МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра архитектуры, градостроительства и графики

УТВЕРЖДАЮ
Проректор по учебной работе
_____О.Г. Локтионова
«____»_____2022 г.

Виртуальное моделирование и компьютерная графика
Виртуальное компьютерное моделирование в архитектуре
Методические указания по выполнению лабораторных и практических работ

для студентов направлений подготовки
07.03.01-Архитектура
07.03.04-Градостроительство

Курск 2022

УДК 72.021.2

Составители: О.В. Будникова, А.С. Великанов
Рецензент: кандидат культурологии М.М. Звягинцева

Современные компьютерные технологии в архитектурном проектировании: методические указания по выполнению лабораторных и практических работ для студентов направлений подготовки 07.03.01-Архитектура, 07.03.04-Градостроительство / Юго-Зап. гос. ун-т; сост., Будникова О.В., А.С. Великанов. Курск, 2022. 245 с. Библиогр.: с. 245.

Разработаны в соответствии с рабочеми программами дисциплин «Виртуальное моделирование и компьютерная графика», «Виртуальное компьютерное моделирование в архитектуре».

Методические указания по выполнению расчётно-графических работ содержат краткие теоретические сведения модулей «Виртуальное моделирование и компьютерная графика», «Виртуальное компьютерное моделирование в архитектуре», охватывающие базовые разделы примения компьтерной графики к архитектурному проектированию и визуализации.

Содержит указания по получению начальных навыков и умений при работе с программным продуктом Unreal Engine.

Методические указания предназначены для студентов направлений подготовки: 07.03.01-Архитектура, 07.03.04-Градостроительство очной и онлайн- форм обучения.

Текст печатается в авторской редакции
Подписано в печать . Формат 60x84 1/16.
Усл. печ. л. . Уч.-изд. л. . Тираж 100 экз. Заказ .
Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Unreal Engine – современный игровой движок, способный визуализировать сложные трёхмерные объекты в реальном времени и с фотореалистичным качеством, работать в режиме виртуальной реальности и обеспечивать высокую степень интерактивности. Совместим со множеством платформ, включая мобильные и WEB.

Создание архитектурных анимаций при помощи стандартных программных пакетов занимает много времени, и они всё равно зачастую представляют неудовлетворительный результат. А чертежи и статические визуализации не всегда в состоянии передать пространственные характеристики здания. Unreal Engine позволяет перенести архитектурный проект из программного пакета трехмерного моделирования в среду реального времени и интерактивно перемещаться в её пространстве с видом от первого лица, либо по заранее заданному пути перемещения, с возможностью применения средств виртуальной реальности или без них. Допускается интерактивно взаимодействовать с объектами и освещением, например, перемещать предметы, менять их материал, включать или выключать свет.

В качестве конечного продукта можно получить как серию изображений, так и видеоролик или интерактивный проект для различных платформ.

Для интерактивного режима поддерживаются различные платформы виртуальной реальности, что позволяет демонстрировать проект пользователям с максимальным эффектом присутствия.

к преимуществам Unreal Engine можно отнести:

– быстрое создание видеороликов высокого качества с облётом здания или помещения: в отличии от классического рендеринга в системах, использующих CPU время просчёта такого ролика будет измеряться минутами, а не часами, днями или неделями;

– создание интерактивных проектов, в которых пользователь сам сможет перемещаться в пределах архитектурного проекта или интерьера: для этого проект будет упакован в исполняемый файл, который можно будет запускать на любом компьютере и интерактивно управлять перемещением камеры;

– подготовка презентаций в режиме виртуальной реальности: при наличии ВР-оборудования можно достичь полного погружения в 3d проект здания или помещения.

1 НАСТРОЙКА ПРОЕКТА

Проект в UE4 фокусируется на акклиматизации к UE4 Editor, Content Browser и Viewport уровня. Данная методичка познакомит с определением объема проекта и созданием всего необходимого для начала создания первого интерактивного мира в UE4.

1.1 Объем проекта

Прежде чем приступить к работе над любым интерактивным приложением, важно сначала определить цели проекта и записать их. Солидный объем работ и четко определенные минимальные требования гарантируют, что разработка проекта будет целенаправленной и методичной.

UE4 включает в себя множество примеров контента, который легко доступен и является отличным способом начать изучение основ UE4. Необходимо использовать этот контент, чтобы построить простой уровень, заполнить его и настроить павн, который может плавно проходить через уровень, сталкиваясь со стенами и полами, давая захватывающее ощущение прочности.

На следующем этапе происходит тестирование приложения с помощью Play-In-Editor (PIE). После удовлетворения всех пунктов, собирается и подготавливается творение для распространения в качестве автономного приложения.

1.2 Создание нового проекта из Launcher

Если вы еще не создали учетную запись на сайте unrealengine.com, не скачали и не установили Epic Games Launcher, а также не установили движок версии 4.25 или более поздней (это методическое указание использует 4.25 во всех примерах), то нужно сделать это перед началом работы.

Самый простой способ создать новый проект UE4 — это использовать окно New Project. Вы можете получить доступ к нему, запустив движок из Launcher (рисунок 1.1).

Упражнение 1. Создание проекта

После запуска Unreal Project Browser выполните следующие действия.

1. В разделе New Project Categories выберите Games.

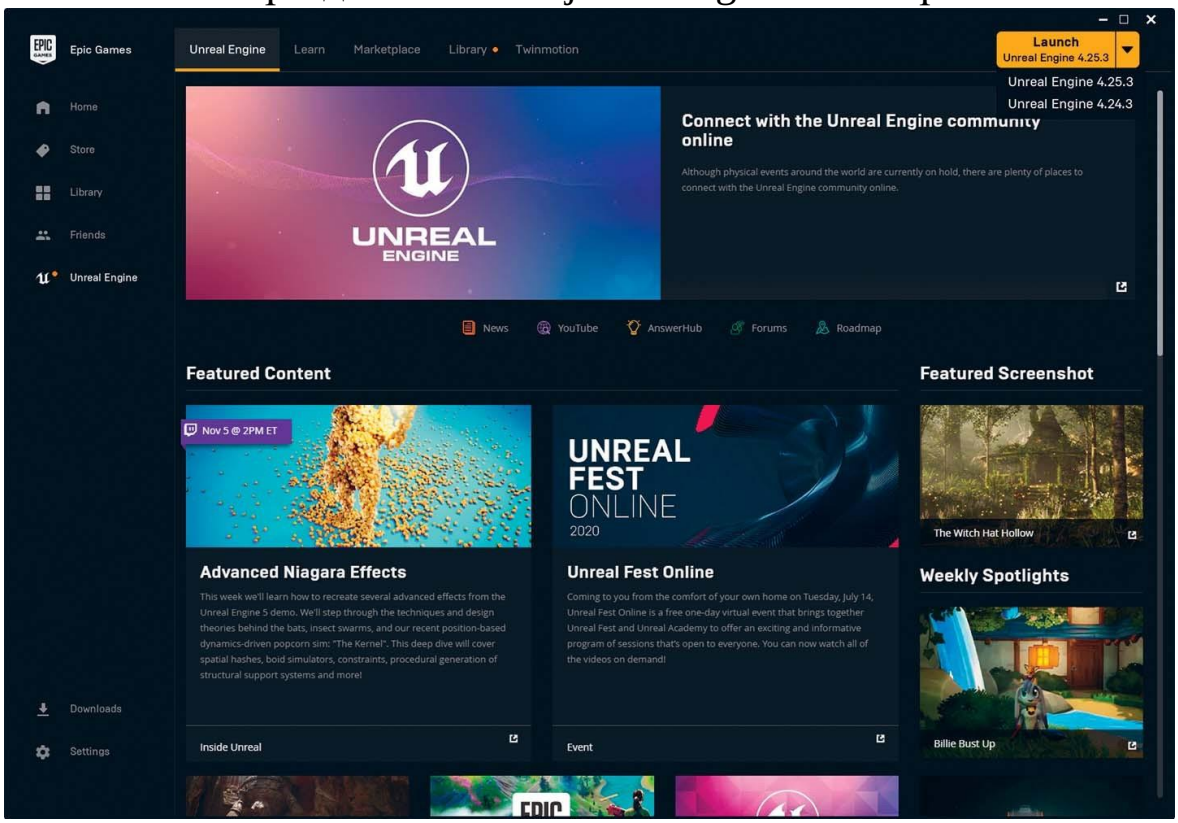

Рисунок 1.1 Запуск движка из Launcher, используя кнопку Launch

Здесь вы можете выбрать один из широкого спектра стартовых шаблонов (в других разделах помимо Games содержатся другие шаблоны, которые в методическом указании не рассматриваются). Каждый из них хорошо написан и может

стать отправной точкой для многих проектов. Они также являются отличным и простым способом играть с различными игровыми режимами и стилями (от первого лица, полет, боковой скроллер и так далее) в UE4.

Вы не будете использовать ни один из них для вашего проекта. Хотя они являются отличными отправными точками, все они ориентированы на игру и вводят вещи, которые вам придется удалить, и другие вещи, которые вы должны уметь настраивать самостоятельно.

2. Выберите шаблон Blank. Этот шаблон проекта не содержит никакого контента или кода вообще — другими словами, это совершенно чистый лист.

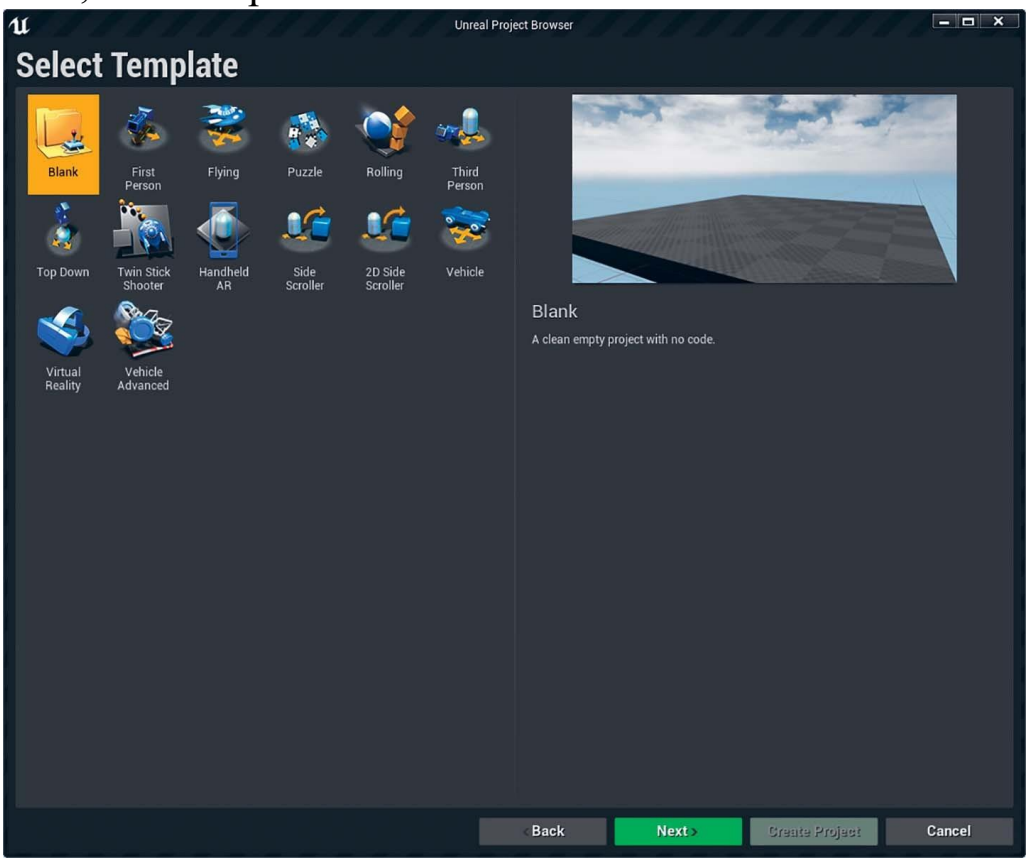

Рисунок 1.2 Панель выбора шаблона в Unreal Project Browser показывает все доступные шаблоны

3. Выберите тип проекта Blueprint и With StarterContent, что добавит в проект несколько простых готовых ассетов, они помогут вам быстро построить свой уровень. Остальные параметры остаются без изменений (рисунок 1.3).

4. Выберите место для хранения вашего проекта и дайте ему имя. Необходимо выбрать быстрый локальный жесткий диск или SSD с большим количеством свободного места (рисунок 1.3).

5. Нажмите Create Project. Движок создает новую папку и структуру папок, необходимую для запуска вашего проекта.

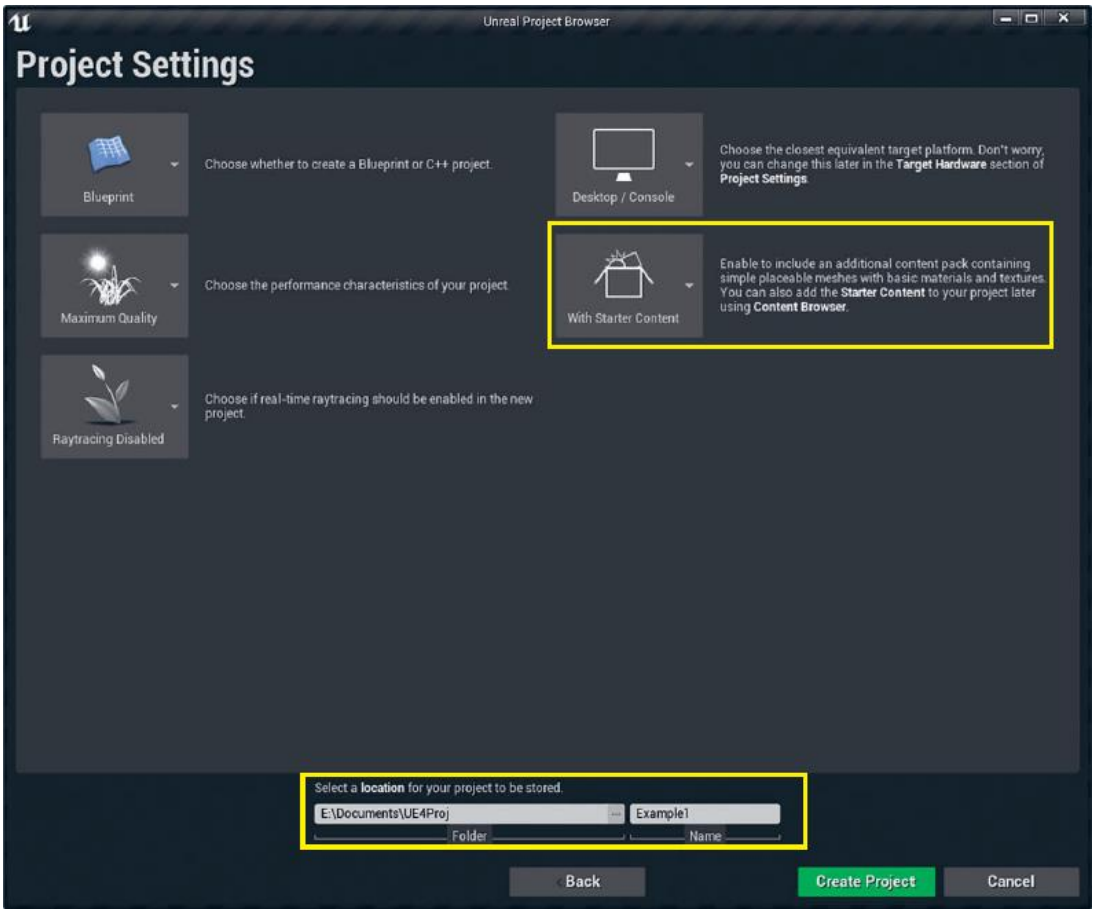

Рисунок 1.3 Панель настроек проекта в Unreal Project Browser

После создания файлов проекта запускается новый экземпляр UE4 Editor (рисунок 1.4), отображающий имя вашего нового проекта.

Рисунок 1.4 Первая загрузка нового проекта

После короткой последовательности загрузки в Editor появляется уровень (рисунок 1.5).

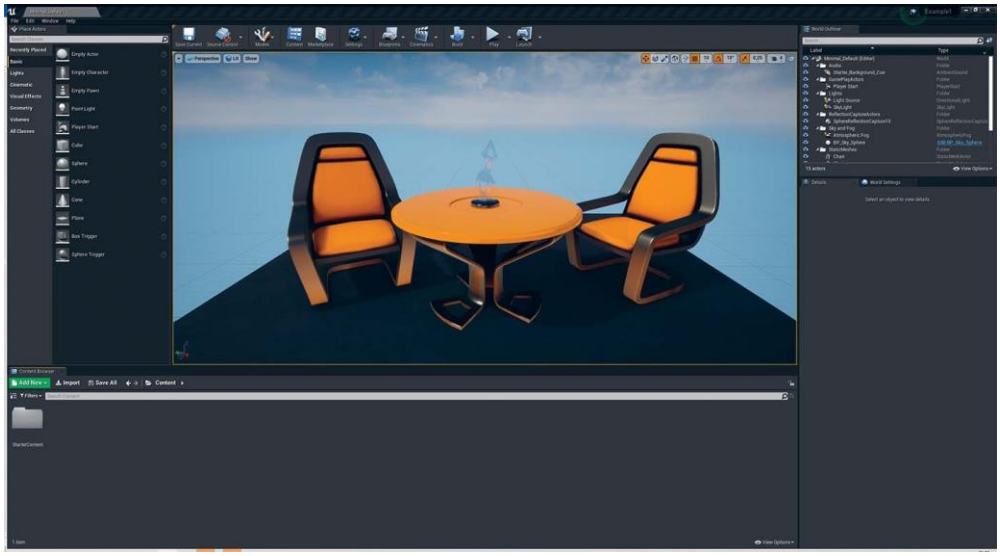

Рисунок 1.5 Открытый Unreal Engine 4 Editor показывает стартовый уровень и папку Starter Content

Когда вы включите Starter Content, уровень по умолчанию вашего проекта установится на простую сцену с размещенными примерами ассетов.

Заключение

Ваш новый проект готов к работе. Вы узнали, как создать новый проект из шаблона и настроить исходную файловую структуру в папке Content. Именно так вы будете начинать большинство своих новых проектов, так что это рабочий процесс, с которым стоит познакомиться.

Различные шаблоны позволяют легко опробовать готовые настройки, они являются отличным учебным ресурсом, а также отличным способом начать работу над конкретным типом проекта.

2 ЗАПОЛНЕНИЕ МИРА

Получение контента на вашем уровне является простым и удобным для художника. UE4 Editor предлагает множество инструментов и функций, которые помогут вам разместить, поменять и организовать ваши ассеты в 3D-пространстве. Далее

вы будете использовать уже созданные ассеты Starter Content, включенные в ваш проект, чтобы построить простой уровень, на котором можно ходить.

2.1 Создание и сохранение нового пустого уровня

Начать с нового уровня и построить уровень с нуля — это лучший способ получить желаемые результаты. Цели визуализации отличаются от целей большинства игр, и большинство доступных шаблонов просто не подходят для типов взаимодействий, необходимых при создании интерактивных визуализаций.

Упражнение 2 Создание и сохранение нового пустого уровня

В меню File выберите пункт New Level и выберите опцию Empty Level.

Сохраните свой уровень, выбрав пункт Save Current в меню File или нажав кнопку Save Current на панели инструментов. Когда появится диалоговое окно Save Level As (рисунок 2.1), назовите уровень MyFirstMap и нажмите кнопку Save, чтобы сохранить карту на диск.

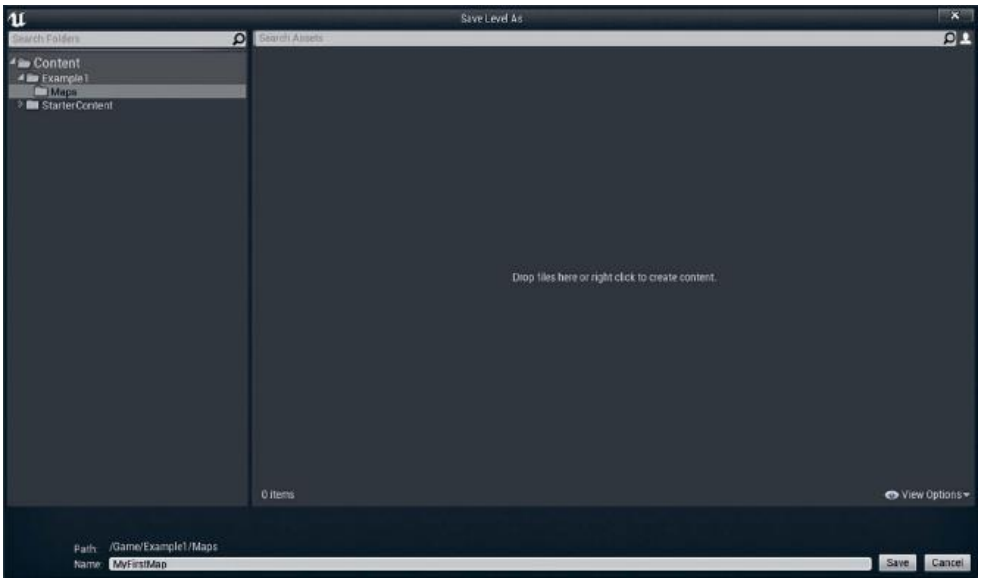

Рисунок 2.1 Диалог сохранения уровня

Как видно из этого примера, создалась папка в корне каталога Content под названием Example1. Вы можете создать новую папку, щелкнув правой кнопкой мыши в различных

логических местах Content Browser или используя кнопку Add New в Content Browser. Вы можете и должны выполнять все функции управления файлами из Content Browser.

Если вы допустили ошибку и нужно переименовать карту или папку, вы можете воспользоваться контекстным меню или выбрать ассет или папку и нажать клавишу F2 на клавиатуре.

Создание каталога для конкретного проекта в папке Content — это обычная практика в UE4, поскольку она гарантирует, что все содержимое, созданное для этого проекта, будет автономно содержаться в его собственном каталоге, в то время как содержимое от третьих лиц или других проектов может быть объединено в ваш проект без опасений, что они будут конфликтовать.

Помните, что уровни и карты — это одно и то же, и термины используются взаимозаменяемо для представления файлов UMAP.

2.2 Расстановка и модификация ассетов

Наиболее распространенный метод расстановки ассетов из Content Browser на уровень — простое перетаскивание ассетов из Content Browser во Viewport (рисунок 2.2).

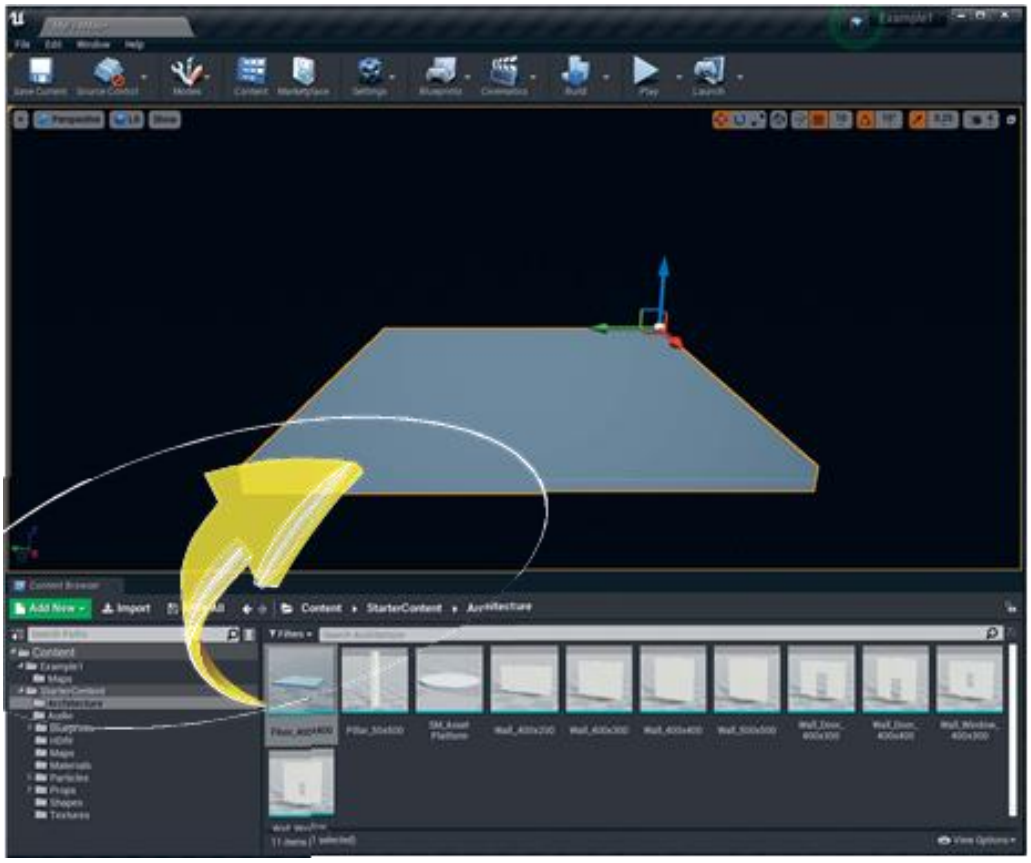

Рисунок 2.2 Перенос ассета из Content Browser на уровень

2.3 Перемещение, масштабирование и вращение

После размещения акторов вы можете легко перемещать, масштабировать и вращать их с помощью знакомых Gizmo. Вы можете легко переключаться между режимами перемещения, поворота и масштабирования с помощью пробела или значков в верхней части Viewport.

Вы также можете использовать клавиши W, E и R для переключения между перемещением, масштабированием и вращением или нажать пробел для циклического переключения между каждым из режимов.

Использование панели Details

На панели Details отображаются все свойства для каждого выбранного актора (рисунок 2.3). Здесь можно непосредственно задать свойства расположения, масштаба и поворота для акторов, а также специфические для класса настройки, такие как Lightmap Resolution, параметры тени и переопределения материала.

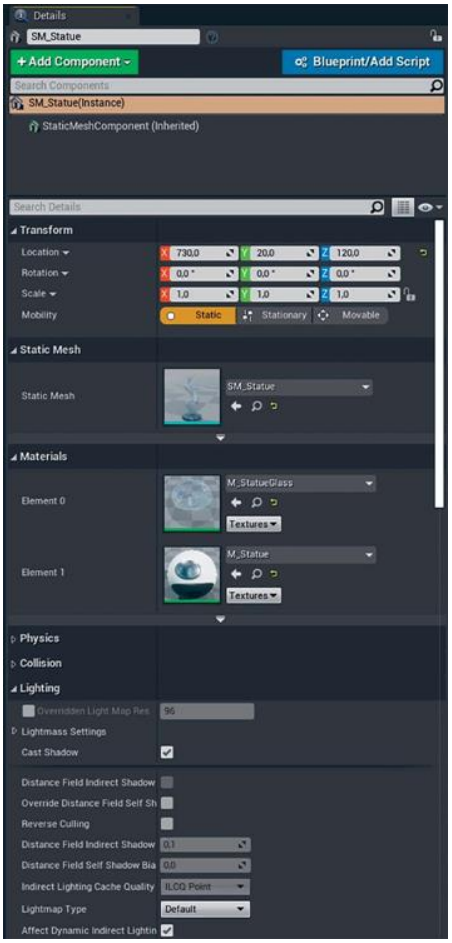

Рисунок 2.3 Панель Details для выбранного Static Mesh Actor

Snapping

Ассеты в Starter Content построены с размером меша 100 units. По этой причине включение Snap в Viewport и установка его на 100 units является хорошей идеей. Таким образом, когда вы перемещаете акторов, они привязываются к сетке единиц измерения 100 x 100 x 100.

Вы также можете установить привязку для Rotation и Scale в настройках Viewport с помощью кнопок-переключателей, расположенных в правом верхнем углу каждого видового экрана (рисунок 2.4). Каждый из них отображает интервал привязки справа. Вы можете настроить этот параметр, нажав на интервал.

Рисунок 2.4 Настройки Snap в Viewport

2.4 Дублирование

У вас есть несколько способов дублировать акторов. Выполнение копирования-вставки или использование команды дублирования в меню на правый клик и Edit — хорошие варианты.

Чаще всего используется сочетание клавиш Alt+D для мгновенного создания копий выбранного объекта. Удерживайте Shift, пока вы дублируете актора, чтобы камера следовала за актором при его перемещении. Вы также можете при помощи Alt + перетащить сразу несколько акторов. Для этого вам нужно ухватиться за трансформирующую gizmo во Viewport, а не просто в любом месте модели. Вы также можете клонировать акторов, подобных этому, с помощью вращающихся и масштабируемых gizmo.

Добавление акторов из Content Browser

Content Browser предназначен для размещения, сгенерированного или импортированного контента на уровне, но вы будете размещать в своих игровых мирах много вещей, которые недоступны в Content Browser.

Вы размещаете такие классы, как Lights, перетаскивая их из Class Browser с помощью Place Mode. Оба метода создают новый актор уровня, который создает экземпляр выбранного класса.

Окно Place Actors расположено в левом верхнем углу редактора по умолчанию. Вы также можете открыть Class Browser, перейдя в пункт Window в верхнем меню и выбрав Place Actors.

Свет в сцене

Прежде чем вы начнете расставлять акторов, нужно немного света на сцене. В отличие от Max или Maya, здесь нет встроенной сцены по умолчанию или подсветки Viewport. Требуется создать системы освещения вручную.

Если на сцене нет света, она будет использовать Unlit режим просмотра. В этом режиме просмотра отображается только незатененный Base Color объектов. Это рабочий вариант,

но так будет трудно работать в долгосрочной перспективе, потому что вы не можете легко увидеть глубину, а объекты, как правило, сливаются друг с другом.

UE4 может похвастаться отличным освещением и тенями в реальном времени. Независимо от того, используете ли вы Lightmass для высокой производительности, статическое освещение GI или динамические тени и системы освещения для прямого освещения, пара ключевых акторов освещения необходимы для достижения наилучших результатов от UE4.

Sun (солнце)

Для солнца давайте выберем Directional Light Actor. Чтобы добавить его, используйте панель Place Actor. Нажмите на Lights и перетащите Directional Light на сцену.

Установите солнце в положение Moveable на панели Details. Это позволяет ему динамически перемещать и менять свойства и заставляет его использовать динамические тени.

В большинстве архитектурных визуализаций вы использовали бы статическое освещение с Lightmass. А пока сосредоточьтесь на главном. Динамическое освещение предлагает быстрый, простой в редактировании способ WYSIWYG экспериментировать и учиться без сложностей Lightmass.

2.5 Атмосферный туман

UE4 включает в себя моделирование атмосферного тумана, или математический способ затенения неба и ослабления и тонирования света на больших расстояниях. Это сродни эффекту величия пурпурных гор, если смотреть на них с расстояния нескольких миль, когда свет рассеивается в атмосфере.

Вы размещаете Atmospheric Fog Actor, как Directional Light, используя Place Mode.

Как только вы помещаете Fog Actor на сцену, сразу же появляется основное небо и горизонт, и если вы смотрите в

правильном направлении, то вы увидите солнечный диск (рисунок 2.7, далее в этой главе).

Прямо сейчас вы заметите, что солнце находится в неправильном месте, и небо выглядит так, как будто солнце садится. Вы можете либо настроить положение солнца и цвета неба вручную, либо назначить Directional Sun Light для определения этих параметров, что даст вам хорошее динамическое небо.

Установка Солнца в атмосфере

Вы должны вручную определить, какой Directional Light Actor используется для определения цветов неба. Сделать это можно в свойствах Directional Light Actor с флажком Atmosphere Sun Light (рисунок 2.5).

Этот флажок немного скрыт в расширенных свойствах Light Actor. Чтобы открыть его, выберите маленькую стрелку вниз в свойствах света на панели Details (рисунок 2.5). Вы также можете использовать строку поиска на панели Details для быстрой фильтрации списка свойств (рисунок 2.6).

Рисунок 2.5 Детали Directional Light с отображаемыми расширенными свойствами и Atmosphere Sun Light, установленным на true

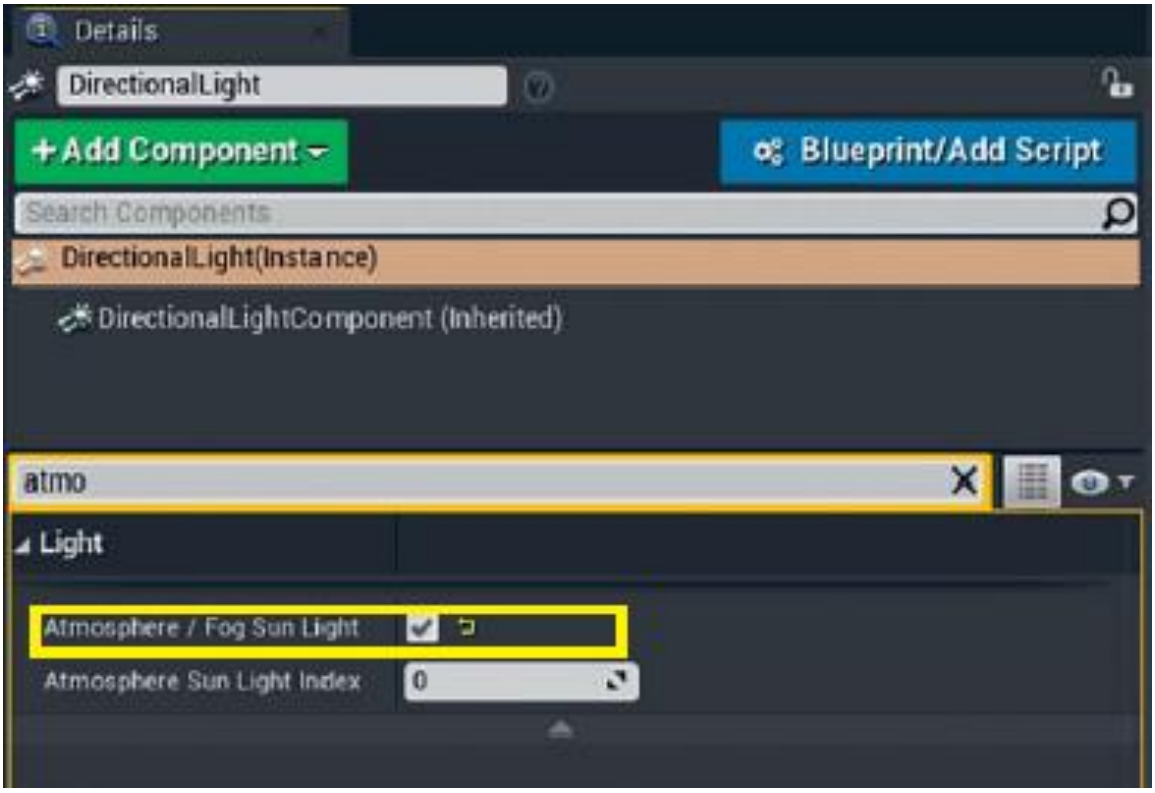

Рисунок 2.6 Отфильтрованные настройки Directional Light

2.6 Sky Light

Sky Light Actor — это третий и последний актор, которого вам нужно добавить. Этот актор захватывает HDR-кубическую карту сцены или использует определенную текстуру HDR-кубической карты для освещения сцены.

После добавления этого актора установите для параметра «мобильность» значение Moveable. Ваши затененные области должны быть заполнены синим цветом с неба. Если он слишком яркий, уменьшите Intensity в SkyLight.

Ваша сцена должна выглядеть более похожей на то, что показано на рисунке 2.7. Directional Light, Sky Light и Atmospheric Fog Actors — каждый установлен на Moveable, чтобы использовать производственный процесс динамического освещения. Макет Editor настроен таким образом, чтобы обеспечить большой Viewport и больше места для изменения свойств.

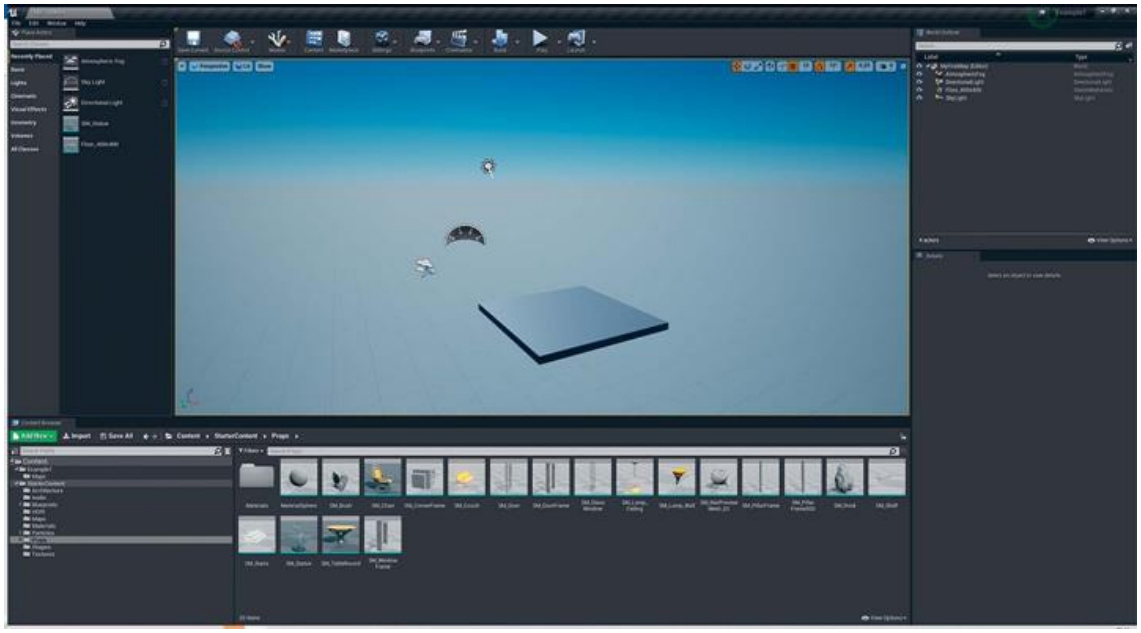

Рисунок 2.7 Основное освещение сцены с Directional Light, Sky Light и Atmospheric Fog Actors

Перемещение по сцене

UE4 имеет отличную комбинацию методов навигации по Viewport, полученных как из игр, так и из приложений 3D-дизайна.

Game Style

Самый распространенный способ перемещения по сцене — это использование игровой навигационной системы.

Удерживая нажатой правую кнопку мыши на Viewport, можно включить Game Navigation Mode. Перетаскивание мыши вокруг (при этом удерживая правую кнопку мыши) вращает вашу камеру. Нажатие W на клавиатуре переместит вас вперед, а S — назад; нажатие A и D переместит вас влево и вправо соответственно.

Вы также можете двигаться вверх и вниз с помощью клавиш E и Q.

Для настройки скорости движения вы можете использовать колесо мыши, чтобы ускорить и замедлить, как быстро вы перемещаетесь по уровню.

Object Focused

Вы также можете сфокусировать камеру на любом акторе (или группе выбранных акторов) с помощью сочетания клавиш F. Эта клавиша центрирует и приближает камеру к выбранным объектам. Когда это произойдет, вы можете вращаться вокруг выбранных акторов, удерживая нажатой клавишу Alt и перетаскивая ее левой кнопкой мыши.

Вид орбиты отлично подходит для осмотра объектов в 3D-режиме.

Вы можете увеличивать и уменьшать масштаб сфокусированных акторов, используя колесо прокрутки на вашей мыши (на этот раз без удержания правой кнопки мыши).

2.7 Создание архитектуры

Упражнение 3. Создание квартиры.

Используя различные Static Meshes в папке Architecture в Starter Content, построим простую квартиру или дом.

Если у вас есть Snapping, установите интервал в 100 units. Это делает каждый меш в этой папке привязанным к сцене как строительные блоки.

Вам понадобится пол, стены и все остальное. Убедитесь, что есть дверь, которая соединяет разные комнаты, чтобы можно было прогуляться. Каждая часть пола составляет 400 x 400 см, поэтому каждая комната должна быть не менее 2 x 2.

На рисунке 2.8 показано, собиранный набор блоков. Необходимо создать то, что удержит пользователя в небольшой области, но будет достаточно открытым для изучения. Используются только статические меши стен и пола из Starter Content, привязанного к сетке 100 × 100. Необходимо выставить Orthographic Viewports, используя Maximize/Minimize в правом верхнем углу каждого Viewport. Настроенный второй перспективный вид, позволит видеть сцену из нескольких 3D-видов.

Ваша область не должна быть такой сложной или может быть гораздо более сложной — зависит от вас. Однако

убедитесь, что у вас есть пол, чтобы пользователь мог стоять на нем, и некоторые стены, чтобы они не могли уйти в пустоту.

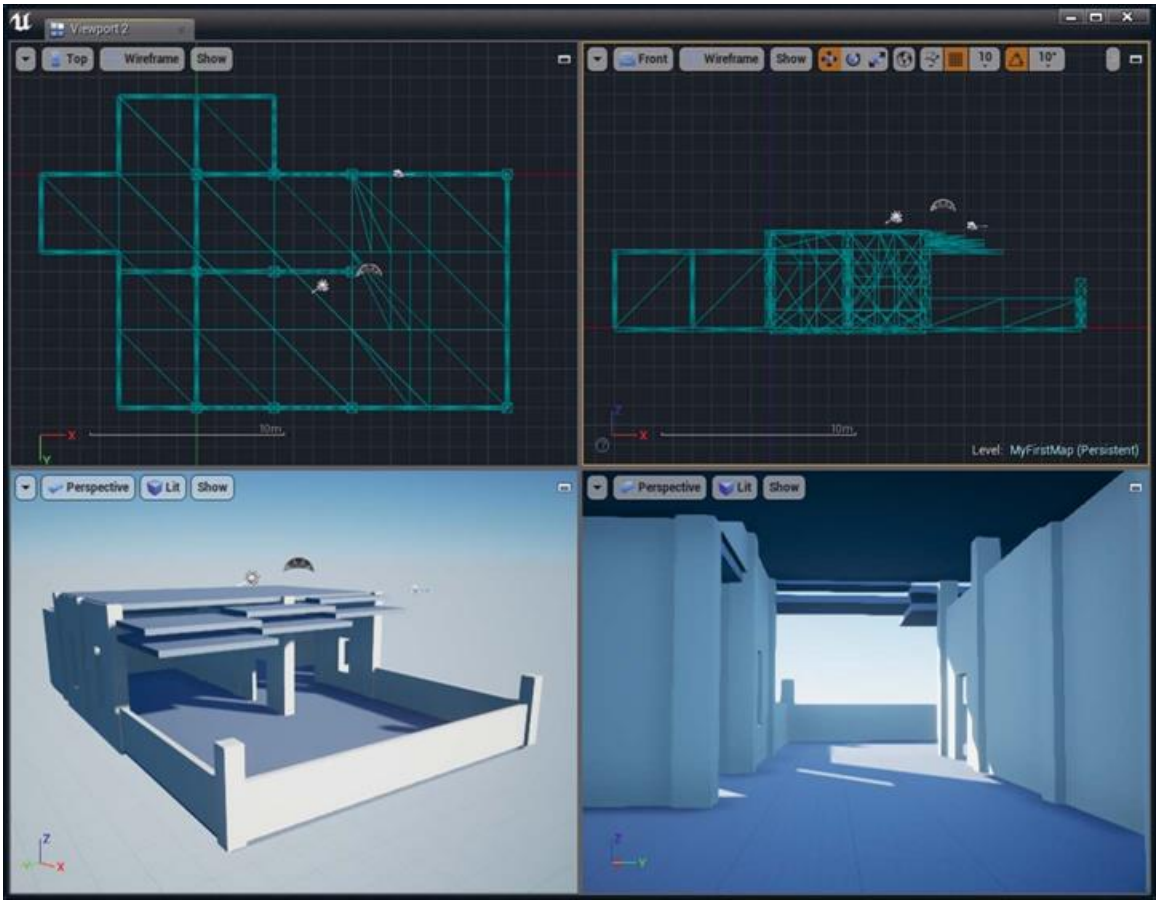

Рисунок 2.8 Дом плавает в бесконечной пустоте

Добавление деталей к структуре

Когда у вас есть структура, рассмотрим некоторые детали. Разместим несколько Static Mesh Actors из Content Browser, а затем добавим Spot Light Actors с профилями освещения и различными цветами, чтобы задействовать забавные световые эффекты.

Placing Props

Props (реквизит) — это Static Mesh Actors на сцене, которые составляют декорации и другие неархитектурные элементы. Sample Content поставляется с хорошим выбором мешей для игры.

Как и другие меши, просто перетащите их из Content Browser, чтобы разместить на вашем уровне. Скопируйте и вставьте, а затем переместите и масштабируйте по вкусу.

Вы также можете разместить материалы и системы частиц на своем уровне. Сходите с ума, держите его минимальным — на ваш вкус. Все, что действительно нужно, — это пол и несколько стен; остальное зависит от вашего воображения.

Рисунок 2.9 Сцена после нескольких минут размещения мешей

Данная сцена сделана еще через несколько минут перетаскивания, копирования, вставки, привязки поверхности и клонирования (рисунок 2.9). Потратив некоторое время на то, получается свой плавучий дом, поместив различный реквизит из Content Browser, а также дублируя и манипулируя мешами. Камни помогают визуально закрепить конструкцию.

Самое время попробовать различные варианты привязки. Необходимо изучить возможность Surface Snapping, чтобы разместить акторов непосредственно на других акторах.

Размещение света

Как и в случае с Directional Sun Light и Sky Light Actors, используйте Class Browser для добавления света на сцену.

При нажатии на вкладку Lights в Class Browser отображается список доступных классов light (рисунок 2.10). Просто перетащите нужный класс источника света на Viewport, чтобы поместить Light Actor на свой уровень.

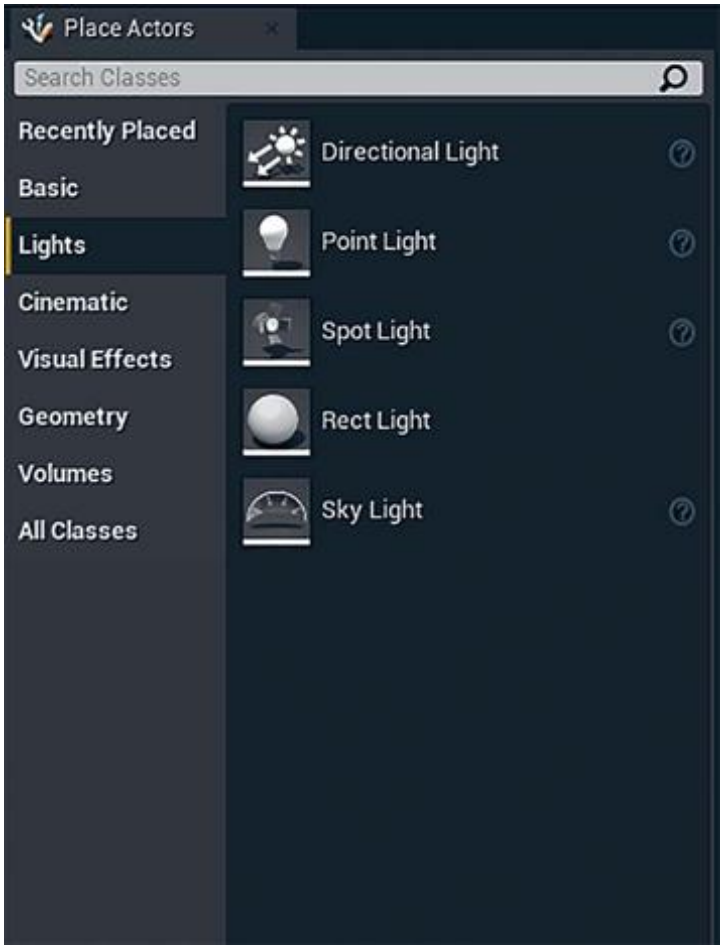

Рисунок 2.10 Light Classes из Placement Actors

Как и в случае со Static Mesh Actors, вы можете вращать и перемещать источники света с помощью Gizmo или элементов управления трансформацией на панели Details. Можно копировать, вставлять и дублировать таким же образом.

Свойства света

Потратьте время на изучение свойств Light Actor на панели Details. Оно предлагает варианты яркости, тени и цвета. Многие из этих вариантов являются чисто визуальными, в то время как многие из них тесно связаны с производительностью.

Динамическое освещение и производительность

Поскольку вы используете динамическое освещение, будьте осторожны с количеством источников света. Хотя deferred renderer UE4 позволяет использовать гораздо больше динамических источников света, чем методы рендеринга предыдущего поколения, они являются дорогостоящими эффектами, особенно когда речь заходит о затенении.

Тени

Динамические тени добавляют много накладных расходов на рендеринг, и вы должны использовать их экономно. Затененные точечные источники света являются самым дорогим видом света для визуализации и должны использоваться наиболее экономно.

Радиус затухания

Вы также можете уменьшить радиус затухания (Attenuation Radius) огней настолько, насколько это возможно, чтобы повысить производительность. Акторы вне радиуса не будут затронуты светом, и освещение и тени от этого источника света не будут вычисляться.

2.8 Добавление профилей IES

UE4 поддерживает 2D-профили IES для точечных и прожекторных источников света. Профили IES модулируют яркость света с помощью текстуры, сгенерированной из импортированного файла IES. UE4 поставляется с несколькими профилями IES, или вы можете импортировать собственный в Content Browser, как и в случае с любым другим типом контента.

Сцена после размещения нескольких источников света (рисунок 2.11). Необходимо добавить некоторые профили IES к точечным источникам света, чтобы сделать их немного более интересными. Вы можете найти свойство IES на панели Details, когда настраиваете Light Actor на своем уровне.

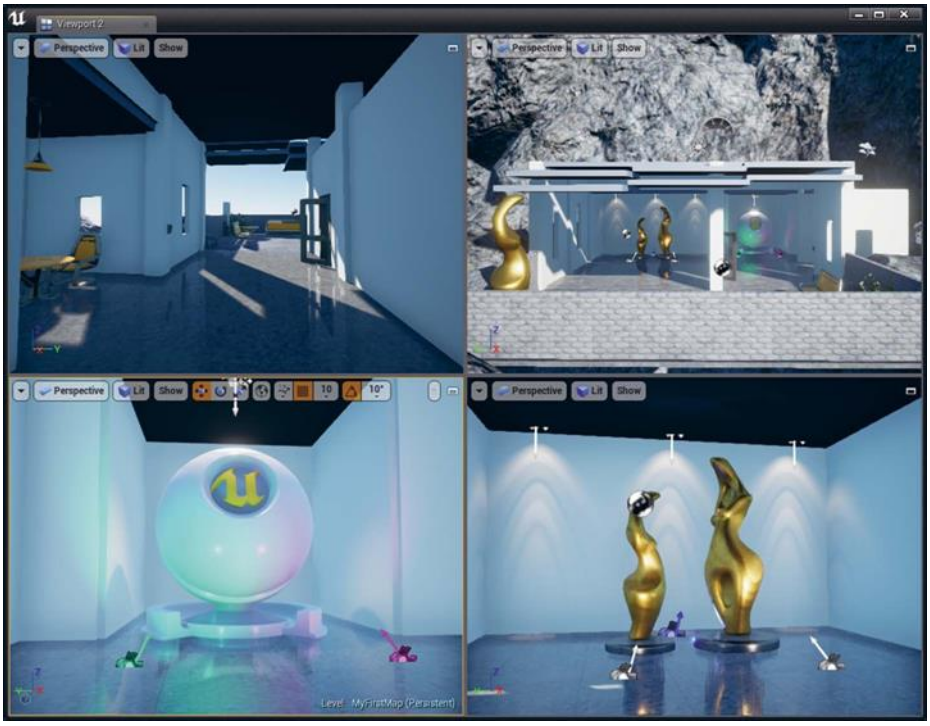

Рисунок 2.11 Финальная сцена в Editor со всеми четырьмя Viewports, настроенными на перспективу, что позволяет видеть уровень во время работы в нескольких проекциях

Заключение

Источники света, материалы и акторов можно перетаскивать и удалять, копировать и перемещать, поворачивать и масштабировать так же легко, как и в любимом 3D-приложении. Построение уровней в UE4 — это весело и интерактивно, и получить отличную настройку освещения очень легко с помощью Atmospheric Fog и Skylight.

Иметь полный контроль над виртуальным пространством и создавать что-то без каких-либо реальных ограничений, кроме простых требований стен и пола.

3 СОЗДАНИЕ ИНТЕРАКТИВНОСТИ С BLUEPRINTS

Вы создали свой первый проект, наполнили мир и установили некоторые источники света и реквизит, чтобы он выглядел красиво. Чтобы сделать проект экстраординарным, нужно добавить ему интерактивности. Далее рассмотрим, как создать первые классы Blueprints, настроить первый Game Mode

и заставить пользователя перемещаться по миру с помощью системы ввода.

3.1 Настройка проекта

Требования проекта включают вид от первого лица и движение в стиле ходьбы. Создание персонажа от первого лица и движения по типу ходьбы — это то, что Unreal Engine может сделать действительно хорошо.

Упражнение 4. Настройка проекта

Сначала создайте павн, Blueprint Class, который получает входные данные, обрабатываемые классом Player Controller, который вы научитесь делать позже. Вы создадите пользовательский класс Game Mode, чтобы определить эти классы как значения по умолчанию для вашего проекта.

С этими объектами вы сможете разместить Player Start Actor, чтобы определить, где ваш пользователь будет спавниться, когда он загрузит этот уровень.

В конце главы ваш павн сможет плавно ходить по уровню, исследуя все вокруг и озираясь.

Процесс настройки для создания Game Mode, в комплекте с Player Controller, отображением входных данных и павном, довольно длительный, вы поймете, как UE4 обрабатывает входные данные и как он ожидает от вас перемещения своих пользователей. Вы также сможете перенести этот контент в новый проект, что позволит вам развивать свою работу по мере добавления возможностей к вашим собственным интерактивным визуализациям.

Нажатие кнопки Play

Нажатие кнопки Play в Editor запускает вашу игру в специальном режиме под названием Play in Editor (PIE), который использует уже загруженные ассеты, чтобы мгновенно запустить мир и заставить вас играть как можно быстрее. Это фантастический способ протестировать ваше приложение без необходимости каждый раз загружать его с нуля.

3.2 Режимы игры

Вы можете выбрать Option справа от кнопки Play, чтобы открыть некоторые дополнительные функции — режимы игры (Play Mode; рисунок 3.1).

Режим по умолчанию — запуск игры на Selected Viewport. Вы также можете запустить его в New Editor Window. Это полезно, если ваш Viewport скрыт или вы хотите протестировать его в определенном разрешении или соотношении сторон.

Выбор Standalone Game запускает сырую сборку вашей игры из командной строки. Это занимает больше времени для запуска, чем PIE (который работает почти мгновенно), но дает более точное представление о том, как ваше финальное приложение будет вести себя вне редактора.

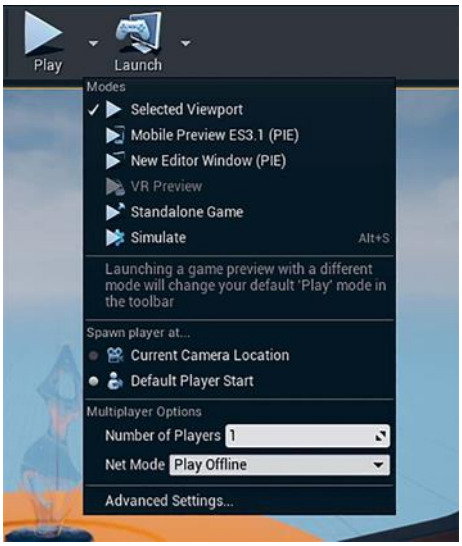

Рисунок 3.1 Параметры PIE

Simulate — это интересный режим, который запускает игру, но не порождает типичный Player Controller и павн, оставляя вам свободу исследовать уровень, поскольку он работает без ограничений, наложенных на обычные павны.

После выбора одного из этих параметров кнопка Play изменится в соответствии с последним выбранным режимом.

Если вы хотите изменить разрешения и другие параметры запуска, выберите Advanced Settings. После этого откроется диалоговое окно Editor Settings, в котором вы сможете изменить их.

Default GameMode

Нажатие кнопки Play на уровне в этот момент спавнит PC по умолчанию и павн, и позволяет вам летать по уровню, используя летающего персонажа с игровым управлением. Это GameMode по умолчанию в UE4.

GameMode — это класс, который определяет, какой Player Controller, Pawn и другие классы будут использоваться при запуске игры.

Обратите внимание, что в этом режиме вы являетесь полноценным пользователем с реакциями на столкновения и физику, в отличие Viewpoint в Editor, где вы рассматриваетесь как призрак. Однако вы также можете летать и двигаться очень быстро. Это не совсем подходящий способ для исследования собственного мира.

Вы должны создать собственный Pawn, PC и GameMode, а затем назначить их своему проекту. После этого вы сможете обойти свой уровень так, как определено в области проекта.

3.3 Создание Pawn

Класс, который представляет ваше физическое присутствие в мире, — это Pawn. Pawn управляет физикой, столкновением и взаимодействием с миром, и другими участниками уровня.

Pawn — это Blueprint-классы. Они содержат компоненты, переменные и Event Graph. Чтобы создать его, выполните следующие действия.

Упражнение 5. Создание Pawn.

1. Выберите Add New в Content Browser и Blueprint Class (рисунок 3.2).

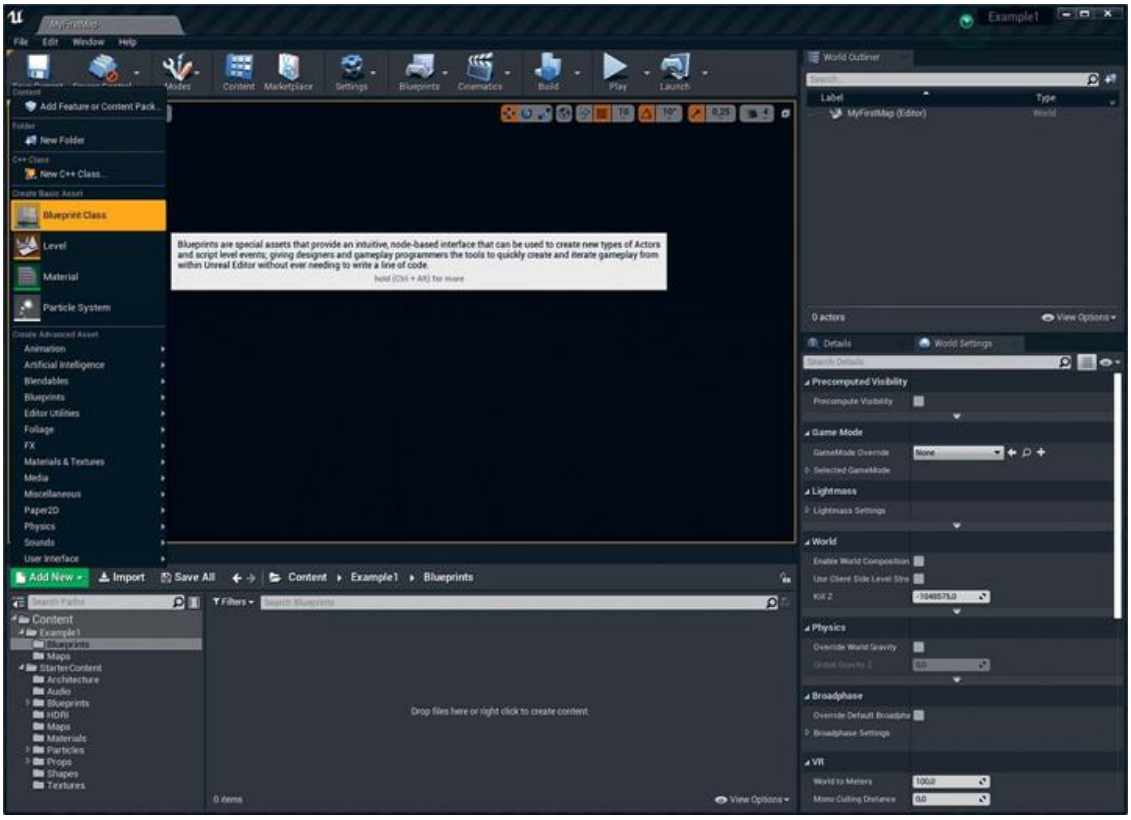

Рисунок 3.2 Создание нового ассета Blueprint в Content Browser

2. Выберите кнопку Character в открывшемся диалоговом окне Pick Parent Class. Класс Character — это улучшенный класс павна, который работает для широкого спектра приложений персонажей от первого и третьего лица, поэтому он идеально подходит для нужд этого проекта и экономит много усилий с вашей стороны.

3. Назовите свой ассет. В данном проекте выбрано имя BP_Interior_Pawn, следуя общепринятым соглашениям об именовании UE4 (рисунок 3.3).

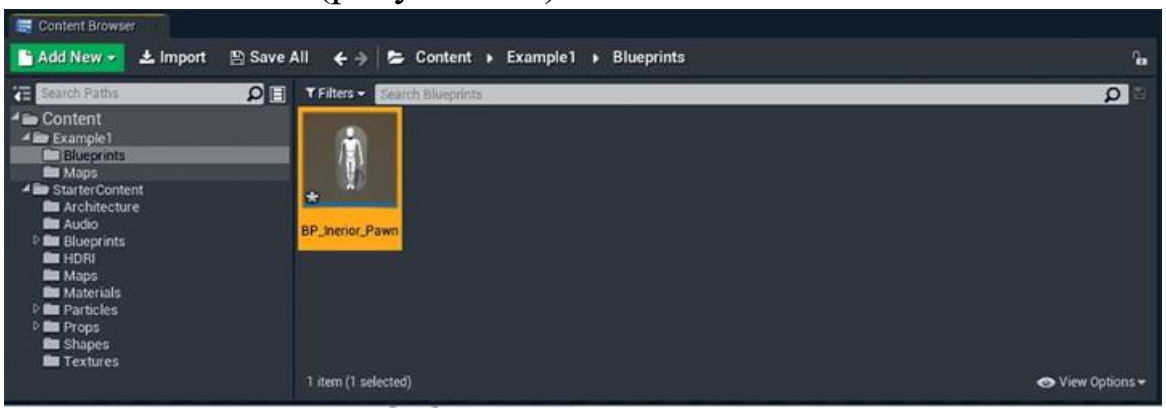

Рисунок 3.3 Созданный и названный BP_Interior_Pawn Blueprint Asset в Content Browser

4. Дважды щелкните по ассету, чтобы открыть Blueprint Editor.

5. Выберите компонент BP_Interior_Pawn из списка Components и убедитесь, что на toolbar выбрана Class Defaults (рисунок 3.4).

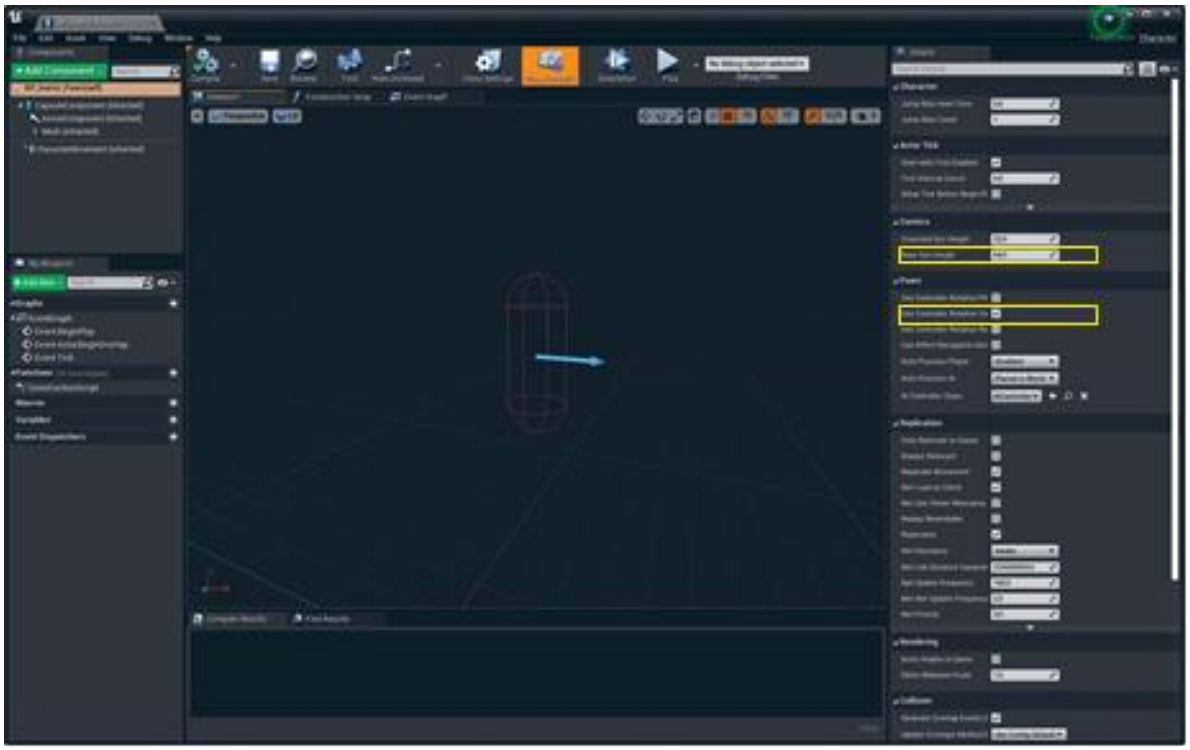

Рисунок 3.4 Значения по умолчанию класса Pawn, отображаемые в Blueprint Editor

3.4 Настройка View Height пользователя

Чтобы установить точку зрения пользователя на разумную высоту, вам нужно настроить Base Eye Height вместе с Half Height у Capsule Component.

Уровень глаз пользователя устанавливается путем добавления свойства Base Eye Height со свойством Half-height у Capsule Component. Вам нужно настроить эти параметры как для достижения нужной высоты, так и для столкновения пользователя.

Регулирование Capsule Component

Pawn использует Capsule Collision Component для имитации тела пользователя в мире. Вы можете увидеть Capsule Component на вкладке Viewport и в списке компонентов (рисунок 3.5 далее в этой главе).

Выберите Capsule Component из списка компонентов, чтобы получить доступ к двум его переменным: Radius и Half-height. Как следует из названия, полувысота представляет собой расстояние от пола до середины Capsule Collision Component. Это значение по умолчанию составляет 88 см, или 176 см (около 5’9”) от пола до верха капсулы. Это низко для большинства людей, но лучше ошибиться немного, чем позволить голове пользователя зацепиться за дверные проемы и светильники. Вы можете сохранить это значение по умолчанию.

Настройка Base Eye Height

UE4 вычисляет высоту взгляда персонажа, добавляя свойство Half-Height Capsule Component к Base Eye Height, чтобы получить окончательную высоту.

Значение базовой высоты взгляда по умолчанию 64 дает высоту всего 152 см, что кажется слишком низким для большинства людей. Средняя высота глаз у мужчин составляет около 175 см, а у женщин — около 160 см. Цель на 168 дает почувствовать разницу, оставляя ее в 80 см Base Eye Height.

Чтобы вернуться к свойствам класса Character по умолчанию, щелкните на корневой компонент (BP_Interior_Pawn) в списке компонентов и установите Base Eye Height равной 80.

Использование Controller Rotation Yaw

Следующая настройка в классе нужна, чтобы знать о том, как будет определяться поворот взгляда пользователя. Yaw — это вращение пользователя по оси Z, контролирующее, как он поворачивается влево и вправо.

Movement component не управляет вращением; это делает Player Controller. Кроме того, поскольку вы не хотите, чтобы вся капсула качалась вверх и вниз, когда вы смотрите вверх и вниз, вы используете только Yaw (вращение по оси Z), оставляя Pitch

для применения к камере, которая является родительским павном.

Наличие двух компонентов, независимо обрабатывающих pitch и yaw, позволяет избежать интерактивной проблемы камеры, обычно известной как Gimbal Lock, когда камера не поддерживает горизонтальную стабильность при вращении.

Включение Controller Rotation Yaw автоматически соотносит yaw объекта Pawn с Control Rotation Parameter y Player Controller.

3.5 Настройка Movement Speed

Компонент CharacterMovement отвечает за перемещение персонажа по сцене и управление состояниями движения. Здесь находятся настройки по умолчанию, относящиеся к движению персонажа, такие как скорость ходьбы, режим движения по умолчанию и так далее.

Найдите компонент CharacterMovement на панели Components и выберите его, чтобы просмотреть свойства по умолчанию на панели Details (рисунок 3.5). Вы можете фильтровать панель Details с помощью символа глаза рядом со строкой поиска в верхней части панели Details.

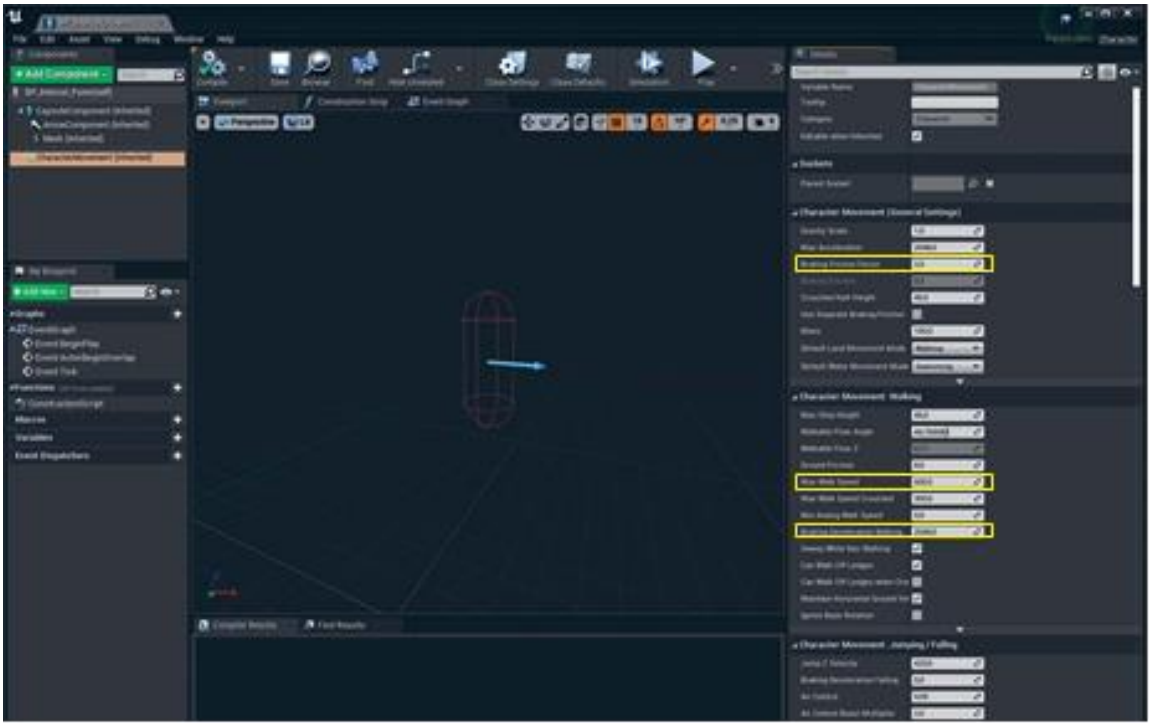

Рисунок 3.5 Измененные свойства CharacterMovement Component

Вам не так уж много нужно изменить для этого простого проекта, но обратите внимание, что варианты обширны, и вы можете использовать их для создания тонны различных типов движений, от полета до падения, плавания и лазания.

Max Walk Speed

По умолчанию пользователь должен двигаться со скоростью 600 см в секунду (13 миль в час). Установите этот параметр где-то в диапазоне 150–175 для медленной прогулки.

Braking Friction и Deceleration

Для замедления персонажа доступны две настройки, и обе по умолчанию имеют высокие значения (предположительно, для игр с быстрым действием, где люди ходят со скоростью 6 метров в секунду).

Установите Braking Friction Factor 1, a Braking Deceleration Walking — 0.0.

Это позволяет пользователю прийти к плавной остановке, а не к резкой, необходимой в большинстве игр для точного управления. Это личное предпочтение и демонстрация того, как вы можете действительно настроить «ощущение» вашего персонажа, используя класс Character.

Теперь, когда вы создали и изменили свой класс спавна, сохраните его, прежде чем продолжить. Помните, что вновь созданные ассеты не записываются автоматически на диск, поэтому вы должны сохранить их вручную.

3.6 Привязка ввода данных

Привязка ввода данных (Input Mappings) — это параметры всего проекта, которые позволяют настраивать общие события ввода, такие как щелчки мыши или нажатия клавиш, и получать к ним доступ из Blueprints.

Чтобы получить доступ к Input Mappings, перейдите в меню Edit и выберите пункт Project Settings.

Выберите Input в левом столбце (рисунок 9.6), чтобы просмотреть Input Mappings.

3.7 Action и Axis Mapping

Action Mappings — это одноразовые события, вызванные событиями одного действия, такими как нажатие кнопок и щелчки мыши. Эти сопоставления срабатывают за один тик.

Axis Mappings представляют собой события, которые могут иметь числовое значение и способны вызываться от кадра к кадру. Это было бы похоже на то, как быстро вы хотите двигаться вперед или повернуть налево или направо. Их называют Axis Mappings, потому что они традиционно относились к игровым джойстикам, которые определяли их движение по своей оси.

В современных интерфейсах это может быть количество пикселей, перемещенных мышью в последнем тике, нажатой и удерживаемой клавишей или положением аналогового джойстика на геймпаде.

Для перемещения и вращения пользователя вы будете использовать исключительно Axis Mappings.

Настройка Mappings (отображений)

Массив Axis Mappings состоит из списка Mappings. Каждый Mapping имеет Label и массив входных данных, каждый из которых имеет значение Scale.

Label — это то, на что ссылается Mapping в Blueprints. Вы можете назвать этот Label как угодно: Walk, Turn, EatShrimp. Нет никакого установленного стандарта. Все, что вы назначаете этому полю, становится доступным в качестве Input Event и переменной Axis Value, которая легко доступна в Player Controller.

Каждый именованный Input Axis может иметь несколько входов. Например, Mapping «MoveForward» может иметь назначенные как клавиши W, так и S. У клавиши S Scale –1, тогда как у W Scale +1. Это означает, что, когда это событие вызывается пользователем, нажимающим W, оно возвращает

значение 1, а пользователем, нажимающим S, возвращает значение –1. Таким образом, вы можете сгруппировать входные данные, чтобы избежать слишком большого количества событий в вашем игровом коде.

Прежде чем продолжить, настройте Input Mappings так, чтобы они выглядели как на рисунке 3.6. Настройки в Project Settings проекта автоматически сохраняются в конфигурационных файлах проекта. Вам не нужно нажимать Save, когда вы закончите; вы можете просто закрыть окно настроек.

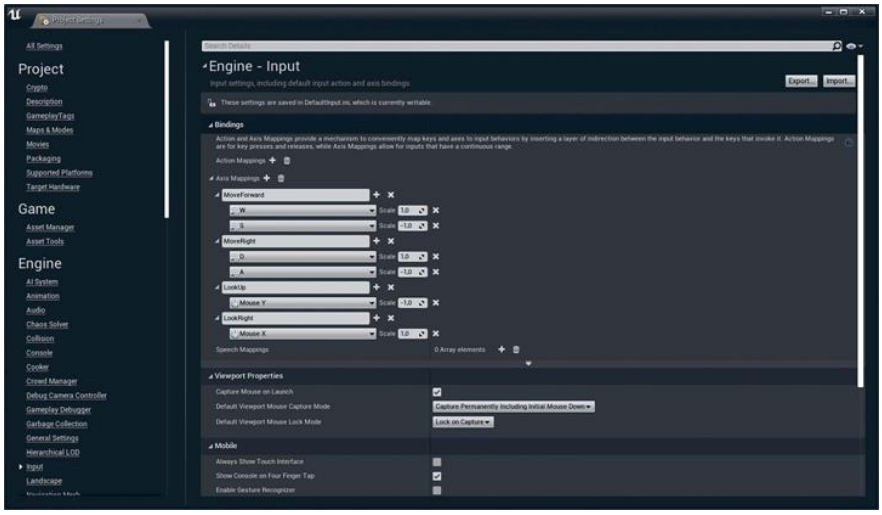

Рисунок 3.6 Диалог настройки Input Mapping

3.8 Input Device Flexibility

Еще одним преимуществом такой настройки ввода является возможность использовать ввод с нескольких устройств в едином виде. Нажатие Enter может вызвать то же событие, что и нажатие кнопки X на геймпаде или щелчок правой кнопкой мыши. Это полностью зависит от вас.

Из-за этой гибкости UE4 не поставляется с настройками по умолчанию. Вы можете сделать Input Mappings точно в соответствии с потребностями вашей системы ввода. Эти настройки хранятся в файле Saved/Config/DefaultInput.ini вашего проекта, и вы можете импортировать и экспортировать его с помощью кнопок в интерфейсе Project Settings или путем копирования и вставки текстового содержимого из других входных файлов.

Создание Player Controller Class

Теперь, когда у вас есть павн и настроен Input Mappings, вам нужно собрать их вместе. Хотя вы можете поместить свою логику ввода и движения непосредственно в класс павна, это не лучший шаблон проектирования.

Для этого у вас есть Player Controller. Как следует из названия, одной из основных функций Player Controller является обработка входных данных от пользователя. Помните, что при запуске приложения UE4 всегда есть Player Controller, поэтому это отличное место для размещения кода, который всегда должен работать, например, обработка входных данных.

Создайте свой Player Controller точно так же, как Pawn и GameMode, которые вы сделали раньше, щелкнув правой кнопкой мыши в Content Browser и выбрав пункт Blueprint в меню Create Basic Asset.

Выберите класс Player Controller в окне Pick Parent Class и назовите вновь созданный ассет BP_UE4Viz_PlayerController.

Обязательно сохраните свой ассет на диске в первый раз.

Добавление Input с Blueprints

Теперь, когда вы определили все входные значения, которые вам нужны, вы можете начать использовать их для управления событиями в вашей игре с помощью сценариев Blueprint. Начнем с открытия Blueprint Editor. Вы делаете это, как и во многих редакторах в UE4, просто дважды щелкнув на свой ассет Player Controller в Content Browser.

Когда ваш PC открыт, вам нужно добраться до Event Graph чтобы начать написание кода обработки входных данных. Если в окне Blueprint Editor отсутствует Event Graph и вверху отображается сообщение о том, что это только Data Only Blueprint, вам нужно нажать кнопку Open Full Blueprint Editor, чтобы увидеть полный интерфейс Blueprint Editor. Когда он будет доступен, перейдите на вкладку Event Graph.

Добавление Axis Events

Вам нужно определить Input Axis Mappings, которые настроили ранее. Когда срабатывает один из Input Axis Mappings

(обычно при выполнении пользователем выбранного действия), оно запускает событие, доступ к которому можно получить из Blueprints. Это называется Axis Event, и оно передает одну переменную: Axis Value.

Axis Value представляет значение ввода, умноженное на Scale, которое вы установили в Input Mappings dialog (см. рисунок 9.6). Итак, нажатие на W запускает input event, возвращающий 1.0 в течение каждого Tick удерживания кнопки, и нажатие S также запускает input event, но возвращает значение –1.0 потому, что Mapping устанавливает Scale value –1. Также эти значения являются совокупными, если Player нажмет W и S одновременно, input axis value вернет 0 потому, что вводы отменяют друг друга.

Клавиши на клавиатуре, конечно, цифровые и могут вернуть только 1 или 0. Тем не менее Axis Value могут работать с аналоговым дивайсом, таким как геймпад, который может вернуть любое значение между –1 и 1 по каждой оси, или с вводом мышью, который возвращает количество пикселей, пройденных мышью с последнего момента считанного ввода (иногда называемых input delta, или разницей между двумя выборками). Эта настройка позволяет множеству различных типов вводов использовать одну и ту же координатную ось, и соответствующее событие, упрощая код обработки ввода.

Для добавления Axis event в ваш Player Controller, нажмите правую кнопку мыши в Event Graph’s Graph Editor и найдите Axis name, определенное Settings (рисунок 3.7). Просто начните печатать, и список будет отфильтрован динамически.

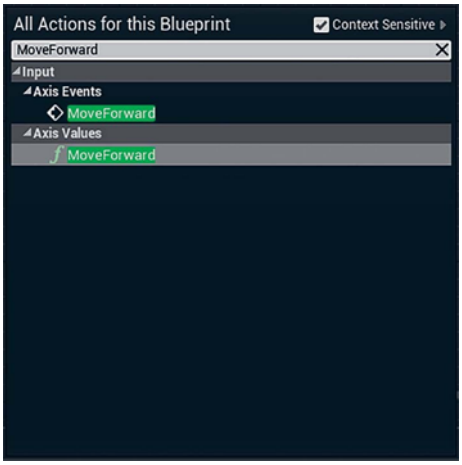

Рисунок 3.7 Поиск Move Forward Axis Event с использованием контекстного меню, доступного при правом клике в Blueprint Event Graph

Добавьте события для каждого из ваших движений Axis Mappings: MoveForward и MoveRight (рисунок 3.8). Добавьте еще два для Rotation Mappings: LookUp и LookRight.

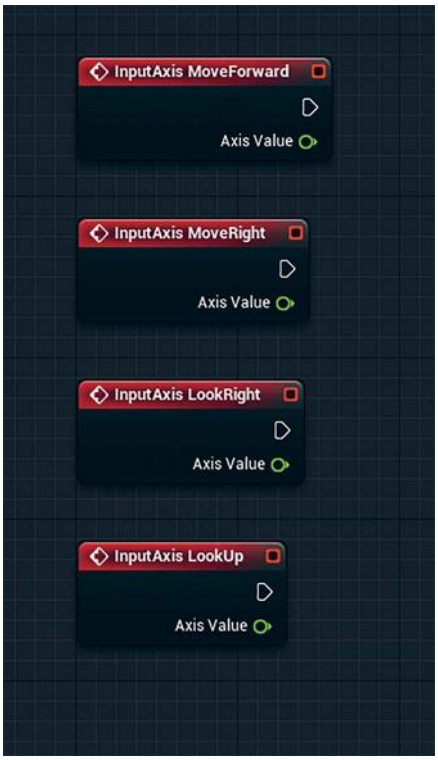

Рисунок 3.8 Добавленные Axis Events

Обратите внимание на вывод Axis value на рисунке 3.8. Оно возвращает значение с плавающей точкой, которое представляет

Player input value масштабируемое Input Scale назначенный в Input Mapping.

Поворот взгляда

Вращать поле обзора легко. Благодаря Character Class, от которого вы наследуете ваш Pawn, вам достаточно вращать Player Controller (помните, что ваш Player Controller управляет полем зрения), и взор будет следовать за ним.

Поскольку это часто используется, UE4 предлагает несколько быстрых решений. Add Yaw Input и Add Pitch Input две готовые функции, принимающие входное значение и записывающие изменение значения вращения в переменную PC’s Control Rotation.

Выполните следующие шаги для создания этих узлов.

1. Нажмите правой кнопкой мыши в the Event Graph и найдите Add Yaw Input и Add Pitch Input.
2. Подключите вывод Exec Out (белая стрелка в правой части узла) от InputAxis LookRight Event и подключите его к входу Add Yaw Input узла Exec In (белая стрелка в левой части узла), перетаскивая из одного в другой. Вы можете перетащить выходы в любом направлении.
3. Соедините выход Axis Value вершины InputAxis LookRight с контактом In Val вершины Add Yaw Input.
4. Повторите эти действия для LookUp axis и функции Add Pitch (рисунок 3.9).

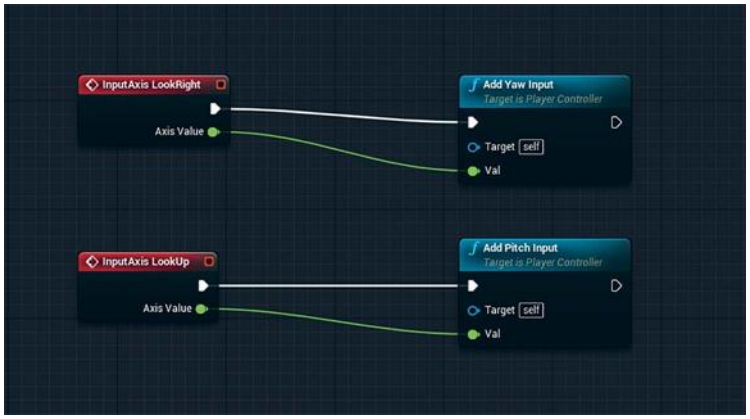

Рисунок 3.9 Завершенный Rotation Input скрипт

3.9 Передвижение пользователя

Движение пользователя немного сложнее настроить, нежели вращение обзора. Player Controller должен сообщить объекту Pawn о движении. Чтобы сделать это, вам нужно использовать модель связи Blueprint для передачи входных данных героя в Pawn, чтобы он мог двигаться.

Ссылка на Pawn

Когда Player нажимает на одно из Axis Mappings и запускает эти события, вы хотите, чтобы Pawn двигался. Чтобы сделать это, вам необходимо связаться с ним и первым шагом получить Reference на Pawn.

Базовый PC-класс, от которого вы наследуете ваш PC, содержит переменную Player Character, заполняемую автоматически с Reference на Character Pawn, которой владеет в данный момент. Это позволяет вам напрямую получить доступ к вашему Character и легко добавить ввод движения.

Вы можете получить ссылку на Player Character, щелкнув правой кнопкой мыши в Event Graph и найдя Character во всплывающем Action List. Выберите функцию Get Player Character.

Is Valid

Так как Player Controller может обладать практически любым типом Actor в мире, переменная Player Character может вернуть none, если она не контролирует Actor или Pawn, наследуемый от Character Class.

Используйте Is Valid branch, чтобы убедиться, что этого не произойдет (рисунок 3.10). Сценарий будет продолжаться только в том случае, если значение Player Character is valid и не выдает ошибку при обращении к нему.

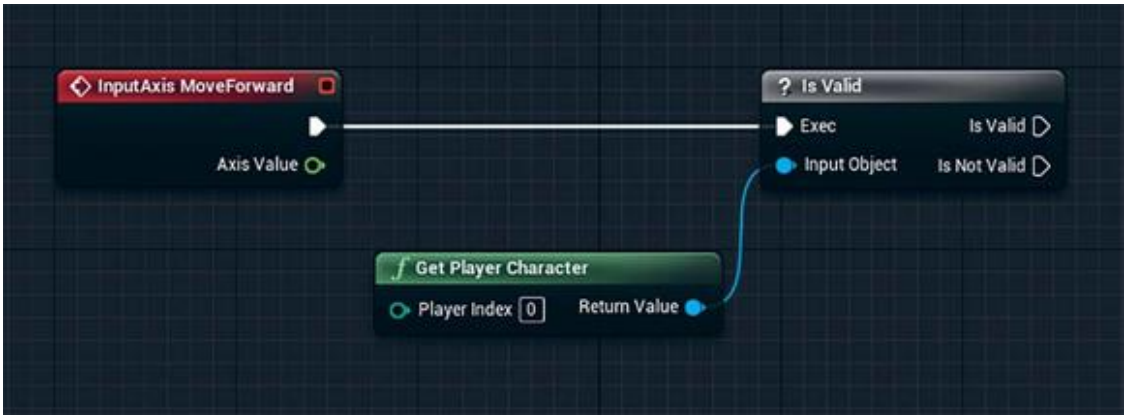

Рисунок 3.10 Получение ссылки на Player Character и проверка его валидности

Всегда проверять, когда «получаете» другие Actor или Objects, которые ненадежно существуют в мире.

Add Movement Input

Character Class уже имеет функции для перемещения Player по сцене с помощью Movement Component. Функция называется Add Movement Input, но вам нужно получить доступ к ней иначе, чем вы делали.

До сих пор вы нажимали правой кнопкой мыши в Viewport и использовали контекстное меню для создания узлов. В этом меню отображаются только те узлы, которые доступны для Blueprint, в котором он находится. Functions, Events и Variables, которые хранятся в других Blueprints, должны быть доступны по ссылке.

Для этого протяните провод от синего контакта Return Value из Get Player Controller в Graph Editor и отпустите. Появится контекстное меню, заполненное Functions, Events и Variables, связанными с указанным классом (рисунок 3.11). Обратите внимание, что в контекстном меню написано Actions taking a(n) Character Reference. Это говорит о том, что вы успешно ссылаетесь на другой Blueprint Class.

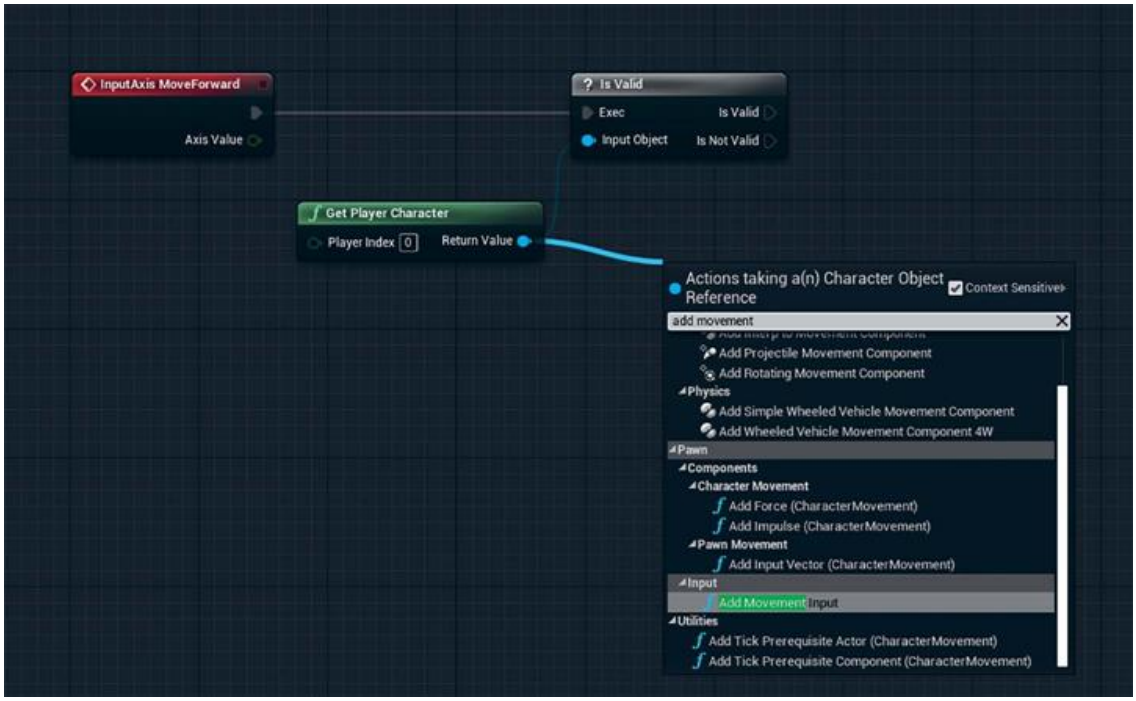

Рисунок 3.11 Добавление Add Movement Input из Character Class Reference

Найдите Add Movement Input и нажмите на него, чтобы разместить в Event Graph, подключив, как показано на рисунке 3.12

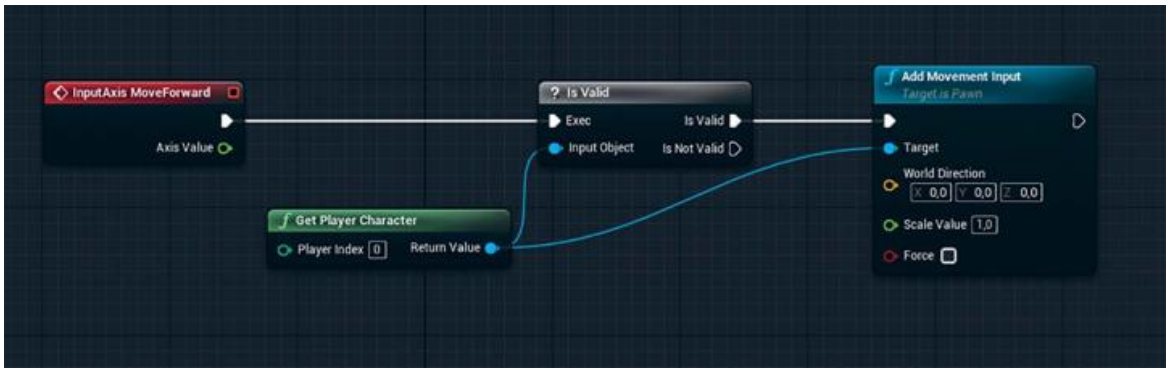

Рисунок 3.12 Размещенная функция Add Movement Input

На рисунке 3.12 обратите внимание на синий провод, соединяющий узел Get Player Controller с Target Add Movement Node. Этот провод представляет собой ссылку на Player Character.

Get Forward и Right Vectors

Функция Add Movement Input использует мировой вектор для перемещения вашего персонажа. Этот вектор меняется в

зависимости от того, куда направлен Player, поэтому необходимо выяснить и использовать это для определения направления движения.

Вот где функции Get Actor Forward Vector и Get Actor Right Vector применяются. Они преобразуют world rotation в нормализованные векторы XYZ для каждого направления.

Как и в случае с узлом Add Movement Input, вы должны вытянуть ссылку из Return Value узла Get Player Character's, чтобы получить доступ к списку функций. Найдите узлы и разместите их в Event Graph (рисунок 3.13). Вы можете увидеть ссылки к Components из Character Class; убедитесь, что выбираете корневой компонент, как показано.

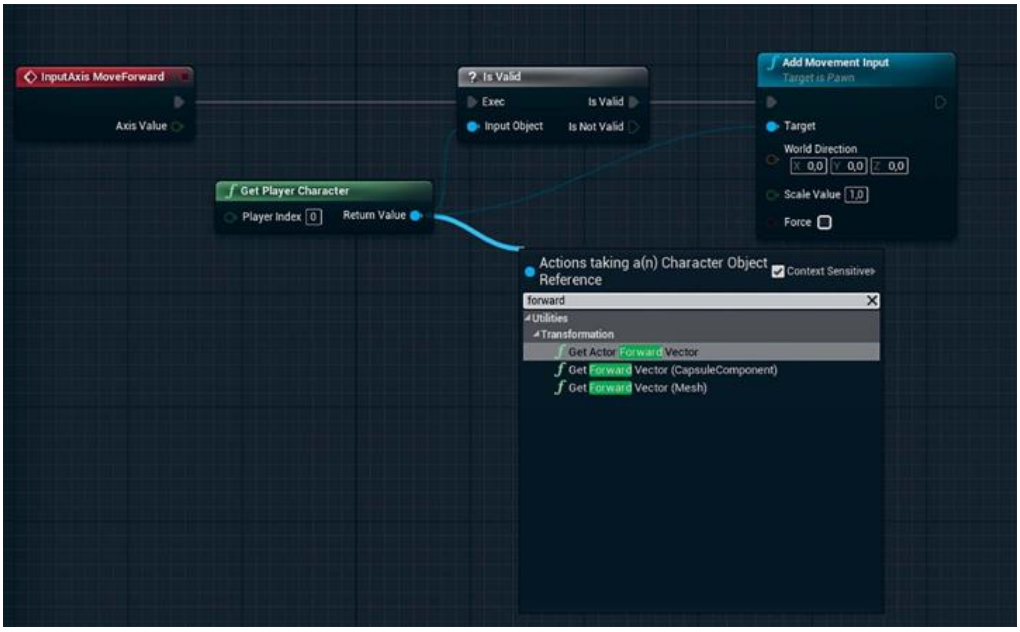

Рисунок 3.13 Добавление функции Get Actor Forward Vector

Scaling Input

Не вдаваясь сильно в векторную математику, вы можете просто умножить вектор (три числа с плавающей запятой, такие как X, Y, Z) по Axis Value из InputAxis Events, масштабируя каждое направление на введенную сумму.

Перетаскивание соединения из Return Value узла Get Actor Forward Vector в Event Graph и отпускание, позволит вам увидеть методы, доступные для Vector Reference. Найдите

*(символ звезды), чтобы увидеть доступные функции умножения и выберите vector * float из списка (рисунок 3.14).

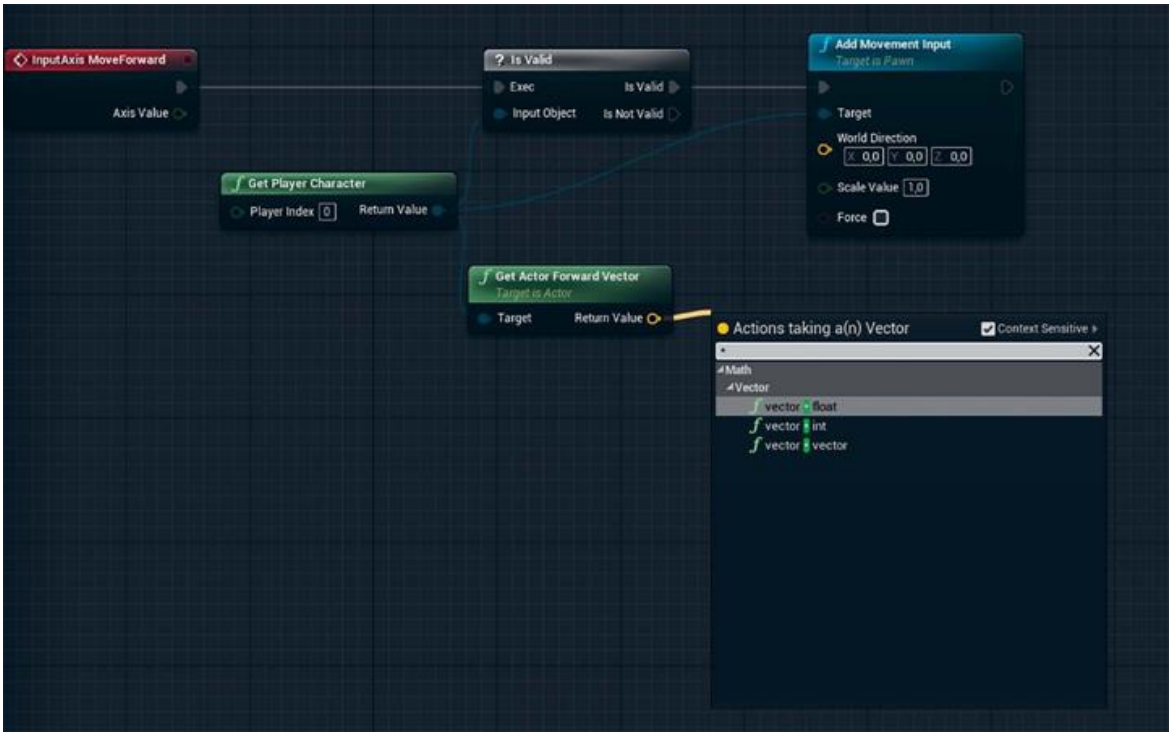

Рисунок 3.14 Добавление математического узла vector float

Затем вы можете подать масштабируемое значение Vector на входные контакты World Direction функции Add Movement Input, завершив ваш код движения вперед. Вам нужно повторить этот процесс для Move Right InputAxis Event. Единственное отличие состоит в том, что вы заменяете функцию Get Actor Forward Vector на узел Get Actor Right Vector (рисунок 3.15).

Рисунок 3.15 Завершенный сценарий movement input

Это все, что нужно сделать, чтобы получить ваш Player's input для вращения PC и перемещения Player's Pawn. Теперь нужно заставить игровой мир использовать ваши Classes, когда вы нажимаете Play, и вы будете готовы к работе.

3.10 GameMode

GameMode Class — это место, в которое заглядывает движок при загрузке уровня (Level), чтобы определить, какой выбрать Player Controller, Pawn и другие необходимые классы.

Упражнение 6. Создание GameMode.

Создайте новый GameMode Asset, выполнив следующие действия.

1. Нажмите Add New button в Content Browser и выберите Blueprint Class
2. Выберите Game Mode Base Class и назовите ваш Asset BP_UE4Viz_Interior_ Game. Нажмите Enter для создания Asset.
3. Откройте ваш только что созданный GameMode Asset с помощью двойного нажатия в Content Browser. Вы увидите список Classes, который можете определить для GameMode. В этом случае назначьте Pawn и Player Controllers, которые вы создали (рисунок 3.16).

4. Закройте Blueprint Editor и сохраните ваш прогресс, выбрав File > Save All.

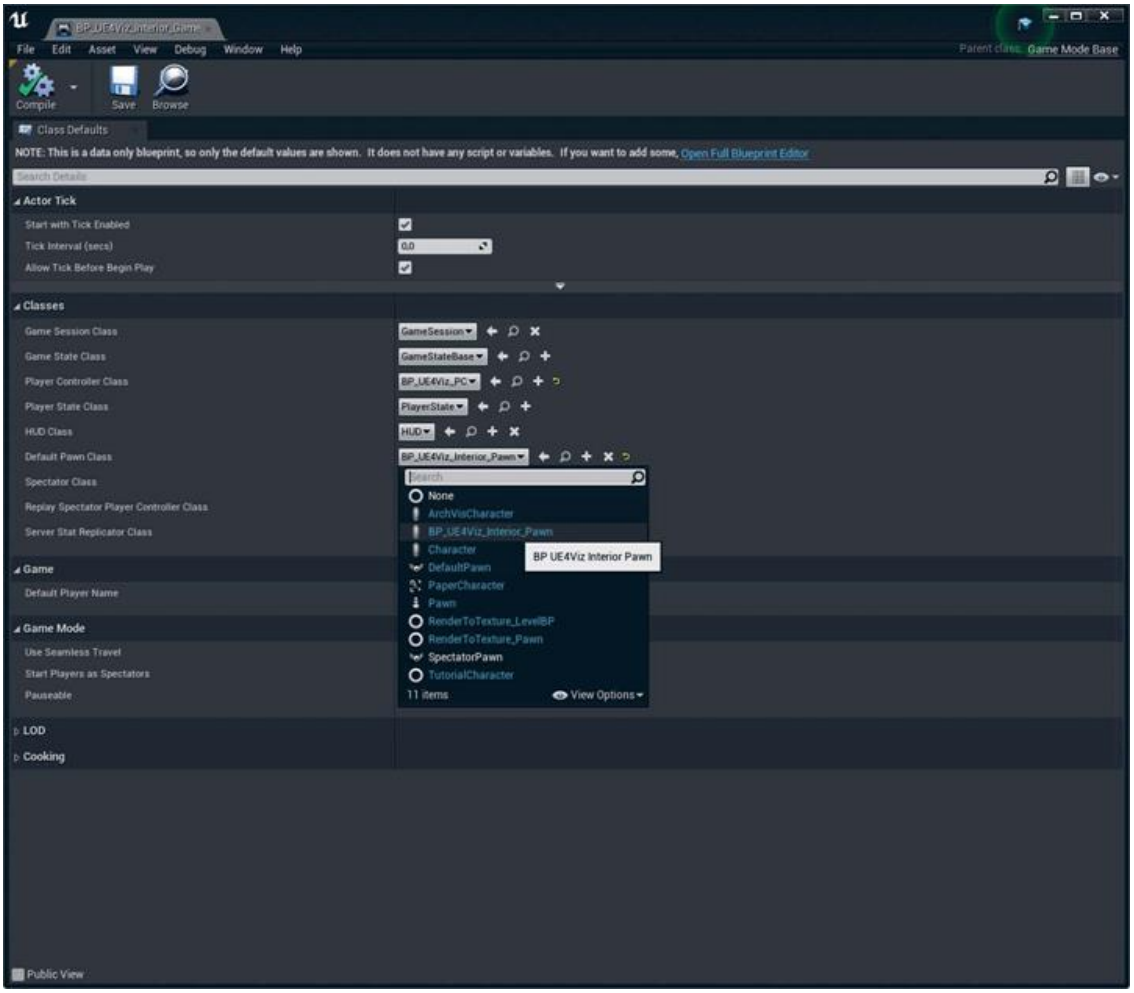

Рисунок 3.16 Назначение Pawn и Player Controller в GameMode Class

Назначение GameMode проекту

У вас есть два способа определить, какой GameMode Class применяется в UE4 Level: как для всего проекта по умолчанию, используя диалоговое окно Project Settings, или через переопределение для Level. Удобнее запускать новый GameMode на любых Level, которые создадите в этом проекте, установите его как проект по умолчанию.

Откройте диалоговое окно Project Settings, выбрав Edit > Project Settings (рисунок 3.17). Edit menu доступно практически из всех окон Editor.

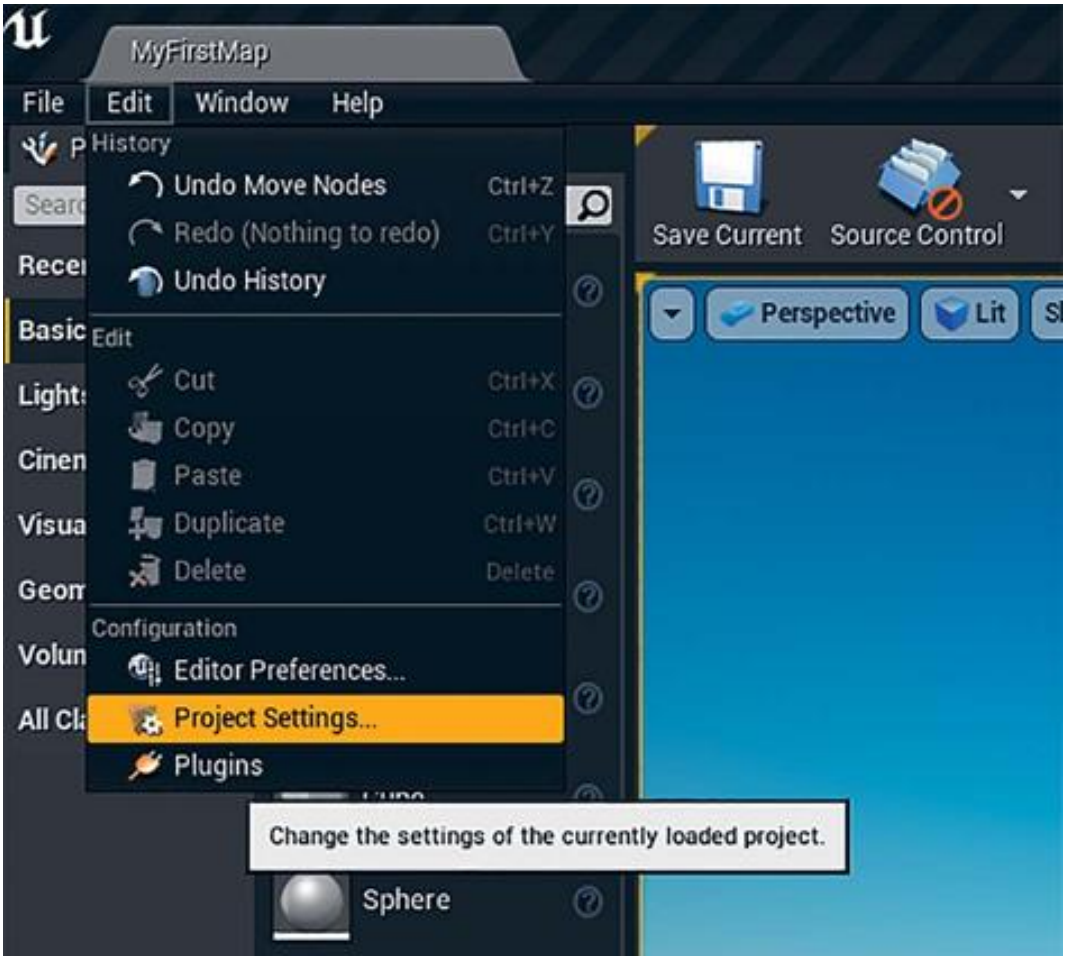

Рисунок 3.17 Переход к диалоговому окну Project Settings

Выберите Maps & Modes под заголовком Project. В разделе Default GameMode выберите только что созданный BP_UE4Viz_Game Class (рисунок 3.18).

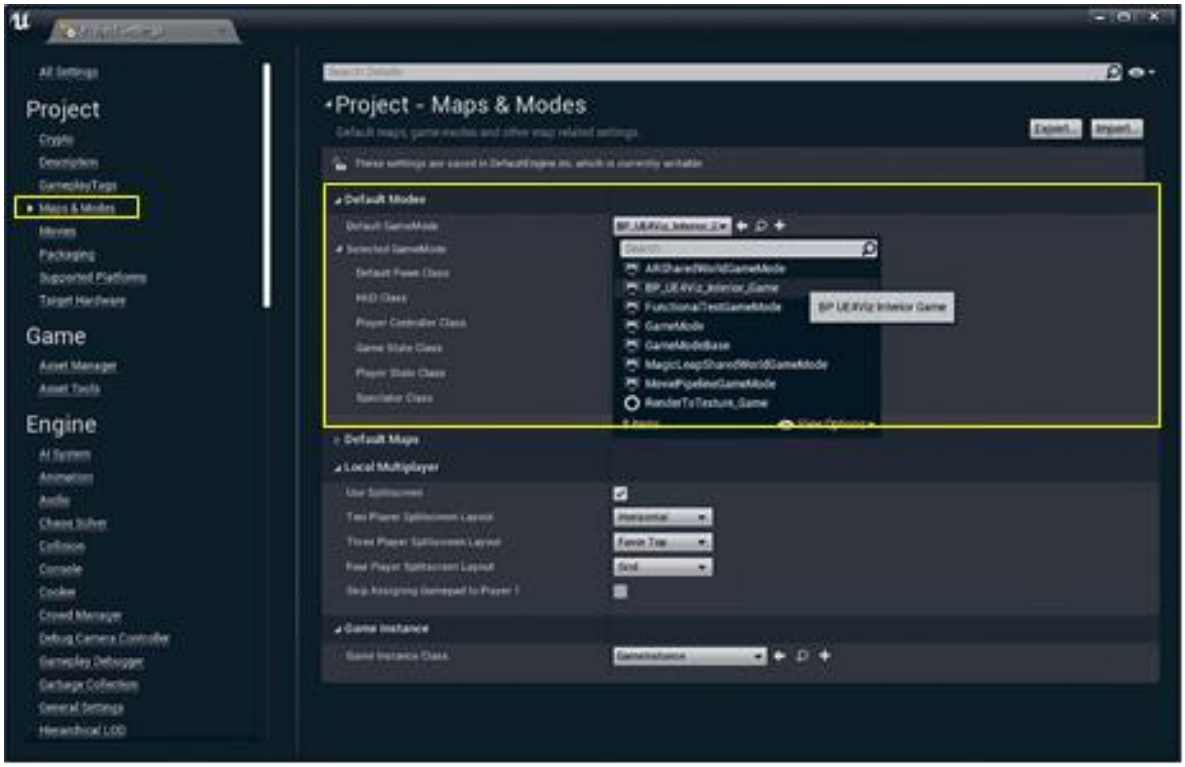

Рисунок 3.18 Назначение проекту по умолчанию BP_UE4Viz_Game Class

Размещение Player Start Actor

Есть одна вещь, которая важна и которую легко забыть, — это Player Start Actor.

Без Player Start Actor проект не будет знать, где появится ваш Pawn class, когда она загружена в качестве standalone или packaged build. Где вы in-Editor и используете PIE без Player Start, ваш Pawn появится по умолчанию в текущей позиции камеры в активном Viewport. Можно легко забыть разместить Player Start так, как ваш Level будет выглядеть рабочим, когда он действительно установлен некорректно.

Для размещения Player Start Actor перетащите его из Class Browser на панель Place Mode (нажмите Shift+1), как показано на рисунке 3.19. Синяя стрелка (3D-стрелка, которая является частью Player Start Actor, а не желтая стрелка) указывает, в каком направлении будет смотреть Player, когда появится.

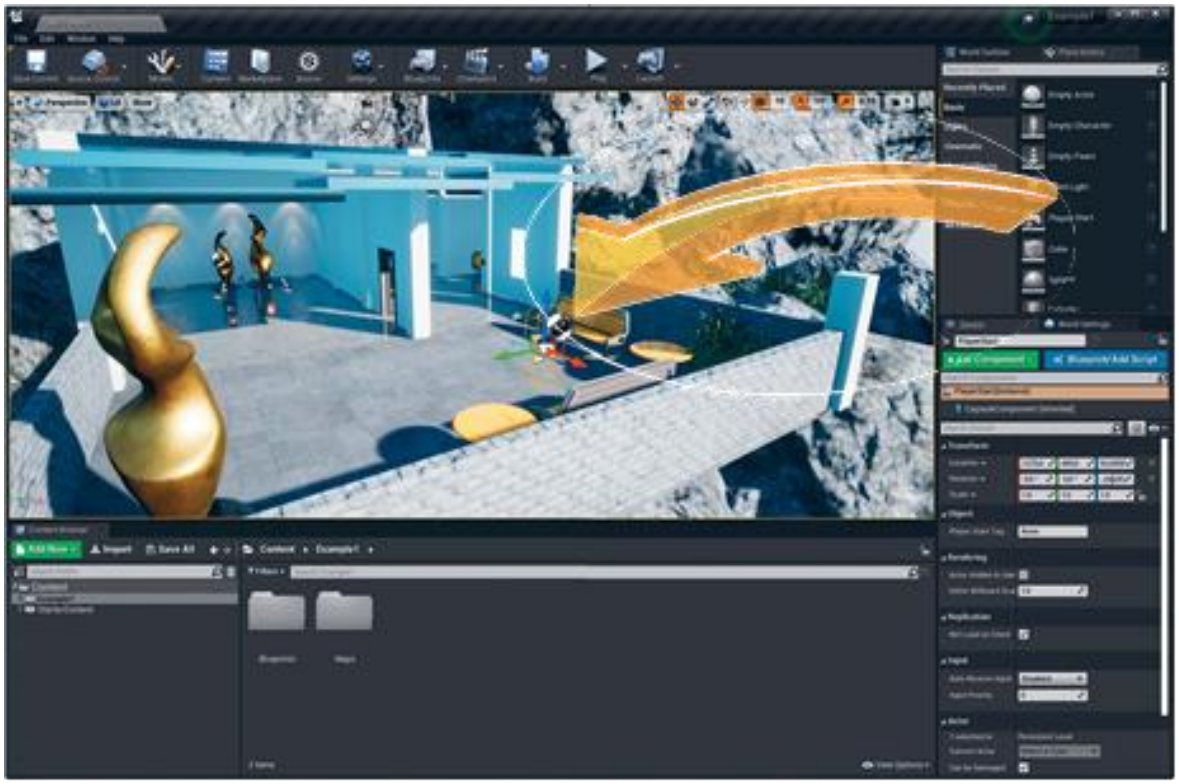

Рисунок 3.19 Поместите Player Start Actor и поверните его лицом к сцене

Теперь вы можете нажать кнопку Play и появиться в игре в качестве вашего пользовательского Character Pawn. Вы сразу же должны заметить, что привязаны к земле с помощью физики и не можете проходить сквозь стены.

Когда вы закончите ходить и захотите вернуться в Editor, вы можете просто нажать Esc на вашей клавиатуре для завершения сессии.

Убедитесь, что сохранили весь ваш прогресс.

Заключение

Вы создали пользовательский UE4 GameMode с Player Controller, Pawn и пользовательским Input Mappings и настроили проект для использования новых Classes в качестве проекта по умолчанию.

Эти Classes будут служить основой для добавления новых функций и дальнейшей работы в UE4 в соответствии с потребностями вашего проекта.

4 УПАКОВКА И РАСПРОСТРАНЕНИЕ

Теперь, когда у вас есть уровень и связанные с ним объекты Pawn, Player Controller и Game Modes, пришло время подготовить его к запуску в качестве отдельного приложения. Это, как правило, называется упаковкой. После упаковки вы сможете поделиться вашим приложением, как любым другим устанавливаемым приложением. Эта глава познакомит вас с упаковкой приложения.

Упаковка против сборок из Editor

Когда вы запускаете приложение с помощью кнопки Play в Editor, вы просто запускаете другое окно в Editor, а не отдельное приложение. Editor использует Assets и Maps из вашей папки Content и то, что уже загружено в оперативную память, чтобы заполнить мир, и код Editor запускает симуляцию. Это позволяет очень быстро запускать, тестировать и выполнять итерации, но для этого требуется установить UE4 Editor целиком.

Чтобы создать приложение под конкретную платформу и облегчить его распространение, ваш контент должен быть упакован. Упаковка обрабатывает контент и создает отдельное приложение, которое легко распространять как любую другую игру или программу.

Вы также можете протестировать свое приложение в Standalone Mode, выбрав эту опцию в меню параметров Editor на панели инструментов. Это запустит загруженный в данный момент уровень в отдельном процессе, который больше похож на упакованную сборку, но это не так. Важно собирать и тестировать свои приложения, поскольку ошибки и несоответствия могут возникать между Editor и упакованными сборками.

Упаковка проекта

В общих чертах, упаковка берет ваш контент и код игры, оптимизирует их и создает исполняемое приложение для выбранной вами платформы.

Упакованные сборки пытаются как можно меньше загружать оперативную память. Это обеспечивает

максимальную производительность и совместимость. Однако это может привести к проблемам, особенно с Level streaming и загрузкой Assets в реальном времени. Для простых проектов это не столь важно, но крупные проекты должны регулярно тестироваться с помощью упакованной сборки, чтобы избежать проблем.

4.1 Готовка контента

Готовка контента (Content Cooking) также проходит через несколько процессов: компиляции всех шейдеров, сжатия всех всех карт текстур и исправления всех редиректов в проект. Подобный подход избавляет пользователей от ожидания завершения всех этих процессов во время загрузки приложения.

Готовка может занять некоторое время и серьезно загрузить CPU и RAM, поэтому запускайте ее на мощной машине. Чем больше ассетов, материалов и текстур содержит ваш проект, тем больше времени уйдет.

В завершение готовки все обработанные ассеты и классы копируются в файл .pak. Этот файл .pak предназначен для очень быстрого чтения даже на самых медленных дисках, что позволяет приложению загружаться быстро.

4.2 Развертывание

Упаковка берет готовый (cooked) контент, объединяет его с бинарными файлами движка и помещает их вместе в одну папку для распространения. Этот последний шаг создает отдельное приложение, которое можно запустить без установки редактора.

4.3 Настройки упаковки

UE4 может генерировать упакованные проекты для множества платформ и систем (iOS, Windows, Mac и т. д.). С такой гибкостью приходит несколько вариантов.

Платформы

Целевая платформа — это комбинация аппаратного (hardware)/программного обеспечения (software), для которой вы пытаетесь создать исполняемый файл проекта.

Вы можете упаковать ваш проект для множества платформ, поддерживаемых UE4. От мобильных девайсов до Windows, Mac и Linux до игровых консолей, UE4 может упаковать ваш проект для практически всех популярных устройств и операционных систем.

Каждый будет иметь собственную оптимизацию, методы ввода и другие ограничения, которые вы должны учитывать. Это методическое указание фокусируется только на настольных платформах (Mac и Windows).

4.4 Конфигурация сборки

Другим важным решением, которое вы должны принять, является конфигурация сборки. Два основных варианта — Shipping и Development.

Наибольшим различием между сборками Shipping и Development является доступ к инструментам отладки, введению логов и командной строке. Сборка Development позволяет это, в то время как сборка Shipping лишена этих возможностей.

Обычно используется Development, если не планируется распространять пакет для публики в качестве широкого выпуска. Консольные команды, введение логов и отладочная информация полезны, если нужно отладить какую-нибудь проблему.

Сборка Shipping хорошая, когда вы хотите распространять ваш пакет публично и не хотите, чтобы он имел доступ к Console.

Как запустить упаковку

Легче всего вы можете упаковать ваш проект через Editor. Просто выберите Package Project из File menu. Это меню содержит несколько настроек и обеспечивает быстрый доступ к

более подробным настройкам в диалоговом окне Project Settings (рисунок 4.1).

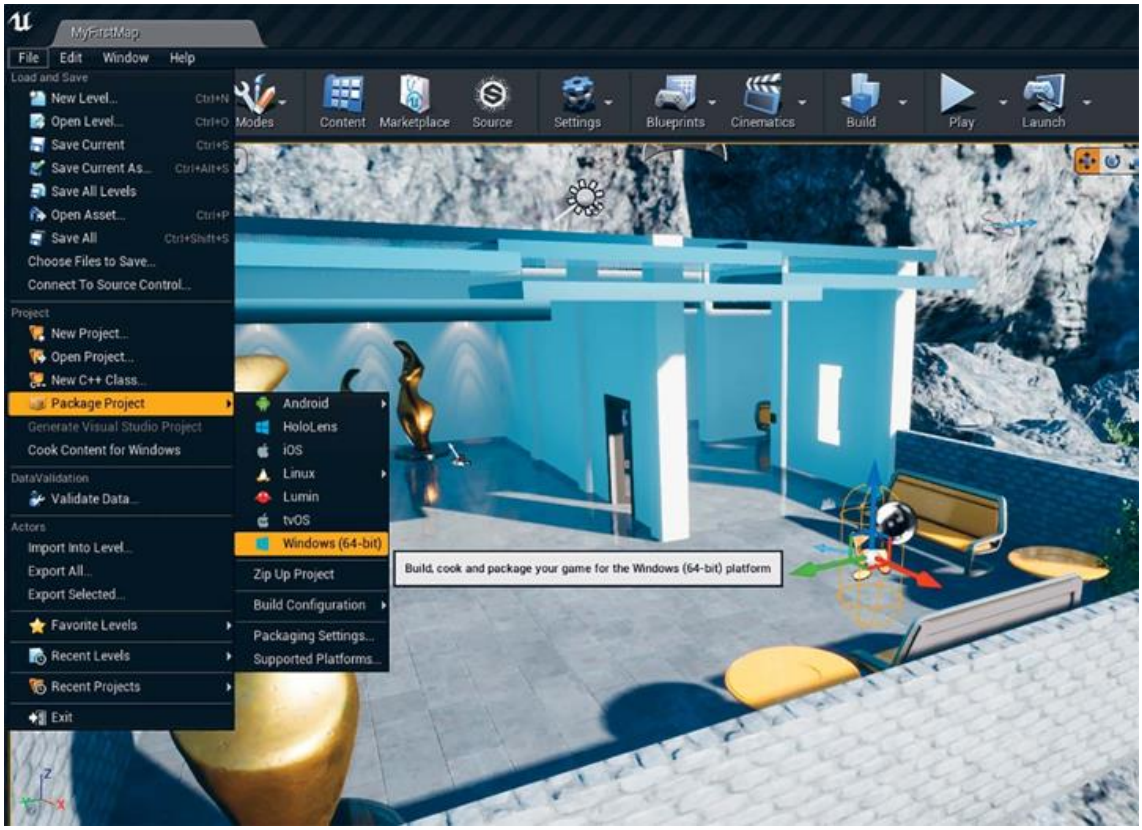

Рисунок 4.1 Меню упаковки в UE4 Editor

После выбора нужной платформы вам предложат указать место для сохранения вашего упакованного проекта. Не помещайте упакованный проект в папку вашего проекта — это может привести к путанице.

Упаковка занимает некоторое время. Появится уведомление, говорящее, что идет упаковка (рисунок 4.2). Этот процесс полностью фоновый, поэтому вы можете продолжить работу над вашим проектом, пока он обрабатывает контент в фоновом режиме.

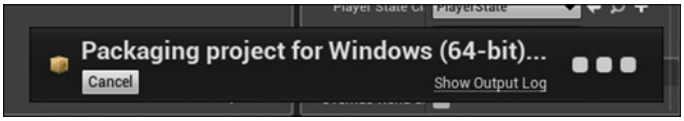

Рисунок 4.2 Уведомление об упаковке UE4 Editor

Упражение 7. Запуск вашего приложения.

После завершения упаковки Editor уведомит вас (рисунок 4.3).

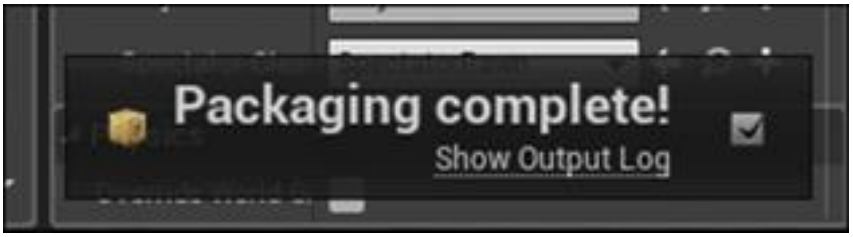

Рисунок 4.3 Уведомление о завершении упаковки в Editor

Пройдите к директории, которую вы определили ранее, и найдите папку с именем WindowsNoEditor (название может различаться на разных платформах). Вы можете переименовать эту папку.

Внутри папки вы найдете исполняемый файл и несколько папок, содержащих все данные и классы, а также облегченную бинарную версию движка (рисунок 4.4).

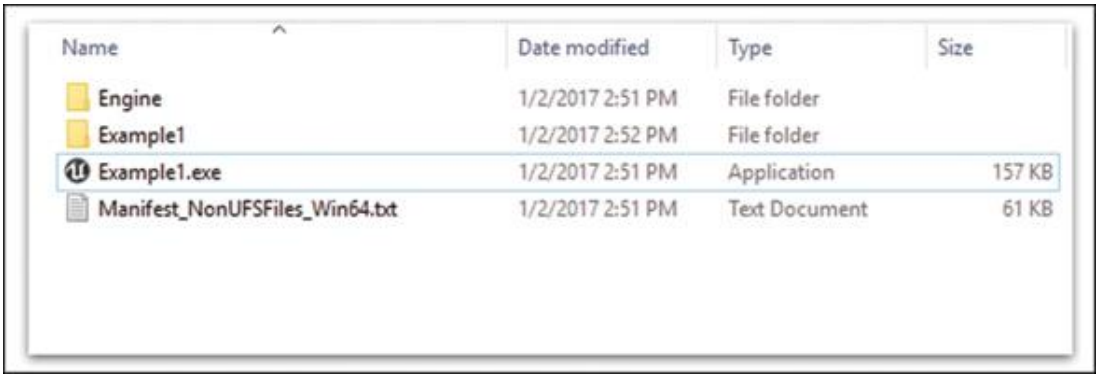

Рисунок 4.4 Упакованный проект в проводнике Windows

Можете запустить ваше приложение, просто дважды кликнув по исполняемому файлу. Приложение должно запуститься, загрузить Level, который вы создали, и вызвать соответствующий Player Controller и Pawn, предоставляя вам доступ к мыши.

4.5 Ошибки упаковки

Иногда упаковка сталкивается с ошибками. Простая ошибка компиляции Blueprint или нехватка места на диске могут остановить весь Packaging process.

Когда это происходит, вам следует обратиться к окну Output Log, которое в большинстве случаев может предоставить полезную информацию.

Для доступа к окну Output Log в Editor перейдите Window > Developer Tools > Output Log (рисунок 4.5). На практике прикреплют данное окно к Content Browser для быстрого доступа.

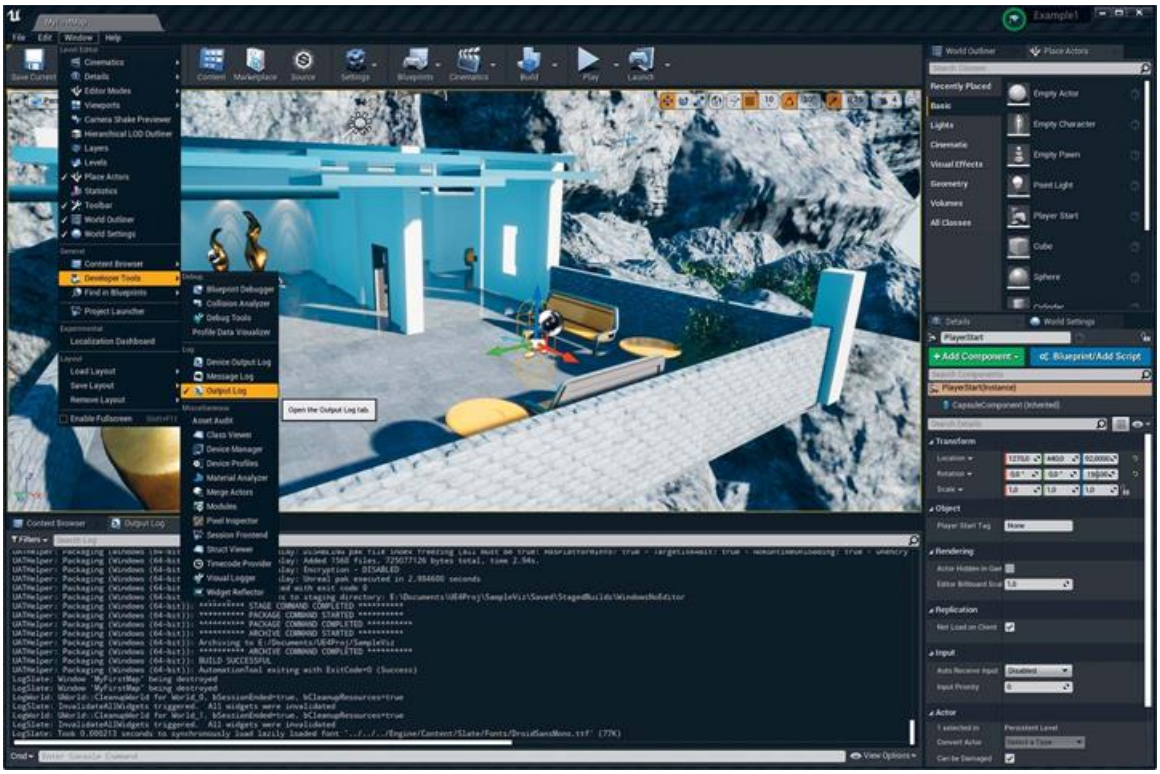

Рисунок 4.5 The Output log

4.6 Распространение проекта

Теперь вы можете копировать, архивировать и отправлять эту папку другим.

Некоторые компьютеры, на которых пытаются запустить ваше приложение, могут не иметь отдельных системных компонентов, которые требует UE4. Они называются Prerequisites. По умолчанию эти файлы включены в ваш упакованный проект. Если ваш упакованный UE4-проект обнаружит, что они не установлены, то предложит пользователю установить их, когда тот попытается запустить исполняемый файл.

Помимо пререквизитов, упакованное UE4 приложение не требует какого-либо формального процесса установки; оно может быть запущено из любого места.

Использование установщиков

UE4 не умеет встраивать установщик при упаковке. Разработка или использование установщика для приложения и платформы разработки зависит от вас. Однако, так как UE4 не требует формальной установки, предоставления файлов проекта в архиве с простыми инструкциями обычно достаточно.

Конечно, если вы хотите настроить ярлыки рабочего стола, возможности установки (install) и деустановки (uninstall), которые делает обычный установщик, то потребуется разработать его.

Хотя эти задачи выходят за рамки этого текста, их довольно просто выполнить, используя одно из множества бесплатных или коммерческих приложений для создания установщика.

5 АРХИТЕКТУРНАЯ ВИЗУАЛИЗАЦИЯ

Теперь, когда у вас есть четкое представление об использовании UE4 для создания простой сцены, вы можете погрузиться в пример из реальной жизни. Визуализация интерьера важна для архитекторов, маркетологов, дизайнеров и потенциальных покупателей, это один из наиболее распространенных типов визуализаций, выполняемых в UE4. В этой части вы исследуете использование Lightmass для создания высококачественного глобального освещения (global illumination (GI)), построения и рендеринга кинематографического пошагового руководства с использованием Sequencer. Затем вы погрузитесь в изучение того, как использовать Blueprints для создания динамических взаимодействий.

5.1 Объем проекта и его требования

Вам предоставили 3D-модель архитектурного интерьера (рисунок 5.1) и вы согласились создать интерактивную визуализацию, которая позволит пользователю свободно исследовать пространство от первого лица с помощью пешей прогулки. пользователь должен иметь возможность менять

определенные материалы с помощью простого пользовательского интерфейса, управляемого мышью. Клиент также запросил архитектурную анимацию, которая будет разработана с использованием

Sequencer и записана на видео для автономного воспроизведения и редактирования.

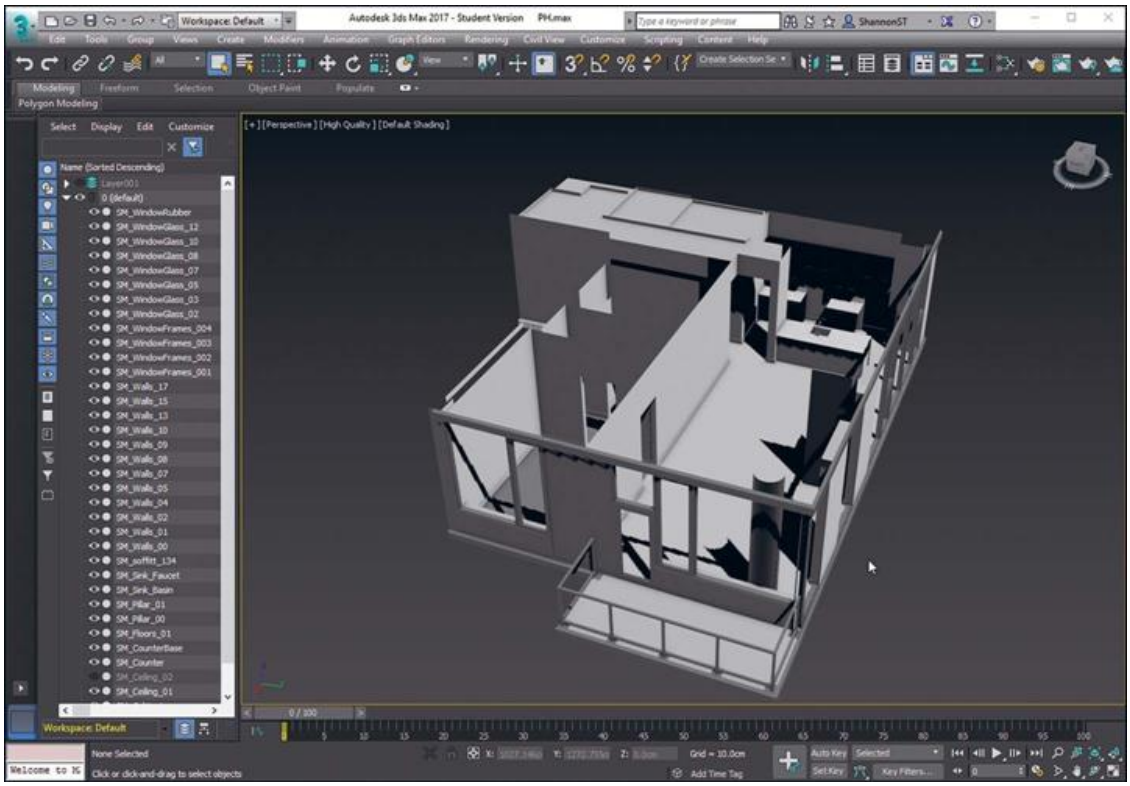

Рисунок 5.1 Предоставленная 3D-модель в 3D Studio Max

Первый шаг — подготовить 3D-модель для импорта в UE4, включая соответствующие материалы, UV-координаты и координаты Lightmap, а также сосредоточиться на создании чистой геометрии, которая импортируется на месте и легко обновляется и меняется. Далее описывается, как использовать автоматическую генерацию координат Lightmap UE4, чтобы сэкономить время и улучшить качество ваших сцен.

После импорта модели в UE4 пришло время начать итерационный процесс нанесения материалов и размещения света, отражающих точек, освещенных пространств и настройки постобработки, чтобы быстро получить отличные результаты.

После того как сцена готова, следующим шагом является создание анимированного пошагового руководства с помощью

Sequencer. Вы узнаете, как легко создавать привлекательные анимации и как заставить их проигрываться и прекратить использование входных данных, поступающих от пользователя.

Затем происходит перенос объекта Pawn, разработанного ранее, в новый проект и настройка сцены, чтобы позволить пользователю ходить по ней, включив столкновение и установив соответствующие акторы и параметры мира.

Завершающим этапом является создание систем переключения материалов и геометрии, которые используют Blueprints и UMG для добавления последнего слоя полировки и подготовки приложения к сборке и отправке заказчику.

Вот требования к этому проекту.

1. Разработка приложения архитектурной визуализации интерьера с использованием предоставленных клиентом 3D-моделей.
2. Использование Lightmass для создания высококачественного освещения GI и стремление к фотореализму.
3. Создание заранее определенной архитектурной анимации с помощью Sequencer и использованием нескольких ракурсов и переходов камеры.
4. Объединение павна и подготовки сцены, чтобы пользователь мог перемещаться по ней.
5. Разработка пользовательского интерфейса на основе UMG, который позволит пользователю переключаться на различные версии сцены.
6. Разработка системы переключения материалов с помощью мыши.
7. Сборка проекта для отправки.

5.2 Настройка проекта

В этом разделе рассматривается создание пустого проекта из Epic Launcher и перенос объектов Player Controller, Game Mode и Pawn в новый проект вместе с необходимыми конфигурационными файлами.

Эта миграция позволяет использовать системы и сосредоточиться на подготовке исходных 3D и 2D данных для UE4.

5.3 Создание проекта

Используя Epic Launcher, запустите версию движка, которую хотите применить, и выберите вкладку «новый проект», когда появится браузер проекта.

На этот раз вы не будете включать Starter Content, поэтому снимите флажок, но оставьте настройки Target Hardware на Maximum Quality.

Проект называется Example2. Сохраните его рядом с предыдущим проектом, Example1.

5.4 Миграция ассетов

Вам наверняка понадобится взять ассеты, Blueprints и другую работу, выполненную в предыдущих проектах, и скопировать их в новые проекты.

UE4 опирается на концепцию ассетов, ссылающихся на другие ассеты. Например, материалы могут ссылаться на несколько текстур. На каждый материал, в свою очередь, будут ссылаться меши, к которым он применяется, и даже Material Instances, производные от него.

Из-за этого следует быть осторожным при ручном копировании содержимого между проектами, чтобы они сохраняли эти ссылки и правильную структуру папок; в противном случае ссылки перестанут работать, что может привести к ошибкам загрузки, багам или даже сбоям в вашем проекте.

Самый надежный способ копирования контента из одного проекта в другой — это использование функции Content Migration. Миграция гарантирует, что все ассеты, на которые ссылаются другие ассеты, которые вы хотите скопировать, будут скопированы в целевой проект.

Чтобы перенести содержимое из одного проекта в другой, откройте исходный проект (проект с файлами, которые вы

хотите скопировать) в Editor. В данном случае это будет проект Example1.

После его открытия выберите ассеты, которые хотите скопировать в Content Browser. Вы можете выбрать отдельные ассеты, уровни или целые папки. В нашем случае выберите павна, PC и GameMode (рисунок 5.2). Щелкните правой кнопкой мыши и выберите Asset Actions > Migrate.

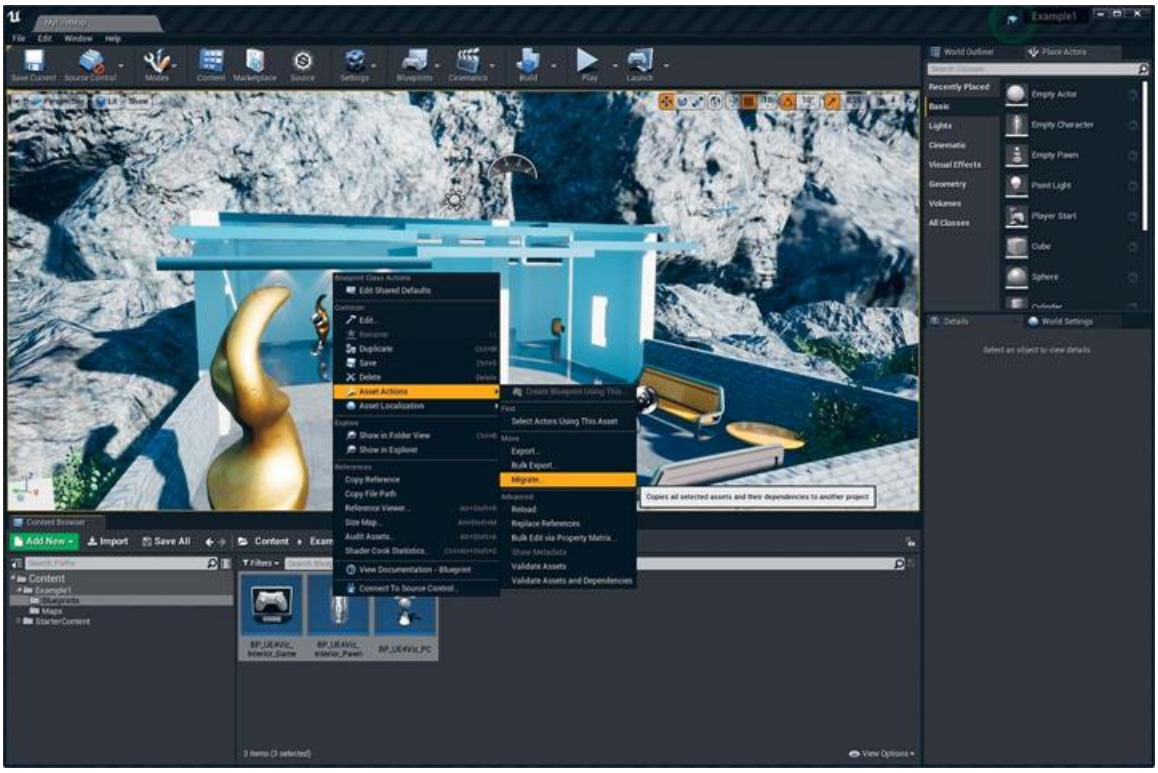

Рисунок 5.2 Миграция ассетов из проекта Example1

Откроется сводка файлов, которые будут скопированы в новый проект. Просмотрите ее и убедитесь, что у вас есть все, что вам нужно, и ничего такого, чего вы не хотите переносить (рисунок 5.3).

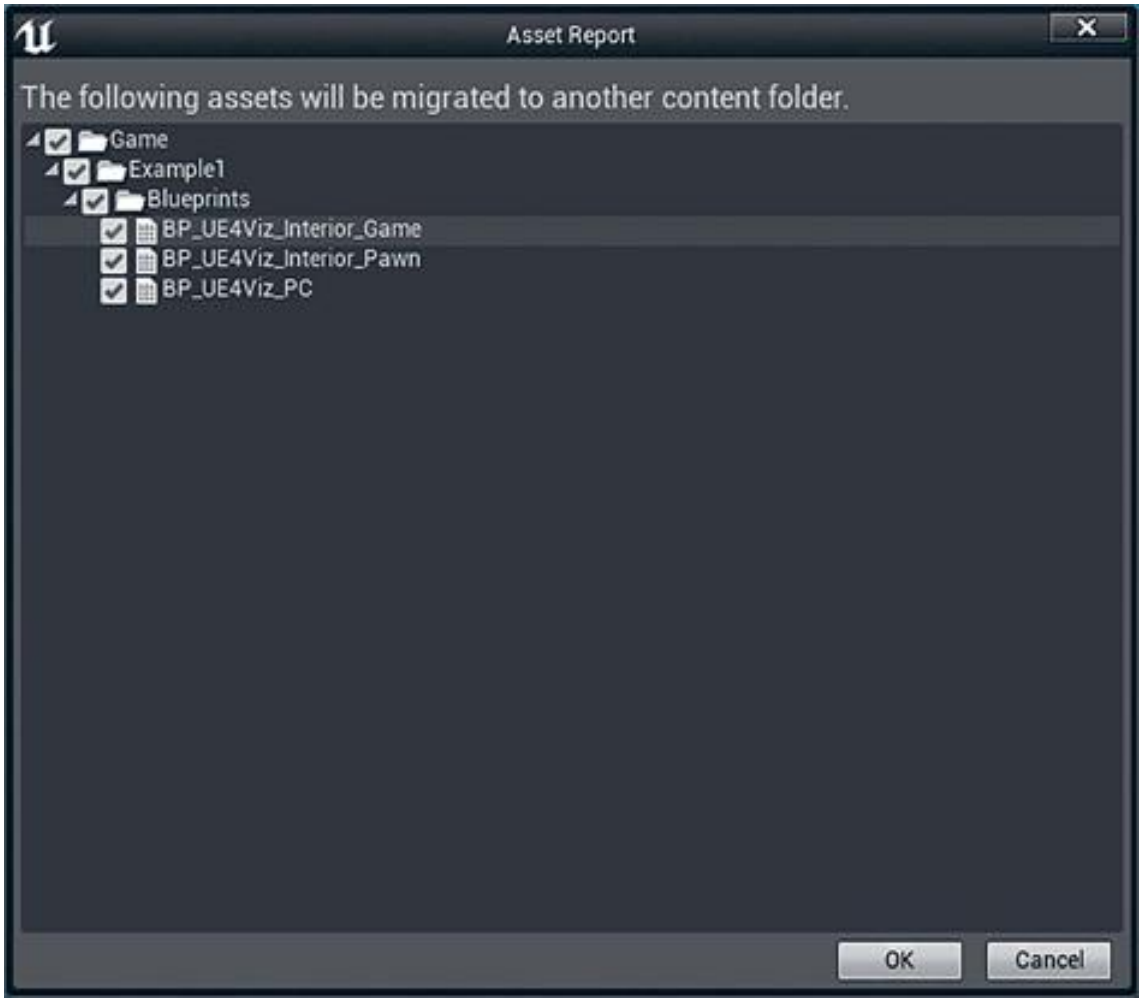

Рисунок 5.3 Список файлов, которые будут мигрированы

После подтверждения сводки появится экран с запросом выбрать каталог Content в рамках нового проекта. Перейдите в папку Content нового проекта и нажмите кнопку ОК (рисунок 5.4).

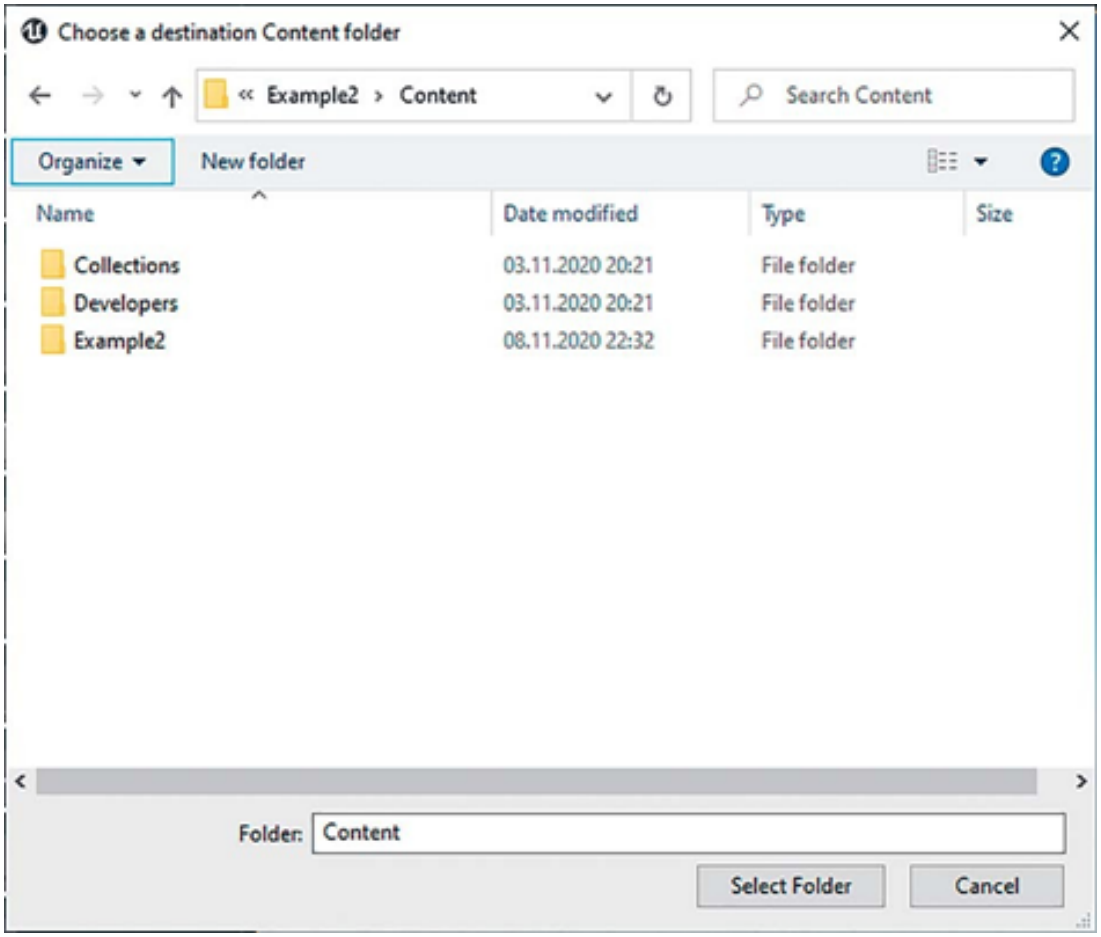

Рисунок 5.4 Указание Content каталога в новом проекте Example2

UE4 скопирует файлы и сообщит об успешном выполнении. Вы должны немедленно увидеть скопированный контент в каталоге Content нового проекта.

Настройку Input

Большая часть работы была сосредоточена на входных данных первого проекта, и получить их в новом проекте легко. На самом деле PC и объект Pawn не смогли бы скомпилироваться, потому что они ссылаются на input bindings, созданные в Example1.

Эти параметры хранятся в папке Config. Эти файлы не управляются из Content Browser и не копируются при использовании инструмента Content Migration, поэтому их необходимо переместить вручную.

Для этого найдите файл DefaultInput.ini в каталоге Config исходного проекта (Example1\ Config\) и скопируйте его в каталог Config целевого проекта (Example2\Config\).

5.5 Копирование, переименование и перемещение ассетов

При открытии проекта Example2 обратите внимание на папку Example1, содержащую павн, PC и GameMode.

Вы хотите переместить их в новую папку, которая не зависит от проекта и ее легко найти, и сейчас самое подходящее время избавиться от папки Example1 в Content Browser. Теперь вы находитесь в Example2!

Всегда выполняйте перемещение, копирование, переименование или любые другие операции с файлами в папке Content с помощью Content Browser. Это гарантирует, что ссылки на перемещенные или переименованные ассеты сохраняются. При необходимости создается redirector. Это небольшие файлы размером 1–2 КБ, которые служат указателем на правильное расположение ассета.

Упражнение 8. перемещение ассетов

Выполните следующие действия.

1. Создайте новую папку в Content directory под названием UE4Viz, а в этой папке — другую под названием Blueprints. Выберите эту папку, чтобы содержать повторно используемый код, который вы могли бы переносить из проекта в проект без необходимости переименовывать или копировать что-либо.
2. Перетащите ассеты Pawn, PC и Game Mode в папку Blueprints, которую вы только что создали. UE4 спросит, хотите ли вы переместить или скопировать ассеты. Выберите Move.
3. Щелкните правой кнопкой мыши на папке Example1 и выберите Fix Up Redirectors in Folder.
4. Удалите старую папку Content1, щелкнув по ней правой кнопкой мыши в Content Browser и выбрав пункт Delete.
5. Выберите Save All на панели инструментов обозревателя содержимого, чтобы убедиться, что все ваши изменения сохранены на диске.

5.6 Работа с Project Settings

Несколько вариантов рендеринга помогут оптимизировать Engine для использования в визуализациях архитектуры. Эти сцены, как правило, гораздо менее интенсивны, чем сцены видеоигр, что позволяет вам увеличить некоторые настройки и все еще поддерживать стабильную частоту кадров.

Упражнение 9. Project Settings

Откройте диалоговое окно Project Settings из меню Edit и перейдите к кнопке Rendering в левом столбце (рисунок 4.5). Измените настройки, как показано на рисунке и описано в следующем списке.

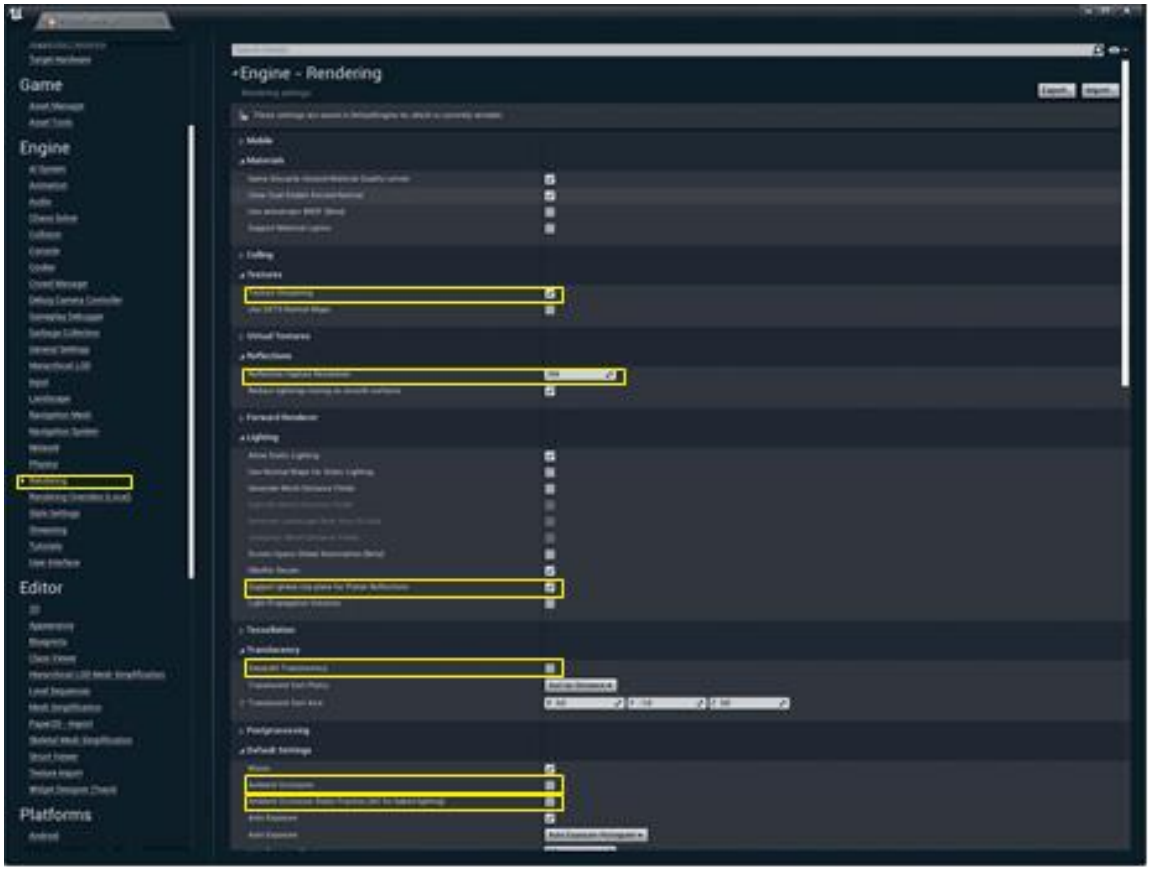

Рисунок 4.5 Настройки рендеринга проекта

1. Disable Texture Streaming: этот параметр предназначен для включения больших игровых миров в ограниченной памяти на консолях и более старом оборудовании ПК. Это может привести к ошибкам отображения света при использовании архитектурных мешей и Lightmaps высокого разрешения Disabling Texture Streaming заставляет все текстуры,

используемые на уровне, загружаться в оперативную память сразу. Используйте эту настройку с некоторой осторожностью. Потоковая передача текстур является важной оптимизацией для сложных сцен, и ее отключение может привести к проблемам с производительностью.

2. Set Reflection Capture Resolution: чтобы получить самые качественные отражения, вы можете увеличить разрешение Reflection Capture Actors, которые размещаете на своих сценах. Будьте осторожны, устанавливая его слишком большим: это может съесть много памяти, если у вас много отражений с высоким разрешением. Не желательно делать значение более 512 для этого параметра, если у вас только не одно или два отражения на каждом из уровней.

3. Turn on Support Global Clip Plane for Planar Reflections: зеркала и отражения являются основным элементом визуализации, и UE4 поддерживает зеркальные Planar Reflections. Это очень дорогая настройка, и вы должны включить ее только в том случае, если она вам абсолютно необходима. Даже если отражения не отображаются на экране, они оказывают значительное влияние на производительность рендеринга в масштабах всего проекта.

Вы можете смягчить это воздействие с помощью панели Details, чтобы отключить множество более тяжелых функций, таких как некоторые настройки постобработки, например, Screen-space Ambient Occlusion и Reflections. Вы даже можете ограничить видимых акторов или расстояние, на котором будет отображаться отраженная сцена.

4. Turn off Separate Translucency: Separate Translucency — это повышение производительности для игр с большим количеством эффектов альфа-прозрачности, таких как искры и взрывы. Эти эффекты могут быть очень дорогими для применения таких эффектов, как Depth of Field, поэтому эти эффекты визуализируются отдельно и игнорируются постобработкой. Вам не нужно, потому что вы наверняка хотите,

чтобы полупрозрачные поверхности вели себя как можно реалистичнее.

Эта настройка не влияет на материалы Masked, поэтому такие вещи, как растительность, обычно не затрагиваются.

5. Turn off Ambient Occlusion and Ambient Occlusion Static Fraction: эти параметры относятся к Screen-space Ambient Occlusion (SSAO) и эффектам постобработки. SSAO — это отличный способ добавить детали к динамически освещенным сценам и объектам. Поскольку ваша сцена почти полностью статична, и вы используете высококачественные настройки Lightmass, вы можете отключить эту настройку для значительного повышения производительности, особенно при более высоких разрешениях.

Эти настройки влияют только на эффект SSAO и не влияют на AO, рассчитанный по Lightmass. Static Fraction контролирует, насколько эффект SSAO сочетается с запеченными световыми картами, а также имеет накладные расходы на производительность, которые можно уменьшить, отключив эффект.

Настройки проекта автоматически записываются в файл проекта DefaultEngine.ini, и нет необходимости сохранять его перед закрытием окна настроек. Однако, поскольку мы изменили такие настройки, как Planar Reflections, вам нужно будет перезапустить Editor, чтобы эти настройки можно было инициализировать. Есть вероятность, что при повторном открытии Editor вы столкнетесь с перекомпиляцией шейдеров, которая может занять некоторое время, но должна произойти только один раз.

Заключение

Теперь ваш проект готов к запуску. Включение некоторых расширенных функций рендеринга UE4 и отключение отдельных ненужных функций улучшает качество изображения и позволяет избежать проблем с потоковой передачей текстур, SSAO и полупрозрачными материалами.

Вы также узнали, как использовать функцию Content Migration UE4 для перемещения готовой работы из одного проекта в другой, гарантируя, что вам не придется заново изобретать колесо для каждого проекта. Входные настройки также были скопированы через .ini-файл.

Теперь вы готовы начать учиться освещать свой мир, применять материалы и использовать Blueprints и Unreal Motion Graphics (UMG), чтобы позволить пользователю вносить интерактивные изменения.

6 ПРОЦЕСС РАБОТЫ С ДАННЫМИ

Получение точных архитектурных данных в UE4 на первый взгляд может показаться сложной задачей и противоречить вашим рабочим процессам. Но с помощью нескольких стратегий и понимания всего процесса работы вы сможете легко получить любой тип данных в UE4. Вы будете импортировать массивные наборы данных в UE4 без труда и в кратчайшие сроки.

6.1 Организация сцены

Для этого проекта вы потратите много времени вне UE4, внутри ваших приложений DCC, создавая 3D-модели и текстуры, которые будут импортированы в UE4.

Сценарий: данные, предоставленные вашим клиентом, оказались хорошего качества. Их можно использовать почти «из коробки», но вы все равно хотите потратить время на подготовку контента, чтобы он импортировался в UE4 как можно проще и надежнее.

Начните с организации сцены в вашем 3D-приложении. САПР и инженерные приложения часто не генерируют наиболее элегантно названные или организованные сцены. Часто специфическая логика используется для именования объектов, которые трудно читать, или они несовместимы с UE4 или конвейером FBX. Длинные или содержащие сложные символы имена могут вызвать проблемы с экспортером FBX и UE4.

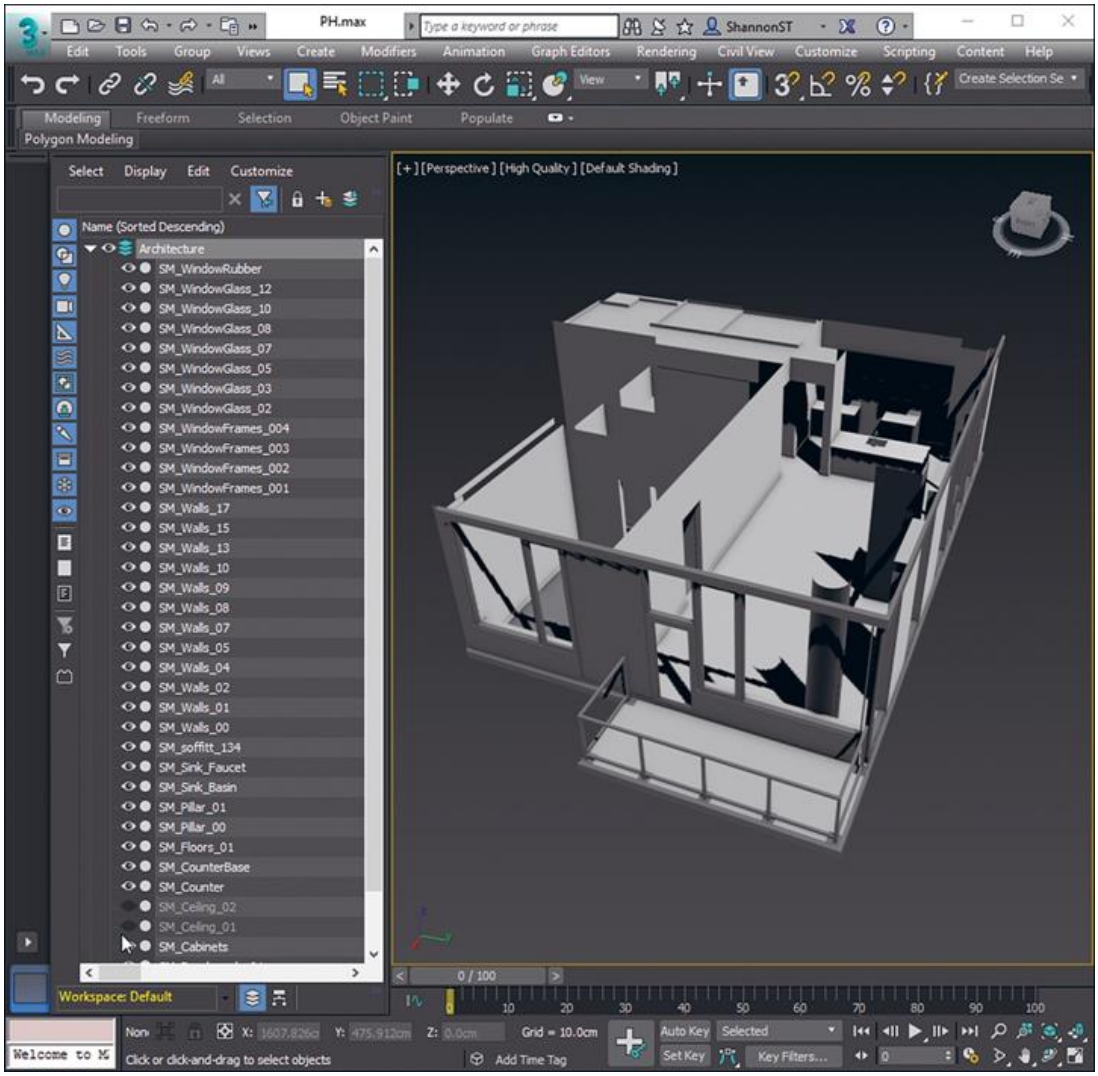

Рисунок 6.1 Организация 3D-сцены в 3D Studio Max

Определение наилучшей организации для вашего конкретного контента и адаптация ваших инструментов и процессов к потребностям конвейера FBX зависят только от вас.

В этом случае вы можете использовать простой сценарий переименования, чтобы упростить имена объектов на сцене и привести их в соответствие с принятой схемой именования UE4, в которой префикс SM_ ставится перед статическими мешами (рисунок 6.1).

Теперь это делается быстро и легко по сравнению с попыткой переименовать все, что было экспортировано, или в UE4, где переименование гораздо более важно.

Вы также можете отсортировать свою сцену по слоям в зависимости от типа модели, наиболее распространенными

типами которой являются реквизит (Props) и архитектура (Architecture). Это позволяет легко выбирать слои для экспорта или управлять видимостью объекта на сцене.

6.2 Материалы

Хотя UE4 не поддерживает полный импорт/экспорт материалов из 3D-приложений, он поддерживает импорт основных функций материала, таких как цветные и Texture Maps. Импорт даже основных материалов экономит много времени на первоначальной настройке сцены.

UE4 также поддерживает назначение атрибутов вашего материала определенным параметрам в указанном родительском материале. Вместо создания материала для каждого импортированного материала делается Material Instance Asset и настраиваются его параметры. Это отличный рабочий процесс для продвинутых пользователей, и необходимо вернуться к нему, когда вы захотите повысить эффективность своих рабочих процессов.

Обратите внимание, что UE4 поддерживает только стандартные материалы для приложений. Материалы из пользовательских рендеров, таких как V-Ray, могут вообще не импортироваться или импортироваться плохо.

В этом случае в исходной модели используются Autodesk Architectural Materials, которые не придерживаются текстурных форматов и входных данных, поддерживаемых UE4. Использовался простой скрипт, который заменял каждый из них стандартным материалом 3D Studio Max с тем же именем, отбрасывая все остальные параметры и назначая простые растровые изображения диффузным контактам входа, когда это было возможно.

Multi-subobject (MSO) Materials также поддерживаются UE4, но мешу нужен как Material ID, назначенный граням, так и материал MSO с уникальным материалом, назначенным каждому ID. Например, меш с четырьмя Material ID и только одним материалом, назначенным в Max, будет импортироваться как меш только с одним Material ID.

Вы также должны найти время, чтобы переименовать свои материалы и текстуры в соответствии с вашим соглашением об именовании. Опять же, это гораздо проще сделать в вашем 3D-приложении, чем в UE4.

6.3 Архитектура и арматура

Сначала вам стоит получить основные Architectural Meshes — стены, полы, потолки, светильники и технику — импортированные в UE4 и помещенные на нашу сцену. Эти меши уникальны и не нуждаются в перемещении. Также важно, чтобы они располагались точно в том же месте, что и исходные данные.

6.4 Обеспечение чистой геометрии

Чистое освещение требует чистой, хорошо подогнанной геометрии. Мерцания, пятна и ошибки освещения деморализуют, поскольку требуют повторной сборки после импорта геометрии и уже проведенной длительной сборки света.

Два самых больших виновника — плохие нормали у граней (перевернутые грани) и совпадающие или сложенные грани. И то, и другое вызывает мерцание и плохое освещение. Lightmass не может правильно рассчитать отраженный свет от этих граней и часто возвращает черный цвет.

Чтобы найти перевернутые нормали, отключите backface culling или double-sided Materials в вашем 3D-приложении. Некоторые приложения также предлагают инструменты визуализации, которые помогут вам искать перевернутые грани.

Найти совпадающие грани может быть сложнее. Ради производительности UE4 использует 16-битный буфер глубины, который более склонен к мерцанию, когда грани расположены более близко, чем вы могли бы использовать с рендерингом трассировки лучей. Эта модель имела много перекрывающихся граней, которые требовали значительной очистки. Многие ошибки не были очевидны до тех пор, пока их не импортировали в UE4 и не увидели там мерцание. Обеспечение хорошей

итеративной настройки рабочего процесса делает устранение подобных проблем гораздо менее болезненным.

Вы также наверняка захотите избежать односторонних мешей. Свет будет проходить через заднюю сторону и вызывать утечки света и другие ошибки рендеринга. В этой сцене вы должены были создать внешние стены, а также потолочные и напольные сооружения, чтобы обеспечить хорошее освещение.

Использование хороших UV-координат

Наличие хороших UV-координат очень важно в UE4. Плохие UV-координаты не только затрудняют нанесение материалов и приводят к плохому внешнему виду, но и могут негативно повлиять на качество рендеринга объектов.

Базовый канал (Channel 0)

Канал 0, базовый UV-канал, является каналом по умолчанию для применения текстур к мешам.

Предоставленная модель содержит меши с хорошими UV-координатами, но выглядит как плитка, повторяясь каждый 1 см (данные CAD часто будут сопоставляться с реальными единицами измерения). Хотя вы можете масштабировать свои текстуры в материалах UE4 для компенсации, потому что это затрудняет чтение меша для предварительного просмотра — текстура растягивается в большом масштабе, необходимом для компенсации.

Стандартный процесс масштабирования UV-координаты примерно до 1 м. Делается это либо с помощью box UV modifier (рисунок 6.2), либо, если уже есть реальные масштабируемые координаты, подобные тем, что в этой модели, использовался модификатор UV map scaler для масштабирования координат до более удобного масштаба UE4.

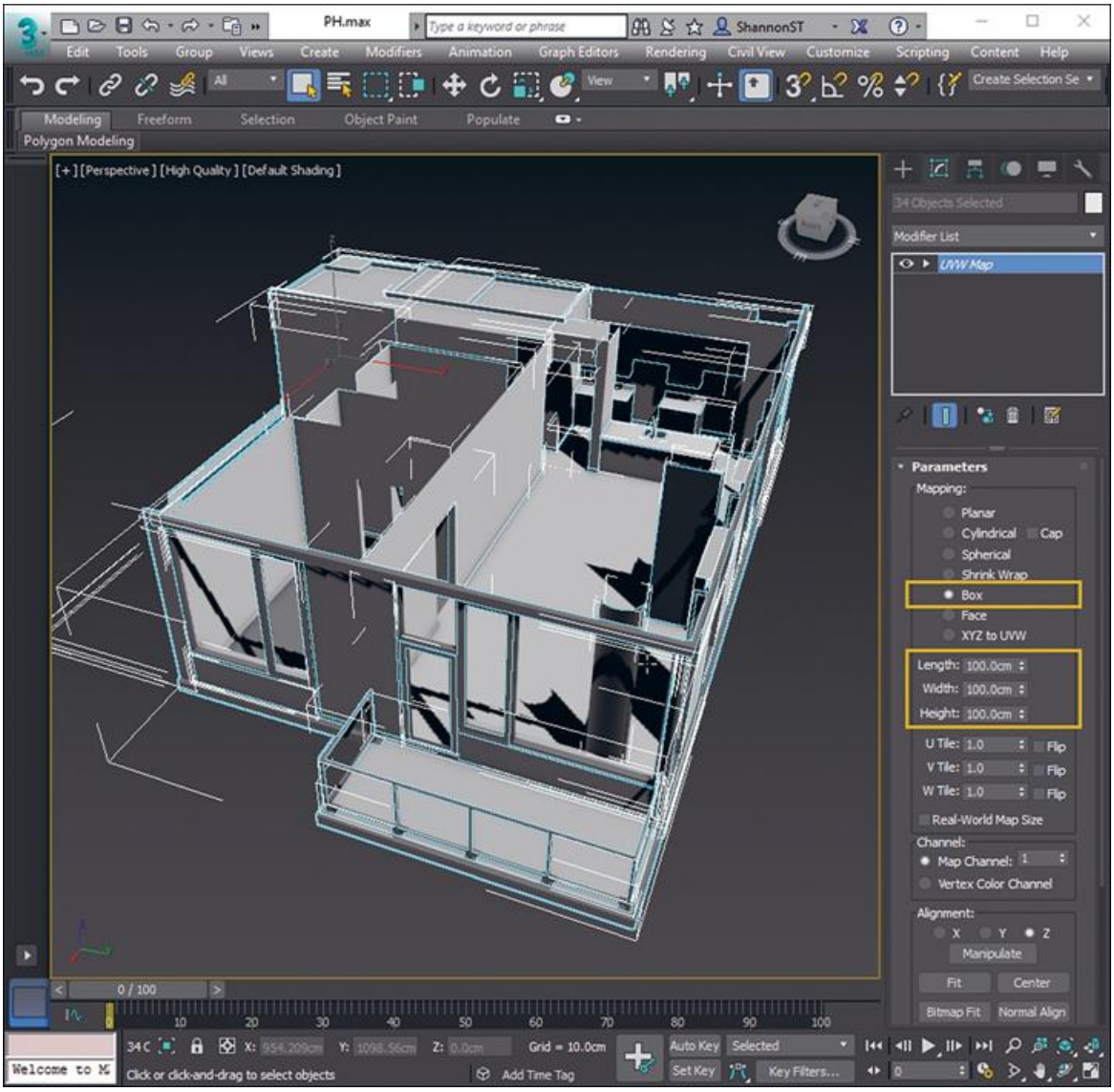

Рисунок 6.2 Применение одного экземпляра Box UVW map modifier ко всем стенам, полам и потолкам

Lightmass (Channel 1)

Lightmass требует чистых, неискаженных UV-координат, чтобы быть эффективным. Вы можете либо повторно упаковать свои координаты в 3D-приложении, либо положиться на систему переупаковки UE4.

Обычно опираются на Automatic Lightmap Coordinate Packing UE4, гарантируя, что модели имеют хорошие базовые координаты канала UV перед экспортом. Более сложные меши могут потребовать ручного редактирования этих координат для достижения наилучших результатов, но большинство мешей работают очень хорошо.

Эта система не разделяет края, не применяет и не меняет ваши UV-диаграммы каким- либо образом. Она просто берет UV-данные канала 0 и переупаковывает их.

Для сцен, которые не имеют UV-координат, или, когда вы моделируете собственные сцены, вы все еще можете использовать автоматическую упаковку в UE4, но вам нужно убедиться, что ваши базовые UV-координаты чисты, без UV-искажений и имеют хорошо расположенные границы.

Большинство стен, полов и потолков можно отобразить с помощью a box-style mapping. Конечно, неквадратные грани могут потребовать дополнительного внимания, но если ваши базовые Texture Maps выглядят хорошо, UE4 должен быть в состоянии сделать хороший набор координат Lightmap для вас.

6.5 Экспортирование сцены

Теперь, когда ваши меши подготовлены, пришло время импортировать их в UE4. Для этого сначала нужно сохранить их в файл формата UE4, который можно импортировать. Для статических мешей предпочтительным форматом файла является FBX.

Для экспорта вашей модели в FBX доступно несколько методов. У каждого из них свои преимущества и недостатки, и каждый может работать лучше для вашей ситуации или сцены.

Использование нескольких FBX-файлов

Стандартно экспортирование объектов на сцены в отдельные файлы FBX, позволяет максимально контролировать данные, и можно легко и надежно обновлять, экспортировать и повторно импортировать свои данные.

Большинство 3D-приложений не поддерживают пакетный экспорт файлов FBX, что вынуждает вас утомительно экспортировать их один за другим. Из-за этого разработался сценарий Max, который позволяет мне быстро сделать это (рисунок 6.3).

Важно отметить, что UE4 использует имя файла FBX в качестве имени импортированного меша и игнорирует имя, назначенное в вашем 3D-приложении.

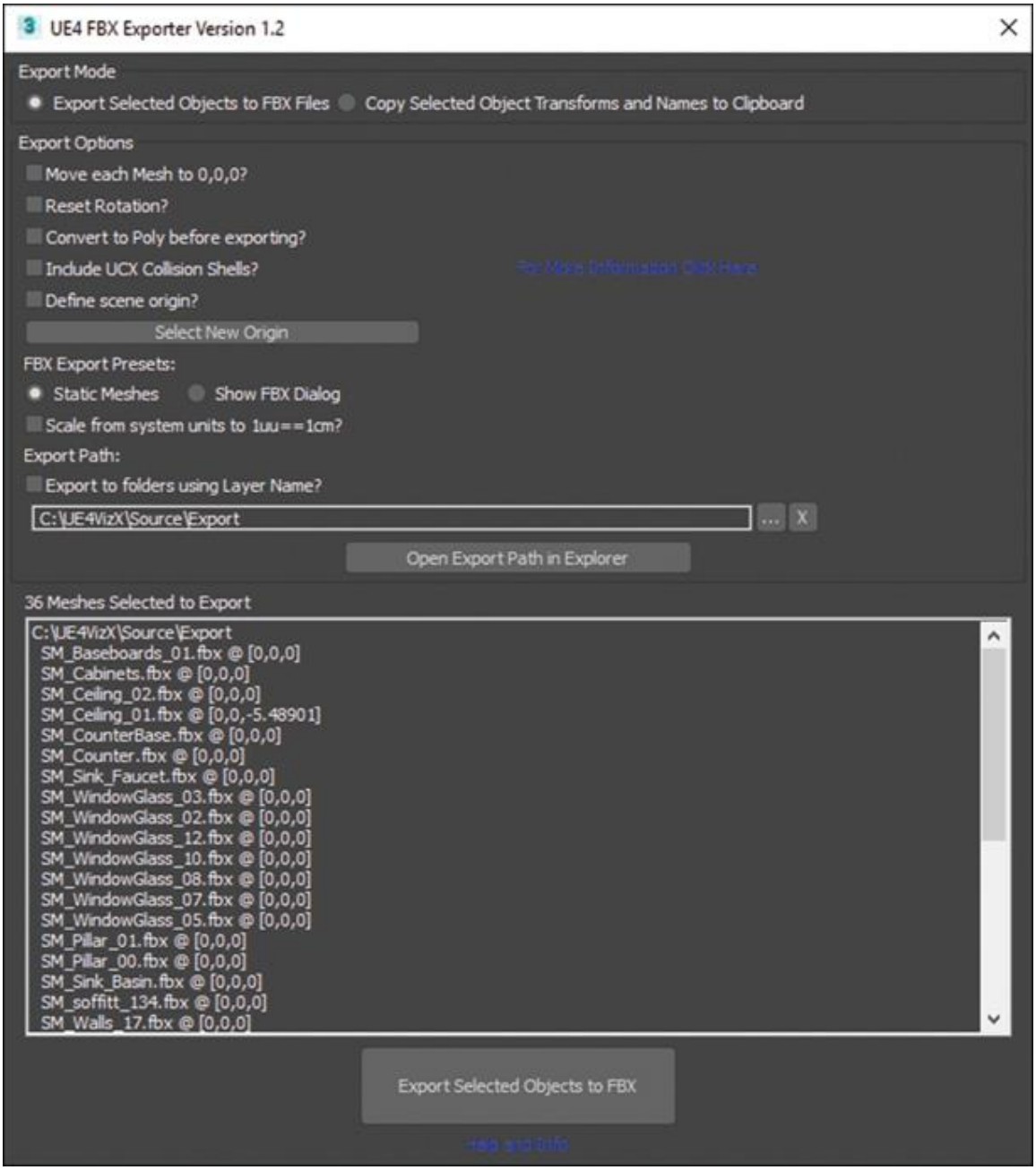

Рисунок 6.3 Скрипт для Max используется для пакетного экспорта файлов FBX для UE4;

Использование одного FBX-файла

Вы также можете экспортировать всю свою геометрию в один файл FBX. UE4 может импортировать файл FBX с несколькими объектами в нем разными способами. Параметры варьируются от создания единого монолитного статического

меша до импорта каждого объекта в виде статического меша в файл по отдельности с помощью Content Browser. UE4 также предлагает функцию Import to Level в меню File, которая позволяет автоматизировать им- порт сложных иерархических моделей в комплекте с камерами, подсветкой и анимацией.

Эти опции очень просты в использовании, и это отличный способ быстро импортировать большие наборы данных в UE4.

Самый большой недостаток, с которым можно столкнуться этими методами, — это трудности с обновлением конкретных мешей или наборов мешей из файла FBX. UE4 ожидает, что структура файла FBX останется неизменной между реимпортами, поэтому реимпорт статических мешей легко вызовет ошибки, если вы не будете очень осторожны с экспортом и не станете поддерживать строгий набор экспортируемых объектов для каждого файла FBX.

6.6 Импорт сцены

В зависимости от того, как вы экспортировали свои данные и как хотите управлять ими на собственной сцене, у вас есть несколько вариантов, когда вы импортируете свой контент в UE4.

Импортирование в Content Browser

Если вы экспортировали содержимое в виде отдельных файлов FBX, вы можете импортировать их по одному или в пакетном режиме, перетаскивая файлы FBX из системных файлов (рисунок 6.4) или используя кнопку Import в Content Browser.

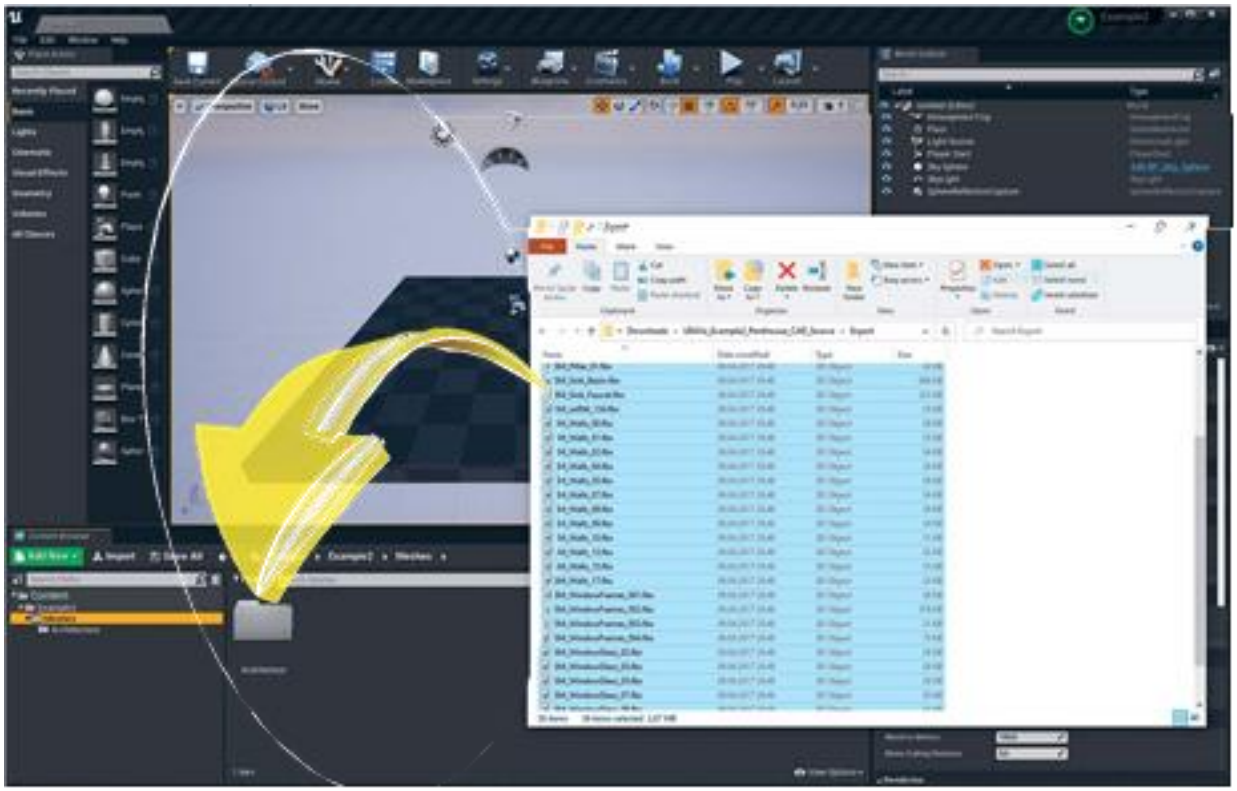

Рисунок 6.4 Импорт нескольких файлов FBX путем перетаскивания из Windows Explorer в Content Browser

Если вы экспортировали ваш контент в один FBX-файл с большим количеством мешей, вы можете импортировать эти файлы в Content Browser как один совмещенный статический меш или в виде отдельных мешей, выбрав Combine Meshes в диалоге FBX Import Options, который появляется при попытке импортировать FBX ассет.

Как видно на рисунке 6.5, этот диалог предлагает множество опций, большинство из которых вы можете оставить по умолчанию, за несколькими заметными исключениями.

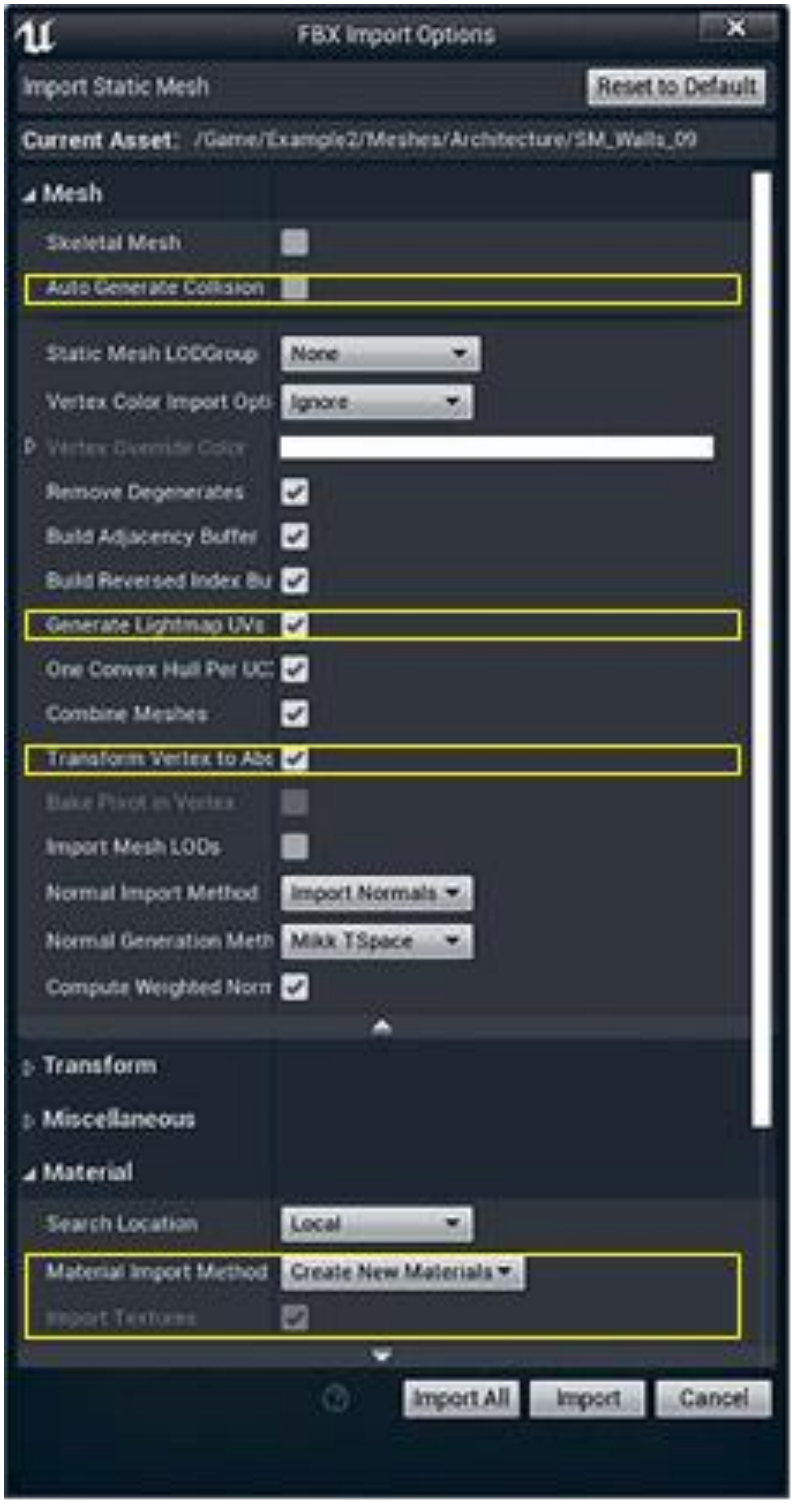

Рисунок 6.5 Диалоговое окно FBX Import Options. Auto Generate Collision отключена, в то время как Generate Lightmap UVs включена, как и параметры Material and Texture import

Auto Generate Collision

Опция Auto Generate Collision создает очень упрощенное поле столкновений вокруг вашего меша и обычно не подходит для визуализации данных. Поскольку эта сцена проста, вы можете положиться на столкновение по полигонам. Это дороже, чем использование примитива коллизии с низким количеством полигонов, и может вызвать проблемы с производительностью в более сложных сценах. Для этого проекта вы должны установить значение false.

Generate Lightmap UVs

Вы уверены, что ваши меши имеют хорошие, чистые UV-координаты, поэтому система Automatic Lightmap Generation должна давать хорошие результаты. Установите для параметра Generate Lightmap UVs значение true.

Transform Vertex to Absolute

Transform Vertex to Absolute позволяет вам решить, использует ли UE4 pivot point, созданную в вашем 3D-приложении, или она будет установлена в значение 0,0,0. Для ваших архитектурных мешей установите этот параметр в значение true.

6.7 Импорт материалов и текстур

Установите значение true, чтобы импортировать применяемые материалы и текстуры вместе с моделью.

Importing into the Level

Другой вариант — это новый Import into Level. Это позволяет импортировать содержимое файла FBX, включая связанные иерархии, анимацию, освещение и камеры.

В UE4 выберите File > Import into Level и выберите файл FBX (рисунок 6.6). Выберите папку для импорта (рисунок 6.7).

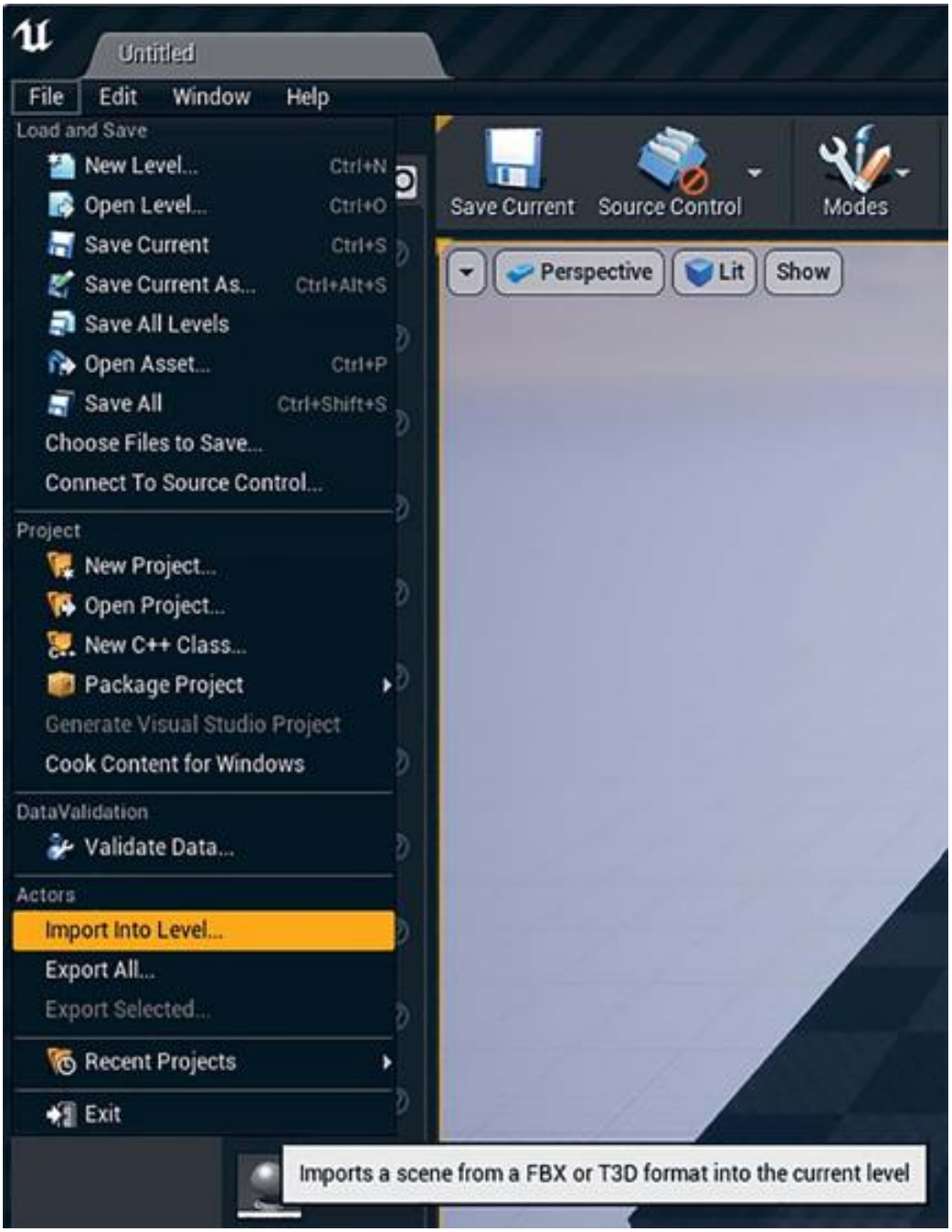

Рисунок 6.6 Импорт сцены FBX непосредственно на уровень

Появится диалоговое окно импорт / реимпорт (рисунок 6.8), позволяющее предварительно просмотреть меши и материалы, которые будут импортированы. Это хорошее время, чтобы быстро взглянуть и убедиться, что вы импортируете то, что хотите.

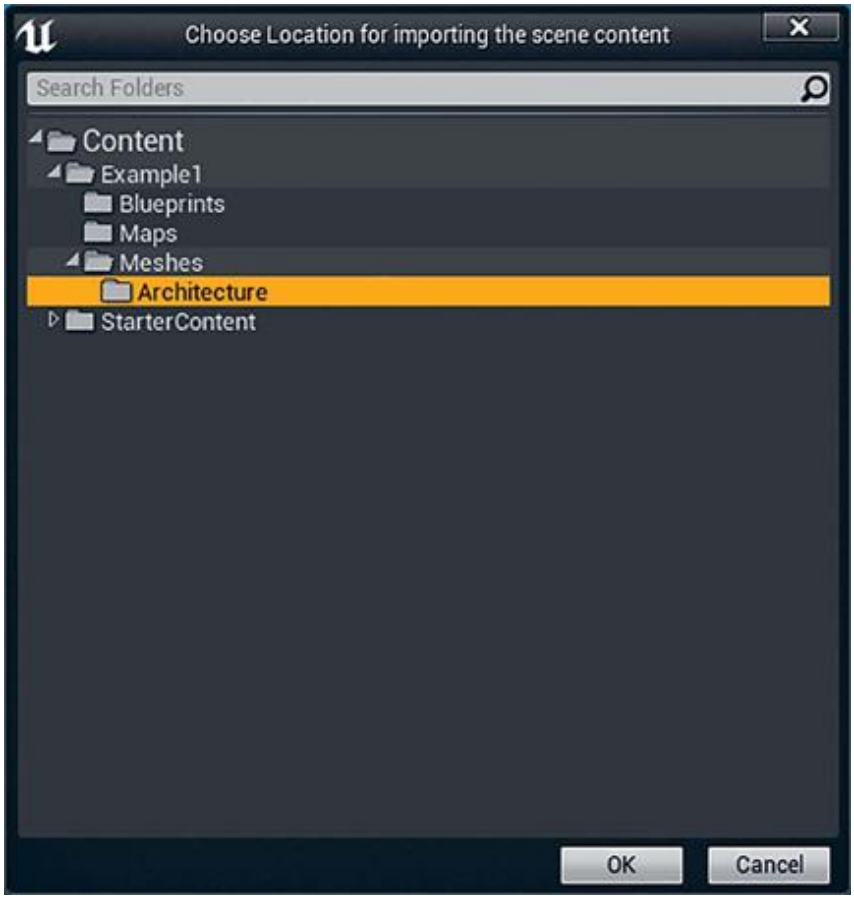

Рисунок 6.7 Выбор целевой папки

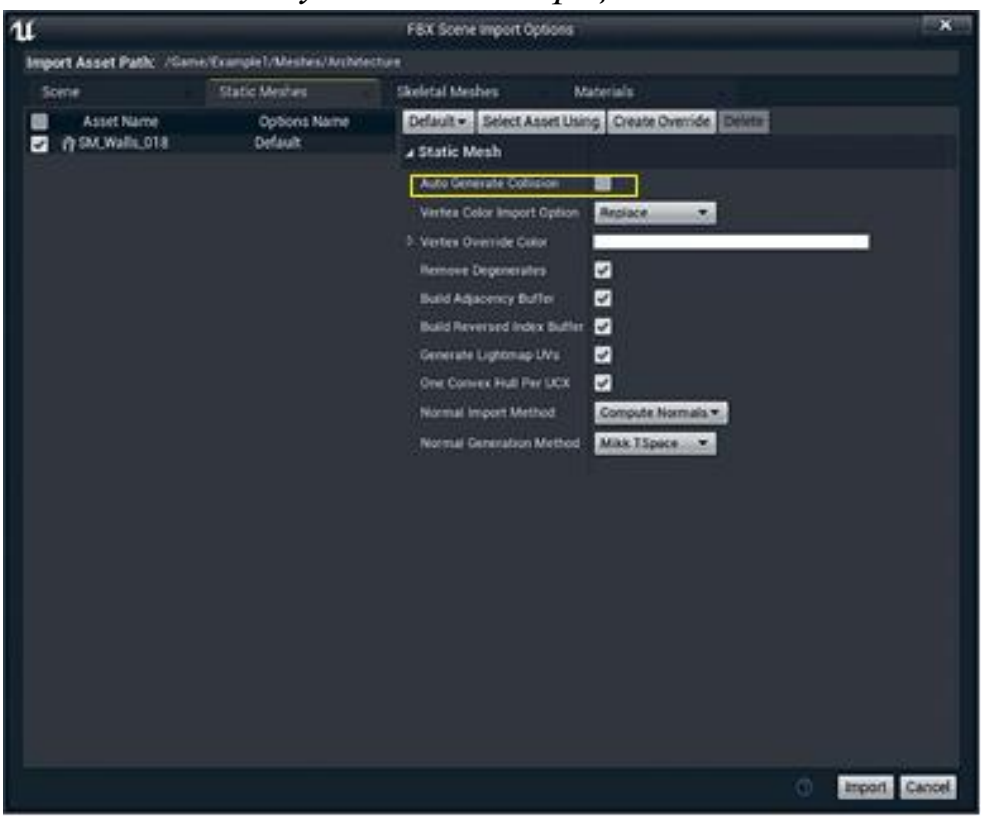

Рисунок 6.8 Диалоговое окно FBX Scene Import Options

Обязательно отключите функцию Auto Generate Collision, поскольку она несовместима с мешами в визуализации архитектуры. Вы будете использовать per-poly collision на своих стенах, а не полагаться на прокси-объекты коллизии.

Каждый меш импортируется как единое целое в файл.uasset вместе с любыми материалами и текстурами, примененными к этим материалам. Вы также можете определить другую папку для импорта ваших материалов.

Создается один Blueprint Actor и помещается в загруженную сцену со ссылками на каждый меш. Этот Blueprint работает в сочетании с FBX Import Data Asset, позволяющим повторно импортировать файл FBX, а также добавлять, удалять и изменять меши и материалы в соответствии с файлом FBX при повторном импорте.

После импорта не забудьте сохранить свою работу, потому что импортированные меши не сохраняются на диске до тех пор, пока вы не сохраните их сами.

Декоративные меши (реквизит)

Стулья, стаканы и тарелки — это все, что можно считать декоративными мешами, или реквизитом: меши, которые дублируются и/или должны быть перемещены или помещены в Editor.

В модели, представленной клиентом в примере этой главы, отсутствовала какая-либо мебель, но вам были даны рекомендации и справочные материалы. Оттуда вы строили 3D-модели специально для UE4 или находили меши в своей библиотеке контента, которые можно было импортировать.

Лучше всего разместить свой реквизит в UE4, а не в 3D-приложении. FBX не поддерживает инстанцирование и будет рассматривать каждую тарелку, стакан и стул как уникальный объект, импортируя каждый из них как уникальный .uasset, каждый из которых занимает память, дисковое пространство и вычислительную мощность. Это также означает, что вы должны обновить каждый экземпляр объекта, если есть что-то, что нужно изменить.

Когда у вас есть много повторяющихся объектов на сцене, имея лишь один.uasset можно использовать его с большим количеством ссылок на этот единственный файл ассета на уровнях.

Кроме того, как будет показано далее, создавая хорошие pivot points, вы можете сделать изменение, перемещение и даже анимацию ваших мешей намного проще, чем их явное размещение.

Установка точек вращения

Вы должны создавать реквизит иначе, чем архитектурные меши, с самой большой разницей в точках вращения (пивотах). Хотя архитектурные меши экспортируются и импортируются на месте, с их пивотами, установленными на 0,0,0 при импорте, опорные меши должны быть импортированы с хорошо авторизованными пивотами.

Пивоты не могут быть отредактированы непосредственно в UE4, поэтому настройка точек разворота в 3D-приложении очень важна.

Поместите свой пивот (точку вращения) туда, где вы хотите, чтобы ваш меш прикрепился к миру. Создание правильного pivot point может сэкономить огромное количество усилий, когда речь заходит о размещении, перемещении и даже анимации ваших мешей в UE4.

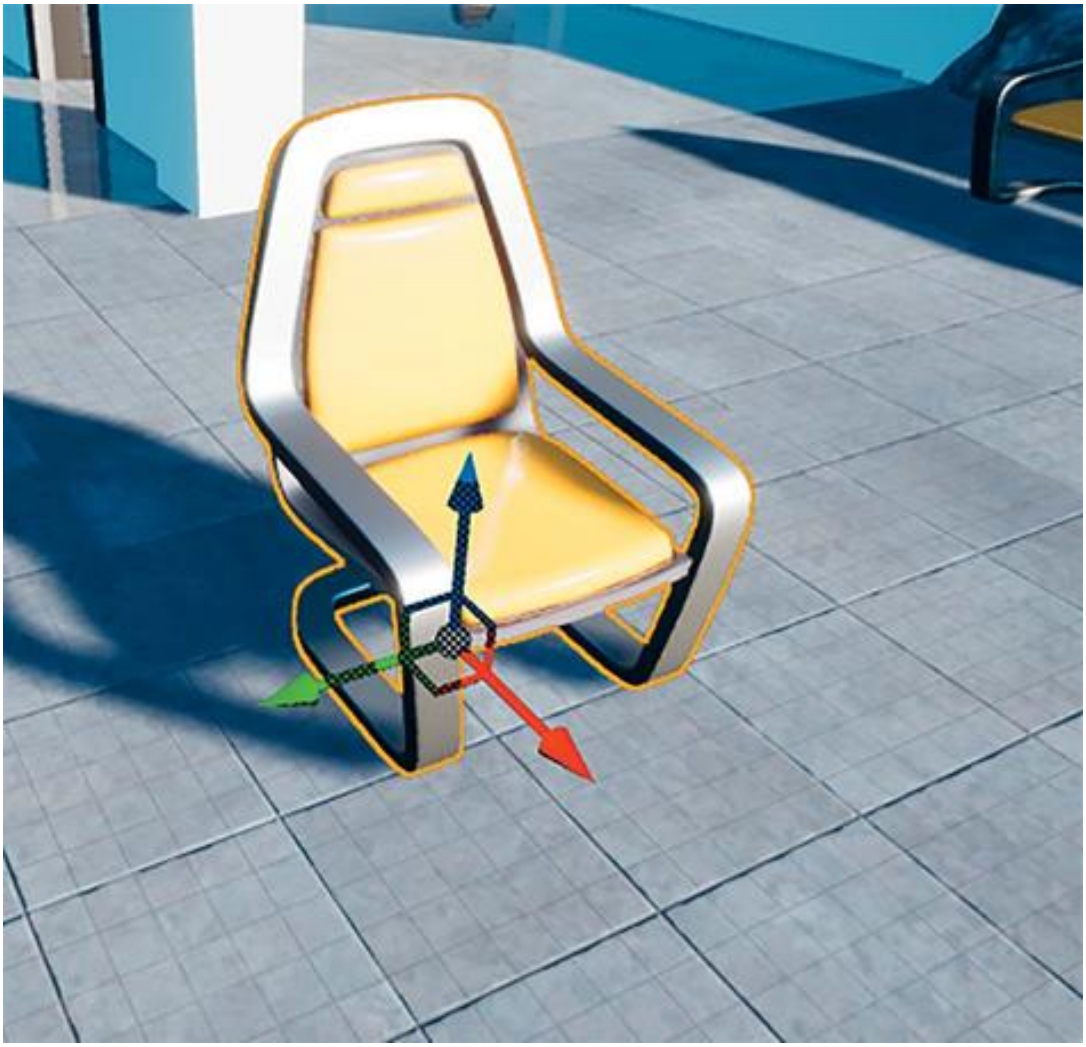

Рисунок 6.9 Pivot point объекта в UE4, где красный — Roll или X, зеленый — Pitch или Y, а синий — Yaw или вращение Z

Наличие пивота в центре объекта, как правило, не лучшее место. Пивот — это место, где актор будет масштабироваться, вращаться и двигаться в UE4. Это также место, где он будет привязываться к поверхностям других акторов, когда вы поместите его на уровень. Для стульев и другой мебели это означает, что вы должны разместить ось на высоте, где мебель ударяется о землю. Вы должны установить у картины в рамке pivot сзади, где она будет прикреплена к стене.

UE4 относится к вращениям как комбинации осей Roll, Pitch и Yaw. Крен — это вращение влево и вправо вокруг оси X, а тангаж — вверх и вниз вокруг оси Y; рыскание вращает актора вокруг оси Z (рисунок 6.9).

Это означает, что вы должны моделировать акторов в своем 3D-приложении, обращенных к положительному X, точно так же, как они появляются в UE4. UE4 будет управлять переводом оси Y.

6.8 Использование осмысленной геометрии

Реквизит, предоставленный клиентом, является отличной отправной точкой, но он часто бывает недостаточно хорош, чтобы использоваться для визуализации высокого класса.

Использовать модели, предоставленные вместе с руководством от клиента, в качестве эталона и переделывать большую часть мебели. Вы видите, что большинство моделей очень просто моделируются. Хотя современные видеоигры используют очень трудоемкий рабочий процесс для мешей, персонажей и окружающей среды, проекты визуализации обычно не имеют времени, бюджета или потребности в таком уровне внимания, чтобы тратить его на каждый ассет.

Геометрия должна быть чистой, хорошо UV-преобразованной и довольно низко полигональной. UE4 и современное оборудование позволяет обрабатывать миллионы треугольников в секунду, и большинство сцен визуализации относительно редки по сравнению с окружением видеоигр. Это позволяет вам использовать больше полигонов для определения ваших мешей, но будьте внимательны, тотальная оптимизация все еще важна.

6.9 Экспорт

Хотя вы можете экспортировать один.FBX со всеми реквизитами в одном файле, — это неэффективно и может затруднить рабочий процесс, так как нужно повторно экспортировать все одинаковые меши в файл FBX, если нужно внести изменения в один меш.

Упражнение 10. Экспорт.

Используйте следующие настройки или аналогичные настройки в экспортере вашего 3D-приложения (рисунок 6.10).

1. Убедитесь, что вы экспортируете Smoothing Groups, Tangents и Binormals, а также что Triangulate и Preserve Edge Orientation установлены на true.

2. Если вы работаете на сцене, которая не масштабируется до сантиметров, вы также можете масштабировать данные с помощью системы масштабирования единиц измерения экспортера. Для этого установите Automatic переключатель в положение false и установите параметр Scene units converted to в Centimeters.

Вы должны экспортировать каждый реквизит как один файл FBX, называя каждый файл FBX так, как вы хотите. UE4 использует имя файла для установки имени меша.

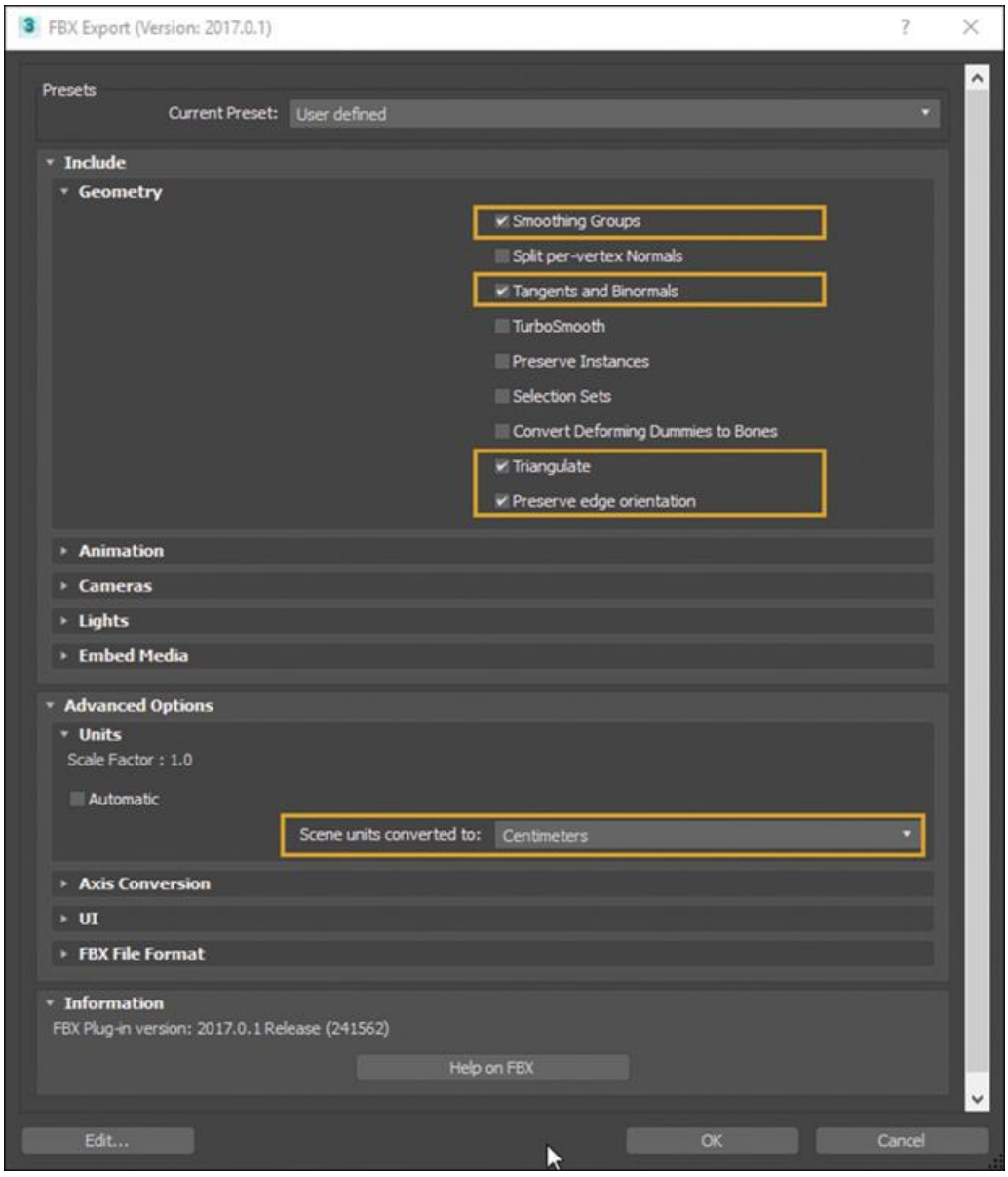

Рисунок 6.10 Диалоговое окно 3D Studio Max FBX Export, показывающее настройки, которые обеспечат возможность импорта вашей модели в UE4 со всеми smoothing groups, UV-координатами и правильной ориентацией ребер, чтобы ваши ассеты отображались в UE4 точно так же, как в вашем 3D-приложении

6.10 Импорт

Как и в случае с мешами архитектуры, импортируйте реквизит с помощью кнопки Import в Content Browser или перетаскивая файлы в Content Browser. Вы можете импортировать все ваши файлы сразу или один за другим.

Как и прежде, появится диалоговое окно Import Dialog for Static Meshes. Используйте настройки, показанные ранее на рисунке 6.5.

Auto Generate Collision

Не рекомендую использовать автоматически создаваемые столкновения. Обычно используются столкновение на своих мешах для реквизита для архитектурных визуализаций, потому что это часто слишком ограничивает движение пользователя.

Transform Vertex to Absolute

Установка Transform Vertex to Absolute в значение false позволяет UE4 использовать авторский пивот для поворота в вашей модели, а не использовать 0,0,0 сцены и запекать вращение и другие преобразования в вершины меша. Если вы не переместили свой реквизит в 0,0,0 в 3D-приложении перед экспортом, то потребуется установить для него значение false.

Заключение

Импорт данных и контента в UE4 может показаться пугающим на первый взгляд, но подготовка и организация вашего контента, а также обеспечение правильных настроек импорта и экспорта для него делают процесс плавным и предсказуемым.

Теперь у вас должно быть хорошее понимание различий между объектами реквизита и архитектуры и того, как надежно

поместить их в UE4 (рисунок 6.11). Теперь вы готовы разместить эти ассеты и начать строить интерактивный мир.

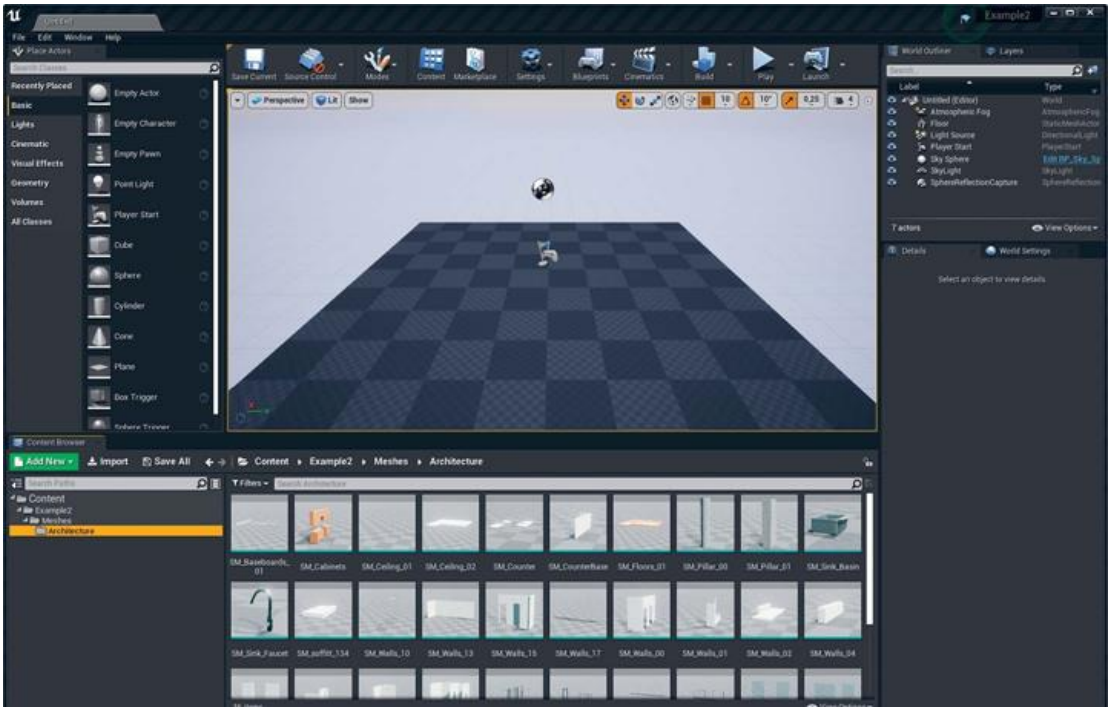

Рисунок 6.11 Content Browser, показывающий импортированные в Architecture Static Mesh Assets

7 НАПОЛНЕНИЕ СЦЕН

Заполнение сцен в UE4 может быть сложной задачей для визуализации. Проектировщикам часто поручают представлять данные или проекты, которые требуют точности и аккуратности.

7.1 Сборка сцен для визуализации

UE4 предлагает мощный набор инструментов для левел-дизайнеров (Level Designers, LDs) и художников, чтобы быстро создавать фантастические игры и визуализации. Художники могут перетаскивать ассеты, огни и другие активы в мир с помощью Content Browser; легко перемещать, масштабировать и поворачивать их с помощью инструментов преобразования, а также легко менять свойства с помощью Details Panel.

Построение сцен для визуализации может быть сложной задачей, поскольку вам часто нужно очень точно отображать данные, ставя точность выше свободы творчества.

Для архитектурных визуализаций часто приходится балансировать между потребностью в точности и художественной свободой для создания невероятно красивых пространств.

Вы должны точно разместить свои архитектурные меши в пределах уровня. Здесь нет никакой художественной свободы, потому что это данные, которые вам дали визуализировать.

Однако ваш реквизит должен быть вручную помещен на уровень, перемещен, повернут и свободно масштабирован, наполняя все жизнью и смыслом.

7.2 Настройка Level

Прежде чем вы построите уровень, нужен сам уровень. По умолчанию UE4 предоставляет простой уровень с облаками, источником света и Player Start Actor, расположенным на Box Static Mesh Actor.

Можно начинать все заново с совершенно нового уровня, избегая любых проблем, которые могут возникнуть при использовании пресета.

Создание New Level

Чтобы начать с чистого листа для нового проекта, выберите File > New Level в Editor (рисунок 7.1).

Это создает новый, совершенно пустой уровень. Сохраните свой уровень, выбрав File > Save Current As.

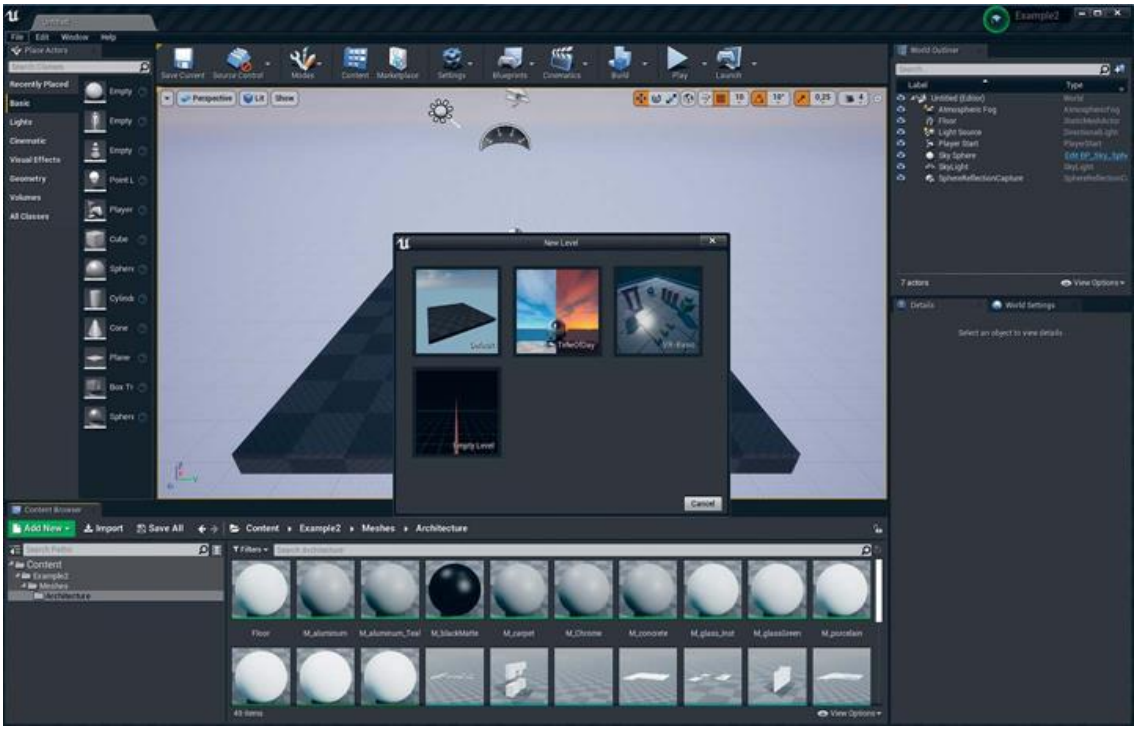

Рисунок 7.1 Создание нового пустого Level; уровень по умолчанию отображается во Viewport

7.3 Добавление простых источников света

Как и в первом проекте, первая задача состоит в том, чтобы поставить несколько простых источников света, прежде чем размещать какие-либо меши. Без света UE4 делает сцену либо черной, либо в Unlit View Mode, что затрудняет просмотр того, что вы делаете. Добавление ряда основных источников света в первую очередь позволяет проще видеть некоторые вещи и работать с ними.

Atmospheric Fog

На панели Place Actors выберите пункт Visual Effects и перетащите Atmospheric Fog в Viewport. Это обеспечивает горизонт, атмосферное рассеивание (голубое небо) и солнечный диск, который делает небо простым и приятным.

Directional Light

Выберите Lights на панели Place Actors и перетащите Directional Light в Viewport. Это будет работать как ваше Солнце.

Выберите Directional Light, найдите опцию Atmosphere / Fog Sun Light и установите ее в true (возможно, вам придется посмотреть в разделе Advanced Options rollout). Это позволяет направленному свету контролировать цвета актора атмосферного тумана и положение солнечного диска.

Sky Light

Перетащите Sky Light на сцену. Это образец окружающей среды вокруг него и использует данную информацию для обеспечения косвенного освещения сцены. Теперь у вас должна быть базовая настройка освещения (рисунок 7.2).

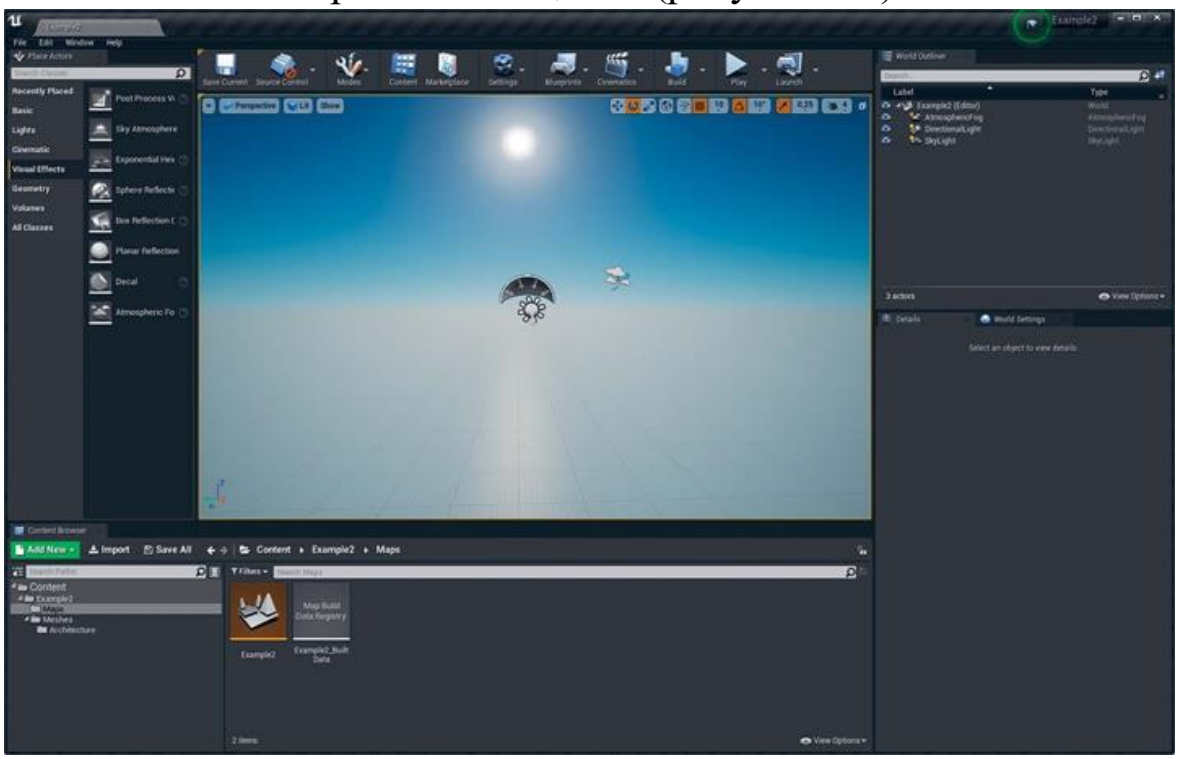

Рисунок 7.2 Базовая настройка освещения показывает выбранный Directional Light и Atmosphere / Fod Sun Light, установленный в true на панели Details, что позволяет установить положение Солнца в небе.

Размещение Architecture Static Meshes

Подготовив свои архитектурные ассеты к использованию origin FBX в качестве pivot point (0,0,0), вы легко разместите их на уровне в том самом месте, куда вы их экспортировали.

Dragging и Dropping мешей

Перетащите ваши меши на Viewport из Content Browser. Вы можете выбрать один или несколько Static Mesh Actor с помощью обычных сочетаний клавиш (Shift-click, Ctrl / Cmd-click и т. д.) и перетащить их в любое место Viewport (рисунок 7.3).

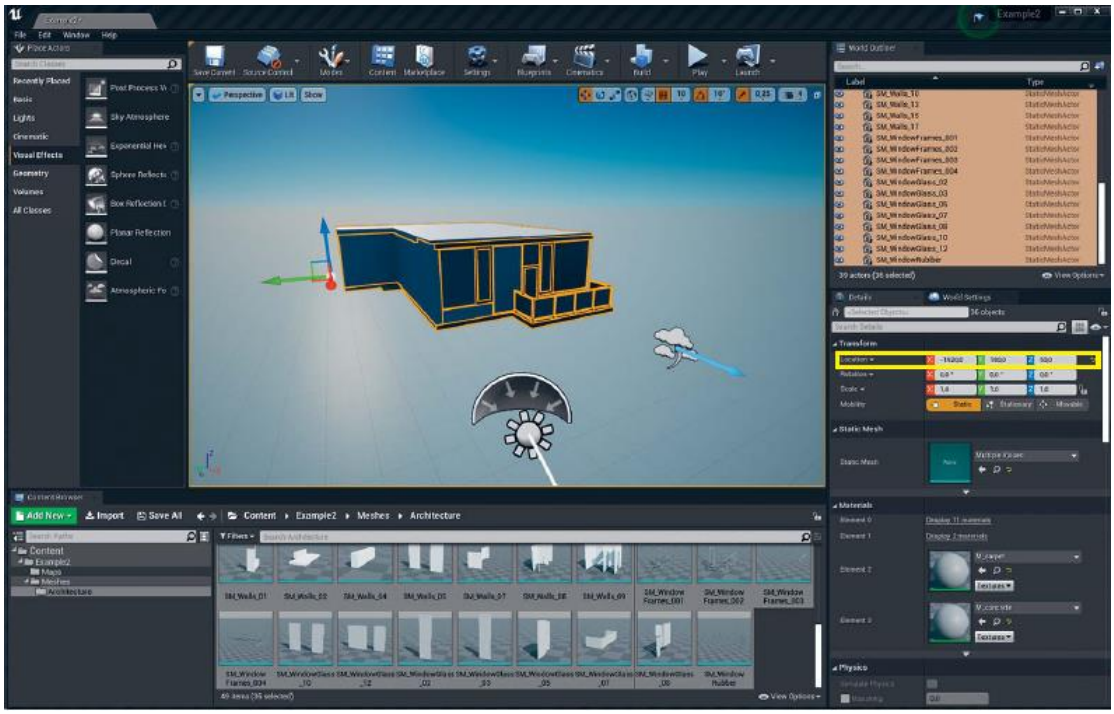

Рисунок 7.3 Результаты операции перетаскивания, а затем установка свойства Location размещенных мешей на 0,0,0

7.4 Настройка Location на ноль

Когда вы перетаскиваете эти меши на уровень, созданные Static Mesh Actors выбираются в World Outliner и Level. Если вы посмотрите на панель Details, то увидите, что можете редактировать свойства всех мешей одновременно. Это позволяет легко установить местоположение на 0,0,0 (рисунок 13.3).

Теперь все ваши архитектурные меши размещаются точно там, где они были в вашем 3D-приложении.

Размещение декоративных мешей

Теперь, когда стены, полы и другая архитектура размещены, пришло время расставить реквизит.

Как и в большинстве случаев с контентом, просто перетащите реквизит, чтобы извлечь его из Content Browser на карту и поместить на свой уровень.

В этом примере мне дали свободу на заполнение сцены мебелью и реквизитом, выбранной клиентом и уже готовым реквизитом из моей библиотеки моделей.

7.5 Привязка к поверхности

Когда вы впервые перетаскиваете модель на уровень, обратите внимание, что она проецируется на сцену и «прилипает» к тому, на которое приземляется. Такая привязка отлично подходит для того, чтобы поместить реквизит на уровне, но что произойдет после подобного размещения?

Включение Surface Snapping на панели Viewport блокирует актора, перемещаемого по поверхности так, как если бы он летал над ней (рисунок 7.4).

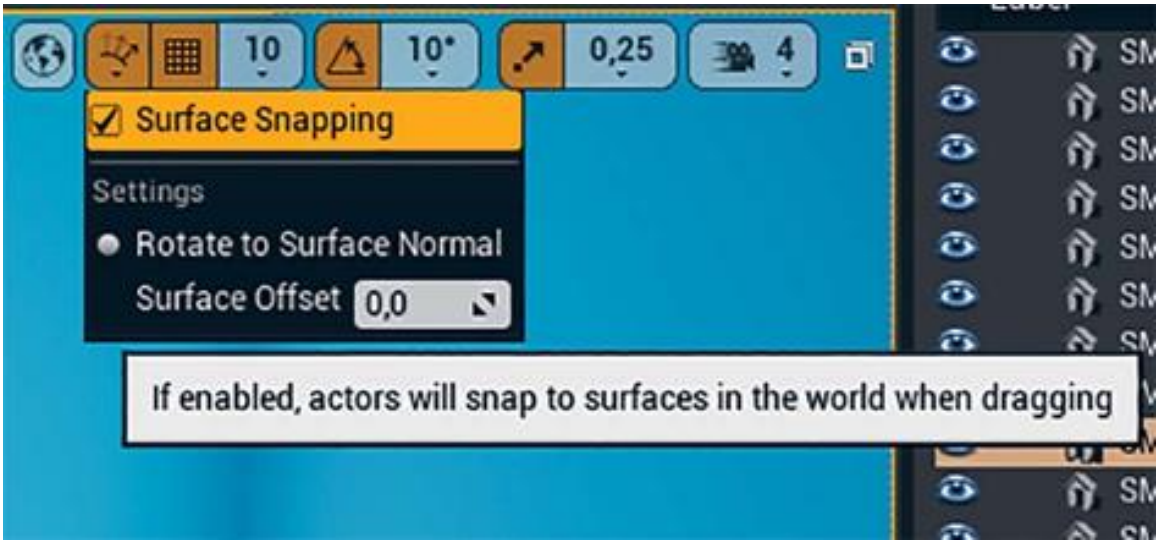

Рисунок 7.4 Выпадающее меню Surface Snapping

Вы также можете включить Rotate to Surface Normal, чтобы размещаемый меш переориентировался на полигональную поверхность, на которой вы его размещаете. Это отлично подходит для размещения акторов на стенах, потому что они будут вращаться, когда вы перетаскиваете их от стены к стене.

7.6 Клонирование и копирование

Обязательно используйте инструменты клонирования, такие как copy-paste и метод Alt-Drag для создания копий. Эти методы делают заполнение сцены очень быстрым.

Shift+Dragging

Если вы удерживаете нажатой клавишу Shift во время перемещения актора с помощью механизма Transform, вид будет следовать за этим актором. Вы можете использовать эту технику в сочетании с методом Alt-Drag — это отличный способ перемещения акторов по большим уровням с помощью перспективного Perspective Viewport.

7.7 Организация сцены

Добавление меша, света и других акторов на сцене вскоре приводит к сотням акторов на ней. Их организация необходима для быстрого рабочего процесса, и UE4 предлагает инструменты, помогающие управлять вашей сценой.

World Outliner

Когда вы добавляете акторов на свой уровень, они появляются в списке World Outliner. Этот список может быстро стать громоздким и трудным для навигации. У вас есть несколько вариантов для его очистки.

Вы можете легко создавать папки с помощью кнопки New Folder (расположенной в правом верхнем углу World Outliner рядом со строкой поиска), но предпочтительнее использовать метод щелчка правой кнопкой мыши. Выберите ассеты, которые хотите поместить в папку в World Outliner, и щелкните правой кнопкой мыши на одном из них, что откроет контекстное меню (рисунок 7.5).

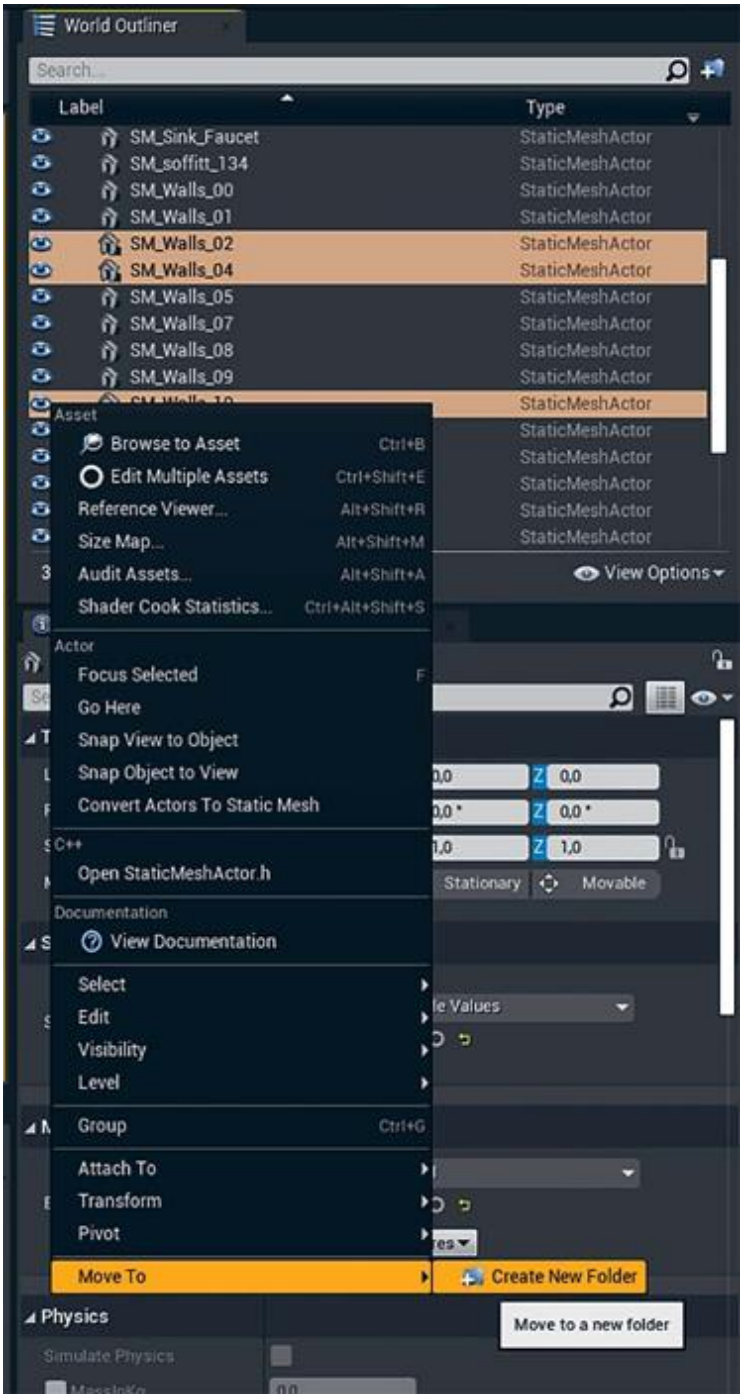

Рисунок 7.5 Добавление акторов в новую папку в World Outliner с помощью меню правой кнопки мыши

Здесь вы можете добавить их в новую папку или в существующую и не искать ее в World Outliner.

7.8 Слои

Менее очевидным организационным методом, но, вероятно, удобным для художников визуализации, является

система слоев (Layer). Как и большинство 3D-приложений, UE4 имеет полноценную систему слоев, в которой вы можете легко скрывать, отображать и выбирать акторов через слой.

Интерфейс слоев не открыт по умолчанию, но вы можете получить доступ к нему через меню Window (рисунок 7.6).

Рисунок 7.6 Включение вкладки Layers; обратите внимание на вкладку Layers справа с размещенными в ней реквизитами

Вы также можете использовать как системы World Outliner, так и системы слоев, поскольку они не влияют на производительность среды выполнения или функции; они просто помогают вам организовать свою сцену.

Самый легкий способ - это использовать систему папок World Outliner, чтобы сохранить свою сцену организованной.

7.9 Группировка

Группировка (Groupping) — это отличный способ сделать многоразовые или легко выбираемые наборы акторов на вашем уровне.

Выберите акторов, которые хотите сгруппировать, щелкните правой кнопкой мыши и выберите пункт Group (рисунок 7.7). Создается Group Actor, и выбранные акторы привязываются к нему. Выбор любого из акторов или группы акторов выбирает их всех.

Группы отлично подходят для построения многоразовых наборов геометрии в ваших сценах.

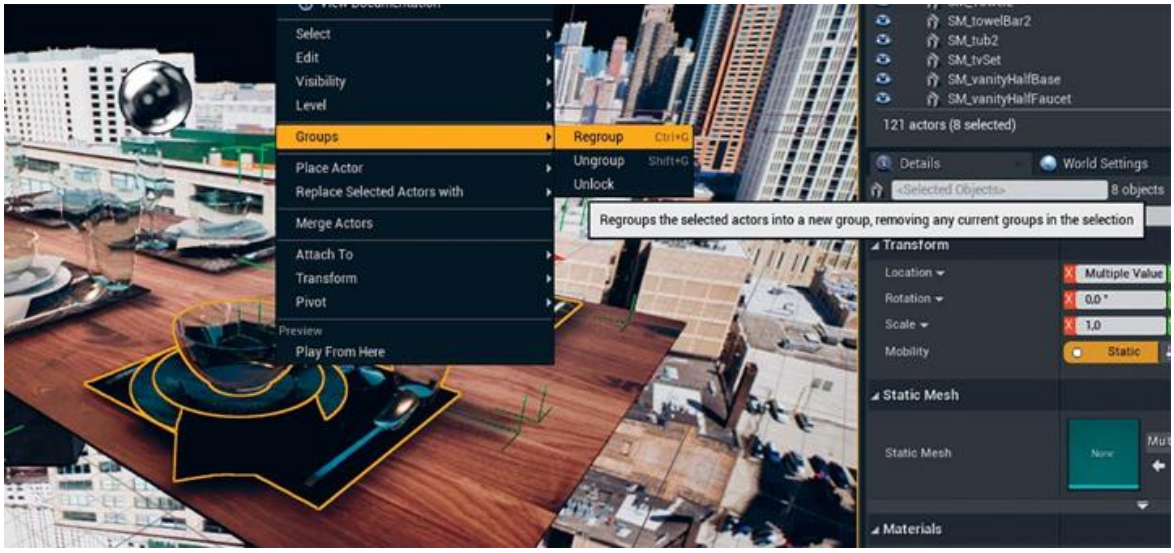

Рисунок 7.7 Группировка мешей вместе для дублирования вокруг сцены

После того как акторы сгруппированы, вы можете Ungroup или Unlock группу для редактирования с помощью контекстного меню. Вы не можете назвать или изменить Group Actor.

Actor Blueprints

Вы можете взять набор акторов (включая свет, частицы, звуки и т. д.) и создать из них объект Blueprint. У этого есть несколько преимуществ перед группами.

Чтобы создать Actor Blueprint из акторов вашего уровня, выберите ассеты, которые вы хотите преобразовать, и выберите Convert Selected Components to Blueprint Class (рисунок 7.8).

Затем вы должны выбрать, где сохранить новый Blueprint Asset в Content Browser. Этот Blueprint содержит компоненты, которые ссылаются на статические меши и другие выбранные вами классы. Он не создает копию или не объединяет их в новый меш.

Теперь вы можете изменить один Blueprint Asset, и все Level Actors, основанные на нем, будут обновлены. Actor Blueprint отлично подходит для создания таких вещей, как свет или другие сложные наборы акторов или даже просто группы акторов, которые вы хотите повторно использовать.

Рисунок 7.8 Преобразование выбранных компонентов в класс Blueprint

Заключение

Размещение контента в UE4 — это одновременно весело и сложно. Теперь вы должны лучше понимать, как поместить архитектурные меши на сцены в точном положении, в котором они были в вашем 3D-приложении, и как организовать и поддерживать сцену по мере добавления все большего количества контента.

Когда сцена заполнена (рисунок 7.9), пришло время заполнять ее освещением, материалами и эффектами постобработки, чтобы достичь своей цели создания фотореалистичной сцены.

Рисунок 7.9 Наполненная сцена с предварительным освещением

8 АРХИТЕКТУРНОЕ ОСВЕЩЕНИЕ

Разработка чистого, точного, реалистичного освещения очень важна для создания визуализаций. Взаимодействие света, теней, отражений и материалов оживляет миры, создавая глубину и рассказывая истории. Далее вы узнаете, как настроить великолепное освещение Global Illumination (GI) с помощью Lightmass.

Получение большего от Lighting

Системы Lighting и Materials в UE4 тесно связаны между собой. Вам нужно работать между ними для достижения конечного результата. Эта глава демонстрирует, как освещать интерьер сцены с помощью UE4.

Высокий динамический диапазон конвейера рендеринга Unreal’a сочетается с такими эффектами, как auto-exposure и bloom, для создания реалистичной, теплой и динамической

среды освещения. Правильно настроить освещение с помощью этих методов может быть сложно, но, используя некоторые базовые настройки, вы сможете быстро получить потрясающий результат.

Вы изучите, как настраивать свойства и параметры Level и Static Mesh для достижения отличного освещения с приемлемым временем на его расчеты.

Будет показано, как использовать эффекты post-processing, такие как vignette, color grading и depth of field, для создания более фотореалистичного изображения (рисунок 8.1).

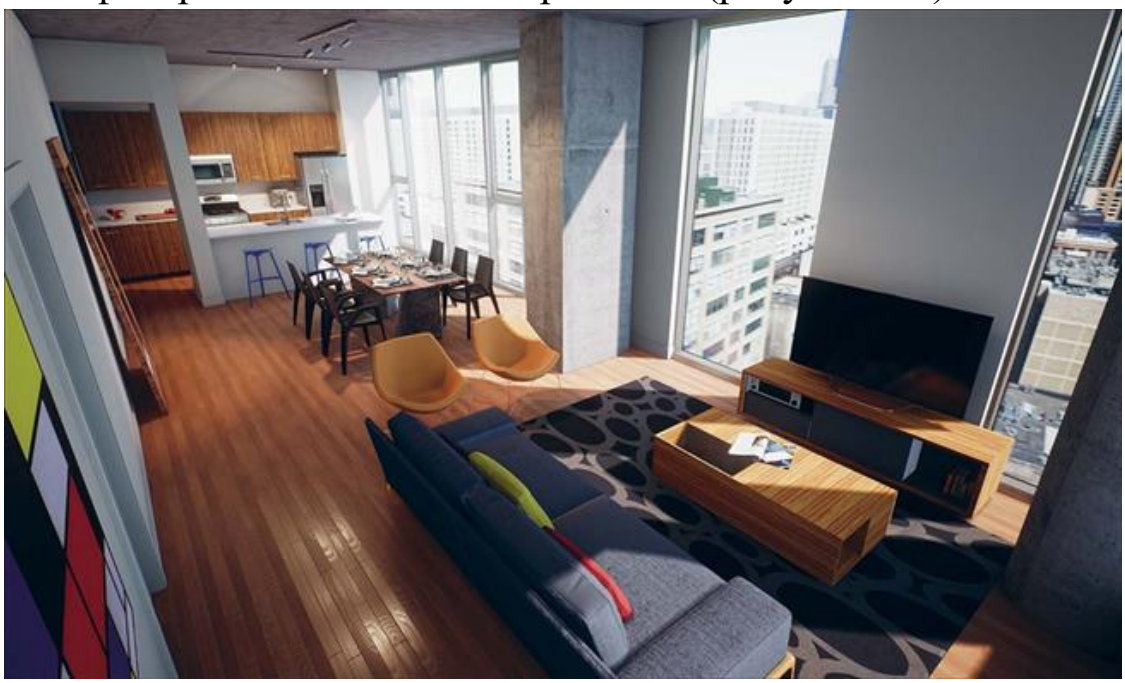

Рисунок 8.1 Предпросмотр финального освещения

Наконец вы изучите, как расположить Reflection Capture Actors на уровне для улучшения освещения и обеспечения более точно выглядящих зеркал.

8.1 Материалы и освещение

Как и весь рендеринг Global Illumination (GI), освещение в UE4 напрямую зависит от материалов, и наоборот. Материалы окрашивают и смягчают отраженный свет, который информирует нас о свойствах поверхности, отражаясь от них.

Из-за этого вы не можете разделить их полностью.

Static Lighting с Lightmass

UE4 произвел огромный резонанс в индустрии архитектурной визуализации благодаря своим впечатляющим визуальным эффектам. Один из способов, благодаря которому UE4 справляется с рендерингом невероятно хорошо освещенных сцен так быстро, является предварительное вычисление освещения с помощью GI рендера, называемого Lightmass.

Lightmass рассчитывает попадающий на поверхность свет для каждого Static Mesh в вашей сцене, используя, к примеру, Mental Ray для вычисления Global Illumination, основанного на фотонах. Затем он записывает эти данные для нескольких Textures, называемых Lightmaps и Shadowmaps. Этот процесс называется построением Static

Lighting. После завершения рендеринга освещения Textures автоматически импортируются и применяются.

Как следует из названия, вы не можете перемещать или менять объекты, освещенные Static Lighting или источниками света, после их построения. Если вы добавите, удалите, переместите или каким-либо еще способом измените Static Lights или Static Meshes на сцене, то нарушите освещение и должны будете повторно построить его.

Обратите внимание, что вы можете добавить источники света к уже построенной сцене с освещением. Несмотря на то что вам нужно будет построить освещение снова, чтобы увидеть влияние GI недавно размещенных источников света, существующий static lighting останется. Это может быть полезно, так как вы добавляете все больше и больше источников света на вашу сцену.

Lightmaps и Shadowmaps хранятся в специальном BuildData UMAP Level файле (пользовательский файл формата UE4, хранящийся как Levels) вместе с вашими сохраненными уровнями. Этот файл и ваш проект будут значительно увеличиваться по мере того, как вы станете добавлять больше Assets, так как каждый из них увеличивает расход памяти на Lighting Textures.

8.2 Настройка Sun и Sky Lights

В предыдущей главе вы разместили Directional Light Actor, Sky Light Actor, и Atmospheric Fog Actor. Эти три актора обеспечивают основу для построения освещения, но они требуют некоторой настройки для работы со Static Lighting.

Солнечный свет

Directional Light будет действовать как солнечный свет, обеспечивая интенсивный прямой свет, который отлично подходит для решения задачи с глобальным светом (GI).

Тремя основными типами или классами источников света в UE4 являются Point, Spot, и Directional. Освещения Point и Spot излучают из одной точки в пространстве. Они отлично подходят для близкого освещения, такого как лампы, светильники, и точечных фонарей. Но они не очень подходят для солнца, которое, с нашей точки зрения, на Земле испускает лучи так, как если бы они шли параллельно друг другу. Для этого вам понадобится свет, который симулирует единственный источник прямого освещения. Внесите класс Directional Light Actor.

Как и большинство Actors, вы можете легко разместить освещение на вашей сцене, перетащив его из Class Browser в Viewport.

Вы можете разместить освещение в любом месте на сцене, так как оно не излучается из этой точки; важен только поворот.

Найдите приятный или реалистичный поворот для освещения. Динамические тени действуют как пред просмотр и дают вам представление, где будут статические тени (рисунок 8.2). Оглянитесь вокруг, когда закончите. Динамические тени UE4 обеспечивают хорошую приблизительную точность, показывая, где будут располагаться запеченные тени на сцене.

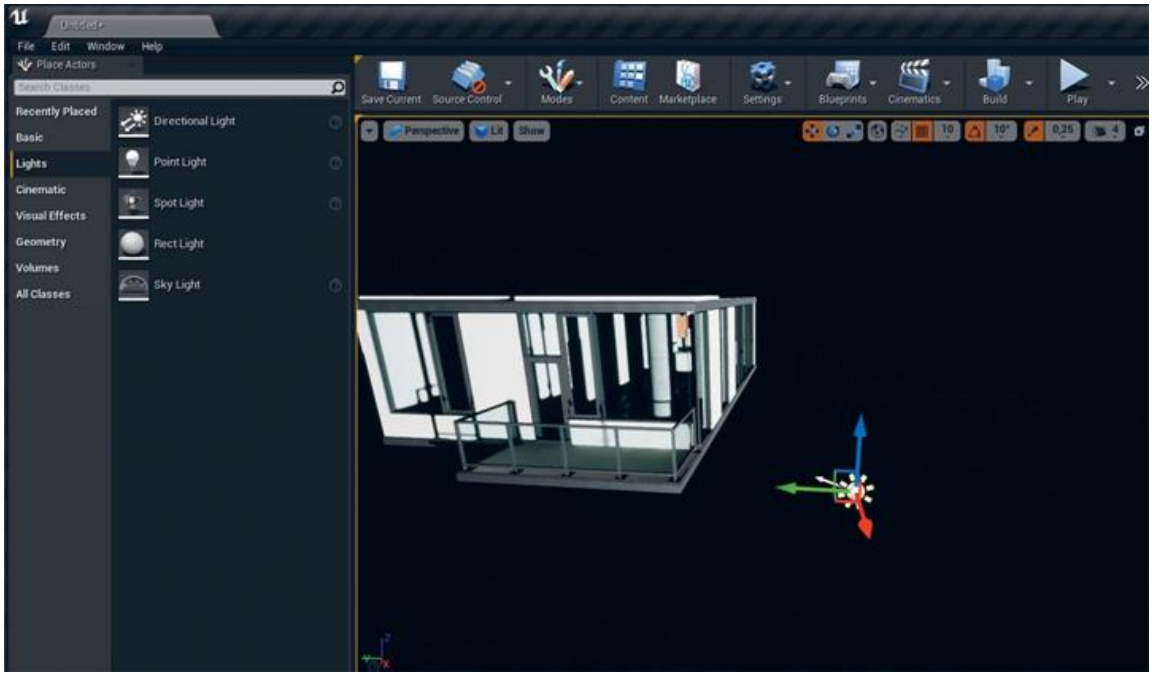

Рисунок 8.2 Размещение Directional Light

В Details Panel показаны различные доступные для выбранного источника света параметры (рисунок 8.3). Вам нужно отрегулировать несколько настроек для получения лучшего качества и производительности от этого Light Actor.

Рисунок 8.3 Свойства Directional Light в Details Panel

Mobility

Первым и наиболее важным параметром, который нужно изменить, является Mobility. Для этого источника света вам нужно использовать Stationary Light. Это как Static Light (он не может перемещаться), но он не записывает свое прямое освещение в Texture; скорее динамически рассчитывает его. Это означает, что вы можете менять яркость и цвет после построения освещения.

Обратите внимание, что изменение Intensity не влияет на освещение GI, так как оно запекается в Static Textures.

Intensity

Для получения эффекта яркого солнечного света используйте значение Intensity 8–10. Это гарантирует, что солнце является самым ярким источником освещения на вашей сцене.

Indirect Lighting Intensity

Поскольку интенсивность Stationary Lights может быть изменена, вам нужно сообщить Lightmass, насколько ярким рассчитывать это освещение. Для этого вы можете использовать параметр Indirect Lighting Intensity для регулировки яркости GI этого источника света. Установите его где-нибудь между 1.0 и Intensity, установленной для вашего источника освещения.

Temperature

Проверьте галочку на Use Temperature и установите Temperature около 5000–5500. Это обеспечит слегка желтое солнце, которое все еще значительно «белее», чем обычный свет от ламп накаливания, у которых значение 2700–3500.

Atmosphere Sun Light

Убедитесь, что выбран параметр Atmosphere Sun Light, который обеспечит совместную работу с Atmosphere Fog Actor для создания физически корректного освещенного неба, которое обеспечит цвет окружения для сцены. Если вы не видите его в начале, то этот параметр может быть скрыт в Advanced Options источника света.

UE4 оснащен системой атмосферного освещения рассеянного тумана, которая симулирует атмосферу земли. Управляемая с помощью Atmospheric Fog Actor и Directional Light Actor. Просто добавьте Environmental Fog Actor на вашу сцену. Если у вашего Directional Light Actor включен Atmosphere Fog Light, он будет определять местонахождение солнечного диска и устанавливать цвета неба в зависимости от позиции солнца, создавая простой, но эффектный skybox.

Sky Light

Для симуляции окружающего света атмосферы UE4 использует Sky Light Actor. Этот специальный источник света может запечатлеть существующую сцену как Cubemap, для использования непрямого освещения, или вы можете предоставить ему HDR Texture.

В данном примере просто запечатлейте сцену.

Разместите этот источник света на вашей сцене там, где вы сможете легко его найти. Вам также нужно убедиться, что он не находится в какой-нибудь геометрии, таким образом захватывая незакрытое небо. Sky Light Actor содержит несколько настроек для получения оптимальных результатов (рисунок 8.4).

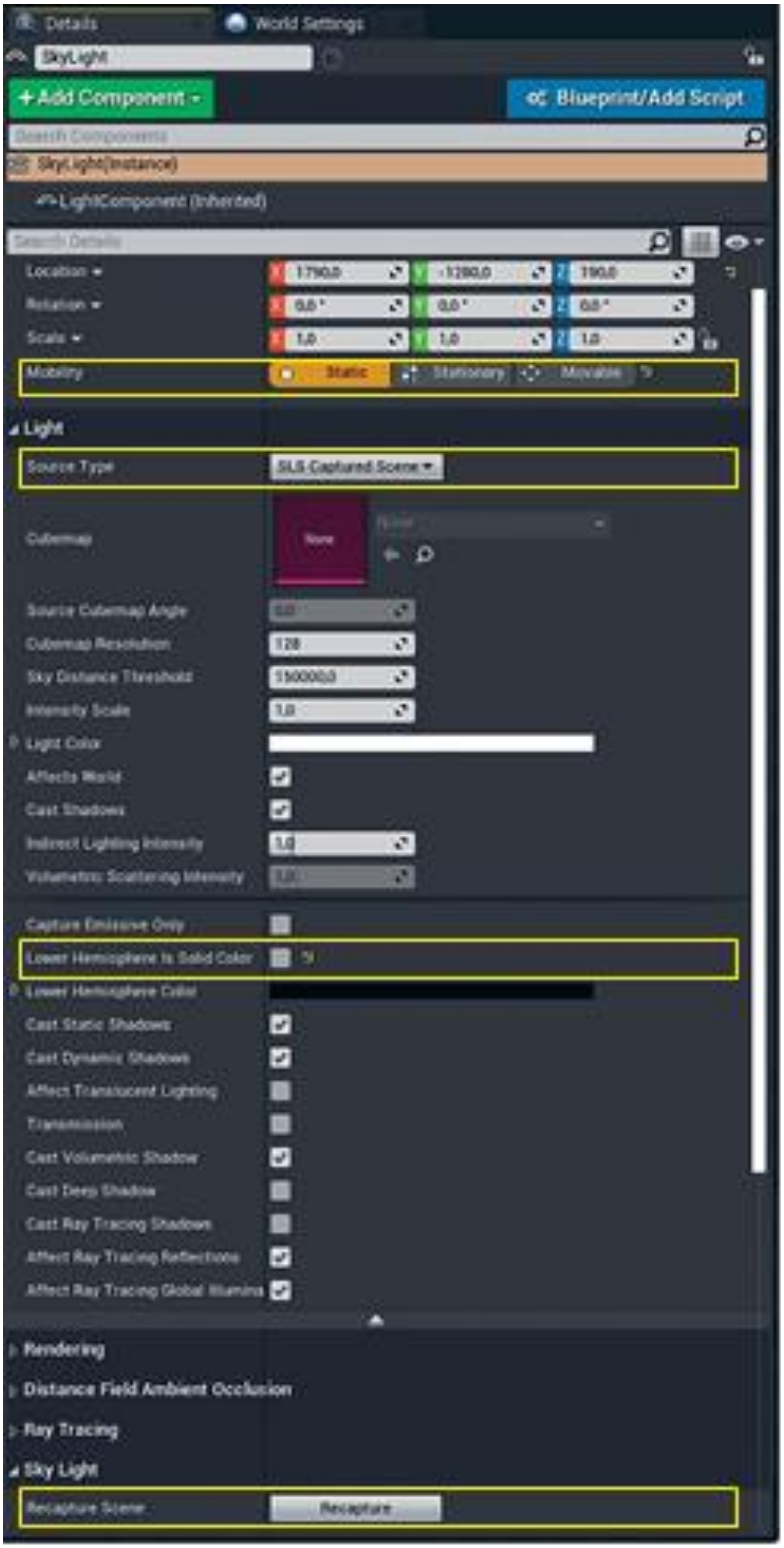

Рисунок 8.4 Расположение Sky Light

Mobility

Sky Light установлен на Static для этой сцены. Вы также можете выбрать Stationary для этой сцены, но вы лишитесь определенного уровня качества и детализации вашего

непрямого освещения неба с возможностью динамического изменения свойств Intensity, Source Cubemap Angle и Cubemap Texture без перерасчета освещения.

При установке Mobility в значение Static запекается все непрямое освещение карт света и тени, созданных для источника света Lightmass. Это делается, чтобы избежать проблем мобильного пайплайна, например, утечек света или неточного освещения.

Source Type

Вы можете выбрать между импортированной HDR Cubemap или Cubemap сцены, взятую Sky Light Actor в Editor. В данном случае используйте сцену в качестве освещения, установив Source Type на SLS Captured Scene.

Lower Hemisphere Is Solid Color

Настройка Lower Hemisphere Is Solid Color является личным предпочтением. Установка его на true заменяет нижнюю половину захваченного Cubemap одним сплошным цветом. Это может ограничить количество света, падающего на площадь, поэтому предпочтительнее оставлять его на false для интерьера сцены, где важно получать как можно больше освещения на сцене.

Recapture Scene

Если вы измените освещение или сцену, Sky Light не будет динамически обновлять Cubemap, используемую для освещения сцены. Вы должны вручную повторно захватить Cubemap с помощью кнопки Recapture Scene.

Билд освещения

Теперь, когда сцена содержит элементарное освещение, вы можете построить освещение в первый раз.

Для построения освещения нажмите кнопку Build в панели инструментов Editor. Вы можете нажать на стрелку рядом с ним, чтобы увидеть настройки. Наиболее важным параметром является Lighting Build Quality. Он позволяет перейти от быстрого превью к хорошо выглядящему, но медленному для рендеринга production quality build (рисунок 8.5).

Lightmass предлагает несколько предустановок для качества освещения от preview до production. Production lighting builds могут занимать на порядок больше времени, чем preview builds, но билды низкого качества могут содержать артефакты и несоответствия, которые трудно отличить от реальных проблем с геометрией или освещением.

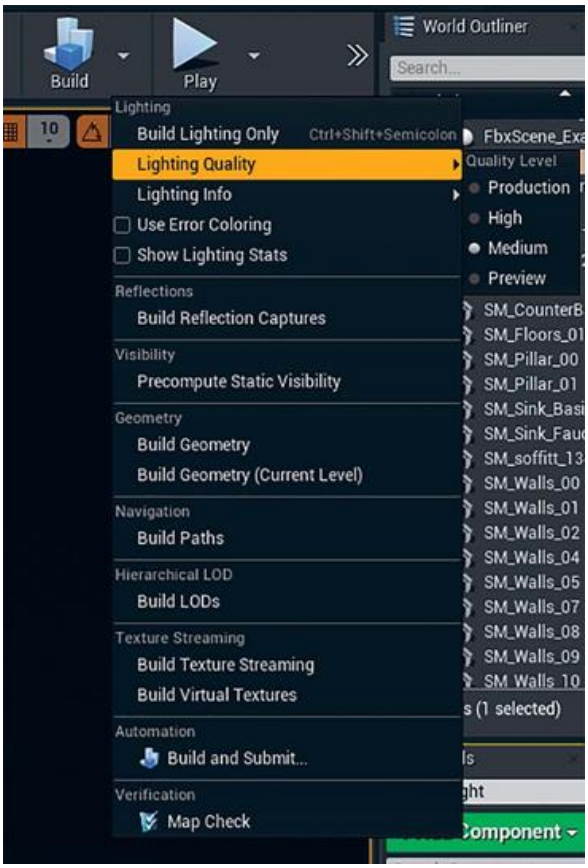

Рисунок 8.5 Настройка Lighting Quality из выпадающего списка Lighting Build

Рисунок 8.6 показывает сравнение построений освещения между preview и production. Вы можете заметить, что точность значительно уменьшена в построении preview, но время на расчет освещения также значительно сократилось.

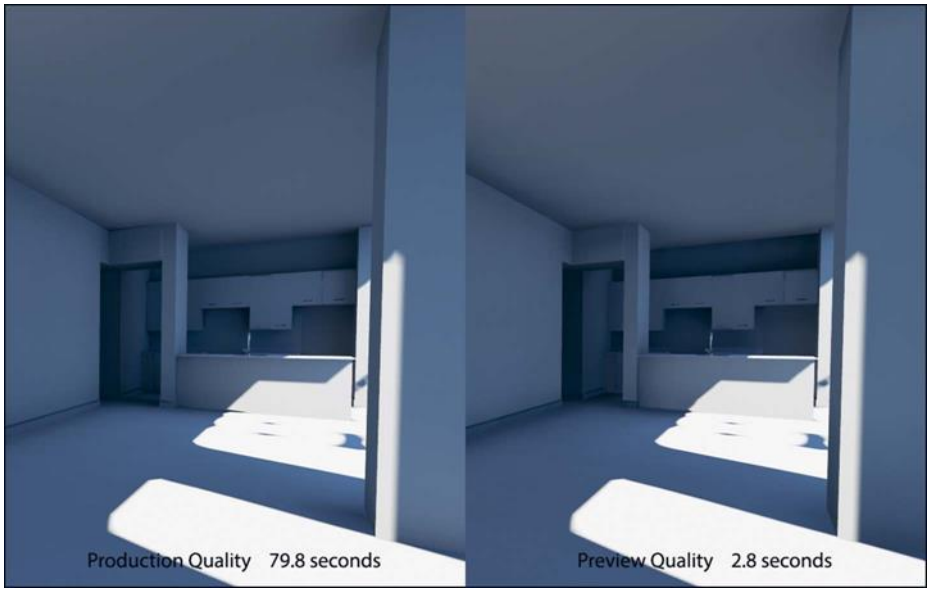

Рисунок 8.6 Первое построение освещения в качествах production и preview

Как вы можете заметить, даже при использовании качества production освещение остается все еще не очень хорошим. Тени являются неопределенными, блочными и в низком разрешении, а также содержат ошибки на плинтусе и в темных областях. Сделав несколько настроек, вы можете достаточно быстро заставить эту сцену выглядеть гораздо лучше.

Настройка Lightmass с архитектурной визуализацией

Первым шагом для «причесывания» вещей является повышение настроек Lightmass. Architecture Visualizations определяет приоритет качества изображения над временем рендеринга, и в UE4 это ничем не отличается. Вы увеличиваете построение освещения, но в итоге получаете сцену, которая выглядит гораздо лучше.

Настройки для Lightmass хранятся на каждом основном Level и могут быть изменены в настройках свойств World. Панель Lightmass предлагает множество настроек (рисунок 8.7). К счастью, настройки по умолчанию UE4 хороши для большинства обстоятельств. Каждый параметр существенно влияет на время построения, поэтому всегда стоит начинать с малого и продвигаться дальше.

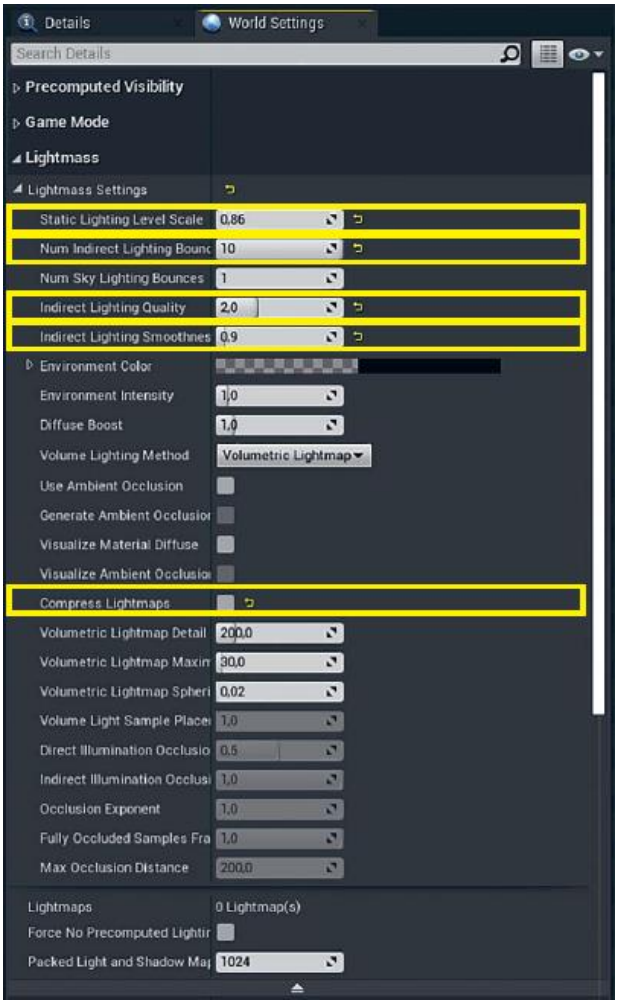

Рисунок 8.7 Настройки Lightmass для мира

Static Lighting Level Scale

Если вы исследовали Lightmass онлайн, вероятно, вы видели некоторые сложные настройки файла .ini для получения Lightmass, которая производит чистые lighting solutions для архитектурных визуализаций.

Эти настройки в значительной степени стали ненужными в более поздних версиях движка (возможно, это вредит качеству изображения и, безусловно, времени на построение). Epic Games проделали огромную работу по улучшению Lightmass для визуализации, и вам нужно всего лишь откорректировать настройки, выставленные в World Settings.

Static Lighting Level Scale

Параметр Static Lighting Level Scale контролирует плотность фотонов на сцене. Маленькие числа означают более

плотные фотоны с большей детализацией. Это может значительно повысить время на построение. Вы можете оставить этот параметр на значении по умолчанию, равном 1,0, или немного уменьшить его. Любое значение ниже 0,8 повысит время на построение, но может появиться шум.

Num Indirect Lighting Bounces

Num Indirect Lighting Bounces фиксирует, сколько раз фотон может отразиться от поверхности на сцене до уничтожения. Значение 10 обеспечивает очень хороший результат, позволяя фотонам проникать в самые темные области вашей сцены. Повысьте этот параметр, если у вас есть пятна в темных областях.

Indirect Lighting Quality

Параметр Indirect Lighting Quality оказывает наибольшее влияние на время построения и общее качество. Это увеличивает количество семплирований, сделанных для сглаживания GI solution. Очень высокое значение обеспечивает очень плавный GI. Оставьте этот параметр в диапазоне от 1 до 4; выше только при шумном GI.

Indirect Lighting Smoothness

Используйте параметр Indirect Lighting Smoothness, чтобы контролировать, насколько мягким или резким является ваше непрямое освещение GI. Маленькие числа делают более резкими, но шумными непрямые тени, тогда как высокие значения дают более мягкие тени. Немного уменьшите этот параметр, если хотите получить эту резкость, но не сильно ниже 0,75, или придется повышать общее качество вашего решения для удаления шумных артефактов.

Compress Lightmaps

Наиболее важным параметром для архитектурной визуализации является Compress Lightmaps, который отключает сжатие Texture для Lightmass Textures. Как видно на рисунке 8.8, сжатие может привести к соединению и блокировке артефактов. Это применяется в большинстве игр, которые не полагаются на чистое освещение, но не для архитектурных визуализаций, где

точность освещения и поверхности имеют наивысший приоритет.

Вам не следует трогать другие настройки, пока не отключите этот параметр. Шум, который он вносит, сделает ваше освещение непригодным для архитектурных визуализаций, и никакие доводки не смогут устранить это.

Использование несжатых Lightmaps сказывается на расходе памяти. Несжатые Lightmaps занимают значительно больше дискового пространства, нежели сжатые Lightmaps. Они также используют больше памяти видео при запуске в игре.

Если ваш Level очень большой (стадион или целое здание) или вы ориентируетесь на нижний сегмент аппаратных средств (ноутбуки, мобильные устройства и т. д.), вы можете использовать параметр Compressed Lightmaps, чтобы позволить этим Levels загружаться в RAM.

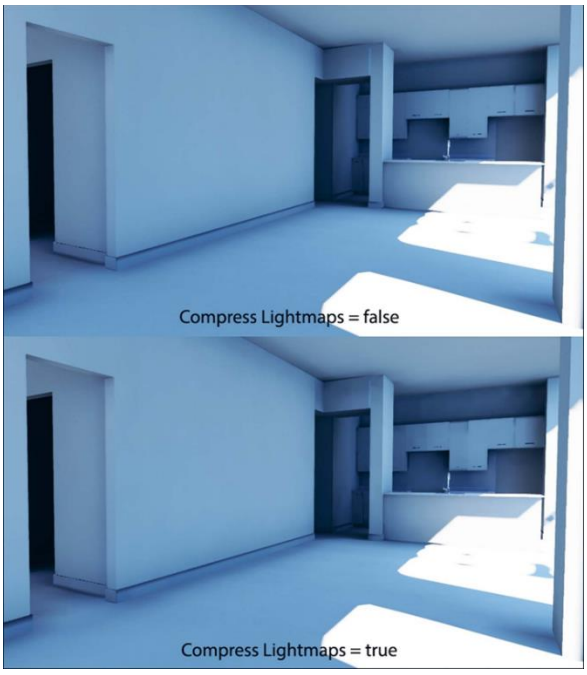

Рисунок 8.8 Сравнение Compress Lightmaps (with increased contrast to enhance artifacts), показывающее артефакты на шкафах, дверных рамах и гладких поверхностях, таких как стена слева

8.3 Настройка Lightmap UV Density

Придание вашим Lightmaps достаточного высокого разрешения важно для получения ровного, детализированного освещения. Однако слишком высокое разрешение может испортить вашу сцену из-за длительного построения и огромного количества используемой памяти.

Упражнение 11. Lightmap в UE4.

1. Включите визуализацию Lightmap Density, выбрав View Mode > Optimization Viewmodes > Lightmap Density (рисунок 8.9).

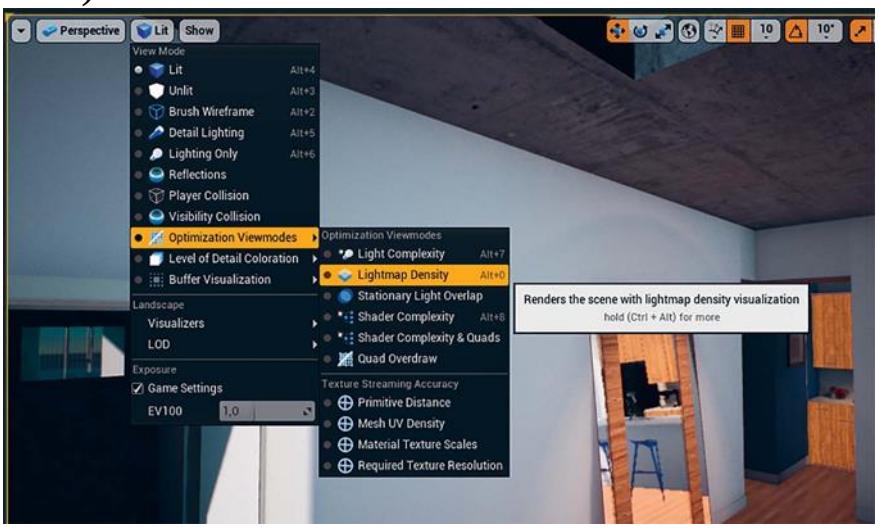

Рисунок 8.9 Переключение режима визуализации на Lightmap Density

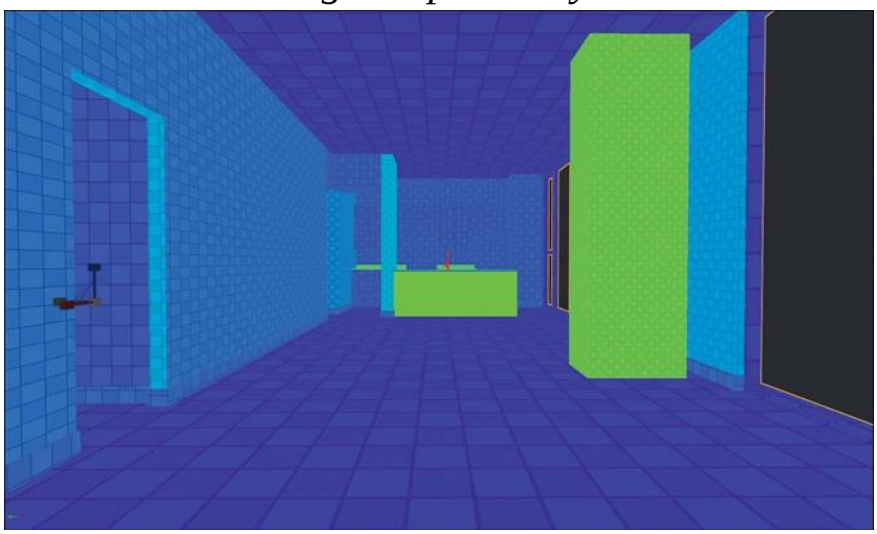

Рисунок 8.10 Визуализация Lightmap Density, показывающая слишком низкое разрешение, которое не

позволяет получить очень детализированные тени и освещение

Когда вы впервые попадете в этот режим, вы увидите что-то, как рисунке 8.10, — множество сеток разных размеров и цветов без информации об освещении.

То, на что вы смотрите, является визуализацией пикселей Lightmap для каждого объекта на сцене. Это поможет вам быстро и интерактивно настроить разрешение.

Синий означает, что разрешение Lightmap слишком низкое, а красный — слишком высокое. Вам нужно стремиться к зеленому или оранжевому на всех ваших поверхностях для получения лучшего качества с приемлемым временем построения.

2. После их настройки Lightmaps должен стать более равномерным по плотности и выше по разрешению, как показано на рисунке 8.11.

Вам не нужно устанавливать разрешение, кратное двойке (64, 128 и т. д.), но не нужно отклоняться слишком сильно. Разрешение более 1024 поддерживаются, но должны быть использованы с осторожностью. Чем выше разрешение, тем дольше время построения и тем больше памяти Level занимает на диске. Лучше всего держать его как можно ниже без нежелательных артефактов.

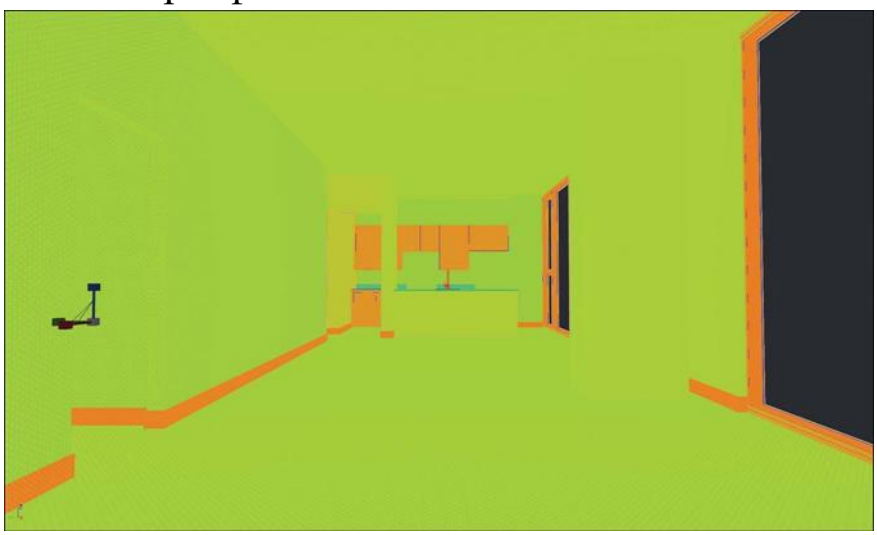

Рисунок 8.11 Режим визуализации с лучшими настройками Lightmap Density visualization

Настройка разрешения Lightmap для Props отличается, поэтому вы настраиваете ее в Asset Level. Это гарантирует, что для каждого реквизита на уровне будет правильное разрешение, так как сделанные вами изменения для исходного Asset отразятся на всех сценах объектов, которые ссылаются на эти Assets.

Имейте в виду, что любые изменения настроек Static Lighting для Mesh Asset нарушат освещение на каждом Level, на который ссылается Asset.

Также вы можете настроить плотность Props для каждого Object basis на вашем уровне с помощью свойства Overridden Light Map Res в Details Panel. Иногда задаются реквизитам (особенно если они содержат множество деталей или органических форм) более высокую плотность, нежели Architecture Meshes, как показано на рисунке 8.12, что делает их красноватыми.

Будьте осторожны; повторяющиеся реквизиты могут быстро генерировать много Texture-данных в высоком разрешении.

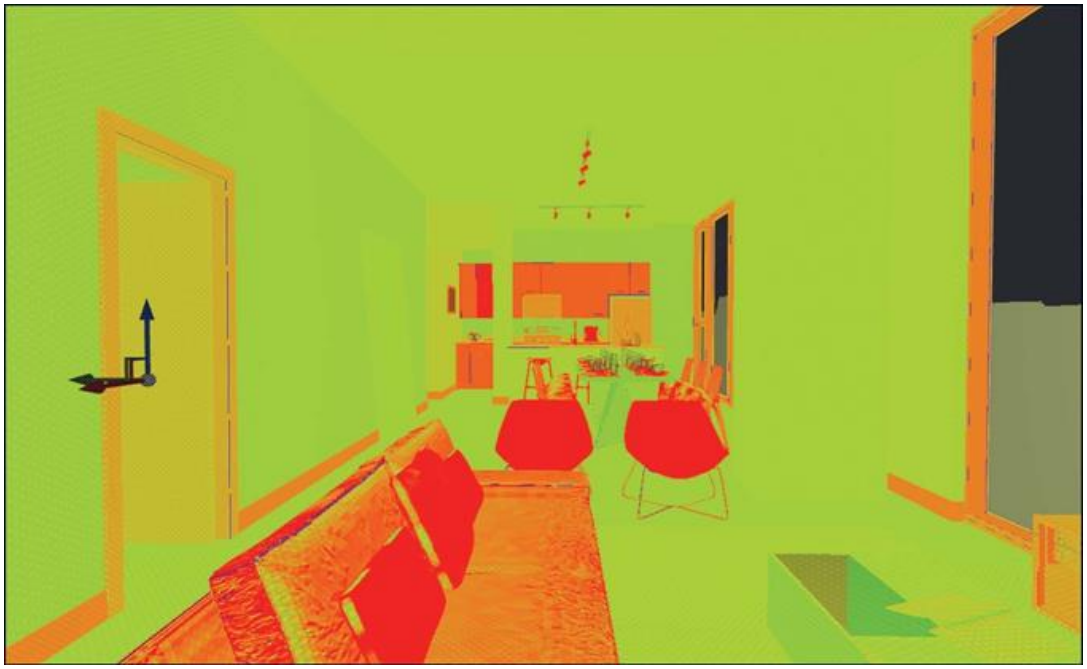

Рисунок 8.12 Конечная плотность Lightmap, показывающая более высокую плотность для органических Props, таких как диван или стулья

Билд и сохранение

Теперь самое время заново построить ваше освещение. Вы сразу заметите, что с увеличением плотности ваших Lightmaps, время построения также значительно увеличилось. Ваше общее качество также должно было значительно вырасти с более детализированными тенями и непрямым светом.

Сохраните вашу работу.

8.4 Размещение интерьерного света

Теперь, когда у вас есть Солнце и небо на сцене, вы можете начать размещать источники света внутри, чтобы помочь осветить некоторые темные уголки.

Spot Lights

Точечные источники света и прожектора имеют те же настройки, что и свет в рендерах с трассировкой лучей, и вы должны чувствовать себя как дома (рисунок 8.13). Есть только несколько различий, о которых следует знать.

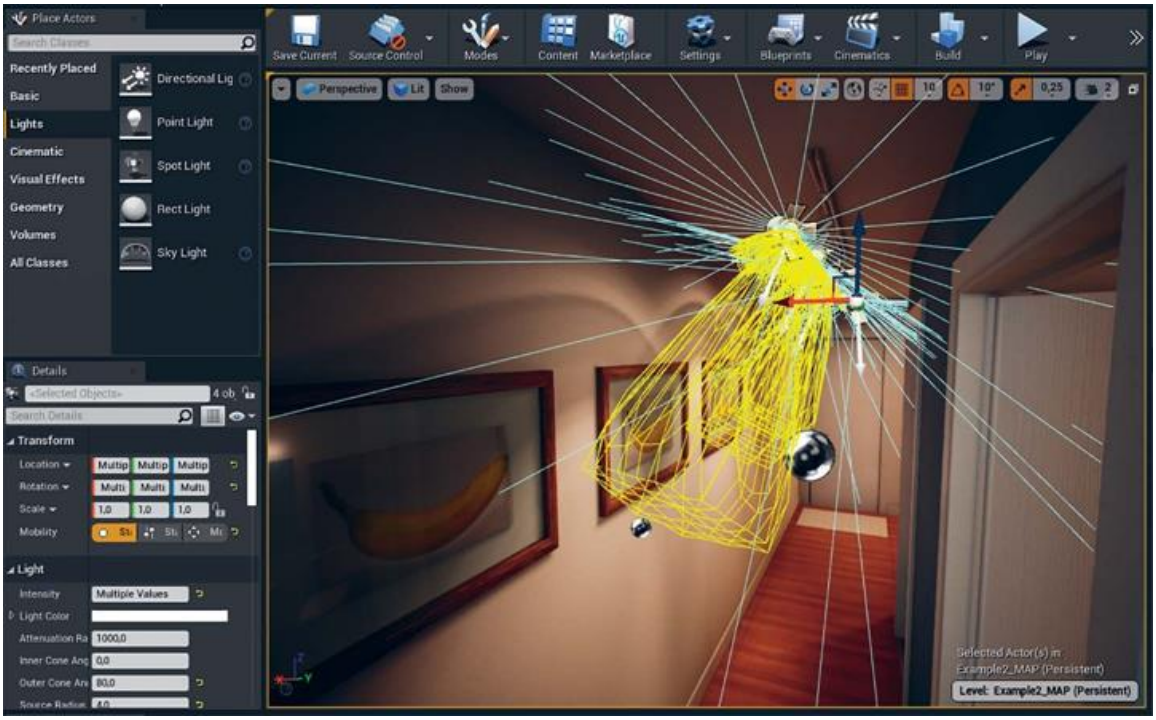

Рисунок 8.13 Размещение Spot Light Actors

Mobility

Можно использовать Static Lights почти исключительно для внутреннего освещения. Хотя стационарные и

динамические источники света дают некоторые преимущества, они добавляют сложность и значительную стоимость производительности вашим сценам.

Intensity

Постарайтесь помнить, что внутреннее освещение должно быть гораздо менее ярким, чем солнечный свет. Даже самый яркий внутренний свет, кажется, не дает много света в залитой солнцем комнате.

Получение правильной яркости освещения — это искусство. Это зависит от количества ИС в области, яркости солнечного света и даже цвета материалов, от которых он отражается. Яркость имеет критически важное влияние на окончательный вид вашего освещения.

Вам нужно повторять настройку освещения, чтобы найти подходящий баланс для вашей сцены и вкусов.

Temperature

В этом примере использовалась настройку температуры для своего света. Это дает реалистичный цвет, основанный на яркости света. Использовать более теплый тон для внутреннего освещения, который позволяет предметам быть более контрастными по отношению к Солнцу и небу.

Attenuation

Ограничение расстояния, на которое может перемещаться прямое освещение, используется как для производительности, так и для художественных целей. Ограничение ослабления света ограничивает количество мешей, на которые он влияет в мире. Это имеет решающее значение для производительности, особенно для динамически затененных огней. В Lightmass ограниченный радиус затухания повлияет на меньшее количество мешей, что ускорит построение освещения; но поскольку данные о свете и тени запекаются в текстуры, нет никакой выгоды для этого в реальном времени.

IES Files

UE4 поддерживает световые профили IES (Illuminating Engineering Society) для точечных светильников и прожекторов.

Вы можете импортировать файлы IES и применять их к лампам, чтобы получить интересные паттерны освещения (как показано на рисунке 8.13). Вы можете найти профили IES по всему интернету от реальных производителей и поставщиков освещения или даже из Unreal marketplace.

Auto Exposure

Временное отключение Auto Exposure может быть большой помощью при настройке освещения на вашей сцене. Автоэкспозиция по умолчанию очень агрессивна и может легко стать очень яркой или очень темной.

Блокировка экспозиции с помощью меню View Mode на Viewport гарантирует, что яркость освещения не будет регулироваться при переходе от светлых областей к темным, и позволяет легче сбалансировать освещение.

8.5 Размещение Light Portals

Light Portals — это специальные акторы, которые помогают Lightmass фокусировать фотоны к отверстиям, в которые поступает много света. Это улучшает время сборки и качество освещения.

Разместите этих акторов на ваших окнах и других отверстиях и свободно отрегулируйте коробку по размеру. Вам не нужно быть ужасно требовательным; это просто помощник для фотонов, и он не нуждается в точности.

8.6 Использование Reflection Probes

Отражения являются неотъемлемой частью PBR. UE4 использует акторов захвата отражения для захвата статических кубических карт сцены и автоматически применяет их к материалам, находящимся под их влиянием.

Хотя вы можете уйти, не размещая их вокруг своего уровня, вы достигнете гораздо более высокого качества с ними.

Два типа Reflection probes — сфера и куб. Проще говоря, для областей квадратной формы используйте Box Reflection Capture Actor. Для всех остальных областей используйте Sphere.

Как видно из рисунка 8.14, для точной настройки отражений на сцене полезно смешение обоих видов.

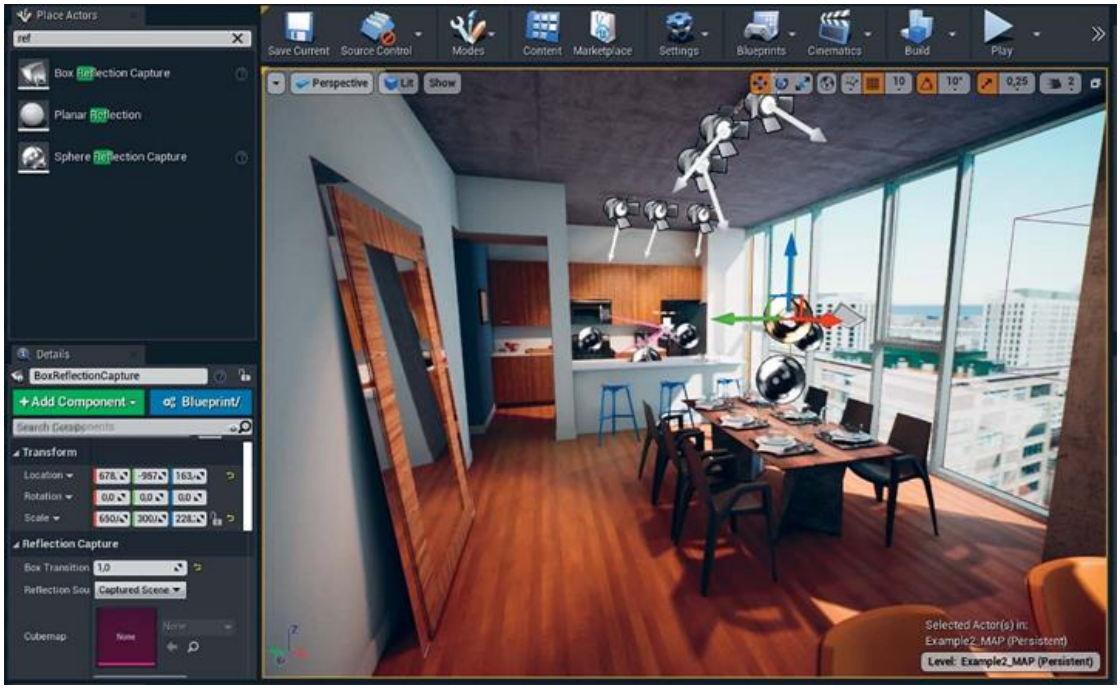

Рисунок 8.14 Reflection Probes

8.7 Размеры

Sphere Reflection Actors используют радиус влияния, чтобы определить, на какие пиксели они будут влиять. Как правило, Reflection Actors меньшего радиуса имеют приоритет над Reflection Actors большего радиуса. Это позволяет вам «гнездить» меньшие Reflection Actors внутри больших, позволяя добавлять детальные отражения в области, которые могут в этом нуждаться.

Box Reflection Capture Actors выставляют значение 3D-масштаба. Вы должны отрегулировать масштаб коробки до тех пор, пока углы коробки, показанной в окне просмотра, не будут максимально приближены к углам вашей комнаты.

Производительность

В дополнение к накладным расходам памяти, захваченной Cubemaps, Reflection Capture Actors автоматически применяют свои отражения к любым пикселям, которые визуализируются внутри них. По этой причине перекрывающиеся захваты могут повлиять на производительность. Вы должны избегать слишком

большого количества накладывающихся друг на друга элементов.

Также стоит отметить, что Box Reflection Capture Actors оказывают более серезное влияние на производительность, чем Sphere Reflection Capture Actors.

Post-Process Volume

Последняя часть головоломки освещения состоит в том, чтобы применить эффекты постобработки, такие как виньетка, блум, размытие движения и глубина резкости к изображению, чтобы придать ему кинематографичный, реалистичный вид.

При создании нового уровня в UE4 настройки постобработки по умолчанию применяются к сцене автоматически. Эти настройки — хорошее начало, но вы почти всегда будете хотеть настроить их в каждой из ваших сцен, чтобы соответствовать ее уникальному освещению и достичь того внешнего вида, который вам нужен.

Чтобы получить доступ к этим настройкам, вам нужно добавить специальный класс актора на вашу сцену: Post-Process Volume.

Volume Actors

Volume — это особый тип класса акторов в UE4, который может определить, находятся ли другие акторы в его пределах. Post-Process Volumes используют эту способность смешиваться между различными постпроцессными настройками, когда камера пользователя входит и выходит из каждого тома.

Вы можете настроить Volumes, установив их настройки Brush. Вы можете определить форму объема, а также его размер. Вы также можете масштабировать, вращать и перемещать их, как и любой другой актор.

Размещение Post-Process Volume

Вы найдете список различных типов классов объемов, доступных на панели Place Actors в разделе Volumes (рисунок 8.15). Перетащите Post-Process Volume в Viewport. Вы можете разместить его в любом месте сцены — просто убедитесь, что он легко выбирается.

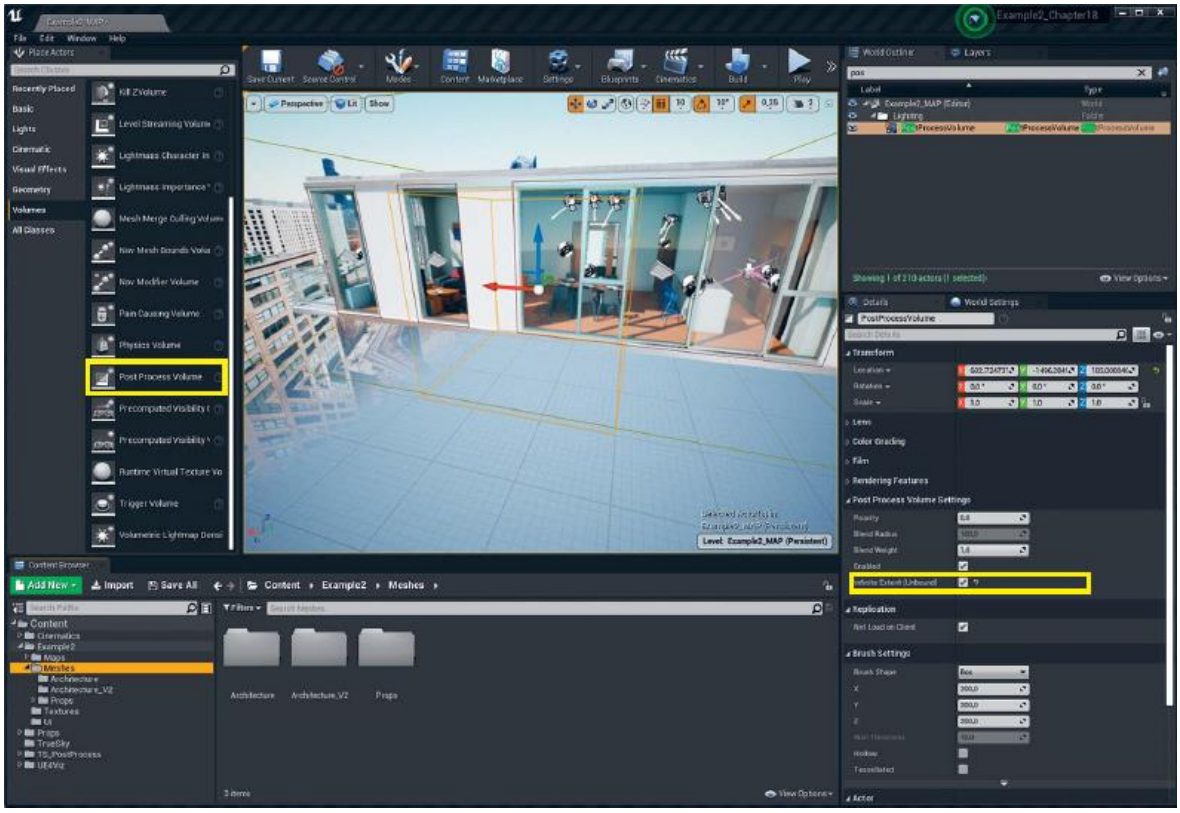

Рисунок 8.15 Размещение Post-Process Volume на Level

Настройка Post-Process Volume

Post-Process Volume имеет множество настроек и позволяет значительно улучшить ваше изображение. В то время как многие настройки — такие как настройка цвета и эффекты вроде цветения — почти полностью зависят от ваших личных вкусов и стилей, другие оказывают прямое влияние на производительность и качество изображения.

Чтобы получить доступ к настройкам, просто выберите Post-Process Volume на уровне и посмотрите на панель Details.

Ниже приведены настройки, которые вы будете настраивать чаще всего, а также те, которые были скорректированы для этого примера.

Unbound

Вы можете настроить свой Post-Process Volume, чтобы пропустить проверку границ и применить его настройки по всему уровню, включив опцию Unbound. Это гораздо проще, чем установить объем, чтобы охватить весь уровень.

Priority

Вы можете иметь Post-Process Volumes, которые перекрывают друг друга. Чтобы определить, какие настройки будут использоваться, примените опцию «приоритет».

Объемы с более высоким приоритетом будут переопределять объемы с более низким приоритетом.

Важно отметить, что будут применены только те свойства, которые переопределены вручную (установив флажок слева от свойства на панели Details для перекрывающегося объема). Это отличный способ настроить только одно свойство, не проверяя соответствие всех остальных параметров.

Blend Radius

Это радиус мирового пространства вокруг объема, который интерполируется между настройками двух объемов. По мере того как камера пользователя перемещается в Volume и из него, настройки будут плавно переходить от одной к другой. У объемов, настроенных как несвязанные, этот параметр неактивен.

White Balance

В UE4 есть много настроек для цветокоррекции, но вам, вероятно, следует начать с баланса белого. Это значение по умолчанию равно 6500, что является очень чистым, сине-белым цветом, который не очень распространен в реальном мире. Это может придать вашим сценам слишком холодный тон и затруднить достижение правильного баланса между теплым светом и прохладными тенями. Вы можете установить значение примерно между 5000 и 6000 для более теплого тона на вашей сцене и выше — для более холодного взгляда.

Saturation and Contrast

Настройки Saturation, Contrast, Crush Highlights и Crush Shadow работают вместе, чтобы помочь контролировать баланс изображения. Вы, вероятно, знакомы с контрастом и насыщенностью, но, возможно, не с настройками подавления. Они просто обрезают черно-белые точки, давая больший видимый контраст.

Настройки по умолчанию часто слишком контрастны, что делает затененные области очень темными. Это очень кинематографичный взгляд, но результат может быть слишком темным для архитектурной визуализации, при чем совсем не там, где предпочтительнее иметь очень темные области.

Эти настройки очень чувствительны, поэтому небольшие корректировки имеют большое значение.

Vignette, Noise и Fringe

Используйте эти эффекты для имитации эффектов объектива камеры.

Виньетка добавляет темный градиент к краям вашего изображения, что может позволить вам увеличить общую яркость изображения — единственные полностью яркие пиксели находятся в самом центре экрана.

Бахрома имитирует эффект хроматической аберрации света, проходящего через объектив камеры, разделяя цвета ближе к краям изображения.

Зернистость добавляет анимированный шум к изображению. Интенсивность зерен управляет непрозрачностью наложения шумовой текстуры, в то время как дрожание зерен управляет тем, насколько зерно смещает изображение. Используйте их осторожно, так как они могут быстро стать сильно заметными.

Color Grading (LUT)

Система Color Grading в UE4 использует специальную текстуру, называемую Color Lookup Table (LUT), для изменения цвета сцены. LUT генерируется путем применения цветовой градации к базовому изображению в приложении для композиции или редактирования изображений.

UE4 считывает разницу между базовым изображением и модифицированным LUT и применяет эту дельту к сцене. Это отличный способ перенести существующий конвейер цветокоррекции в UE4. Вы можете найти более подробную информацию о LUTs и некоторых файлах LUT для загрузки на

Bloom и Lens Flares

Некоторые из наиболее распространенных эффектов постобработки как в играх, так и в традиционно визуализируемом контенте, цветение и вспышки линз помогают изображать чрезмерно яркие области, имитируя объективы камер и человеческий глаз.

Настройки цветения и бликов объектива по умолчанию в UE4 немного агрессивны и могут уменьшить контрастность и четкость сцен, если они используются слишком часто. Уменьшение интенсивности или увеличение порога являются хорошими способами уменьшения общего количества цветения и бликов объектива в сцене.

Вы также можете настроить размер эффектов. Большие размеры будут иметь более значительное влияние на производительность.

Часто полностью отключаются вспышки объектива в сценах или выключаются их до тех пор, пока они происходят только с очень яркими пикселями.

Auto Exposure

Поскольку UE4 визуализирует сцены с использованием среды HDR-освещения, вы можете иметь очень разные уровни яркости освещения от одной области вашей карты до другой. По этой причине UE4 имеет сложную систему автоматической экспозиции.

Эта система поможет создать динамическое, привлекательное взаимодействие освещения, когда пользователь перемещается из одной области в другую и видит, как экспозиция настраивается в ответ так же, как человеческий глаз или камера с автоматической экспозицией. Это также может сделать настройку вашего освещения довольно сложной задачей.

Рекомендую настроить освещение с отключенной автоматической экспозицией, а затем включить его по мере необходимости для сцены. Этот эффект можно отключить либо с помощью настроек экспозиции в раскрывающемся списке View Mode на Viewport, либо установив параметры Min and Max

Brightness в Post-Process Volume равными 1,0. Затем вы можете использовать Exposure Bias, чтобы вручную установить экспозицию камеры.

После настройки освещения начните играть с Min and Max Brightness, чтобы настроить экспозицию камеры. Чтобы разрешить камере переэкспонировать, осветляя темные области, установите минимальную яркость на значение ниже 1,0. Чтобы уменьшить экспозицию камеры при ярком освещении, увеличьте максимальную яркость выше 1,0.

Ambient Occlusion

Хотя в данном примере мы не используем Screen Space Ambient Occlusion (SSAO), это очень важный эффект, который широко используется во всех видах визуализаций и игр. Если вы создаете сцены с динамическим освещением или большим количеством динамических акторов, вы, вероятно, захотите включить это, потому что это значительно увеличивает глубину сцены и качество освещения.

Глобальное освещение

Это управляет интенсивностью и цветом Lightmaps, генерируемых Lightmass. Вы можете использовать это, чтобы быстро изменить свое испеченное освещение. Это не управляет никаким видом GI в реальном времени в UE4.

Глубина резкости (Depth of Field, DOF)

UE4 предоставляет несколько методов для создания эффектов глубины резкости. Для визуализации круг DOF легко является лучшим. Это физически точный эффект, который имитирует фактические характеристики размытия диафрагмы объектива и создает очень тонкий, реалистичный эффект. Он также довольно эффективен по сравнению с другими эффектами. Однако, как и многие эффекты постобработки, он может быть ресурсоемким при более высоких разрешениях.

Чтобы увеличить эффект размытия, уменьшите значение параметра Aperture F-Stop. Более низкие настройки приведут к большему размытию изображения (рисунок 8.16). Важно

отметить, что это реалистичный эффект, поэтому вы можете не заметить его, пока не подойдете очень близко к объекту.

Для внутренней сцены вам нужно будет убедиться, что ваше фокальное расстояние установлено примерно на 300–500 (от 3 до 5 метров), а Aperture F-Stop довольно велика на уровне 4–8.

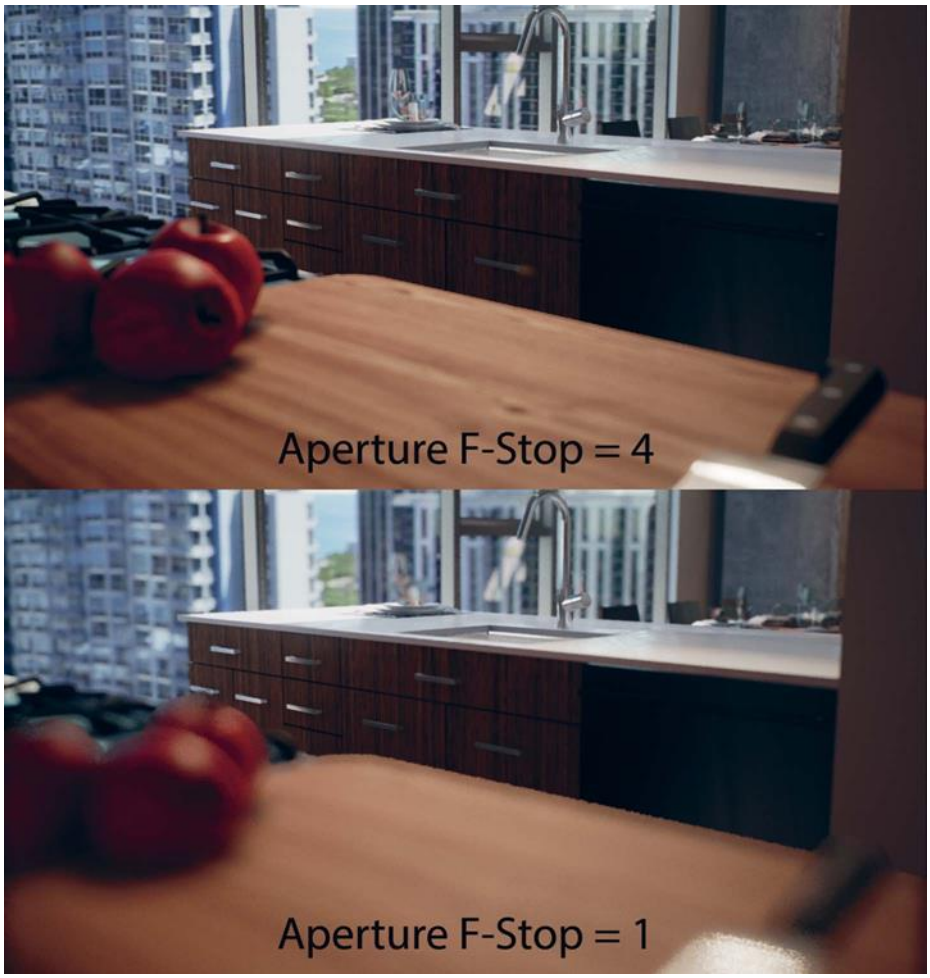

Рисунок 8.16 Circle Depth сравнение Field F-Stop

Размытие изображения при повороте камеры (Motion Blur)

UE4 использует высококачественную систему размытия движения, которая генерирует карту скорости каждого кадра и использует ее для соответствующего размытия сцены. Настройки по умолчанию, как правило, очень хороши для большинства сцен; однако, если вы работаете с более высокой частотой кадров или хотите более чистую презентацию, то

можете отключить эту функцию или уменьшить ее значение, уменьшив параметр Max.

Отражения в экранном пространстве (Screen Space Reflections)

Screen Space Reflections обеспечивают детальные динамические отражения, основанные на визуализированном изображении. Они необходимы для достижения самого высокого качества, но вы захотите увеличить Quality и Max Roughness. Установите

Quality на 100 и Max Roughness между 0,6 и 1,0. Более высокие значения дороже визуализировать, но они могут выглядеть более точными. Устанавливайте значения как можно ниже, сохраняя при этом внешний вид вашей сцены.

Anti-Aliasing

UE4 предлагает несколько методов сглаживания (AA). Для большинства визуализаций Temporal AA (TAA) система в UE4 дает превосходные результаты и имеет очень мало накладных расходов на производительность.

В следующем примере сцены показан результат включения параметров окружающей окклюзии, виньетки, зернистости и глубины резкости. Каждый из них улучшает внешний вид изображения, добавляя несовершенства и другие эффекты, которые можно увидеть на фотографии.

Как видите, настройки в постобработке пространства оказывают серьезное влияние на внешний вид сцены. Даже тонкие настройки, используемые здесь, значительно изменяют внешний вид сцены (рисунки 8.17 и 8.18).

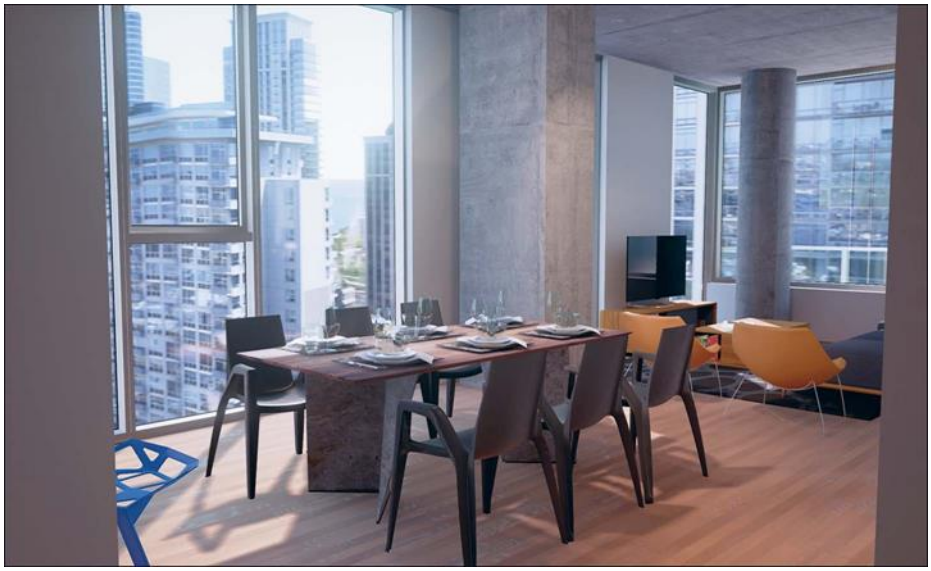

Рисунок 8.17 Сцена с настройками по умолчанию у постобработки

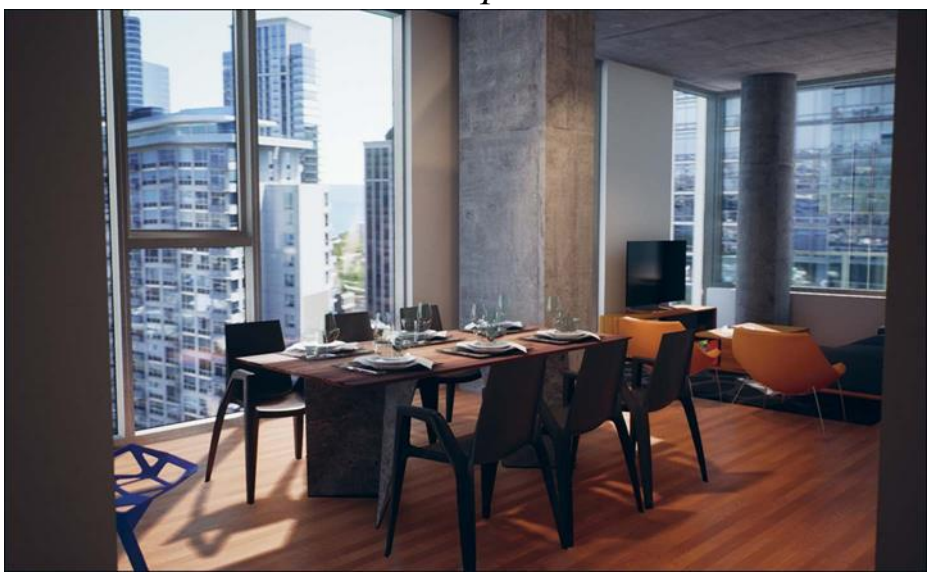

Рисунок 8.18 Сцена с настройками постобработки, показывающими заметное изменение контраста, баланса белого и насыщенности

Заключение

Освещение в UE4 имеет много компонентов, которые работают вместе, чтобы создать единое целое. Хотя эти понятия были представлены в том порядке, в котором вы могли бы с ними столкнуться, по мере изучения и повторения на вашей сцене вы станете перемещаться между их настройками, потому что настройка одного влияет на другое.

Подобное взаимодействие между светом, цветом и материалами знакомо художникам визуализации, и именно в этой точке технология расходится с художественностью.

Благодаря практике вы научитесь понимать свой инструмент и начнете легко создавать теплое освещение в UE4.

9 АРХИТЕКТУРНЫЕ МАТЕРИАЛЫ

Система материалов в UE4 как хорошая видеоигра: проста для изучения, но занимает целую жизнь, если ты надумал стать мастером.

Создание хороших материалов UE4 для производства — это не только о качествах художника, но и о создании повторно используемых материалов, получении лучшей из возможных производительности и изучении новых техник, которые не используются при рендеринге с трассировкой лучей, таких как Parallax Occlusion Mapping.

Материалы и освещение работают вместе для создания богатого и реалистичного окружения для пользователя. Вы можете достичь потрясающих результатов с помощью простых материалов. Теперь, когда запущено освещение, вы можете добавить несколько материалов на сцену и оживить их с помощью цвета, отражения, деталями и вариациями поверхности.

Получить хорошо выглядящий материал в UE4 просто, потому что использовать PBR (physically based rendering) легко. Определите параметры Base Color, Roughness,

Metallic, Normal, а движок сделает сложную работу и породит физически корректный материал для потрясающих поверхностей ваших сцен.

Что такое мастер-материал?

Как уже обсуждалось, вы можете создать уникальный материал для каждой поверхности, как и в обычном 3D-редакторе. Это может занять очень много времени, особенно когда вы начинаете добавлять в материалы много функционала.

Вместо этого вы будете использовать специальные параметры, Material Parameters, для создания одного материала (Material) и затем нескольких экземпляров материала (Material Instances), назначая текстуры и переопределяя свойства для создания практически каждого материала, используемого на сцене.

Этот единственный материал часто называют Master Material. Master Material не является конкретным ассетом в Unreal Engine. Скорее это идея. Любой материал может быть Master Material, если вы создадите его с параметрами Material Parameters. Material Parameters предоставляют переменные для Material Instances, которые могут быть изменены на лету, как в редакторе, так и в процессе работы с помощью Blueprints.

Благодаря простоте основанного на физике рендеринга сеть ваших материалов не будет слишком сложной. Используя только Color, Normal, Metallic и Roughness Textures вместе со случайной картой высот, вы сможете определить практически любую поверхность.

Вы даже сможете создать обычно сложные поверхности, такие как металл, стекло, и архитектурные настенные покрытия с минимальными усилиями или без необходимости создавать сложные пользовательские материалы.

9.1 Обзор работы материалов

На рисунке 9.1 представлен Master Material. На первый взгляд он может показаться сложным, но на самом деле все просто.

Материалы похожи на Blueprints и используют основанный на узлах граф (Material Graph), чтобы помочь визуализировать то, что на самом деле является понятиями из программирования. Узлы, соединенные с другими узлами и потоками данных слева направо, в конце сходятся в одном из различных атрибутов материала, таких как Base Color или Roughness.

Вы возьмете концепции, описанные в главе 5, и расширите их, добавив немного новых типов узлов, которые обеспечат большую гибкость для экземпляров материалов или предоставят

продвинутые функции рендеринга, такие как Parallax Occlusion Mapping.

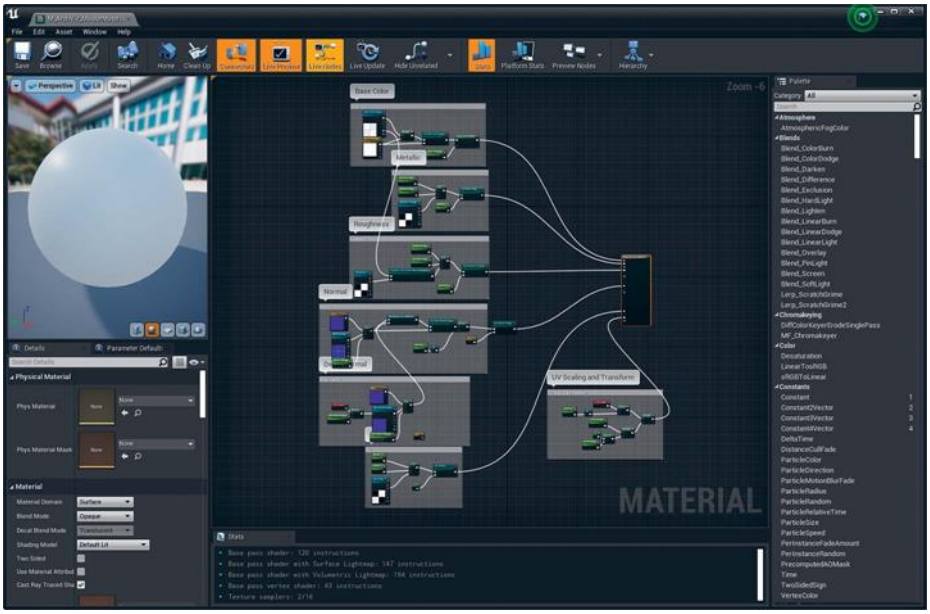

Рисунок 9.1 Завершенный M_ArchViz_MasterMaterial shader graph, представляющий один Material, который может быть использован для почти каждого Material на сцене через Material Instances

Узлы параметров

Существуют особые виды вершин, Material Expression Nodes, которые вы можете разместить в Material Graph, называемые Parameters. Эти узлы позволяют вам раскрывать некоторые аспекты вашего материала, который может быть динамически изменен с помощью Material Instance assets или в процессе работы с помощью Blueprints.

Параметры, созданные в материале, отображаются как редактируемые Properties в Ma- terial Instances, полученные из этого материала. В Material Instance Editor (рисунок 9.2), вам нужно установить галочку слева от свойства, которые вы хотите переопределить до изменения значения. Для возврата значения по умолчанию нажмите на маленькую желтую стрелку рядом с измененным свойством. Вы также можете убрать галочку для отмены переопределения.

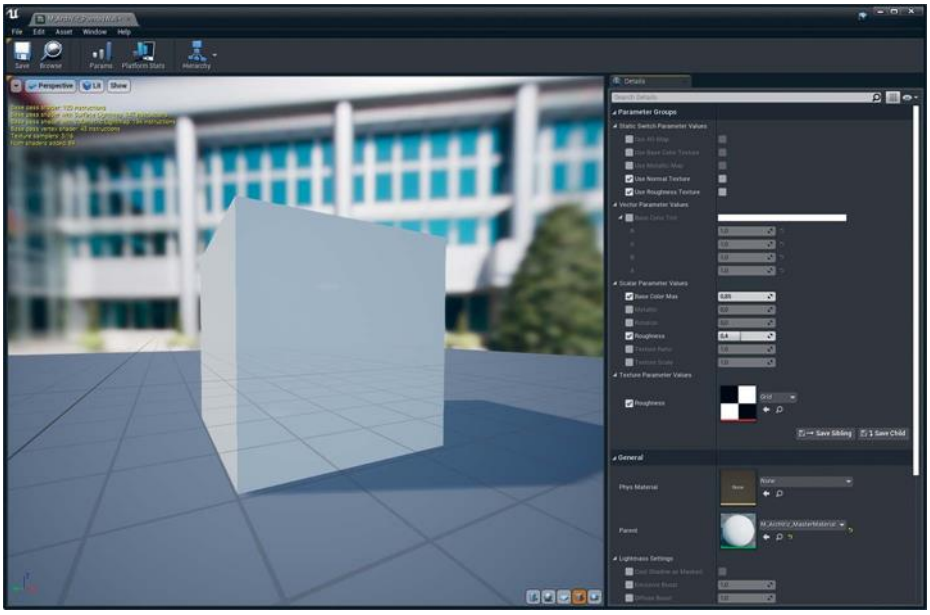

Рисунок 9.2 Раскрашенный Wall Material Instance, основанный на M_ArchViz_MasterMaterial

Создание мастер-материала

Как показано на рисунке 9.1, нужно очень многое настроить для создания хорошего Master Material. Каждый Material input (Base Color, Metallic, Roughness, and Normal) имеет ряд узлов, соединенных с ним, что позволяет вам создавать бесконечные вариации ваших материалов с Material Instances.

Base Color

На рисунке 9.3 показан Texture Parameter 2D, называемый Base Color Texture, который умножается на Vector Parameter, называемый Base Color Tint. Materials — это Math, а Colors как RGB Vectors. Это значит, что вы можете выполнять все возможные математические операции для вектора цвета в текстурах, и это позволит вам делать корректировки на лету.

Adding Parameters

Для размещения узла Texture Parameter просто используйте палитру или меню, вызванное нажатием правой кнопкой мыши, и выберите Texture Parameter из списка. Затем назовите Parameter чем-нибудь значимым. Parameters могут содержать пробелы и знаки препинания в их названиях, делая их легко читаемыми, но это может затруднить доступ к этим переменным кода.

Узлы Parameter могут быть переименованы и изменены с помощью панели Details. Используя панель Details, вам также нужно определить Texture asset, который будет использоваться для этого узла по умолчанию. Вы можете либо нажать по миниатюре для вызова browser, либо перетащить Texture из Content Browser в свойства узла.

Разместите Vector Parameter таким же образом, снова присвоив ему осмысленное название. Vector Parameter можно отредактировать, изменив значения RGBA в панели Details, или двойным нажатием на образец цвета в узле. Это вызовет палитру цветов, которой гораздо проще пользоваться.

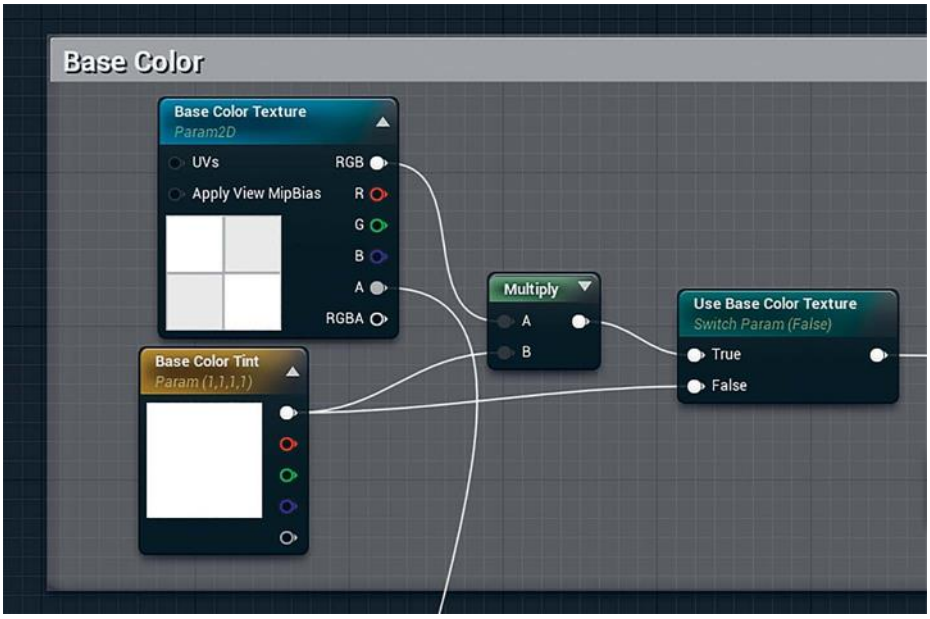

Рисунок 9.3 Узел Base Color

Кроме того, вы можете создать вершину Texture 2D, перетащив текстуру из Content Browser в граф материала. Сразу такая вершина не станет вершиной Parameter, но ее можно легко преобразовать в таковой двойной щелчок ПКМ на вершене Texture 2D и выбором команды Convert to Parameter в контекстном меню.

Multiplying Colors

Добавьте узел Multiply с помощью Palette, или меню, вызываемого двойным нажатием правой кнопки мыши, и подключите выход Base Color Texture к входу A и выход Base Color Tint к входу B.

Узел Multiply умножает каждый канал входа A на вход B и возвращает результат. Умножение вектора (выходной узел base Color Texture’s RGB) на другой вектор (выходной узел BaseColor Tint’s RGB) означает, что каждый из отдельных RGB-каналов умножается друг на друга (RedA x RedB, BlueA x BlueB, GreenA x GreenB). Вы также можете умножить один тип данных на другой, к примеру, Vector и Scalar (float), — в данном случае будет умножаться каждый канал Vector на значение Scalar.

Использование узла Multiply для цветов аналогично Multiply blend mode, с которым вы, возможно, сталкивались во многих приложениях. Черный (0,0,0), полностью окрашивает ваш Base Color в черный, так как все умноженное на 0 равно 0. Чистый белый (1,1,1) не изменяет входящее значение вообще. Конечно, вы можете выбрать цвет, тонируя пиксели. Вы также можете превысить 1 (или 0) в вашем Vector Parameter, действуя как регулировка яркости для Texture. Это иногда может привести к физически неточным результатам, поэтому будьте осторожны, когда переопределяете ваши входные данные этим способом.

Static Switch Parameters

Узел Static Switch Parameter отображает Boolean флажок в Material Instances. Если параметр установлен на true (в Material или Material Instances, от которого получает его), Material будет оценивать путь кода, соединенный с входом True (и с False в противном случае).

Изменение значения Static Switch также может обновить интерфейс Material Instance.

Parameters, которые не вызываются, такие как Base Color Texture, не будут показаны в Material Instance Editor, что приведет к уменьшению помех и избеганию показывания пользователю параметров, которые ничего не делают.

Соедините результат узла Multiply с входом True, а выход Base Color Tint Parameter — с входом False в Static Switch Parameter.

В данном случае, если Use Base Color Texture равен false, Material будет использовать Base Color Tint для определения Base Color, минуя узлы Texture и Multiply. Это экономит на чтении Texture и вычислении Shader, делая более производительный Material.

Metallic

Входной канал Metallic является одним из наименее используемых атрибутов и может быть установлен на 0 или 1 через Metallic Scalar Parameter (рисунок 9.4).

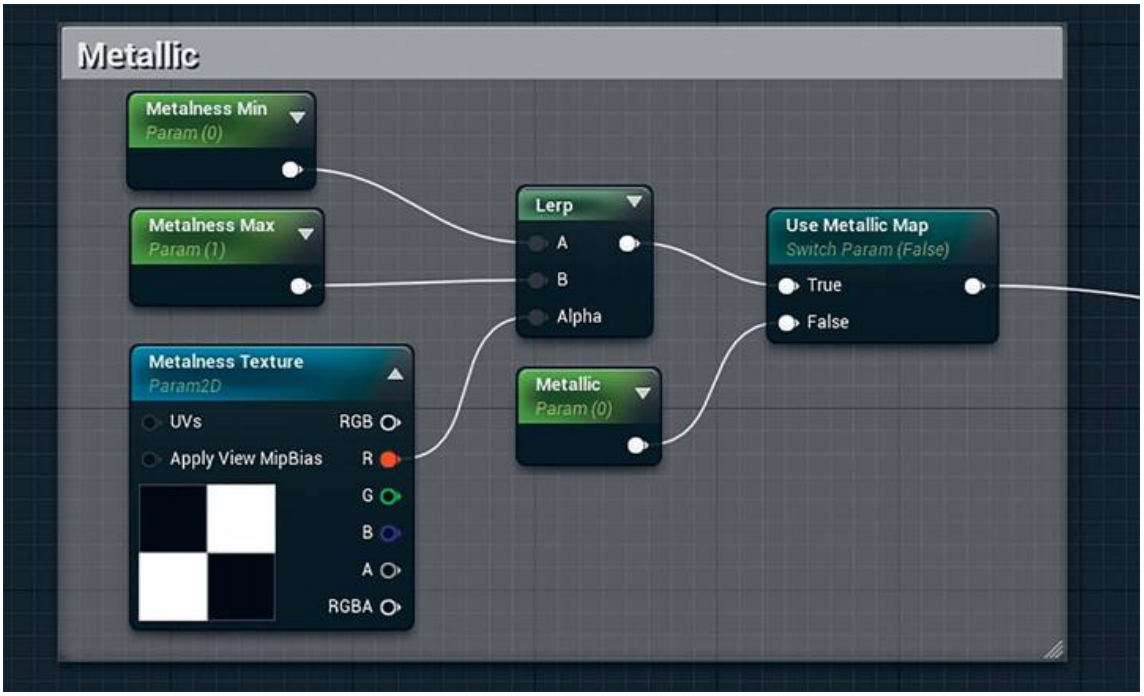

Рисунок 9.4 Узел Metallic

Однако иногда вы можете захотеть использовать Texture как маску для определения области модели, какие являются metallic, а какие нет; к примеру, текстура древесины с видимыми гвоздями или шурупами или материал, представляющий сколотую краску с металлической поверхностью под ней.

Для учета этого создайте узел Texture Parameter 2D под названием Metalness Texture.

Расположите узел Linear Interpolate (или Lerp, как его обычно называют), который позволит вам переназначать значения в другие значения, используя простые Min и Max Parameters для каждого. Alpha-вход действует как процентный

вес между входами A и B, значение 0.0 соответствует входу A и значение 1.0 соответствует входу B. Регулируя эти значения, вы можете легко интерактивно настроить поверхности ваших сцен.

Создайте два Scalar Parameters, назовите их Metalness Min и Metalness Max и подключите их к входам A и B узла Lerp. Установите Default Value для Metalness Max на 1.0 с помощью панели details. Настройка параметров Metalness Min и Metalness Max позволит вам легко изменять значение маски без изменения Texture.

Чтобы полностью отключить Texture, разместите другой Static Switch Parameter в Graph и назовите его. Если для этого Parameter установлено true, то выбирается красный канал Metalness Texture и затем изменяется с помощью узла Lerp. Поскольку для входа

Metallic требуется только градация серого или скаляр (0–1), только красный канал текстуры используется, как альфа узла15 Lerp.

Если Use Metallic Map установлен на false, он будет использовать Metallic Scalar, который установлен на 0,0 и подключен к выходу False.

Roughness

Канал Roughness, возможно, является самым важным каналом, не считая Base Color. Однако, как вы можете заметить на рисунке 9.5, он настроен так же, как и другие каналы, с Use Roughness Texture Static Switch Parameter переключаясь между Texture-based roughness и Roughness Scalar Parameter.

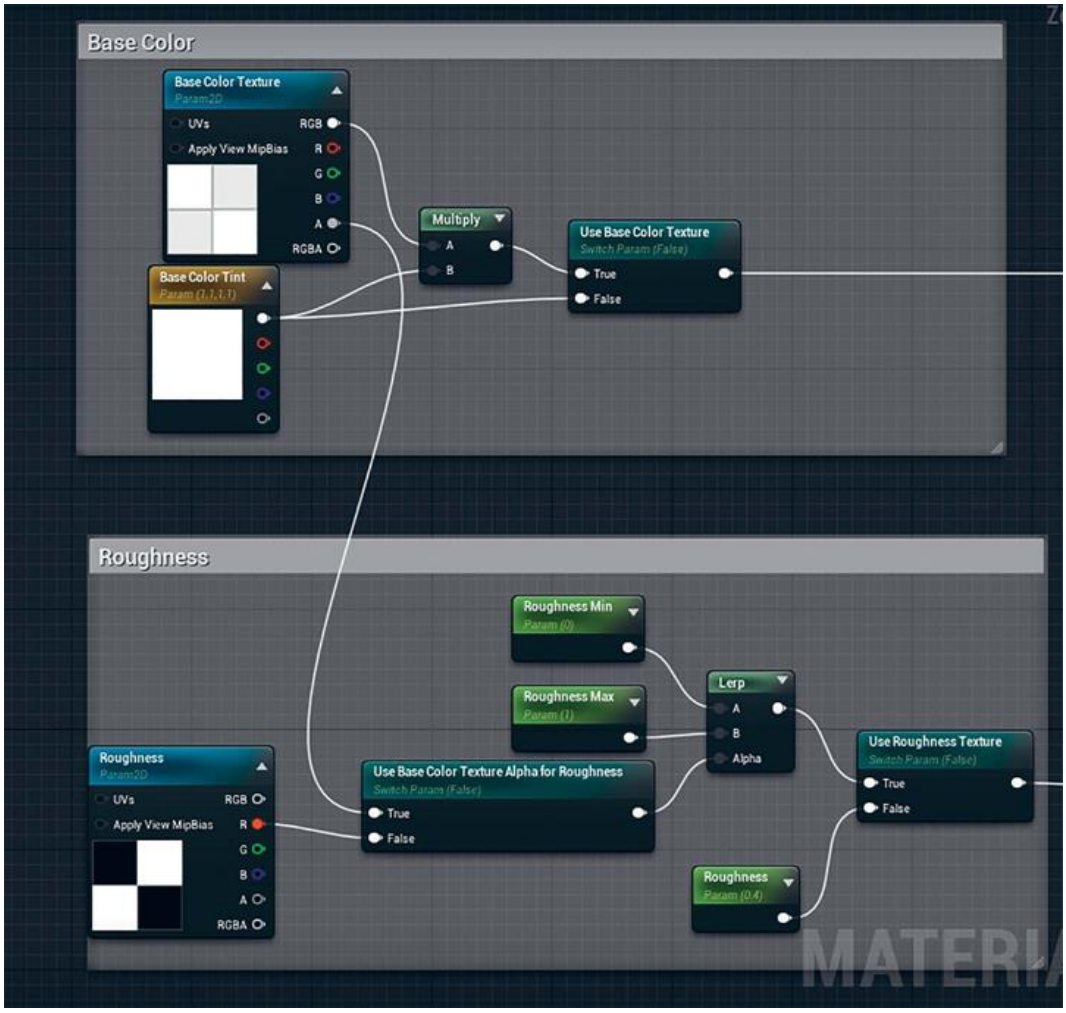

Рисунок 9.5 Узел графа Roughness также показывает Base Color, поэтому вы можете заметить, как Alpha из Base Color Texture может быть использована в качестве Roughness map

Использование только одного канала из Texture позволяет продвинутым пользователям применять каждый канал для хранения RGB-текстуры различных изображений в градациях серого.

Упражнение 12. Material Graph.

Создайте ваш Material Graph, как показано на рисунке 9.5. Обратите внимание на значение по умолчанию для каждого узла Parameter при подготовке построения вашей сети.

Use Base Color Texture Alpha for Roughness Static Switch Parameter позволяет пользовать Alpha канал Base Color Texture как маску Roughness или красный канал Roughness Texture

Parameter. Использование Alpha из Base Color Texture является обычным рабочим процессом, и многие Assets, доступные в Marketplace и в сообществе, используют этот метод для определения Roughness в Materials.

Другой Static Switch Parameter, называемый Use Roughness Texture, позволит вам переключаться между одним скалярным значением Roughness, который соединен с входом False, и путем на основе Texture, который подходит к входу True. Если Use Roughness Texture установлен на true, то данные текстуры вернутся к Use Base Color Texture Alpha для Roughness Parameter, управляемым с помощью узла Lerp и Roughness Min и Roughness Max Scalar Parameters.

Normal

Канал Normal, вероятно, наименее знаком большинству визуализирующих художников. Большинство 3D-приложений полагаются на карты рельефа и высот для определения поверхности. Работающие в реальном времени приложения, включая UE4, используют карту нормалей, потому что они быстрее вычисляются, чем карта рельефа, и могут определить кривизну поверхности, что делает их более качественными, чем карты рельефа.

Создайте узел Normal Texture Parameter, как и любой другой Texture Parameter. Однако вы должны установить Sampler Type на Normal в панели Details узла. Это делается автоматически, если вы назначили карту нормалей текстуре для свойства узла Texture.

Для контроля интенсивности карты нормалей используйте Lerp между значением Normal Texture, подключенным к входу A, и значением Constant Vector 0,0,1 (значение для не измененной нормали), подключенным к входу B (рисунок 9.6).

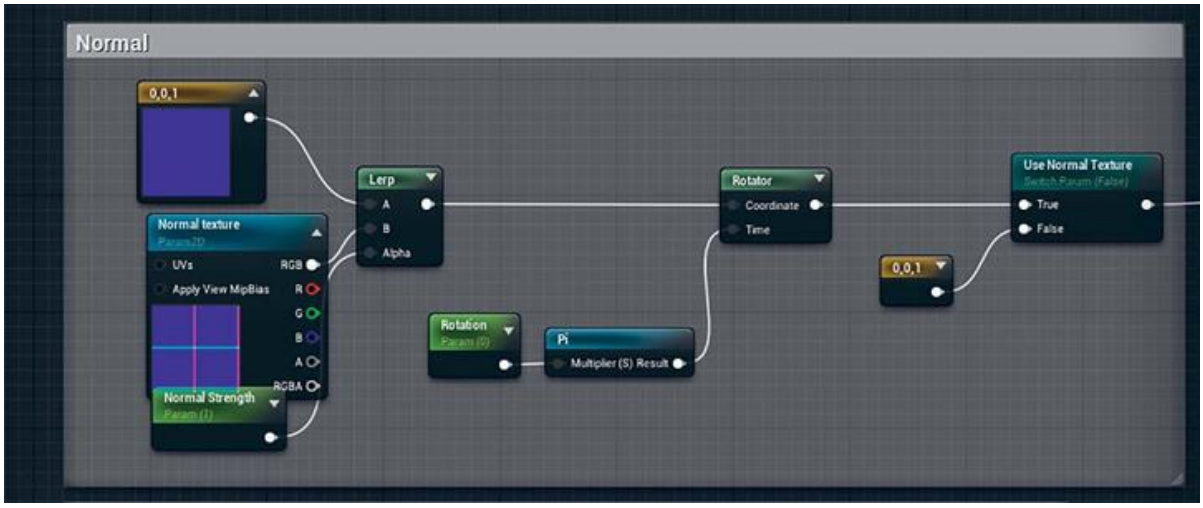

Рисунок 9.6 Узел Normal

Для учета вращения карты нормалей требуется изменить сохранения правильной векторной математики. Используйте узел Rotator для вращения данных, возвращенных из Lerp. Scalar Parameter, называемый Rotation, умножается на Pi и используется как Time input для узла Rotator, связанного с UV и Texture.

Преобразования (объясню позже). Чтобы было понятно, вы поворачиваете значение карты нормалей, а не Texture.

Векторная математика, как эта, является основой 3D-приложений. Все, от перемещения в 3D-пространстве до цветов материалов, является векторами. Изучение векторной математики является одной из лучших вещей, которую вы можете сделать как художник, стремящийся улучшить свои возможности и навыки в каждом аспекте для UE4.

Ambient Occlusion

Дополнительная карта Ambient Occlusion (AO) предназначена для ручного определения микроповерхностной ambient occlusion of a Material (рисунок 15.7). Если этот ввод не определен, UE4 будет динамически генерировать эти данные с помощью карты нормали, но иногда они не совсем точные или это не то, что хочет художник. Канал AO чаще всего используется, когда наборы текстур уже запечены с помощью таких программ, как Substance designer или XNormal, и может быть отключен в большинстве случаев.

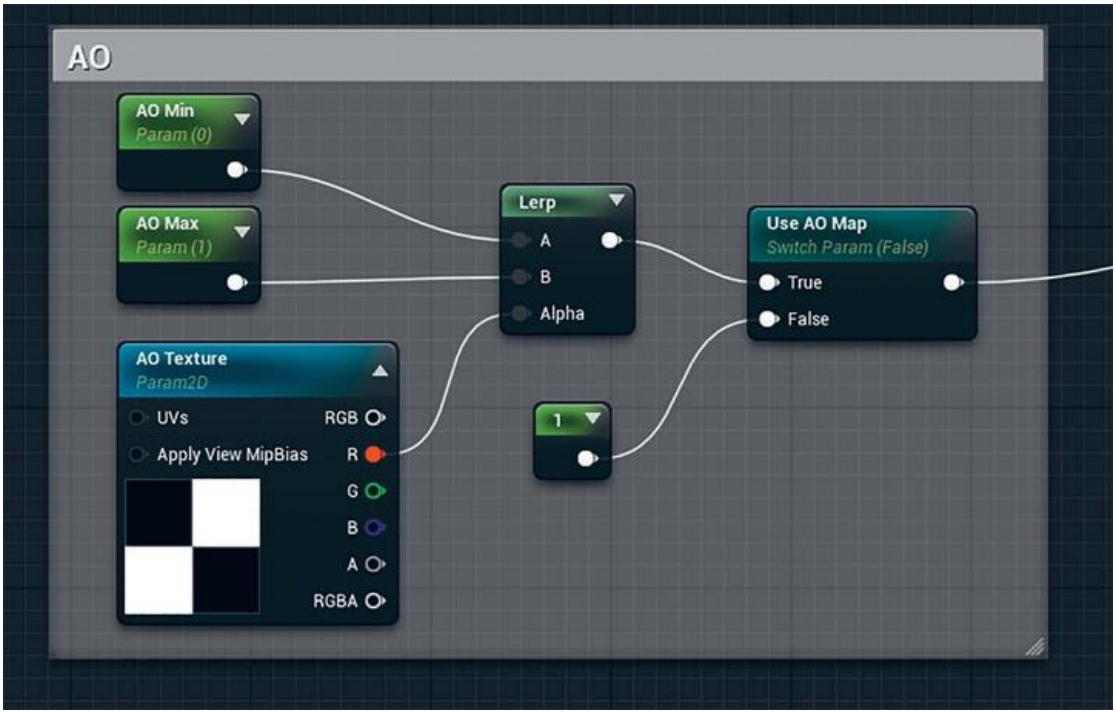

Рисунок 9.7 Узел Ambient Occlusion обычно используется только при особых обстоятельствах, когда материалы UER4 нуждаются в небольшой помощи

Настройте граф, как показано на рисунке 9.7. Как входы roughness и Metallic, AO нужен только скаляр или градация серого; следовательно, используется только красный канал AO Texture Parameter.

Использование AO Map Static Switch Parameter позволяет Material Instances пропустить использование AO map вообще. В отличие от некоторых других Switch Parameters, где на вход False подается Scalar Parameter, здесь мы будем просто назначать узел Constant и устанавливать его 1.0. В отличие от Scalar Parameters, Constant переменные не могут быть изменены в процессе работы.

Texture Scaling и Transform

Настройка масштаба и позиций Textures необходима для правильного оформления материалов. В UE4 вместо настройки масштаба Texture, как в 3D-приложении, вы изменяете UV-координаты поверхности в реальном времени с помощью shader network (рисунок 9.8).

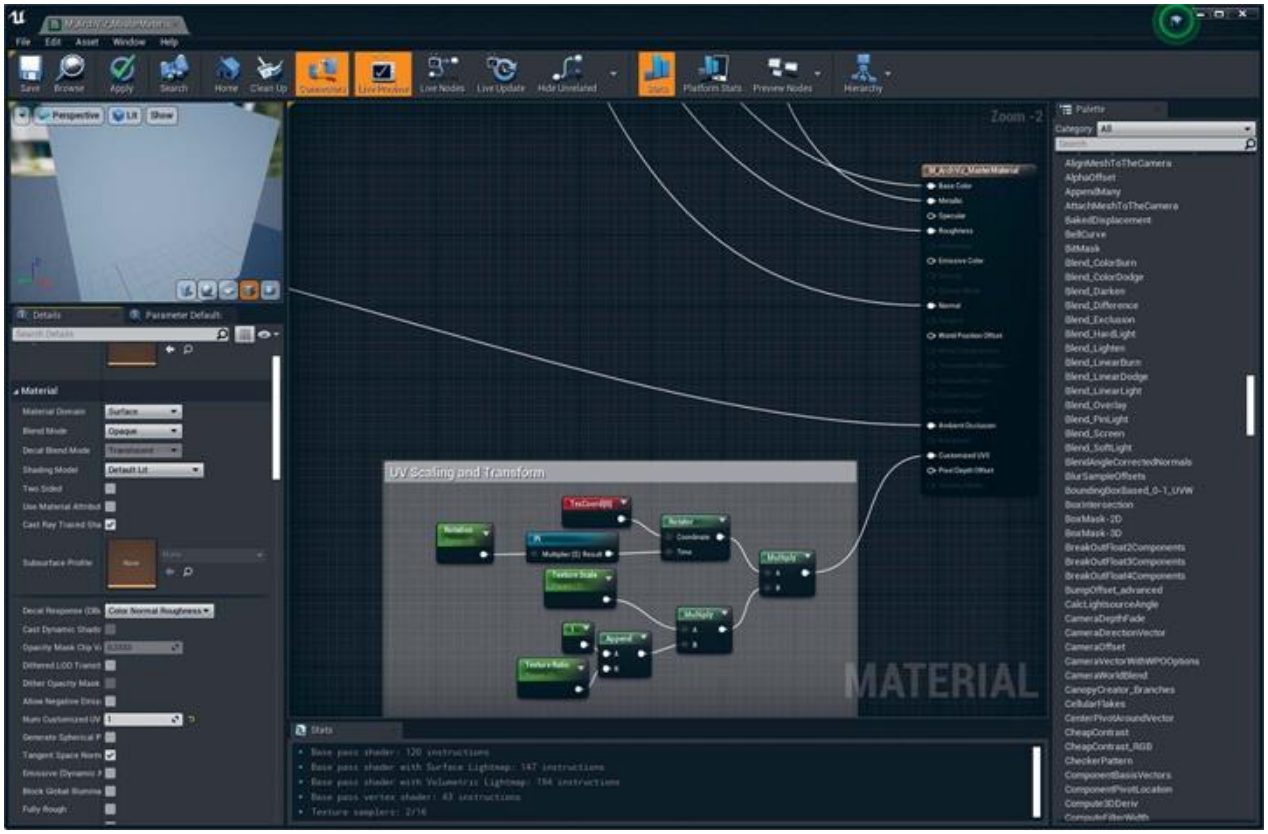

Рисунок 9.8 UV Scaling и Transform

Начните с добавление узла Texture Coordinate (названного TexCoord на рисунке 9.8). Этот узел возвращает UV-координаты в определенном UV-канале.

Затем изменяете координаты UV, которые он возвращает от первого поворота его вокруг центра Texture, используя Rotator. Этот узел приводится в действие путем подключения Rotation Scalar Parameter к Time input16. Для упрощения создания Rotation Parameter в Material Instances он умножается на число pi, превращая Rotation в диапазон 0,0–1,0, где 0,5 равно 180 градусам.

Для масштабирования координат и, следовательно, масштабирования/наложения Texture, они должны быть умножены. Texture Scale Parameter выполняет эту задачу. Значения, превышающие 1, приводят к более крупным тайлам в мозаичной Texture, в то время как значения меньше 0 приводят к тому, что Texture кажется больше, а тай- лы меньше.

Texture Ratio Parameter позволяет текстуре быть неоднородно масштабированной. Узел Append создает значение

Vector2 (значение с двумя float; в данном случае каналы U и V, например, 0, 0 или 0,2, 1,0) с Constant 1,0 в качестве первого значения и Texture Ratio в качестве второго. Это возвращает значение vector2D, например, 1,0, 0,5, которые затем умножаются на Texture Scale Parameter, и этот результат умножается на rotated coordinates.

Результатом всего этого являются измененные UV-координаты, которые вы затем подключаете к атрибуту Customized UV0 для Material. Этот вход не виден по умолчанию. Для включения этой опции вы должны сначала установить свойство Num Customized UVs для Material. Установка его на 1 или больше добавит входные узлы Customized UV для Material.

Custom UVs позволяет избежать необходимости соединять преобразованные UV-ко- ординаты к каждому входу UV всех узлов Texture в вашем Material. Вместо этого данные, поступающие к входам Custom UV, меняют координаты канала UV. Теперь к любой Texture, которая настроена для использования UV-координат канала, будут применены эти изменения значений UV.

Создание Material Instances

На рисунке 9.9 показывается множество Material Instances, используемых на сцене. Почти все из этих Instances основаны на одном Master Material, описанном ранее.

К каждому Material Instance применили новые Textures и настройки Parameters, создавая разнообразную Material Library, доступную здесь.

Rotation Parameter используется несколько раз в этом шейдере. Если у параметров одинаковые названия, изменение одного значения повлияет на все узлы с этим названием.

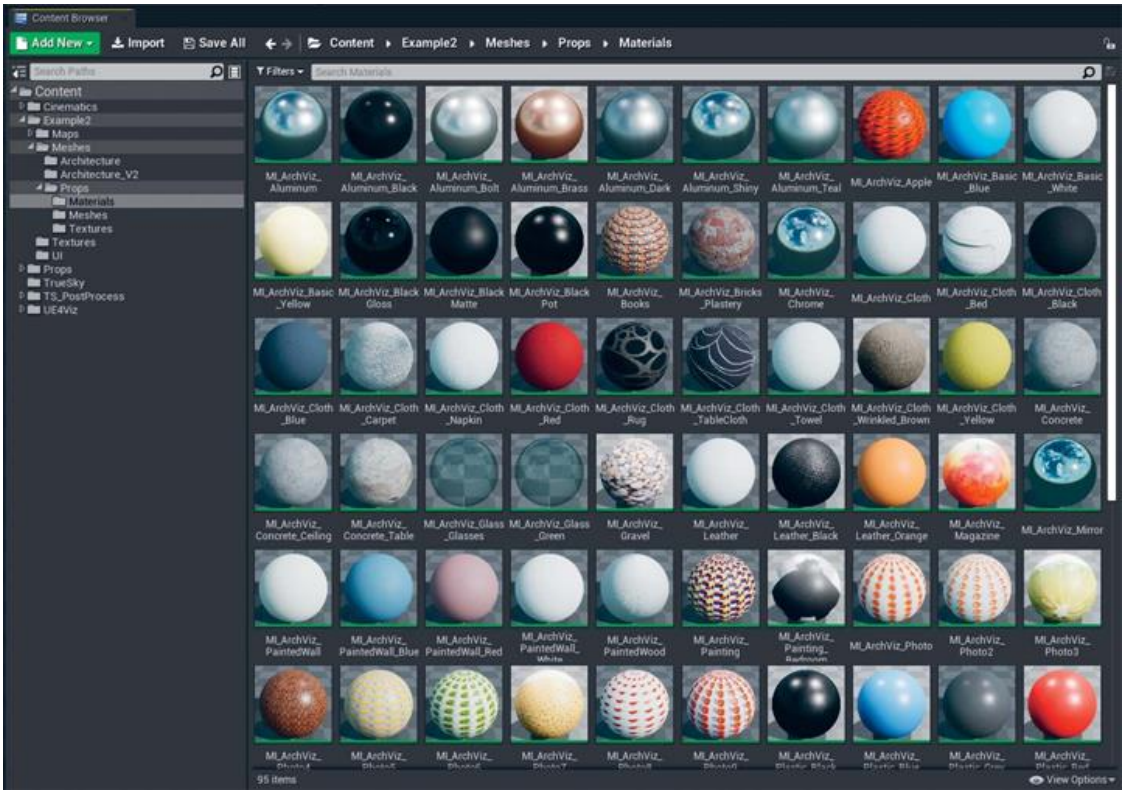

Рисунок 9.9 Content Browser с Materials и Material Instances, унаследованные от Master Material

9.2 Раскраска стен

Для этой сцены потребуется множество красок разных цветов. Для каждого создается Material Instance, каждый цвет определяется с помощью изменения Base Color Tint Parameter. Однако, как вы можете заметить на рисунке 15.10, мне также пришлось настроить множество Parameters, чтобы получить правильно выглядящий Material Instance: мне пришлось назначить Textures, переключить Static Switch Parameters и изменить скалярные значения, что в результате дало богато выглядящую поверхность стены.

Все различные цветовые вариации окрашенных стен происходят от одного Master Material; однако они унаследованы от другого Material Instance, а не напрямую от Master Material (рисунок 9.11). Это демонстрирует одну из самых мощных возможностей Material Instances: способность наследовать Material Instances от других Material Instances.

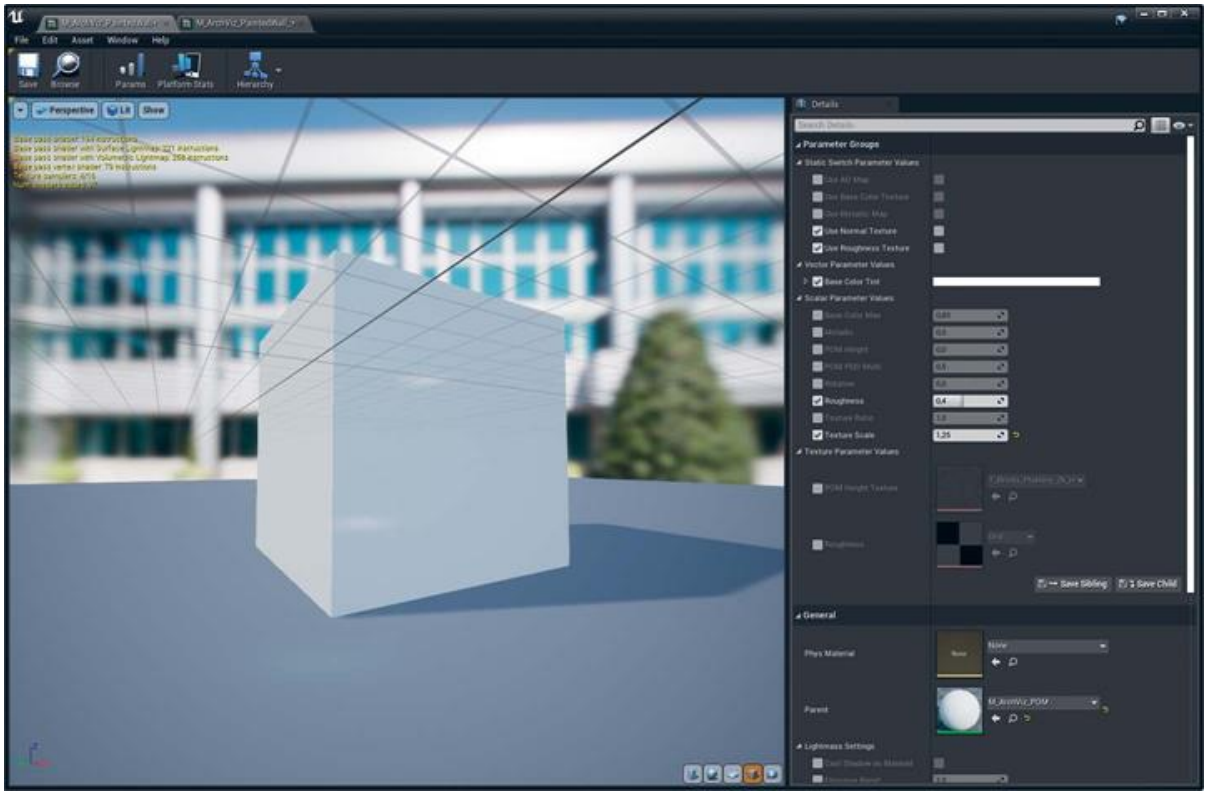

Рисунок 9.10 Раскрашенный Wall Material Instance с множеством измененных Parameters, точно настроенных для нужного вида Material

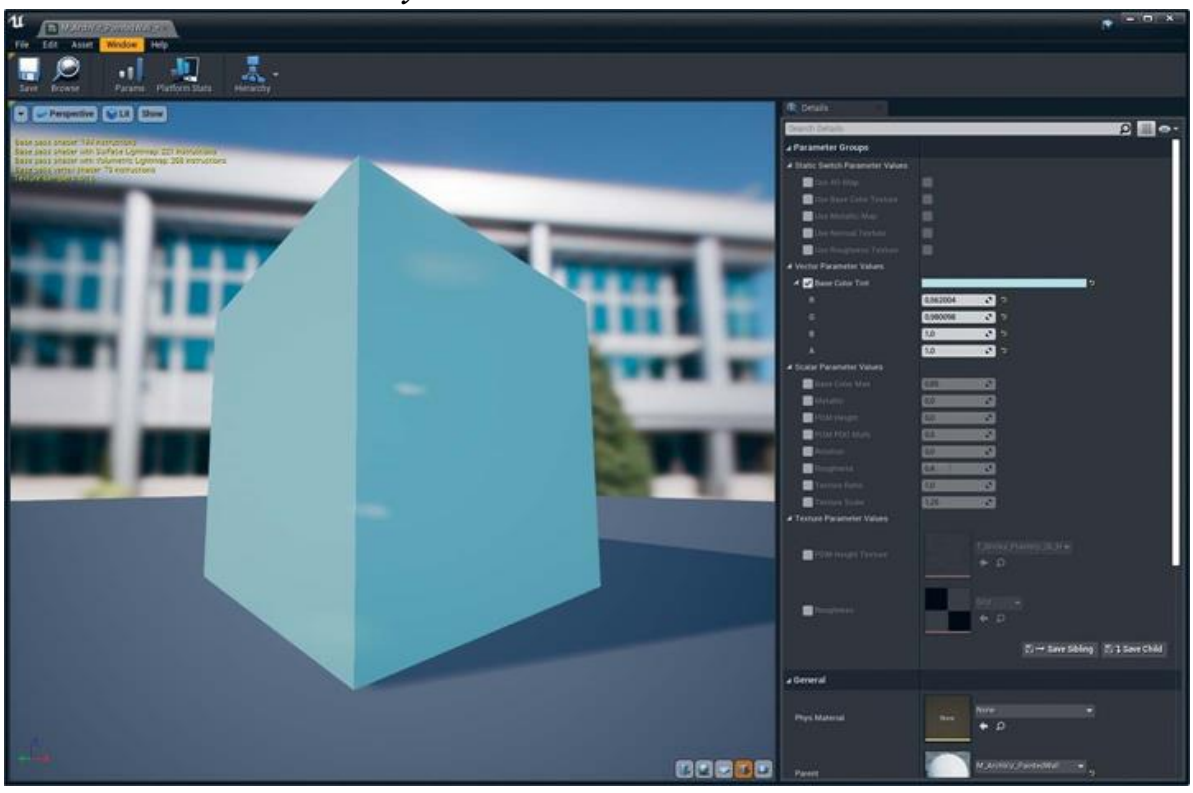

Рисунок 9.11 M_ArchViz_PaintedWall _Blue Material Instance, унаследованные от другого Material Instance: M_ArchViz_PaintedWall

Создание Material Instances от другого Material Instances позволяет вам настроить один Master Material Instance, изменение которого будет затрагивать все Material Instances, унаследованные от него.

Это позволяет создавать бесконечные вариации по данной теме и затем корректировать их все на лету без необходимости изменять каждый отдельно. Свойства, переопределенные в наследном Material Instances, такие как Base Color Tint, не будут изменены изменениями в родителе и сохранят любые сделанные вами изменения.

9.3 Полы

Полы могут быть одними из самых простых Material Instances, но в тоже время одними из самых важных. Получение отражений и деталей прямо на полах в визуализациях важно, так как это обеспечивает контакт с мебелью, стенами и другими Props (рисунок 9.12). Пол составляет огромный процент от привычного вида окружения, таким образом настройка и улучшение напольных материалов стоит усилий.

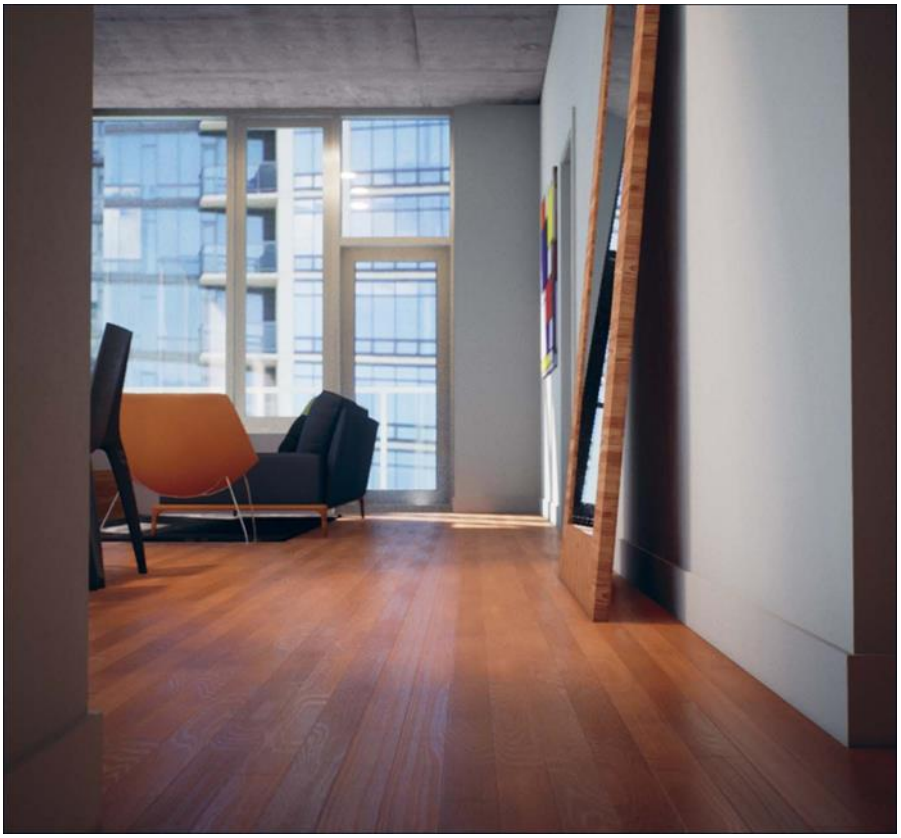

Рисунок 9.12 Напольный Material Instance зависит от хороших карт нормалей и Roughness, которые были сильно изменены с помощью Material Instance Parameters

Используя образец Texture для половых досок, которые использовались клиентом, сгенерируйте карты Normal, Base Color и Roughness Texture с помощью Substance

Bitmap 2 Material (B2M). Это одно из множества доступных платных приложений, которое использует методы обработки на простом изображении и пытается сгенерировать PBR Textures для использования в таких игровых движках, как UE4.

Как вы можете заметить на рисунке 15.13, несмотря на то что Textures, созданные с B2M, хороши, их все еще нужно много дорабатывать. Вот почему настройка

Parameters в Materials, позволяющая вносить изменения, так важна. Без нее вам нужно будет изменять Texture Assets, повторно импортируя каждый раз, чтобы увидеть изменения. Используя этот способ, вы можете посмотреть, как должен выглядеть материал в контексте.

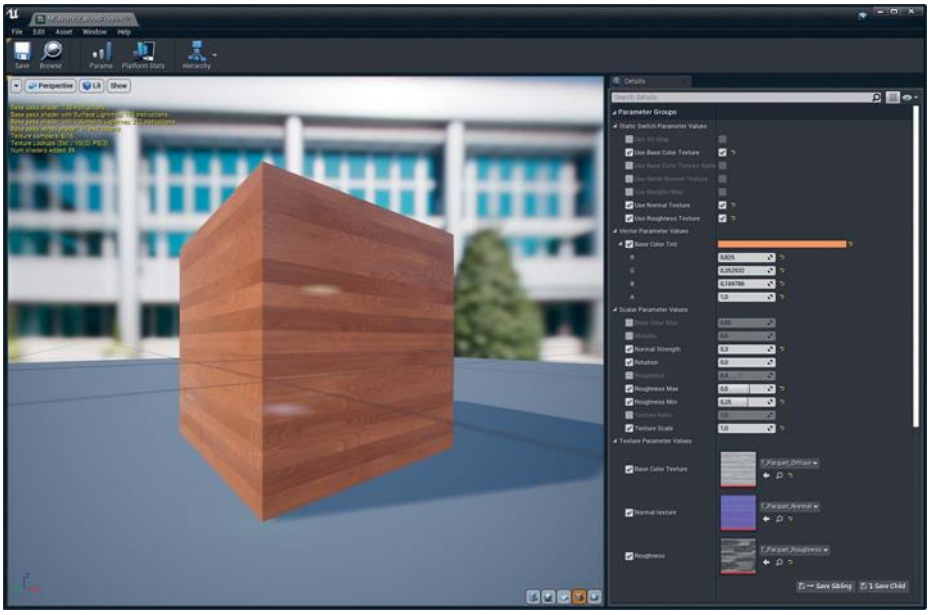

Рисунок 9.13 M_ArchViz_WoodFlooring Material Instance

9.4 Сложные материалы

Несмотря на то что мастер-материал превосходно подходит для большинства непрозрачных поверхностей, он не может справиться со всем. Такие материалы, как стекло или кирпич, требуют более специфичных материалов, чтобы выглядеть лучше, и использования преимущества от возможностей рендеринга UE4.

Parallax Occlusion Mapping

Parallax Occlusion Mapping (POM) является методом для создания похожих на смещение эффектов, на который уходит часть от стоимости рендеринга. Посредством некоторой причудливой математики можно использовать карту высот для смещения Textures, создавая иллюзию глубины (рисунок 9.14).

Рисунок 9.14 Тот же Material, только с включенным POM, показывает, как создается глубина, и помогает лучше определить поверхность, чем только одна карта нормалей

POM создает искаженные UV-координаты, которые затем подаются в каждый Texture’s UV input. Вся эта сложная математика сделана для вас и настроена Material Function, которой требуется только несколько входных данных, например, Height Map для корректной работы.

Так как POM влияет на каждый канал вашего Material, разбить его на собственный Material обычно является хорошей идеей. Настройка Material graph с учетом всех ваших других параметров может привести к чрезмерно сложному Material, который выполняет слишком многое и может стать бременем для обслуживания.

На рисунке 9.15, версия POM для Master Material почти такая же, как и у стандартного Master Material. Создать его можно, дублировав Master Material в Content Browser и затем отредактировав его. Самым большим дополнением является Parallax Occlusion Mapping Function вдалеке слева. Если вы присмотритесь (рисунок 9.16), то увидите, что его легко настроить.

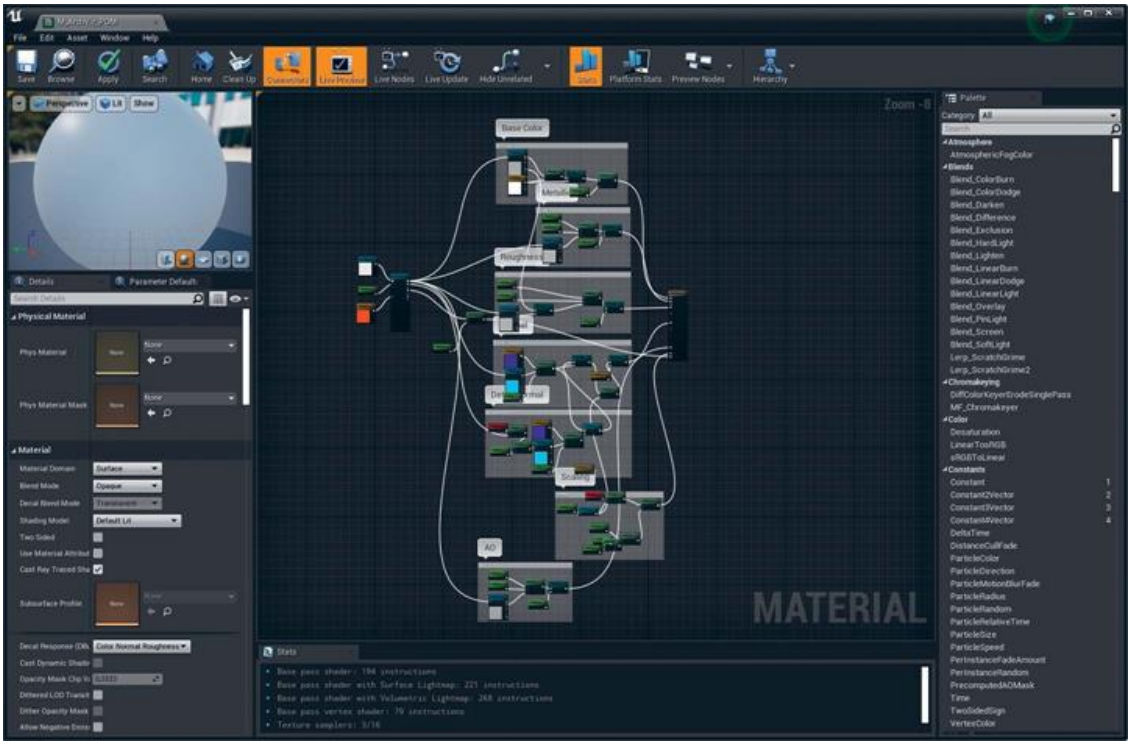

Рисунок 9.15 Обзор POM-версии для Master Material

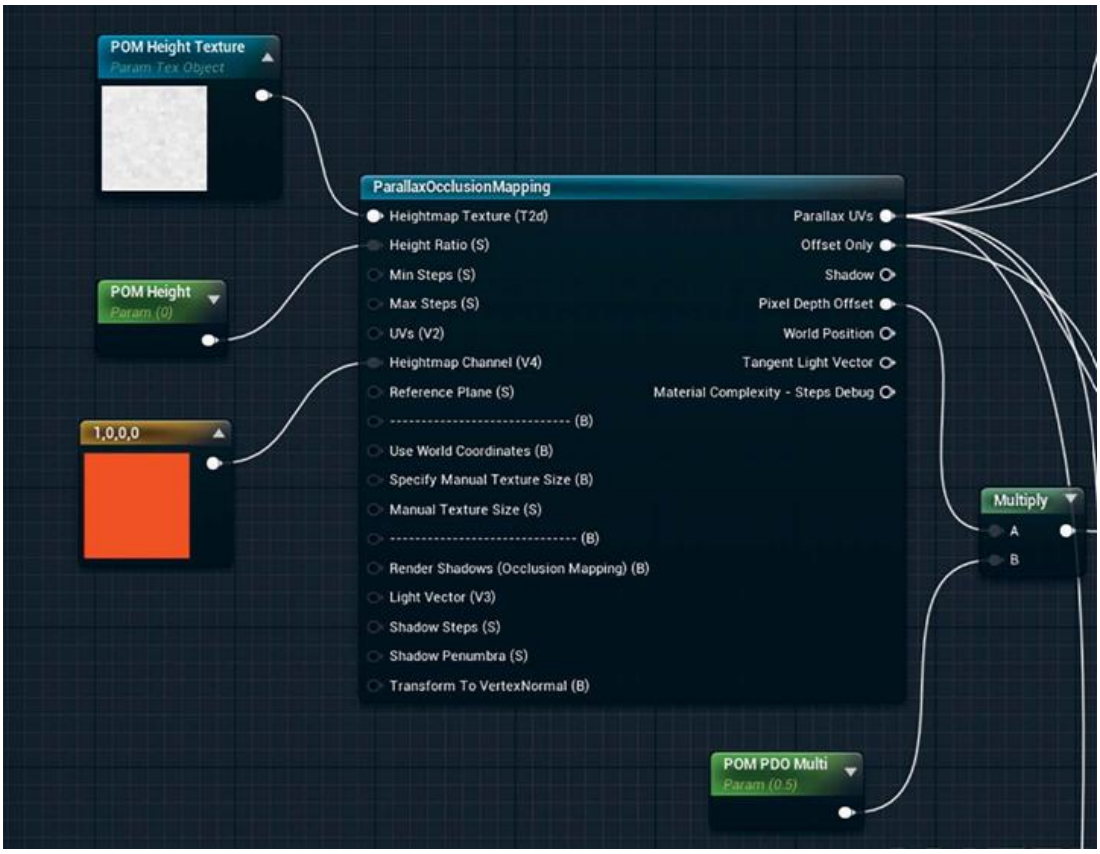

Рисунок 9.16 Детальный обзор функции Parallax Occlusion Mapping; получившиеся искажения UV затем подключаются к каждому входу Texture Sample Parameter's UV

Для настройки функции подключите Texture Object Parameter к входу Heightmap Texture. Texture Object Parameter отличается от других Texture Parameters, которые вы применяли где-то в другом месте. Texture Parameters, который вы используете, возвращают цвет, основанный на назначенной Texture; Texture Object Parameter выдает ссылку на Texture, и ее используют для взаимодействия с функциями материала, а не напрямую для рендеринга.

Для его создания вы должны явно выполнить поиск Texture Object Parameter в Palette или открыть правой кнопкой мыши меню в Material Editor.

Подключите POM Height Scalar Parameter, соединенный с входом Height Ratio, и настройте силу воздействия. Оно должно быть очень низким. Если установить 0,01, то это приведет к очень сильному эффекту. Хорошие значения обычно находятся в диапазоне от 0,001 до 0,005.

Подключите узел Constant Vector 3 к входу Heightmap Channel. Это определит, какой цветовой канал будет выбран из узла POM Height Texture. В данном случае установите его на красный канал, установив для Constant 3 значение 1,0,0.

9.5 Ковры

Использование POM с высокочастотной Texture, такой как карта высот ковра, прекрасно работает, давая глубину и насыщенность даже простым материалам (рисунок 9.17).

Рисунок 9.17 Material ковра использует POM для смещения волокон, что помогает придать ковру глубину и мягкость, особенно в движении и в VR

9.6 Кирпичи

Кирпичи являются классическим примером для демонстрации возможностей рендеринга движка— по веской причине. Их обманчиво сложно рендерить. POM дает хорошие результаты и обеспечивает качество, которое устраивает большинство художников визуализации (рисунок 9.18).

Есть много онлайн-источников для отличных кирпичей и других Texture ресурсов.

Главное — иметь точную карту высот. Получение определения между кирпичом и осадком имеет огромное значение.

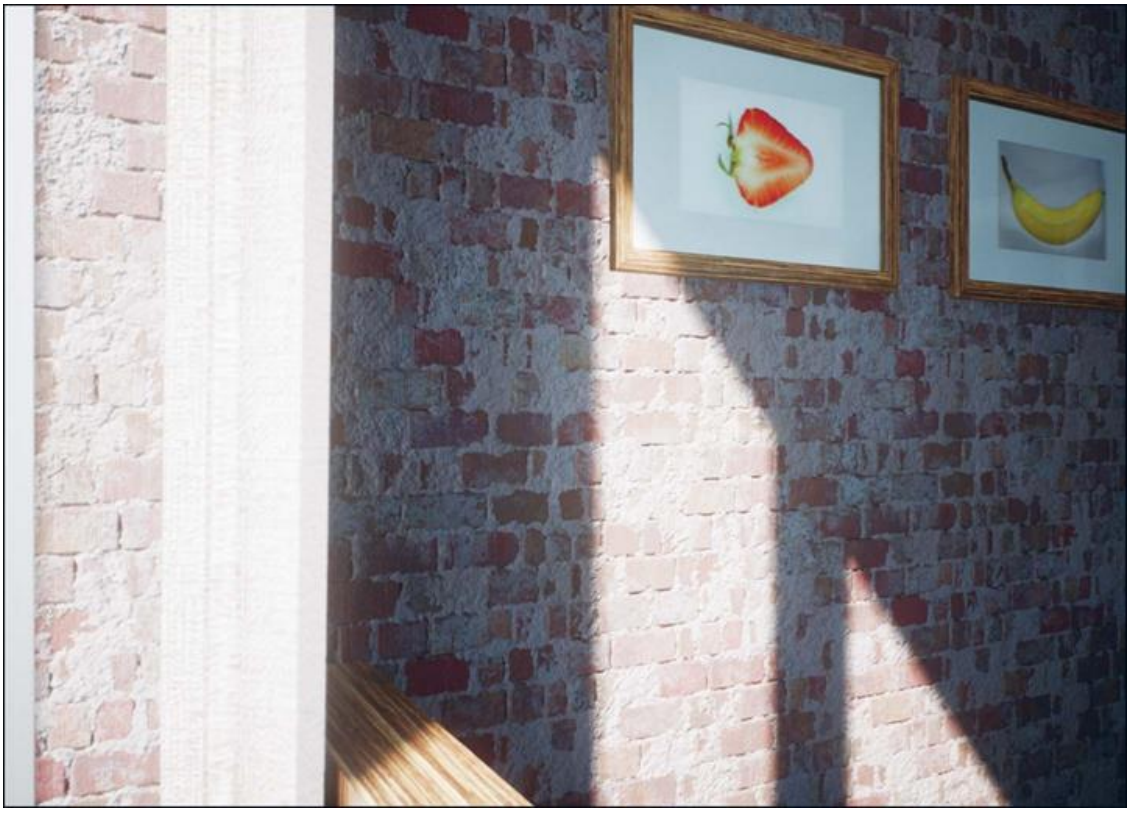

Рисунок 9.18 Brick Wall с использованием POM

9.7 Стекло

Игровые движки долго боролись за создание убедительных полупрозрачных эффектов стекла. Эффективный рендеринг стекла основан на преломлении и отражении, и оба являются дорогими и затратными по времени эффектами для рендеринга (рисунок 9.19). С осторожностью вы можете достичь потрясающего стеклянного Material в UE4.

Рисунок 9.19 Один стеклянный Material, используемый для стаканов, мисок и окон из листового стекла на заднем плане

Существует несколько способов создания Glass Materials, каждый со своими преимуществами и недостатками. Обычно чем точнее материал, тем больше времени требуется для рендеринга. Материал, показанный на рисунке 9.19, является довольно дорогим для рендеринга, но это того стоит, так как повышается качество простых полупрозрачных материалов.

Так как нам понадобится установить Material на Translucent, мы не можем просто использовать Master Material в качестве основы для Glass Materials. Нам нужен новый Material.

Первое, что вам нужно сделать, — Material, который вы создаете, Translucent. Это позволит рендерить его после основной сцены и смешивать поверх после. В панели Material’s Detail без выбранных узлов установите Material’s Blend Mode на Translucent и Lighting Mode на Surface Translucency Volume (рисунок 9.20).

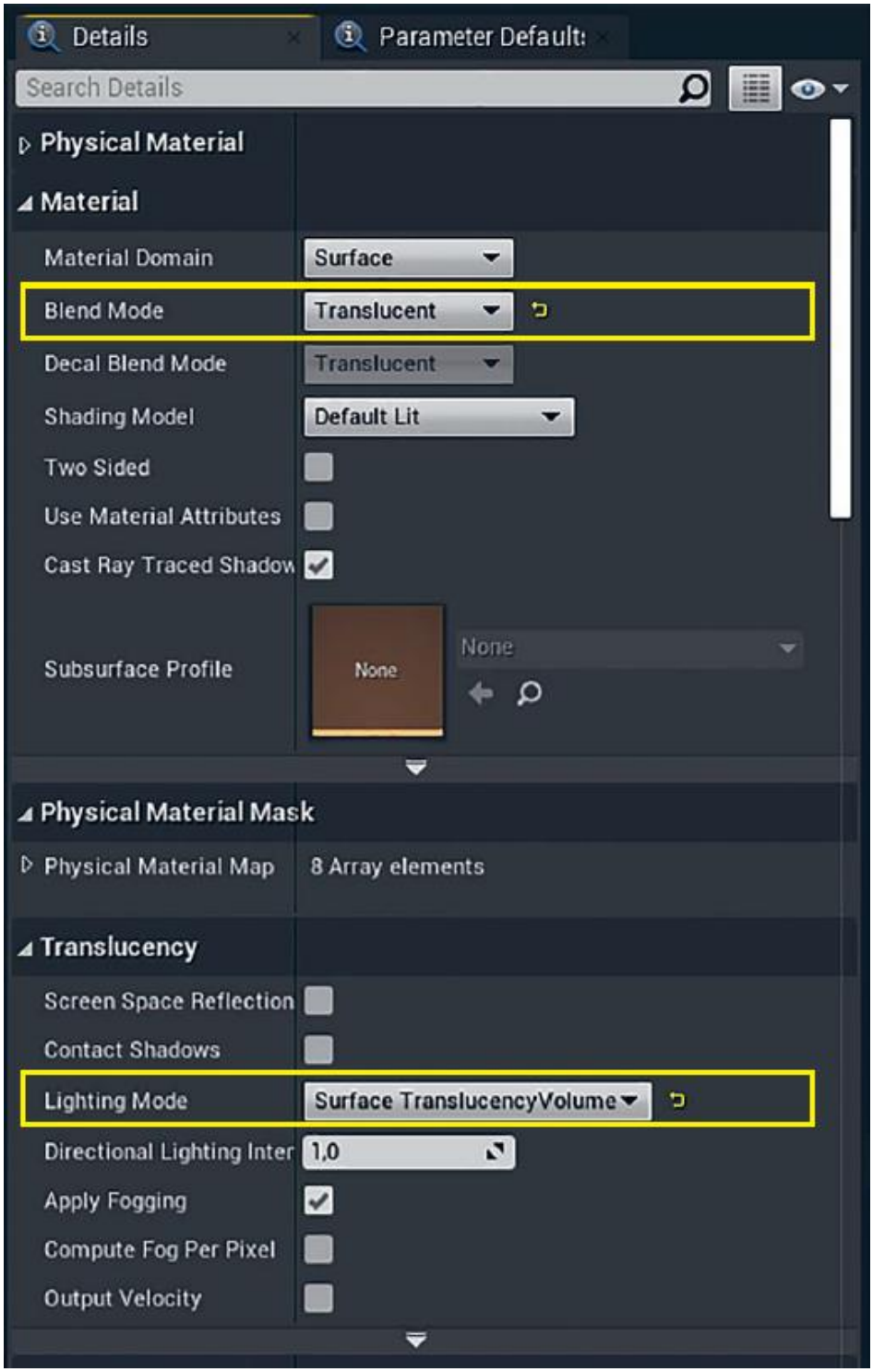

Рисунок 9.20 Detail Panel стекла с Blend Mode, установленным на translucent, и Lighting Mode, установленным на Surface Translucency Volume

Большинство прозрачных материалов полагаются на атрибут Opacity, чтобы модулировать непрозрачность объекта, позволяя отрендерить сцену позади него. В данном случае вы

обойдете это (установите Opacity почти на 1,0) и соедините со своей версией сцены за стеклами, раскрасив и исказив.

Для этого нужно выбрать Scene Texture (рисунок 9.21).

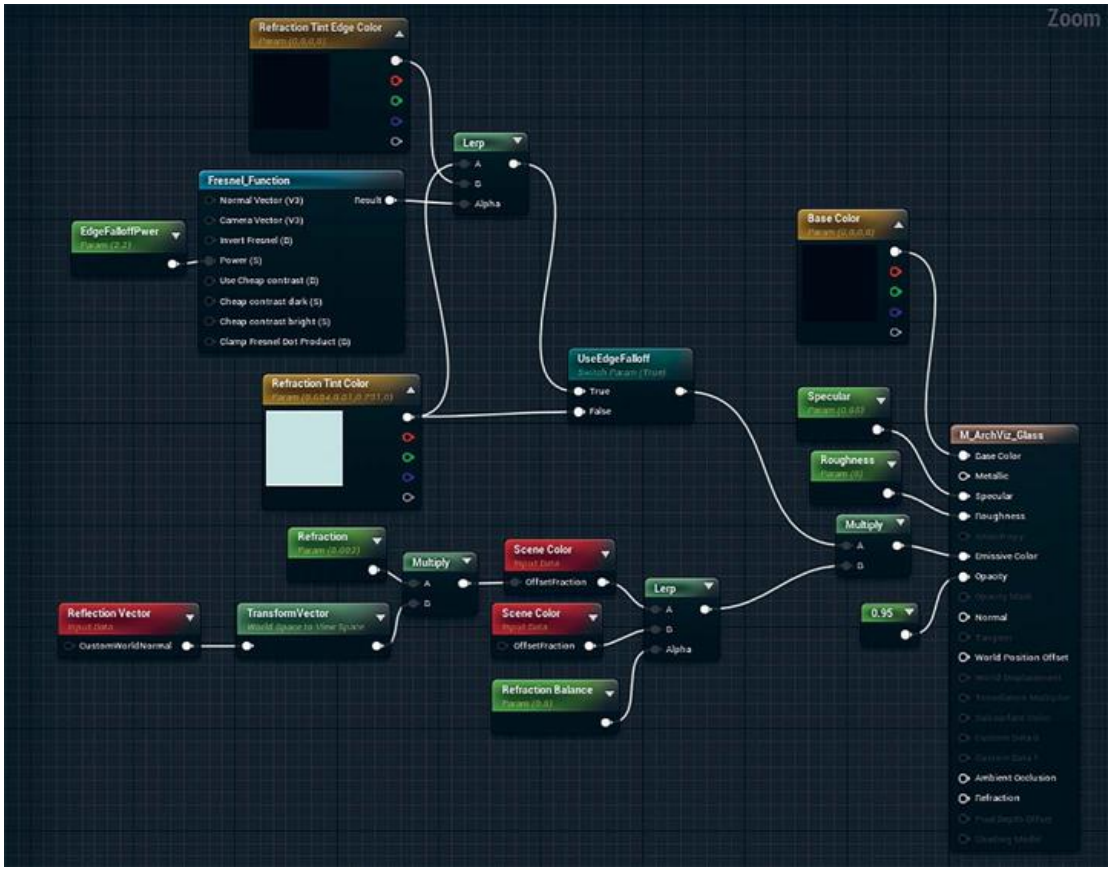

Рисунок 9.21 Сеть ArchViz Glass Material, минуя канал Opacity и создавая вашу собственную измененную Scene Texture, позволяет вам более эффективно контролировать преломление, а отражениям поверхности оставаться яркими

Scene Texture — это отрендеренная сцена до применения полупрозрачности (полупрозрачные объекты в UE4 рендерятся после остальной части сцены и затем компонуются в финальный кадр). Если бы вы передавали Scene Color напрямую атрибуту Emissive для материала, это заставило бы стекло выглядеть полностью прозрачным, не считая отражения от поверхности.

Для искажения Scene Texture возьмите Reflection Vector (входной Vector предоставленный движком), преобразуйте его к вашему виду (переводя его из 3D мирового пространства в значение 2D пространства обзора) и затем используйте эти данные для искажения Scene Color с помощью Offset Fraction.

Offset fraction — это процент (0,0–1,0), который позволяет вам выбрать Scene Color в другом месте пикселя на экране.

Это абсолютно физически не точная техника преломления, но она обеспечивает хорошо выглядящие результаты.

Вы заметите два узла Scene Color на рисунке 9.21, которые связаны с помощью Reflection Balance. Один из них — искаженный Scene Color, а другой без изменений.

Это дает двойной вид панели, который отлично подходит для архитектурной визуализации листового стекла.

Затем вы можете легко подкрасить искаженные и удвоенные образцы Scene Color.

В предположении, что вы установили Use Edge Falloff на true в экземпляре материала, основанном на некотором материале, Freshnel обеспечивает простой спад по Френелю, основанный на нормали поверхности относительно камеры. Используя этот спад, Tint интерполируется между определенным Refraction Tint Color Parameter к Refraction Tint Edge Color Parameter. В противном случае Scene Color окрашивается

Refraction Tint Color Parameter.

Установите атрибуты материала Roughness и Specular с помощью простого Scalar Parameters. Вы можете расширить этот материал для использования текстур, чтобы управлять любым из этих параметров; однако это сделает материал более дорогим для рендеринга и должно использоваться с осторожностью.

Заключение

Благодаря системе PBR в UE4, визуальному редактору материалов и мощности, а также удобству экземпляров материалов, создание красивых материалов происходит быстро и легко. Возврат к созданию материалов в 3D-при- ложениях может быть трудным.

Использование параметров материалов и экземпляров материалов делает повторно используемыми ваши материалы по щелчку, а интерактивный основанный на узлах редактор материалов позволяет вам экспериментировать, изучать и

расширять используемые материалы по мере роста ваших способностей и требований к проекту.

10 СОЗДАНИЕ КИНЕМАТИКИ С SEQUENCER

Интерактивность и исследование являются отличительными чертами интерактивной визуализации в UE4. Однако UE4 также хорошо приспособлен для создания предварительно обработанных статических анимаций, которые соперничают с качеством рендеринга, полученного с использованием трассировки лучей и могут сделать это быстро. С помощью Sequencer, инновационного инструмента анимации Unreal Engine 4, вы можете комбинировать интерактивные миры с ключевыми кадрами анимации камер и акторов для создания превосходных анимаций.

Начало работы с Sequencer

Sequencer Editor является кинематографическим инструментом для редактирования в UE4, позволяющим вам редактировать Sequences.

Sequences являются ассетами, которые вы можете размещать на уровнях, как акто- ра, содержащего ключевые кадры анимационных Tracks. Сиквенсер черпает вдохновение из таких инструментов, как After Effects, Final Cut и других приложений для редактирования и создания видео. Знакомство с этими приложениями облегчает изучение многим художникам визуализации.

Сиквенсор заменяет предыдущий инструмент кинематографического редактирования в UE4, Matinee. Matinee продолжает сосуществовать вместе с Sequencer, но он устарел и в сравнении с Sequencer очень ограничен.

10.1 Master Sequence

В отличие от большинства ассетов, которые вы либо импортируете, либо создаете в Content Browser, Sequences вы создаете на уровне с помощью раскрывающегося меню Cinematics (рисунок 10.1).

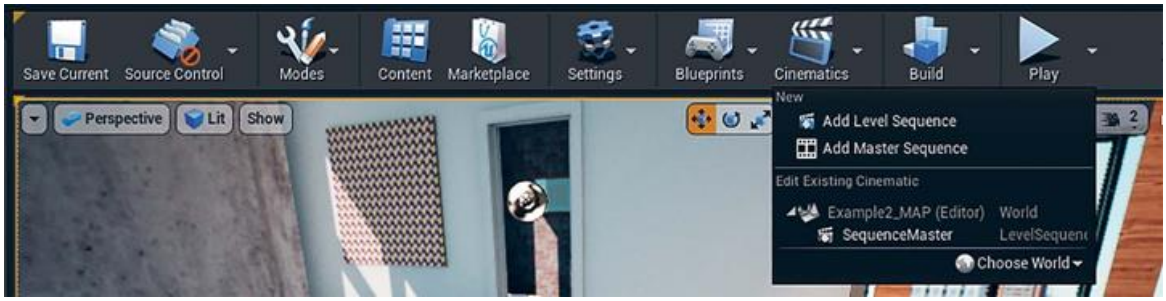

Рисунок 10.1 Создание Master Sequence из выпадающего меню Cinematics в Editor

Вам предоставляется возможность создать Master Sequence, одиночный Level Sequence или устаревший сиквенсор Matinee. Master Sequence — это в основном мастер, который создает Sequence с несколькими sub-sequences (рисунок 10.2). Мастер также создает различные Sequence Assets и сохраняет их в специальном месте в Content directory.

Вы также можете начать с одним Level Sequence, если нужно создать что-то простое. Вы можете включить отдельные Sequences в другие Sequences в любое время.

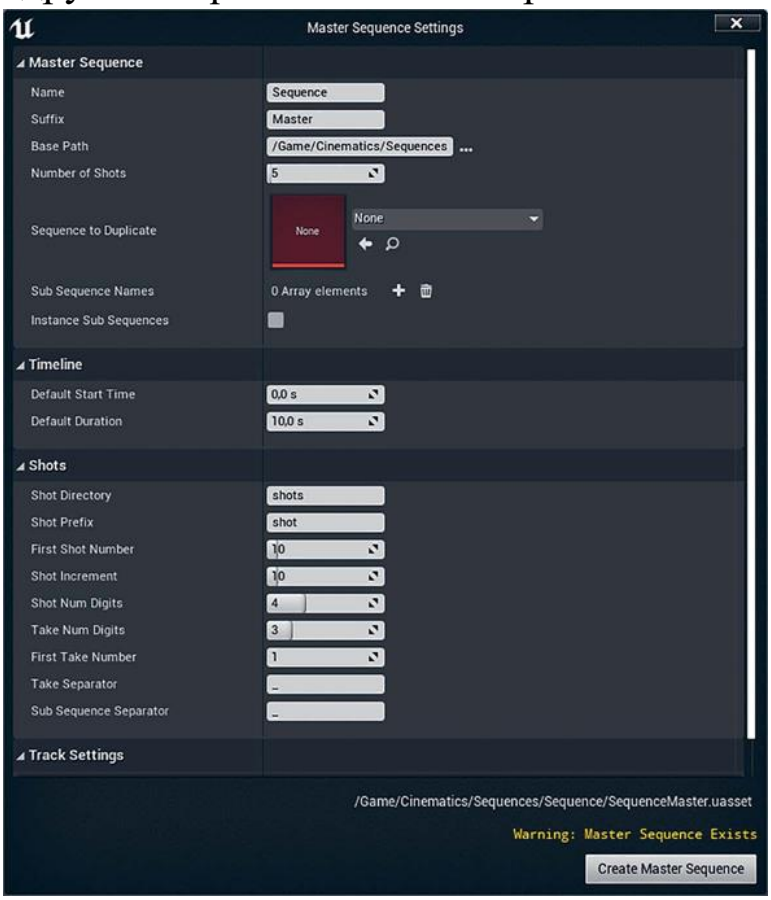

Рисунок 10.2 Настройки мастера Master Sequence

Вы можете принять стандартные настройки для большинства проектов, если не нужно следовать схеме именования или другим стандартам. Единственной измененной настройкой является Default Duration, которую необходимо установить на 10 секунд.

Когда вы создаете Master Sequence, появится окно Sequencer показывающее все дорожки Sequences, расположенные в ряд (рисунок 10.3).

Любой, кто знаком с приложениями нелинейного редактирования, почувствует себя как дома. Вы можете перемещать кадры (Shots) и добавлять, удалять и просто менять продолжительность каждого трека, что делает Shots Track Editor похожим на Adobe Premiere: быстрый, оптимизированный, но с нехваткой точного контроля.

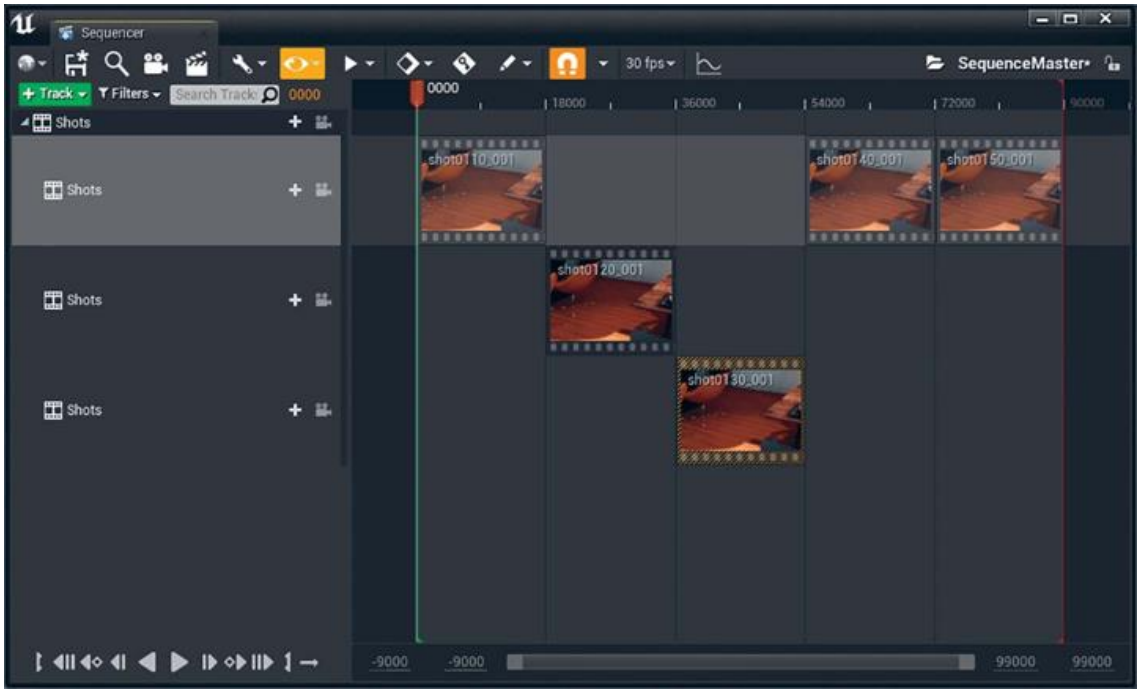

Рисунок 10.3 Master Sequence Editor, в котором вы можете перемещать и редактировать Shots как в программе для нелинейного редактирования видео

Чтобы получить доступ к управлению, дважды нажмите на любой Sequences, чтобы открыть кадр в Sequence Editor (рисунок 10.4). Хотя это тот же Sequencer Editor, но он выглядит по-другому и служит другой цели. Предоставляя Camera Cuts и Camera Actor Tracks, этот редактор больше похож на Adobe After

Effects: управляемый ключевыми кадрами интерфейс анимации и эффектов.

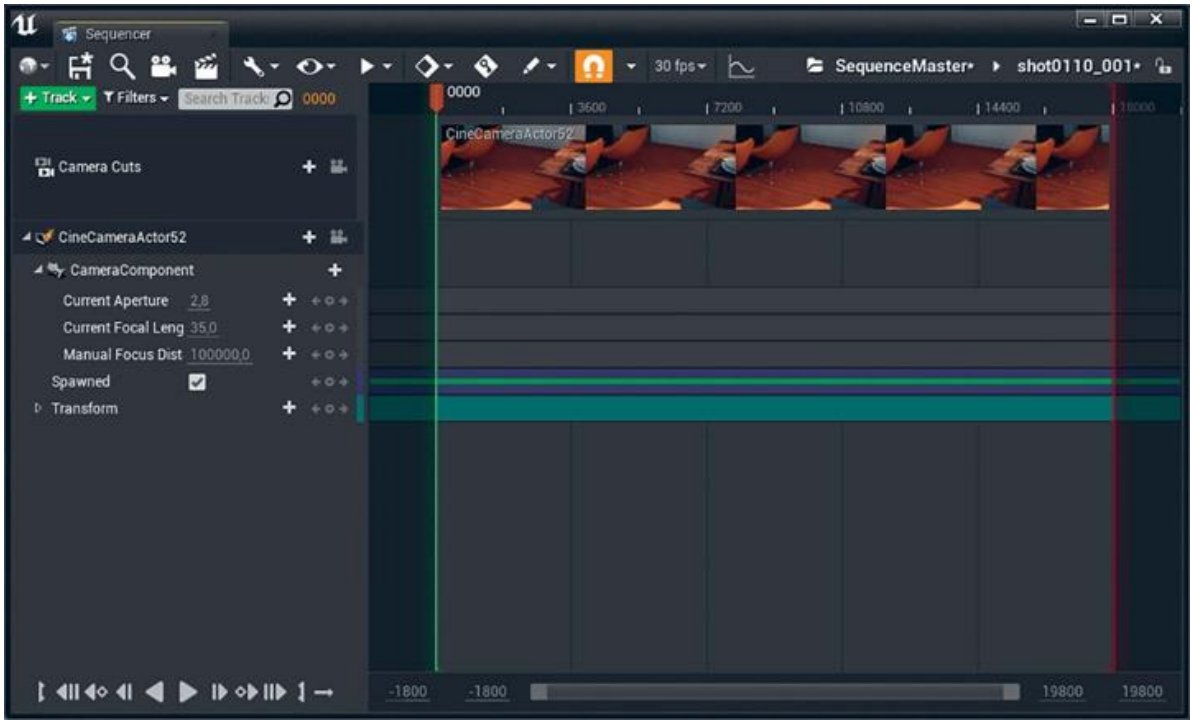

Рисунок 10.4 Единственный кадр, открытый для редактирования в Sequence Editor, показывающий, как меняется редактор, чтобы выглядеть более похожим на пакет эффектов ключевых кадров, например, After Effects

10.2 Динамические камеры

В каждом из этих кадров Sequences, CineCameraActor появляется динамически в Sequence и уничтожается, когда Sequence завершается. Эта особенность по-настоящему мощная и означает, что не нужно размещать одиночного Camera Actor на вашем уровне, так как они создаются и уничтожаются на лету в сиквенсоре.

Эти камеры действуют как любой другой Camera Actor и отображают настройки объектива, глубину резкости и настройки фокуса, как и все доступные эффекты постпроцессинга, позволяя настроить каждую камеру, как вам нужно (рисунок 10.5).

Обратите внимание на Details Panel и кнопку Add Keyframe, которая появляется слева для каждого из связанного с

анимацией свойства. Как и большинство окон в UE4, вы можете закрепить сиквенсор в окне UE4 для наведения порядка.

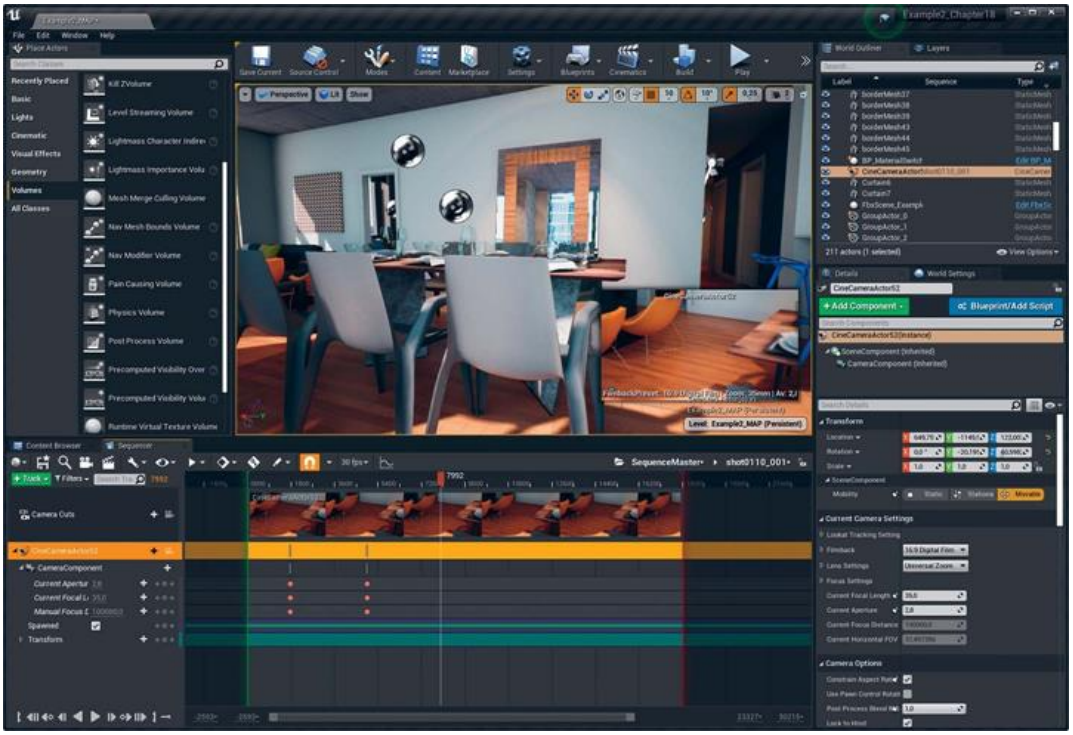

Рисунок 10.5 Закрепленный Sequencer в UE4 Editor

10.3 Анимация камеры

Заставить камеру передвигаться и меняться со временем — следующий очевидный шаг, но сделать это может быть не так легко.

Настройка ключевых кадров

У вас есть несколько способов установить ключевые кадры в сиквенсоре. Вы можете включить Auto-Key, создавая ключевой кадр каждый раз, когда меняете свойство или перемещаете, поворачиваете или масштабируете актора. Это может быстро привести к беспорядку.

Вы также можете вручную установить ключевые кадры. Вы можете сделать это либо в интерфейсе сиквенсора, либо напрямую в свойствах и Viewports. Когда сиквенсор активирован, связанные с анимацией свойства для объектов Actors показывают кнопку Add Keyframe (рисунок 16.5), позволяя вам легко добавлять ключевые кадры, когда это нужно.

Третий способ является смесью из предыдущих вариантов. Вы можете автоматически устанавливать ключевые кадры, но ограничивать их свойствами, которые вы уже добавили к ключевым кадрам (рисунок 10.6). Обычно используется этот вариант, так как он сочетает простоту в использовании auto-keyframing, не беспокоясь о случайном добавлении ключевых кадров на дорожки, которые не нужно анимировать.

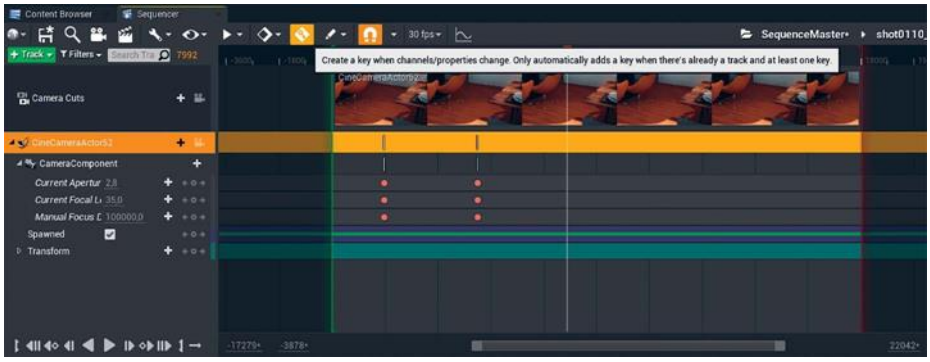

Рисунок 10.6 Настройка параметров Auto-Keyframe для добавления только ключевых кадров к уже анимированным дорожкам и свойствам

Упражнение 13. Установка камеры

Теперь, когда вы можете установить несколько ключевых кадров, давайте поставим камеру на место.

Для просмотра через камеру выберите справа значок камеры, которую хотите анимировать. Это установит Viewport для управление камерой, позволит летать Actor по сцене, прикрепив его к вашему обзору.

Значки камеры появляются во многих областях интерфейса сиквенсора, и это может немного сбивать с толку (рисунок 10.6). Вы заметите, что дорожка Camera Cuts также содержит значок камеры. Если вы нажмете на нее, вам откроется вид с перспективы Sequence, включающий любые camera cuts, которые вы создали.

Как правило, вы будете переключаться между этими видами, пока работаете, настраивая ключевые кадры камеры и просматривая контекстный результат с другими кадрами.

При управлении камерой вы смотрите через объектив камеры и перемещаете ее по сцене с помощью стандартного

управления Perspective Viewport. Вы можете настроить ключевые кадры камеры в Orthographic-видах или свойства Camera Actor в панели Details.

Дорожки и название камеры

Вы можете обратить внимание на название дорожки в сиквенсоре, отличное от метки в предпросмотре ленты времени. Это немного сбивает с толку, так как они именуются независимо (вы можете, к примеру, использовать один и тот же Camera Actor для различных дорожек в разных Sequences). Название в предпросмотре является названием актора в мире. Вы можете переименовать ваш Cine Camera Actor с помощью панели Details для обновления названия в ленте времени.

Название в левой панели является названием дорожки. Вы можете дважды нажать на это имя, чтобы переименовать дорожку.

10.4 Переходы

Единственный переход, который предлагает UE4, — дорожка Fade. Недоступны cross- dissolve или другие эффекты перехода. Это в основном из-за соображений о производительности. Выполнение cross-dissolve заставляет дважды отрисовывать сцену и совмещает их во время растворения. Хотя некоторые сцены могут справиться с этим, но большинству будет тяжело отрендерить так много информации.

Редактирование кадров

Для возврата к Master Sequence, выберите заголовок в правом верхнем углу окна си- квенсора (рисунок 10.7).

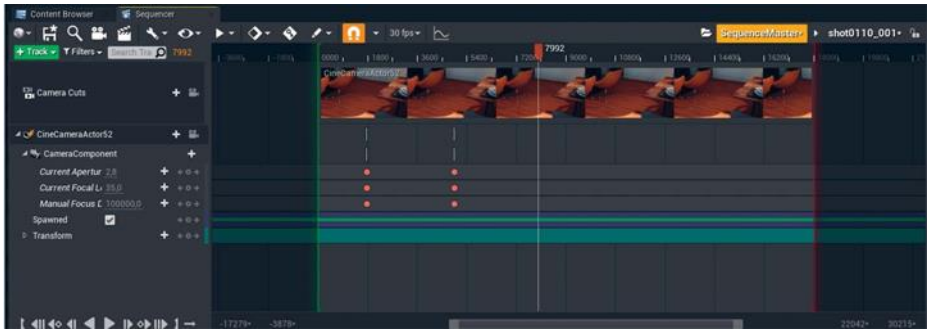

Рисунок 10.7 Завершенный кадр с SequenceMaster-треком, выделенным в правом верхнем углу окна секвенсора

Вы увидите, что предварительный просмотр снимка был обновлен. Вы можете использовать это как руководство для настройки входных и выходных точек в Sequence. Чтобы лучше увидеть, увеличьте масштаб timeline с помощью ползунка диапазона внизу окна сиквенсора.

Продолжайте редактирование оставшихся снимков. Отредактируйте снимок, затем вернитесь к основной последовательности, чтобы увидеть ее в контексте, измените ее и повторяйте до тех пор, пока не будете довольны всеми кадрами в последовательности.

Сохранение

Не забывайте периодически сохраняться. Sequences сохраняются как UASSET-файлы в Content directory вашего проекта.

Вам не нужно сохранять уровень, на котором находится Sequence после того, как вы впервые разместили Sequence Actor. Этот актор ведет себя как ссылка на ваш Master Sequence, позволяя инициализировать данные Sequence, когда уровень загружается, и тем самым дает Blueprints получить доступ к нему, просто ссылаясь на объект Actor.

10.5 Совместная работа

Так как кадры хранятся как отдельные пакеты UASSET, несколько членов команды могут одновременно работать с одним Sequence. Также установка света, реквизита и другое декорирование может продолжаться, пока анимация разрабатывается параллельно. Это экономит время и позволяет быстрее создавать наброски в производственном конвейере.

Рендеринг в видео

После усовершенствования Sequence время вывести его на диск. Именно здесь скорость UE4 высока. Целые анимации могут быть доставлены (не только отрисованы, но и на YouTube) за время, необходимое одному кадру для отрисовки в стандартном рендеринге с трассировкой лучей, в таком как Mental Ray или V-Ray.

Открыв ваш Sequence, нажмите на кнопку Render to Video (рисунок 10.8) в панели инструментов Sequencer. Откроется диалоговое окно Render Movie Settings (рисунок 10.9). Здесь вы можете установить широкий спектр параметров рендеринга и экспорта. От рамок бампера до предварительной прокрутки сжигаемых модулей UE4 предлагает инструменты профессионального уровня для видеоредакторов и композиторов.

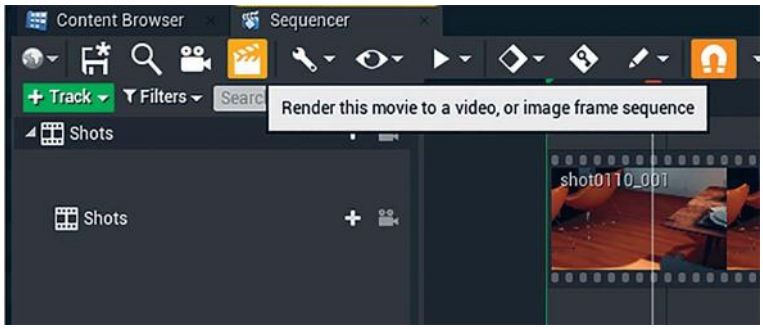

Рисунок 10.8 Выбор кнопки Render to Movie в Sequencer

Render Movie Settings

Можно установить его на рендеринг в разрешении 4K с 60 кадрами в секунду (рисунок 10.9). Это создает шелковисто-гладкие, резкие анимации, которые действительно привлекают внимание. Эти файлы весят более 100 Гб за минуту без сжатия.

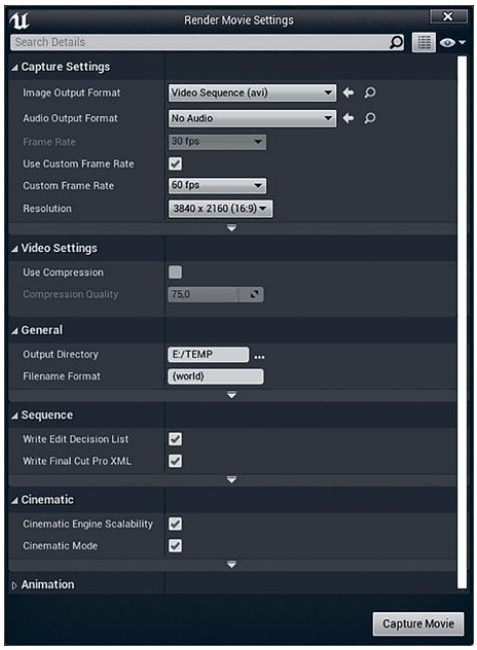

Рисунок 10.9 Диалоговое окно Render Movie Settings, установленное на разрешение 4K (3840×2160) и 60 кадров в секунду

Можно рендерить напрямую в AVI (Video Sequence), но, если в вашей студии есть более обширный процесс постпродакшена, основанный на линейном цвете, вы также можете рендерить буферы HDR-изображений отдельно, используя опцию Custom outputs.

Вы также можете выбрать сжатие ваших AVI для сохранения места (не рекомендуется) или экспортировать на отдельные кадры в распространенных форматах, таких как BMP и EXR.

10.6 Процесс рендеринга

Когда UE4 начинает рендеринг, появляется маленькое окно рендеринга и начинается запись на диск (рисунок 10.10). Окно рендеринга расположено в верхнем левом углу окна, а редактор находится в фоновом режиме. Обратите внимание на уведомление Capturing video в нижнем правом углу. Также обратите внимание на высокое использование диска для записи — чем быстрее, тем лучше.

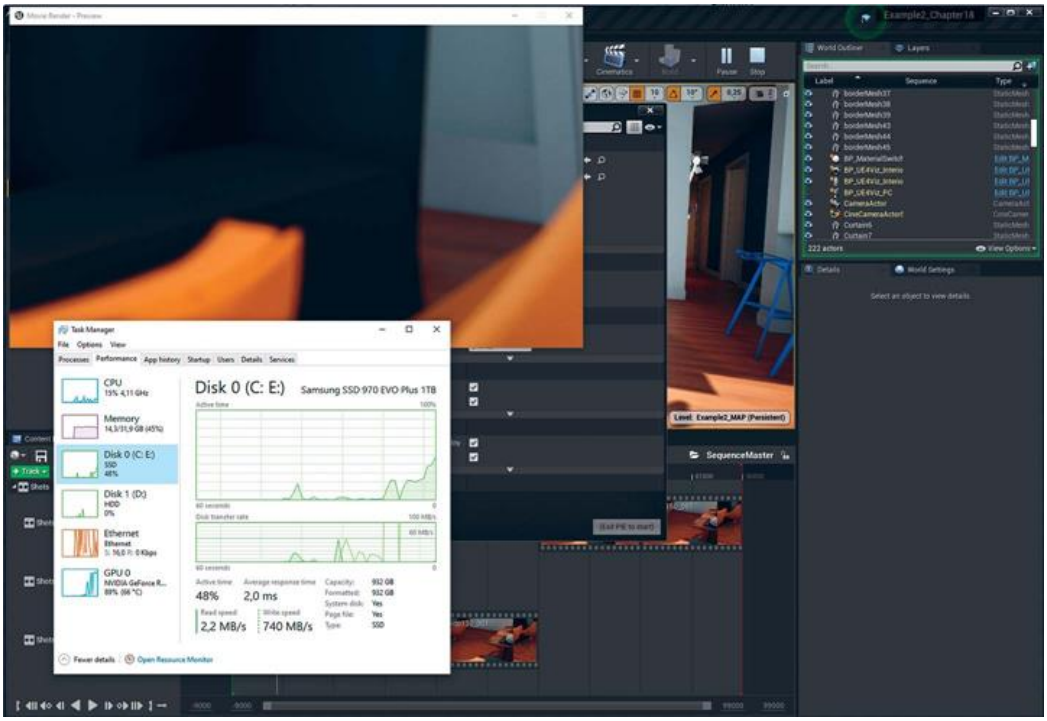

Рисунок 10.10 UE4 рендерит Sequence на диск

На данном этапе рендеринг ограничивается не визуальными эффектами, а скоростью жесткого диска, на который он записывается. Запись кадра на диск длится дольше, чем рендеринг кадра.

Записывайте на диск, который дает самую быструю и стабильную производительность. Обязательно проверьте. Диски с более устойчивой скоростью записи принесут огромные выгоды.

После завершения анимации обработайте ее так же, как и любой другой видеофайл. Он может напрямую отправиться в интернет или включаться в более крупное видео, точно так же, как и ваши стандартные видео визуализации.

Заключение

Возможность адаптироваться к нуждам клиента в момент обращения является еще одной причиной, почему так много студий визуализации переходят на UE4 не только для создания интерактивности, но и для предоставления, потрясающего предварительно отрендеренного контента.

Эта 90-секундная последовательность требует примерно 10 минут на рендеринг — в разрешении 4K с 60 кадрами в секунду. Больше времени уходит на сжатие, копирование и загрузку видеофайлов, созданных с помощью UE4, нежели на их рендеринг.

Благодаря мощным возможностям Sequencer, способности точно видеть, что будет в кадре вашей камеры все время, и физически корректировать модель Cine Camera, вы можете начать создавать видеоконтент в UE4, который соответствует, а иногда и превосходит качество традиционных рендеров.

11 ПОДГОТОВКА УРОВНЯ К ИНТЕРАКТИВНОСТИ

Основываясь на простом Pawn, Game Mode и Player Controller из первого примера проекта, мы легко настроим возможность исследования уровня. Однако вы должны настроить ваш уровень с учетом коллизий, запуска пользователя

и других настроек, прежде чем пользователь начнет успешно перемещаться по нему.

11.1 Настройка Level

Для настройки интерактивности уровня вам нужно настроить использование курсора мыши и сенсорного ввода для Level и Player Controller. Вам также станут доступны события взаимодействия с мышью, которые позволят вам выделять и нажимать на акто- ров в игровом мире в процессе работы.

Конечно, вы захотите открыть уровень, прежде чем продолжить. Вы, возможно, как и я, захотите сохранить новую версию уровня, которая позволит легко вернуться к его неизмененной версии.

Упражнение 14. Добавление Player Start Actor

Каждый UE4 Level требует Player Start Actor. Это простой Actor, который сообщает UE4, где расположить пользователя на уровне при начале игры.

Перетащите Player Start Actor из панели Place Actors в Level рядом с тем местом, где вы хотите, чтобы пользователь появлялся в мире каждый раз (рисунок 11.1).

Также вы должны повернуть Player Start Actor лицом по направлению, в котором хотите, чтобы пользователь начинал. Синяя стрелка в середине Player Start Actor указывает на Player Start Actor’s forward vector.

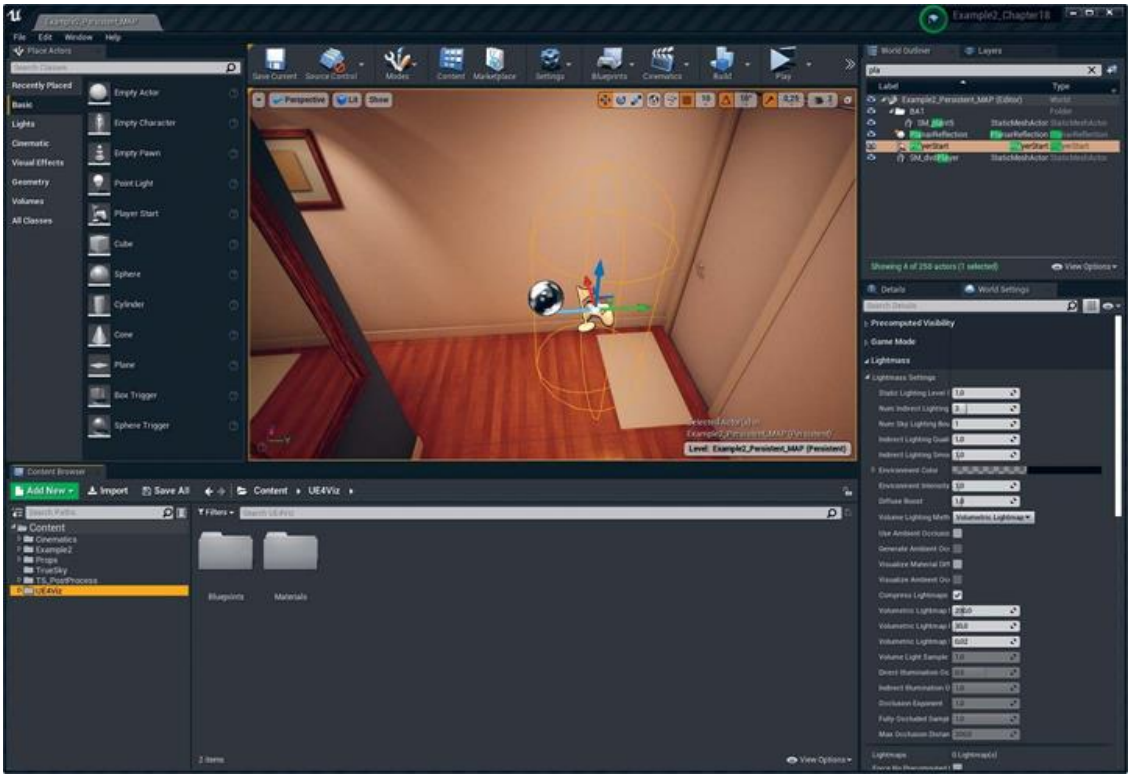

Рисунок 11.1 Расположение Player Start на Level

11.2 Добавление коллизий

Теперь вы можете протестировать ваше приложение, нажав кнопку Play in Editor. Однако высока вероятность, что вы обнаружите себя упавшим ниже вашего пола на уровне, когда сделаете это. Это может быть связано с отсутствием информации о коллизии.

Обнаружение коллизий является серьезной задачей в интерактивных играх и симуляциях. Знание, когда один актор «пересекся» или столкнулся с другим актором или зашел в некоторую область, является важным для создания живых миров.

В UE4 вы можете использовать Player Collision View Mode, как показано на рисунке 11.2, для предпросмотра столкнувшихся акторов. В Viewport выберите кнопку

ViewMode и из выпавшего списка — Player Collison. Чтобы вернуться к вашему обычному виду, установите обратно Lit для Viewmode.

Как видите (или, точнее, не видите), на рисунке нет этажей или стен, но у некоторого реквизита и дверных мешей уже есть коллизии. Они представлены с помощью плоских затененных версий ваших примитивов коллизий.

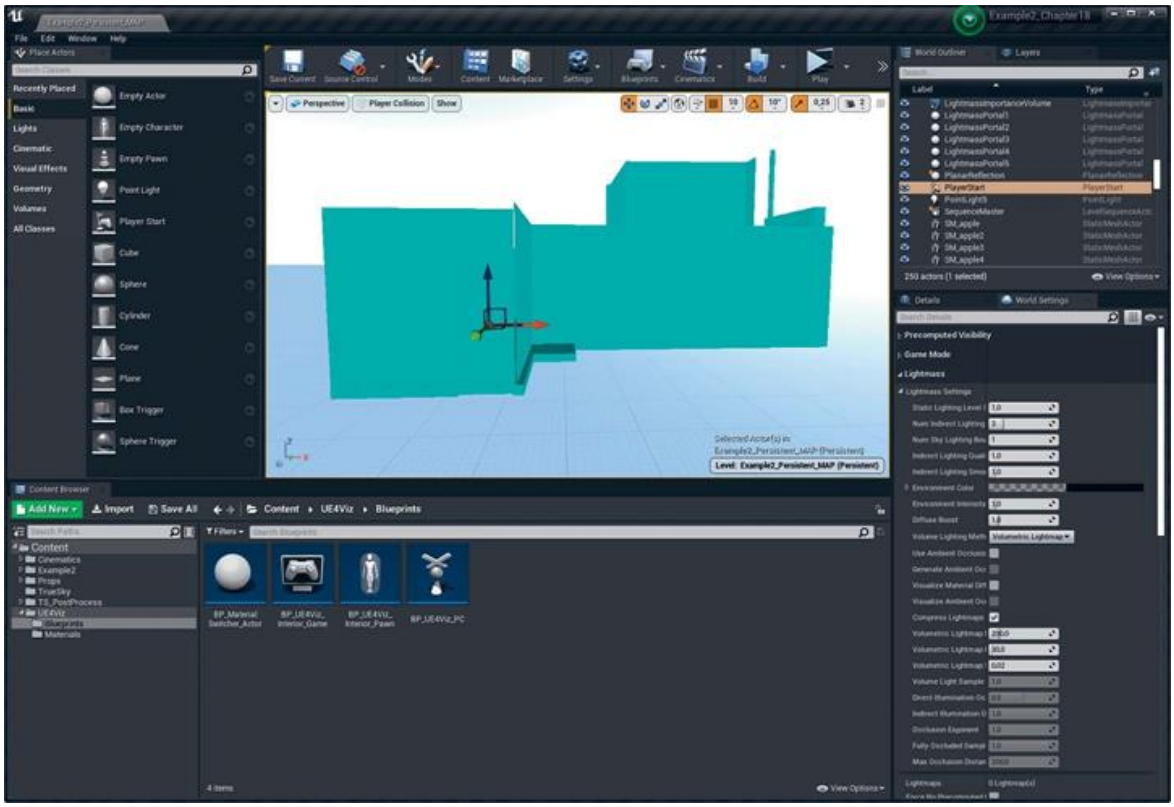

Рисунок 11.2 Collision, отображаемая с помощью Player Collision View Mode

Collision в UE4 может быть сложной для понимания на первый взгляд. К счастью, визуализации имеют тенденцию становиться более простыми, статичными, с небольшим количеством динамичных Actors (в сравнении с игрой с десятками или сотнями героев на экране и в игровом мире одновременно), чтобы помочь вам избежать некоторых сложностей, с которыми вы можете столкнуться при настройке коллизий.

Сложные коллизии против простых

Игры, как правило, полагаются на упрощенную версию моделей для выполнения расчетов столкновений. Это связано с тем, что эти вычисления являются и становятся все более

дорогостоящими, чем больше полигонов и информации требуется обработать.

UE4 использует collision primitives — простые фигуры, такие как кубики, сферы, и капсулы, — которые можно создать в Engine или с помощью низкополигональных фигур, сделанных в 3D-приложении, для использования в качестве simple collision. Это позволяет графически применить на несколько порядков больше полигонов, так как физический движок использует оптимизированную версию сцены для своих вычислений.

UE4 также может выполнять per-polygon collision по мере необходимости. Это часто косметически используют в играх для добавления определенных графических эффектов, таких как повреждение стен или транспортных средств, которые происходят точно в месте видимого удара, в то время как фактическая физическая коллизия использует упрощенную модель для скорости.

Так как сцены визуализации могут быть относительно простыми с несколькими взаимодействующими элементами, вы часто можете использовать per-poly или complex collision (как это известно в UE4) вместо простого collision, позволяющего вам пропустить время и усилия по созданию собственной геометрии коллизий.

11.3 Стены и пол

Главное, чтобы ваши пользователи не проходили через стены и полы в симуляции, так как эти меши геометрически просты и имеют неудобную форму, мы можем безопасно установить их для использования обычного per-polygon collision.

Для этого откройте Static Mesh Asset в Static Mesh Editor. Во вкладке Details раскройте Static Mesh Settings, установите Collision Complexity на Use Complex Collision as Simple и убедитесь, что Collision Preset установлен на BlockAll (рисунок 11.3).

Вы можете переопределить Collision Preset для каждого основного Actor с помощью панели Details; однако нельзя изменить свойство Collision Complexity. Можно сделать это

только в интерфейсе Static Mesh редактора или в матрице свойств.

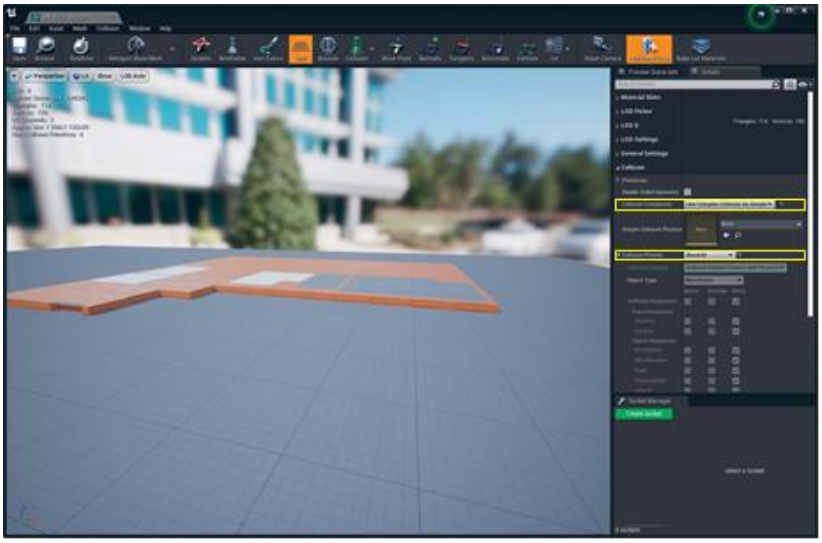

Рисунок 11.3 Настройка Floor Mesh для использования per-polygon collision

Упражнение 15. Bulk Editing с Property Matrix

Вы должны настроить collision complexity для каждого объекта Mesh, с которым хотите столкнуть вашего аватара. Это займет очень много времени, если вам придется открывать все ассеты один за другим.

К счастью, UE4 содержит функцию Bulk Edit.

Выберите все меши, которые хотите включить для коллизий в Content Browser. Нажмите правой кнопкой мыши на один из значков Asset для открытия контекстного меню.

В разделе Asset actions выберите Bulk Edit via Property Matrix (рисунок 11.4).

Матрица свойств показывает все общие свойства в измененной Details Panel. Отсюда вы можете установить Collision Complexity для всех ваших Meshes за раз (рисунок 11.5). Вы также можете отобразить свойства в виде столбцов в списке слева и таким образом визуально сравнить Assets и изменить отдельные Assets, как в электронной таблице.

Иногда свойства содержат разные названия в разных интерфейсах. Collision Preset является одним из таких. Для изменения этого свойства с помощью матрицы свойств

посмотрите на свойство под названием Collision Profile Name. Фильтрация свойств для collision помогает сузить список. Для настройки Collision Preset введите BlockAll в текстовое поле свойства Collision Profile Name.

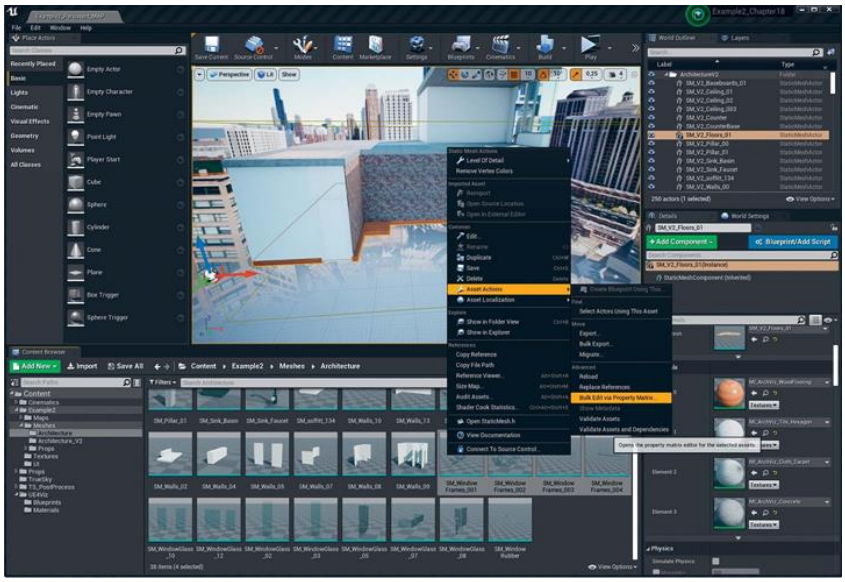

Рисунок 11.4 Выбор нескольких Assets в Content Browser и их одновременное редактирование с помощью матрицы свойств

Когда вы закончите редактирование ваших Assets, обязательно сохраните их, чтобы изменения записались на диск.

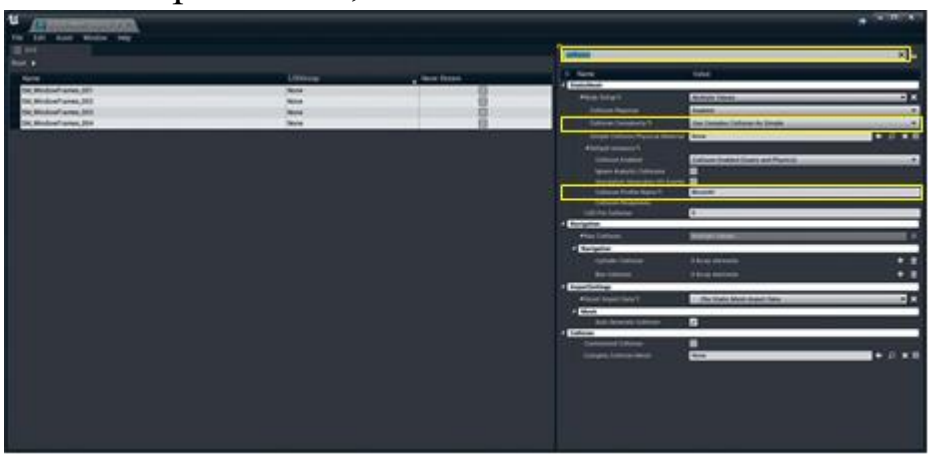

Рисунок 11.5 Использование матрицы свойств для настройки Collision Complexity для всех стен и полов одновременно

Визуализация столкновения

После того как вы настроили столкновения, проверьте вашу инфраструктуру коллизий, установив Perspective Viewport

для Player Collision View mode (рисунок 11.6). Это позволит вам увидеть инфраструктуру коллизий, которую физический движок UE4 использует для расчетов столкновений.

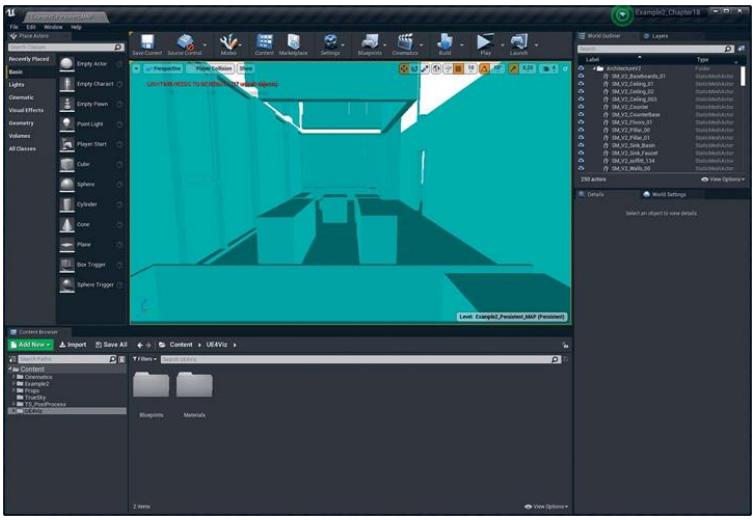

Рисунок 11.6 Player Collision View mode показывает per-polygon collision, успешно включенную для стен и полов.

11.4 Настройка Prop Collision

Настройка столкновений для реквизита немного сложнее. Предпочитительнее отключать коллизии на большинстве объектов Props в моих симуляциях, позволяя пользователю свободно проходить через и над большинством препятствий.

На рисунке 11.7 вы можете увидеть, что большая часть мелкого реквизита все еще имеет коллизии, что потенциально может затруднить перемещение автара в этом пространстве.

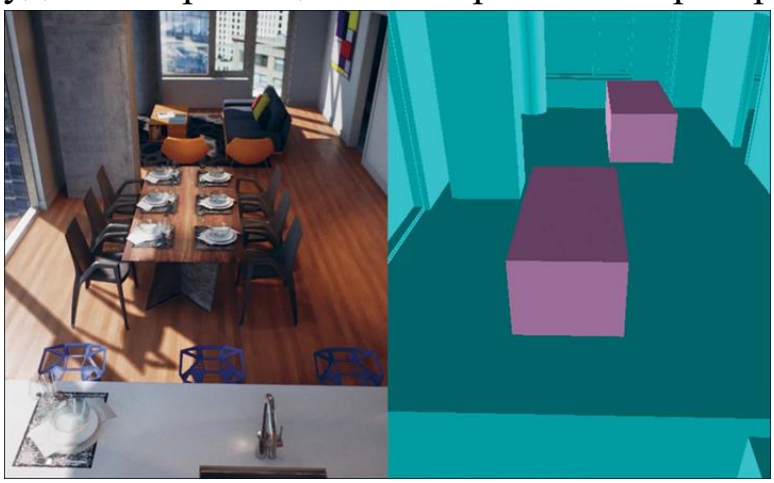

Рисунок 11.7 Окончательная настройка Collision, показывающая только стены, полы, окна и огромную мебель,

содержащую player collision enabled, позволяющая пользователю свободнее перемещаться по пространству.

Чтобы разрешить реквизиту иметь коллизию (важно для взаимодействий, таких, например, как нажатие мышью), измените Collision Preset в Static Mesh Editor (или Collision Profile Name в матрице свойств) на Ignore Only Pawn. Это позволит статическому мешу реагировать на все другие физические события, но не помешает перемещению пользователя. Вы также можете установить это свойство каждому актору через панель Details.

Теперь вы можете нажать Play и походить вокруг по уровню без страха навсегда упасть в пустоту. Стены и пол должны быть цельными, и на уровне должно быть легко ориентироваться.

Если вы застреваете или падаете, вам нужно посмотреть на настройку коллизий на уровне.

Включение курсора мыши

Поскольку вы хотите, чтобы это было управляемое мышью приложение, вам нужно видеть курсор. Откройте Player Controller и найдите группу Mouse Interface в свойствах по умолчанию класса (рисунок 11.8). Включите Show Mouse Cursor, Enable Click Events и Enable Mouse Over Events, которые позволят 3D Actors на сцене взаимодействовать с курсором мыши.

Если вы сделаете тест сейчас, вы обратите внимание на то, что можете вращать камерой, только когда удерживаете кнопки мыши. Отпускание кнопки отобразит курсор и отключит контроль вращения.

Вы также можете заметить, что ось вращения ощущается «выключенной». Камера повернет направо, когда вы ожидаете, что она повернет влево или вверх, когда вы хотите посмотреть вниз.

Это из-за того, что пользователи ожидают управления камерой. Когда пользователю не требуется удерживать кнопку для поворота, это привычно для горизонтальной оси, следовать

направлению мыши. Правый поворот — направо, левый поворот — налево.

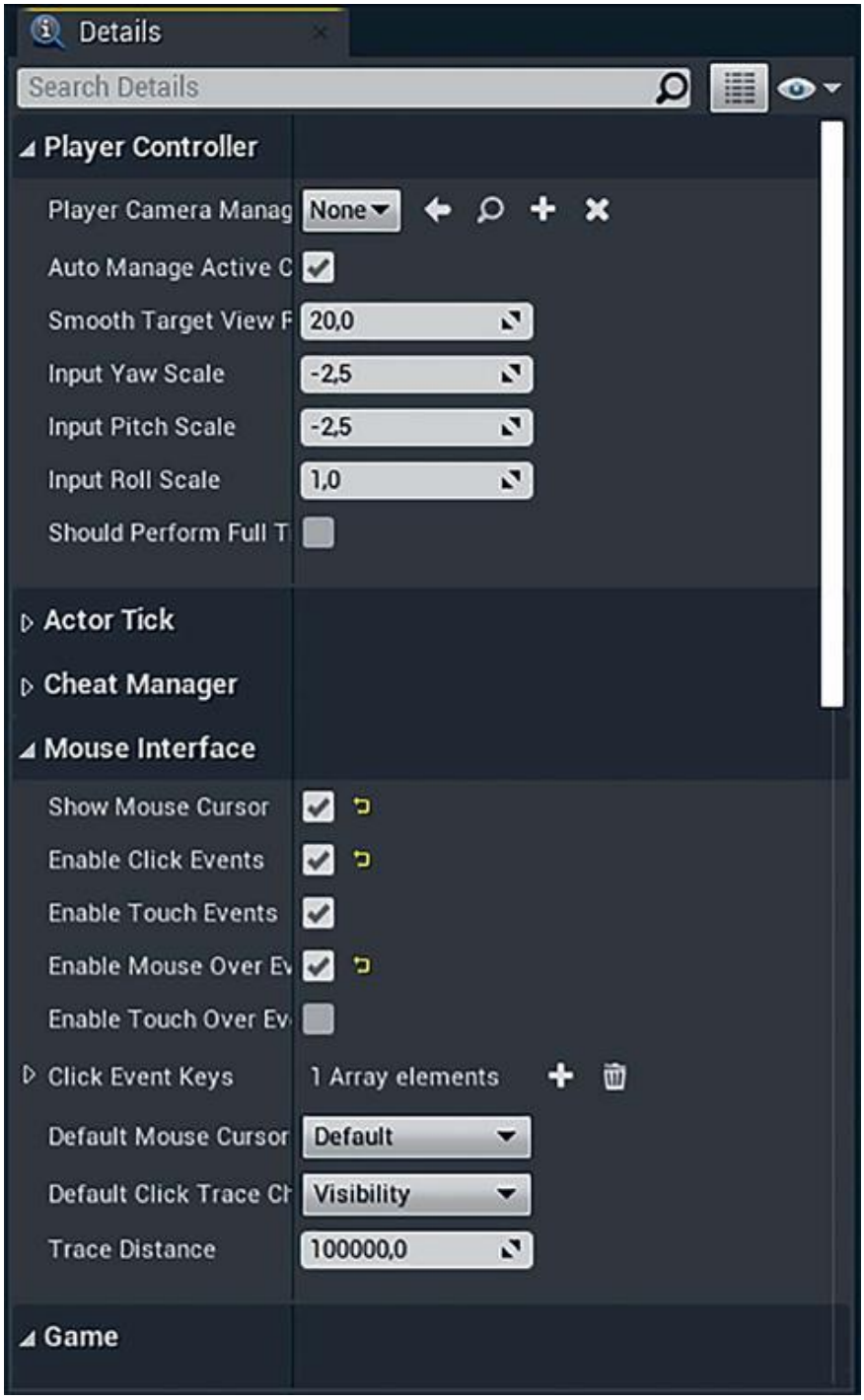

Рисунок 11.8 Настройки Player Controller для использования Mouse Cursor, Mouse Click, и Mouse Over events, а также Input Yaw и Pitch для компенсации ощущения инвертированного поворота

Однако, перетаскивание и отпускание (drag and drop) для поворота камеры ощущается естественно и выглядит как работа на тачскрине с привязкой курсора к точке в 3D-пространстве и поворотом камеры, как если бы пользователь использовал трекбол. Это означает, что перетаскивание мышью вправо должно поворачивать обзор влево, когда перетаскивание вниз должно поднять камеру.

Вы можете регулировать эти изменения в зависимости от нужд пользователя через значения Input Yaw и Input Pitch Scale в Player Controller (рисунок 11.8). Причина, по которой Yaw (горизонтальное вращение) быстрее, связана с человеческим восприятием и «правильным ощущением». Наличие симметричных осевых скоростей «ощущается» неправильно.

Стоит отметить, что интерактивность камеры действительно зависит от личных предпочтений. Возраст, опыт работы с различными технологиями и даже любимая игра людей сообщит им, как интерактивная камера должна вести себя.

11.5 Создание Post-Process Outlines

Чтобы было понятно, какой актор выбран, используется очертание, похожее на используемое в редакторе. Этот эффект недоступен во время работы, так как использует совершенно другую систему рендеринга, чем главный Viewport.

Взамен вы можете использовать post-process Material для рисования контура. Для образца слили собственный Marketplace content в этот проект, создал экземпляр материала и изменил его под свои потребности (рисунок 11.9).

Чтобы назначить post-process Material, вы должны сделать это в настройках Post- Process Volume, включив Blendables array.

Для добавления записи в массив задайте для нее Asset Reference и либо выберите материал постобработки из списка, либо перетащите его из Content Browser в свойство в панели Details.

Этот материал использует Custom Depth буфер для определения, какие объекты выбраны, а какие нет. Вы

устанавливаете Custom Depth для каждого актора в Levels Editor (рисунок 11.9).

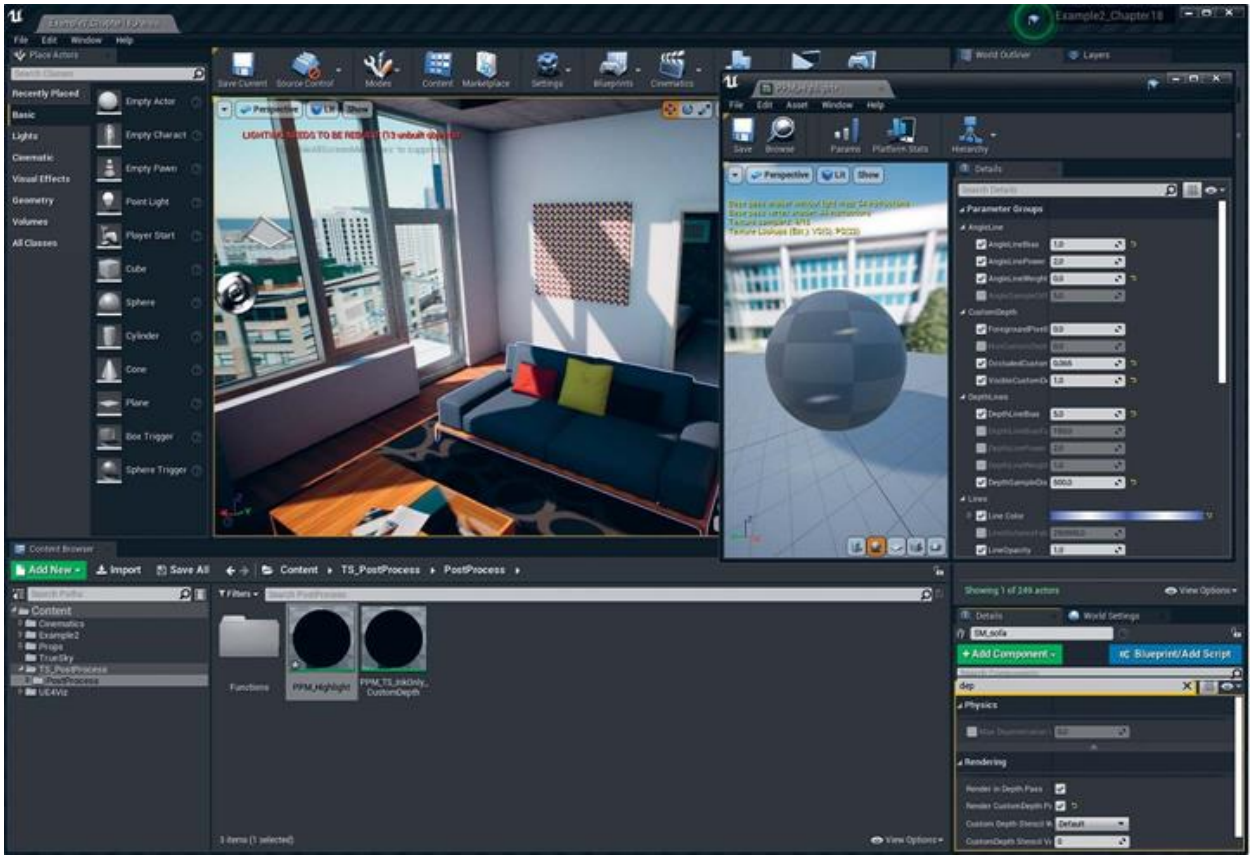

Рисунок 11.9 Тестирование Post-Process Material с помощью включения свойства Render Custom Depth для Static Mesh Actor дивана

Вам может потребоваться включить этот эффект в ваших Project Settings в разделе Rendering (рисунок 11.10).

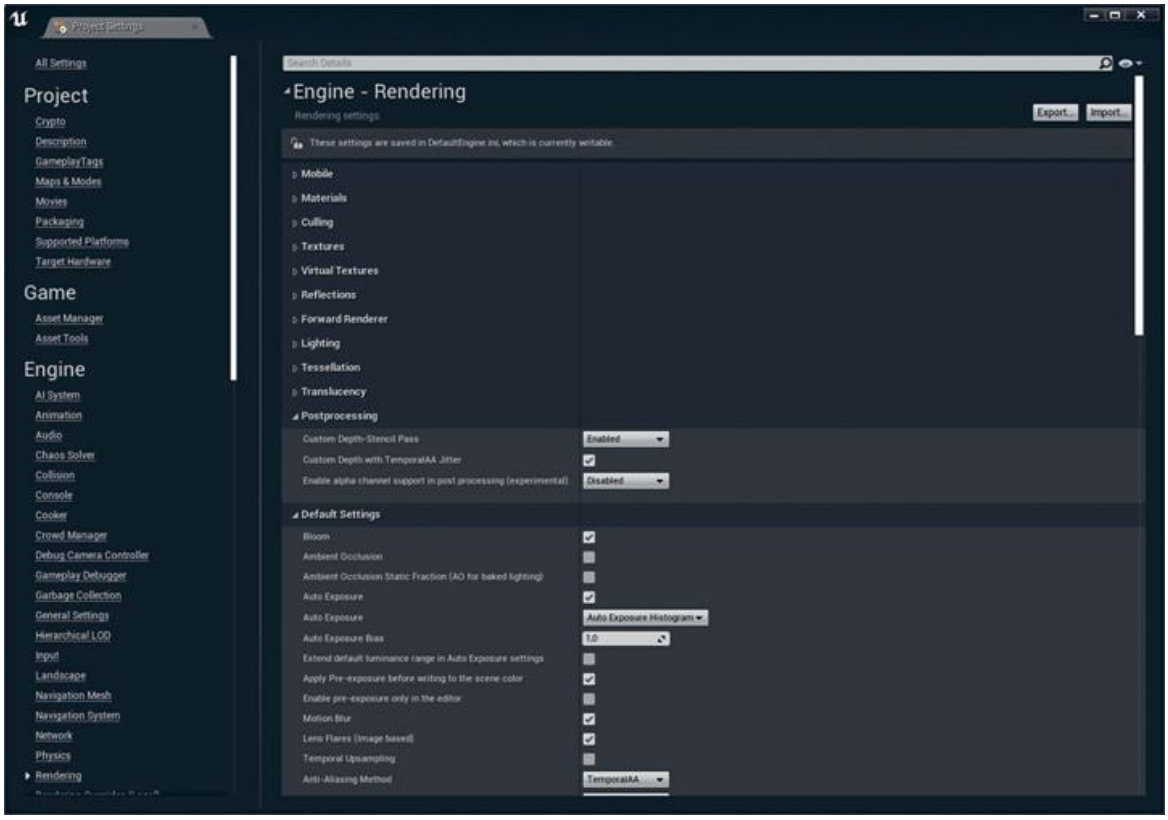

Рисунок 11.10 Включение Custom Depth буфера в диалоговом окне Project Settings

Заключение

Настройка ваших данных, позволяющая взаимодействовать с пользователем, является важным процессом. Установка коллизий гарантирует.

Включение курсора мыши открывает множество возможностей взаимодействия Player, включая разработку интерфейса в UMG и взаимодействие с внутриигровыми мешами с использованием событий ввода.

Теперь ваш проект готов, чтобы начать программировать интерактивность с помощью Blueprints.

12 БОЛЕЕ СЛОЖНЫЕ BLUEPRITNS: ВЗАИМОДЕЙСТВИЕ С UMG

Создание пользовательских интерфейсов — это вызов, с которым большинству профессионалов визуализации, вероятно, не приходилось сталкиваться раньше. Без правильных инструментов это может стать непростой задачей. UE4 представляет UMG (Unreal Motion Graphics), комплексное

решение, которое может быть использовано для разработки полноценных data-driven интерфейсов или простых кнопок переключения и логотипов, которые необходимы для создания отточенной и простой в использовании интерактивной визуализации.

Переключение данных

Следующая установленная цель этого проекта позволит пользователю переключаться на альтернативный вариант вашего уровня, сохраняя одно и то же положение камеры. Такое контекстное переключение — одно из самых мощных преимуществ интерактивной визуализации. Предоставление пользователю возможности сравнивать различные данные с одного и того же вида является отличным способом анализировать альтернативы и позволяет ему выбирать, где разместить камеру, тем самым сильно расширяя возможности.

UE4 содержит систему под названием Level Streaming, предназначенную для загрузки и выгрузки целых уровней во время работы. Она разработана, чтобы позволить играм иметь очень большие уровни, разбитые на секции, которые загружаются и выгружаются, когда пользователь перемещается по ним без экранов загрузки, прерывающих ход игры.

Вы изучите, как пользоваться этой системой для одновременной загрузки двух версий вашей карты и скрытия и отображения их сначала с помощью простого переключения клавишами для тестирования, а затем путем создания UMG UI для использования пользователями.

Чтобы сделать это, вы сначала должны разработать новый уровень, основанный на ваших новых данных, вместе с Lighting, Materials и Props. Вы будете использовать уже существующий уровень как основу, многократно используя ваши работы из предыдущих глав, сохраняя много времени, которое тратится на разработку.

После этого вы создадите на основе UMG (Unreal Motion Graphics) пользовательский интерфейс, который позволит пользователю просто нажимать кнопки для мгновенного

переключения между различными вариантами. Вы изучите, как создавать UMG Widget Blueprint и связывать его с Viewport и как он принимает пользовательский ввод и выдает команды для изменения игрового мира.

12.1 Создание вариаций уровней

Клиент предоставил вам альтернативную планировку пространства. В ней присутствует превосходный лофт над спальней (рисунок 12.1).

Это значительное изменение, которое повлияет на внешний вид, ощущение и освещение на уровне, поэтому лучше всего создать совершенно новый уровень, с собственными освещением и геометрией.

Упражнение 16. Создание вариаций уровней.

загрузите уровень в редактор, если еще не сделали это. Вы можете загружать уровни, перейдя в File > Open Level или найдя UMAP Asset в Content Browser и дважды нажав на него.

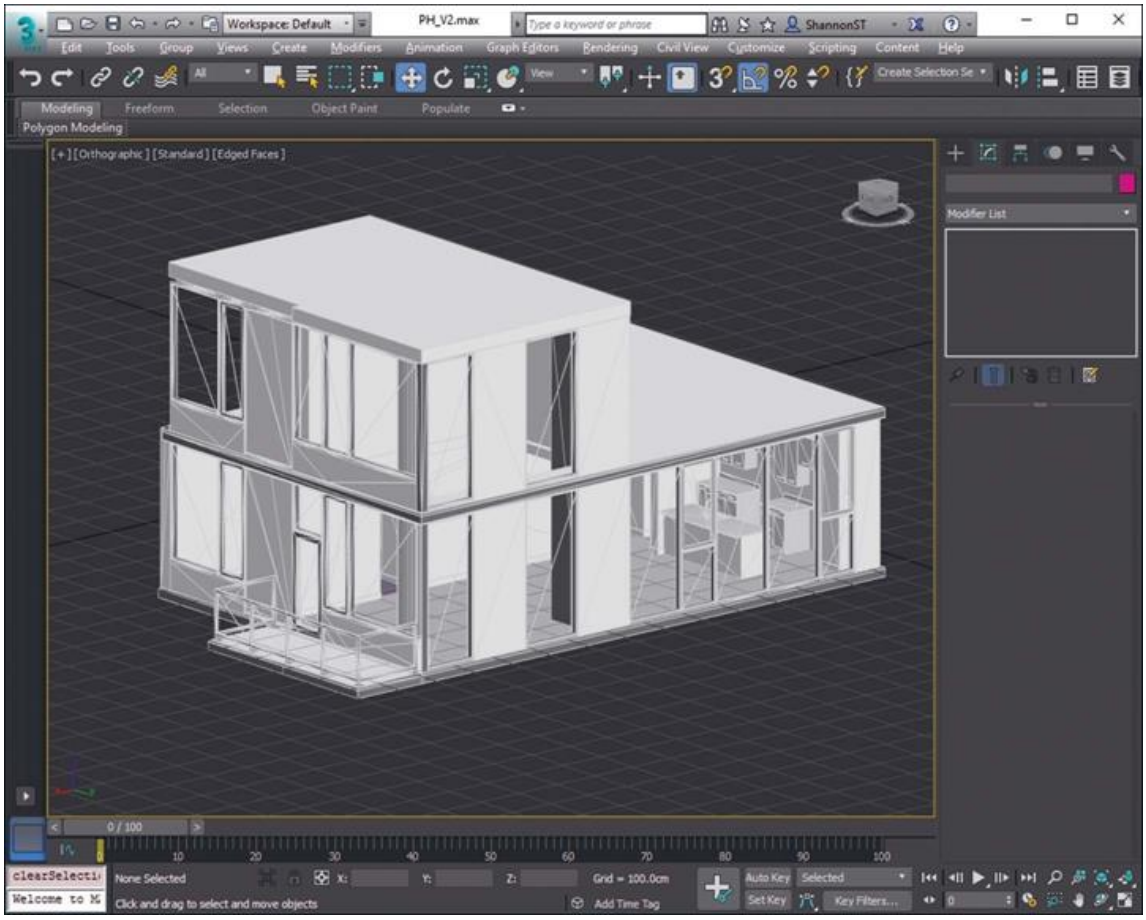

Рисунок 12.1 Обновленные данные в 3DS Max

Making a Copy with Save As

Создайте копию уровня, выбрав File > Save Current As, и затем назовите его как-ни- будь наглядно, например Example2_V2_MAP.

Так вы получите два уровня в вашем проекте: Example2_MAP и Example2_V2_MAP. На данном этапе уровни идентичны, за исключением названия.

12.2 Импорт новой архитектуры

Поскольку ваши уровни имеют много общего, включая расположение опор, освещение и другие элементы, вам нужно только поменять архитектурные меши. Вы также можете выбрать для замены только меши, которые изменяются, но для примера давайте создадим полностью уникальный уровень с полностью уникальной геометрией.

Как в главе 12 «Процесс работы с данными», приготовьте ваш контент в 3D-приложении. Примените UVW mapping для проверки на плохую геометрию и упорядочьте ваш контент, как вы делали ранее. Вам следует придумать новый вариант наименования для этих мешей, чтобы избежать возможных конфликтов.

После подготовки экспортируйте в формате FBX и импортируйте в UE4.

Когда вы импортируете FBX-файлы в UE4, вам следует переместить контент в новую папку, отдельно от предыдущих архитектурных мешей; это позволит поддерживать два набора данных раздельно, избегая конфликтов данных.

Как и раньше, импортируйте FBX-файлы в Content Browser используя предложенные параметры для статических архитектурных мешей — очень важно убедиться, что Auto Generate Collision установлен на false, Generate Lightmap UVs установлен на true, и для Transform Vertex to Absolute также стоит true (рисунок 12.5 в главе 12).

После импорта FBX-файлов не забудьте сохранить Static Mesh Assets, которые вы только что создали, прежде чем двигаться дальше.

12.3 Замена архитектурных мешей

В вашем новом уровне (Example2_V2_MAP) выберите все Architecture Static Mesh Actors на сцене. Нажмите кнопку Del или правой кнопкой мыши на акторах и выберите Delete для их удаления, оставив только Props, Lights и других Actors.

Для расположения новых архитектурных мешей перетащите их во Viewport и затем переустановите их позицию на 0,0,0 с помощью свойства Location в панели Details. Теперь вы обновили архитектуру.

Пока ваши недавно расположенные Static Mesh Actors все еще выбраны, самое время также систематизировать их в папку в World Outliner, чтобы вы могли в будущем легко выбрать их.

Настройка Lightmap Density

Теперь, когда вы заменили стены, полы и потолки, они вернулись к своему стандартному Lightmap-разрешению. Вы должны просмотреть и изменить во вновь размещенных мешах свойство Overridden Light Map Res, используя Lightmap Density Optimization View Mode.

Вы должны попытаться сделать так, чтобы все как можно лучше соответствовало плотности, которую вы установили на предыдущей карте. Для гарантии, что билд света согласуется между двумя уровнями.

Применение материалов

Ваши недавно импортированные меши, несомненно, будут либо лишены материалов, либо будут применены импортированные материалы по умолчанию. Потратьте время, чтобы применить материалы на первом уровне.

Включение Collision

Последнее, что вам осталось сделать, — это убедиться, что ваша Collision установлена правильно. Используйте Player

Collision View Mode, чтобы определить, для каких мешей нужно включить Collision.

Декорации (опционально)

Воспользуйтесь этой возможностью, чтобы внести столько изменений уровня, сколько захотите, — попробуйте различные Lighting, Materials, Props, назовите его. В этом примере Props, Lighting и Materials остались прежними, изменившись только в архитектуре. Это позволит пользователям сконцентрироваться на различиях, не отвлекаясь на другие изменения.

Билд освещение

Освещение хранится отдельно для каждого уровня. Таким образом у вас могут быть одинаковые ассеты для разных уровней с сильно различающимися настройками освещения и Lightmaps. Это позволяет запекать освещение для каждого уровня, пока используются одинаковые ссылки на ассеты.

Рисунок 12.2 Готовое освещение, построенное для новых данных

Преимущество использования метода Save As для создания нового уровня состоит в том, что он сохраняет все World Settings, которые вы применили ранее, включая настройки Lightmass. Это делает создание освещения действительно очень простым; вы просто должны иметь возможность установить Lighting Build Quality для какого хотите уровня и нажать кнопку Build.

После создания освещения у вас должен быть новый вариант вашего уровня с потрясающими сводчатыми потолками и соответствующими изменениями освещения (рисунок 12.2).

Сохраните уровень. Карты света и теней для уровня будут записаны (или, начиная с версии 4.15, в отдельный файл Build Data рядом с уровнем, который виден только в Content Browser или File Explorer).

Level Streaming

UE4 может загружать и выгружать целые уровни на лету. Это называется Level Streaming, и эта система может использоваться для перемещения больших наборов с помощью команд Blueprint.

Способ работы Level Streaming довольно прост. Сначала загружается уровень, который называется Persistent Level. Этот уровень часто очень прост или даже почти полностью пустой.

Сам этот уровень или объект, или актор внутри него (к примеру, Blueprint или Player Controller), может загрузить другой уровень или уровни в или из него. Вы также можете переключать видимость загруженных уровней в редакторе и в процессе исполнения, обеспечивая удобный способ быстрого переключения наборов данных.

Для настройки вашего Level Streaming сначала создайте Persistent Level, затем добавьте ваши две версии в качестве Streaming Levels с помощью интерфейса Levels (рисунок 12.3).

Создание нового уровня

Давайте сначала создадим новый, пустой уровень для вашего Persistent Level. Перейдите в File > New Level и выберите Empty Level из доступных вариантов.

Когда ваш новый, пустой уровень откроется, сохраните и назовите его (к примеру, Example2_Persistent_ MAP). Эта карта содержит только некоторый код Blueprint в Level Blueprint для управления переключения ваших Streaming Levels.

Доступ к Levels Interface

Editor предоставляет доступ к системе Level Streaming через список Levels. Вы може- те получить доступ к нему,

перейдя в Window > Level. Это откроет окно Levels (рисунок 12.3)

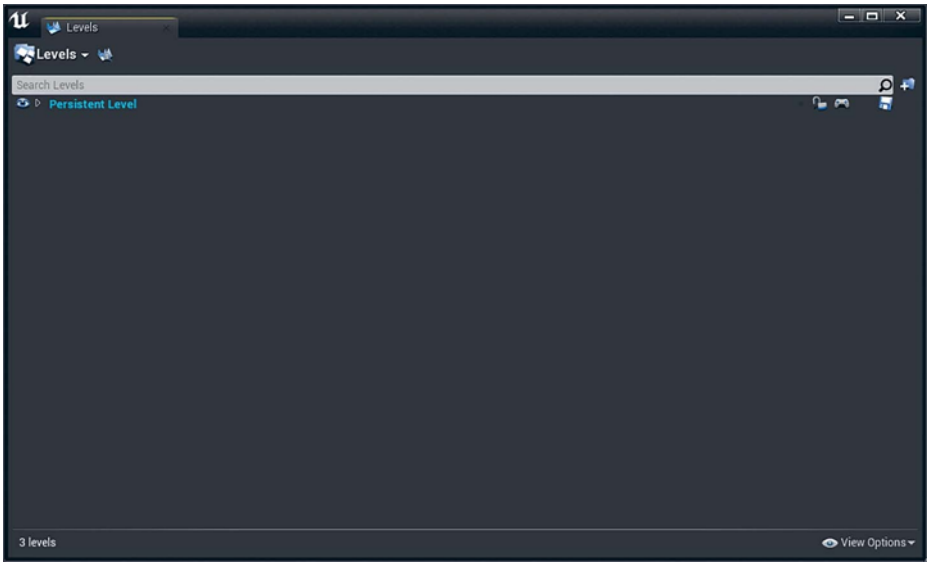

Рисунок 12.3 Окно Levels

Вы увидите текущий загруженный уровень в списке как Persistent Level.

Упражнение 17. Добавление Streaming Levels.

Нажмите кнопку Levels в левом верхнем углу окна Levels и выберите Add Existing из раскрывшегося меню. Выберите исходный уровень (Example2 _MAP), который вы сделали (рисунок 12.4). Вы также можете создать новую, пустую карту streaming или новую карту с выбранными Actors.

Карта загрузится в Viewport, и вы увидите ее в списке в окне Levels.

Проделайте это снова для добавления новой версии (Example2 _V2_MAP) в окно Levels. Теперь вы должны увидеть оба уровня в окне Levels, как и во Viewport.

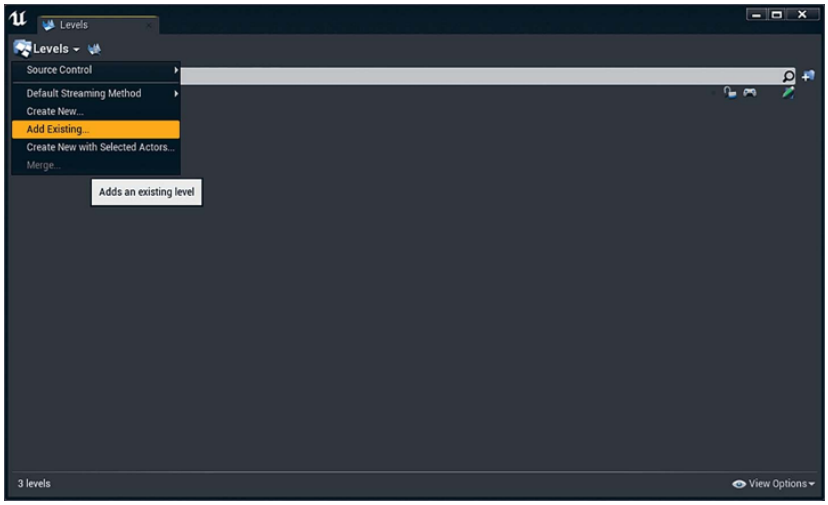

Рисунок 12.4 Добавление streaming Level с существующей картой

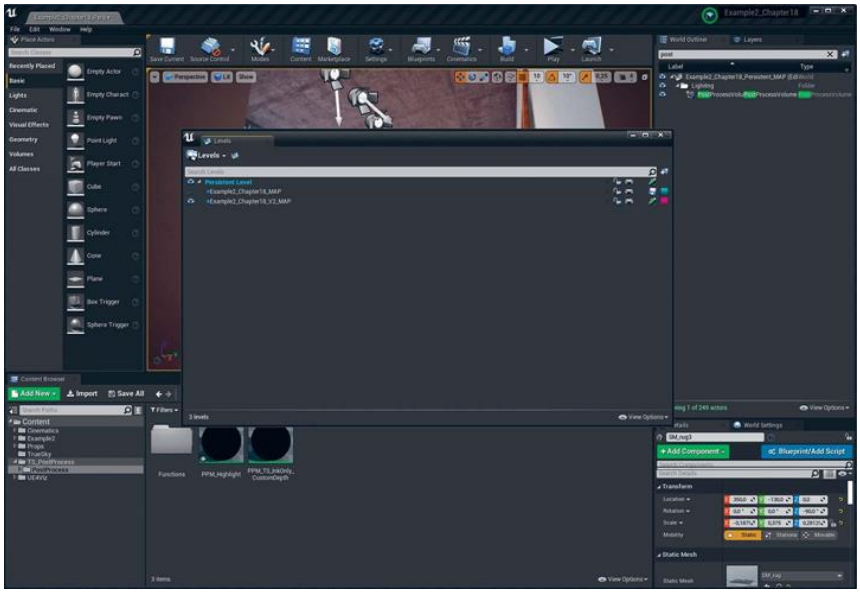

Рисунок 12.5 Обе версии уровня загружены в окно Levels

Вы можете скрывать и отображать уровни с помощью значка глаза для каждого уровня. Вы также можете сохранять уровни, открыв их Level Blueprints и переключив блокировку редактирования для предотвращения изменений. Даже если ваш проект не использует Level Streaming в процессе работы, его использование может быть хорошим способом разбить большую сцену на маленькие файлы или для организации всего (рисунок 12.5). Обратите внимание, что на рисунке Example2_MAP скрыт с помощью символа глаза, который теперь отображается как закрытая иконка глаза.

Помните, что переключатели обзора, установленные для каждого уровня в окне Levels, доступны только для редактора и что видимость уровня в процессе работы обрабатывается с помощью Blueprint.

Теперь вам нужно сохранить ваш Persistent Level, так как вы изменили его добавлением Streaming Levels с помощью интрейфейса Levels.

Использование Blueprints и постоянно загруженные уровни

Обратите внимание на маленькую синюю точку рядом с вашими двумя новыми уровнями в окне Levels. Это значит, что уровни загружаются с помощью Blueprints и что они не будут загружаться до тех пор, пока вы не сообщите им об этом с помощью Blueprints. Вы также можете выгружать, скрывать и отображать эти уровни в процессе работы.

Кроме того, можно установить уровни в состояние Always Loaded. Как следует из названия, эти уровни всегда загружаются и не могут быть скрыты или выгружены в процессе работы.

Уровни, которые необходимо всегда загружать, можно установить с помощью двой- ного нажатия по уровню в окне Levels, выбрав Change Streaming Method > Always Loaded. Эти уровни будут загружаться так, как если бы они были частью Persistent Level, когда игра запустилась.

Сохраняйте ваш Streaming Method для уровня как Blueprints, чтобы иметь возможность скрыть или отобразить их во время работы с помощью Blueprints.

Определение Player Start Actor

Даже несмотря на наличие Player Start Actors на двух ваших Streaming Levels, нужно определить одного для нового Persistent Level. Иногда это несколько сбивает с толку, так как вы можете ясно видеть Player Start Actors и выбрать их в Editor Viewport, но они не появятся здесь при начальной загрузке уровня. Это связано с тем, что Player Controller создается до того, как Streaming Level сможет загрузиться.

Выгрузка и загрузка Levels удаляет их из памяти и может занять некоторое время в процессе работы. Скрытие или отображение сохраняет Level загруженным, но просто включает рендеринг содержимого этого Level. Это происходит мгновенно.

Вы можете легко копировать и вставлять один за другим ваших Player Start Actors с одного из ваших Streaming Levels в Persistent Level. Выберите Player Start из Content Browser или Viewport и затем перейдите в Edit > Copy или выберите Edit > Copy из меню, открывающегося нажатием правой кнопки.

Вставка требует немного больше внимания. Когда у вас есть несколько уровней, загруженных в редактор, вам нужно определить, какой уровень является активным, прежде чем ставить Player Start.

Чтобы этот Player Start был расположен в Persistent Level, нажмите дважды по Persistent Level в окне Levels, сделав его активным. Вы сможете узнать активный слой, так как название станет синим.

Вставьте Player Start Actor в Persistent Level используя меню Edit, меню, вызванное правой кнопкой мыши, или просто используя Crtl/Cmd+V.

Вы также можете разместить Player Start используя Class Browser, как вы делали раньше, но вам все равно нужно убедиться, что активен правильный уровень.

Теперь у вас есть Player Start, размещенный в вашем Persistent Level. Однако если вы нажмете Play сейчас, то вы просто загрузитесь в темный пустой мир. Вам нужно настроить Level Blueprint для загрузки streaming Levels.

12.4 Настройка Level Blueprint

Теперь, когда ваши Streaming Levels настроены в редакторе, вы можете написать Blueprint логику для их переключения с помощью Level Blueprint.

Открытие Level Blueprint

Каждый уровень содержит собственный Blueprint Event Graph, называемый Level Blueprint. Этот Blueprint является

отличным способом сделать Level-specific действия; вещи, которые будут происходить только на одном уровне, основанном на особых акторах или событиях этого уровня. В качестве примера можно привести срабатывание открытия двери пользователем или настройку воспроизведения конкретной музыки, когда уровень открывается.

Любая функциональность, которая должна быть общей от уровня к уровню (например, движение пользователя) должна обрабатываться обычным Blueprint Class, так как вам не нужно дублировать и поддерживать этот код для каждого созданного Level18.

Во время работы все ваши уровни загружены в единый мир, и все могут получить доступ ко всему остальному. Однако в редакторе акторы и уровни могут получить доступ только к другим акторам на том же уровне.

Для доступа к Level Blueprint вы можете нажать на значок Gamepad в окне Levels, соответствующий уровень, который хотите редактировать, или нажать кнопку Blueprints из панели инструментов и выбрать Open Level Blueprint (рисунок 12.6).

Вы также можете получить доступ к Level Blueprints загруженных уровней и получить быстрый доступ к вашим Game Mode’s Classes.

Рисунок 12.6 Открытие Persistent Level Blueprint

Level Blueprint открывается в окне Blueprint Editor (рисунок 12.7). Этот редактор немного отличается от Blueprint Editors, который вы использовали до сих пор. В нем отсутствуют вкладки Viewport, Components и Construction Script. Level

Blueprints содержит только Event Graph, потому что они не могут иметь компоненты и не собираются, как классы Actor.

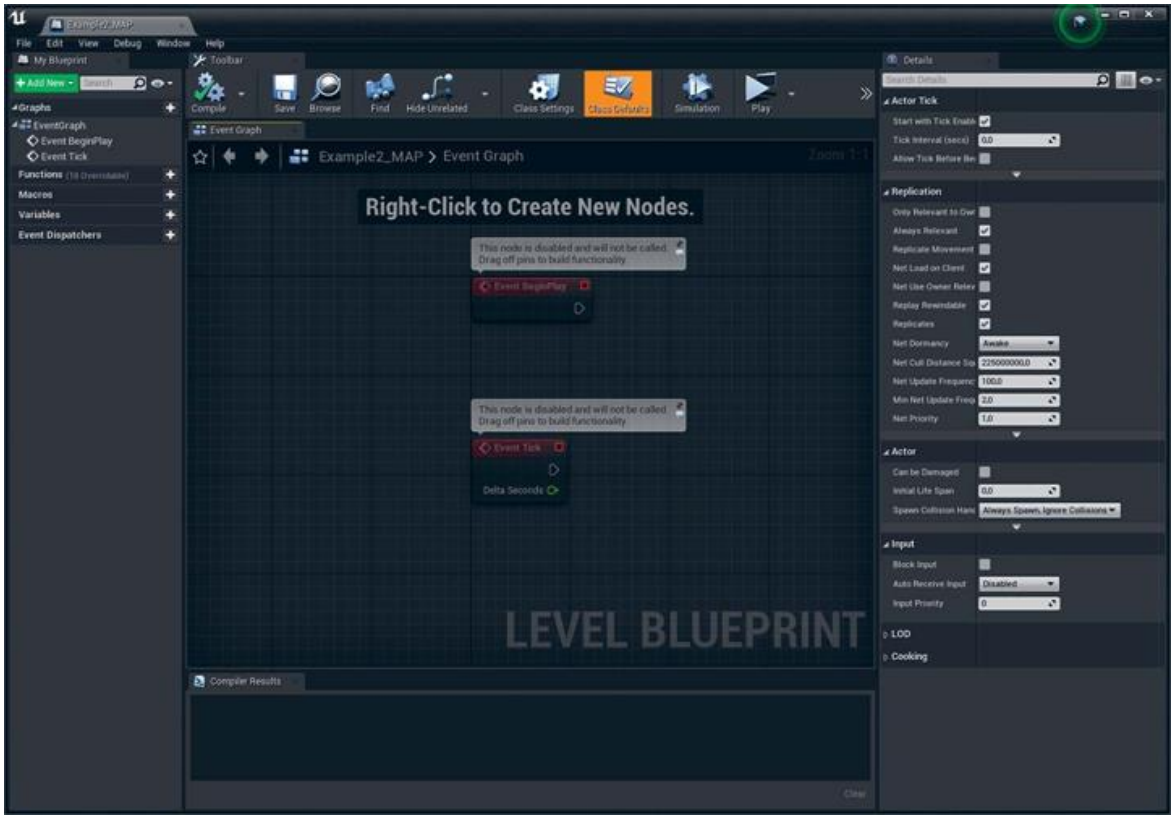

Рисунок 12.7 Level Blueprint, открытый в Blueprint Editor

Использование Events

Event является специальным узлом, который вызывается из кода игрового процесса. При вызове он запускает граф узла, соединенный с ним выходным контактом выполнения (белая стрелка). Эти события могут вызываться в ответ на различные события игрового процесса, такие как начало игры, перезапуск уровня или если пользователь нажал на определенную клавишу.

UE4 содержит множество предопределенных узлов событий. Два наиболее распространенных изображены на рисунке 12.7 выше: BeginPlay и Tick. Также помните, что ранее вы использовали InputAxis Events для настройки ввода вашего Player Controller.

BeginPlay

BeginPlay Event вызывается автоматически игрой один раз, когда уровень впервые загрузился, после загрузки и инициализации всех акторов и мировых объектов.

Здесь вы устанавливаете ваш начальный код Level Streaming. Как уже упоминалось, ваш Persistent Level пуст и должен содержаться в Levels streamed.

Для этого используйте функцию Load Stream Level. Вы можете увидеть на рисунке 12.8, что расположенно два этих узла. Как и в большинстве узлов, нажмите правой кнопкой мыши в Event Graph и найдите узел Load Stream Level.

Рисунок 12.8 Узлы Load Stream Level, соединенные с узлом BeginPlay Event

После размещения обоих узлов вам нужно заполнить настройки для каждого из них, чтобы они соответствовали рисунку 12.8. Example2_MAP загружается первым, используя узел Load Stream Level с Should Block on Load и Make Visible After Load, установленным на True. Затем загружается Example2_V2_MAP, но изначально с помощью параметра Make Visible After Load, установленного на False.

Вы должны установить для свойства Name точно такое же имя, как и для уровня в окне Levels. Также уровень уже должен быть загружен как Streaming Level в окне Levels для доступа к потоковой передаче Blueprints.

Should Block on Load вынуждает UE4 ожидать загрузки первого уровня, прежде чем продолжить выполнение любого игрового кода (это и есть block — он блокирует продолжение игры). Блокировка предотвращает падение вашего объекта Pawn в пустой Persistent Level до загрузки Streaming Level и связанной с ним коллизии.

Latent-функции

Обратите внимание на значок Clock на функциях Load Stream Level. Это указывает на то, что узел является Latent-

функцией. Latent-функции требуют времени для выполнения и продолжают рабочий процесс по графу событий (Event Graph), только когда их задача выполнена.

Загрузка уровней может занять несколько секунд. Большие уровни и медленные жесткие диски могут увеличить это время. Вот почему надо загружать их одновременно при начальной загрузке Persistent Level и переключать их видимость. Сохраняя уровни в памяти и просто меняя их видимость, мы можем избежать задержек и переключаться почти мгновенно.

Если нажать Play сейчас, ваше приложение должно загружать карту так же, как и раньше. Однако это, возможно, заняло немного больше времени по сравнению с тестированием с PIE до этого.

Это потому, что вы ждете, пока загрузятся уровни; даже карта V2, которая еще не отображается, требует времени для загрузки.

После загрузки уровней вы сможете перемещаться в пространстве, как это было в предыдущих главах, но на этот раз — с помощью Level Streaming.

Настройка Level Blueprint

Теперь, когда у вас есть загруженные карты, давайте заставим их переключаться. Это получится всего с несколькими узлами в вашем Level Blueprint. Вы также увидите, как создавать простые сочетания клавиш для тестирования переключения уровня, прежде чем приступить к разработке UMG-интерфейса.

12.5 Создание Custom Events

Вы можете создать собственный узел Custom Event, который можно вызвать в любой момент в Blueprint из другого Blueprint в вашем мире, тем самым предоставляя возможность для организации Event Graphs и позволяя Blueprints взаимодействовать друг с другом.

Вам требуется несколько Custom Events, которые могут переключать видимость уровней. Вы будете использовать эти события для тестирования переключения уровней и затем

используете тот же Events позже, когда разработаете UMG-интерфейс.

Двойным нажатием по заднему плану Event Graph создайте в Persistent Level Blueprint узел Custom Event и выберите Add Custom Event из контекстного меню (рисунок 12.9).

Когда вы создаете Event, присвойте ему уникальное имя и нажмите Enter для подтверждения.

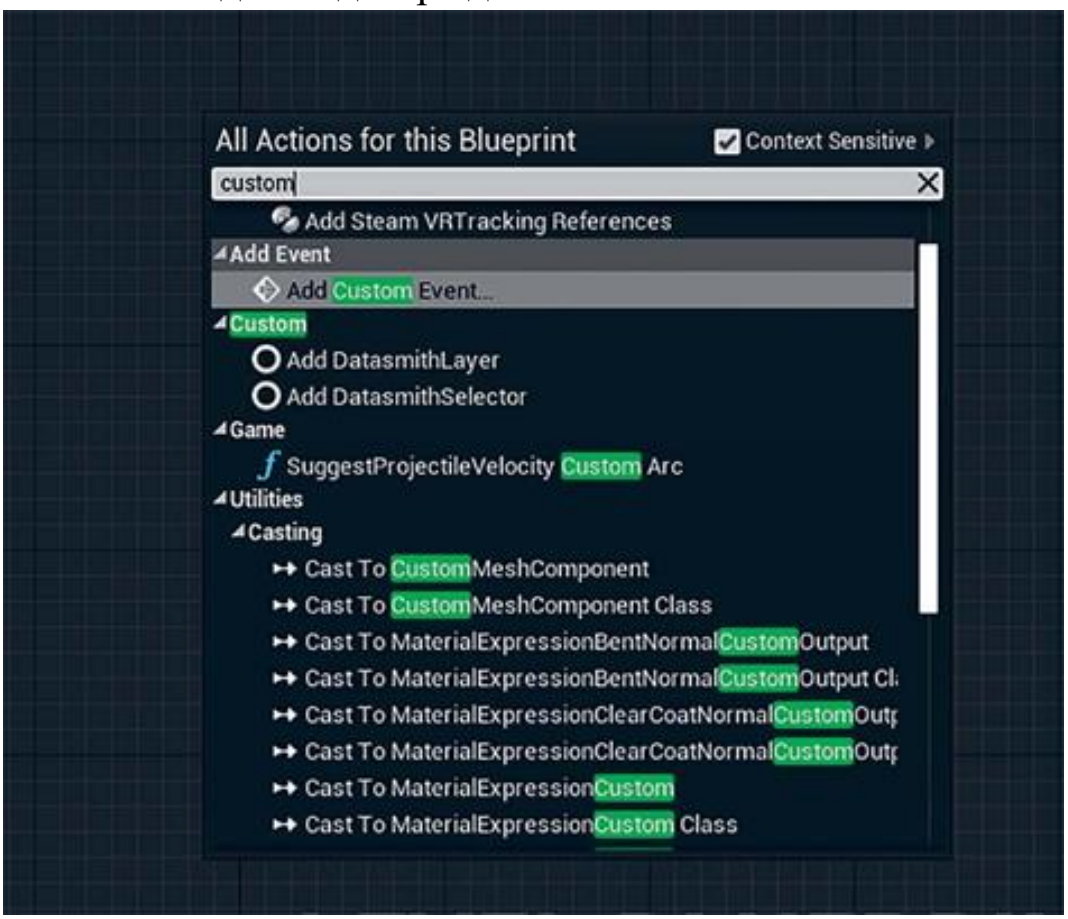

Рисунок 12.9 Добавление Custom Event для Level Blueprint

Вам требуется три события для работы системы переключения, поэтому разместите три узла Custom Event, назвав первый ShowVersion1, второй ShowVersion2 и третий — ToggleVersions. Ваш Event Graph должен выглядеть так, как показано на рисунке 12.10.

Рисунок 12.10 Три узла Custom Event, добавленные в Event Graph

Настройка Show Versions Events

Функции Show Version 1 и Show Version 2 просто устанавливают видимость для Streaming Levels, позволяя переключаться между ними.

Настройте Event Graph, как показано на рисунке 12.11.

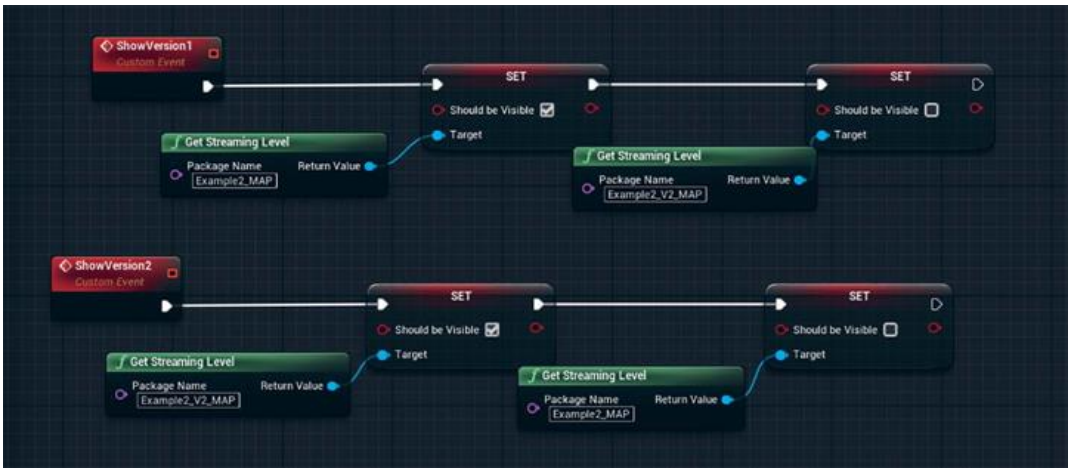

Рисунок 12.11 ShowVersion1 и ShowVersion2 custom Events и их графы выполнения

Эти два почти зеркальных изображения графов выполнения скрывают один уровень, пока показывают другой. Функция Get Streaming Level возвращает ссылку на уровень, определенный в свойстве Package Name. Вы должны вручную ввести это имя, и оно должно в точности совпадать со Streaming Levels.

Для доступа к свойству Set Should be Visible для уровня протяните соединение из синего контакта Return Value узла функции Get Streaming Level в Event Graph и отпустите для доступа контекстному меню для Streaming Levels, затем найдите Set Visible.

Настройте ваш первый не скрытый уровень так, чтобы скрывать другие. Вы увидите, что название уровня объявлено в Package Name узла Get Streaming Level.

Упражнение 18. Toggle Versions Event.

Toggle Versions Event переключает между двумя вариантами при каждом вызове (рисунок 12.12). Узел Flip Flop специально разработан для этого, а еще он смешно читается.

При первом вызове Flip Flop он вызовет только вывод A. В следующий раз он вызовет только выход B, затем A и так далее, чередуясь между вызовами двух Custom Events, которые вы создали ранее.

Для создания вызова Custom Event нажмите правой кнопкой мыши в Event Graph и найдите имя, которое дали вашему Custom Event, когда создали его. Чтобы убедиться, что ваш Custom Events находится в списке, вы должны скомпилировать и сохранить Blueprint.

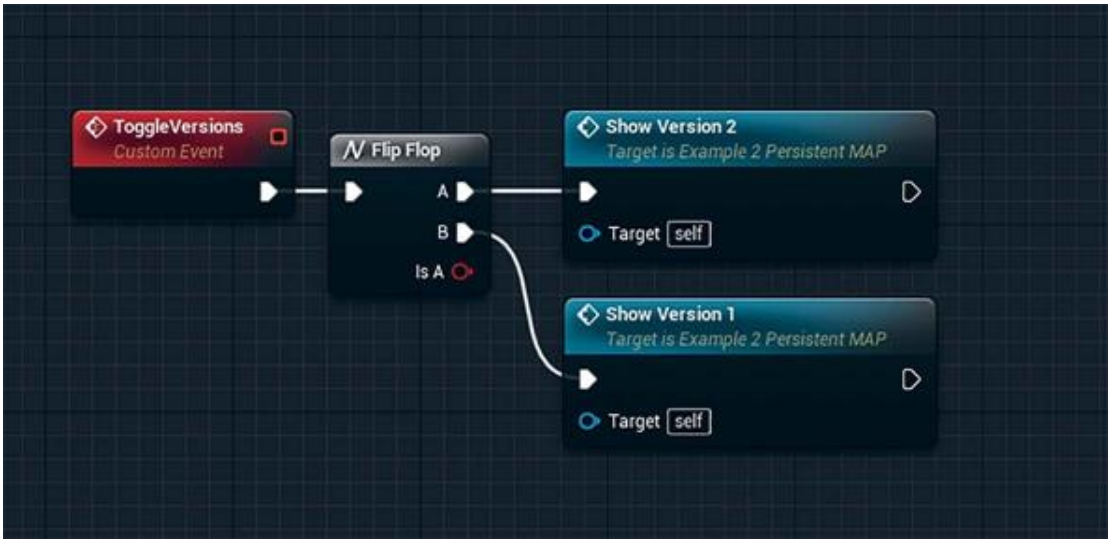

Рисунок 12.12 ToggleVersions и узел Flip Flop для переключения между двумя Custom Events

Время тестов

Теперь, когда вы все настроили, потратьте время на тестирование, прежде чем приступать к UMG-интерфейсу. Как вы можете проверить свою работу сейчас без UMG-интерфейса? Проще всего — создайте сочетание клавиш в Level Blueprint.

Создание шорткатов

Level Blueprints умеет перехватывать входные события от пользователя так же, как Player Controller. Вы можете использовать это, чтобы легко настроить сочетания клавиш для вызова Toggle Versions Event.

В вызванном нажатием правой кнопки мыши контекстном меню найдите Input L (рисунок 12.13) и выберите L из списка Keyboard Events.

Рисунок 12.13 Создание Input Event для клавиши L

Затем просто соедините вызов ToggleVersions Event с контактом выполнения Pressed в L Keyboard Input Event, как показано на рисунке 12.14.

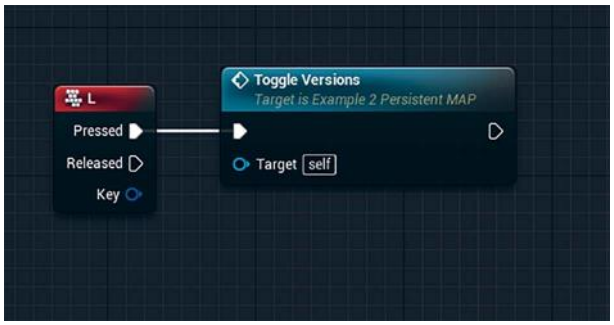

Рисунок 12.14 Абсурдно сложный код сочетания клавиш

Компиляция и сохранение

Теперь вы должны скомпилировать Level Blueprint, убедившись, что в нем нет ошибок или предупреждений, затем сохраните Persistent Level для сохранения кода, который написали.

Нажатие Play

Теперь, когда вы нажимаете Play, вам должно быть доступно нажатие кнопки L на клавиатуре, и ваш уровень немедленно переключится между двумя версиями.

Вы заметите, какое существенное изменение вносят дополнительные окна для всей сцены, даже в коридорах, далеко от основных изменений. Такая чрезвычайно мощная возможность переключения опций в контенте, как эта, дает пользователь личный опыт.

Однако не ждите, что все знают, что нужно нажать L, чтобы эти изменения произошли. Вам нужно предоставить пользовательский интерфейс для пользователя, который даст ему понятные опции. Для этого примените встроенную в Unreal систему пользовательского интерфейса UMG.

12.6 Unreal Motion Graphics (UMG)

Unreal Motion Graphics UI Designer (UMG) является инструментом визуализации UI, который вы можете использовать для создания игровых элементов, например, меню, названий и кнопок. UMG аппаратно-ускоренный, современный и не зависящий от платформы, что означает, что он работает быстро, отлично выглядит и может быть применен на любой платформе, поддерживающей UE4. Вы можете создать один интерфейс для использования на чем угодно, от PC и Mac до Nintendo Switch и всем, что между ними.

Использование Widgets

UMG полагается на виджеты (Widgets), готовые элементы, которые вы можете использовать для создания вашего интерфейса. Предопределенные виджеты доступны для большинства элементов UI, включая кнопки, ползунки, выпадающие списки и текстовые метки, а также виджеты, помогающие организовать и упорядочить другие виджеты в UI.

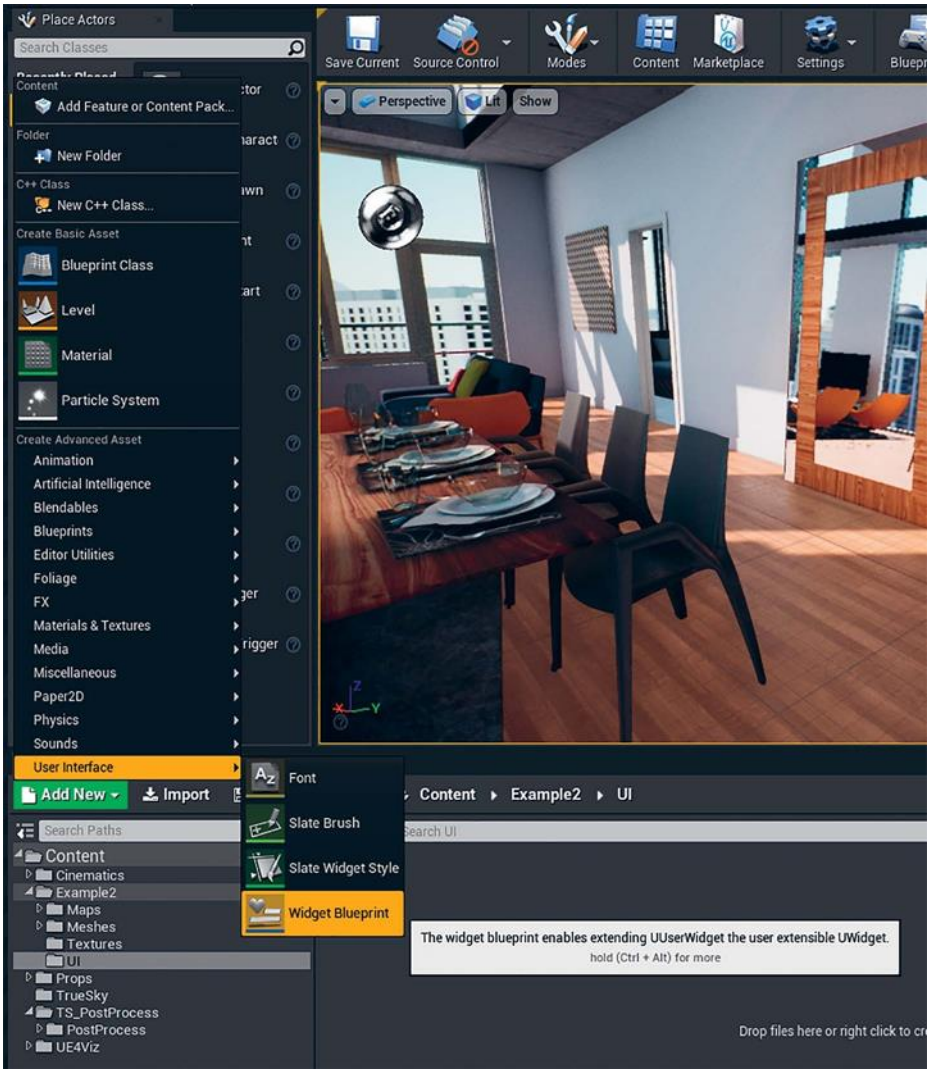

Рисунок 12.15 Создание Widget Blueprint

Виджеты собраны в Widget Blueprint, специальном Blueprint Class с настраиваемым Editor.

Вы создадите Widget Blueprints, как и большинство UE4 Classes, в Content Browser.

В меню Add New перейдите в User Interface > Widget Blueprint (рисунок 12.15). Назовите ваш новый Widget UI_Example2_HUD. HUD означает Heads up Display и обычно называется интерфейсом, который всегда показывается во время игры. Другими распространенными примерами UI могут быть главное меню или меню паузы.

Теперь нажмите дважды по недавно созданному Widget Blueprint, чтобы открыть его для редактирования (рисунок 12.16).

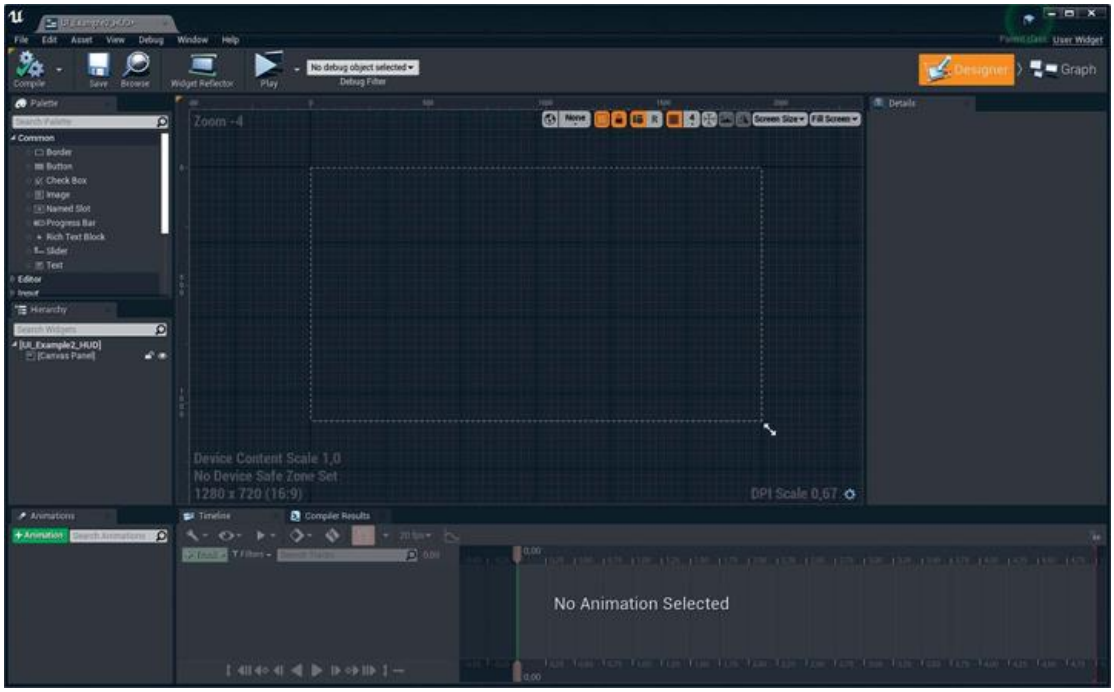

Рисунок 12.16 Окно Widget Blueprint Editor

Этот редактор состоит из двух основных разделов: вкладки Designer, которая позволяет вам визуально настраивать элементы UI, и вкладки Graph, где вы можете добавить функциональность для ваших UIs. В правом верхнем углу вы можете увидеть вкладки Designer и Graph. Нажмите на них, чтобы переключиться между двумя режимами интерфейса. В центре Stage, а слева Palette со всеми виджетами, доступными создания вашего UI. Ниже находится Hierarchy, которая показывает расположение Widgets во вложенном списке. Внизу находится Animation List и Timeline. Вы можете использовать их для разработки ключевой анимации для ваших элементов UI. Наконец, справа находится Details Panel. Как и все остальные панели Details в UE4, она контекстная и отображает детали выбранного в данный момент виджета.

Horizontal Box

Для этого UI вам нужны две кнопки, позволяющие переключаться между вашими вариантами. Вы наверняка хотите, чтобы кнопки были аккуратно расположены вдоль нижней части экрана с равномерными интервалами, чтобы все смотрелось чисто и профессионально.

UE4 поставляется с виджетом, который помогает с этим: Horizontal Box Widget. Он может содержать несколько дочерних виджетов, вложенных в него, равномерно распределяя каждый по горизонтали.

Найдите Horizontal Box Widget, указанный в окне Palette, под группой Panel. Перетащите его в Stage. Виджет появится в Stage и в списке Hierarchy.

Ваш виджет, вероятно, не попал в действительно правильное место, таким образом, вам нужно передвинуть его в нижнюю часть экрана, куда вы хотите. Вы можете просто перетащить его, но это не лучшая практика.

UMG поддерживает произвольное разрешение и масштабирование, поэтому он может быть использован на достаточно большом количестве устройств с любыми экранами.

Рисунок 12.17 Настройки Widget Anchor

Одним из способов, применяемых для работы UMG, является использование Anchors для Widgets. Anchor гарантирует, что Widget всегда привязан к относительной части экрана: стороне, углу или центру.

В данном случае прикрепите его к нижнему центру, нажав на раскрывающийся список Anchors в панели Details (рисунок 12.17).

Когда вы сделаете это, то не увидите больших изменений в Viewport, потому что UE4 пытается регулировать все свойства расположения так, чтобы Widget оставался в том же месте. Вы можете изменить эти свойства, чтобы получить Horizontal Box Widget там, где хотите, и такого размера, который хотите (рисунок 12.18).

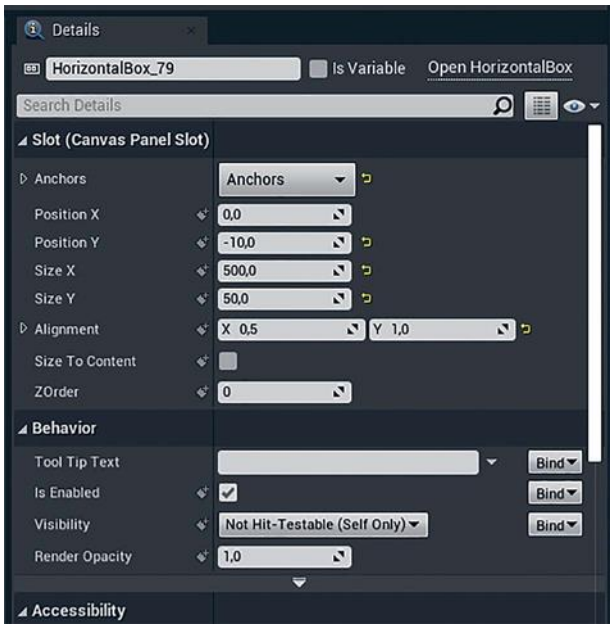

Рисунок 12.18 Настройка свойств Slot для Horizontal Box Widget's

Упражнение 19. Horizontal Box Widget

1. Установите Position X на 0 и Position Y на –10. Это центр Widget, но сдвинутый на 10 пикселей от нижней части экрана.

2. Установите Size X на 500 и Size Y на 50 для определения размера поля.

3. Установите Alignment. Это в основном точка опоры смещения. Если вы установите ее на 0,0, Widget будет перемещен к верхнему левому углу, а при 1,1 будет перемещен к нижнему правому углу. Alignment установленный на 0,5; 1,0 переместит опорную точку Widget к центру низа.

Кнопки

Просто перетащите два Button Widgets из Palette в Horizontal Box Widget. Вы можете сделать это либо в Stage, либо в окне Hierarchy (рисунок 12.19). Перетаскивание в окно Hierarchy может очень помочь, когда ваши UI Widget становятся сложными.

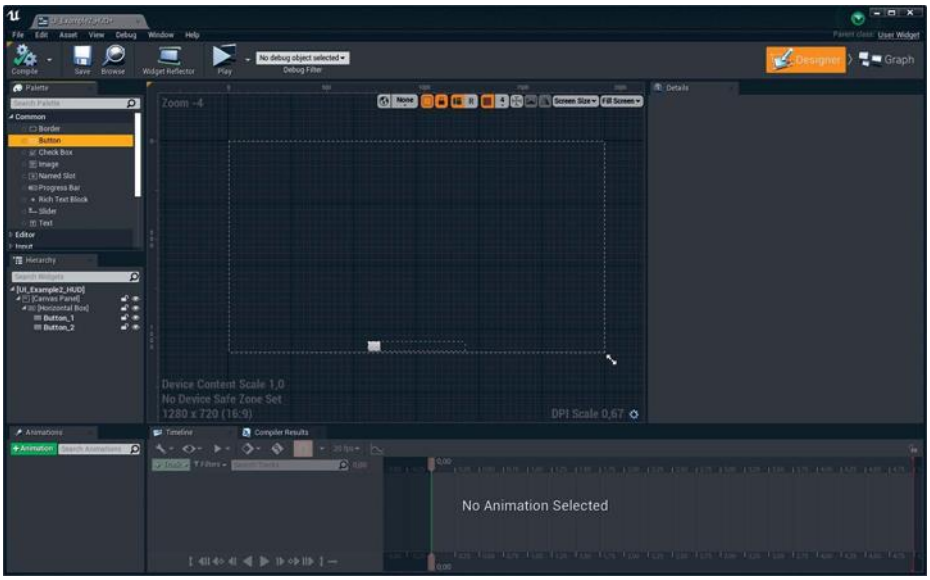

Рисунок 12.19 Кнопки в Horizontal Box Widget выглядят неидеально

Вам нужно немного стилизовать и настроить кнопки, чтобы они выглядели и вели себя правильно.

Выбирайте кнопки одну за другой и настройте их следующим образом (рисунок 12.20).

1. Главное — вы должны дать каждой вашей кнопке уникальное название. Если вы этого не сделаете, попытка получить доступ к Widgets из Blueprints может стать проблемой.
2. Установите для Padding 10 пикселей слева и справа. Вам, возможно, потребуется развернуть параметр Padding, нажав по стрелке рядом с ним, чтобы отобразить отдельные параметры.
3. Установите Size для Fill в значение 1,0. Это заставит кнопки заполнять любую доступную площадь, а не пытаться быть как можно меньше (Auto).

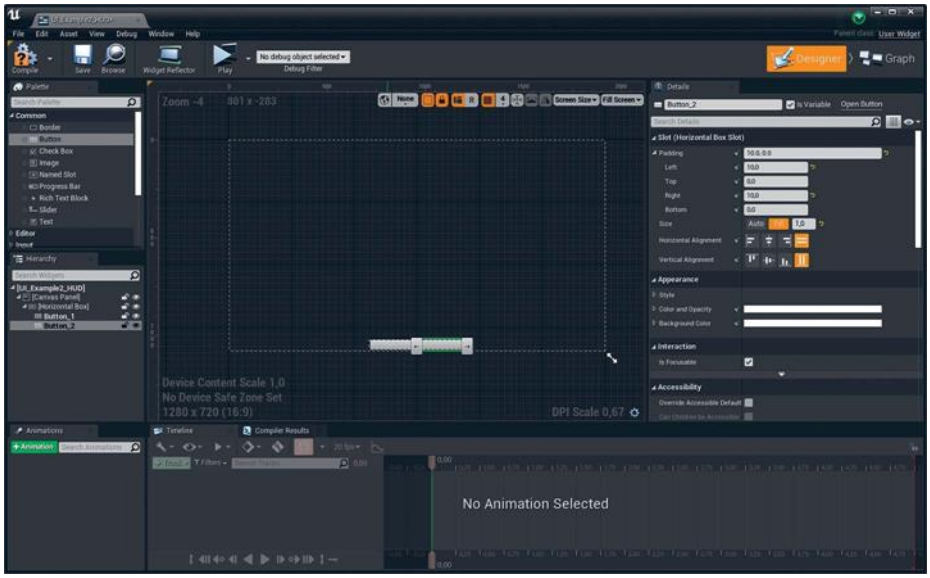

Рисунок 12.20 Настройка свойств Slot для Button Widget's

Метки

Вам нужны некоторые метки (Labels) для ваших кнопок. К счастью, Button Widget Class может иметь один дочерний класс — Text Widget, который идеально подходит для использования в качестве метки.

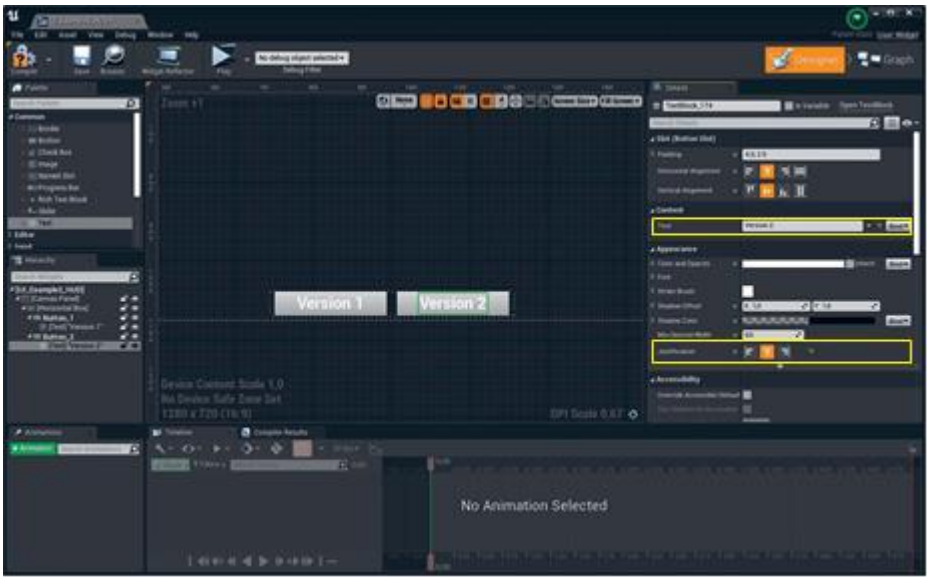

Рисунок 12.21 Добавление Labels

Найдите Text Widget в Palette и перетащите его в каждую из ваших кнопок. Установите в Label свойство Text для чтения Version 1 и Version 2. Вы также, возможно, захотите добавить небольшую тень для текста, чтобы сделать его более разборчивым (рисунок 12.21).

Самое время сохранить вашу работу, если вы еще не этого сделали. Ваш интерфейс готов, и вам не надо добавлять какой-либо код непосредственно в графе этого Widget Blueprint, так как вы будете обрабатывать все это в Level Blueprint.

Level Blueprint

У вас есть все части, необходимые для работы этой системы; теперь вам осталось только собрать их. Всю финальную сборку вы сделаете в Level Blueprint.

Откройте Level Blueprint, нажав кнопку Blueprints в Toolbar, и выберите Open Level Blueprint.

Наиболее очевидный вопрос на данном этапе: как получить UI, который вы только что создали, и отобразить его? Для этого в UE4 присоедините ваш Widget Blueprint к Viewport пользователя с помощью Blueprints, как показано на рисунке 12.22.

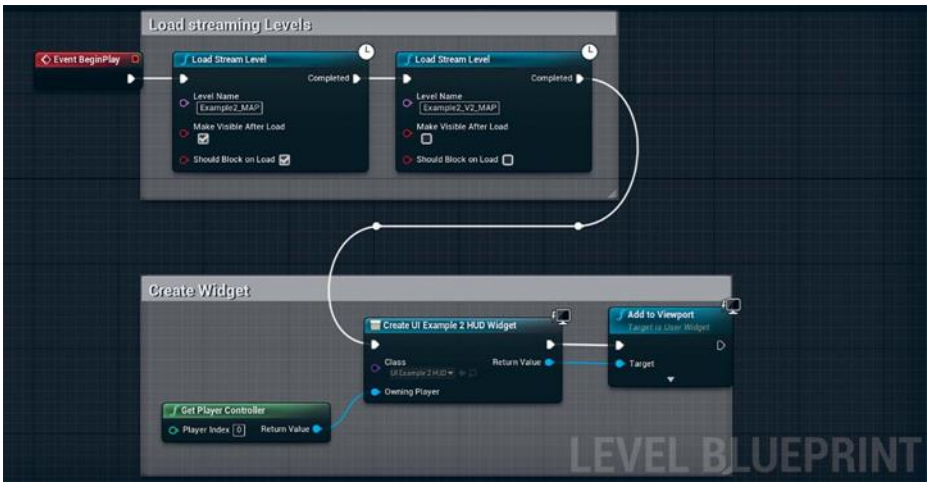

Рисунок 12.22 Создание экземпляра вашего Widget Blueprint и добавление его к Viewport

Лучшим местом, чтобы сделать это, является Begin Play Event, так как вы наверняка хотите, чтобы HUD отображался при загрузке уровня. Добавьте этот код после 19. Обратите внимание на узлы Redirect в белом потоке исполнения. Они позволяют программисту сделать узлы графов более читабельными, определяя положение исполнительных линий. Для добавления одного дважды нажмите на соединение.

Завершения загрузки вашего уровня, таким образом вы не будете показывать UI пользователю прежде, чем тот сможет нажать на него.

Следующие разделы проведут вас через соединение всех компонентов вместе.

Упражнение 20. Создание Widget

Первым шагом является создание объекта Widget, который загружает класс виджета с диска и создает его экземпляр в памяти (вы, конечно, можете создать больше одного экземпляра любого Widget, все они являются экземплярами одного и того же Widget Blueprint Class).

Как обычно, нажмите правой кнопкой мыши в Event Graph и найдите Create Widget для создания этого узла.

После размещения узла Create Widget вам нужно сделать две вещи.

1. Назначить класс, который вы хотите создать; в данном случае UI_Example2_HUD, выбрав из раскрывающегося списка Class.
2. Предоставьте этой функции ссылку на ваш Player Controller. Просто нажмите правой кнопкой мыши в Viewport, найдите Get Player Controller из списка и затем соедините его с контактом Owning Player.

Вам необходимо это сделать, так как в многопользовательской среде может быть больше одного Player Controller, и вам нужно знать, кому принадлежит HUD. В данном случае у вас есть только один, поэтому вы можете просто ссылаться на первый доступный Player Controller.

Добавление Widget во Viewport

Чтобы прикрепить Widget к Viewport и быть видимым и не взаимодействующим, вам нужно добавить его в Viewport.

Для доступа к узлу Add to Viewport перетащите соединение из Return Value узла Create Widget (который теперь должен читаться как Create UI Example 2 HUD Widget) и отпустите в Graph Editor, который откроет контекстное меню для вашего Widget Class. Найдите узел Add to Viewport и добавьте его в

граф. Синее соединение автоматически подключится к контакту Target недавно созданной функции Add to Viewport.

А теперь пора нажать на кнопку Compile и, если ошибок нет, сохраните ваш Level Blueprint.

Если вы запустите игру сейчас, вы увидите, что ваши две кнопки отображаются внизу экрана. Если вы щелкните по ним, то ничего не произойдет. К счастью, вы наконец-то настроили функции переключения; вам осталось только установить вызов этих функций при нажатии кнопок.

Event Binding

Мощной особенностью UE4 является способность для одного Blueprint привязываться к Event в другом. Это позволяет одному Blueprint обрабатывать несколько взаимодействий или Events с помощью одного унифицированного кода.

Для обнаружения событий вы сначала должны получить ссылку на кнопки, которые сделали (рисунок 12.23). Это легко, потому что метод Create Widget возвращает ссылку на Object, который создал. Таким образом, перетащите соединение из Return Value в graph для доступа к контекстному меню вашего Widget Class.

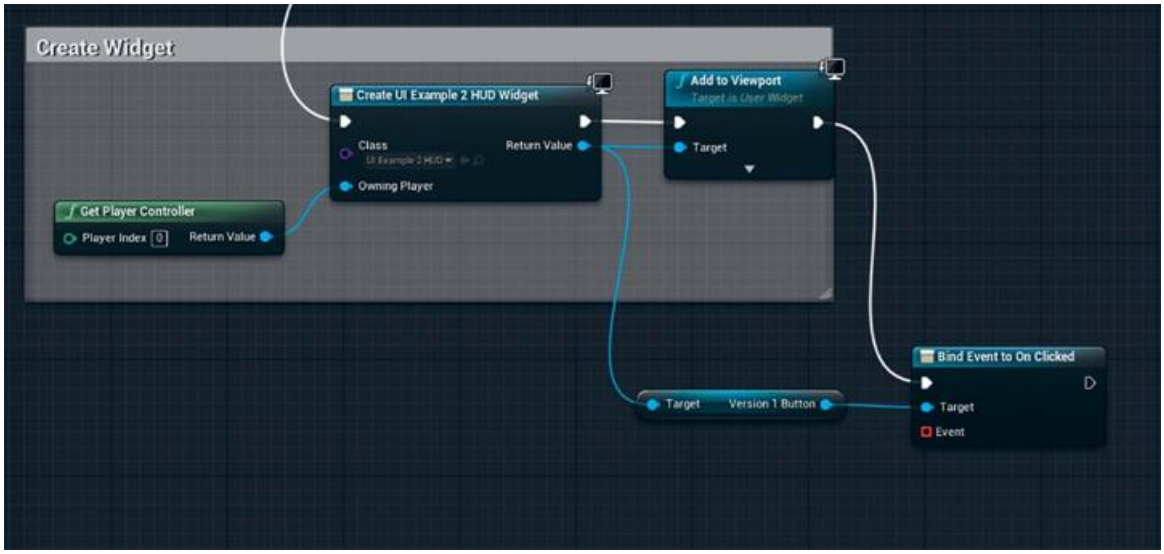

Рисунок 12.23 Получение ссылки на кнопку, которую вы создали в HUD Widget

Найдите Get Version и выберите Get Version 1 Button. Из этого узла Get снова вытяните синее соединение и найдите Bind.

Вы можете привязать несколько событий, и так же вы можете отвязаться от Events. Выберите Bind Event to OnClicked для вашей кнопки. Подключите его к функции Add To Viewport, потому что он должен быть вызван для Bind, которую вы настроите для получения эффекта.

Теперь вам нужно определить, какое событие будет вызваться при запуске привязанного OnClicked Event. Вы сделаете это с красным/оранжевым контактом ссылки на Event.

Перетащите из этого контакта, похожего на контакт в ShowVersion1 Custom Event. Вам может понадобиться передвинуть ваши узлы поближе, чтобы делать это было проще (рисунок 12.24).

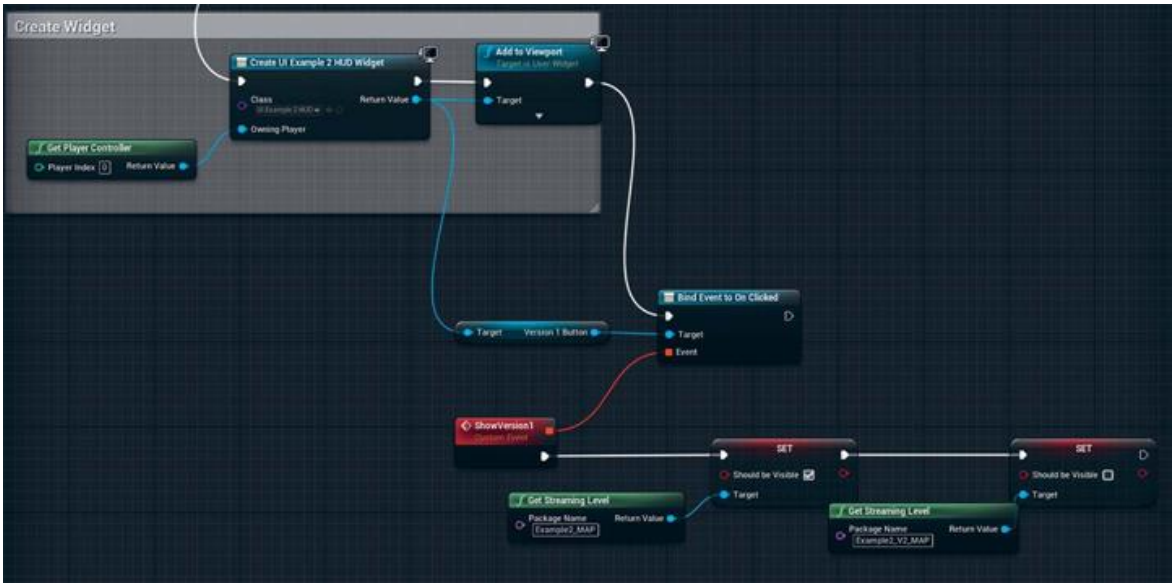

Рисунок 12.24 Привяжите ваш ранее созданный ShowVersion1 Event к OnClicked Event для Version1Button Widget

Теперь, конечно, повторите для другой кнопки, получив ссылку из узла Create Widget и назначив ваш другой Custom Event для OnClicked Event этой кнопки (рисунок 12.25).

Теперь нажатие кнопок вызывает Events, переключающие уровни.

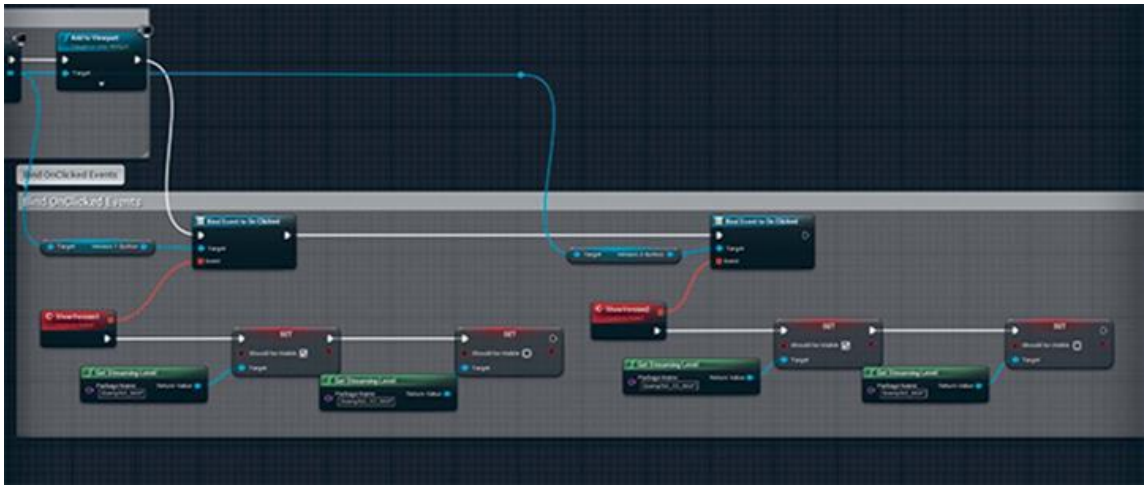

Рисунок 12.25 Обе кнопки связаны с переключающими Events в Persistent Level Blueprint

Компиляция и сохранение

Скомпилировав ваш Level Blueprint, убедитесь, что ошибок нет, и сохраните вашу работу.

Запуск уровня

Теперь вы можете нажать Play на уровне и испытать полностью собственный, управляемый мышью пользовательский интерфейс. Вы должны быть способны перемещаться с помощью клавиатуры и мыши, переключаться между двумя версиями с помощью UI (рисунок 12.26).

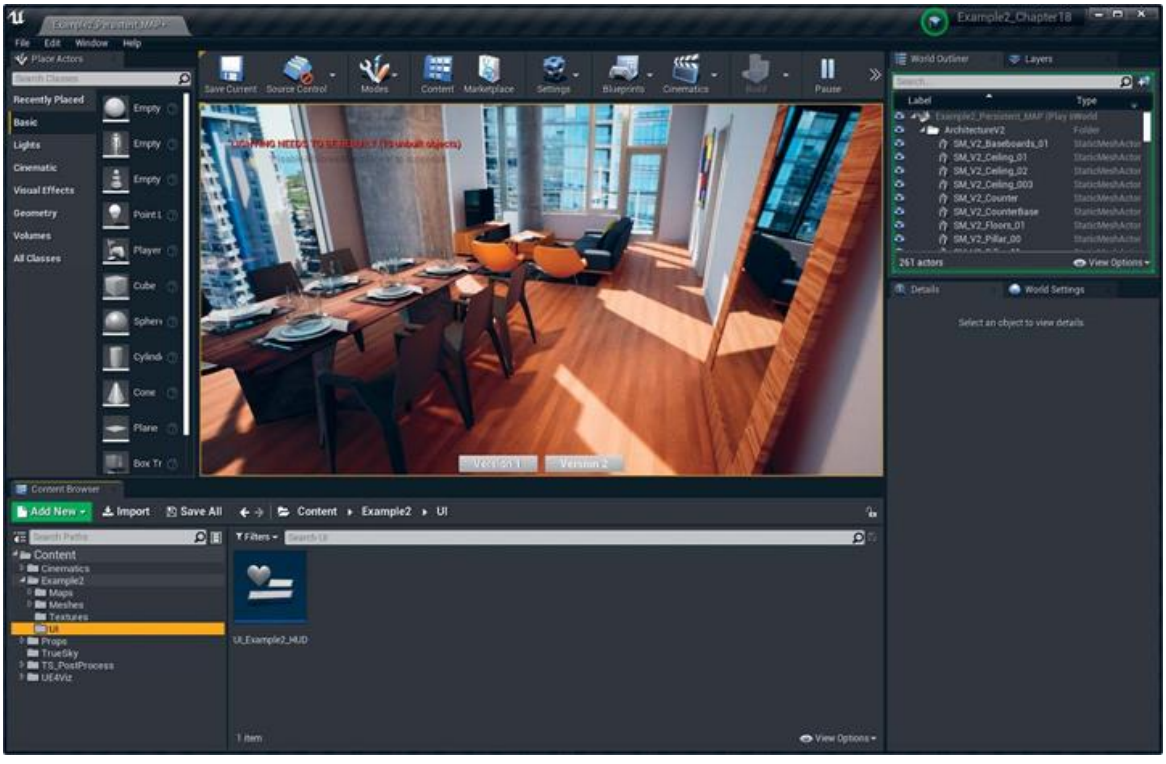

Рисунок 12.26 Запущенная игра в PIE, с полностью функциональными кнопками и мгновенным переключением

Заключение

Вы проделали огромный путь и узнали много нового в этой главе.

Вы изучили систему Level Streaming в UE4 и использовали ее для мгновенного переключения целых уровней в процессе работы. Это отличный способ для смены наборов данных в приложениях для визуализации, который прост в настройке и поддержке.

Вы также рассмотрели, как Blueprints могут взаимодействовать друг с другом с помощью привязки события и использования этой способности для построения простого, но эффективного пользовательского интерфейса с UMG.

UMG является отличным ресурсом и одним из лучших доступных инструментов создания пользовательского интерфейса. В сочетании с гибкостью и мощностью UE4, вы в силах разработать огромное количество приложений, которые только можете себе представить.

Хотя эта игра является лишь простым примером, UMG использовался для таких успешных проектов, как полномасштабная AAA-видеоигра, блокбастер с простым инструментом сравнения A/B, как этот. Его производительность, гибкость и простота в использовании — одна из причин, по которой UE4 становится доминирующей силой практически в каждой визуальной индустрии.

13 ДОПОЛНИТЕЛЬНЫЙ УРОВЕНЬ BLUEPRINTS: ПЕРЕКЛЮЧАТЕЛЬ МАТЕРИАЛОВ

Создание готовых Blueprints необходимо для успешной работы с UE4. Хорошие системы Blueprint создают новую функциональность для приложения и построены как инструмент, который прост в использовании для вас или вашей команды. Далее вы узнаете, как создать один Actor Blueprint, который позволит пользователю нажимать на любой меш на

сцене и переключаться на следующий определенный материал в списке. Рассматривая этот Blueprint, вы сможете заметить, как Blueprints могут взаимодействовать друг с другом и менять свойства на лету.

Теперь, когда вы увидели, как создать приложение для интерактивной визуализации, создать Player Controller и Pawn, переключение между уровнями и как разрабатывается пользовательский интерфейс в UMG, настало время сделать следующий шаг — ну ладно, может быть, короткий лестничный пролет.

Эта глава предназначена для людей, у которых есть опыт программирования в UE4 в целом, или для тех, кто хочет увидеть, как создается сложная система. В любом случае, вы должны иметь четкое представление обо всех темах, затронутых ранее, прежде чем пытаться их воссоздать или следовать этой главе.

Эта глава демонстрирует наиболее важную часть этой системы и дает рекомендации о том, как выполнить некоторые обычные программные паттерны UE4 и Blueprints.

12.7 Настройка цели

Цель — создать Blueprint Actor Class, который:

- позволит левел-дизайнеру (LD) разместить Blueprint на уровне, используя редактор;
- определит список сцен с Static Mesh Actors, которые меняют материалы при нажатии;
- позволит LD определять список материалов, которые циклически повторяются, когда пользователь кликает на один из мешей;
- выделит перечисленных Actors, когда курсор пользователя находится над Actor, демонстрируя, что он может быть изменен.

Можно придумать пять способов заставить систему работать так же. В программировании редко бывают идеальные решения. Каждый программист подходит к проблеме по-своему.

Для этого проекта мы создадим один Actor Blueprint, содержащий несколько переменных для хранения списка мешей и материалов. Этот Blueprint четко покажет LD, что настраивается с помощью визуальных подсказок, что сделает настройку простой и надежной, а также облегчит процесс отладки (рисунок 13.1).

Перенесем часть контента с Marketplace в наш проект и модифицируем его, для размещения объектов с помощью Post-process Domain Material, который будет переключен с использованием Blueprint.

Рисунок 13.1 Material switcher Blueprint, расположенный на Level, показывает LD-friendly design

Создание Actor Blueprint

Вы можете использовать один Blueprint для добавления всего функционала, который нужен для создания простой в использовании системы для пользователя и дизайнера, которому поручено настроить уровень (Level Designer или LD).

Actor Blueprint — это то, что вы можете разместить на уровне и у чего есть 3D-transform, который вы можете изменить. Вы можете прикрепить Components к Blueprint Actors для

расширения их возможностей. Components включают в себя Meshes, Particle Effects, Sound Cues и многое другое.

Actor Blueprints также могут сделать что-то особенное. Они могут ссылаться на других акторов уровня. Это важно, потому что вам нужен этот Blueprint для взаимодействия с различными статическими мешами на уровне, чтобы менять их.

Создайте Blueprint в Content Browser. Когда появятся опции выбора Class, от которого вы наследуете ваш новый Blueprint, выберите Actor Class (рисунок 13.2).

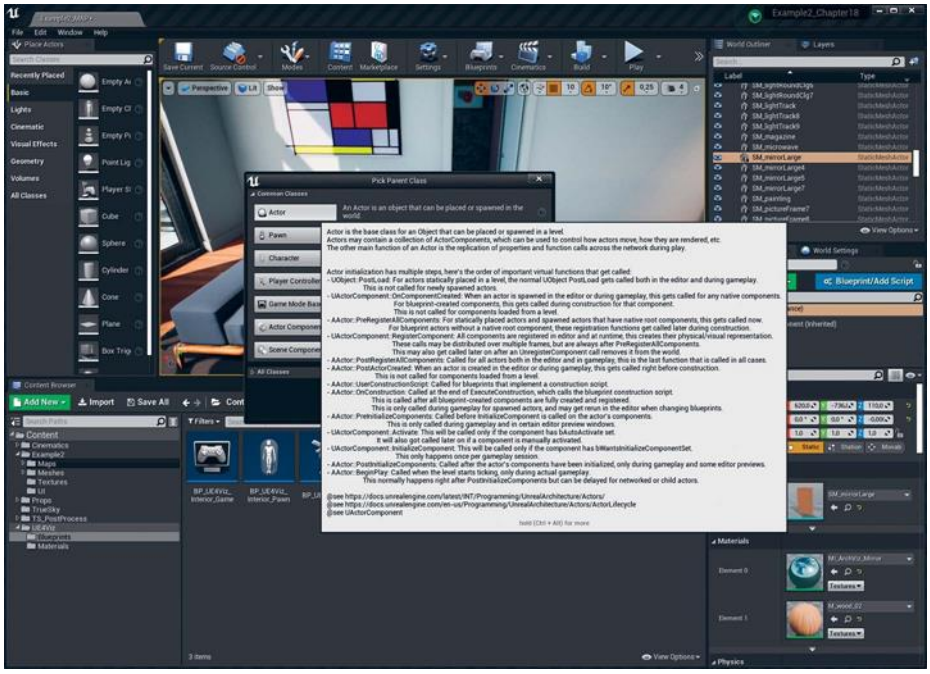

Рисунок 13.2 Создание нового Actor Blueprint из Content Browser

Создание Variables

Вам нужно хранить некоторую информацию или данные в Blueprint Class. Для сохранения данных в Blueprint используйте переменную. Переменные бывают множества разных типов (Classes), включая простые, с которыми вы, скорее всего, знакомы, такие как булевые (boolean) или число с плавающей точкой (float), а также любых других классов, включенных в ваш проект.

Эта возможность использовать классы и акторы в качестве переменных позволяет вам ссылаться на Static Mesh Actors, которые вы хотите изменить. Вы также можете ссылаться на

Assets, хранящиеся в Content Browser. Это позволит вам определять различные материалы, которые будут применены к статическим моделям. Для работы Switcher Blueprint требуется всего несколько переменных (рисунок 13.3).

Вы создаете переменные в Blueprint Editor с помощью кнопки Add New на панели My Blueprint или нажатием на иконку + рядом с разделом Variables на этой панели (рисунок 13.3).

Когда вы создаете переменную, нужно назвать ее и присвоить тип. Вы можете переименовать и присвоить тип позже в панели My Blueprint или в панели Details переменной.

Вы можете не только определить тип (float, string, vector и т. д.) переменной, вы также можете определить ее как массив. Массив переменных становится списком любого типа, на который она установлена. Вы можете редактировать или менять этот список в Details актора, если включен параметр переменной Allow Editing.

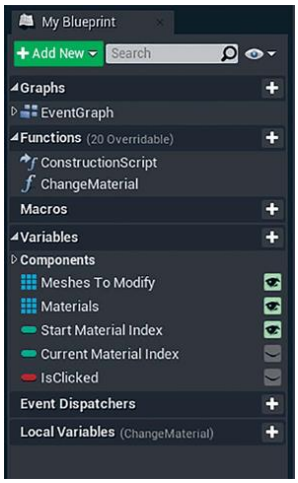

Рисунок 13.3 Переключатель материалов переменных и функций Blueprint

Каждый раз, когда вы создаете новую переменную, у нее по умолчанию будет тип и настройки массива, которые вы установили для последней созданной переменной.

Meshes To Modify Array

The Meshes To Modify Variable — это массив Static Mesh Actors (Static Meshes, который размещен на уровне, а не ссылка на Static Mesh Asset из Content Browser), который должен менять материал как группу при нажатии. Это позволяет LD определить

несколько предметов мебели (например, все стулья для столовой) в массиве, и при нажатии на один все они совместно изменят материал.

Массив Variable может содержать список Level Actors, к которому обращаются узлы Event Graph nodes (рисунок 13.4).

После создания, присвоения имени и назначения типа переменной нажмите на значок Grid рядом с ним, чтобы конвертировать в массив. Обратите внимание, что при нажатии на Variable Details Panel заполняется свойствами этой переменной.

Если для переменной установлено Editable, то она отобразится в панели Details в Level Actors этого класса. Это позволяет каждому экземпляру уровня содержать различный набор мешей, назначенных массиву. Вы можете увидеть на рисунке 13.4, что свойство Meshes To Modify отображается в панели Details.

Expose on Spawn является специальным параметром, который позволяет вам легко установить переменную, если вы программно спавните этого актора в процессе исполнения. Так как вы не будете этого делать, ни у одной из ваших переменных не должно быть установлено это.

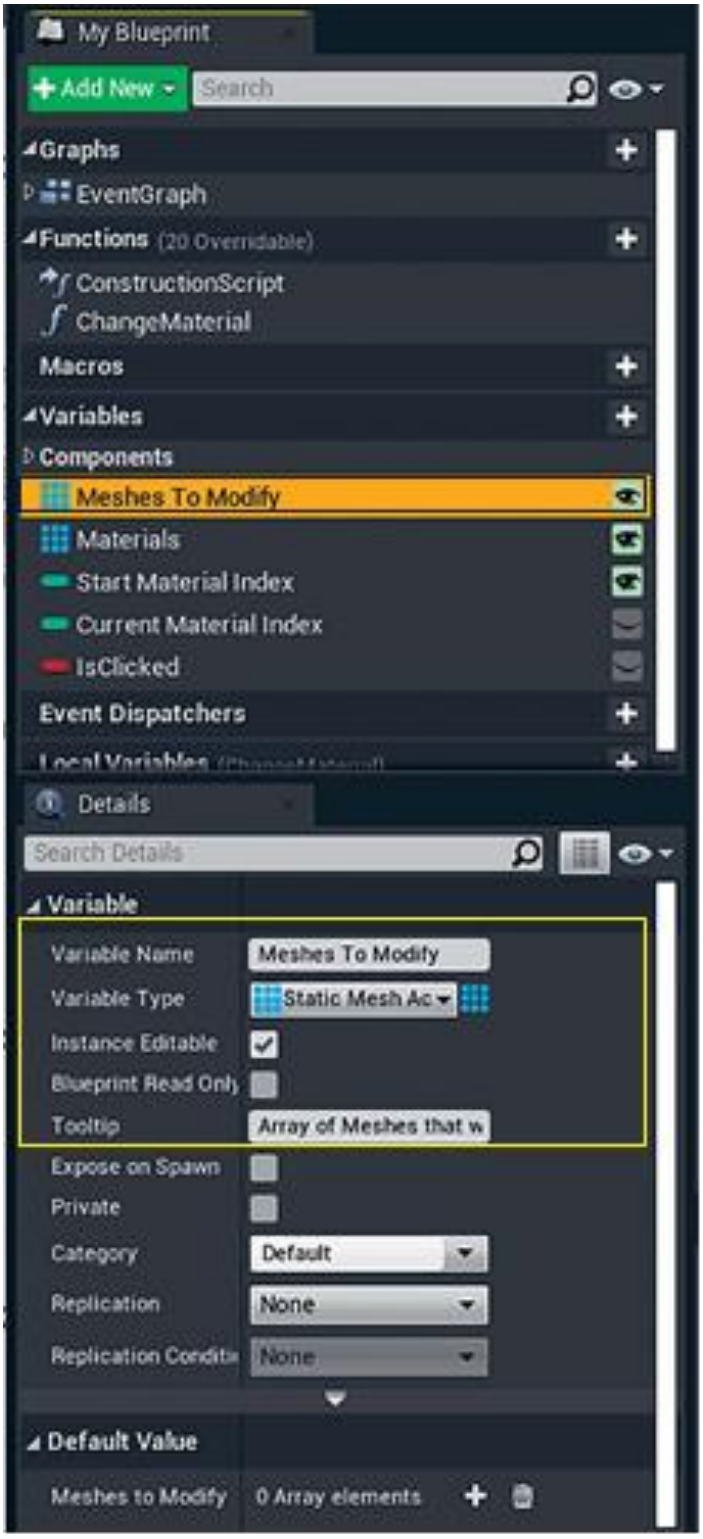

Рисунок 13.4 Свойства Meshes To Modify Variable

12.8 Массив материалов

Materials Variable — это массив ссылок на Material Interface. Тип Material Interface для переменной позволяет

использовать как обычные материалы, так и объекты материалов.

В отличие от Meshes to Modify Variable, которые ссылаются на Static Mesh Actors на уровне, массив Materials Variable ссылается на Assets в Content Browser. Это позволяет системе считывать из библиотеки материалов в вашем проекте, даже если материалы не были назначены для чего-нибудь на сцене.

Установите Editable для этой переменной, позволяющий LD изменять эти значения для каждого актора в пределах каждого уровня.

Переменная Start Material Index

Свойство переменной Start Material Index является типом Integer. Integers — это все целые числа, такие как 1, 2 и 8675309. Они не могут иметь дробной части, но могут быть отрицательными. В этом случае вы можете использовать их для определения, какой материал из массива материалов выбрать.

Индекс — это термин, который относится к числовой позиции значения в массиве.

В UE4 индекс начинает отсчет с 0, поэтому индекс 0 является первым элементом в массиве, а индекс 3 — четвертым.

В этом случае вы будете использовать его, чтобы определить, какой материал применить для вашего Meshes to Modify Actors из массива Materials.

Также установите для него Editable (рисунок 13.5), так вы можете решить, как создатель вашего уровня, с какого индекса начинать и какой материал применять по умолчанию.

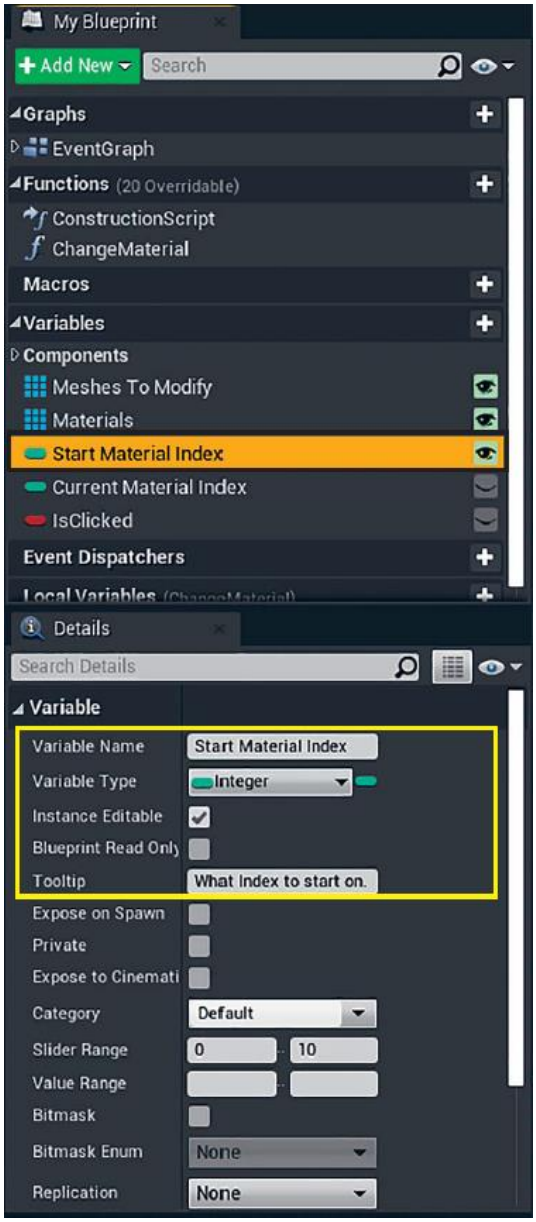

Рисунок 13.5 Свойства Start Material Index Variable, установленные как Integer (целые числа) и Editable

Current Material Index Variable

Установите для Current Material Index другой Integer точно так же, как вы делали в Start Material Index, но отключите свойство Editable, потому что Current Material Index используется только кодом Blueprint в процессе исполнения и никогда не будет напрямую устанавливаться пользователем. Вы станете применять эту Variable для отслеживания пользовательских нажатий, увеличивая его на единицу каждый раз при нажатии пользователя.

IsClicked Variable

Свойство IsClicked переменной имеет тип Boolean. Boolean определен только двумя возможными значениями: true или false. Как и Current Material Index, это не редактируемая переменная (свойство Editable отключено), так как код использует ее во время выполнения для определения, является ли клик пользователя по Actor намеренным или случайным нажатием.

Добавление компонентов

Components похожи на sub-Actors для ваших Blueprints. Это C++ и Blueprint-классы, которые могут содержать все, что может другой Blueprint-класс, включая код, эффекты и ввод.

Есть два способа создать Components в Blueprints — программно во время исполнения или вручную, используя Components и окна Viewport в Blueprint Editor (рисунок 13.6).

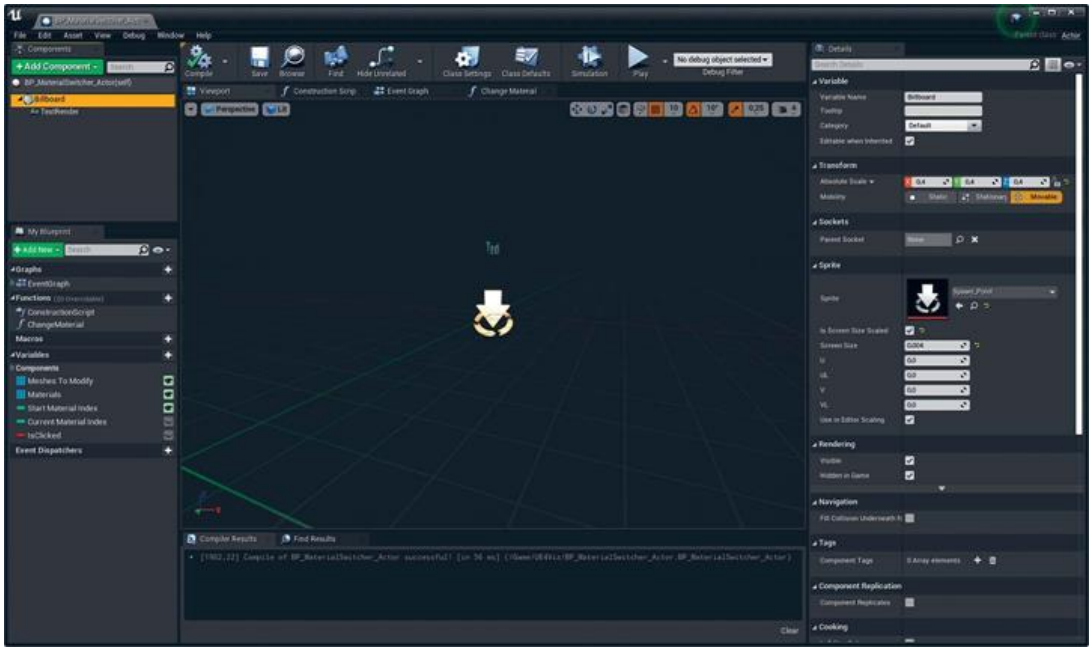

Рисунок 13.6 Детали Billboard Component

Billboard Component

Billboard Component выступает в качестве основы вашего Actor. Billboard Class отображает Texture, которую всегда видит пользователь. Вы можете использовать это для таких вещей, как вспышка объектива или спецэффектов. В этом случае у вас есть что-то, что LD может увидеть и выбрать в уровне.

Для создания компонентов используйте раскрывающийся список Add Component и выберите тип, который вам нужен, в данном случае Billboard Component.

Для замены DefaultSceneRoot Component, который установлен по умолчанию, перетащите только что созданный Billboard Component в DefaultSceneRoot Component, заменив его.

Вы также можете определить собственную Sprite Texture для отображения, как показано на рисунке 13.6. Показанный Spawn_Point из содержимого движка поставляется вместе с UE420. Вы также можете создать и импортировать вашу собственную текстуру для использования здесь.

Параметр Is Screen Size Scaled, установленный на true, гарантирует, что ваш спрайт никогда не будет превосходить определенного размера экрана. Это полезно, чтобы экран не заполнялся огромными спрайтами, так как Viewport camera приближается к ним.

TextRender Component

TextRender Component удобен — он отображает текстовые 2D-строки в 3D-простран- стве, что может быть изменено на лету (рисунок 13.7). Использование его для отображения названия первого выбранного меша в массиве Meshes To Modify. Это только помогает LD при создании уровня.

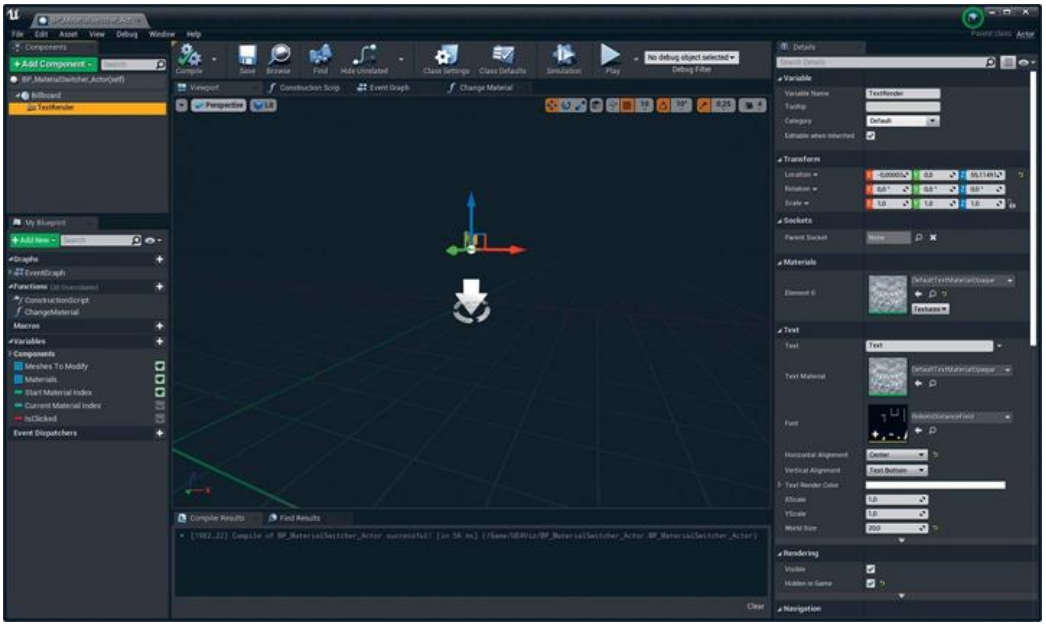

Рисунок 13.7 Детали TextRender Component

Вы можете получить доступ к контенту, который входит в состав Engine, выбрав значок Eye в Content Browser, а затем — Show Engine Content. Будьте осторожны, чтобы ничего не изменить в содержимом вашего Engine, так как это может привести к проблемам со стабильностью и обмену контентом между членами команды.

12.9 Создание функции Change Material

Функция Change Material (рисунок 13.8) действительно является ядром всего этого Blueprint. Она обрабатывает работу по назначению материалов для назначенных Static Meshes. Также она содержит некоторую логику для избегания ошибок и может гарантировать, что материалы всегда назначены.

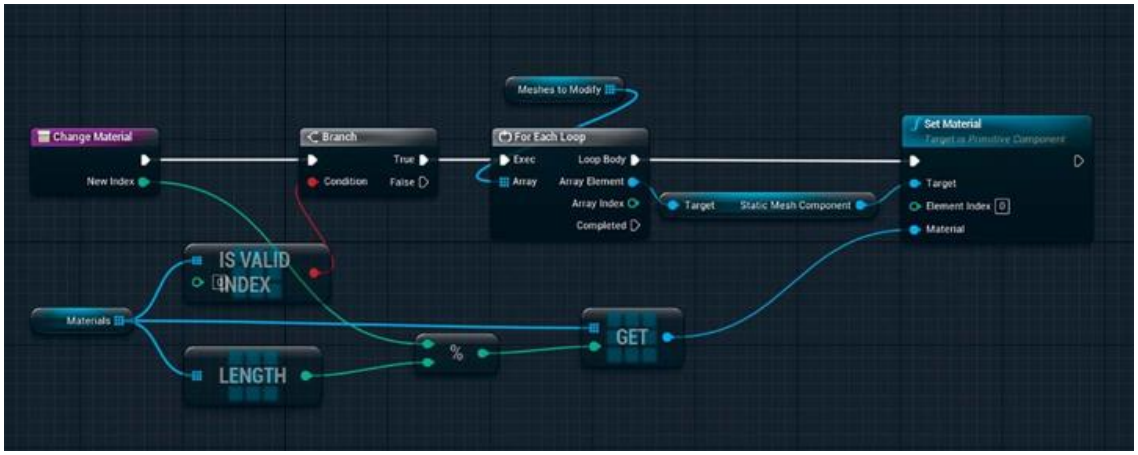

Рисунок 13.8 Изменение Material Function

Функции используются для инкапсуляции определенного функционала, особенно когда вы хотите его повторно использовать. В данном случае вам нужно изменить материал в процессе Construction Script, после Begin Play Event в Event Graph и вовремя выполнения, когда пользователь нажимает на перечисленный меш.

Вы создаете функции аналогично тому, как создаете переменные, используя панель My Blueprint.

Нажмите на кнопку Add New и выберите Function из выпадающего меню. Назовите ее Change Material.

После создания функции она сразу же откроется для редактирования в пустом Event Graph. Чтобы открыть уже

созданные функции, дважды нажмите на них для входа в панель My Blueprints.

New Index Input

Функция может содержать входные и выходные параметры, которые позволяют обрабатывать и возвращать данные. Входные и выходные параметры даже могут быть разных типов данных. Простой функцией, которая делает это, является функция Get массива. Вы предоставляете ей целочисленный индекс, и она вернет что угодно, что массив определил, как Static Mesh или Material.

Для добавления входных параметров нажмите на узел функции в Graph Editor; это отобразит свойства функции, включающие область для добавления входных и выходных параметров.

Нажмите на кнопку + в Details Panel рядом с Inputs для добавления нового входного параметра. Назовите его New Index и установите ему тип Integer.

Этот входной параметр будет определять, какой индекс массива Materials использовать. Это позволит функции стать многофункциональной. Вместо того чтобы писать новую функцию для каждого номера материала, вы можете просто использовать эту переменную для определения желаемого индекса.

Is Valid Index

Проверка Is Valid Index выполняется в массиве материалов перед продолжением, чтобы удостовериться, что ваш массив валидный и что вы не обращаетесь к пустой записи массива.

12.10 Цикл For Each

Если Is Valid прошла, тогда функция проходит через каждый меш в массиве Meshes To Modify и вызывает для него функцию Set Material.

Чтобы определить, какой материал применить, вы должны получить ссылку на него из массива материалов с помощью предоставленного значения New Index.

Узел Modulo (%) используется, чтобы гарантировать, что любое целое число будет в пределах доступного диапазона в массиве Materials. Он делает это, возвращая остаток от деления A на B — к примеру, 1 mod 4 = 1, 4 mod 4 = 0, 5 mod 4 = 1 и 55 mod 4 = 3.

Понимание Construction Script

Два случая, когда код Blueprint запускается во время конструирования акторов,

Construction Script и в процессе исполнения Event Graph.

Construction Script запускается только при появлении класса впервые в мире. Это может случиться либо, когда вы размещаете или меняете актора в редакторе, либо, когда актор или объект появятся в процессе выполнения. Construction Script — это место, где вы можете запрограммировать Blueprint до того, как игра запустится. Это включает появление компонентов, изменение других акторов и так далее.

Акторы, размещенные на уровне (в отличие от появляющихся во время выполнения), запускают только их Construction Script в Editor, и результаты сохраняются в файл, когда вы сохраняете уровень. Construction Script больше не запустится, даже если перегрузить уровень.

Construction Script в нашем классе делает три основных действия: устанавливает text label, рисует линии от актора до мешей, которые будет изменять, и устанавливает материалы для меша в соответствии со Start Index Variable (рисунок 13.9).

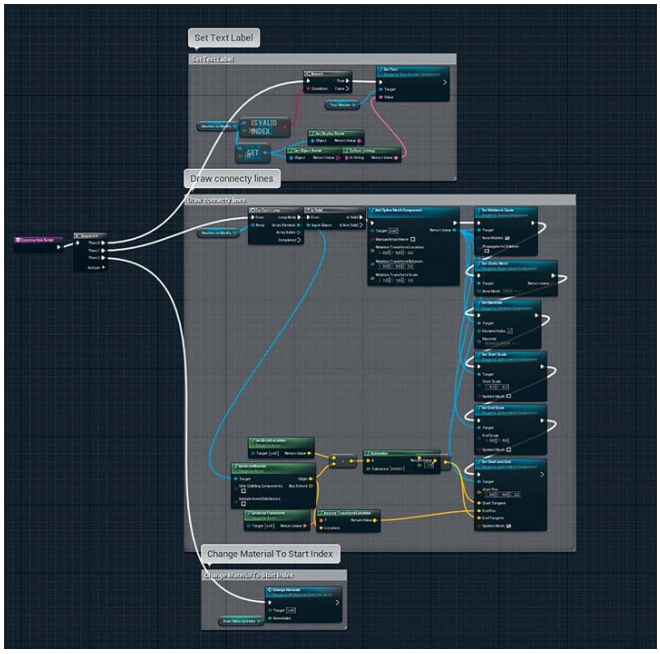

Рисунок 13.9 Окончательный Construction Script с тремя основными функциями, выполняемыми Construction Script

Установка Text Label

Construction Script сначала пытается установить в TextRender свойство Text на основе выбранных мешей (рисунок 13.10). Если Meshes не выбраны, он определяет, как None, помогая LD понять, правильно ли настроены его системы.

Чтобы сделать это, Construction Script должен получить ссылку на первый элемент массива Meshes To Modify, для этого получите имя с помощью Get Display Name и соедините его с Set Text Function с текстом в Render Component.

Как и множество узлов Blueprint, которые ссылаются на определенный класс, вам нужно сначала получить ссылку на ваш Text Render Component и затем перетащить соединение из него в Editor, чтобы увидеть контекстный список узлов, доступных для этого класса.

Вершина Branch требуется, чтобы гарантировать, что вы не вызовете Get для пустого массива, таким образом, исполнение продолжится, только если первая запись в массиве Meshes To Modify (Index 0) действительна.

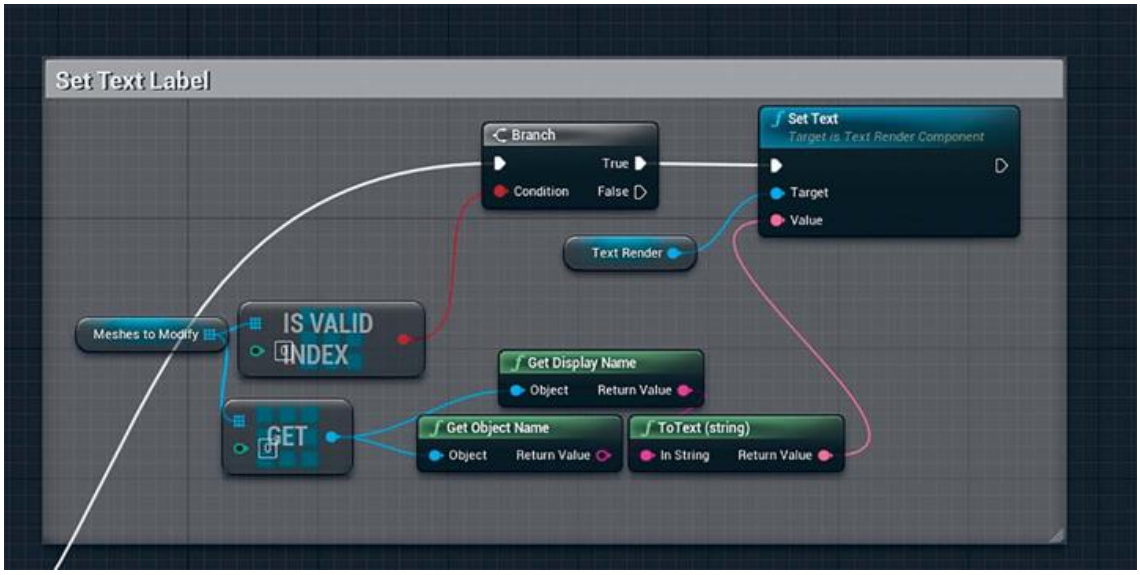

Рисунок 13.10 Детали TextRender Component

12.11 Рисование коннекторов

При разработке Blueprints для использования в производстве вы должны учитывать не только как пользователь взаимодействует с Blueprints, но еще и как человек, настраивающий уровень, должен взаимодействовать с ним. Чрезмерно сложные или трудные в использовании инструменты не востребованы, и время, затраченное на их создание, теряется впустую, поэтому наведение лоска функциональности в редакторе может гарантировать, что практически каждому будет легко использовать.

Чтобы дать возможность LD способы визуализации того, как Static Mesh Actors ссылается на каждый Material Switcher Actor, их расположили в специально выделенной планке для каждого перечисленного меша из массива Meshes to Modify (рисунок 13.11).

Так как вы хотите нарисовать линию для каждого меша в вашем массиве Meshes To Modify, вам нужно перебрать массив, используя цикл For Each.

Каждый цикл сначала проверяет, возвращает ли допустимый Array Element (возможно, в вашем массиве могут быть пустые записи). Если допустимый, то используется узел

Add Spline Mesh Component для добавления нового Spline Mesh Component в ваш Blueprint.

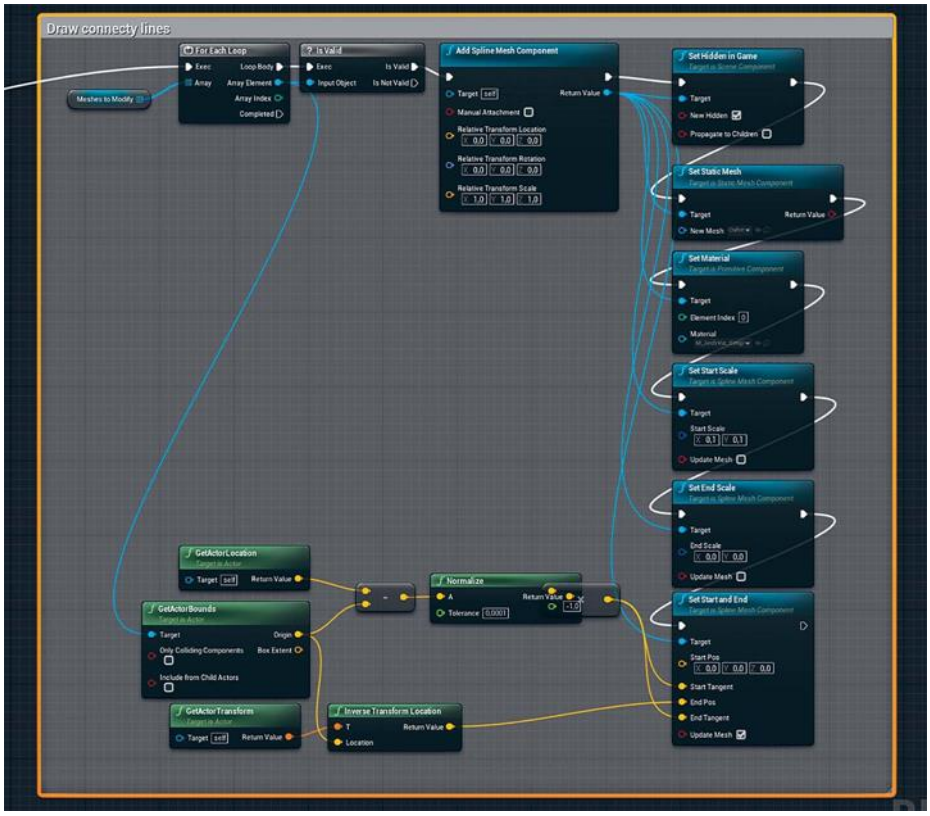

Рисунок 13.11 Рисование 3D-линий для визуализации назначенных Meshes

Затем каждая линия оформляется, так чтобы она указывала от Switcher Blueprint на каждый ссылочный объект Mesh.

Set Hidden in Game позволяет Component быть видимым в редакторе, но, когда вы переходите в Game Mode (с помощью симуляции, нажатия на Play или нажатия G для предпросмотра Game Mode), он будет скрыт. Множество Editor-only-акторов используют этот параметр, например, значки и стрелки в Light Actors.

Затем Cube Mesh Static Mesh Asset присваивает Spline Mesh Component вместе с материалом, используя Set Static Mesh и Set Material Functions.

Начальный и конечный масштаб устанавливается, создавая форму, подобную стрелке, используя Set Start Scale и Set End Scale.

Наконец, используя функцию Set Start and End, чтобы установить начальные и конечные позиции сплайна относительно мировой позиции Blueprint Actor. Так как эти местоположения относятся к позиции Blueprint в мире, значение 0,0,0 представляет место точки опоры вашего Blueprint.

Мировая позиция Static Mesh Actor запрашивается с помощью функции Get Actor Bounds. Затем она преобразуется в локальные с помощью узла Inverse Transform Positon и передается в свойства End узла Set Start and End.

Сплайн в этой точке будет выглядеть очень странно, так как сначала вы должны определить направление касательной для каждой точки на сплайн. Поскольку вы наверняка хотите, чтобы касательные просто шли вниз по направлению сплайна, вы вычисляете касательную между ними, вычитая их позиции и затем нормализуя результат для получения Unit vector, который можете соединить с двумя свойствами касательной Start and End.

Использование Context Sensitive Checkbox

Помните, что для доступа к этим функциям вам нужно перетащить соединение из Return Value в Add Spline Component для получения доступных для Class-specific методов. Вы также можете не проверять флажок Context Sensitive во всплывающем окне. Это перечислит все доступные методы, которые вы можете расположить в Blueprint; однако вы можете запутаться, так как вам будет предложено гораздо больше вариантов, чем если бы вы использовали контекстно отсортированный список.

Техника перетаскивания соединений полезна для всех видов узлов. Например, чтобы легко найти математический узел Vector * vector, вы можете перетащить желтое соединение из выхода vector и напечатать *, чтобы получить список всех возможных функций умножения, которые может использовать vector.

Замена материала по Start Material Index

Чтобы разрешить переключателю материалов (Material Switcher) работать в редакторе, вы можете вызвать Change Material Function, которую вы создали в Construction Script

(рисунок 13.12). Теперь, когда LD устанавливает Start Material Index Variable, он увидит соответствующее изменение материала в редакторе.

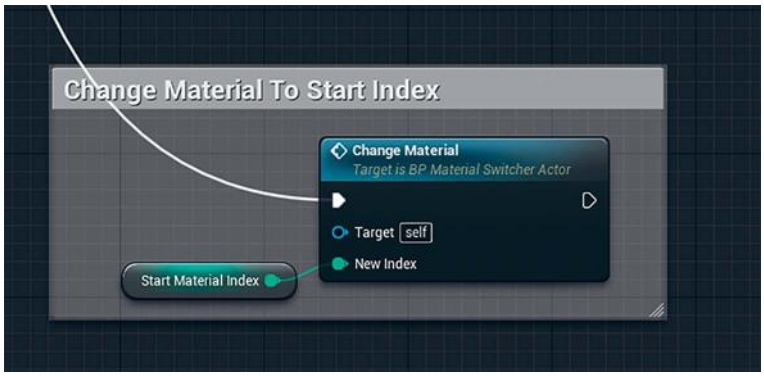

Рисунок 13.12 Вызов функции Change Material в Construction Script

Добавление функции Change Material в Construction Script также является удобным способом протестировать вашу функцию, поскольку вы можете вручную установить Index и увидеть, действительно ли она изменит материалы, как предполагается.

Понимание Event Graph

Граф событий (Event Graph) является частью Blueprint, которая запускается во время игры, в котором у вас есть функции, события и узлы, отвечающие за разные игровые события, такие как Tick и Input Events. В данном Blueprint, Event Graph обрабатывает выделение объектов Actors и меняет материалы на них, если пользователь нажимает на актора (рисунок 13.13).

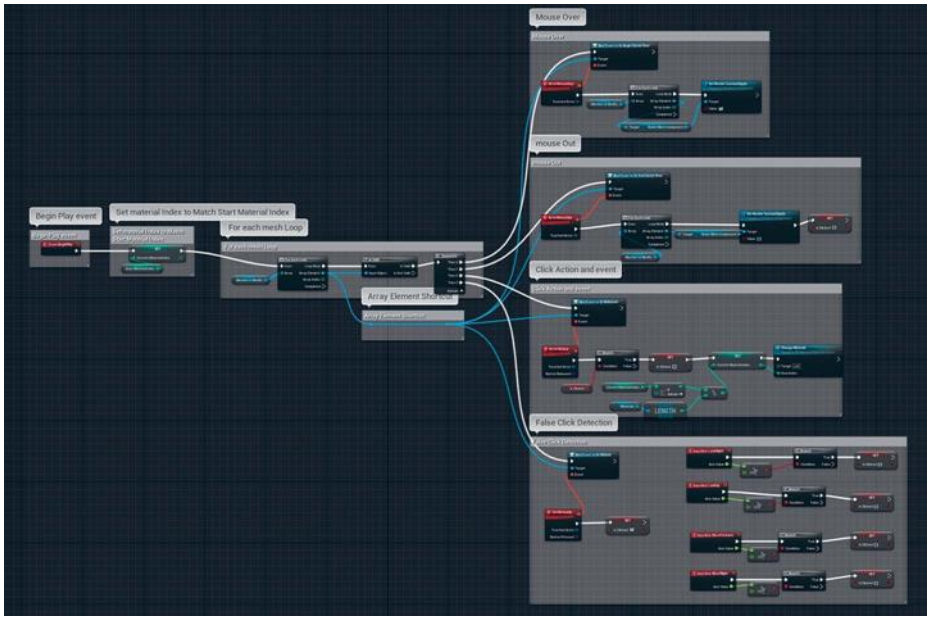

Рисунок 13.13 Обзор Event Graph с показом главных Functions

Прежде чем двигаться дальше, убедитесь, что вы работаете во вкладке Event Graph в вашем Blueprint Editor.

Основное назначение Event Graph в классе Material Switcher Blueprint — настроить все Meshes To Modify Actors для получения Events ввода мыши. Для этого код циклом проходит через каждый меш в массиве Meshes To Modify и использует Event Binding для сообщения этим мешам, как вести себя, когда на них нажимает пользователь.

12.12 Событие Begin Play

Код Event Graph в этом Blueprint начинается с события Begin Play. Это событие вызывается только один раз, когда Actor или Object впервые появляются в игре в процессе исполнения.

Первая часть Graph из узла Event Begin Play, скорее всего, теперь понятна. Для каждого Static Mesh Actor в Meshes To Modify циклом проходит каждый меш в массиве, выполняя проверку на доступность, чтобы избежать обращения к null указателя. Если он доступен, то функционал доступен различным событиям взаимодействия мыши.

Привязка события

Blueprints можно связать с событиями, которые происходят в других классах вашей игры, и это очень полезно для сохранения простоты и краткости ваших Blueprints. Вам не надо добавлять код в каждый объект Mesh, в котором вы хотите зарегистрировать события мыши, — просто используйте функционал Bind to Event.

Как и прежде, вы должны перетянуть соединение из красного/оранжевого Bind To Function's поля делегата в Graph Editor, отпустить и выбрать Add Custom Event. Это создаст новый Custom Event, который содержит Inputs и другие параметры, правильно настроенные для этого Event. В данном случае вы можете видеть, что Custom Event возвращает ссылку на Touched Actor, который добавлен автоматически.

Mouse Over и Mouse Out

Подобно множеству приложений, которые содержат управляемые мышью интерфейсы, UE4 предлагает Events, когда курсор мыши начинает и заканчивает висеть на объекте Actor.

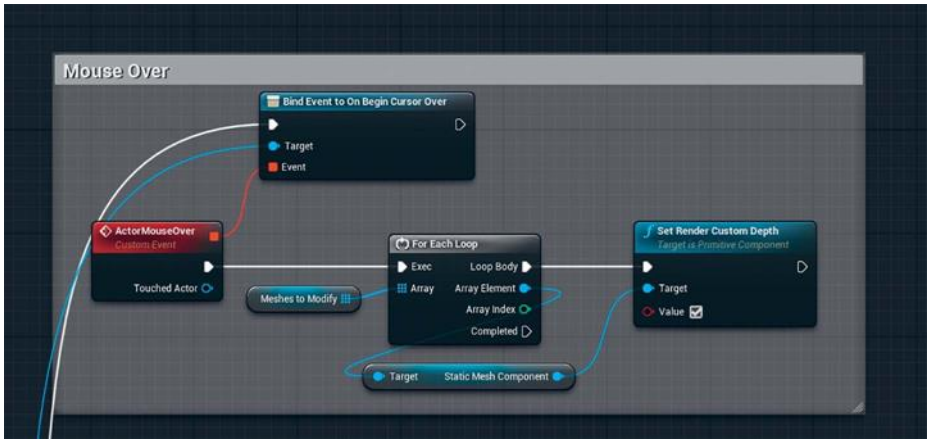

Рисунок 13.14 Граф события Mouse Over Event

В данном случае Event используется для установки Custom Depth на каждый меш в массиве Meshes To Modify, что позволяет post-process Material обнаруживать их. Вы можете заметить на картинках (рисунок 13.14 и 13.15), что, как только пользователь наводит курсор на акторов, код в цикле просто перебирает массив Meshes To Modify, устанавливает свойство Custom Depth для каждого меша на true и возвращает false, когда курсор покидает.

Custom Depth затем считывается Post-Process Material, и генерируется контур вокруг поверхности актора.

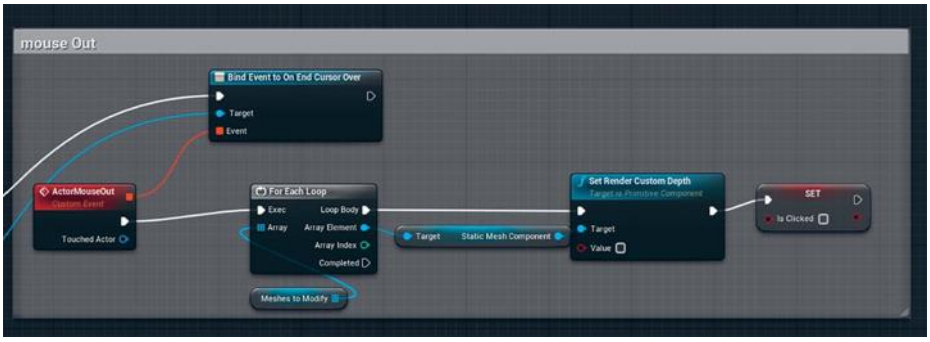

Рисунок 13.15 Mouse Out Event Graph

Click Action

Когда пользователь нажимает на конкретный меш в массиве Meshes To Modify, нужно изменить материалы. Это делается с помощью увеличения Current Material Index Variable, снова используется оператор взятия по модулю (%) (рисунок 13.16), чтобы убедиться, что значение остается в пределах диапазона; в основном зацикливая целое число по определенному номеру.

Затем результат взятия по модулю отправляется в Change Material Function, которая, в свою очередь, меняет все меши в массиве Meshes To Modify.

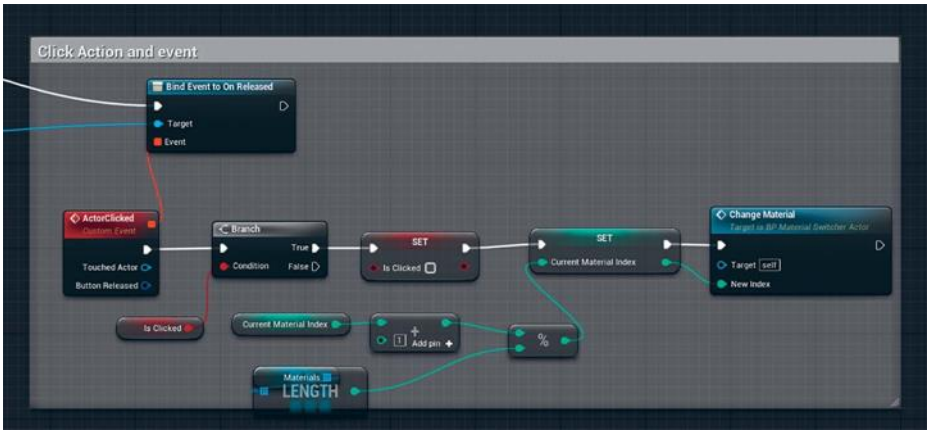

Рисунок 13.16 Click Action и Event

Определение False Clicks

Иногда пользователь захочет повернуть камеру с помощью мышки и не нажимать на интерактивные меши, размещенные на уровне. Если вы позволяете менять стены и полы, могут

существовать несколько мест, куда пользователь может нажать без случайного переключения материала.

Для решения этой проблемы Blueprint обнаруживает входные данные, которые могут указывать пользователю не только нажимать, но нажимать и перетаскивать, предположительно для поворота обзора (рисунок 13.17).

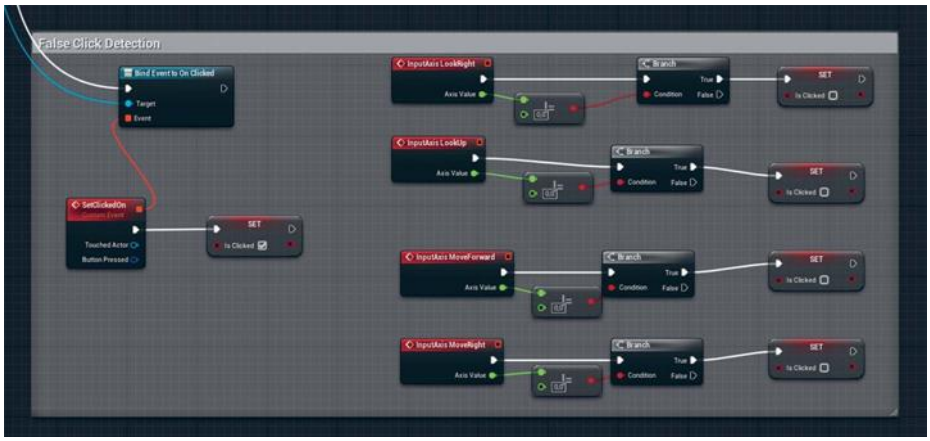

Рисунок 13.17 Обнаружение ложного нажатия

Это происходит, когда IsClicked Variable применяется. Присвоение значения true этой переменной происходит, только когда пользователь напрямую нажимает на меш, и после этого присваивается false для любых других вводов мыши, избегая вызова функции Change Material.

12.13 Заполнение уровней

Как и все ассеты, которые вы хотите взять из Content Browser для уровня, лучше всего просто перетащить их из Content Browser в Viewport.

Начните с размещение одного из Material Switcher Blueprints рядом с Meshes, которые хотите изменить (рисунок 13.18). Обратите внимание на Details Panel справа, что Variables, которые помечены как Editable, видны и готовы для изменений LD.

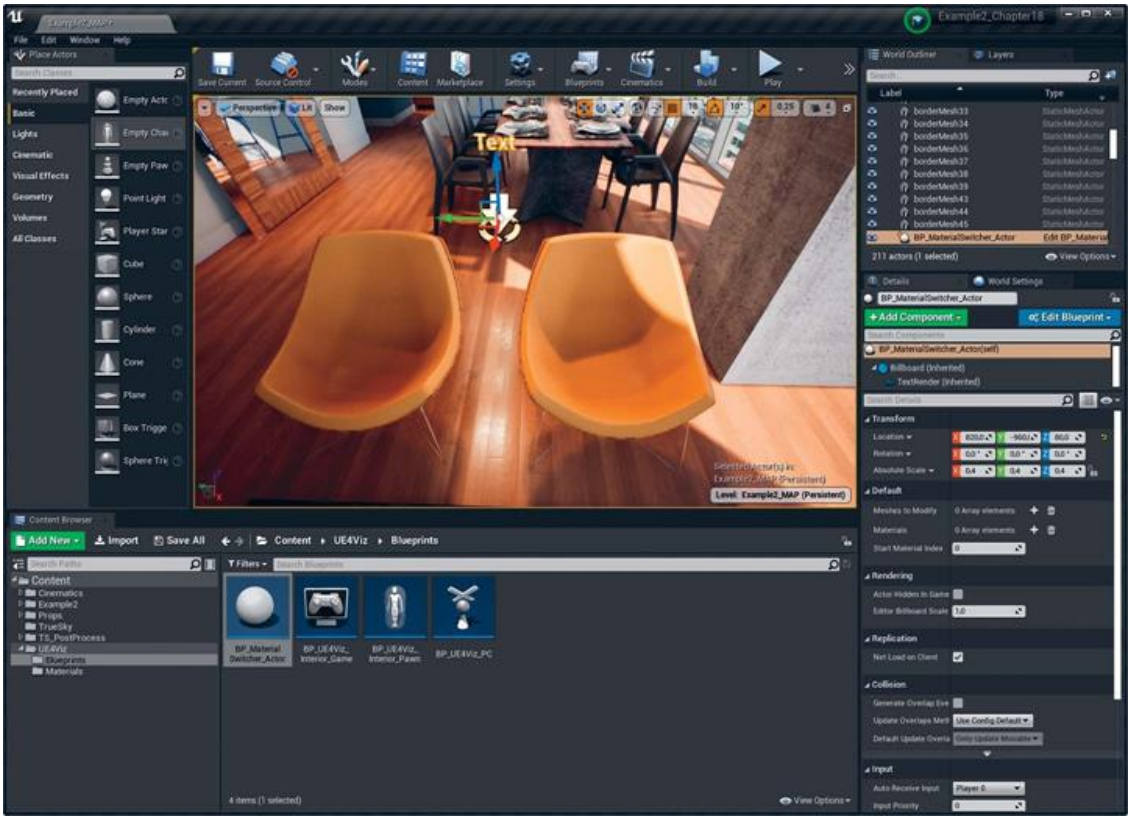

Рисунок 13.18 Material Switcher добавлен на Level рядом с парой стульев

Добавление мешей

Теперь определите меши, которыми хотите повлиять на Blueprint. В Details Panel, найдите массив Meshes To Modify и нажмите на кнопку «+» для добавления нового элемента в массив. Появится выпадающий список, показывающий каждый статический меш на вашем уровне. Это может быть трудный способ выбора мешей; попробуйте лучше возможность использования инструмента eyedropper (рисунок 13.19).

Как только вы добавите меши в список, запустится Construction Script, и вы увидите линию, проведенную между Blueprint и каждым мешем в списке (рисунок 13.20).

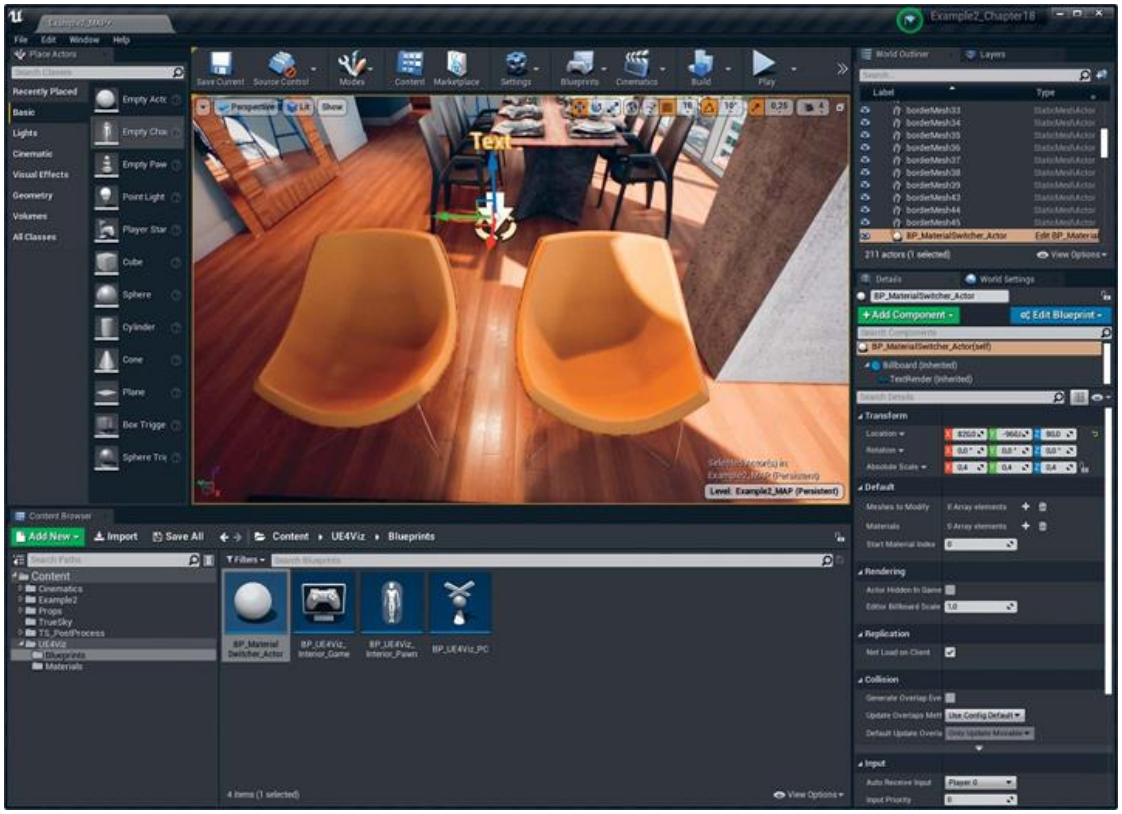

Рисунок 13.19 Использование инструмента Eyedropper для выбора Meshes

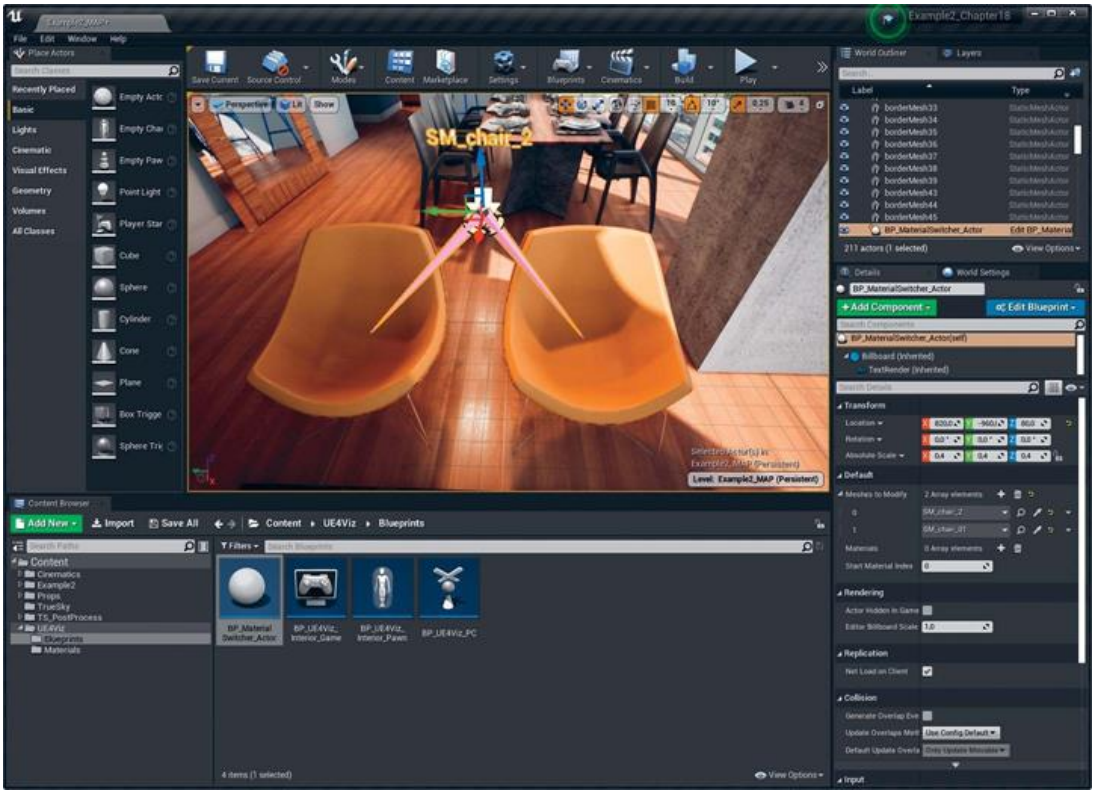

Рисунок 13.20 Добавленный Meshes в массив Meshes to Modify. Линии соединения и Text label также видны

Добавление материалов

После того как вы заполнили список Mesh, проделайте аналогичную настройку для материалов. Вместо выбора мешей из мира используйте eye dropper, как раньше, выберите ваш материал или объект материала из Content Browser, так как массив Materials скорее всего является ссылкой на Asset, нежели на Actor (рисунок 13.21).

Рисунок 13.21 Полностью настроенный Blueprint с Materials, назначенными массиву Materials

12.14 Настройка параметров по умолчанию

Не так много параметров доступно другим в этом Blueprint, не считая массивы. Конечно, вы можете назначить переменную Start Material Index. Это легкий способ настроить внешний вид вашей области по умолчанию и способ предпросмотра Materials до нажатия Play.

Copying и Pasting

Отличная короткая дорожка для настройки акторов — возможность копировать и вставлять их как переменные. К примеру, вы можете выбрать запись массива Materials в Details

Panel и выбрать Copy перед вставкой в то же поле свойства другого Actor.

Этот трюк не ограничивается Blueprints. Вы будете удивлены, как много вещей можно копировать и вставлять в UE4.

Запуск приложения

Все, что осталось сделать, — это протестировать, нажав кнопку Play. Вы должны иметь возможность перемещаться в пространстве, нажимать на различных акторов на уровне и менять их материалы, создавать персонализируемое внутреннее пространство (рисунок 13.22).

Рисунок 13.22 Использование Play в Editor для тестирования системы

После установки достаточного количества мешей и материалов вы можете резко сменить внешний вид сцены с помощью всего лишь нескольких нажатий (рисунки 13.23 и 13.24).

Рисунок 13.23 До нажатия

Рисунок 13.24 После нажатия

Заключение

Теперь у вас есть законченное приложение с полностью проработанным функционалом, которое можно отправить вашему клиенту, радуясь тому, что выполнили все свои обязанности. Вы увидели, как простой Blueprints может содержать мощные эффекты в игровом мире только с несколькими узлами.

Вы увидели два способа создания интерактивности: используя UMG и напрямую взаимодействуя с акторами на уровне.

Вы также увидели пару способов переключения данных UE4, оба с их собственными преимуществами и рабочими процессами.

14 ЗАКЛЮЧЕНИЕ

С доступом к исходному коду C++, огромному массиву плагинов и интеграций, а также лучшим разработчикам на Земле, которые улучшают его каждый день, UE4 является действительно мощной силой, способной изменить индустрию визуализации.

UE4 продолжает меняться

Unreal Engine 4 развивается быстрее, чем любой другой программный пакет, который когда-либо использовали. Эволюция редактора и инструментов с первого знакомства с UE4 Rocket Beta поразительна.

Постоянные улучшения, инновации и поддерживаемые рабочие процессы добавляются с такой регулярностью, что весомой частью работы стало просто поспевать за всем, что выходит.

Очень интересно работать с таким ярким сообществом и видеть, как каждый день тысячи людей и команд со всего мира делают невероятные вещи.

Будущее визуализации

У визуализации всегда было несколько основных проблем. Предоставление качественного продукта, который эффективно рассказывает истории и сообщает сложную информацию визуально, может стать чрезвычайно сложной задачей. Визуализация часто разрабатывается в немыслимо короткие сроки для сильно ограниченных бюджетов с удивительно неполными или меняющимися (или оба варианта) наборами данных.

UE4 поможет значительно облегчить это противостояние. Интерактивность позволяет пользователю визуализировать данные по своему усмотрению в собственном и стиле. Как и в игре пользователь использует интерактивную визуализацию для создания своей уникального проекта и делает это со своей точки зрения.

Затронуты только самые основы того, что может сделать UE4 и что вы можете с ним делать. Показанные примеры также представляют только маленькую часть типов визуализации, создаваемых сегодня. Наборы данных и требования клиентов широко варьируются от одной индустрии к другой и даже от одного проекта к другому. Потому что многие люди используют и изучают UE4, так что вы никогда не останетесь один на один с трудностями. И Epic Games, и сообщество UE4 стремятся сделать UE4 лучшим движком для всех.

Поскольку вы все больше и больше будете использовать UE4, можно взглянуть на процесс работы и подумать, как можно его адаптировать из UE4 в общий студийный подход. Подобно тому как объединяются подходы Голливуда и игр, вы можете извлечь много уроков из разработки игр и интерактивных визуализаций, которые можно применить к повседневной работе архитектора, что не только улучшит работу, но и упростит ее интеграцию в UE4.

РЕКОМЕНДАТЕЛЬНЫЙ СПИСОК ЛИТЕРАТУРЫ

Основная

1. ЕСКД. Общие правила выполнения чертежей. М.: Государственный комитет СССР по стандартизации, 1983
2. Единая система конструкторской документации. ГОСТ 2.101-68 – 2.109-68, ГОСТ 2.301-68 – 2.317-69.
3. Начертательная геометрия. Инженерная и компьютерная графика [Текст]: учебник под общ. ред. П.Н. Учаева и В.И. Якунина, – М.: Академия, 2008– Т1: Начертательная геометрия, геометрическое и проекционное черчение. – 304 с.
4. Начертательная геометрия. Инженерная и компьютерная графика [Текст]: учебник под общ. ред. П.Н. Учаева и В.И. Якунина, – М.: Академия, 2008.– Т2: Машиностроительное черчение. – 344 с.
5. Конакова, И.П. Инженерная и компьютерная графика [Электронный ресурс]: учебное пособие / И.П. Конакова, И.И. Пирогова ; Министерство образования и науки Российской Федерации, Уральский федеральный университет имени первого президента России Б.Н. Ельцина.- Екатеринбург : Издательство Уральского университета. 2014.-91 с. Режим доступа: –biblioclub.ru
6. Рочегова, Н. А. Основы архитектурной композиции. Курс виртуального моделирования [Текст] : учебное пособие / Н. А. Рочегова, Е. В. Барчугова. - М. : Академия, 2010. - 320с.
7.

Дополнительная

8. Начертательная геометрия. Инженерная и компьютерная графика в задачах и примерах [Текст]: учебное пособие / ред. П.Н.Учаева. Старый Оскол: ТНТ, 2011.-288 с.
9. Компьютерные технологии и графика [Текст]: учебное пособие / ред. П.Н.Учаева. Старый Оскол: ТНТ, 2011. – 280 с.

10. Пронин, Г. Технология дизайна в 3ds Max 2011. От моделирования до визуализации [Текст] / Г. Пронин. - СПб. : Питер, 2011. - 384 с.

11. Единая система конструкторской документации. ГОСТ 2.301-68 - 2.317-68.

12. Иванова, Светлана Ивановна. Построение изображений [Электронный ресурс]: учебное пособие / ЮЗГУ; Министерство образования и науки Российской Федерации, Юго-Западный государственный университет. – Курск: ЮЗГУ, 2011.-102 с.

Перечень ресурсов информационно-телекоммуникационной сети Интернет, необходимые для освоения дисциплины

1. http://window.edu.ru – Бесплатная электронная библиотека онлайн «Единое окно к образовательным ресурсам».
2. http://www.edu.ru – Российское образование. Федеральный образовательный портал: учреждения, программы, стандарты, ВУЗы, тесты ЕГЭ, ГИА.
3. http://www.mon.gov.ru – Министерство образования и науки Российской Федерации.
4. http://biblioclub.ru – Электронно-библиотечная система «Университетская библиотека онлайн»
5. https://docs.unrealengine.com– Официальная справка Unreal Engine.