



С Л А В А   Г Р И С

# ОШИБКИ РАЗРАБОТЧИКОВ ВИДЕОИГР



ОТ ИДЕИ ДО ПРОВАЛА

Видеоигры: Глубокое погружение

Слава Грис

**Ошибки разработчиков  
видеоигр. От идеи до провала**

«Издательство АСТ»

2023

УДК 004.9  
ББК 32.973.26

**Грис С.**

Ошибки разработчиков видеоигр. От идеи до провала / С. Грис —  
«Издательство АСТ», 2023 — (Видеоигры: Глубокое погружение)

ISBN 978-5-17-157167-2

Как трезво оценивать свой проект? Что мешает грамотно планировать свою работу? Какие психологические особенности игроков стоит учитывать? Создание видеоигр — особое искусство. Инструмент разработчика — не компьютер, не игровые движки, а мозг. Книга «Ошибки разработчиков видеоигр. От идеи до провала» объяснит, как избежать самых распространенных ошибок, научит грамотно анализировать выбранное направление, четко оценивать свои силы и не опускать руки на пути к своей цели. Слава Грис — разработчик-одиночка, создатель Fearmonium, Catmaze и Reflection of Mine. Он прошел путь от фрилансера до организатора собственного видеоигрового издательства. Его первая книга «Сделай видеоигру один и не свихнись» стала очень популярна среди читателей. Будучи магистром психологии и куратором множества команд, делающих видеоигры, автор рассмотрит проблемы разработчиков с учетом особенностей человеческого мышления и расскажет о неочевидных способах воздействия на игрока. Эта книга будет полезна начинающим разработчикам, которые не хотят ничего упустить. В формате PDF A4 сохранён издательский дизайн.

УДК 004.9  
ББК 32.973.26

ISBN 978-5-17-157167-2

© Грис С., 2023

© Издательство АСТ, 2023

# Содержание

Ошибка 0	7
Ошибка 1	11
Ошибка 2	15
Ошибка 3	20
Конец ознакомительного фрагмента.	22

# Слава Грис

## Ошибки разработчиков видеоигр

### *От идеи до провала*

Дизайн серии Дмитрия Агапонова

Разработка серийного макета Елены Горячкиной

В оформлении переплета издания использована иллюстрация автора.

© Слава Грис, 2023

© ООО «Издательство АСТ», 2023

\* \* \*



## **Ошибка 0**

### **Жить в иллюзиях**

#### **Страшные цифры**

---

В 2022 году в Steam вышло более 12 тысяч игр, но только 500 из них собрали больше 1000 отзывов. Это означает, что лишь 4 % выпущенных проектов разошлись тиражом более 50 000 цифровых копий. Остальные 96 % смогли окупить разработку только в том случае, если создание игры обошлось авторам в крайне скромную сумму. Для менее экономных студий 2022 год ознаменовал полный крах. Даже если учесть, что часть этих игр вышла в конце года и их цифровые тиражи считать еще рано, эту приблизительную статистику едва ли можно назвать обнадеживающей. Не спасают положение и выходы на консоли с мобильными устройствами.

Взглянем правде в глаза: выпуск видеоигры в наше время в подавляющем большинстве случаев становится разорительной ошибкой. К провалу разработчиков приводит целый ряд досадных упущений и просчетов, которые и стали темой этой книги.

Не допускать ошибок, как известно, возможно только в процессе полного бездействия (но даже его можно рассмотреть как неверный способ распоряжаться своей жизнью). В ходе разработки видеоигр дров могут наломать как разработчики-одиночки, так и небольшие команды и даже огромные студии, вливающие миллионы долларов в проекты, которым всё равно суждено сгнить на задворках платформы Steam.

Окидывая взглядом индустрию или натываясь на невразумительные моменты в конкретных видеоиграх, так и хочется воскликнуть: «О чем они, черт возьми, думали? Как можно было такое допустить?» И дело далеко не всегда в ужасном игровом дизайне, плохом маркетинге или неумелом управлении персоналом. Проблема появления ошибок куда глубже и интереснее. Она кроется не только в небрежном отношении к невидимым нормам создания видеоигр. Мы допускаем ошибки из-за собственного замутнённого взгляда на действительность, ведь каждый из нас смотрит на мир через десятки и сотни когнитивных искажений.

#### **Чужие ошибки**

Когнитивным искажением можно назвать отклонение в мышлении и анализе информации в сторону полной иррациональности и нелогичности. На эту скользкую дорожку нас уведут стереотипы, убеждения, эмоции и, будем честны, простая глупость и необразованность.

Одно из самых важных искажений в контексте данной книги заключается в следующем: стороннее наблюдение за провалами коллег по цеху создает у наблюдателя иллюзию того, что он бы на их месте никогда не допустил такого недоразумения. Он бы играючи учел сотни факторов, ставших причиной провала коллег, и обязательно бы сделал всё лучше, чем получилось у них. Особенно легко выйти на такой ход мысли тогда, когда катастрофа уже случилась.

Под влиянием этого искажения написаны сотни комментариев в интернете под новостями о провалившихся играх, а также тысячи негативных отзывов к откровенно плохим проектам, причем обычно авторы таких отзывов убеждены, что сделать всё «хорошо и правильно» было не такой уж сложной задачей.

Мысленно учесть все нюансы и тонкости в каждой ситуации на самом деле невозможно: «оперативная память» человеческого мозга весьма ограничена. Принимая решение или фор-

мируя оценочное суждение, человек способен распоряжаться лишь приблизительно пятью фактами одновременно. Восприятие такого сложного продукта, как видеоигра, состоит из сотен факторов. Мысленно прикинуть, как именно изменение одного из них повлияет на все остальные, абсолютно невозможно.

## Свои ошибки

Распространенная убежденность в том, что обучаться на собственных ошибках – это самый действенный способ приобрести нужные навыки и опыт, еще сильнее изгибает и без того кривую призму, через которую мы воспринимаем объективную реальность: результаты множества исследований доказывают, что человек чертовски хорош в оправданиях. У нас буквально ничто на этом свете не получается так же хорошо, как врать окружающим (да и самим себе) об истинных причинах наших провалов и ошибок в суждениях.

При этом делать выводы из собственных ошибок кажется чем-то логичным и само собой разумеющимся, однако уже упомянутые когнитивные искажения мешают нам и в этом светлом начинании. Если мы и сделаем какие-то выводы из собственных неудач, то они всё равно с огромной долей вероятности не будут иметь никакого отношения к действительности.

Очень показательный эксперимент, доказывающий удивительную способность человека игнорировать свои прошлые упущения, американский психолог Гарри Бахрик провел на студентах университета Огайо. Учащихся попросили вспомнить свои финальные школьные оценки по нескольким предметам. Студенты знали, что их ответы будут сверять с их настоящими аттестатами, и врать не было никакого смысла.

Правильно свои оценки назвали лишь 70 % респондентов, но это не особо интересные данные: оценки можно было просто забыть. Интересно другое.

Студенты помнили о 89 % своих лучших оценок («А») – и лишь о 29 % своих худших оценок («D»). Их мозгу не нравилось находиться в черепной коробке неудачника, и он попросту вычеркнул из памяти часть данных, характеризующих своего носителя как тупого неуча.

На своих ошибках учатся, говорите? Свои ошибки как раз забывают или интерпретируют иначе, и только в очень редких случаях делают из них правильные выводы, для принятия которых нужен кристально чистый ум и понимание странных особенностей нашего сознания.

Когда публичное лицо внезапно меняет свою точку зрения и озвучивает совсем иные убеждения, якобы противоречащие предыдущим высказываниям, комментаторами в интернете такое поведение саркастично называется «переобучанием» и обыкновенно строго порицается. Но на деле мы все – мастера «переобучания» и отлично умеем оправдывать свои ошибочные суждения.

Это доказывают результаты одного длительного эксперимента, в ходе которого американских старшеклассников сначала опросили на тему того, нужно ли развозить чернокожих и белых ребят по домам на разных автобусах, а чуть позже разделили их на группы, в каждую из которых включили подставного участника, вооруженного неоспоримыми аргументами в пользу одного из двух взглядов на эту проблему. Практически в каждой группе школьники склонились к тому мнению, которое они приняли совместно с другими членами группы. Их первоначальные ответы потеряли всякое значение.

Третья стадия эксперимента была самой интересной: учеников, поменявших свою точку зрения, ткнули носом в то, что мнение, высказанное ими в индивидуальном порядке, отличалось от мнения, которое они защищали в составе группы.

Никто из школьников не признавал, что после убеждений подставного лица он начал думать иначе. Все поголовно утверждали, что они и раньше думали точно так же, как думают сейчас. Они предоставляли испытателям такое многообразие причин, по которым их перво-



начальный ответ отличался от итогового, что собрать их в какую-то статистику просто невозможно.

Брали ли они? Нет, все школьники действительно были твердо убеждены, что они не «переобувались».

Наш мозг, отчаянно цепляющийся за свою идентичность, крайне редко позволяет нам признать, что мы ошибались. Вместо этого он будет скрывать в нашей памяти часть данных, коверкать воспоминания и интерпретации. Мозг будет нашептывать и придумывать нам оправдание за оправданием: «Мы чего-то не учли, потому что это было невозможно в тот момент», «Нас подвели коллеги», «У нас не было необходимого опыта», «А вот если бы обстоятельства сложились иначе...» и т. д.

Я слышал массу причин и отговорок от людей, чьи игры настиг сокрушительный провал. После выслушивания множества подобных монологов мне стало стыдно за то, что я и сам частенько адресую себе такие же неуклюжие оправдательные формулировки, вместо того чтобы признать очевидную истину: меня зовут Слава Грис, и я ошибался тысячу раз.

Я вхожу в 96 % провалившихся разработчиков. Я выжил только потому, что разработка моих игр не опустошала мои карманы. Мне есть на что жить не благодаря тому, что я делаю, а вопреки.

## Самообман

Однажды мне написал один разработчик и спросил, возможно ли на Construct 2 сделать «кадр батлер». Я вежливо ответил, что не знаю, что такое «кадр батлер», после чего он заметил свою опечатку и уточнил, что он имел в виду жанр «кард-батлер», к которому относится, например, Hearthstone.

Допустил ли он ошибку? Очевидно. Но дело было в выходные, и я не отвечал на его сообщение достаточно долго, чтобы он успел обидеться. Тогда он оскорбил меня и вдогонку саркастично пожелал мне «удачи с английским». Иными словами, спустя какое-то время его мозг интерпретировал его собственную весьма очевидную ошибку таким образом, что виноватым в нашем недопонимании оказался я, обладающий якобы недостаточными знаниями английского, чтобы понять его достаточно невнятную речь.

Дурак ли он? Нет, мы все такие. И я в том числе. Мои игры не стали суперхитами из-за моих собственных ошибок. А мой мозг, не желающий находиться в черепной коробке неудачника, постоянно мне твердит, что я занял ту нишу, которую желал: «Я ведь никогда не хотел быть разработчиком популярных игр! Я же сам не люблю популярные игры! Я борюсь со злом, а не примыкаю к нему!»

Часть ошибок мой мозг, может, и признает, но посчитает в них виноватым кого угодно, кроме себя. Он сочтет, что я попросту не располагал теми знаниями и опытом, которые помогли бы делать более популярные игры, а значит, находился в среде, в которой разработать суперхит было попросту невозможно. Ему абсолютно плевать, что часть успешных разработчиков тоже ничего толком не умела в самом начале своего пути к шумному дебюту в Steam.

Врет ли мне собственный мозг? Ошибается ли? Я не знаю. Во мне засела убежденность в том, что я не сделал хита по той причине, что даже не пытался понравиться абсолютному большинству игроков. Я пытался встать рядом со своими собственными любимыми независимыми играми: *Chronicles of Teddy*, *Cineris Somnia*, *Aggelos*... Слышали о них? Вот о моей *Fearmonium*, наверное, тоже не слышали.

В этой книге мы попробуем разобраться, через какие искажения вы смотрите на действительность. А в дополнение к этому мы под лупой рассмотрим кочующие из игры в игру элементы, от которых игроки уже давно плюются, а разработчики продолжают использовать

их словно «по инерции», не утруждая себя рассуждениями о необходимости этих решений в видеоиграх.

Но сначала ответим на небольшой, но важный вопрос: зачем вам вообще слушать человека, который не стоит за созданием общепризнанных шедевров игровой индустрии? Чему может научить тот, кто не вошел в 4 % успешных счастливыхчиков?

# Ошибка 1

## Смотреть только вверх

### Мировые хиты

---

Разработке видеоигр посвящено множество книг. Вы можете почитать о том, как Джон Ромеро и Джон Кармак создавали Doom, ставший поистине монументальной работой в нашей индустрии и породивший целую волну видеоигр, отчаянно пытавшихся уподобиться оригиналу. Уже давно опубликованы дневники Джордана Мечнера, стоявшего за созданием гениальной Prince of Persia, с грандиозным качеством которой не может сравниться большинство современных платформеров. В вашем распоряжении литература про World of Warcraft, вселенная которой стала настоящим вторым домом для сотен тысяч игроков по всему миру. Я не буду пытаться перекричать этих талантливых людей и обесценить их гений, но и молчать не собираюсь.

Их опыт, безусловно, ценен и важен, а произведения про талантливых игровых дизайнеров и истории успеха целых студий действительно помогают сформировать более точное представление об игровой индустрии. На такой истоптанной тропе, как разработка видеоигр, очень важно уметь перенимать и проецировать на себя чужой опыт, а для этого необходимо учитывать, какие факторы, нюансы и обстановка влияли на ваш пример для подражания.

Возможно ли учесть все тонкости, способствующие созданию суперхитов, о которых написаны десятки книг? Можно ли было в свое время предположить, что выход симулятора фермера Stardew Valley на рынок, где уже давно существует множество частей Harvest Moon, обеспечит разработчика славой и успехом? А иной раз даже колоссальные бюджеты и опыт команды не гарантируют игре огромное количество продаж: ярким примером служит высокобюджетный проект Babylon's Fall от студии Platinum Games, стоявшей за такими хитами, как Bayonetta и NieR: Automata. Ни талант членов огромной команды, ни колоссальные средства, потраченные на разработку и маркетинг, не помогли выпустить очередной успешный продукт – Babylon's Fall с треском провалился.

Если бы существовала некая формула создания успешной игры, то наша индустрия выглядела бы совсем иначе и в свет не вышел бы такой спорный продукт, как Nosgoth, за авторством которого стояли опытные разработчики из Psyonix (авторы Rocket League) при поддержке Square Enix. Серверы их игры в жанре МОБА уже давно закрыты, а сам проект принес компании сокрушительные убытки. Очередная часть некогда популярной Gungrave не собрала бы всего пару сотен отзывов, большинство из которых были отрицательными, а Anthem не растерял бы так быстро интерес игроков.

Если в поисках применимой к нашим реалиям «формулы успеха» смотреть только наверх, мы заблудимся в лабиринте из неучтенных фактов, устаревших порядков и нераскрытых деталей. Из этой путаницы, включающей в себя и долю недосказанности, и непредсказуемую реакцию игроков, успешно не выбирался никто.

Джон Ромеро, один из авторов Doom, – определенно гений, но даже если он телепатически передаст вам абсолютно все свои знания и свой уникальный метод мышления, который помог ему разработать суперхит в начале 90-х в США, применить его навыки в 2020-х годах в России у вас едва ли получится. Более того: спустя тридцать лет с момента выхода игры

ни один из последовавших проектов Ромеро так и не приблизился по популярности к Doom. Получается, весь этот опыт и багаж знаний не помогает в наши дни даже ему самому.

## Нишевые игры

На момент написания этой книги я сделал три игры: Reflection of Mine, Catmaze и Fearmonium. В разработке я не отвечал только за музыку – рисовал и программировал всё самостоятельно. Есть случаи, когда созданные аналогичным образом произведения становились хитами – Undertale, Stardew Valley, Cave Story и т. д. Мои работы такая завидная доля обошла стороной. Я не пытаюсь снискать себе лавры выдающегося разработчика.

Но создание видеоигр всё равно стало моим основным источником дохода. Я продолжу ими заниматься, покуда мне хватает сил, потому что, выпустив на рынок свой первый продукт, я с удивлением открыл для себя существование ниши видеоигр, в которую мало кто метит, но многие попадают. Находится она между успехом и провалом.

Громкие разговоры о популярных проектах лишь подливают масла в огонь классического когнитивного искажения, подразумевающего категоричное разделение вещей и явлений на абсолютные категории: что-то может быть только очень хорошим или очень плохим, очень успешным или с треском провалившимся, невероятно выдающимся или абсолютно непримечательным.

Яркий пример такого искажения наблюдается на сайтах, где у пользователей есть возможность ставить оценки играм, фильмам или продуктам. Самыми популярными оценками станут 10, 9 и 0. Что-то среднее – 5 или 6 – игроки выбирают очень редко, потому что многие из нас мыслят абсолютными понятиями, с трудом признавая существование серого цвета в черно-белой картине мира. Оценка 5 не воспринимается как средняя.

Взглянуть на окружающую действительность без искажений сложнее, чем кажется. Абсолютно естественно, что многие люди мыслят таким образом. Я и сам излишне категоричен во многих своих суждениях.

Оттого мой первый доход с видеоигр стал для меня откровением: я не смог купить себе яхту, но и переезжать жить в коробку под мостом мне тоже не пришлось. Я просто получил весьма среднюю по меркам моего города зарплату, и она позволила мне спокойно уделить полтора года разработке следующей игры.

Всё это время я «смотрел наверх», что привело к двум проблемам: во-первых, я пытался скопировать программу действий, абсолютно неприменимую к той действительности, в которой я на тот момент находился. Эдмунд Макмиллен (автор SuperMeat Boy) вполне мог оставаться моим вдохновителем, но вот конкретный пример для подражания мне стоило подыскать скорее в пределах собственных реалий: я был не первым, кто выпускал свою нишевую игру в Steam в 2017 году, и кто-то уже преодолевал те трудности, с которыми столкнулся я. И это был не Эдмунд. У него были свои трудности, свое время и свои площадки. Например, он обрел известность на платформе Newgrounds, которая в наши дни едва ли позволит набрать аудиторию и заявить о себе достаточно громко. Его опыт работы с Newgrounds в контексте наших дней не имеет никакой ценности.

Вторым последствием «взгляда наверх» стали весьма абстрактные ожидания от выпуска игры. Если не знать, каких последствий мы ожидаем от своих действий, эти последствия в любом случае нас разочаруют. Эмоциональная составляющая разработки так же важна, как и наши технические и художественные навыки. Разочарование, провоцирующее подавленность и выгорание, явно не способствует тому, чтобы оказать своему проекту достойную поддержку после выхода в свет, а именно поддержка, обновления и терпеливое общение даже с самыми капризными игроками определяет продолжительность жизненного цикла вашего проекта.

Не имея в радиусе своего познания примера разработчиков, равных со мной по умениям и опыту, я не знал, чего ждать от выпуска игры и какая сумма определит меня как «провалившегося разработчика», а какая поставит на один уровень с Эдмундом.

К уровню Эдмунда я, разумеется, даже не приблизился. Моя первая игра – *Reflection of Mine* – «блистает» кучей ошибок как в вопросе продвижения, так и в игровом дизайне. Если предположить, что вы так и не узнаете или просто не учтете всего, что я описываю на последующих страницах, можете взять за ориентир доход от моего первого продукта: на протяжении первых полутора лет в среднем игра приносила \$800–900 в месяц. О том, сколько она заработала за весь свой жизненный цикл, я расскажу чуть позже.

Я смог рассчитать свой годовой доход в первые же дни после выпуска игры, потому что со мной поделились формулой приблизительного расчета долгосрочной выручки: сумма, заработанная на первой неделе продаж в Steam и умноженная на пять, приблизительно обозначит ваш доход за год. Таким образом, я прикинул, что на эти не особо-то и большие деньги я успею сделать вторую игру, в разработке которой уже смогу обойти часть разбросанных на моем пути граблей и выпустить таким образом нечто более успешное.

Так появился *Catmaze*, при разработке которого, я, если честно, тоже наломал дров, потому что смотрел не на те проекты и прислушивался не к тем людям: мое представление о мире тоже сформировано из заблуждений, и одним из них был «эффект авторитета». Существование этого искажения является еще одной причиной, по которой людям так сложно отрывать свой взор от общепризнанных гениев и ставить под сомнение их слова и решения.

## Авторитеты

Почему мы смотрим только наверх? Дело в том, что наш мозг весьма ленив. Он стремится как можно скорее сделать выводы и разделить всю получаемую информацию на «правильную» и «неправильную», а помогают ему в этом стереотипы. Они подталкивают нас к тому, чтобы придавать больший вес словам выдающихся и значимых людей. Уроки и утверждения всем известного разработчика Кодзимы воспринимаются нами как более значимые, нежели нравоучения того, кто провалился.

Дизайнерские решения в видеоиграх тоже можно относить к своеобразным урокам от мастеров и слепо верить в то, что каждый отдельно взятый элемент в популярной игре – это своеобразное утверждение автора на тему того, как нужно делать видеоигры.

Пример искаженного мышления может выглядеть так: «Если признанные гении решили, что открытые миры – это весело, то я буду делать игру с открытым миром». Провальные же игры, в свою очередь, могут восприниматься как набор абсолютно неправильных решений, ни одно из которых не является жизнеспособным: «Я видел множество невзрачных и провалившихся платформеров, а значит, делать платформер – это плохая идея».

Эффект авторитета является искажением по той причине, что слова, действия и решения авторитетной персоны чаще всего воспринимаются нами вне всякого контекста.

Мы можем очень сильно ошибиться, выбирая себе пример для подражания, который существует совсем в других реалиях. Мы едва ли сможем учесть все факторы, которые сделали решения авторитетных персон такими изящными и правильными. Мы можем подсмотреть у Кодзимы некий элемент из его видеоигры и слепо перенести его в наш собственный продукт, состоящий из совсем других идей и создаваемый в совсем других условиях. Разумеется, этот элемент может просто перестать работать.

Решение Кодзимы заставить игрока ходить по пустошам на протяжении тридцати часов в *Death Stranding* может произвести положительное впечатление на игрока только в совокупности с тысячей других элементов: музыкой, системой управления, внешним видом локаций, мотивацией героя, степенью сопереживания протагонисту и т. д. Без стройного оркестра игро-

вых элементов и без нужного контекста механика доставки посылок станет просто утомительной.

Разработчики, добившиеся авторитета, в подавляющем большинстве случаев стоят за созданием игр, разработка которых обошлась в миллионы долларов. Если смотреть на эту закономерность через «эффект авторитета», то единственным верным способом создать игру может показаться бездумное вливание в нее кучи денег. Многие знакомые мне разработчики действительно погрязали в долгах и даже не приблизились к тому, чтобы окупить затраты на свои проекты.

Если же иногда все-таки опускать глаза, то мы увидим вполне достойные игры, созданные без огромных финансовых затрат и способные при этом прокормить своих авторов, а иной раз и обеспечить их достаточным количеством средств на создание более крупнобюджетных проектов.

Статистика говорит о том, что хитом становится, в среднем, четвертый проект студии. Не стоит метить сразу на вершины хит-парадов, потому что если кто-то и умудрился туда попасть с первым же своим проектом, то следующий разработчик окажется там только со своей седьмой игрой. Сколько игр мне еще предстоит сделать, чтобы взгляд на меня оказался «взглядом наверх»? Я не знаю. Всё, что я могу, – это продолжать заниматься разработкой.

---

**Подсматривая решение или в игровом дизайне, или просто в ведении дел, ориентируйтесь не только на авторитет вашего идейного вдохновителя, но и на условия, в которых это решение приводило к положительным результатам. Возможно, в ваших реалиях и в контексте вашей узкой задачи это решение будет не столь удачным.**

## Ошибка 2

### Собрать команду неучей

#### Зачем нужна команда

---

Путь разработчика-одиночки кажется непреодолимым лабиринтом из миллиона задач, которые предстоит решить, и миллиона навыков, которыми предстоит овладеть. Видеоигра является комплексным произведением и требует недюжинных умений во множестве сфер: в программировании, в рисовании, в написании музыки и сценариев... Весьма естественно испытать леденящий душу ужас даже при беглом взгляде на перечень необходимых умений.

Причиной этого ужаса является не только неправильная оценка собственных сил, но и отягощающая наше существование черта человеческого мозга: он боится за нашу идентичность. Обрастая навыками, умениями и начиная работу в другой сфере, мы преобразуемся как личность и формируем вокруг себя новую среду. Мозгу в незнакомой среде будет некомфортно, ему станет гораздо сложнее просчитывать риски. Если при текущем положении дел у нас всё хорошо и относительно стабильно – есть жилье, какие-то деньги и работа, – то вместо мотивации создавать видеоигры наш мозг даст нам только страх и ступор. У него нет причин менять среду существования. У него и так всё нормально. Это вашему эго не сидится на месте. Оно стремится к большему, а вот мозг в сытом теле не так отягощен желанием самовыражаться и развиваться, а мотивацию вам выдать должен именно он. Страдающее эго тут ни при чем.

Работа в команде подразумевает меньшее количество изменений в нашей личности и необходимость приобретения якобы меньшего количества навыков, потому идея собраться с другими в команду кажется мозгу не такой уж и плохой. Мотивацию на это благое дело он выдаст с большей вероятностью.

Проще ведь прыгать в мутный омут разработки видеоигр, разделяя с кем-нибудь ответственность за потенциальный провал, да? Удар о твердое дно принесет не такое внушительное количество боли: виноватыми в неудачном приземлении окажетесь не только вы. А может, вы вообще будете не виноваты, и провал получится свалить целиком на кого-то из членов вашей команды: например, из-за программиста игрокам не понравился отклик от действий персонажа вашей игры, из-за художника вы привлекли к проекту слишком мало внимания, а по вине плохого сценариста игроки не прониклись сюжетом и не полюбили ваших героев.

Вы можете даже не осознавать часть причин, которые подталкивают нас к объединению в группу, но где-то в вашем бессознательном могут существовать и еще более грустные и лицемерные причины стать частью стаи. Не стоит думать, что ко всем нашим решениям нас подталкивает сухая логика и рациональное сознание. Чаще всего логика и сознание как раз стоят в стороне именно в те моменты, когда мы совершаем тот или иной выбор. Важно отдавать себе точный и полный отчет в том, почему конкретно вы выбираете работу в одиночку или работу в команде. Содержание этого отчета будет зависеть от ваших личностных особенностей и убеждений, и составить его за вас у меня, увы, не получится.

В данном произведении я не агитирую бросаться в пучину соло-разработки. Напротив, я готов поддержать тех, кто решил вступить на наш тернистый путь в компании коллег и сейчас выбирает один из двух вариантов: найти таких же пылких и воодушевленных ребят, готовых работать на честном слове, а потом весело разделить с ними прибыль от будущего хита, или же отложить немного денег и нанять людей, которым можно делегировать обязанности по

созданию графики, написанию кода или музыки. О варианте с инвесторами и издателями мы подробно поговорим в следующих главах.

## **Работа с новичками**

Первый вариант, подразумевающий работу с друзьями на честном слове, является наиболее популярным и предпочтительным у новичков, незнакомых со статистикой, раскрывающей страшную истину: подавляющее большинство таких начинаний заканчивается провалом. Собравшиеся вместе друзья, не имеющие опыта в разработке, едва ли доведут проект до ума, потратив больше времени на бесконечную переписку в «рабочем» чате, нежели на разработку.

Безусловно, можно найти примеры, когда сплоченная ответственностью и обещаниями группа друзей выпускала успешные игры на скопленные средства, скинутые в «общак» или просто расходуемые на жизнь в период разработки. Один из самых ярких кандидатов на образец для подражания на этом пути – это студия Four Quarters, прославившаяся за счет своей последней игры Loop Hero, изданной Devolver Digital (теми же издателями, которые стояли за выпуском Hotline Miami). Свою первую игру – Please, Don't Touch Anything – они делали в свободное от основной работы время без финансирования извне. Только с ее выпуском они смогли с головой погрузиться в разработку игр, существуя на доход с первого коммерческого проекта.

Подробно вдаваясь в анализ деятельности этой студии, важно отметить, что их первая игра вышла из-под руки всё же не совсем уж юных и неопытных авторов, никогда раньше не работавших вместе. Узнать свои слабые и сильные стороны им помогло участие в игровых джемах, на которых они набили массу шишек и из которых извлекли множество полезных уроков. Их проекты зарождались именно на джемах, потому что расширить маленькую игру гораздо проще, чем справиться с необузданным роем идей о создании очередного «убийцы GTA», с которого вы в итоге будете стряхивать механику за механикой.

Если вы твердо решили продержаться на плаву бравым отрядом друзей, вам нужно научиться справляться с несколькими искажениями нашего мышления, которые могут стать причиной провала. Отсутствие этих искажений вполне могло сыграть роль в успехе тех немногочисленных команд-новичков, у которых получилось вывести свой продукт на рынок, в том числе и у Four Quarters.

## **Цикл ошибок**

В работе почти каждого из нас можно наблюдать явление, которое носит название эффекта Даннинга – Крюгера. Это искажение представляет собой любопытную петлю из ошибочных суждений, в которую можно угодить, если взяться за какое-либо дело без достаточного багажа знаний.

Человек, обладающий низкой компетенцией в вопросе, например, разработки видеоигр, по понятным причинам способен принимать неудачные решения и делать ошибочные выводы. Но что иронично, всё те же низкая компетенция и дефицит знаний не позволяют ему воспринять свои решения как неудачные и трезво взглянуть на проделанную работу.

Эффект Даннинга – Крюгера заключается в том, что низкая компетенция выступает причиной принятия неверных решений и она же потом не позволяет трактовать эти решения как «неверные». Этот эффект и не дает нам учитывать часть совершенных ошибок.

Я не самый рукастый парень на свете. Вместо забивания гвоздей я иной раз отчаянно бью молотком мимо шляпки, а сами гвозди у меня просто гнутся, вместо того чтобы войти в стену. Если собрать команду таких же криворуких неучей, как я, поместить нас в информационный вакуум и заставить построить дом, то спустя какое-то время мы будем ошалевшими глазами



лицезреть перекошенную полуторазтажную халупу, болезненно наклоненную на бок. Поймем ли мы, что наше строение – это уродливая пародия на шалаш, а не настоящий дом? Возможно. Поймем ли мы, что конкретно нужно сделать, чтобы привести наше убогое произведение в человеческий вид? Нет, не поймем: наш уровень компетенции по ходу сгибания гвоздей если и вырастет, то незначительно. Мы так и останемся неучами, игнорирующими многовековой опыт постройки домов. Мы лишь продолжим бултыхаться в соку собственной некомпетентности.

В начале своего пути разработчика видеоигр я работал в команде. Мы с товарищами втроем потратили шесть месяцев на создание едва играбельного прототипа очень скучной и непонятной игры. Равнодушие игроков, отсутствие заинтересованности блогеров и общий дефицит восторга масс от нашего произведения вводили нас в ступор. Мы оказались неспособны определиться с тем, что конкретно нужно поменять в нашем проекте, чтобы он расцвел новыми красками и принес нам всемирную известность.

Вместо этого мы окрестили себя непризнанными гениями и сели делать вторую игру. Так зловещее колесо сансары пошло на второй круг: мы ошибались из-за низкой компетентности и за счет нее же не могли исправить свои ошибки. Вторая игра оказалась чуть лучше, но всё еще была чудовищно плохой: мы еще не осмыслили тот опыт, которого набрались и который помог нам слегка улучшить навыки.

Более того, эффект Даннинга – Крюгера определяет низкоквалифицированных людей как более уверенных в своих действиях. Мы работали над двумя так и не вышедшими играми с непоколебимой убежденностью в том, что разрабатываем суперхиты. О многих вещах, которые определяли наши идеи как «чудовищные», мы даже не задумывались. Мы не подвергали сомнению большинство своих решений, потому что нам не хватало на это ни знаний, ни опыта, ни, будем честны, мозгов.

Высокое мнение о своих способностях при полном отсутствии необходимых знаний и опыта – это как раз та ловушка, в которую попадает большинство начинающих команд. Эффект Даннинга – Крюгера чудесно объясняет, почему новички часто берутся в рамках своего первого проекта за реализацию идеи по созданию нелинейной RPG в открытом мире. Человеку достаточной компетенции идея сесть за разработку столь комплексного проекта уже не кажется такой светлой. Тем более, следуя всё тому же эффекту, компетентные люди свои возможности обыкновенно, наоборот, недооценивают.

Именно эффект Даннинга – Крюгера объясняет такую внушительную диаспору деятельных неучей и бездействующих гениев.

## Экспертная оценка

Не подумайте, что я подвергаю людей жесткой сегрегации по признаку интеллектуальных способностей. Истина в том, что гений в одном ремесле обязательно окажется неучем в другом. То, что я компетентен в вопросе разработки видеоигр, не делает меня отличным строителем, а отличного строителя не делает идиотом неспособность работать с игровыми движками. Мы все обладаем очень низким уровнем компетенции в большинстве существующих сфер труда.

Спрыгнуть с этой карусели ошибок и неправильных решений можно несколькими способами. Первый из них потребует меньше усилий, поэтому наверняка многим он покажется более соблазнительным. Я говорю об *экспертной оценке*.

Игровая индустрия состоит из огромного количества людей, среди которых обязательно найдется специалист, способный указать вам на те места, где ваш проект трещит по швам. Опытные разработчики не являются труднодостижимыми богами, восседающими на далеком Олимпе из признания, денег и славы. Компетентные в разработке видеоигр люди ничем не отличаются от тех, кто отлично разбирается в какой-нибудь другой области, – каждый из нас

что-то умеет лучше, чем большинство других. Уж в забивании гвоздей вы меня точно превосходите. Но это же не делает вас гордым зазнайкой, верно?

Разработка видеоигры – это такая же работа с гвоздями. Только интереснее.

Секрет получения хорошей экспертной оценки кроется там же, где и секрет получения хорошей критики, – в деньгах. Никто не будет тратить свое время бесплатно, а если даже и согласится на столь щедрый жест, то внимание, которое вы получите, окажется незначительным по сравнению с реальными возможностями вашего будущего наставника. Кураторские и менторские системы подразумевают почасовую оплату консультаций специалиста, но поверьте: даже час трезвого анализа вашей игры специалистом способен поменять ваше отношение к проекту, команде и к идее делать игры в целом.

Иной раз платить кураторам можете даже не вы: многие образовательные курсы предлагают не только набор уроков с домашними заданиями, а еще и систему наставничества. Организаторы курса сами найдут для вас знающего специалиста, и оплата его труда войдет в стоимость курса.

Опытный строитель тут же объяснит, почему наша несчастная изба накренилась, и посоветует, каким образом ее выровнять, в то время как комментаторы в интернете будут и дальше бездумно строчить оскорбления в адрес неучей, построивших этот ужас. Они не укажут нам на те способы, которые могли бы действительно помочь решить проблему. А если и укажут, то бегло и поверхностно, без учета всего контекста ситуации.

Найти опытного разработчика, готового заниматься кураторством, можно или в специализированных пабликах по разработке видеоигр в «ВК», или в телеграм-каналах, посвященных той же теме, – например, «Ступени Геймдева». Также никто не мешает вам лично обратиться к уважаемому вами разработчику с просьбой уделить вам время за пару золотых. Кто-то, быть может, согласится и бесплатно, если проект его заинтересует и если вы будете доносить свои просьбы понятно и вежливо.

Основная цель таких консультаций – не критика. Критиковать некомпетентного специалиста нет никакого смысла. В детстве у меня не получалось вырезать снежинки из бумаги: я просто ее мял и гнул, отчаянно вопрошая учительницу, что не так с моими ножницами. Она нервно отвечала мне, что я «неправильно их держу», и на этом замолкала. Помогал ли мне этот ответ резать бумагу? Нет – я продолжал ее мять. Я всё еще понятия не имел, каким именно образом мне нужно держать ножницы. Я был уверен, что это инструменты у меня кривые, бумага сделана из резины, а учительница – дура. Я никоим образом не ставил под сомнение собственные навыки и в итоге, к слову, до сих пор ненавижу что-то вырезать.

Критика полезна лишь тогда, когда у нас уже есть хотя бы минимальные умения и знания. Новичкам же нужно показать направление и объяснить, почему принятые ими решения и используемые ими методы ошибочны. Вам нужно услышать не то, что ваша игра плохая, а то, по какой причине она кажется плохой.

Если бы мне шесть лет назад сказали, что в моем рисунке есть проблемы с композицией, и на этом бы закончили формулировку своих претензий, то я просто пожал бы плечами и не сделал из этого замечания никаких выводов. Мне было бы необходимо выслушать более подробные объяснения и попросить критика ткнуть меня носом в конкретные элементы, акцент на которых был излишним либо недостаточным. Сейчас же, когда я знаком с построением композиций и обладаю хоть какими-то сведениями по этому вопросу, подобное лаконичное замечание заставит меня уже самостоятельно перебрать свои собственные познания и прикинуть, где же я всё-таки неправильно расставил акценты и правду ли говорит критик.

При наличии опыта эффект Даннинга – Крюгера не даст о себе знать. Критик должен в первую очередь ориентироваться на развитие компетенции того, кому он помогает. Неопытному сценаристу видеоигр нет смысла говорить, что его персонажи плоские и неинтересные. Ему нужно рассказать про существование архетипов, про то, какие черты могут сделать его

героя более запоминающимся, и только после того, как компетенция критикуемого будет расширена новыми знаниями, указывать на конкретные ошибки.

Расширение компетенций – это необходимая составляющая в улучшении навыков и приобретении опыта.

В ходе бесед с куратором не забывайте и про эффект авторитета: основная задача во время беседы с наставником – это получить новые знания, обратить внимание на недочеты в вашем проекте и выслушать рекомендации по их устранению. Не нужно воспринимать каждое слово как инструкцию к действию, надиктованную вам высшими силами. С кем бы вы ни говорили, мнение останется всего лишь мнением: в разработке видеоигр не существует универсально-правильных решений тех или иных проблем, что встанут на вашем пути. Ментор тоже может предложить вам отнюдь не самый эффективный вариант преодоления трудностей. Обязательно попробуйте его, но не рассчитывайте на максимально эффективный результат.

Одно только знание о многообразии способов, которыми вы можете реализовать задуманное, значительно повышает вашу собственную компетенцию. Простое выслушивание советов от действительно опытных людей уже сделает вас умнее и позволит взглянуть на проблемы под другим углом или же просто обнаружить слабые места в своем проекте на самых начальных этапах.

## Наблюдательность

Второй способ избавиться от эффекта Даннинга – Крюгера заключается в том, чтобы более внимательно анализировать поведение разработчиков при работе в команде. Я достаточно долго подрабатывал фриланс-художником в различных игровых проектах, но ни один из них так и не увидел свет. У меня заказывали графику на сумму, превышавшую мой тогдашний годовой доход на основной работе. Однако проекты всё равно откладывались авторами в долгий ящик из-за массы ошибок: кому-то в итоге не хватало денег – ошибка планирования давала о себе знать; кто-то, уже оплатив всю графику, понимал, что его проект – это сущее недоразумение и окупиться он никогда не сможет (качественная игра будет хороша и увлекательна даже на заглушках и кубиках, а графика и прочие элементы лишь расширят тот спектр эмоций, которые игрок от нее получит).

Без внимательного анализа того опыта, который я приобрел в ходе работы фрилансером, я едва ли довел бы до ума *Reflection of Mine*. Я обрстал компетенциями, анализируя последствия решений, принятых игровыми дизайнерами и программистами. Мой низкий уровень вовлеченности в эти проекты помогал смотреть на них гораздо трезвее, чем на продукт, разрабатываемый в собственной команде. Мне было по большому счету всё равно, выстрелит игра или нет, и страсть к этим проектам не затмевала мне глаза – я относился к ним с равнодушием и скепсисом.

---

**Нарабатывая опыт, наибольшее внимание уделяйте расширению своих компетенций. Хороший результат в любом деле всегда достигается комбинацией навыков и знаний. Если вырабатывать только навык, постоянно делая одно и то же и не обращаясь за подсказками к более опытным специалистам, то никакого прогресса вам не видать. Ровно как не видать его и при бездумном накоплении знаний без их дальнейшего использования. Если вы узнали что-то новое в ходе сотрудничества с другими людьми – тут же применяйте это знание, преобразуя его в навык. Но с чего же начать?**

## **Ошибка 3**

### **Брезговать вареньем**

#### **Опыт без работодателя**

---

Хорошей страховкой, позволяющей не угодить в замкнутый круг проб и ошибок при работе в команде, является наличие в ней хотя бы одного участника, обладающего базовым набором компетенций. Тогда разработка пойдет по принципу «Один делает – остальные помогают» и иерархия в проекте выстроится более строгая, нежели в обычной компании ничего не умеющих друзей. От горизонтальной иерархии всё перейдет к вертикальной.

Есть два способа заполучить в команду компетентного человека. Первый: на специализированных форумах можно найти уже готовую команду разработчиков и впитывать информацию, подключившись к их работе. Недостатком этого метода является то, что с нулевым багажом знаний вас вряд ли возьмет на работу тот, у кого уже есть опыт по выпуску игр, оттого первый шаг в этом направлении является и первым шагом в реализации второго способа – самому стать опытным лидером. В обоих случаях вам нужно этого опыта где-нибудь набираться, и набираться придется самостоятельно.

Новички часто видят перед собой непреодолимую преграду в виде замкнутого круга: чтобы устроиться на работу, нужен опыт работы, а чтобы набраться опыта, нужно устроиться на работу. Благо, в нашей сфере всё устроено немного иначе. Для приобретения опыта вам не нужны какие-то особые условия, которые может предоставить только работодатель. Если у вас есть компьютер с выходом в сеть – дорога к получению опыта открыта и все условия для работы у вас уже есть.

При любой удобной возможности я рекомендую начинающим разработчикам принимать участие в игровых джемах – массовых мероприятиях, где несколько команд (или соло-разработчиков) за ограниченный период времени создают видеоигру на заданную тему. Разумеется, разработанные на джемах продукты едва ли можно назвать полноценными играми, готовыми к выпуску в Steam, но в нашем случае многого и не надо: основной целью в начале пути является приобретение навыков планирования, получение способности правильно оценивать собственные силы и резвое обучение в стрессовых условиях, обеспеченных ограниченным временем.

#### **Базовые навыки**

Набраться совсем базовых умений, которые позволят приподнять планку собственных компетенций до участия в джеме, можно, бездумно повторяя за разработчиками игр уроки на YouTube: их там развелось великое множество, и освоить любой игровой движок на базовом уровне в наши дни не составляет труда. Это вопрос одной-двух недель, по прошествии которых вы сможете реализовать несколько простеньких механик, достаточных для того, чтобы разработать крошечную игру для джема.

Если иллюзия насчет того, что научиться работать в движке – задача непосильная, никак не может развеяться, вспомните, что я говорил о сохранении собственной идентичности: это ваш мозг не хочет менять условия своего существования и допускать изменений в вашей личности. Работу нужно вести не в сторону того, чтобы на одной силе воле изучать на YouTube

урок за уроком, а в сторону формирования убеждений о том, что разработка видеоигр откроет вам двери к более привлекательной карьере и к самореализации.

Я всегда рекомендую приглядеться к такому онлайн-мероприятию, как Ludum Dare, проводимому несколько раз в год. С нашей текущей задачей он поможет справиться на все сто благодаря интересной системе получения отзывов к проектам: чем больше вы комментируете работы других ребят, тем больше комментариев к своей игре получаете сами.

Таким образом, посвятив по прошествии джема пару деньков изучению проектов своих конкурентов и их активному комментированию, вы сможете получить целый вагон отзывов от разработчиков видеоигр. Они будут рассуждать на тему того, что в вашей игре хорошо, а что плохо. Разумеется, некоторые из отзывов будут оставлены начинающими неучами, которые преследуют те же цели, что и вы, но это никак не умаляет ценности других комментариев. Безусловно, они не будут в достаточной мере развернуты и так же ценны, как мнение куратора или ментора, но для начала это лучше, чем ничего.

## Графическое оформление

Участие в джемах может показаться недоступной роскошью тем, кто не обременен хотя бы базовыми навыками в рисовании. Будем честны, в наши дни работу в движке на уровне, требуемом для джема, можно освоить за несколько недель, тогда как освоение изобразительного искусства для получения приятной картинки может занять несколько лет.

Но игры для джемов представляют собой некоммерческие продукты, а это значит, что вы можете использовать почти любые материалы, найденные на просторах интернета. Работа с уже готовыми графическими ассетами мало того что окажется простой и быстрой, так еще и научит вас обращаться с графикой в принципе. Как платные, так и бесплатные ассеты вы можете найти на таких ресурсах, как unity asset store, envato, itch.io, 3dmodelhaven, unreal marketplace, kenney, opengameart. А если вы захотите использовать графику из уже вышедших хитов прошедших лет, то добро пожаловать на [spriters-resource](#) или [spritedatabase](#).

Одна из немногих причин радоваться тому, что на дворе уже не 2007 год, заключается в том, что в наши дни в моде достаточно лаконичный дизайн. Вы, возможно, вспомните стиль, носящий название Frutiger Aero (рис. 1), знакомый по оформлению устаревшего плеера Winamp и теме Windows Aero. Этот стиль состоял из объемных элементов с градиентами и полупрозрачностью. Шаг влево, в сторону лаконичности, сделал бы его совсем холодным, безликим и даже пугающим, а шаг вправо превратил бы его в нагромождение излишне сложных структур. В наши дни, несмотря на всю трудность исполнения, он воспринимается устаревшим и безвкусным, и у большинства современных пользователей найдет отклик более лаконичное графическое исполнение. Учитывая, что графика – это одна из самых дорогих сторон разработки видеоигры, можно считать, что современные реалии идут нам навстречу.

## **Конец ознакомительного фрагмента.**

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.