствующих знаний все еще остается скрытой или слишком объемной, чтобы ее можно было передать. Вот почему лучше, если человек с естественной властью принимает окончательное решение — дизайнер уровней принимает решение о своем уровне, писатель — по диалогу, директор — по общей структуре.

Присвоение и доверие

Слово «arrogant» происходит от французского *s'arroger* и означает «присваивать чужие полномочия». Люди присваивают полномочия по принятию решений, когда отнимают их у человека, над которым у них естественная власть.

Присвоение часто принимает форму *микроменеджмента*. Микроменеджмент — стиль управления, при котором руководитель отдает распоряжения, связанные с незначительной специальной информацией, которую подчиненный понимает лучше, чем руководитель. Обычно это приводит к плохим решениям, потому что специальные знания подчиненного о работе не принимаются во внимание.

Например, руководитель может выслушать часовой обзор разрабатываемой игровой системы и потребовать внести ряд изменений, которые представляются сотрудникам «на передовой» крайне глупыми. Таких руководителей ласково называют «чайками»¹. Но руководители делают так не по собственной глупости. Они не потратили сто часов на итерации и тестирования, как сотрудники «на передовой», и не знают нюансов каждого теста, эксперимента или обсуждения. Руководитель может быть очень опытным, но общего опыта редко бывает достаточно, чтобы превысить уровень знаний людей, которые провели за этой работой гораздо больше времени.

Это распространенная проблема, которая возникает из глубинных когнитивных искажений. Геймдизайн полон скрытых возможностей. Один уровень или система могут быть сыграны сотнями или тысячами разных способов, и чтобы понять игру, недостаточно посмотреть ее обзор. Однако кажется, что все именно так и происходит из-за глубоко укоренившегося предположения WYSIATI (что видишь, то и есть). Руководитель не знает, сколько он не знает, так как никогда не видел того, чего не знает. Поскольку он сам не осознает, что информации не хватает, то считает, что видит полную картину. Таким образом, он присваивает себе полномочия по принятию решений, считая, что может сделать лучше, а подчиненные возмущаются и уходят.

Основная причина присвоения заключается в том, что лидеры, зная, что у них больше опыта, не доверяют своим подчиненным. Поэтому они и не хотят давать другим возможность принять важные решения и берут это на себя. Но такой подход не ра-

ПРИСВОЕНИЕ — это предъявление прав на принятие решения, которое подпадает под естественную власть другого человека.

¹ Чайка-менеджер описывается простой формулой: «прилететь, поорать, нагадить и улететь». — *Примеч. ред.*

ботаает — руководитель не может знать о процессе все. В итоге решение становится хуже.

Чтобы вместе работать над игрой, мы должны доверять друг другу. Это не призыв к командной работе или мотивационный лозунг. Это голые факты, и слово «должны» я использую в буквальном смысле. Доверие — это не опция. Мы не можем исправлять чужие ошибки просто потому, что не в состоянии понять работу друг друга. Поэтому лидер, руководящий командой болванов, обречен. Он не может защититься от идиотизма, когда во всей студии принимается слишком много решений, находящихся вне его понимания и юрисдикции. Любые усилия, которые он предпринимает, чтобы защититься от людей, которых он считает тупее, чем они есть, только помешают команде. Единственный вариант — привлечь людей, которым вы можете доверять, и начать доверять.

Задать направление

Мы хотим распределить полномочия на принятие решений, но это не значит, что можно просто закрыть команду в комнате с пиццей и компьютерами и вернуться через два года за готовой игрой. Даже при распределении власти лидеры играют свою роль.

Руководители знают макроструктуру игры. Они знают ее эмоциональные цели, рыночную стратегию, бизнес-стратегию, нарратив, настроение, стиль дизайнера, фокусы дизайнера и общую структуру механики лучше, чем кто-либо. В этих сферах у руководителей естественная власть, и они принимают решения.

Чтобы эти решения были результативными, у лидеров должна быть возможность управлять подчиненными. Они должны направлять их к поставленным целям, а не указывать, что кон-

кретно нужно делать. Они не должны требовать поменять цвет сапог персонажа. Они должны объяснить причину, по которой сапоги должны выглядеть иначе, и дать художнику возможность решить, как это сделать.

Направление — это цель работы. Какие задачи решает внешность этого персонажа? Какова цель этого уровня? Какова роль этой технологии в общем дизайне? Что понимает руководитель, но не понимает подчиненный?

Лидеры не могут указывать подчиненным, как выполнять любую мелкую задачу. Необходимо задать масштабное НАПРАВЛЕНИЕ.

Направление — это концепция, которую я заимствую из военного управления. Капитан не говорит сержанту, куда именно отправить каждого солдата во время штурма холма. Он думает только о том, какой именно холм будет штурмовать и в какое время. Сержант не думает о других холмах, кроме того, куда его отправили. Он думает о том, как направить своих солдат на этот холм, на который ему указали. А каждый солдат самостоятельно решает, куда целиться и как бежать, чтобы добраться туда, куда его отправил сержант.

Задавая направление, руководитель использует свои уникальные знания о более широких структурах проекта, чтобы обеспечить подчиненного информацией, которой тот не располагает. В то же время, придерживаясь общих соображений и не присваивая себе право принимать мелкие решения, руководитель не отказывается от информации, предоставляемой подчиненным в подробнейших деталях.

Руководитель не может быть готовым ко всем неизбежным проблемам. У него нет суперспособностей разобраться в деталях всего проекта. Но до тех пор, пока подчиненный понимает направление, он может решать задачи и использовать возможности так, чтобы наилучшим образом достичь поставленной цели. Сотрудник может сделать что-то совершенно не так, как предполагал руководитель, но пока он не сходит с заданного ему направления, игра становится только лучше.

Например, руководя уровнем в игре ужасов, высокомерный лидер и приверженец тейлоризма может сказать: «Игрок просыпается в ванне кровавой воды. На зеркале нацарапано: “Ты не сбежишь”. Игрок осматривает дом и находит тело своей подруги. Вскоре после этого он встречается врага с мачете, и поскольку у игрока нет оружия, ему придется бежать из дома, а противник будет его преследовать». Вот список задач, которые необходимо выполнить. И если бы все работали так, как запланировано, все было бы гораздо проще. Но что происходит, если тестировщики отказываются бежать от мачете или смеются над дурацким посланием на зеркале? Сотрудники не могут решить эти проблемы самостоятельно. Они не понимают, что делают, и у них нет полномочий что-либо менять. Следовательно, они должны вернуться к руководителю, а тот, вероятно, предложит им плохое

решение, потому что ему не хватает глубоких знаний об уровне и он не присутствовал на провалившемся плейтесте. Специальные знания сотрудников не учитываются, и процесс срывается.

Лидер должен задать направление: «Этот уровень должен научить игрока всем основным элементам управления, кроме управления оружием, которое следует оставить до следующего уровня. Он должен включать историю о мертвой девушке, чтобы мы могли развернуть ее позже. Следует также добавить героя с мачете. Поскольку следующий уровень начинается за пределами заброшенного дома, этот должен закончиться тем, что игрок выходит через входную дверь дома. Это первый уровень, так что игра может начинаться с любой локации. Наконец, не показывайте главного героя, я хочу, чтобы на этом этапе он оставался загадкой». Теперь команда может приниматься за работу. Они могут начать с того же дизайна, который первоначально предлагал менеджер проекта. Но во время итерации они, скорее всего, найдут лучший способ достичь тех же целей. Плейтесты и мозговой штурм раскроют новые уровни проблем и решений, которые лидер никогда не смог бы найти в одиночку. Освободившись от строго регламентированного списка задач, разработчики могут использовать возможности и решать проблемы по мере их возникновения. Дизайн изменится, но поскольку разработчики понимают его роль в более крупной структуре, то смогут убедиться, что он хорошо и правильно вписывается в следующий уровень. А руководитель освобождается от обязанности следить за деталями.

Чтобы задать направление, руководитель должен сначала сам его понять. Иными словами, он должен понимать структуру игры и цели работы каждого разработчика. Поскольку это предполагает сложные мыслительные процессы, плохие руководители часто включают микроменеджмент. Они задают неопределенные направления, а затем критикуют то, с чем приходят подчиненные. Такая «оценка сверху» приносит мало пользы. Руководитель — единственный, кто способен усовершенствовать широкую структуру игры.

Он должен думать об этом, анализировать ежедневно. Если он хорошо справляется с этой работой, у него не останется времени придирается к мелочам в работе подчиненных. Если

руководителю нужно оценивать работу для поддержания стандартов качества, его замечания должны ограничиться общими утверждениями о недостатках и предложениями помощи, а не конкретными указаниями, что должно быть сделано.

Как только направления понятны, о них нужно четко сообщить. К сожалению, в разработке игр слишком часто звучит фраза: «Это не то, что я хотел». Результат, который вы видите, означает, что сотрудник не понял направления и поэтому не смог выполнить задачу. Задать направление без микроменеджмента — это навык, который требует внимания и практики. Требуется усилие, чтобы понять и выразить цель, а не просто указать, что конкретно нужно делать.

Руководители тоже итерируют. Они итерируют не отдельные уровни или механику, а продвижение уровня, структуры сюжета или позиционирование на рынке. Как и всем остальным, им нужно знать результаты своих решений. Им нужен не подробный посекундный темп или отдельные решения геймплея, а сжатая информация более высокого уровня, которая должна отразиться на более широкой структуре игры. Если одна основная игровая система просто не работает, лидер должен об этом знать. Если тестировщики привязываются к второстепенному персонажу на одном уровне, лидер должен об этом знать, так как подобные результаты могут приводить к изменениям в широкой структуре игры.

Сотрудники должны РЕЗЮМИРОВАТЬ новую полученную информацию и предоставлять лидеру соответствующие выводы.

В команде, где руководитель задает понятные, четко выраженные направления, а подчиненные предоставляют ему на рассмотрение соответствующие выводы и итоги о проделанной работе, каждый знает ровно то, что ему нужно знать, и не более, а решения принимают люди, обладающие наиболее актуальными знаниями и, следовательно, естественной властью. Руководителю не обязательно везде присутствовать и решать все проблемы, но процесс разработки должен быть целенаправленным и структурированным. Каждый итерирует свою работу — руководитель делает широкие мазки, а рабочие добавляют к ним детали. Благодаря тому что все знания применяются эффективно, вам не нужны гениальные разработчики для того, чтобы создать выдающуюся игру.

Мотивация

Здесь суета! Здесь праздник жизни!

Но он не найден влюбленным народом.

Он на вершине, что видят реже

Пустынных комнат. Последний взмах кисти...

Мотивация геймдизайнера должна быть и сильной, и тщательно контролируемой. Сильной, потому что нужен мощный драйв, чтобы решать проблемы в процессе разработки игр. Контролируемой, потому что легко можно пойти в неправильном направлении и случайно сделать что-то, что повредит игре. Эта глава посвящена мотивации — ее возникновению, возвращению и управлению в себе и в других.

Внешние поощрения

Казалось бы, чтобы люди работали лучше, нужно просто щедро их вознаграждать. Трудолюбивые разработчики должны получать большую зарплату, возможность покупать акции компании, иметь парковочное место, медицинскую страховку, большой офис или множество других плюшек. Это похоже на то, как владелец завода платит рабочим один доллар за каждую тонну чугуна, которую они загрузят в вагон. Такие стимулы называются внешними вознаграждениями.

ВНЕШНЕЕ ПООЩРЕ-
НИЕ — это награда,
которая отделена от
самой работы, обычно
в обмен на некоторые
измеримые результа-
ты работы.

Внешние поощрения широко распространены в различных сферах: от финансов до правительства и промышленности. Но в геймдизайне они провальны. На это есть четыре основные причины.

Первая причина. Геймдизайн сложно оценить извне, поэтому внешнее поощрение и не работает в играх. Чтобы

дать кому-то денежную «морковку» за хорошую работу, мы должны знать, когда этот человек поработал хорошо, а когда — плохо. Оценить работу рабочего с чугуном легко. Мы можем увидеть результат, и все, что нужно, — это подсчитать количество слитков в вагоне. Но в играх очень трудно увидеть качество приложенных усилий. Разные разработчики вносят разный вклад, а разработка игр настолько неопределенна, что даже хорошая работа может привести к катастрофе в результате неудачного стечения обстоятельств. Хорошие дизайнеры, готовые рисковать, могут оказаться на плохом счету по сравнению с теми, кто работает посредственно и не желает идти на риск, так как некоторые из их попыток неизбежно заканчиваются неудачей. Не существует хорошего способа измерить производительность, следовательно, и не существует способа определить поощрение.

Вторая причина. Поощрения вытесняют нашу внутреннюю любовь к работе. Хорошим разработчикам нужны деньги, но для многих из них это не главная причина работать в этой сфере. Большая часть их мотивации гораздо более субъективна. Они хотят создать что-то великое, явить это миру и, возможно, получить признание. Хотят решать сложные задачи. Хотят уважения, авторства и автономии. В повседневной работе они, возможно, не думают о какой-либо более высокой цели, кроме как не подвести своих коллег. Деньги нужны им только для того, чтобы прокормить семью. Если сделать заработную плату определяющим фактором, то легко разрушить внутреннюю мотивацию работать в свое удовольствие. Люди забывают, что они работают из-за интереса, и начинают верить, что работают ради денег. Бонусы в игре могут вытеснить внутреннее удовольствие от процесса игры. Вот почему дорогостоящие юристы не будут предлагать свои услуги нуждающимся пенсионерам за 30 долларов в час, они предложат их бесплатно, а случайные прохожие бесплатно помогут вам выгрузить диван из грузовика, и не возьмут с вас и цента. Пол Маккартни был прав — любовь не купить.

Отрицательное влияние внешних вознаграждений сильнее всего проявляется в творческих задачах. Тереза Амабайл, профессор Гарвардской школы бизнеса, провела многолетнее исследование креативности на реальных рабочих местах. На основании 12 ты-

сяч ежедневных записей в журналах, полученных от 238 креативных сотрудников из семи различных компаний, она искала взаимосвязь между эмоциями, событиями и творческим результатом. Большинство сотрудников говорили о том, что внешнее поощрение вообще не мотивировало их, а те, кого больше всего интересовали деньги, не очень преуспели в творческих задачах. Респондентов в большей степени мотивировали сложные задачи, общность, командный дух и ответственность.

Третья причина. Внешнее поощрение часто создает искаженные стимулы, которые не идут на пользу проекту. Когда каждый сосредоточен на максимальном увеличении своего бонуса, разработка игр превращается в политическую игру. Политика и сплетни ведут к разрушительной конкуренции среди разработчиков. Страх наказания заставляет скрывать результат и избегать риска. Чтобы оставаться в безопасности или завоевать уважение, разработчики могут объединяться в «банды» и отказываться предоставить информацию тем, кто в них не входит. Эта модель подтверждается исследованиями. Амабайл выяснила, что конкуренция подавляет творчество и движение идей и не позволяет людям помогать друг другу. Дэн Ариэли обнаружил, что участники теста, в котором нужно составить предложения, расставив слова в правильном порядке, гораздо реже помогают другим, если расшифровывают слова о деньгах. Наполнение сознания мыслями о долларах делает нас склонными к риску и эгоистичными — худшее поведение для группового творчества.

Четвертая причина. Внешние стимулы отвлекают от работы. Разработчики перестают думать о том, как улучшить игру, их мозг занят мыслями о том, как заполучить бонус больше. А каждая минута, потраченная на мысли о вознаграждении, — это упущенное время, в течение которого разработчик не решал задачи разработки.

Еще хуже, когда внешними стимулами является наказание. Босс может угрожать уволить кого-то, урезать зарплату, наорать или даже проявить неуважение, закатывая глаза в ответ на любые предложения. Подобные угрозы могут привести к немедленной бурной деятельности, но одной деятельности недостаточно, чтобы сделать крутую игру. Геймдизайн требует открытого об-

суждения и глубоких размышлений, а под давлением это невозможно. Помните, что страх и гнев подавляют верхнюю височную извилину, отвечающую за творчество? Как писал Фрэнк Герберт в «Дюне», страх убивает сознание — если мы боимся, то неспособны в полной мере раскрыть свои творческие возможности. Вот почему Амабайл обнаружила, что люди, вероятнее всего, совершат прорыв, будучи радостными и счастливыми. Работа в радость позволяет разуму расслабиться и способствует высокоэффективным размышлениям. Страх, вызванный внешними угрозами, может мотивировать, но при этом он может подавлять способность свободно мыслить и выполнять работу должным образом.

Все это может случиться сразу. Например, рассмотрим систему, о которой я когда-то слышал: если разработчики представляли что-то классное на совещаниях в стиле «покажи и скажи», то получали за это ваучеры на покупки в интернете. Эта система — почти абсолютное зло. Люди начинают чувствовать, что работают ради ваучеров, а не ради работы, поэтому теряют интерес к тому, что делают. Сотрудники, которые выполняют неочевидные технические задачи «за кадром», обижаются на тех, кто делает визуально заметную работу, которую показывают на презентациях. Сотрудники тратят время на усовершенствование своей презентации «покажи и скажи», а не делают работу, которая действительно улучшит игру. Им приходится тратить время на размышления о дурацких ваучерах и о том, смогут ли они их получить. В итоге они будут ненавидеть систему, которая обращается с ними как с собакой Павлова.

Значимая работа

Так как же нужно мотивировать разработчиков? Не стоит поощрять геймдизайнеров деньгами за механику или за идею — это приводит к появлению бесполезных механик или идей. Наказание за неудачные идеи, прототипы или тесты только подавляет готовность рисковать, а это вредит инновациям. Награда за отработанное время приводит к появлению большого количества тел на стульях и не способствует реальной производительности. Поощрение отдельных сотрудников за что-либо во-

обще приводит к зависти и нездоровой конкуренции в команде. Поощрение части сотрудников делит команду на бездельников и неофициальных рабовладельцев. Кажется, что все приведет к неминуемой катастрофе.

Тем не менее в мире появляются новые игры, и некоторые весьма хороши. Хотя проблема мотивации структурно очень сложна, ее возможно решить, зная кое-что о разработчиках.

Разработчики хотят
выполнять значимую
работу.

Больше всего творческие люди хотят делать работу, которая в мире будет значима. Разработчики будут из всех сил искать подходящий способ улучшить игру. Никому больше даже не нужно подробно разбираться в том, что он делает, потому что мотивация идет изнутри. Если разработчик знает, что успешно справился с задачей, он будет рад. Если нет, то расстроится.

Но не всегда можно предложить такую значимую работу, особенно в больших студиях. Существует неочевидный способ создания значимой работы. Задача должна приносить пользу. Однако есть и другие важные свойства значимой работы. В идеале она должна предлагать творческий труд, сбалансированные сложные задачи, гордость, признание, право собственности, ощущение востребованности, ответственность и свободу. Задача организации команды заключается в создании среды, которая последовательно обеспечивает работу, предполагающую эти ценности. Джон Лассетер, аниматор студии Pixar, выразил это так:

«Творческим людям быстро становится скучно, они зависимы от настроения и работать с ними непросто. Необходимо позаботиться о них, а также о том, чтобы им было весело. Творческие люди делают действительно хорошую работу, только если перед ними поставлена интересная задача. Им должно нравиться то, над чем они работают. Они должны сильно гордиться тем, что являются частью проекта. Это задача менеджера. Каждый раз вы должны ставить перед ними творческие задачи. Это сложно, но никто не говорил, что работать с творческими людьми легко».

Чтобы креативный разработчик был доволен, не нужна «денежная морковка». Нужно сочетание ответственности, доверия,

сложных задач и веры в проект. Кроме того, как отмечает Лассетер, чем лучше люди, тем труднее их мотивировать. Незаинтересованные, посредственные разработчики не чувствуют неудовлетворения, работая над неинтересными задачами, потому что незаинтересованность является их нормальным состоянием. Они — как разнорабочие, работают исключительно за деньги. Великие дизайнеры живут неконтролируемым потоком идей и амбиций. Они должны давать им выход, в противном случае они становятся несчастными. Эта черта характера является как источником их способностей, так и причиной, по которой им «быстро становится скучно, они зависимы от настроения и работать с ними непросто».

Самоопределение предполагает, что разработчик воспринимает себя как часть той работы, которую он делает. Если мотивация находится на этом уровне, разработчик будет размышлять над проектом в душе, в машине и даже когда спит. Он будет тратить любую свободную минуту на размышления над идеями и поиски возможностей и решений. Будет настолько глубоко погружен в этот процесс, что окружающих будет раздражать его рассеянность. Он будет записывать идеи на салфетках или носить с собой блокнот. Будет искать варианты решения задачи, может прийти на работу в любое нерабочее время или встать посреди ночи, чтобы записать идею, которая ему приснилась. Работа больше не просто работа. Это его гордость и цель.

Наиболее трудно-
достижимая цель
в системе поощрения
разработчиков —
САМООПРЕДЕЛЕНИЕ.

Самоопределение — это целая несокрушимая стихия. Именно так стартапы побеждают гигантские корпорации, как и братья Райт опередили богатого Сэмюэля Лэнгли. Большинство компаний никогда не видят этого, потому что они убивают самоопределение денежной «морковкой» или «эффективными» управленческими методами. Берегите самоопределение, оно может творить чудеса.

Настроение

В благоприятной среде люди чувствуют прилив энергии и безопасность. Они достаточно защищены, чтобы рисковать и задавать вопросы, а их мозг сосредоточен на работе. В плохой

неблагоприятной среде они злятся и чувствуют страх. Страх нейтрализует их способность к творчеству, и они тратят всю свою энергию на уклонение от ответственности.

НАСТРОЕНИЕ — это эмоции, которые люди ежедневно испытывают на работе.

Среда зависит от ожиданий людей того, как к ним будут относиться. Если они рискуют, а идея проваливается, ожидают ли они того, что их обвинят, или же ждут, что их успокоят? Если они спрашивают о чем-то руководителя, ожидают ли они подробного ответа или словесной пощечины? Если разработчики чувствуют опасность, они будут работать в страхе и намного хуже своих возможностей. Если они чувствуют благоприятную обстановку и поддержку, они будут идти к цели и рисковать.

Рассмотрим разработчика с крутой, но рискованной идеей. Он решает, рассказать о ней или забыть ее. Он хочет сделать игру хорошей, но его также волнует его социальный статус и эмоциональное благополучие. Таким образом, он инстинктивно просчитывает плюсы и минусы. В благоприятной творческой среде это выглядит так:

За	Против
• Если идея будет успешной, мне припишут ее успех и игра принесет пользу • Я смогу выразить себя • Я смогу лучше понять свою идею, проанализировав умные ответы коллег, и, следовательно, лучше пойму, как я думаю и как мои идеи связаны с другими • Мне будет весело и интересно обсудить идею со своими друзьями-дизайнерами	• Я потрачу как минимум минуту на описание и обсуждение идеи, и она может ни к чему не привести

Очевидно, что дизайнер расскажет об идее. Это лучший результат для игры и студии. Идея может провалиться, но решение рассказать о ней было правильным. И остается шанс, что эта идея превратит игру в хит.

Теперь рассмотрим пример студии, в которой витает атмосфера вины, страха и неправомерного распределения власти:

За	Против
• Если идея успешна, возможно, мне припишут ее успех (если кто-то не украл ее и если люди не забыли, откуда она взялась)	• Я потрачу хотя бы минуту на описание и обсуждение идеи, и она может ни к чему не привести • Меня могут неправильно понять или прервать, прежде чем я смогу выразить свою мысль. Это было бы хуже, чем молчать • Меня разорвут на части, если в моей идее дыра, о которой я даже не подумал. Это не пойдет на пользу моему общественному положению • Если моя идея хороша и ее примут, ее может растоптать «эффективный» менеджер, как только человек, наделенный большими полномочиями, заинтересуется ею

В такой атмосфере идея остается похороненной. И поскольку этот шаблон повторяется тысячи раз, игра не может развиваться, потому что никто не рискует. Через год или два все задаются вопросом, почему игра такая легкая, такая неоригинальная, такая безопасная. Обзорщики не дают ей положительных отзывов, и игра умирает.

Обычно никто не понимает, что произошло, потому что плохая атмосфера — молчаливый убийца. Она не приводит к катастрофам разработки. Скорее она делает так, что хорошие идеи не реализуются. Она делает так, что люди не предлагают рискованные идеи и не ведут необходимые дебаты. И хотя никто этого не замечает, отсутствие вышеуказанных параметров ежедневно наносит игре непоправимый вред.

Среда является одним из самых важных факторов, определяющих качество игры, потому что наполняет все. Она меняет отношение людей, их мысли и действия в любом месте, каждую секунду, каждый день.

Страх и любовь

Бывают такие руководители, которые в качестве мотивации используют гнев. Чтобы подтолкнуть людей, они любят прибегать к таким методам, как крик, изощренные оскорбления и вздохи разочарования. Суть состоит в том, что угроза эмоциональной боли не даст людям лениться.

В других областях такой подход может быть эффективным. Как-то я смотрел документальный фильм о том, как знаменитый британский шеф-повар Гордон Рамзи руководил подчиненными на своей кухне. Он все проверял. Малейшая ошибка в блюде или его приготовлении приводила его в бешенство. Он кричал на поваров и официантов за мельчайшие ошибки. Он увольнял по человеку каждую неделю. И это хорошо работало — Рамзи получил самые высокие награды. Но рассмотрим характер работы. Метод Рамзи заключался в поддержании стандартов. Его ресторан был великолепен не благодаря креативности его команды, а благодаря способности прекрасно выполнять четко определенные процедуры приготовления и подачи блюд. Проверки Рамзи были эффективными, потому что опытный шеф-повар легко замечал ошибки в приготовлении пищи. Нет никакого смысла в том, чтобы персонал кухни был креативным или рисковал во время приготовления или подачи ужина.

Но разработка игр — это другое. Дизайн требует рисков, неудач и креативности на всех этапах разработки. Гнев и страх разрушают готовность людей идти на риск и их способность к творческой работе. Самоопределение становится невозможным, если вас всегда заставляют танцевать, стреляя при этом в ноги.

Гнев манит, потому что на первый взгляд выглядит эффективным. Руководитель кричит и видит, как ленивые разработчики начинают суетиться. Но он не видит, как меняется желание 30 сотрудников, наблюдающих за этой живописной картиной, рисковать в дальнейшем. Он не понимает рискованную, но интересную мысль о том, что подчиненный теперь будет бояться его.

Великий кукловод Джим Хенсон всегда хорошо обращался с сотрудниками и никогда не кричал на них. Джим говорил: «Вы хотите на Луну. Вы должны стремиться к Юпитеру. Если вашей целью будет Юпитер, вы обязательно попадете на Луну». Он руководствовался вдохновением, совместной работой и признательностью. Он сосредоточился на других, а не на себе. Находясь на вершине власти, владея миллионами долларов и домами на нескольких континентах, он оставался добрым и отзывчивым. Люди работали на него из любви. Они рисковали, потому что им было нечего бояться. Они отдавали ему все, потому что им

не нужно было тратить энергию на то, чтобы защищаться. Их самоопределение сработало, и команда Джима оставалась на вершине в течение десятилетий.

Джим не направил все свои силы на немедленное поощрение. Он не поддавался желанию властвовать над сотрудниками. Он понимал, что его работа состоит в том, чтобы «создавать» других для достижения успеха, а не использовать их в качестве инструментов, что его команда была гораздо важнее, чем он сам.

Чтобы собрать урожай щедрости любви, нужно терпение. Страх — это легко, быстро и очевидно, но, в конце концов, это слабое топливо для творчества. Любовь медленная, косвенная и тихая, но будучи возвращенной, необычайно сильна. Только любовь может выпустить самоопределение, благодаря которому каждый разработчик сможет извлечь творческую силу из своей сути.

Вот почему в приемной своей студии Хенсон установил гигантскую табличку со старой цитатой Г. К. Честертона: «Художественный темперамент — болезнь, которой подвержены все любители».

Социальная мотивация

Мы уже видели, что внешние поощрения могут нарушить мотивацию разработчиков игр, но в то же время это единственный внутренний стимул выполнять значимую работу, благодаря которому можно создать наше лучшее произведение. Но иногда все еще есть место для особых, мотивирующих стимулов для вдохновения. Ключевым моментом здесь является не экономическое поощрение или наказание, а более тонкие социальные сигналы. Давайте рассмотрим некоторые виды мотивации, которые настраивают разработчиков игр на нужный лад.

Плейтесты как вид мотивации

Конечно, хотелось бы, чтобы тестировщикам нравилось то, что мы сделали. Здорово, когда они просят поиграть еще. С другой стороны, это ужасно — видеть, что тестировщику скучно, слишком сложно или он находит в игре баги. Удовольствие от успешного и боль от неудачного плейтеста являются сильнейшей мотивацией.

Плейтесты хорошо мотивируют, создавая естественные, неограниченные, заслуживающие доверия результаты принятых решений разработки.

Это похоже на метод кнута и пряника. Но этот метод отличается от стимулов, которые предлагает начальник.

Во-первых, результаты плейтеста являются прямым, естественным следствием. Это не система поощрений, созданная одним человеком с целью манипуляции. Это реальные результаты из реальной жизни. Другими словами, плейтест не работает как кнут и пряник.

Во-вторых, если мы используем плейтест как средство мотивации, не нужно ограничиваться мыслями о том, чего от нас ждет начальник, или полагаться на его понимание. В погоне за одобрением босса мы ограничены его пониманием игры. Выше головы не прыгнуть. Вы всегда ограничены пониманием человека, который вас оценивает. Но если вы работаете над реальностью, а не гонитесь за оценкой, то ничем не ограничены и можете раскрыть весь свой творческий потенциал. Игра всегда может понравиться тестировщику еще сильнее, чем раньше.

Наконец, тестировщики заслуживают доверия. Они не поощряют нас за то, что мы придерживались того же мнения или предположений, что и босс. Они поощряют нас за объективно хорошую работу. Разработчики не сомневаются и не бунтуют против реакции тестировщиков на игру так, как если бы они сомневались или бунтовали против манипулятивных мнений руководителя.

Ожидания как вид мотивации

Если все относятся к вам как к дураку, вы можете просто начать играть роль дурака или даже поверить в то, что так оно и есть. Если все смотрят на вас с восхищением, вы начнете копаться в своих идеях, говорить уверенно и раскрывать свои способности думать лучше, так как хотите соответствовать этому образу.

Если вы относитесь к людям так, как будто они работают хорошо, они будут работать хорошо.

Этот эффект повсеместен. Например, исследователи выяснили, что женщинам сложнее сдать экзамен по математике, чем мужчинам, только если вы не скажете им, что и у женщин, и у мужчин одинаковые способности к этой науке. Их ожидания меняют их убеждения в том, что они могут сделать на самом деле, что, в свою очередь, влияет на то, что они делают. Другое исследование показало, что

если учителям говорили, что какие-то дети, возможно, являются гениями, то эти ученики показывали более высокий результат, чем остальные дети, даже если это были совершенно обычные дети, выбранные случайным образом. Учитель относился к ним как к умным, и вели они себя соответственно.

Некоторые пытаются делать наоборот. Они плохо относятся к другим, пытаются заставить их работать лучше. Такие люди слепо верят, что человек разозлится и, пытаясь всем доказать их неправоту, станет работать гораздо лучше и усерднее. Но это не работает. Хуже некуда, когда с тобой обращаются как с некомпетентным сотрудником. Человек злится, это разрушает творческие способности и создает чувство беспомощности. Если вы проделали хорошую работу, а с вами обращаются как с идиотом, зачем продолжать эмоционально вкладываться в нее?

Лучшие игровые команды стремятся стимулировать чувство элитарности. Они развивают дух товарищества вокруг определенного знака или идеи, которые отделяют их от других. При этом они могут быть разработчиками среднего уровня, но вера в избранность команды побуждает их делать больше. Именно поэтому у Уолта Диснея не было обычных дизайнеров тематических парков, у него были имаджинеры (Imagineers). Вот почему армии постоянно подчеркивают уникальную историю каждого подразделения. Чувство элитарности создает прецедент быть достойным и укрепляет внутреннее чувство индивидуальности. У разработчиков, к которым относятся как к посредственностям в сером типичном офисе с перегородками, нет причин быть лучше. Выделите этих людей, позвольте им создать что-то уникальное, и они оправдают ожидания.

Резиновая курица как мотиватор

Рассмотрим пример «сломанных компиляций». В разработке видеоигр ошибка в одной строке кода, или, иначе говоря, плохо сконфигурированная часть содержимого либо недопустимый сюжет, может испортить весь код в случае попадания в центральную базу данных, вследствие чего игра ломается. В результате команда несет большие потери, поскольку продолжение процесса разработки возможно только после устранения ошибки.

Несерьезные, не-явные, случайные социальные награды и наказания могут донести информацию, не нарушая творческую среду.

Чтобы избежать «сломанных компиляций», естественной мотивации недостаточно. Ответственность за «сломанную компиляцию» обычно несет вся команда, а не конкретный человек, по чьей вине это произошло. В результате создаются условия для систематических мотивационных искажений, которые приводят к более частому появлению «сломанных компиляций», чем было бы оптимально для нормального рабочего процесса. Как же можно это предотвратить?

Виновников «сломанных компиляций» можно наказывать денежными штрафами, но это усилит гнев и паранойю, поскольку люди слишком уверены в безопасности своей работы. Выговор не работает — он может легко ухудшить взаимоотношения и атмосферу. Лучше использовать резиновую курицу.

Это работает так: если кто-то «сломал компиляцию», то должен самостоятельно устранить ошибку, и пока не сделает этого, на его столе будет лежать резиновая курица в качестве символа (незначительной) оплошности. Курица остается до тех пор, пока кто-то еще не «сломает компиляцию». Здесь нет споров и прямой конфронтации. Вы просто садитесь за рабочий стол и находите на клавиатуре курицу, глядящую на вас своими резиновыми глазами.

Человек, у которого лежит курица (как было много раз и у меня), становится предметом дружеских подколов, но его никто не осуждает всерьез. Никто не хочет получить курицу, но никто и не сердится и не впадает в уныние, если это происходит. Люди не сходят с ума из-за «сломанной компиляции», но и не забывают о ней. Резиновая курица — это не только смешной, но еще и идеально сбалансированный, бесконфликтный инструмент мотивации.

Как однажды заметил Наполеон, «солдат будет бороться долго и упорно ради кусочка цветной ленточки». Курица — один из инструментов социальной мотивации. Для других целей могут использоваться разные методы. Пospорьте с кем-нибудь на кружку пива, что человек не сможет придумать, как решить задачу дизайна (в любом случае, вы выпиваете вместе). Найдите в игровых файлах что-то новое и интересное и выведете находку на мониторы в офисе, чтобы все видели. Мы можем задействовать сотни небольших мер и системы, с помощью которых можно

создавать и передавать социальные сигналы неочевидными, ненавязчивыми способами, которые дадут плодотворный положительный результат.

Принцип прогресса

Исследовав 12 тысяч записей в журналах сотен творческих работников, Тереза Амабайл тщательно проанализировала данные в поисках закономерностей, которые активизировали творческий потенциал. Вы уже можете догадаться, что это были не деньги или страх. Она нашла кое-что удивительное: это было не общественное одобрение и не стремление создать хороший продукт. Амабайл обнаружила, что мотивация в основном заключается в наличии ежедневного прогресса в работе. Она назвала это принципом прогресса.

Амабайл выявила, что размер прогресса не так важен, как его частота. Мотивация лучше всего поддерживается небольшими победами — нахождением алгоритма, завершением анимации или наблюдением за тем, как тестирующий понимает некоторые детали, которых раньше не понимал. Даже если игра в целом все еще ужасна (а они всегда находятся на ранней стадии разработки), такие мелкие победы создают ежедневную вовлеченность.

Применение принципа прогресса означает необходимость организовать процесс разработки таким образом, чтобы каждый имел возможность получать регулярные, видимые маленькие победы.

Все основы хорошей разработки способствуют этому: итерируя, вы сможете наблюдать постоянный прогресс на примере частых плейтестов, а благодаря естественной власти люди могут наслаждаться своими победами, а не приписывать их боссам. Но мы также можем пойти дальше, организуя и отслеживая работу так, чтобы прогресс был частым и заметным. Даже такая простая вещь, как работа с TODO-листом, наглядно показывает прогресс. Каждое вычеркнутое задание дает небольшой эмоциональный заряд. Если ежедневно вычеркивать по две-три задачи, команда будет ощущать непрерывное движение вперед.

ПРИНЦИП ПРОГРЕССА предполагает, что наиболее сильным источником хорошей внутренней трудовой жизни является регулярный заметный ежедневный прогресс.

Такой процесс схож с разработкой обратной связи с поощрением для игроков. Мы могли бы дать игрокам возможность прокачивать персонажей ежечасно, так чтобы это было заметно, потому что мы знаем, что такой регулярный прогресс весьма приятен. Та же психология распространяется и на самих разработчиков игр. Даже самые маленькие победы могут поддержать, если будут частыми и заметными.

Работу можно организовать так, чтобы люди могли самостоятельно получать прямую обратную связь о своих успехах. Дизайнер может добиться результатов в разработке системы, но если его будет оценивать только другой человек в обзоре или в плейтесте, дизайнер никогда не увидит свой прогресс и не прочувствует его. У дизайнера должна быть возможность проводить плейтесты своей работы, как у программиста должна быть возможность наблюдать за автоматическим тестированием разработанного им кода, а у художника — возможность самостоятельно решать, когда персонаж вписывается в разработанный мир. Они создают свой собственный прогресс.

Принцип прогресса подходит маленьким проектам даже больше, чем масштабным. В большой команде один человек может работать на основании импульса всей группы. А в команде из одного-двух человек нужен еще один источник мотивационного топлива. Вот почему для крошечных команд подробные списки дел так полезны. Речь идет не только об организации задач. Каждое выполненное задание немного повышает настроение.

Процесс разработки в маленьких командах похож на написание книги. За годы работы с текстом я научился трепетно относиться к количеству написанных страниц. Я не трачу дополнительные четыре часа на переписывание, потому что в будущем хочу выпустить книгу. До этого еще слишком далеко, и мой мозг пока не воспринимает это событие как эмоционально значимое. Но я радуюсь, когда вижу, что число страниц увеличилось.

Сложные решения

Один мудрец покинул свой монастырь высоко в горах и отправился в Нью-Йорк. Он призвал всю свою накопленную мудрость. Вскоре он устроился поваром в забегаловку, спился и умер в одиночестве.

Последствия решений

Когда мы принимаем проектные решения, часто приходится думать не только о влиянии решения на саму игру. Например, изменение пола или этнической принадлежности главного героя может принести пользу истории игры, но отрицательно сказаться на продажах. Отказ от плохой идеи разработчика может в краткосрочной перспективе снизить загруженность, но ухудшить атмосферу в студии и усложнить удержание сотрудников. Программный продукт может быть дешевым в реализации, но дорогостоящим в долгосрочной перспективе. Другой программный продукт может быть дешевым, но позже приведет к риску появления критического отказа в проекте. Чтобы принимать правильные решения, часто приходится учитывать эти сложные последствия, влияющие на процессы, людей, бизнес и рынки. Давайте рассмотрим некоторые из этих последствий.

Эффект дизайна — это все, что связано с влиянием решения на игроков. Большая часть книги была посвящена оценке и прогнозированию эффектов дизайна.

Требуется время, чтобы написать код, сделать анимацию, записать диалог и выполнить множество других задач, которые заставляют дизайн работать. В эту категорию также входят стоимость исправления багов, настройки систем в рамках предусмотренных параметров и прочие простые в прогнозировании технические задачи. Поскольку затраты на реализацию настолько очевидны, их почти всегда планируют, хотя часто недооценивают.

ЭФФЕКТ ДИЗАЙНА — это влияние решения на саму игру.

СТОИМОСТЬ РЕАЛИЗАЦИИ — это ресурсы, необходимые для реализации решения.

Для большинства разработчиков работа со стабильным, совершенным ПО и игровыми системами — непозволительная роскошь. Обычно инструменты глючные и плохо задокументированы, игровая механика сделана наполовину, а история меняется ежедневно. Эти сырые элементы обременяют всех, кто с ними работает. Не смертельные, но раздражающие баги тормозят

БРЕМЯ НЕЗАВЕРШЕННОСТИ — цена, которую платят люди, вынужденные выполнять работу, которая зависит от незавершенных частей игры.

работу. Незавершенная история тормозит дизайн уровней, внося неопределенность в проектные решения уровней. Несбалансированная игровая механика мешает дизайнерам уровней оттачивать и балансировать соответствующие задачи.

Благодаря стеку зависимостей можно избежать сырого дизайна. Впрочем, даже если стек зависимостей будет четким и понятным, полностью устранить бремя незавершенности невозможно.

РИСКИ КРИТИЧЕСКОГО ОТКАЗА — это затраты, вызванные критическими отказами сырой системы.

Существенные недостатки могут появиться в любой сырой системе в самое неподходящее время. Серьезные, сложные баги могут сорвать тестирование. Непредсказуемые доминирующие стратегии могут разрушить месяцы работы над балансом. В течение нескольких месяцев разработчики могут не замечать пробелов в нарративе. Как бомбы замедленного действия, эти проблемы скрыты глубоко в игре, могут взорваться в любой момент и вызвать критический сбой.

Цена таких ошибок — не столько реальная работа по их устранению, сколько их влияние на другие процессы. Простой баг может привести к тому, что художнику в другом офисе придется потратить два дня на обнаружение ошибки импорта данных. Даже если исправление ошибки представляет собой изменение одной строчки кода, ущерб все равно нанесен: замедлилась работа художника и все то, что от него зависит.

Наибольший ущерб наносится тогда, когда такие бомбы срабатывают в критических ситуациях. Иногда случается так, что художник работал над графикой, которая была необходима для запланированного на следующий день плейтеста нового игрового режима. Из-за отсутствия графики приходится отменять

плейтест, в результате дизайнеры лишаются данных, а это, в свою очередь, тормозит процесс принятия решений по следующей итерации кода, оставляя программистов геймплея без четких инструкций. Такая цепная реакция нередка в больших и сложных процессах.

У каждого процесса разработки существует определенный организационный уровень, который помогает соблюдать координацию. Один разработчик может вести для себя заметки. Команда из трех человек может ежедневно собираться с целью обсудить координацию работы. По мере роста команды стоимость и сложность координации возрастают. Большие команды используют специально обученных разработчиков, системы отслеживания ошибок и вики по дизайну. Затраченные усилия являются нагрузками, связанными с процессом разработки.

НАГРУЗКИ, СВЯЗАННЫЕ С ПРОЦЕССОМ РАЗРАБОТКИ, — цена отслеживания и составления графика работ.

Низкий уровень нагрузок, связанных с процессом разработки, является одним из самых больших преимуществ небольших команд. В то время, когда я работал один над уровнями Unreal Tournament ради развлечения, четыре часа рабочего времени означали десять минут изучения своих заметок и три часа пятьдесят минут работы в редакторе уровней. Я знал о дизайне все и ни от кого не зависел. Если вы работаете над большими студийными проектами, четыре часа рабочего времени часто означают час написания спецификаций, еще один час обсуждения и два часа на саму работу в редакторе. В первом случае нагрузки, связанные с процессом разработки, занимают 4% моего времени. Во втором — 50%.

Процесс групповой разработки всегда является в некотором смысле продажей. У каждого есть доля влияния, которое он может потратить и обменять на улучшение игры. Это влияет на проектные решения.

ТАКТИЧЕСКИЕ ПОСЛЕДСТВИЯ влияют на отношения между разработчиками.

Например, в студии может работать очень опытный старший программист, который основал компанию 15 лет назад. Он не уделяет особого внимания сюжету, но любит работать с новыми графическими технологиями. В этой студии дизайн, который не раздвигает границы графических возможностей,

может иметь большие преимущества, но за это приходится платить раздражением опытного программиста, который может ополчиться против такого решения.

Работать с тактическими последствиями часто означает думать на несколько шагов вперед. У дизайнера может быть материал, который он хочет попробовать реализовать, но он понимает, что никогда не сможет урезать его, если идея не сработает, потому что в нее будет вложено слишком много сил других людей. Если вероятность того, что материал урежут, слишком велика, дизайнер не станет продвигать эту идею. Это неприятный результат, обусловленный недостатками организации, однако с точки зрения дизайнера так будет лучше для процесса разработки и самой игры.

КУЛЬТУРНЫЕ ПОСЛЕД-
СТВИЯ меняют при-
вычки разработчиков
и атмосферу развития.

Культура студии — это коллективный набор ожиданий, предположений и привычек команды. Принятые сейчас решения оказывают долгосрочное влияние на культуру и, следовательно, на дальнейшую работу. Некоторые решения могут улучшить культуру, другие — уничтожить ее.

Например, слишком частое необоснованное изменение деталей сюжета и основных идей дизайна приводит к вероятности дальнейших изменений. Разработчики уже не рассчитывают на то, что все останется без изменений, поэтому не вкладываются в идеи по максимуму. Они эмоционально страдают при виде того, как их любимую работу убивали слишком частые изменения истории. Личные вклады и творческая активность команды постепенно ухудшаются. Студия потеряла что-то ценное как в социальном, так и в денежном смысле, что не отражается на балансе игры. Они потеряли свою культуру.

Именно культура в долгосрочной перспективе определяет судьбу студии. Она управляет каждым действием, предпринимаемым отдельно взятым человеком. Чтобы сохранить ее, стоит пожертвовать краткосрочными достижениями.

ЦЕНА РЕШЕНИЯ — это
цена принятия пра-
вильного решения.

Мозги сотрудников, организация исследования и письменный анализ не бесплатны. За принятие правильного решения нужно что-то заплатить, а иногда лучшее решение — не тратить слишком много усилий на принятие решения.

Если решения незначительные, лучше, чтобы их принимал один разработчик. Было бы глупо потратить полчаса на анализ крошечного решения, которое не окажет существенного влияния на проект. В данном случае то, что выглядит как лень, на самом деле является рациональным безразличием. Вы видели это на примере итерации боевого дизайна.

И напротив, важные решения должны быть тщательно продуманы. Необходимо собрать всех участников проекта, провести несколько совещаний, сдать письменные отчеты, исследования и обзоры. Правильное решение стоит своих затрат.

Последствия решений. Кейс

Представьте, что вы дизайнер, работающий над шутером с элементами фэнтезийной RPG. Ваша студия, Dragon Brain Games, была основана два года назад бизнесменом и известным автором жанра фэнтези Алланом Макреем, который воспользовался своей популярностью, чтобы обеспечить финансирование проекта *Talmirian Gods: The Journey of an Age*.

Проект почти завершен. Через месяц контент будет заблокирован, после чего вы сможете только устранять баги и не сможете выполнять какую-либо значимую разработку. Так как в прошлом Макрей был писателем, то использовал авторский подход к игре, поэтому до недавнего времени не было никаких плейтестов. Вы считаете, что Макрей прекрасно справился с написанием текста. Вы не совсем довольны боевой механикой, так как у вас мало времени для итерации, но считаете, что все сделали правильно, учитывая, что этот процесс был ориентирован на нарратив.

В игре существует главный враг по имени Валрог. Этот гигантский зверь является центральной частью ряда боев. Вы были уверены, что будете комбинировать его с другими врагами как можно чаще, чтобы создавать элегантные боевые вариации, и лично проверили Валрога на отсутствие вырожденной стратегии.

Но у вас не получилось. Поскольку ранее в проекте не было систематического плейтестинга, пришлось тестировать боевые столкновения в одиночку. Как только вы начали запускать

плейтесты всей игры, тестировщик быстро нашел вырожденную стратегию. Одним из первых доступных в игре прокачек персонажа является Эликсир скорости, который позволяет постоянно немного увеличивать скорость передвижения персонажа. Благодаря этому Валрога можно бесконечно гонять по кругу, самому оставаясь при этом невредимым. Бои превращаются в цирк, когда игрок бежит вокруг Валрога, гоняя его до смерти. Тестировщики громко смеются, а грозный Валрог беспомощно вращается по кругу. Потом тестировщикам становится скучно, поскольку они снова и снова используют одну и ту же повторяющуюся тактику.

Четверть тестировщиков выбирают Эликсир скорости, и большинство из них, естественно, находят вырожденную стратегию. Что же делать в таком случае?

Очевидный выбор — сделать Валрога шустрее. Кажется, что, увеличив скорость поворота, можно решить проблему. Но это рискованно. Сырой код анимации игры не позволяет произвольно растягивать и уменьшать имеющуюся анимацию. Чтобы изменить скорость Валрога, придется использовать новую анимацию. Вы не уверены, действительно ли это решит проблему, какая необходима корректировка или как это изменение повлияет на баланс боев с участием Валрога у игроков без Эликсира. Валрог является ключевым элементом многих боев, и изменить баланс их всех будет непросто. Кроме того, команда аниматоров перегружена работой, так как им нужно срочно внести изменения, чтобы игра соответствовала книгам, на которых основана, а руководитель ушел на неделю в отпуск. Более быстрая анимация поворота может решить проблему, но вы рискуете получить критический сбой из-за других проблем с балансом. Кроме того, существует вероятность, что изменения анимации не решат проблему вырожденной стратегии полностью, но пока не протестируете, ничего не узнаете.

Другой вариант — убрать Эликсир скорости из игры. Это простое решение осложняется тем, что Эликсир был в презентации игры на игровой конференции и фанаты рассчитывают увидеть его. Эликсир играет второстепенную роль в книгах, на которых основана игра, но вы не уверены, захочет ли Макрей отказаться от него — раньше он не всегда был открыт для изменений нарратива. Есть и небольшая последовательность геймплея, основанная

на Эликсире, а еще он используется в качестве награды за один квест. Дизайнер этого квеста — ваш друг, и он хочет сократить эту последовательность, чтобы выручить вас, но в квесте тогда должна быть другая награда. Эликсир можно заменить золотом, но это может нарушить экономический баланс, если мы дадим игрокам слишком много золота.

Другой вариант — не делать ничего. Это однопользовательская ролевая игра, а не мультиплеер, поэтому идеальный баланс с запредельно высоким потолком мастерства тут ни к чему. Даже если Валрога можно будет победить, игра не провалится. Обычно такой вариант является самым привлекательным. Тем не менее Макрей в ужасе от тестировщиков, смеющихся над самыми эффектными сценами. Продажа бездействия стала бы стратегической задачей.

Можно пока ничего не делать и потратить время на анализ в поисках других вариантов. Проведите мозговой штурм, проведите исследование, напишите идеи на карточках и перемешайте их или просто поработайте над чем-то еще и подождите, пока в работу включится бессознательное мышление.

Эта сложная ситуация включает в себя почти все последствия, которые может иметь принятое решение.

Различные решения имеют разные последствия влияния на различные закономерности игрового баланса и экономику. У них разная стоимость реализации: новая анимация стоит дорого; удаление Эликсира — дешево.

Бремя незавершенности, результат дисгармоничной системы анимации игры, лишает вас столь необходимой информации о возможных последствиях изменения скорости Валрога, а это значит, что новая анимация создает риск критического отказа, если в других смежных местах она не сработает. Замена Эликсира на золото влечет за собой риск нарушения игровой экономики — еще один потенциальный дизайн-эффект.

Взаимоотношения с аниматорами, Макреем, программистами и другими дизайнерами также влияют на тактические последствия. Отказ от Эликсира кажется намного привле-

кательнее, так как вы дружите с дизайнером, который работал квест. Изменение истории кажется намного менее привлекательным, так как Макрей — не ваш друг, а босс.

Наконец, предыдущая реакция Макрея на изменение нарратива создала атмосферу неопределенности. Эта дополнительная неопределенность привела к тому, что стало сложнее принять решение, что для Макрея неочевидно.

Все это «обернуто в слои неопределенности». Вы не знаете, как фанаты и журналисты отреагируют на манипуляции с Эликсиром, что о нем думает Макрей, какие именно изменения анимации необходимы для нормальной работы Валрога и так далее.

Эта ситуация может показаться чересчур сложной. Но этот уровень сложности принятия решений не редкость при создании сложных продуктов для многосегментного рынка, особенно при наличии команды людей, у каждого из которых свои навыки, взаимоотношения и желания. Настоящие дизайнеры в больших студиях сталкиваются с этим ежедневно; решение таких кафкианских задач — их основная часть работы. А заведомо правильного ответа не существует.

Могу сказать следующее: слишком много неопределенности. Лучшее решение, на мой взгляд, — собрать информацию, задав несколько нерискованных вопросов. Встретьтесь с Макреем и не стесняйтесь невзначай упомянуть про «убийство» Эликсира, чтобы прочувствовать его реакцию. Как правило, можно узнать мнение человека, фактически ничего не делая. Поговорите с аниматором, который вам больше всего симпатизирует, о возможной необходимости изменений Валрога, и сделайте соответствующие выводы. Может, это окажется проще, чем предполагалось, или об этом вообще не может быть и речи.

Информацию вы получите в любом случае. Наконец, подумайте над задачей и выжмите все соки из руминации, напишите другим разработчикам, чтобы они поделились идеями. Если в течение дня вы будете собирать информацию всеми способами, то сможете рассмотреть задачу уже с большей уверенностью. Потрачен один день, но, скорее всего, решение будет намного лучше, и я считаю, что это того стоило.

Ценности

Скульптура была идеальной — синусоидное воссоздание человеческой фигуры, величавой и совершенной. Чтобы выразить свою благодарность, горожане предложили скульптору все, что он пожелает. Он мог бы получить стада баранов, богато украшенное золотом оружие или жениться на дочери градоначальника.

Но скульптор все равно выглядел равнодушным. Он прошел мимо богатств и зашел в мастерскую деревенского кузнеца. Там он начал рассматривать полки, подбирая новые инструменты для своей работы.

Из этой книги вы многое узнали о том, как быть геймдизайнером. Добавьте к этим знаниям практику и приобретете навыки. Впрочем, не думаю, что знания и навыки — это все, что нужно для идеального геймдизайна. Думаю, что вам также понадобятся ценности.

Каждая профессия имеет свои ценности. У солдат есть боевые ценности, такие как верность, честь и отвага. Ученые ценят устойчивость, беспристрастность и честность. Промышленность ценит тяжелый труд, умение думать наперед и выполнять свою работу. Но ни одна из этих ценностей не подходит идеально для геймдизайна, потому что перед нами стоят другие задачи. Было бы глупо применять военные ценности к научной работе, и глупо применять любую из их ценностей к тому, что делаем мы. Нам нужен другой набор ценностей. Но какими они должны быть?

Я не считаю, что кто-то может указать набор лучших ценностей для всех дизайнеров, но думаю, что каждому дизайнеру будет полезно поразмышлять о том, каких ценностей он придерживается. Ценности поддерживают нас и являются неизменными стандартами, которые позволяют нам устоять перед тактическими и эмоциональными кризисами в работе.

Вот те ценности, в которые верю я. А во что верите вы?

ЦЕННОСТЬ — это эмоциональный выбор того, кем мы хотим быть. Человеческое качество, к которому нужно стремиться. Никто и никогда не сможет полностью воплотить свои ценности, но мы стараемся сделать себя и свою работу лучше.

Открытость

Открытость означает уважительное принятие идей, с которыми вы не согласны. Без открытости студия должна либо унифицировать мнения всех сотрудников и прекратить обсуждения, заставив людей замолчать, либо утонуть в неистовом потоке разногласий.

Открытость не означает просто позволить людям говорить, она предполагает уважение их вклада, искренность и готовность к диалогу. Не нужно принимать любую идею, важно думать о каждой идее осмысленно. Это не всегда легко. Как сказал Аристотель: «Признак образованного ума — развлекать себя мыслью, не принимая ее за истину».

Открытость произрастает из веры в неопределенность ваших собственных идей. Лучшие дизайнеры, которых я знаю, в начале почти каждого предложения скажут: «Моя интуиция подсказывает мне» или «Из того, что я видел», «Я почти уверен в этом» или «То, как я это вижу». Это не просто слова-паразиты. Эти дизайнеры поняли, что быть неуверенным — это честно. Мы всегда не уверены. Постоянно выражая неуверенность, они оставляют открытую возможность для обсуждения. Они оставляют место для других, более совершенных идей, и в итоге становятся мудрее.

Прямота

Плыть по течению чужих мнений весьма заманчиво. Не нужно много думать и рисковать положением в обществе. Все довольны. Но если вы постоянно со всем соглашаетесь, то в долгосрочной перспективе это убивает игру. Если никто не указывает на несоответствия, они проникают в проект и разрастаются как раковая опухоль. Команда разработчиков соглашается и соглашается, в итоге игра выходит в продажу с множеством недочетов. На первый взгляд постоянное «единодушие» можно принять за сплоченность команды. В действительности же это сигнализирует либо о процветающей лени и нежелании думать, либо о страхе.

Если открытость заключается в принятии идей, то искренность и прямота предполагают наличие собственного мнения и готовность им поделиться. Это значит иметь мужество высказывать

противоположное мнение, даже если было бы легче промолчать. Честный разработчик укажет на ошибку в концепции, предложенной боссом, или найдет в себе смелость не согласиться с мнением большинства людей в конференц-зале. Честные разработчики — вдумчивые, искренние и прямолинейные, а не желающие угодить. Они генерируют новые идеи и прикладывают эмоциональные усилия для их воплощения.

В конце концов, достойные люди уважают и ценят тех, кто поддерживает их идеями, даже если они не всегда соглашались с самими идеями. Глубоко в душе всем нам хочется быть тем, у кого есть сила сформировать независимое мнение и уметь высказать его.

Смирение

Геймдизайн — это очень, очень сложно. Игра — это сотни игровых механик, взаимодействующих с миллиардом компьютерных транзисторов, использующих десять различных основополагающих технологий, каждая из которых реализует тысячи алгоритмов, все они взаимодействуют с живыми игроками, которые воплощают всю сложность и разнообразие человеческой натуры и живут в мире безумной культуры и рыночной экономики. Такая система непостижима. Наш разум изначально не предназначен для того, чтобы разбираться в таких сложных системах.

Источником многих ошибок геймдизайна являются убеждения, что мы понимаем то, чего на самом деле не можем понять. Самонадеянность приводит к избыточному планированию и заставляет нас делать поспешные самоуверенные выводы. Она заставляет нас делать меньше плейтестов, чем необходимо, и из-за него мы пропускаем серьезные ошибки. Я считаю, что одна из самых важных базовых вещей, которые мы можем сделать, чтобы стать геймдизайнером, — это не выучить как можно больше, а осознать, как мало из того, что мы делаем, можно по-настоящему понять. Следует быть смиренными перед задачей, которую мы пытаемся выполнить.

Смирение позволяет принять то, как мало мы знаем. Так можно постичь не только маленькие берега понимания; смирение

открывает перед нами непостижимые океаны неведения. Оно помогает нам поймать волну серендипности, когда мир пытается нас чему-то научить. Оно противостоит нашему природному заблуждению «что вижу, то и существует», делает нас наблюдательнее, вдумчивее и, возможно, даже немного мудрее. Нет человека глупее, чем тот, который считает, что знает все.

Голод

Представьте, что вы создали что-то превосходное — отличный уровень, прекрасную песню, испекли идеальное печенье. Все аплодируют вашему успеху. Вы победили. Вы лучший. Что вы делаете теперь?

На этом этапе большинство людей останавливаются. Как только они превзошли все стандарты и обошли всех конкурентов, то решили, что теперь уж точно все. Таковая человеческая природа — делать только то, что нужно для достижения цели. Если этой целью является одобрение или поражение конкурента или денежное поощрение, нет никаких оснований продолжать, если вы добьетесь своего. Так поступают нормальные люди.

Однако существует и другой тип людей. Их не волнуют стандарты или конкуренты. Они не работают ради внешних наград или одобрения. Они работают ради работы. Они хотят получить незаменяемое чувство удовольствия делать что-то лучше, чем раньше. Они испытывают чувство голода.

Голод — это вера в то, что независимо от полученного результата мы можем добиться большего. Это постоянное желание сделать как можно лучше независимо от внешних ожиданий. Голод — не талант, голодный дизайнер может быть не более способным, чем его более «сытые» товарищи. Но он не привязан к стандартам окружающего его мира. Неважно, насколько хороша работа, он постарается сделать ее еще лучше.

Это сложно. Трудно превзойти то, что было сделано раньше. Часто даже бывает непонятно, возможно ли превзойти предыдущие стандарты. В своей книге «Талант ни при чем» Джефф Колвин пишет:

«Олимпийские рекорды столетней давности — лучшие результаты на планете — сегодня часто достойны лишь старшекласников. Победитель забега на 200 метров среди мужчин на Олимпиаде 1908 года пробежал дистанцию за 22,6 секунды. Лучшее время, показанное на сегодняшний день старшекласником, меньше на две с лишним секунды. Огромная разница. “Школьный” рекорд в марафоне сегодня превосходит результат победителя Олимпийских игр 1908 года более чем на 20 минут».

В гимнастике, музыке, шахматах и многих других областях то, что было необычным несколько десятков лет назад, теперь кажется простым и банальным. Это значит, что в 1908 году любой старшекласник, приложив достаточно усилий, мог победить олимпийского чемпиона. Но никто этого не делал, так как все были привязаны к определенным ожиданиям.

Голод — это постоянная неудовлетворенность не только работой, но и самим собой. Существует целый мир идей, которые нужно изучать, и вселенная информации, которую необходимо получать. «Голодный» дизайнер всегда видит себя в будущем как человека с большим количеством навыков и знаний, серьезным эмоциональным диапазоном и ответственной работой. Он стремится к этому ежедневно, постоянно совершенствуя свои навыки. Скорее всего, эти усилия в ближайшее время ни к чему не приведут. Но на протяжении многих лет или десятилетий они накапливаются. Каждый способен на большее, чем кто-либо мог бы от нас ожидать. Если мы сможем постоянно подпитывать в себе чувство голода, границы ожиданий исчезнут.

Эндшпиль

Реальность игр больше и шире, чем можно вместить в книгу или разум. Игры пронизывают умы и культуры нитями причинно-

Это книга моделей
и гипотез, а не реаль-
ностей и истин.

следственных связей, возвращаясь к истории людей и их подвидов, опутывают их жизни, на которые они будут влиять, и будущие культуры, которые будут их критиковать. Все это невозможно передать в книге. Я даже не пытался. Скорее я попытался создать руководство по ис-

кусству геймдизайна, в котором можно найти наиболее полезное описание игр. Но руководство не является абсолютной истиной. Это простая карта удивительно богатой и разнообразной территории. Независимо от того, сколько мы знаем, не следует забывать, что реальность намного шире.

Игра, в отличие от истории, — это не цепочка событий. Это система. Она превращает часть мира в набор механики и оборачивает ее так, чтобы мы могли играть. Игра не показывает нам

Игры — это мыслен-
ные модели разных
сфер жизни.

только одну нить событий, как это делает история, а позволяет нам снова и снова проживать этот кусочек мира в сотнях вариаций. Взаимодействие, в основе которого лежит исследование, дает информацию, которую вы не узнаете из историй.

Когда после неудачной попытки я перестаю доверять своей интуиции, то вспоминаю покер. В покере есть четкий месседж. В общих чертах он гласит, что никто не выигрывает каждую партию. Но просто читать — это не то же самое, что играть в игру. Фраза: «Никто не выигрывает каждую партию» — просто фрагмент текста, который можно положить на полку памяти. Только если мы сыграем в игру, то наш мозг прокручивает этот паттерн снова и снова, тысячами разных способов. Это повторяющееся взаимодействие не просто создает воспоминания, оно перестраивает образ нашего мышления. Мы не просто знаем, что никто не выигрывает каждую партию. Мы это понимаем, верим в это и чувствуем, потому что пережили такой опыт много раз.

Я есть то, в какие игры играл. Если я рискую, то думаю о шахматах и не забываю думать на два или три шага вперед. Если ленюсь, то думаю о футболе и о том, как важен каждый сантиметр, на который продвигаешься к цели. Если начинаю думать о задаче слишком узко, то вспоминаю о StarCraft и о том, как важно следить за общей картиной, потому что вокруг постоянно подстерегают именно такие факторы риска, о которых вы даже не задумывались. Это не просто воспоминания. Спустя годы работы с играми они стали частью меня самого.

Идеальный геймдизайн красив, но его нельзя увидеть или потрогать. Его красота в пространстве, которое он создает. Мы воспринимаем эту красоту, не просто просматривая события в определенной последовательности, а исследуя предложенные возможности. Как и мы, система игры меняет часть нашего разума в своем собственном образе. Система игры меняет часть нашего сознания. Вот как мы воспринимаем послания игр и их уникальную диалектическую силу. Как сказал Конфуций, «я слышу и забываю, я вижу и забываю, я делаю и понимаю».

Разработчики The Sims изобрели свой собственный жанр и продали сто миллионов копий. Half-Life и Counter-Strike произвели революцию в боях и подаче истории от первого лица. Dwarf Fortress подарила возможность строить фэнтезийные миры со своей политикой, экономикой и историей. Игровая механика Braid пронизана лирикой. Minecraft раскрыл креативные способности миллионов. А пока вы читаете эту книгу, кто-то где-то набрасывает идеи, создает прототип или размышляет над игрой, которая изменит мир.

Лучшие игры все еще
ждут своего появления.

В работе над играми присутствует особая свобода. Их границы так неопределенны, что я иногда задаюсь вопросом, существуют ли они вообще? Мы не ограничены бумагой или кинолентой. Игра может быть такой же простой, как детские кубики, или такой же сложной, как виртуальный мир, населенный миллионами существ. Ее можно исследовать, наблюдать, делиться и побеждать в ней. Она может длиться минуту или целую вечность.

На такой неисследованной территории нет ни избитых путей, ни ориентиров, ни границ. Здесь нет никого, кто мог бы вам помочь или удержать. Как сказал Эйнштейн, «воображение важнее знания». А ведь столько всего еще предстоит вообразить...

Рекомендуемая литература

Я в большом долгу перед всеми теми, на чьи идеи опирался, когда писал эту книгу. Без них она бы не появилась на свет. Идеи, которые я позаимствовал, настолько популярны, что я не могу перечислить все источники, которые повлияли на мою точку зрения. Мне с трудом удалось сократить список до десяти самых важных книг, откуда я почерпнул наиболее подходящие идеи, которые когда-либо находил. Эти книги более подробно раскрывают и разъясняют многие понятия, которые я затрагивал.

- **«Думай медленно... решай быстро»** Даниэля Канемана. В основном геймдизайн зависит от понимания человеческого мышления — игроков, команды и нас самих. Книга Канемана — это руководство по мышлению. Это лучшее из того, что я встречал о взаимодействии интуитивного и системного мышления, порождающего наши уникальные человеческие способности и чудачества.
- **«История на миллион долларов: Мастер-класс для сценаристов, писателей и не только»** Роберта Макки. Эта книга — лучшее руководство по искусству написания историй, которую я нашел. На первый взгляд кажется, что речь идет о сценарии, но уроки книги о структуре истории имеют более широкое применение.
- **«Искусство геймдизайна»** Джесси Шелла. Шелл — опытный геймдизайнер, который серьезно относится к играм, и это видно. Он затрагивает многие темы, которые я не рассматриваю, и его выводы и модели иногда отличаются от моих. Контраст — это пища для размышлений.
- **«Черный лебедь. Под знаком непредсказуемости»** Нассима Николаса Талеба. У Талеба другой взгляд на риск и случай-

ности. В своем фирменном, дерзком стиле он описывает идею события «Черный лебедь» — те невероятно важные и совершенно непредсказуемые события, которые управляют всем. Эта книга заставит вас усомниться в своей способности прогнозировать будущее.

- **«Логика неудачи»** Дитриха Дернера. Неудача — это не странное событие вроде удара молнии. Она медленно нарастает по собственной мрачной логике. Используя игры, которые имитируют сложные ситуации вроде африканских деревень и экологических заповедников, Дернер исследует то, как люди постоянно неправильно решают сложные проблемы, и подсказывает, как избавиться от укоренившихся вредных привычек. Это хороший старт для понимания проблем процесса разработки игр.
- **«Как работает мозг»** Стивена Пинкера. В этом произведении на 700 страниц звездный психолог Гарварда Стивен Пинкер описывает мышление в мельчайших деталях с точки зрения вычислительной техники и эволюции. Некоторые из его выводов вызвали немало противоречивых отзывов в научных кругах; все они заставляют задуматься.
- **«Переговоры без поражения. Гарвардский метод»** Роджера Фишера и Уильяма Ури. Большая часть настоящего геймдизайна связана с переговорами. Никого не волнует ваша способность анализировать, если вы не можете прийти к конструктивному компромиссу для реализации проекта.
- **«Теория игр. Искусство стратегического мышления в бизнесе и жизни»** Авинаша Диксита и Барри Дж. Нейлбаффа. Каждый геймдизайнер должен иметь базовое представление о математических концепциях теории игр, таких как преобладающие стратегии и равновесия Нэша. В этой книге вы найдете подробное и доступное объяснение этих понятий.
- **«Выдающиеся результаты. Талант ни при чем!»** Джеффа Колвина. Книга Колвина об обучении. Не важно, что вы изучаете — игру в гольф или игру на скрипке, программирование или геймдизайн. Колвин показывает, что ключом к лучшей эффективности является не талант или время, а целенаправленная

практика. Практика — это не просто выполнение какой-либо работы; это определенным образом стремление стать лучше. Эти высокоэффективные идеи и примеры будут ценными для тех, кто хочет усовершенствовать свои навыки геймдизайна.

- **«Властелины Doom. Как двое парней создали игровую индустрию и воспитали целое поколение геймеров»** Дэвида Кушнера. Эта книга о создании игры Doom в начале 1990-х. Но она не о мозге, она о сердце. Кушнер описывает команду разработчиков с легендарными уровнями драйва и страсти. В мире, где мы редко встречаем действительно необычных людей, подобные истории служат примерами невероятных возможностей.

Ответы на вопросы

Ответы на вопросы из главы 11:

Вопрос	Ответ
В каком году родился Архимед?	287 г. до н. э.
Сколько видов муравьев существует?	12 500
Какова была численность населения в 1900 году?	1,6 миллиарда
Каков диаметр Солнца?	1 392 684 км
Сколько лун у Сатурна?	53
При какой температуре плавится железо?	1538 °C
Каково количество погибших в Первой мировой войне?	9 911 000
Какова площадь Антарктиды?	14 000 000 км ²
На какой широте находится город Сантьяго, Чили?	33° 27 'S
Каков общий вес золота, добытого за всю историю человечества?	165 000 метрических тонн

Об авторе

Тайнан Сильвестр занимается разработкой игр с 2000 года. Работал над проектами всех типов — от индивидуальных проектов, разрабатываемых одним человеком, до студийных кассовых проектов категории AAA. Оставьте свой комментарий на его веб-сайте tynansylvester.com или отправьте письмо по адресу tynan.sylvester@gmail.com.