

Содержание

Об авторе	7
О технических рецензентах	8
Предисловие	9
Глава 1 ❖ Основные принципы анимации	15
Основные понятия анимации	15
Кадры	16
Ключевые кадры	17
Типы анимации	18
Анимация твердого тела	18
Оснащенная анимация, или анимация на основе скелета	19
Анимация спрайтами	20
Анимация, основанная на физике	21
Анимации превращения	22
Видеоанимация	23
Анимация частицами	25
Программная анимация	25
Анимация с помощью кода – придание предметам движения	27
Совместимая анимация – скорость, время и deltaTime	29
Движение в направлении	31
Программирование автогенерации с помощью анимационных кривых	34
Вращение объектов – анимация с помощью сопрограмм	38
Материалы и рельефная анимация	41
Дрожание камеры – анимационные эффекты	43
Итоги	45
Глава 2 ❖ Анимация спрайтами	46
Спрайты, их импорт и настройка	47
Отдельные спрайты	47
Атлас спрайтов	50
Анимация с помощью спрайтов	54
Анимация спрайтами слишком медленная или слишком быстрая	56
Анимация должна быть цикличной	58
Кадры воспроизводятся в неправильном порядке	58
Итоги	61
Глава 3 ❖ Встроенная анимация	63
Окно анимации – создание полета	63
Анимация нескольких объектов сразу	71
Вызовы функций из анимации	72

Системы частиц	75
Система частиц для имитации светлячков.....	77
Общие свойства системы частиц.....	79
Форма эмиттера и скорость эмиссии.....	80
Внешний вид частиц.....	82
Скорость частиц	83
Цвет частиц и их исчезновение.....	85
Итоги	87

Глава 4 ❖ Анимация предметов с помощью системы Mecanim

89

Подготовка сцены с помощью прототипирования активов	89
Создание анимаций для кнопки и двери	90
Начало работы с системой Mecanim	95
Переходы и параметры системы Mecanim	98
Создание графа системы Mecanim для открывания двери	105
Создание интерактивных сцен	105
Итоги	113

Глава 5 ❖ Основы анимации персонажей

114

Создание оснащенных персонажей	114
Импорт оснащенных персонажей	116
Аватары и ретаргейтинг	118
Ретаргейтинг анимации	124
Перемещение корневого объекта	130
Исправление смещения при движении	130
Итоги	135

Глава 6 ❖ Продвинутая анимация персонажей

137

Создание управляемого персонажа.....	137
Смешивание деревьев	139
Размерности	142
Отображение десятичных чисел с дробной частью	146
Подготовка к написанию скрипта для смешивания деревьев анимаций.....	149
Написание скрипта для управления смешиванием деревьев системы Mecanim.....	150
Тестирование смешивания деревьев системы Mecanim.....	152
Итоги	152

Глава 7 ❖ Смешивание форм, инверсная кинематика и анимированные текстуры

154

Смешивание форм	154
Инверсная кинематика	157
Анимированные текстуры	168
Итоги	173

Предметный указатель

175

Об авторе

Алан Торн (Alan Thorn) – независимый разработчик игр и писатель, более 13 лет занимающийся промышленной разработкой игр. Он основатель компании Wax Lygical Games и создатель игры *Baron Wittard: Nemesis of Ragnarok*, отмеченной многочисленными призами. Он также автор более десятка видеокурсов и пятнадцати книг о разработке игр, среди которых – *Mastering Unity Scripting*, *How to Cheat in Unity 5* и *UDK Game Development*. Кроме того, он является приглашенным лектором магистерского курса дизайна и разработки игр в Национальной школе кино и телевидения в Лондоне.

Алан участвовал как независимый разработчик в более чем 500 проектах по созданию игр, симуляторов, игровых киосков, сложных игр, программ расширения реальности, программного обеспечения для игровых студий, музеев и тематических парков по всему миру. В настоящее время он работает в двух игровых проектах. Он также увлекается графикой, философией, йогой и пешими походами. Больше информации о нем можно найти на сайте <http://www.alanthorn.net>.

О технических рецензентах

Адам Сингл (Adam Single) – муж, отец, профессиональный разработчик, инди-разработчик, любитель музыки и геймер. Он программист игры **7bit Hero**; программист в команде Real Serious Games в Брисбене, Австралия; соучредитель, программист и содизайнер Sly Budgie; организатор компании Game Technology Brisbane meetup.

В области профессиональной разработки игр он с 2011 года, он работал над многими мобильными играми, в том числе над хитом для андроида *Photon* и предустановленными играми для мобильных телефонов Disney Japan's. Адам был программистом в команде, которая создала огромный интерактивный дисплей в технологическом университете Квинсленда (Queensland University of Technology's), поразительную мультитач-инсталляцию, которая была названа The Cube. Это было сделано в рамках Australia's first Digital Writing Residency. Он также работал в команде Real Serious Games над созданием масштабных интерактивных демонстраций для горнодобывающей и строительной промышленности, некоторые из которых являются виртуальными реальностями и основаны на Oculus Rift. Все это было сделано с помощью игрового движка Unity.

У Адама страсть к уникальным возможностям современных технологий. Когда он не работает над новой перспективной игровой механикой для компании Sly Budgie, он экспериментирует с домашней виртуальной реальностью с использованием технологии мобильного телефона, создает мобильные приложения, применяя самые современные методы веб-разработки и выдвигая захватывающие идеи.

Брайан Р. Вензен (Bryan R. Venzen) – художник визуальных эффектов и аниматор из Тампы, штат Флорида. После окончания Florida Interactive Entertainment Academy (FIEA) в 2011 году он увлекся спецэффектами и анимацией для мобильных телефонов, ПК и игровых приставок. Благодаря своим способностям и опыту он вносит свой вклад во многие проекты, создавая спецэффекты, включая спецэффекты для производства, работающие на физическом движке NVIDIA (PhysX), анимацию, и выполняя моделирование в пакетах Autodesk 3ds Max и Maya, разрабатывая уровни на Unity и UDK, активно участвуя в обсуждениях дизайна, чтобы улучшить общее качество продукции. Брайан быстро обучается и целеустремленно работает в команде, владея многими 3D-пакетами, движками и инструментами для игр. Он является специалистом в вопросах поиска и устранения неисправностей и хорошо себя чувствует в быстро развивающейся рабочей среде. Создание развлекательного контента – его страсть.

Предисловие

Ничто не сравнится с ней! Анимация играет в играх главную роль практически везде, от простого применения, такого как, например, полет космического корабля, до сложного, как мимика лица. Вы просто не сможете разработать полноценную игру и по-настоящему интерактивное приложение без хоть какого-то использования анимации. В этой книге мы шаг за шагом обойдем всю обширную область, предлагаемую Unity, для создания анимации в реальном времени. Мы начнем свой путь с самого простого. Эта книга не предполагает предварительного наличия каких-либо знаний об анимации или опыта применения анимации в Unity. Однако книга предполагает хотя бы некоторое знакомство с разработкой в Unity, умение пользоваться ее основным интерфейсом и хотя бы начальные навыки написания кода на языке программирования C#. Используя все это в качестве отправной точки, вы с помощью данной книги научитесь создавать множество удивительных анимаций и применять их для отображения реального мира, сможете достичь ощутимых результатов быстро и легко. А сейчас давайте посмотрим, что ждет нас впереди...

Что охватывает эта книга

Глава 1 «Основы анимации» начинается с формального описания анимации как понятия, раскрывается сущность анимации. Затем в этой главе рассматриваются применяемые в Unity типы анимаций и описание связанных с ними скриптов. Сюда включены описания работы с `deltaTime`, анимационными кривыми и даже рельефная анимация мешей. В общем, эта глава начинает наше путешествие по анимации и охватывает все, что нам необходимо знать для продвижения вперед.

Глава 2 «Анимация спрайтами» исследует анимацию в мире 2D, свойства спрайтов, используемых для создания 2D-игр, текстуры книг с двигающимися картинками и плоскую анимацию. Рассматриваются редактор спрайтов Unity, кадры анимации, частота кадров и исправление общих проблем анимации.

Глава 3 «Встроенная анимация» позволит вам ознакомиться с анимацией и для 2D, и для 3D в Unity, научит вас использованию окна Animation и системы частиц. Мы рассмотрим два реальных и полезных проекта. Во-первых, мы заставим камеру облететь уровень. Во-вторых, мы создадим систему частиц светящейся пыли, которая нередко встречается в фантастических играх.

Глава 4 «Анимация предметов с помощью системы Mecanim» научит вас обращению с флагманским инструментом Unity, носящим имя Mecanim. Mecanim включает в себя набор функций, совместное использование которых делает возможным создание современной плавной анимации, особенно для персонажей. В этой главе мы рассмотрим несколько нетрадиционные примеры использования системы Mecanim для оживления дверей и других интерактивных реквизитов уровней.

Глава 5 «Анимация персонажей: основы» начинается с анализа оснащенной анимации персонажей. В этой главе мы рассмотрим, как анимировать гуманоидный скелет и персонаж в реальном времени. Мы также рассмотрим, как импортировать готовые персонажи и оптимально настроить их для анимации.

Глава 6 «Продвинутая анимация персонажей» логически вытекает из предыдущей главы. Итак, возьмем импортированный и настроенный в Unity персонаж и анимируем его как управляемый игроком. Этот персонаж будет поддерживать анимацию ожидания, анимацию ходьбы и бега, а также анимацию поворотов влево и вправо.

Глава 7 «Смешивание форм, инверсные кинематики и анимированные текстуры» последняя в этой книге. В этой главе рассматриваются следующие темы: смешивание форм для анимации лица и других телодвижений, обратная кинематика для точного позиционирования рук и ног персонажа в режиме реального времени, анимированные текстуры для воспроизведения видеофайлов в виде проецирования на геометрических поверхностях.

Что вам необходимо при чтении ЭТОЙ КНИГИ

Эта книга содержит почти все, что вам нужно при работе с ней. Каждая глава рассматривает практические, реальные проекты анимации и включает сопутствующие файлы, которые можно загрузить и использовать. Одно только вам нужно, не считая этой книги и вашего внимания к ней, – это копия самой последней версии Unity; на время написания данной книги это Unity 5. Приложение Unity 5 бесплатно для личного использования и может быть загружено с сайта Unity <https://unity3d.com/>. В дополнение к Unity, если вы захотите создать движущиеся модели персонажей и прочие анимированные 3D-активы, вам понадобится программное обеспечение для 3D-моделирования и анимации, такое как 3ds Max, Maya или Blender. Blender можно бес-

платно использовать, загрузив его с сайта <http://www.blender.org/>. Однако в этой книге описана только анимация в Unity.

Для кого эта книга

Если вы имеете базовые знания о Unity и собираетесь их расширить, хотите узнать больше об анимации в реальном времени, тогда эта книга для вас. Автор книги предполагает, что вы уже разрабатываете простые приложения, пишете несложные скрипты и желаете научиться чему-то большему, например понимать наиболее продвинутые концепции анимации. Особенно вы, возможно, хотите узнать, как инструменты, такие как Mecanim, помогут вам не только создать невероятную анимацию, но и сделать вашу работу проще и эффективнее.

Соглашения

В этой книге использовано несколько стилей текста для выделения разных видов информации. Здесь приведены примеры этих стилей с объяснением их назначения.

Тексты программ, имена таблиц баз данных, имена папок, файлов, расширения файлов, параметры, просто URL, ввод пользователя и заголовки в Twitter будут выглядеть так: «По умолчанию все новые скрипты создаются с двумя функциями Start и Update».

Блок программного кода будет приведен в следующем виде:

```
using UnityEngine;
using System.Collections;
public class MoviePlay : MonoBehaviour
{
    //Reference to movie to play
    public MovieTexture Movie = null;
    // Use this for initialization
    void Start ()
    {
        //Get Mesh Renderer Component
        MeshRenderer MeshR = GetComponent<MeshRenderer>();
        //Assign movie texture
        MeshR.material.mainTexture = Movie;
        GetComponent<AudioSource>().clip = Movie.audioClip;
        GetComponent<AudioSource>().spatialBlend=0;
        Movie.Play();
        GetComponent<AudioSource>().Play();
    }
}
```

Когда мы захотим обратить ваше внимание на отдельные части программного кода, соответствующие строки или слова будут выделены полужирным шрифтом:

```
using UnityEngine;
using System.Collections;
public class Mover : MonoBehaviour
{
    // Use this for initialization
    void Start () {
    }
    // Update is called once per frame
    void Update ()
    {
        //Transform component on this object
        Transform ThisTransform = GetComponent<Transform>();
        //Add 1 to x axis position
        ThisTransform.position += new Vector3(1f,0f,0f);
    }
}
```

Новые термины и важные слова будут выделены полужирным шрифтом. Слова, которые вы увидите на экране, например в меню или в диалоговых окнах, будут отображены в нашем тексте так: «Как только кривая будет закончена, для тестирования кода запустите проект в Unity, и вы увидите эффект на закладке **Game**».



Так будут оформляться предупреждения и важные примечания.



Так будут оформляться советы и рекомендации.

Обратная связь с читателями

Для нас важны отзывы наших читателей. Дайте нам знать, что вы думаете об этой книге, что вам понравилось или, может быть, не понравилось. Отзывы читателей важны для определения наиболее нужных вам направлений.

Чтобы связаться с нами, просто отправьте электронное письмо по адресу feedback@packtpub.com, указав название книги в теме вашего сообщения.

Если есть область, которую вы хорошо знаете, и хотите написать книгу или поспособствовать выходу книги, ознакомьтесь с нашим руководством для авторов на www.packtpub.com/authors.

Поддержка пользователей

Если вы гордый владелец книги, изданной в Packt, мы готовы предоставить вам дополнительные услуги, чтобы ваша покупка принесла вам максимальную пользу.

Загрузка активов к книге

Вы сможете загрузить файлы с примерами кода из вашего аккаунта на <http://www.packtpub.com> для всех купленных вами книг издательства Packt Publishing. Где бы вы не купили эту книгу, вы можете посетить страницу сайта <http://www.packtpub.com/support> и зарегистрироваться для получения файлов по электронной почте.

Скачивание цветных изображений из этой книги

Мы также предоставляем в ваше распоряжение файл формата PDF, который содержит цветные изображения скриншотов и диаграмм, используемых в этой книге. Цветные изображения помогут вам лучше понять содержание книги. Вы можете загрузить этот файл по ссылке https://www.packtpub.com/sites/default/files/downloads/48130T_Colored-Images.pdf.

Опечатки

Хотя мы тщательно проверили содержание нашей книги, ошибки все же случаются. Если вы нашли ошибку в любой из наших книг, в тексте или программном коде, мы будем благодарны, если вы сообщите об этом нам. Сделав это, вы оградите других читателей от разочарования и поможете нам в улучшении следующих изданий данной книги. Если вы нашли опечатку, пожалуйста, сообщите о ней, для этого посетите сайт <http://www.packtpub.com/support>, выберите вашу книгу, щелкните по ссылке *let us know* и введите описание опечатки. Как только ваше сообщение об опечатке будет проверено, ваше обращение будет принято, и опечатка будет добавлена в общий список опечаток. Общий список опечаток можно просмотреть, выбрав нужную книгу, на <http://www.packtpub.com/support>.

Пиратство

Пиратство в отношении авторского права в Интернете является общей проблемой всех издателей.

В Packt мы относимся к защите авторских прав очень серьезно. Если вы столкнетесь с любыми нелегальными копиями в Интернете, сообщите нам адрес или название веб-сайта, чтобы мы могли принять необходимые меры.

Пожалуйста, сообщите нам о пиратских копиях по электронному адресу copyright@packtpub.

Мы ценим вашу помощь в защите наших авторов и наших возможностей в предоставлении вам ценной информации.

Вопросы

Если у вас возникнут любые вопросы по книге, вы можете связаться с нами по адресу questions@packtpub.com.

Глава 1

Основные принципы анимации

Привет и добро пожаловать в путешествие по множеству функций анимации в мире Unity. Важность анимации нельзя переоценить. Без анимации все в игре будет статичным, безжизненным и даже скучным. Это справедливо абсолютно для всего в играх: двери должны открываться, персонажи – двигаться, листья – колебаться ветром, искорки и частицы – вспыхивать и гаснуть и т. д. Следовательно, обучение анимации, безусловно, даст вам как разработчику перспективы карьерного роста. Так как анимация присутствует на игровом поле большинства игр, то она имеет отношение ко всем членам команды, это очевидно для художников и аниматоров, но касается и программистов, дизайнеров звука и дизайнеров оформления уровней. Эта книга одновременно ценна и актуальна для большинства разработчиков, она направлена на быстрое и эффективное внедрение основных концепций и практики анимации реального времени в игры, в частности в Unity. К концу этой книги, если вы прочтете каждую главу внимательно, вы получите прочные знания и навыки в области анимации. Вы сможете добиться того качества анимации, которое отражает ваше художественное видение, а также ознакомиться с тем, как и где вы можете расширить свои знания до следующего уровня. Но для того, чтобы достигнуть этой стадии, мы начнем здесь, в первой главе, с самых основных понятий анимации – основ для понимания любой анимации.

Основные понятия анимации

На фундаментальном уровне анимация основывается на отношении между двумя специфичными и независимыми свойствами, а именно **изменениями**, с одной стороны, и **временем** – с другой. Технически анимация определяет изменение во времени, то есть как свойство или величина изменяется во времени, например как положение автомо-

бия меняется с течением времени или как цвет светофора переходит с течением времени с красного в зеленый. Таким образом, каждая анимация внутри промежутка времени (продолжительности) или на протяжении всей своей жизни меняет свойства объектов в определенные моменты (кадры), помещенные в любом месте от начала до конца анимации.

Это определение само по себе техническое и несколько сухое, но актуально и важно. Хотя и не позволяет должным образом охватить эстетические и художественные свойства анимации. Через анимацию и через креативное изменение свойств с течением времени можно эффективно передать настроение, атмосферу, вселенные и идеи. Невероятно, но эмоциональные и художественные образы, которые обеспечивает анимация, в конечном счете являются продуктами главного соотношения изменений с течением времени. В рамках этой же смены в течение долгого времени мы можем определить дальнейшие ключевые термины, в частности термины компьютерной анимации. Вы, возможно, уже знакомы с этими понятиями, но давайте определим их более формально.

Кадры

В анимации время должно обязательно быть разделено на отдельные и дискретные единицы, где могут произойти изменения. Эти моменты называются **кадрами** (frames). Время, по существу, есть непрерывное и неуничтожимое понятие, поскольку всегда можно поделить единицу времени (например, секунды), чтобы получить еще меньшую единицу времени (например, миллисекунды), и т. д. В теории этот процесс деления, по существу, может продолжаться до бесконечности, в результате чего будут получены все меньшие и меньшие доли времени. Понятие **момента** времени или **события** во времени, в отличие от этого, человек сделал дискретным и автономным объектом. Оно является дискретным, чтобы соответствовать нашему восприятию, согласоваться с нашим реальным опытом. В отличие от времени, момент – это нечто, что не может быть разбито на что-то меньшее, не переставая существовать вообще. Внутри момента или кадра что-то происходит. Кадр предоставляет возможностям свойствам измениться: двери – открыться, персонажу – передвинуться, цвету – поменяться и т. д. В видеоанимации, в частности, каждая секунда содержит определенное количество кадров. Количество кадров в одной секунде будет варьироваться от компьютера к компьютеру, в зависимости от количественных характеристик аппаратуры, установленного про-

граммного обеспечения и других факторов. Емкость кадров в секунду называется частотой кадров в секунду (FPS). Она часто используется в качестве меры производительности игр, более низкая частота кадров, как правило, связана с подергиванием и низкой производительностью. Рассмотрим рис. 1.1, показывающий, как кадры делят время на периоды.

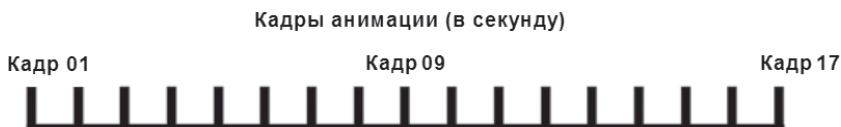


Рис. 1.1. Разделение времени кадрами

Ключевые кадры

Хотя кадр и предоставляет возможность для изменений, но это не обязательно значит, что изменение *произойдет*. Секунда может включать в себя множество кадров, но не в каждом кадре нужны будут изменения. Более того, даже если изменения и необходимы в каждом кадре, аниматорам было бы очень утомительно отдельно определять каждый кадр действия. Одним из преимуществ компьютерного метода анимации, в отличие от ручного, или «старого», метода анимации, является то, что он может сделать нашу жизнь проще. Аниматоры могут определить только ключевые или важные кадры в анимационной последовательности, а затем компьютер автоматически сгенерирует промежуточные кадры. Рассмотрим простую анимацию открытия стандартной межкомнатной двери наружу путем поворота ее на петлях на 90 градусов. Анимация начинается с двери в закрытом положении и заканчивается в открытом положении. Здесь мы определили два основных состояния двери (открыта и закрыта), и эти положения отмечают начало и конец последовательности анимации. Они называются **ключевыми кадрами** (key frames), потому что они ключевые моменты в течение анимации. На основании ключевых кадров Unity (как мы увидим) может осуществлять **автоматическую генерацию кадрами анимации** (tweens), плавно поворачивая дверь от его положения в начальном кадре до положения в конечном кадре. Математический процесс генерации отрезков называется интерполяцией. На рис. 1.2 показано, как кадры генерируются между ключевыми кадрами.

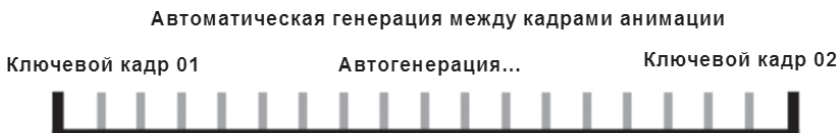


Рис. 1.2. Автоматическая генерация между ключевыми кадрами с использованием интерполяции

Типы анимации

В предыдущем разделе были определены основные понятия, лежащие в основе анимации в целом. В частности, она охватывает изменения, время, кадры, ключевые кадры анимации и интерполяцию. На основании этого мы можем выделить несколько типов анимации в видеоиграх с технической точки зрения, в отличие от взгляда с художественной точки зрения. Все эти варианты основаны на концепции, которую мы рассмотрели, но они осуществляют ее разными способами. Разделение анимаций на типы важно для Unity, потому что различие в их природе требует, чтобы мы управляли и работали с ними по-разному, с использованием конкретных рабочих процессов и методов, которые мы рассмотрим в следующих главах. Типы анимации перечислены ниже в этом разделе.

Анимация твердого тела

Анимация твердого тела используется для создания готовых последовательностей анимации, которые перемещают или изменяют свойства объектов, с учетом того, что объекты являются единым целым, в отличие от объектов, содержащих движущиеся относительно самих объектов части. Некоторыми из примеров этого типа анимации являются движение гоночного автомобиля по дороге, открытие двери на петлях, полет космического корабля сквозь пространство по своей траектории и полет пианино, падающего из окна здания. Несмотря на различия между этими примерами, у всех из них есть важное общее свойство. В частности, хотя и происходят изменения объекта в ключевых кадрах, но они производятся для единого и цельного объекта. Другими словами, хотя дверь и поворачивается на петлях из закрытого состояния в открытое состояние, она заканчивает анимацию по-прежнему как дверь, с той же внутренней структурой и тем же составом, что и раньше. Она не превращается в тигра или льва. Она не взрывается или не превращается в желе. Она не растворяется

в каплях дождя. В течение анимации дверь сохраняет свою физическую структуру. Она изменяется только с точки зрения ее положения, скорости вращения и масштаба. Таким образом, при анимации твердого тела изменения в ключевых кадрах распространяются только на целые объекты и их свойства высокого уровня. Они не касаются под свойств и внутренних компонентов и не меняют сущности или собственной формы объектов. Эти виды анимации могут быть определены либо непосредственно в редакторе анимации Unity, как мы это увидим в последующих главах, либо в программах 3D-анимации (таких как Maya, Max или Blender), а затем импортированы в Unity посредством файлов мешей. В *главе 3 «Встроенная анимация»* будет продолжено знакомство с анимацией твердого тела.

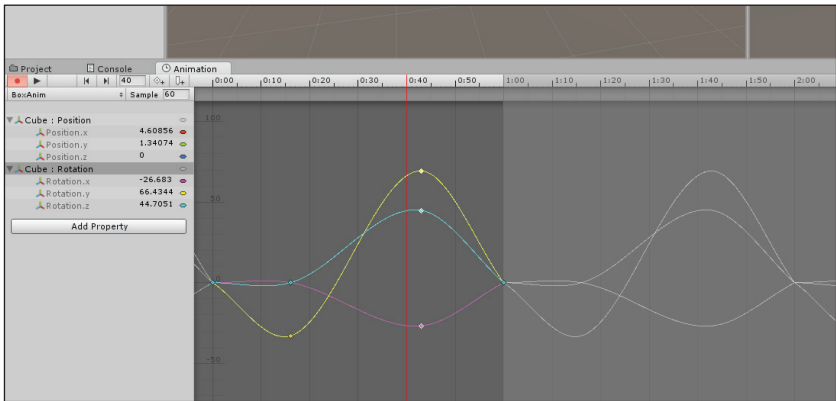


Рис. 1.3. Ключевые кадры для анимации твердого тела

Оснащенная анимация, или анимация на основе скелета

Если вам нужно оживить персонажи человека, животного, питающуюся плотью слизь, взрыв или деформацию объектов, то анимации жесткого тела, скорее всего, будет недостаточно. Вы должны будете использовать анимацию на основе скелета (также называемую изменяющей анимацией). Этот тип анимации изменяет в ключевых кадрах не положение, вращение или масштаб объекта, а движение и деформацию его внутренних частей. Вот как это работает: анимационный художник создает набор специальных объектов костей, подгоняя основной скелет к мешу, что позволит независимо и просто

управлять как внешней, так и внутренней геометрией. Это подходит для создания анимации рук, ног, поворота головы, движений рта, шелеста листьев дерева и многого другого. Как правило, анимация на основе скелета создается как полная последовательность анимации в программе 3D-моделирования и импортируется в Unity как часть файла меша, который может быть обработан и доступен с помощью системы анимации Mecanim Unity. В главах 5, 6 и 7 анимация на основе скелета будет рассмотрена более подробно.

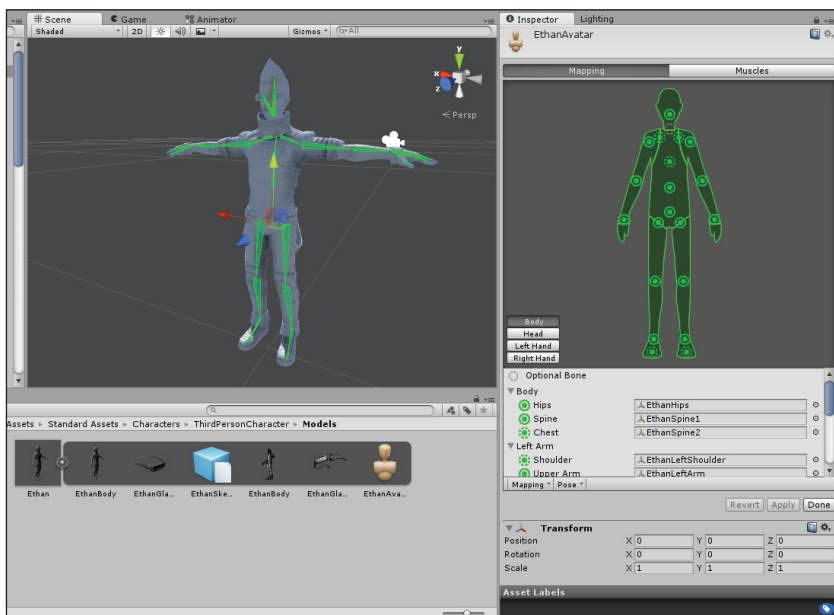


Рис. 1.4. Основанная на скелете анимация, применяемая для персонажа с мешами

Анимация спрайтами

Для 2D-игр, графических пользовательских интерфейсов, а также ряда специальных эффектов в 3D (таких как текстура воды) вам иногда понадобятся стандартные меши с анимированными текстурами. В этом случае объект движется, используя методы анимации твердого тела, но его внутренние части изменяются, как при изменяющей анимации. Это больше похоже на анимацию собственно текстур. Этот тип анимации называется **анимацией псевдоспрайтами** (sprite animation).

animation). Анимация состоит из последовательности изображений или кадров и проигрывает их в нужном порядке и с заданной скоростью для получения непрерывной анимации, например цикл ходьбы персонажа в 2D-игре с горизонтальной прокруткой. Более подробная информация об анимации спрайтами приведена в следующей главе.

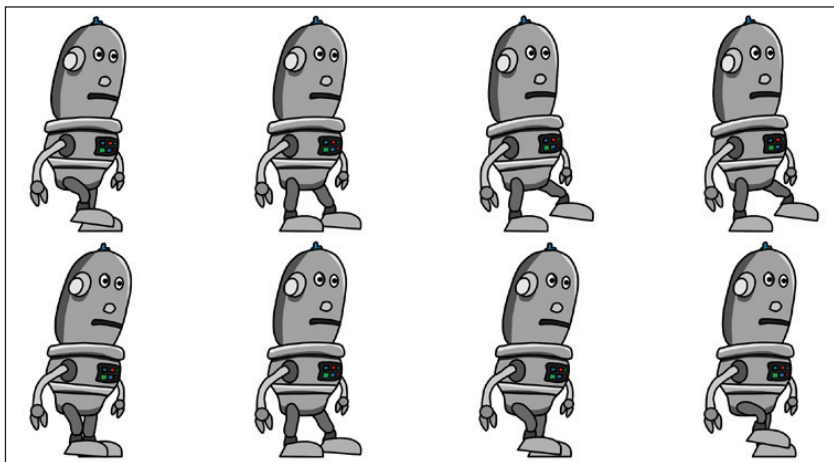


Рис. 1.5. Анимация спрайтами

Анимация, основанная на физике

Во многих случаях вы можете заранее полностью определить вашу анимацию. То есть вы можете ее полностью спланировать и создавать анимационные последовательности для объектов, которые будут проигрываться заданным образом во время выполнения, например цикл ходьбы, открытие двери, взрыв и др. Но иногда вам понадобится анимация, которая выглядит реалистично и, кроме того, еще и реагирует динамично на внешние воздействия, основываясь на решениях, принятых игроком, и других переменных факторах внешнего мира, которые не могут быть предсказаны заранее. Есть разные способы управлять таким поведением, но один из них – это использование физической системы Unity, которая включает в себя компоненты и другие данные, при присоединении их к объектам они заставляют объекты вести себя реалистично. Примерами могут послужить падение объекта на землю под действием силы тяжести или изгиб и кручение объекта, подобно ткани на ветру.

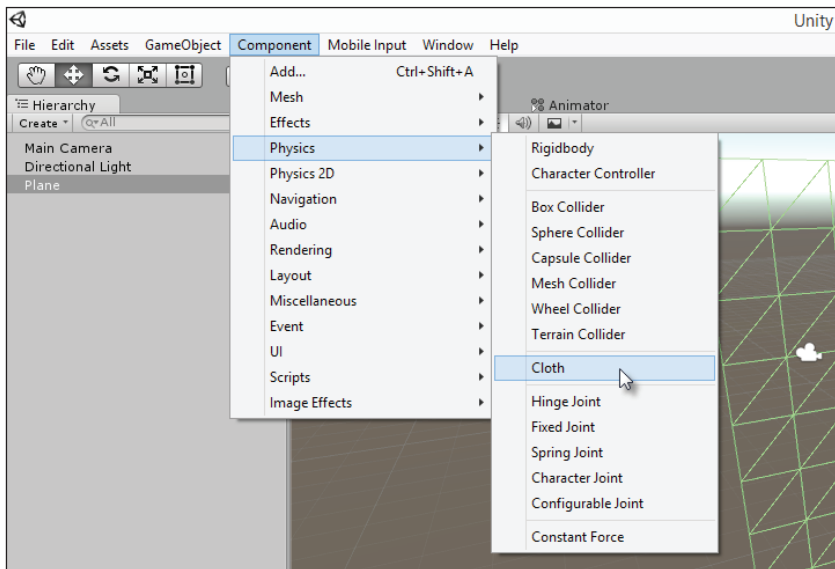


Рис. 1.6. Анимация, основанная на физике



Вы сможете загрузить файлы с примерами кода из вашего аккаунта на <http://www.packtpub.com> для всех купленных вами книг издательства Packt Publishing. Где бы вы не купили эту книгу, вы можете посетить страницу сайта <http://www.packtpub.com/support> и зарегистрироваться для получения файлов по электронной почте.

Анимации превращения

Иногда ни один из методов анимации, о которых вы уже прочли, – ни анимация твердого тела, ни анимация, основанная на физике, ни изменяющая анимация, ни анимация спрайтами – не может предоставить вам того, что вам необходимо. Может быть, вы должны превратить что-то одно в совершенно другое, например человека в оборотня, жабу в принцессу или плитку шоколада в замок. В некоторых случаях вам нужно смешать или плавно объединить состояние меша в одном кадре и другое его состояние в другом кадре. Это называется **анимацией превращения** (Morph), или смешиванием форм. По сути, этот метод опирается на фиксацию положения вершин меша в ключевых кадрах анимации и смешивает их состояния с помощью анимации. Недостатком этого метода являются затраты на вычисления. Его применение отрицательно сказывается на производительности, но

результаты могут быть впечатляющими и очень реалистичными. Мы рассмотрим смешивание форм подробно позднее в *главе 7 «Смешивание форм, инверсные кинематики и текстуры с видео»*. На рис. 1.7 показан эффект смешивания форм.

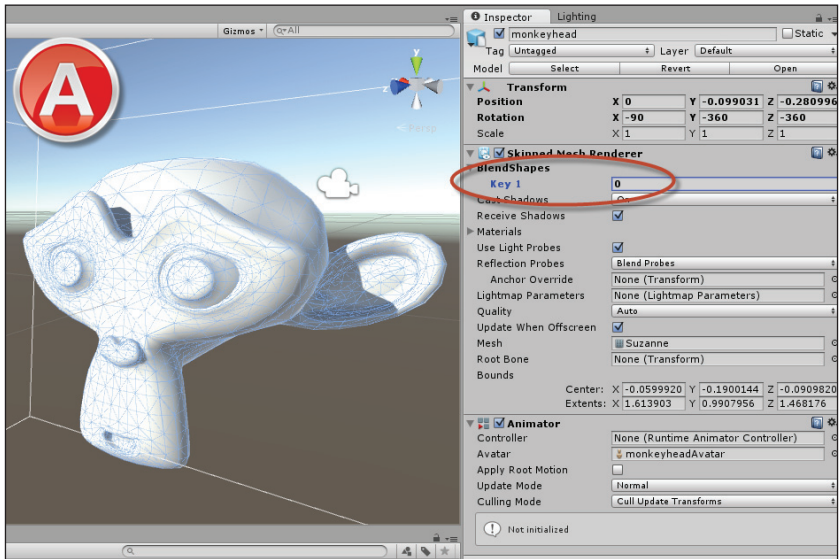


Рис. 1.7. Начальный этап анимации превращения

Смешивание форм переводит модель из одного состояния в другое. Посмотрите на изображение формы для конечного состояния (рис. 1.8).

Видеоанимация

Наверное, одной из наименее известных особенностей анимации в Unity является возможность воспроизводить видеофайлы в виде анимированных текстур как на настольных платформах, так и на мобильных устройствах, таких как iOS и Android. Unity обрабатывает файлы видеоформата OGV (Ogg Theora video) как активы и может воспроизводить видео и звук из этих файлов как анимированные текстуры на меш-объектах в сцене. Это позволяет разработчикам использовать предварительно подготовленные в любом из анимационных пакетов видеофайлы в своих играх (рис. 1.9).

Конец ознакомительного фрагмента.
Приобрести книгу можно
в интернет-магазине
«Электронный универс»
e-Univers.ru