

*Прие – моей опоре и источнику вдохновения.
Моим ученикам из Научной академии STEM Magnet,
которые стали моими первыми «подопытными»
и научили меня учить.*

– Крис Брэдфилд

Оглавление

Об авторе.....	11
О рецензенте.....	12
Предисловие	13
Глава 1. Введение в Godot 4.0	16
Общие рекомендации	16
Как учиться эффективно.....	17
Что такое игровой движок?.....	18
Что такое Godot?	19
Загрузка Godot	20
Альтернативные способы загрузки	21
Обзор интерфейса Godot.....	22
Диспетчер проектов.....	22
Окно редактора	24
Ноды и сцены.....	27
Создание скриптов в Godot.....	28
О языке GDScript	28
Итоги главы.....	29
Глава 2. Coin Dash – создание первой 2D-игры	30
Технические требования.....	31
Настройка проекта	31
Векторы и 2D-координатные системы	33
Часть 1 – Сцена игрока	35
Создание сцены.....	35
Анимация спрайтов	36
Форма столкновений	39
Скрипт поведения игрока.....	40
Управление движением игрока.....	41
Переключение анимации	43
Управление началом и завершением движения игрока	44
Подготовка к обработке столкновений	44
Часть 2 – Сцена монеты.....	47
Настройка нод	47
Скрипт монеты	48
Часть 3 – Главная сцена	49
Настройка нод	49
Скрипт главной сцены	50
Подготовка к новой игре	50
Часть 4 – Пользовательский интерфейс.....	52
Настройка нод	52
Настройка сообщений.....	53
Отображение счета и времени	54
Динамическое обновление интерфейса с помощью GDScript.....	55

Завершение игры	56
Подключение HUD к Main	56
Часть 5 – Завершающие штрихи	58
Визуальные эффекты	58
Звук	59
Усиления	59
Анимация монет	62
Препятствия	62
Итоги главы	64
Глава 3. Space Rocks – создание классической 2D-аркады с физикой	65
Технические требования	66
Настройка проекта	66
Физика твердых тел	67
Корабль игрока	69
Настройка тела и физики	70
Конечные автоматы	72
Управление кораблем игрока	74
Обработка выхода за границы игровой области	76
Реализация стрельбы	77
Создание астероидов	81
Настройка сцены	81
Астероиды разного размера	81
Создание экземпляров астероидов	82
Взрывающиеся астероиды	84
Создание пользовательского интерфейса	89
Компоновка	89
Скрипт пользовательского интерфейса	91
Код интерфейса главной сцены	93
Код игрока	94
Завершение игры	96
Обнаружение столкновений твердых тел	97
Пауза в игре	97
Противники	98
Движение по контуру	99
Сцена противника	101
Движение противника	101
Генерация противников	102
Стрельба и столкновения	103
Щит игрока	106
Звуковые и визуальные эффекты	109
Звуки и музыка	109
Частицы	110
След противника	113
Итоги главы	113
Глава 4. Jungle Jump – создание 2D-платформера с бегом и прыжками	115
Технические требования	116

Настройка проекта	116
Знакомство с кинематическими телами	118
Реакция на столкновение	118
Создание сцены игрока.....	119
Слои и маски столкновений	120
Об AnimationPlayer.....	120
Анимации	121
Форма столкновений	122
Завершение сцены игрока	123
Состояния игрока	123
Скрипт игрока	124
Движение игрока.....	125
Здоровье игрока	128
Предметы для сбора	129
Настройка сцены.....	129
Скрипт предметов.....	130
Проектирование уровня.....	131
Использование TileMap	132
Проектирование первого уровня	135
Добавление ловушек	137
Прокручивающийся фон	138
Добавление противников	141
Настройка сцены.....	141
Скрипт противника.....	142
Нанесение урона противнику	143
Скрипт игрока	144
Создание интерфейса.....	147
Настройка сцены.....	147
Скрипт HUD	148
Подключение HUD.....	149
Стартовый экран.....	150
Настройка сцены.....	150
Настройка главной сцены	151
Переход между уровнями	153
Сцена двери	153
Настройки экрана.....	155
Завершающие штрихи	155
Звуковые эффекты	155
Двойной прыжок.....	155
Частицы пыли.....	156
Лестницы	158
Движущиеся платформы	160
Итоги главы.....	162
Глава 5. 3D Minigolf – погружение в 3D и создание поля	
для мини-гольфа.....	163
Технические требования.....	163
Введение в 3D.....	164
Ориентация в трехмерном пространстве.....	164

3D-редактор Godot	165
Добавление 3D-объектов	166
Глобальное и локальное пространство	167
Трансформации	168
Меши	168
Камеры	170
Настройка проекта	172
Создание поля	173
Принцип работы GridMaps	173
Создание первой лунки	174
Окружение и освещение	175
Добавление лунки	176
Создание мяча	177
Тестирование мяча	177
Улучшение физики мяча	177
Создание интерфейса	179
Создание стрелки прицеливания	179
Реализация интерфейса	181
Создание скриптов	183
Скрипт интерфейса	183
Главный скрипт	184
Скрипт мяча	186
Тестирование	187
Улучшение системы прицеливания	187
Улучшение системы прицеливания (вариант 2)	188
Настройка камеры	190
Создание полноценного поля	193
Визуальные эффекты	194
Добавление материалов	195
Освещение и окружение	197
Итоги главы	199
Глава 6. Infinite Flyer	200
Технические требования	201
Настройка проекта	201
Настройка ввода	201
Сцена самолета	202
Формы столкновений	203
Скрипт самолета	204
Создание игрового мира	207
Объекты мира	207
Чанки	210
Главная сцена	214
Создание новых чанков	216
Усложнение	217
Столкновения	219
Топливо и счет	221
Стартовый экран	224
Звуковое оформление	225

Сохранение рекорда	225
О расположении файлов	226
Доступ к файлам	226
Дополнительные функции	228
Итоги главы	228
Глава 7. Дальнейшие шаги и дополнительные ресурсы	229
Работа со встроенной документацией Godot	229
Чтение документации API	231
Контроль версий и использование Git с Godot	232
Использование Blender с Godot	233
Подсказки для импорта	233
Работа с файлами .blend	234
Экспорт проектов	235
Шаблоны экспорта	236
Пресеты экспорта	236
Как экспортировать проект	237
Экспорт для различных платформ	237
Введение в шейдеры	238
Создание 2D-шейдеров	240
3D-шейдеры	245
Дополнительные материалы	247
Использование других языков программирования в Godot	247
C#	247
Другие языки и GDExtension	249
Помощь и ресурсы сообщества	250
Godot Recipes	251
Участие в развитии Godot	252
Участие в разработке движка	252
Написание документации	252
Финансовая поддержка	253
Итоги главы	253
Заключение	254
Предметный указатель	255

Об авторе



Крис Брэдфилд (Chris Bradfield) работает в сфере интернет-технологий более 25 лет. Он участвовал в разработке ряда успешных MMO- и социальных онлайн-игр для издателей в Южной Корее и США. За время карьеры в игровой индустрии он успел поработать гейм-дизайнером, разработчиком, продуктовым менеджером и руководителем по развитию бизнеса.

В 2012 году он увлекся преподаванием и основал KidsCanCode, где обучает программированию учащихся средних и старших классов. Кроме того, Крис создает видео- и текстовые учебные материалы для студентов, изучающих разработку игр по всему миру.

Выражаем благодарность Кенни Влэгелсу (Kenney Vleugels, @kenneyNL) за 3D-тайлы для мини-гольфа, skorpio – за арт космического корабля, Луису Зуно (Luis Zuno, @ansimuz) и авторам Sunny Land за их работу.

О рецензенте

В Индонезии живет и творит Исайя Джамиль (Isaiah Jamiel) – мастер своего дела, виртуозно владеющий технологиями .NET и Godot. Он руководит студией NTC Studios, где вместе с Изрой, Хансеном, Сачио, Шэннон, Деллой и Есикой стремится к успеху в игровой индустрии. Творчество Джамиля не ограничивается программированием: он также создает аранжировки, вдохновляющие традиционную индонезийскую музыку. Но важнее любых профессиональных увлечений для него остаются любовь к Валери и глубокая связь с семьей. Именно стремление сочетать инновации, творчество и теплые человеческие отношения и движет его впечатляющими работами – как в технологиях, так и в музыке.

Предисловие

Эта книга – введение в игровой движок Godot и его новейшую версию, Godot 4. Godot 4 предлагает огромное количество функций и возможностей, представляя собой серьезную альтернативу дорогим коммерческим решениям. Для начинающих это удобный и эффективный способ освоить основы разработки игр. Опытным же разработчикам Godot предлагает мощный, гибкий и настраиваемый инструмент для воплощения творческих идей.

В основе книги лежит проектный подход. Вы последовательно создадите пять игровых проектов, а дополнительные ресурсы помогут глубже разобраться в принципах создания игр в Godot.

Для кого эта книга

Эта книга предназначена для всех, кто хочет научиться создавать игры на современном игровом движке. Она станет полезным ресурсом как для начинающих, так и для опытных разработчиков. Предварительный опыт программирования приветствуется.

Содержание книги

Эта книга предлагает проектно-ориентированный подход к изучению Godot. Пять игровых проектов выстроены по принципу «от простого к сложному», где каждый следующий опирается на концепции, изученные в предыдущем.

Глава 1 «Введение в Godot 4.0». Знакомит с концепцией игровых движков в целом и Godot в частности. Здесь вы найдете инструкции по установке Godot, обзор интерфейса редактора и рекомендации по эффективной работе с книгой.

Глава 2 «Coin Dash – создание первой 2D-игры». Небольшой 2D-проект, в котором вы научитесь создавать сцены, работать с системой нод, ориентироваться в редакторе и писать первые скрипты на GDScript.

Глава 3 «Space Rocks – создание классической 2D-аркады с физикой». Посвящена работе с физическими телами и созданию космического шутера в стиле Asteroids.

Глава 4 «Jungle Jump – создание 2D-платформера с бегом и прыжками». Создание платформера с боковой прокруткой в духе Super Mario Bros. Вы познакомитесь с кинематическими телами, анимационными состояниями и дизайном уровней с помощью тайловых карт.

Глава 5 «3D Minigolf – погружение в 3D и создание поля для мини-гольфа». Переносит освоенные 2D-концепции в трехмерное пространство. Вы поработаете с 3D-мешами, освещением и управлением камерой.

Глава 6 «Infinite Flyer». Посвящена дальнейшему изучению 3D-разработки. Вы узнаете о создании динамического контента, процедурной генерации и других продвинутых техниках.

Глава 7 «Дальнейшие шаги и дополнительные ресурсы». Предлагает темы для самостоятельного изучения после прохождения основных проектов, а также полезные ссылки и советы для расширения ваших навыков.

Как извлечь из книги максимум пользы

Для лучшего понимания примеров кода желательно иметь общие представления о программировании. Идеально, если это будет опыт работы с современным языком с динамической типизацией, таким как Python или JavaScript. Если же вы только начинаете свой путь в программировании, будет полезно сначала изучить основы.

Godot запустится на большинстве современных компьютеров под управлением Windows, macOS или Linux.

Если вы читаете электронную версию книги, советуем набирать код вручную или использовать файлы из репозитория книги на GitHub (ссылка приведена в следующем разделе). Это поможет избежать возможных ошибок из-за копирования и вставки кода.

Исходный код проектов

Исходный код для проектов доступен в репозитории GitHub по адресу <https://github.com/PacktPublishing/Godot-4-Game-Development-Projects-Second-Edition>. Репозиторий поддерживается в актуальном состоянии.

Загрузка цветных изображений

Все скриншоты и схемы, использованные в книге, также доступны в виде отдельного цветного PDF-файла. Загрузить его можно по адресу: <https://packt.link/1Y2hq>.

Принятые обозначения

В книге используется ряд принятых текстовых соглашений.

Код в тексте обозначает элементы кода, имена таблиц баз данных, папок, файлов, расширения файлов, пути, примеры URL-адресов и ввод пользователя. Например: «В Godot 4 появилась возможность импортировать файлы .blend напрямую в проект».

Блоки кода оформляются так:

```
shader_type canvas_item;

void fragment() {
    // Place fragment code here.
}
```

Полужирным выделяются названия элементов интерфейса: кнопки, вкладки, свойства, пункты меню. Пример: «Первым свойством является **Shader** (Шейдер), где можно выбрать **New Shader** (Новый шейдер). При этом откроется панель **Create Shader** (Создание шейдера)».

Курсивом выделяются ключевые термины при первом упоминании и важные смысловые акценты. Пример: «Важным понятием является *конечный автомат (Finite State Machine – FSM)*».

Советы и важные примечания выделяются таким образом.

Глава 1

Введение в Godot 4.0

Разработка игр – увлекательное и благодарное дело, будь то профессиональная карьера или хобби. Сейчас идеальное время для старта: современные инструменты и языки программирования позволяют создавать качественные игры и распространять их по всему миру легче, чем когда-либо прежде. Если вы читаете эту книгу, значит, вы готовы сделать первый шаг к созданию игры (или игр) своей мечты.

Эта книга представляет собой введение в игровой движок Godot, а именно в его версию 4.0, выпущенную в 2023 году. Godot 4 предлагает множество нововведений, что делает его мощной альтернативой платным коммерческим движкам. Начинающим разработчикам он позволяет легко освоить фундаментальные концепции разработки игр, а опытным служит гибким, расширяемым и *открытым* инструментом для воплощения творческих идей.

В основе книги лежит проектный подход: вы освоите ключевые аспекты движка, создавая пять игровых проектов. Это поможет сформировать глубокое понимание как теории разработки игр, так и ее практического применения в Godot. Вы познакомитесь с архитектурой движка и освоите важные техники для ваших будущих проектов.

В этой главе мы рассмотрим следующие темы:

- общие рекомендации по работе с книгой;
- что такое игровой движок;
- знакомство с Godot;
- загрузка и установка Godot;
- обзор пользовательского интерфейса Godot;
- ноды и сцены;
- написание скриптов в Godot.

Общие рекомендации

В этом разделе собраны советы, основанные на педагогическом опыте автора. Имейте их в виду, работая с книгой, особенно если вы только начинаете свой путь в программировании.

Старайтесь выполнять проекты из книги в предложенном порядке. Содержание глав взаимосвязано: более поздние опираются на темы, рассмотренные в ранних. Если что-то забыли, просто вернитесь и повторите тему. Ваша главная цель – понять материал, а не поставить рекорд скорости.

Материала много, поэтому не отчаивайтесь, если что-то не дается сразу. Помните, что, как и в любом ремесле, для достижения мастерства нужны годы практики. Стать экспертом за одну ночь невозможно. Ключ к освоению сложных тем – повторение. Чем больше вы работаете с Godot, тем привычнее и понятнее он становится. Попробуйте перечитать одну из первых глав, когда закончите книгу. Вы удивитесь, насколько углубилось ваше понимание.

Если вы читаете электронную версию, не поддавайтесь искушению копировать код. Набирая его вручную, вы заставляете мозг работать активнее – подобно тому, как ведение конспекта на лекции помогает усвоить материал лучше, чем простое слушание, даже если вы никогда не вернетесь к этим записям. Для тех, кто печатает медленно, это также станет отличной тренировкой скорости. Коротко говоря, вы программист, поэтому привыкайте набирать код.

Одна из самых частых ошибок начинающих – браться за проекты, которые им не по силам. На старте очень важно ограничивать масштаб. Гораздо полезнее и результативнее завершить две-три небольшие игры, чем забросить один неподъемный проект.

Пять игр в этой книге строго следуют этой стратегии. Их объем сознательно ограничен – не только чтобы уместить материал в рамки урока, но и чтобы фокус оставался на отработке базовых навыков. В процессе разработки у вас наверняка появятся идеи для новых функций и игровых механик: *«Что, если добавить улучшения для космического корабля?»* или *«А если научить персонажа отталкиваться от стен?»*.

Идеи – это прекрасно, но помните, что первостепенная задача – выполнить основной проект. Записывайте интересные мысли, но не позволяйте им вас отвлекать. Разработчики называют это *«раздуванием функционала»* – бесконечным добавлением функций, которое является самой распространенной причиной незавершенных проектов. Не попадайте в эту ловушку.

И наконец, не пренебрегайте перерывами. Не пытайтесь выполнить проект и тем более прочесть всю книгу за пару подходов. После изучения каждой новой концепции, и особенно после прочтения главы, дайте себе время осмыслить информацию. Это не только улучшит усвоение материала, но и сделает процесс обучения комфортнее и приятнее.

Как учиться эффективно

Вот секрет, который поможет извлечь из книги максимум пользы и надежно закрепить навыки: завершив работу над главой и создав проект, удалите его и начните с нуля. Теперь постарайтесь воссоздать проект, не заглядыва-

вая в книгу. Если зайдете в тупик, обратитесь к книге, а затем продолжите самостоятельно. Если почувствуете себя уверенно, попробуйте добавить в игру что-то свое, например изменить геймплей или добавить новую механику.

Проделывайте это с каждым проектом, и вы заметите, что обращаетесь к книге все реже. Если вы сможете самостоятельно воссоздать все проекты из этой книги, значит, вы готовы к созданию собственных уникальных игр.

Помните об этих советах, читая следующие разделы. Далее вы узнаете, что такое игровые движки и почему разработчики выбирают их для создания игр.

Что такое игровой движок?

Разработка игр – это сложный процесс, требующий широкого спектра знаний. Кроме того, для создания современной игры необходим солидный технологический фундамент. Представьте, что перед программированием вам пришлось бы сначала собрать компьютер и написать операционную систему. Примерно так выглядела бы разработка игр, если бы каждый раз приходилось начинать с чистого листа.

К тому же существует ряд общих задач, присущих каждой игре. Например, вне зависимости от жанра в любом проекте необходимо выводить изображение на экран. Если код для этого уже написан, гораздо эффективнее использовать его повторно, чем создавать заново. Именно для этого и существуют игровые фреймворки и движки.

Игровой фреймворк (game framework) – это набор библиотек со вспомогательным кодом для создания базовых компонентов игры. Однако он редко предоставляет все необходимые модули, и для их интеграции часто приходится писать много дополнительного кода. Вот почему создание игры на фреймворке обычно занимает больше времени, чем с использованием полноценного движка.

Игровой движок (game engine) – это комплекс инструментов и технологий, который значительно упрощает создание игр, избавляя разработчиков от необходимости изобретать велосипед для каждого нового проекта. Он предоставляет набор стандартных функций, реализация которых с нуля потребовала бы огромных затрат времени и сил.

Типичный игровой движок обычно включает следующие возможности:

- **рендеринг (2D и 3D):** рендеринг – это процесс отображения игры на экране. Эффективный рендеринг подразумевает поддержку современных графических процессоров (*GPU – Graphics Processing Unit*), работу с высокими разрешениями, а также реализацию эффектов, таких как освещение, перспектива и настройка областей просмотра (вьюпортов), – и все это при условии высокой частоты кадров;
- **физика:** хотя это и базовая потребность, создание точного и стабильного физического движка – сложнейшая задача. Почти любой

игре нужна система обнаружения столкновений, а часто и продвинутая физическая симуляция. При этом найдется мало разработчиков, кто станет писать подобный код с нуля, особенно без соответствующего опыта;

- **кросс-платформенность:** на современном рынке большинство разработчиков стремятся выпускать игры сразу для нескольких платформ: ПК, консолей, мобильных устройств и/или веб-браузеров. Движок решает эту задачу, предлагая единый процесс экспорта для публикации на разных платформах без необходимости переписывать код и поддерживать несколько его версий;
- **единая среда разработки:** работая в одном знакомом интерфейсе над разными играми, разработчику не приходится осваивать новый рабочий процесс для каждого проекта.

Помимо перечисленного, в движок встроены инструменты для реализации сетевых функций, управления изображениями, звуком и анимацией, а также средства отладки и многое другое. Часто движки позволяют импортировать контент, созданный в других программах (например, для анимации или 3D-моделирования).

Использование движка позволяет разработчикам сосредоточиться на самой игре, а не на разработке ее базовой архитектуры. Небольшим командам и независимым разработчикам это дает возможность выпустить игру за год вместо трех или вовсе не забросить проект.

Сегодня на рынке представлены десятки популярных игровых движков, таких как Unity, Unreal Engine, GameMaker Studio и др. Важно понимать, что большинство из них – коммерческие продукты. Они могут не требовать первоначальных вложений, однако, если ваша игра начнет приносить доход, с большой вероятностью придется платить лицензионные отчисления и/или роялти. Вне зависимости от выбора движка первым делом внимательно изучите лицензионное соглашение, чтобы четко понимать условия использования движка и возможные издержки.

С другой стороны, существуют и некоммерческие движки с *открытым исходным кодом*, такие как Godot, которому и посвящена эта книга.

Что такое Godot?

Godot – это полнофункциональный современный игровой движок, предоставляющий все описанные возможности и даже больше. Он распространяется бесплатно и имеет открытый исходный код под лицензией MIT. Это означает отсутствие взносов, скрытых издержек и роялти. Все, что вы создаете на Godot, на 100 % принадлежит вам. Для многих разработчиков это ключевое преимущество.

Если вы впервые сталкиваетесь с концепцией *открытого исходного кода* (*open-source*), это может показаться необычным. Однако Godot, как

и ядро Linux или браузер Firefox, не коммерческий продукт корпорации. Его создает и развивает сообщество энтузиастов, которые вкладывают время, знания и силы в улучшение движка, поиск ошибок и написание документации.

Для разработчиков Godot предлагает ряд важных преимуществ. Отсутствие коммерческой лицензии дает полную свободу в выборе способа и площадок для распространения игры. В отличие от многих коммерческих движков, которые могут ограничивать типы проектов или требовать приобретения дорогостоящей лицензии для определенных жанров (например, азартных игр), Godot не накладывает таких ограничений.

Открытый исходный код Godot обеспечивает прозрачность, недоступную в коммерческих движках. Это также означает, что, если возможностей движка недостаточно для ваших задач, вы можете модифицировать его исходный код и добавить нужный функционал. Кроме того, открытый код значительно упрощает отладку сложных проектов, поскольку вы получаете полный доступ к внутренней архитектуре движка.

Вы также можете напрямую участвовать в развитии Godot. Подробнее о том, как внести вклад в разработку движка, читайте в главе 7.

Теперь, когда вы понимаете, что такое Godot и как он помогает в создании игр, самое время приступить к работе. В следующем разделе вы узнаете, как скачать Godot и подготовить его к работе на вашем компьютере.

Загрузка Godot

Чтобы скачать последнюю версию Godot, перейдите на официальный сайт <https://godotengine.org/> и нажмите кнопку **Download Latest** (Скачать последнюю версию). Эта книга написана для версии 4.0. Если вы скачаете версию с дополнительными цифрами в названии (например, 4.0.3) – не беспокойтесь, это актуальный релиз с исправлениями.

На странице загрузки представлены два варианта: стандартная версия и версия .NET. Версия .NET предназначена для разработки на языке C#. Выбирайте стандартную сборку, если не планируете его использовать. Все проекты в этой книге созданы в стандартной версии движка.



Рис. 1.1. Страница загрузки Godot

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru