

Оглавление

| | |
|---|----|
| Предисловие от издательства | 10 |
| Об авторе | 11 |
| О рецензентах | 12 |
| Благодарности | 13 |
| Предисловие | 14 |
| Для кого предназначена эта книга | 14 |
| О чем эта книга расскажет..... | 14 |
| Как извлечь из книги максимум | 15 |
| Загрузка примеров кода | 15 |
| Загрузите цветные изображения | 16 |
| Условные обозначения..... | 16 |
| Отзывы..... | 16 |
| Часть I. Введение в разработку на Roblox | 17 |
| Глава 1. Введение в разработку на Roblox | 18 |
| Технические требования..... | 18 |
| Преимущества разработки на Roblox..... | 19 |
| Финансовые возможности в Roblox | 19 |
| Развитие профессиональных навыков | 20 |
| Преимущества совместной работы..... | 20 |
| Типы разработчиков | 21 |
| Программисты..... | 21 |
| Моделисты | 22 |
| Дизайнеры уровней | 22 |
| Дизайнеры UI/UX | 22 |
| Чего ожидать от ранних проектов..... | 23 |
| Резюме..... | 24 |
| Глава 2. Знакомство с рабочей средой | 26 |
| Технические требования..... | 26 |
| Страница создания игры..... | 26 |
| Настройка параметров игры и плейса..... | 27 |
| Меню Configure Game | 28 |
| Меню Configure Start Place | 29 |
| Другие настройки плейса | 31 |
| Библиотека и магазин аватаров..... | 33 |
| Начало работы с Roblox Studio..... | 35 |
| Меню File и настройки | 35 |
| Движение и управление камерой | 37 |
| Панель Explorer | 38 |
| Работа с инструментами в Studio..... | 40 |

| | |
|--|----|
| Настройки в меню Game Settings | 41 |
| Вкладка View | 43 |
| Вкладка Test | 44 |
| Настройка Studio для облегчения рабочего процесса | 46 |
| Использование ресурсов Roblox | 47 |
| Учебники и ресурсы | 47 |
| Форум разработчиков | 47 |
| Резюме | 48 |
| Часть II. Программирование в Roblox | 49 |
| Глава 3. Введение в язык Roblox Lua | 50 |
| Технические требования | 50 |
| Создание переменных и условные операторы | 51 |
| Типы данных | 51 |
| Определение переменных и работа с ними | 54 |
| Числа | 54 |
| Логические типы | 55 |
| Строки | 55 |
| Таблицы | 57 |
| Словари | 59 |
| Векторы | 61 |
| CFrame | 62 |
| Экземпляры | 66 |
| Условные операторы | 66 |
| Объявление и использование циклов | 70 |
| Циклы for | 70 |
| Цикл while | 72 |
| Цикл repeat | 73 |
| Функции и события | 74 |
| Функции в программировании | 74 |
| Рекурсия | 77 |
| События и методы экземпляров | 80 |
| Стиль программирования и эффективность кода | 81 |
| Общие правила стиля | 82 |
| Специфичные для Roblox принципы | 82 |
| Резюме | 83 |
| Глава 4. Сценарии программирования в Roblox | 84 |
| Технические требования | 84 |
| Основы модели «клиент–сервер» | 84 |
| Различные типы сценариев | 85 |
| Вкладка Script Menu | 87 |
| Фильтрация | 89 |
| Удаленные события | 89 |
| Удаленные функции | 91 |
| Привязываемые функции и события | 92 |

| | |
|---|------------|
| Работа с сервисами Roblox | 93 |
| Сервис Players | 93 |
| Сервисы ReplicatedStorage и ServerStorage | 96 |
| Сервис StarterGui | 96 |
| Сервисы StarterPack и StarterPlayer | 97 |
| Сервис PhysicsService | 98 |
| Сервис UserInputService | 99 |
| Работа с физикой | 100 |
| Ограничения | 100 |
| Перемещение тел | 102 |
| Добавление второстепенных аспектов игры | 105 |
| Звук | 106 |
| Освещение | 108 |
| Другие эффекты | 109 |
| Резюме | 111 |
| Глава 5. Пишем обби-игру | 112 |
| Технические требования | 112 |
| Настройка серверной части | 112 |
| Управление данными игрока | 114 |
| Обработка троттинга и граничных случаев | 119 |
| Управление столкновениями и персонажами игроков | 122 |
| Создание этапов для обби-игры | 123 |
| Создание поведения объекта | 124 |
| Создание наград | 129 |
| Магазины и покупки | 131 |
| Премиальные покупки за робаксы | 131 |
| Создание магазинов с внутриигровой валютой | 137 |
| Борьба с эксплойтами | 139 |
| Настройка внешнего интерфейса (фронтенд) | 140 |
| Создание эффектов | 140 |
| Тестирование и публикация | 144 |
| Резюме | 145 |
| Глава 6. Создание игры в жанре «Королевская битва» | 147 |
| Технические требования | 147 |
| Настройка серверной части | 147 |
| Управление данными игрока | 148 |
| Настройка системы раундов | 149 |
| Создание оружия | 155 |
| Локальная репликация | 167 |
| Спаун лута | 169 |
| Настройка внешнего интерфейса | 172 |
| Работа с пользовательским интерфейсом | 172 |
| Создание магазина | 182 |
| Резюме | 182 |

| | |
|---|------------|
| Часть III. Логистика производства игр..... | 183 |
| Глава 7. Три «М»..... | 184 |
| Технические требования..... | 184 |
| Механика..... | 184 |
| Симуляторы..... | 185 |
| RP-игры..... | 186 |
| Тайкуны..... | 186 |
| Мини-игры..... | 187 |
| Монетизация..... | 187 |
| Маркетинг..... | 189 |
| Система продвижения Roblox..... | 189 |
| Ютуберы..... | 191 |
| Проверим, чему вы научились..... | 192 |
| Резюме..... | 193 |
| Предметный указатель..... | 194 |

Об авторе

Зандер Брамбо – программист, менеджер проектов и разработчик игр. В игры, которые были созданы им самим или при его участии, играло более 200 млн человек. Зандер активно взаимодействует с сообществом Roblox и командой по связям с разработчиками Roblox, содействуя в продвижении и поддерживая платформу Roblox. Он является автором популярных игр, таких как **Power Simulator**, **Munching Masters** и **Magical Simulator**. Благодаря своим наработкам в создании игр он окончил колледж, а в данный момент учится в Вашингтонском университете и является частью Школы компьютерных наук и инженерии Пола Г. Аллена. На момент написания этой книги Зандеру 18 лет.

О рецензентах

Эндрю Береза – человек с глубоким стратегическим мышлением и специалист по решению проблем, обладающий особым умением выявлять закономерности и связи в абстрактных и запутанных системах. Он занимается разработкой игр на всех уровнях, от творческого руководства и дизайна уровней до пользовательского интерфейса и программирования. Является признанным экспертом по платформе Roblox с более чем 10-летним опытом разработки и создал 3-месячный акселератор (2017 г.) и 5-месячный инкубатор (2018 г.), размещенный в корпоративной штаб-квартире Roblox в Сан-Матео, Калифорния. На платформе Roblox к числу его игр относятся **2PGFT, Miner’s Haven, Azure Mines, War Games, Vesteria** и **Build Island**. Еще до Roblox он много работал с играми, создавая читы для картриджей Super Mario World и моды для Minecraft.

Ян Ханф, также известный на платформе Roblox как **Hanfian**, занимается созданием игр с 2012 года. Он учился в колледже Тихоокеанского университета и за 5 лет получил степень магистра компьютерных наук. Основным направлением его работы были графика и симуляции. В 2019 году Ян работал над популярной в Roblox игрой Anime Fighting Simulator, которая завоевала международный успех. В настоящее время Ян проживает в Калифорнии и борется с раком мозга четвертой стадии, но надеется продолжить делать игры в будущем.

Предисловие

По мере чтения этой книги вы получите практический опыт работы на платформе Roblox. Мы начнем с обзора разработки Roblox, а затем научимся работать с Roblox Studio. По мере продвижения вы узнаете все необходимое: от программирования в Roblox Lua до создания игр в жанре обби и «Королевская битва». Ближе к концу мы рассмотрим вопросы разработки игр, поговорим о том, как достигнуть максимальной популярности игры путем внедрения хорошей механики, монетизации и маркетинговых методов.

После изучения этого руководства Roblox у вас появятся навыки, необходимые для работы в команде или даже для руководства над командой, и вы сможете создавать игровые миры из своих фантазий и позволите игрокам по всему миру окупнуться в них.

Для кого предназначена эта книга

Эта книга предназначена для всех, кто интересуется разработкой игр на платформе Roblox, а также для тех, кто уже знаком с Roblox и хочет изучить лучшие рекомендации, приемы и практики для разработки в Roblox.

О чем эта книга расскажет

В *главе 1* мы познакомимся с основными концепциями разработки Roblox, возможностями разработки на этой платформе, с тем, как зарабатывать деньги на играх и чего стоит ожидать с учетом опыта проектов других разработчиков.

В *главе 2* мы будем учиться работать в Roblox Studio. Рассмотрим основные элементы управления, такие как движение и управление камерой, взаимодействие с экземплярами в рабочем пространстве, как использовать бесплатные ресурсы и редактировать информацию об игре.

Для изучения *главы 3* вам не понадобится знание языков программирования. Вы узнаете, как программировать в Roblox Lua, начиная с простых инструкций типа `print("Hello, world")` и заканчивая универсальными конструкциями программирования.

В *главе 4* акцент делается на сценариях программирования, специфичных для Roblox, с которыми люди, даже обладающие общими знаниями в области программирования, незнакомы.

В *главе 5* мы применим все уже приобретенные знания и создадим простую, но рабочую и полноценную игру. В качестве жанра выберем обби, и для создания игры вам нужно будет использовать переменные, события, функции и свойства.

В главе 6 вам нужно будет собрать воедино все, чему вы научились, чтобы создать игру в жанре «Королевская битва». Вам нужно будет воспользоваться всем, что вы узнали из книги, и одновременно с этим вы получите новые знания в области безопасности и организации. К концу этой главы вы прочувствуете весь процесс создания игры с нуля.

Глава 7 – это наиболее обширная часть книги, посвященная вопросам, не связанным непосредственно с программированием. В этой главе основное внимание будет уделено расширению тех навыков, которые позволят вам стать не только программистом, но и специалистом по продвижению игр, для чего мы воспользуемся тремя «М»: Механикой, Монетизацией и Маркетингом.

Как извлечь из книги максимум

Для выполнения упражнений из этой книги вам потребуется следующее аппаратное и программное обеспечение:

| Аппаратное и программное обеспечение, рассматриваемое в книге | Необходимая операционная система |
|---|----------------------------------|
| Roblox Studio | Windows, macOS, Chrome OS |
| Roblox Player | Windows, macOS, Chrome OS |

Полный список системных требований для всех компонентов Roblox можно найти по следующей ссылке: <https://en.help.roblox.com/hc/en-us/articles/203312800>.

Если вы используете электронную версию этой книги, мы советуем вам набирать код примеров самостоятельно или взять его из нашего репозитория GitHub (ссылка доступна в следующем разделе). Это поможет вам избежать возможных ошибок, связанных с копированием и вставкой кода.

Загрузка примеров кода

Вы можете загрузить примеры кода для этой книги на сайте GitHub по адресу <https://github.com/PacktPublishing/Coding-Roblox-Games-Made-Easy>. В случае обновления кода вы увидите изменения в файлах в репозитории GitHub.

Чтобы скачать файлы кода, выполните следующие действия:

- 1) перейдите по указанной ссылке на сайте GitHub;
- 2) нажмите кнопку **Code** и выберите пункт **Download ZIP**.

Загрузив файл архива, распакуйте его, используя последнюю версию следующего ПО:

- WinRAR/7-Zip для Windows;
- Zipeg/iZip/UnRarX для Mac;
- 7-Zip/PeaZip для Linux.

Загрузите цветные изображения

Мы также предоставляем файл PDF с цветными изображениями скриншотов/диаграмм, использованных в этой книге. Вы можете скачать его по ссылке: https://static.packt-cdn.com/downloads/9781800561991_ColorImages.pdf.

Цветные изображения также можно найти здесь: <https://github.com/PacktPublishing/Coding-Roblox-Games-Made-Easy/tree/main/Color%20Images>

Условные обозначения

В этой книге используется несколько специальных обозначений.

Моноширинный шрифт: фрагменты кода в тексте, имена таблиц базы данных, имена папок и файлов, расширения файлов, пути, URL-адреса, пользовательский ввод и ссылки Twitter. Пример: «Когда игрок присоединяется к игре, его данные нужно добавить в словарь `sessionData`».

Блок кода оформляется следующим образом:

```
dataMod.removeSessionData = function(player)
    local key = player.UserId
    sessionData[key] = nil
end
```

Полужирный текст: обозначает новый термин, важное слово или слова, которые вы видите на экране. Это могут быть, например, элементы меню или фразы в диалоговых окнах. Например: «Чтобы создать пользовательский интерфейс для игры, перейдите к службе **StarterGui** на панели **Explorer**».

Советы или важные примечания

...выглядят вот так.

Отзывы

Пожалуйста, оставьте отзыв о книге. Почему бы не оставить отзыв на сайте, где вы приобрели книгу, после ее прочтения? Благодаря этому потенциальные читатели смогут увидеть ваше непредвзятое мнение и использовать его для принятия будущего решения о покупке, мы в издательстве сможем понять, что вы думаете о наших продуктах, а наши авторы увидят ваши отзывы о своем творении. Спасибо!

Часть



Введение в разработку на Roblox

В этой части мы познакомимся с основными концепциями разработки на Roblox. Затем изучим принципы работы в ПО Roblox Studio, его возможности в разработке и увидим, чего стоит ожидать от первых проектов.

Эта часть состоит из следующих глав:

- *глава 1 «Введение в разработку на Roblox»;*
- *глава 2 «Знакомство с рабочей средой».*

Глава 1

Введение в разработку на Roblox

Roblox – это крупнейшая развлекательная платформа, не имеющая аналогов как для игроков, так и для создателей игр. На Roblox заходит более 100 млн активных пользователей в месяц, что позволяет как новым, так и опытным разработчикам создавать успешные игры, в которые можно играть по всему миру бесплатно. Благодаря множеству ресурсов и огромному дружелюбному сообществу разработчиков со всего мира, с которыми можно связаться, выход в сферу разработки игр Roblox – это возможность, которую нельзя найти больше нигде.

Перед тем как окунуться в технические детали работы с Roblox Studio и программирования на Lua, нам сперва нужно ознакомиться с тем, что вообще может дать вам разработка игр на Roblox и из чего обычно состоит повседневное взаимодействие разработчика с платформой. К концу этой главы у вас будет более полное представление о различных типах ролей, которые разработчики могут выполнять в игровых проектах, о том, как становление разработчиком Roblox может принести вам финансовую выгоду и чего ожидать после разработки и публикации своих первых игр.

В этой главе мы рассмотрим следующие основные темы:

- узнаем о преимуществах разработки на Roblox;
- выделим несколько типов разработчиков;
- рассмотрим перспективы первых проектов.

Технические требования

Для изучения этого раздела вам не потребуется никакого программного обеспечения или дополнительных материалов, так как в основном здесь будет информация и теория. Возможно, вам будет полезно иметь под рукой компьютер с доступом в интернет, чтобы вы могли зайти на некоторые сайты или в приложения, обсуждаемые в данной главе.

Преимущества разработки на Roblox

Платформа Roblox непрерывно развивается и растет с момента ее создания в 2006 году, и в последние годы этот рост значительно ускорился. По состоянию на 2020 год в Roblox ежемесячно играют более 100 млн уникальных пользователей. Сейчас – лучший момент, чтобы стать разработчиком Roblox, так как множество новых игроков ищут новые игры от новых авторов.

Финансовые возможности в Roblox

Вероятно, одним из самых больших факторов, способствующих общему росту популярности Roblox среди разработчиков, является возможная финансовая выгода. Может быть, именно это и есть один из ваших основных мотивов к изучению данной платформы, и в этом нет ничего удивительного.

Сейчас самые популярные игры на платформе Roblox ежегодно приносят разработчикам десятки миллионов долларов США на внутриигровых покупках.

Как разработчик вы можете зарабатывать деньги на играх с помощью процесса под названием **Developer Exchange**, или, если кратко, **DevEx**. В Roblox есть виртуальная валюта под названием **робаксы**, которую можно покупать за реальные деньги. Когда игрок покупает робаксы, баланс его учетной записи пополняется, и он может свободно тратить валюту на любую игру, которую захочет. Когда игрок тратит робаксы на одну из внутриигровых покупок, 70 % робаксов от этой покупки получает разработчик игры, а Roblox берет 30%-ную комиссию.

Выделим два типа внутриигровых покупок:

- **игровые пропуски, или гейм-пассы:** разовые покупки;
- **продукты для разработчиков:** покупаются несколько раз за что-то вроде внутриигровой валюты.

Вы должны накопить в общей сложности 100 000 робаксов и быть старше 13 лет, чтобы иметь право на участие в программе DevEx. В дополнение к продажам, которые вы делаете напрямую, поклонники ваших игр, у которых есть премиальная подписка Roblox, приносят вам дополнительные робаксы в зависимости от того, сколько времени они проводят в вашей игре, хотя эта сумма обычно составляет лишь часть дохода. Кроме того, есть некоторые закрытые программы, в которые можно попасть только по приглашению и которые позволяют вам продавать за робаксы аксессуары для персонажей, плагины и многое другое. Однако скоро эти программы планируется сделать общедоступными.

Работая в команде разработчиков в группе Roblox, разработчики могут получать доход в робаксах напрямую либо через процент от дохода от игры. Прямые выплаты организуются просто, и их можно настроить на странице

Configure Group, но платежи не регулярны. Выплата разработчикам процента от дохода от игры выполняется автоматически после проверки. Важно помнить, что такая выплата возможна только в том случае, если игра размещена в группе, а не в личном профиле.

Важная заметка

После того как игрок совершает внутриигровую покупку, Roblox выдерживает 3-дневный период ожидания, прежде чем эти робаксы будут перечислены в групповой или личный аккаунт, чтобы убедиться, что покупка была законной.

Больше информации о программе обмена разработчиками можно найти здесь: <https://www.roblox.com/developer-exchange/help>.

Развитие профессиональных навыков

Помимо финансовой прибыли, разработчик Roblox получает уникальную возможность развивать другие свои навыки, которые могут оказаться полезными в иных профессиональных средах. Независимо от того, выполняете ли вы роль менеджера проекта или обычного программиста, у вас будут развиваться навыки работы в команде и общения. Одно из самых востребованных качеств, которое ищут работодатели, особенно в STEM, где большая часть работы выполняется в команде, – это способность координировать свою работу и общаться внутри команды. Разработка на Roblox, на мой взгляд, является одним из лучших мест для студентов, изучающих информатику, где можно начать осваивать принципы совместной работы. Платформа помогает развить не только способности в программировании, но и лидерские качества и, в конечном итоге, навыки управления финансами.

Преимущества совместной работы

В целом работать с другими людьми не всегда обязательно, и существуют популярные игры, созданные в одиночку. Но в Roblox сама система настоятельно побуждает разработчиков работать над играми вместе, при этом каждый разработчик имеет в проекте одну или несколько ролей. В настоящее время почти все топовые игры в разделе **Popular** были созданы командой из двух или более человек. Благодаря большому успеху и последующему росту команды разработчиков некоторых игр выросли до более чем 20 человек, хотя это пока редкое явление.

Лучший способ присоединиться к сообществу и найти других разработчиков для сотрудничества – это Twitter или Discord. Имея в Twitter аккаунт, посвященный вашей разработке, вы можете публиковать свои лучшие творения и общаться с другими, более популярными разработчиками. Новые

связи с другими разработчиками в **Roblox Twitter Community (RTC)** могут дать вам возможность поработать с более известными людьми и набрать популярность. **Discord** – это приложение для общения, похожее на **Slack**. В **Discord** есть множество серверов, ориентированных на разработку в **Roblox**, и на них вы можете поделиться своими трудами, обсудить работу с другими членами сообщества или найти новых людей, с которыми можно объединиться.

Не менее важную роль в сообществе разработчиков играют ютуберы, создающие контент по **Roblox**. Когда пользователи **YouTube** создают контент, демонстрирующий ваши игры, это привлекает аудиторию, и количество ваших игроков, скорее всего, увеличится благодаря такой рекламе. Контакт с такими людьми может обеспечить продвижение ваших проектов в будущем, а также найти единомышленников. С создателями контента не всегда просто связаться, но они часто сидят в **Discord** или **Twitter**, а также отвечают на запросы по электронной почте.

В целом описанный здесь тип сетевого взаимодействия одинаков во многих сферах деятельности. Создание имиджа, популярности и репутации играет ключевую роль в вашей карьере. Законное и профессиональное выполнение вашей работы тоже приносит плоды, но не столь быстро. А коммуникативные способности позволят вам легче общаться с новыми людьми, которые могут принести пользу как вашей работе, так и репутации.

Типы разработчиков

Как упоминалось ранее, сообщество разработчиков **Roblox** весьма разнообразно. Каждый разработчик привносит на платформу свой уникальный стиль и технику. В разработке игр есть множество ролей, которые может выполнять каждый член команды. Чаще всего в **Roblox** у каждого разработчика имеется один основной навык, в рамках которого он работает в проекте. Наиболее распространенными типами разработчиков в **Roblox** являются программисты, дизайнеры уровней, разработчики 3D-моделей, дизайнеры UI/UX (интерфейса) и различные художники. Каждый разработчик, выполняющий свою роль, одинаково важен для создания качественного готового продукта.

Программисты

Программисты создают ядро любой игры. Программист реализует все: от хранения данных игрока и создания работающего оружия до любых более сложных и редких игровых функций. В **Roblox** программисты используют язык **Lua**, быстрый процедурный язык на основе **C**, который обычно применяется в индустрии разработки игр. Вы увидите, что синтаксис языка **Lua** проще других языков и более ориентирован на человека, чем большинство языков высокого уровня. Из-за этого многие программисты

считают кривую обучения довольно плавной, и если вы начинаете изучать программирование с Lua, в будущем переход на другие языки будет прост, поскольку в нем используется общий синтаксис из нескольких языков программирования.

Моделисты

3D-моделисты создают все, что вы видите в игре: от мебели, домашних животных и предметов до любых других визуальных элементов разного размера. Модели можно создавать прямо в Roblox Studio, но большинство 3D-моделей создают в бесплатном приложении под названием Blender. Тому есть много причин, в частности детали, используемые для создания моделей внутри Roblox Studio, блочные, с их помощью не удастся получить гладкие или сложные формы, а в специальном ПО для моделирования это делается легко.

Дизайнеры уровней

Дизайнеры уровней создают миры ваших игр. Будь то холодный коридор в дрейфующем космическом корабле или жаркая засушливая пустыня с далеким оазисом – именно дизайнеры уровней создают первое впечатление об игре, когда игроки присоединяются к ней.

Важно, чтобы дизайнер уровней проекта был достаточно опытным, чтобы он умел отразить желаемое видение в игре. Может показаться, что дизайнеры уровней и моделисты – это одно и то же, но это не так. Дизайнеры уровней больше сосредоточиваются на общей карте и большом мире для проектов, что, впрочем, не мешает им использовать Blender и создавать 3D-модели отдельных ресурсов.

Дизайнеры UI/UX

Дизайнеры UI/UX (или дизайнеры интерфейса) создают экраны, с которыми взаимодействуют игроки внутри вашей игры. Например, дизайнер пользовательского интерфейса создает экран инвентаря игрока, полосу здоровья и другие видимые элементы интерфейса.

Зачастую UI – это первое, что игрок увидит в вашей игре, поэтому важно, чтобы дизайнер мог создать визуально привлекательный интерфейс, соответствующий стилю самой игры.

Если и другие типы разработчиков – аниматоры, музыкальные продюсеры, графические дизайнеры и иные художники. Все перечисленные типы разработчиков важны для создания сильного, законченного продукта, и для успеха проекта важно, чтобы все они были одинаково компетентны в своей сфере. Вы должны определить, какие из предыдущих ролей вас интересуют больше всего, чтобы затем узнать о них больше и развить соответствующие навыки для будущего использования.

Чего ожидать от ранних проектов

Начало карьеры разработчика у всех разное, но уверенно можно сказать одно: ваш первый игровой проект не будет номером один на странице популярных игр, и это нормально! Как бы обидно это ни звучало, золотые горы не свалятся на вас легко и просто. Мой личный опыт подсказывает, что поддерживать популярную игру, в которую одновременно играют десятки тысяч игроков (или больше), довольно сложно. Новым разработчикам сперва нужно набраться опыта и привыкнуть к платформе, и лишь потом начать создавать популярные игры. На рис. 1.1 изображена миниатюра моей первой игры под названием **Endure**. Здесь невооруженным глазом видно, что игра сделана любителем.



Рис. 1.1. **Endure** была одной из моих первых игр, и она требовала явной доработки

Многие начинающие разработчики часто сталкиваются с препятствиями, когда начинают работу над проектом с огромным азартом, но слабым техническим пониманием, и впоследствии оказываются вынуждены отказаться от проекта, так как не удастся реализовать все задуманное. Лучший способ обойти эти проблемы – составить план разработки и задокументировать функции и механики, которые должны быть включены в ваш проект. В процессе вы можете пересматривать и реструктурировать свое видение по мере необходимости, учитывая, что популярно среди игроков Roblox и что вы вообще можете реализовать с имеющимися способностями команды.

Важнее всего, конечно, мотивация и преданность делу, ибо без них проект так и останется в столе. Важно помнить, что игры не всегда доставляют удовольствие игрокам, если их создание не приносит удовольствия разработчикам. Работая с другими людьми, убедитесь, что все в команде согласны с направлением и примерной дорожной картой разработки игры. Оптимальная цель как для вашего психического здоровья, так и для качества ваших проектов – делать каждую новую игру лучше, чем предыдущая. Вы можете потратить месяцы на разработку довольно посредственной игры, но это будет необходимая для будущей работы основа. Каждый новый проект дает вам немного больше опыта и признания. На рис. 1.2 показан прямой результат применения этих методов – мою игру **Power Simulator** запустили более 100 млн раз, а ее миниатюра гораздо более привлекательна для потенциальных игроков благодаря профессиональному дизайну.



Рис. 1.2. **Power Simulator** – это моя самая успешная на сегодняшний день игра и результат двухлетнего опыта

Какая бы судьба ни ожидала ваши ранние проекты, важно всегда смотреть в будущее и пытаться улучшить игровой опыт для ваших игроков и процесс разработки для себя. Только работой и стараниями можно достичь успеха.

Резюме

Прочитав эту главу, вы должны запомнить несколько ключевых моментов: какие возможности существуют на платформе Roblox, чем занимаются разные типы разработчиков и как найти в сообществе людей, с которыми можно будет создавать совместные проекты.

Теперь, когда вы узнали много нового о работе в сети, о том, чего ожидать от ранней разработки, а также о том, как начать развиваться как разработчик, я верю, что вы сумеете совершить удивительные вещи на платформе

Roblox и развить навыки, которые пригодятся вам в любой профессиональной среде.

В следующей главе мы начнем знакомиться с Roblox Studio, программой, в которой вы будете создавать свои игры. Знание всех особенностей вашей рабочей среды, в том числе тех, которые сперва могут быть незаметны, поможет повысить эффективность вашей разработки и общую производительность в будущих проектах.