

14-Й ЧАС

Введение в систему визуального программирования блюпринтов

Что вы узнаете в этом часе

- ▶ Изучение интерфейса редактора блюпринтов
- ▶ Как использовать события, функции и переменные
- ▶ Добавление события
- ▶ Объявление переменной

Практически каждый игровой движок имеет скриптовый язык, позволяющий разработчикам добавлять и изменять функционал их игр. Некоторые движки используют существующую скриптовую среду, например LUA*, другие же создают собственную. UE4 предоставляет два способа создания контента: язык C++ и блюпринт (Blueprint). Система визуального программирования блюпринтов — это мощная и полнофункциональная среда разработки скриптов, работа с которой осуществляется посредством редактора. Она предоставляет художникам и дизайнерам возможность создавать полноценные игры, прототипировать идеи и изменять существующие элементы геймплея. В этом часе представлен редактор блюпринтов и базовые понятия разработки скриптов.

ПРИМЕЧАНИЕ

Подготовка к практике

Для этого часа создайте новый пустой проект без стартового контента.

* Скриптовый язык программирования. — Прим. ред.

Основы визуального программирования

Разработка на языке C++ требует использования интегрированной среды разработки (IDE, Integrated Development Environment), такой как Microsoft Visual Studio, в которой доступна разработка любых программных элементов от новых классов и геймплея до модификаций компонентов ядра движка. Блюпринты, напротив, представляют собой визуальную среду разработки скриптов. Хотя вы не можете использовать блюпринт для написания графического движка, вы можете использовать его и для создания ваших собственных классов и игрового функционала. Визуальная среда программирования, такая как блюпринт, не использует традиционную текстовую среду, но вместо этого предлагает ноды и провода. *Ноды (nodes)* — это визуальные выражения функций (участков кода, выполняющих определенные операции), переменных (для хранения данных), операторов (выполняющих математические операции) и условий (позволяющих проверять и сравнивать переменные).

ПРИМЕЧАНИЕ

Когда использовать C++

Язык C++ необходимо использовать, когда ваша игра требует 100% эффективности или некоторых модификаций компонентов базового рендеринга, физики, аудио или сетевого движка. Компания Epic предоставляет полный доступ ко всему исходному коду, используемому для создания всех базовых компонентов движка. Некоторые предпочитают использовать текстовую скриптовую среду или среду программирования, такую как C++. Если вы мало знакомы с программированием, работа в визуальной скриптовой среде, такой как блюпринт, — отличный способ изучить базовые концепции программирования, не изучая синтаксис.

Визуальное программирование позволяет художникам и дизайнерам разрабатывать игровой функционал, давая возможность программистам работать над более сложными задачами. Многие игры могут быть созданы полностью с помощью блюпринта, и, поскольку он производит компиляцию на уровне байткода, блюпринт-скрипты весьма эффективны. Вы можете использовать блюпринт для создания полноценных игр для всех платформ, поддерживаемых UE4.

ПРИМЕЧАНИЕ

Компиляция блюпринт-скриптов

Хотя блюпринт — это визуальная среда, блюпринты все же нуждаются в компиляции. Блюпринты компилируются до уровня байткода. Важно понимать следующие термины.

- ▶ **Компилятор.** Программное обеспечение, используемое для компилирования инструкций (исходного кода), написанных на языке программирования.

- ▶ **Компиляция.** Процесс преобразования инструкций в машинный язык (код), который может быть выполнен центральным процессором. Технические требования для выполнения компиляции варьируются в зависимости от аппаратного обеспечения и операционной системы.
 - ▶ **Байткод.** Скомпилированный исходный код, который обрабатывается виртуальной машиной, а не аппаратными средствами. Это означает, что исходный код может быть один раз скомпилирован и запущен на любом аппаратном обеспечении, имеющем виртуальную машину для обработки байткода.
 - ▶ **Виртуальная машина.** Программное обеспечение, транслирующее байткод в инструкции, которые аппаратные средства могут понять и обработать.
-

Изучение редактора блюпринтов

Визуальная среда разработки блюпринтов — ключевой компонент редактора UE4, и даже в основанных на C++ проектах вы, скорее всего, до некоторой степени будете использовать блюпринты. В UE4 существует пять типов блюпринтов.

- ▶ **Блюпринт уровня (Level Blueprint).** Этот блюпринт используется для управления глобальными событиями на уровне. На одном уровне может быть только один блюпринт уровня, и он автоматически сохраняется при сохранении уровня.
- ▶ **Блюпринт-класс (Blueprint class).** Это производный класс от другого существующего класса, созданного с помощью C++ или другого блюпринт-класса. Он используется для кодирования функционала актеров, помещенных на уровень.
- ▶ **Data-Only блюпринт (Data-Only Blueprint).** Этот блюпринт хранит только измененные свойства унаследованного блюпринта.
- ▶ **Блюпринт-интерфейс (Blueprint Interface, BPI).** Блюпринт-интерфейсы используются для хранения коллекции определенных пользователем функций, которые могут быть назначены другим блюпринтам. Блюпринт-интерфейсы позволяют другим блюпринтам обмениваться данными друг с другом.
- ▶ **Блюпринт-макрос (Blueprint Macros).** Блюпринт-макросы — независимые графы часто используемых последовательностей узлов, которые могут многократно использоваться другими блюпринтами. Блюпринт-макросы хранятся в библиотеке макрос-блюпринтов (Blueprint Macro Library).

Блюпринты уровней и блюпринт-классы — два наиболее часто используемых типа блюпринтов. В этом часе вы сосредоточитесь на знакомстве с редактором блюпринтов. В последующих часах вы узнаете больше о работе с блюпринт-классами.

ПРИМЕЧАНИЕ

Работа с блюпринтами

Ниже приведены некоторые базовые понятия, необходимые при разговоре о блюпринтах и программировании в целом.

- ▶ **Блюпринт.** Ассет блюпринт-класса, хранящийся на панели **Content Browser**.
- ▶ **Актор блюпринта.** Экземпляр ассета блюпринт-класса, помещенный на уровень.
- ▶ **Объект.** Переменная или коллекция переменных, например структуры данных или функции, хранящиеся в памяти.
- ▶ **Класс.** Шаблон для создания объектов, который хранит исходные значения, присвоенные переменным, а также функции и операции для работы с этими объектами.
- ▶ **Синтаксис.** В традиционном программировании и скриптовых средах синтаксис означает орфографическую и грамматическую структуру, которая ожидается компилятором языка, чтобы он смог корректно перевести код в машинный язык.

Интерфейс редактора блюпринтов

Чтобы открыть блюпринт уровня и увидеть интерфейс редактора (**Blueprint Editor**) выберите на панели инструментов редактора уровней пункт **Blueprints** ⇒ **Open Level Blueprint** (см. рис. 14.1). Интерфейс редактора блюпринтов и рабочий процесс просты для запоминания, но процессом разработки овладеть достаточно сложно. Несмотря на то что даже сейчас вам не нужно беспокоиться о синтаксисе в визуальной среде программирования, вам все же необходимо работать с логикой и порядком операций. Это требует практики в любой среде программирования.

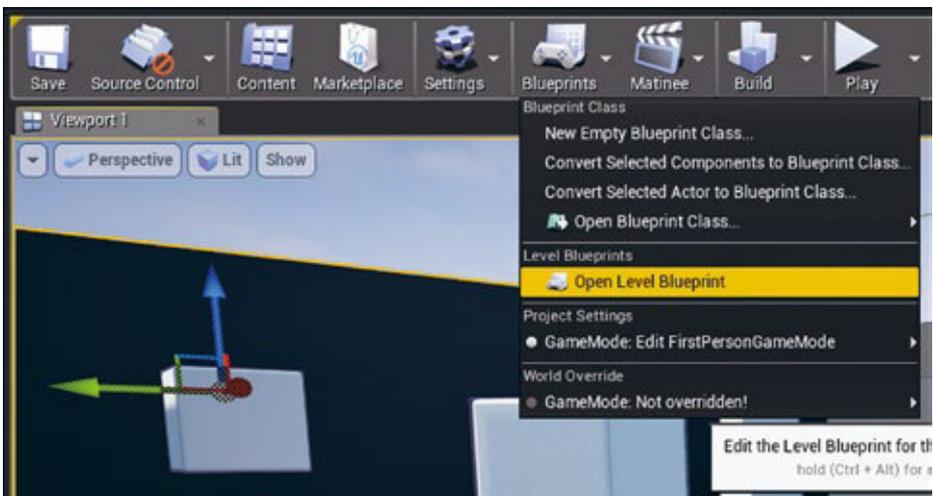


Рис. 14.1. Открытие блюпринта уровня в редакторе блюпринтов для текущего уровня

Интерфейс редактора блюпринтов содержит строку меню, панель инструментов для быстрого доступа к основным инструментам и операциям, панель **Event Graph** для проектирования скриптов, панель **Details** для отображения свойств выбранных в редакторе блюпринта объектов и панель **My Blueprint**, которая используется для управления и отслеживания графов узлов, функций, макросов и переменных, используемых в выбранном блюпринте. Возможности редактора блюпринтов выделены на рис. 14.2 и описаны ниже.

1. **Панель инструментов.** Панель инструментов содержит кнопки (с коротким описанием) для управления редактором блюпринтов.
2. **Панель My Blueprint.** Используется для управления графами, функциями, макросами и переменными, которые содержатся в вашем блюпринте.
3. **Панель Details.** Вы можете управлять свойствами добавленных в блюпринт компонентов, переменных и функций с помощью панели **Details**.
4. **Панель Event Graph (граф событий).** Эта панель используется для программирования базового функционала блюпринта.

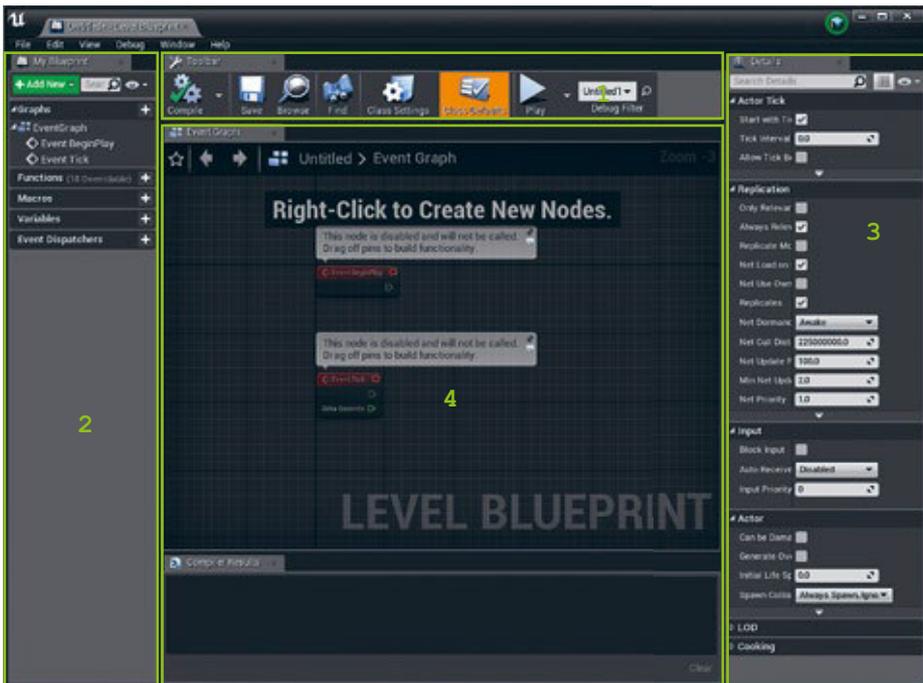


Рис. 14.2. Интерфейс редактора блюпринтов

ПРИМЕЧАНИЕ

Панель инструментов редактора блюпринтов

При работе с блюпринтами уровней, панель инструментов редактора блюпринтов не имеет функции сохранения или поиска на панели **Content Browser**, поскольку блюпринт уровня связан с уровнем. Чтобы сохранить блюпринт, просто сохраните уровень.

Панель инструментов редактора блюпринтов

Панель инструментов редактора блюпринтов включает в себя только пять инструментов. Два инструмента сосредоточены на кнопках **Compile** и **Play**. Вы нажимаете кнопку **Compile**, чтобы скомпилировать скрипт и увидеть проблемы в окне результатов компиляции внизу панели **Event Graph**. Кнопка **Play** в этом редакторе аналогична кнопке **Play** в редакторе уровней для запуска уровня. Обратите внимание, что на панели инструментов нет кнопки **Save**. Это связано с тем, что блюпринт уровня связан с уровнем, поэтому, если вы хотите сохранить блюпринт, просто сохраните уровень.

Панель инструментов редактора блюпринтов имеет следующие кнопки для управления блюпринтами.

- ▶ **Compile**. Компилирует блюпринт.
- ▶ **Search**. Открывает панель **Find Results** с поисковой строкой для поиска узлов в блюпринте.
- ▶ **Class Settings**. Показывает опции блюпринта на панели **Details**.
- ▶ **Class Defaults**. Отображает свойства блюпринта на панели **Details**.
- ▶ **Play**. Запускает уровень.

Панель My Blueprint

Панель **My Blueprint** отслеживает все графы узлов, функции, макросы и переменные, которые используются в блюпринте. Каждая категория разделена заголовком, и справа от каждого заголовка есть символ **+**, по которому можно щелкнуть, чтобы добавить элемент при необходимости. Вы можете использовать панель **My Blueprint**, чтобы добавить, переименовать или удалить все эти элементы.

Панель Event Graph

Панель **Event Graph** — это граф узлов по умолчанию, который используется для кодирования блюпринтов. **Event Graph** — это та область, в которой производится значительная часть работы при использовании редактора блюпринтов. Вы можете

добавлять больше графов узлов в существующий блюпринт при необходимости. Эта панель похожа на листок бумаги в клетку. Вы можете добавлять в блюпринт столько графов, сколько потребуется для сохранения организованности. В табл. 14.1 указаны комбинации клавиш, используемые при работе с узлами в **Event Graph**.

ТАБЛ. 14.1. Комбинации клавиш редактора блюпринтов

| Комбинация клавиш | Команда или действие |
|--|--|
| ПКМ по пустому пространству | Открывает контекстное меню блюпринта |
| ПКМ + перетаскивание по пустому пространству | Перемещает Event Graph |
| ПКМ по узлу | Отображает действия с узлом и контактами |
| ЛКМ по узлу | Выбирает узел |
| ЛКМ + перетаскивание узла | Перемещает узел |
| ЛКМ + перетаскивание по пустому пространству | Выбирает область |
| Ctrl+ЛКМ | Добавляет или удаляет текущий выбранный узел из группы выбранных узлов |
| Прокручивание колеса мыши | Приближает и отдаляет Event Graph |
| Home | Выравнивает Event Graph по центру экрана |
| Delete | Удаляет выбранные узлы |
| Ctrl+X | Вырезает выбранные узлы |
| Ctrl+C | Копирует выбранные узлы |
| Ctrl+V | Вставляет выбранные узлы |
| Ctrl+W | Копирует и вставляет выбранные узлы |

Контекстное меню блюпринта

Контекстное меню блюпринта (Blueprint Context Menu) — это одно из наиболее часто используемых меню при работе в редакторе блюпринтов. Если вы щелкните правой кнопкой мыши по пустой области или перетащите контакт, откроется контекстное меню блюпринта (см. рис. 14.3), которое позволяет добавлять в граф события, функции, переменные и условия. Это меню по умолчанию чувствительно к контексту, отчего оно отображает только действия, которые соответствуют выбранному элементу или элементу, от которого вы протягиваете контакт.

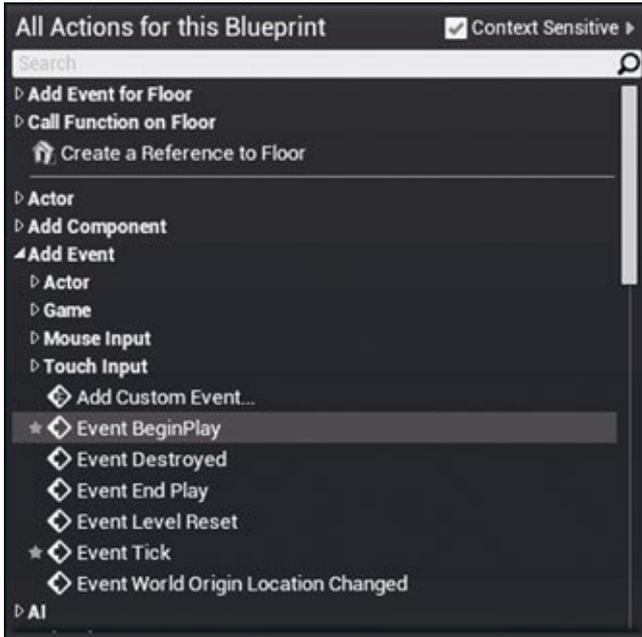


Рис. 14.3. Контекстное меню блюпринта

Ноды, провода и контакты

Можно представить процесс визуального программирования как работу электрической цепи. Красный нод события отправляет сигнал, идущий по проводам и запускающий выполнение любого нода, через который этот сигнал проходит. Когда нод получает сигнал, он извлекает данные, которые ему требуются в контактах данных в левой части нода. После чего нод выполняет свою операцию, отправляет сигнал дальше и возвращает результаты через контакты данных в правой части нода. Ниже приведена информация, которую нужно знать об этом процессе.

- ▶ Ноды — это визуальные представления событий, функций и переменных. Они имеют цветовой код, выражающий их назначение. Красный нод — это нод события, используемый для инициирования выполнения последовательности нодов. Синие ноды — это функции для выполнения определенных операций. Цветные овальные ноды, каждый из которых имеет только по одному контакту данных, представляют переменные.
- ▶ Контакты выполнения (ехес-контакты) «ввода» и «вывода» представляют собой белые треугольники, один из углов которых направлен вправо, они находятся в верхней части нода и указывают направление последовательности. Красный нод события имеет только один ехес-контакт «вывода»,

поскольку используется для инициирования последовательности, в то время как синие ноды имеют как ехес-контакты «ввода», так и «вывода» (в большинстве случаев), чтобы пропускать сигнал через себя.

- ▶ Контакты данных имеют цветовой код, основывающийся на типе используемых ими данных. Контакты данных в левой части нода извлекают данные, в то время как контакты данных в правой части нода возвращают данные.
- ▶ Провода соединяют ноды. Белые провода соединяют ехес-контакты «ввода» и «вывода», а цветные соединяют контакты данных. Цвет каждого провода отражает тип используемых им данных.

Чтобы установить соединение между контактом выполнения или контактом данных и проводом, щелкните по контакту и перетащите его в другой контакт того же типа. Чтобы оборвать провод, идущий в контакт или из контакта, нажмите клавишу **Alt** и щелкните по контакту. Нажмите клавишу **Ctrl**, затем щелкните по контакту или проводу, чтобы перетащить его в новый контакт.

Фундаментальные понятия разработки скриптов

Все среды программирования используют события, функции, переменные и операторы условий. Следующие страницы познакомят вас с этими базовыми понятиями.

События

Блюпринты в UE4 основаны на событиях. *Событие* (event) — это то, что происходит в процессе игры, от нажатия игроком клавиши на клавиатуре или попадания аватара в какое-либо помещение до столкновения актера с другим актером или начала игры. Большинство событий попадает под основные категории, описанные в табл. 14.2. События используются для инициирования последовательности в блюпринте. При запуске события из вывода ехес-контакта посылается сигнал, который проходит по проводам и обрабатывается всеми функциями, встреченными им на пути. Когда сигнал доходит до конца последовательности нодов, он пропадает.

Некоторые события требуют назначения им определенных актеров или компонентов, таких как события коллизий (Collision Events) — как правило, в одно и то же время транслируется более одного события. Например, если у вас на уровне есть актеры **Vox Trigger** и **Sphere Trigger** и необходимо, чтобы каждый из них реагировал, когда какой-либо актер перекрывает хотя бы один из них, вам необходимо назначить каждому из них событие коллизии **OnActorBeginOverlap**. Готовность назначать события определенным актерам позволяет вам программировать

реакцию каждого актера. Чтобы назначить актер на уровне событию коллизии, выберите актер и затем щелкните правой кнопкой мыши по пустому пространству на панели **Event Graph** блюпринта уровня. Затем в поисковой строке контекстного меню блюпринта введите словосочетание **on Actor begin** и выберите вариант **OnActorBeginOverlap** из списка, чтобы поместить нод события. Когда нод события коллизии помещен, вы увидите название актера, присвоенное ноду события, так вы узнаете, что актер был назначен ноду. Теперь при наложении оболочки коллизии актера будет запускаться этот нод события коллизии.

ТАБЛ. 14.2. Основные события

| Название события | Описание события |
|------------------|--|
| BeginOverlap | Срабатывает, когда оболочки коллизий двух актеров перекрываются. (Назначается актеру или компоненту.) |
| EndOverlap | Срабатывает, когда оболочки коллизий двух актеров перестают перекрываться. (Назначается актеру или компоненту.) |
| Hit | Срабатывает, когда оболочки коллизий двух актеров соприкасаются, но не перекрываются. (Назначается актеру или компоненту.) |
| BeginPlay | Срабатывает каждый раз, когда уровень загружается в память и запускается |
| EndPlay | Срабатывает, когда уровень закрывается |
| Destroyed | Срабатывает, когда актер удаляется из памяти |
| Tick | Срабатывает при каждом цикле центрального процессора |
| Custom | Работает так, как определил пользователь в зависимости от определенных потребностей |

ПРИМЕЧАНИЕ

Компоненты

Компоненты в UE4 — это подобъектные элементы, находящиеся в блюпринт-классах. Они раскрываются в 16-м часе «Работа с блюпринт-классами».

Редактор блюпринтов предоставляет предопределенные ноды событий, но вы можете также создать собственные пользовательские события, которые могут быть вызваны в любой точке последовательности блюпринта. Создание пользовательского события позволяет определять название события и любые проходящие при вызове события данные. На рис. 14.4 показаны два существующих события (EventBeginPlay

и `OnActorBeginOverlap`), которые вызывают пользовательское событие и передают переменную строкового типа. Пользовательское событие получает сигнал и использует функцию `Print String` для отображения строковых данных на экране. Пользовательские события могут помочь вам управлять блюпринтами и их хранением.

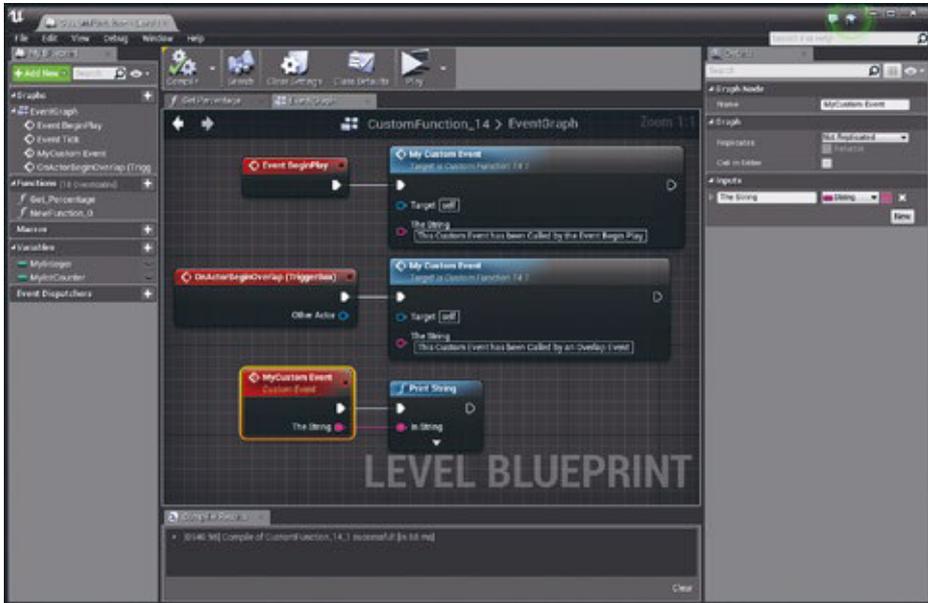


Рис. 14.4. Пользовательские события блюпринта

Чтобы создать пользовательское событие, щелкните правой кнопкой мыши по панели **Event Graph** и в поисковой строке контекстного меню блюпринта напечатайте слово **custom**. Затем выберите вариант **Custom Event** из списка и поместите нод пользовательского события. Переименуйте событие, щелкнув по его названию по умолчанию. Чтобы назначить событию переменную, выберите нод и на панели **Details** добавьте переменную. Создав пользовательское событие, вы можете вызвать его из другой последовательности, открыв контекстное меню и напечатав в поисковой строке название, присвоенное вами событию. Выберите пользовательское событие из списка, чтобы поместить нод, и затем вы можете соединить его проводами с последовательностью.

Функции

Функция — это фрагмент кода, выполняющий определенные операции. Она получает данные, хранящиеся в переменных, обрабатывает информацию и, если это требуется, возвращает результат. Редактор блюпринтов имеет полный набор

предопределенных функций, аналогичных таковым в любой другой среде программирования. Когда функция помещена на панель **Event Graph**, вы, как правило, можете увидеть целевой контакт данных в левой части нода функции. В блюпринте цель обычно представляет собой переменную, хранящую ссылку на актер или компонент актера на уровне, на котором будут выполняться операции этой функции. В примере функции, показанном на рис. 14.5, вы можете увидеть функцию `SetActorLocation`, которая используется для смены местоположения актера на уровне.

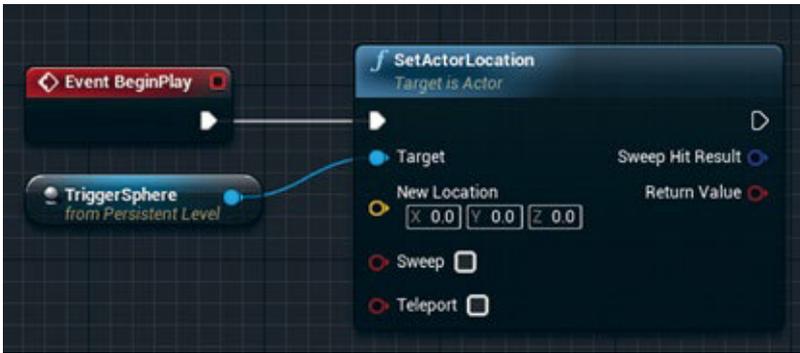


Рис. 14.5. Функция блюпринта

Несмотря на то что блюпринт имеет обширный список готовых к работе функций, вы также можете создавать собственные пользовательские функции в редакторе блюпринтов для отдельных блюпринтов, или даже создать свою библиотеку функций блюпринтов, которая позволит создавать коллекции функций, пригодных для многократного использования в любом блюпринте проекта. На рис. 14.6 показан пример пользовательской функции, созданной в редакторе блюпринтов. Эта пользовательская функция, которая называется `Get Percentage`, получает две переменных с плавающей запятой (A и B), где A — это общее значение, а B — текущее значение. Функция делит текущее значение (B) на общее значение (A) и умножает результат на 100, после чего возвращает итог в процентах в виде числа с плавающей запятой.

Создав пользовательскую функцию, вы можете использовать ее столько раз, сколько захотите, просто перетащив ее из панели **My Blueprint** в граф нодов. Чтобы создать пользовательскую функцию в блюпринте, просто щелкните по символу **+** в строке **Functions** на панели **My Blueprint**. Таким образом, вы получите граф нодов, определенный для вашей функции. В графе нодов вы можете увидеть два фиолетовых нода для присвоения переменных ввода и вывода. Вы можете определить переменную ввода/вывода для функции, выбрав ноды ввода или вывода в пользовательском графе функции и на панели **Details**,

при необходимости вы можете создать переменные с различными типами данных. Создав переменные ввода и вывода для функции, вы можете закриптовать последовательность, как вы сделали бы в любом другом графе узлов, но вы должны будете соединить последовательность проводами с узлами ввода и вывода, когда закончите.

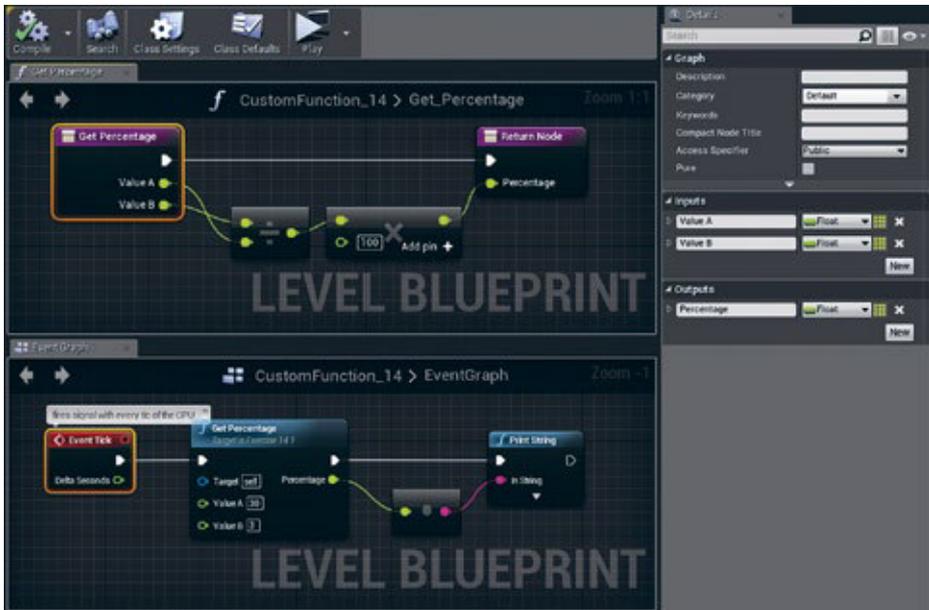


Рис. 14.6. Пользовательская функция блюпринта

ПРИМЕЧАНИЕ

Пользовательские функции

Вы также можете создать пользовательскую функцию, выбрав последовательность уже помещенных узлов и щелкнув правой кнопкой мыши по одному из узлов и выбрав вариант **Collapse to Function** из открывшегося меню. Так у вас появится новая функция, которую можно переименовать. Если вы обычно повторяете последовательность из трех или более предопределенных функций в блюпринте, весьма вероятно, что вы извлечете пользу из «схлопывания» последовательности в пользовательскую функцию.

ПОПРОБУЙТЕ САМИ

Добавьте событие

Выполните следующие шаги, чтобы добавить событие **Event BeginPlay** и используйте функцию **Print String**, чтобы вывести текст на экран.

1. В меню редактора уровней выберите пункт **File** ⇒ **New**, чтобы создать новый уровень по умолчанию.
2. На панели инструментов редактора уровней выберите пункт **Blueprints** ⇒ **Open Level Blueprint**.
3. Выберите события **BeginPlay** и **Event Tick**, которые уже добавлены, и нажмите клавишу **Delete**.
4. Щелкните правой кнопкой мыши по панели **Event Graph** и выберите вариант **Event BeginPlay**. (Используйте поисковую строку, если затрудняетесь найти его.)
5. Щелкните по ехес-контакту в событии **Event BeginPlay**, перетащите его вправо и отпустите.
6. Щелкните правой кнопкой мыши по панели **Event Graph** и в поисковой строке контекстного меню напечатайте словосочетание **print string**.
7. Выберите функцию **Print String**, чтобы поместить для нее нод.
8. Справа от контакта данных **String**, в строке ввода с уже имеющейся надписью **Hello**, напечатайте **Hello level**.
9. Щелкните по клавише **Compile** на панели инструментов редактора блюпринтов и запустите уровень.
10. Каждый раз при запуске уровня вы будете видеть надпись **Hello level** в левом верхнем углу вьюпорта уровня в течение нескольких секунд, после чего надпись будет исчезать.

ПРИМЕЧАНИЕ

Функция Print String

Использование функции **Print String** — не самый подходящий способ коммуникации с игроками. Эта функция обычно используется при разработке в качестве инструмента отладки для коммуникации с происходящим в блюпринте. Если вы хотите отправлять сообщения игрокам, вам необходимо использовать класс **Blueprint HUD** или научиться пользоваться редактором **Unreal Motion Graphics**. См. 22-й час, «Работа с UMG».

Переменные

Переменные хранят данные различных типов. Когда переменная объявлена (создана), компьютер резервирует определенный объем памяти в зависимости от ее типа данных. Эта память далее используется для хранения или извлечения

информации из этой ячейки памяти. Различные типы переменных используют различный объем памяти. Некоторые переменные хранят маленький объем информации, такой как бит, а некоторые хранят значительные объемы, такие как целый актер. В редакторе блюпринтов переменные имеют цветовой код, чтобы вы могли быстро определить, какой тип переменной вам потребуется при работе с функциями.

В табл. 14.3 перечислено большинство часто используемых типов переменных, их цветовой код и тип данных, хранящихся в них.

ТАБЛ. 14.3. Основные типы данных

| Тип переменной | Цвет | Описание |
|--|-----------|--|
| Boolean или bool (логический или булев) | красный | Хранит значение 0 («выключено» или ложь) или 1 («включено» или истина) |
| Integer или int (целочисленный) | циановый | Хранит любое круглое число, такое как 1 , 0 , -100 или 376 |
| Float (число с плавающей запятой) | зеленый | Хранит любое значение с десятичной дробью, такое как 1,0 , -64,12 или 3,14159 |
| String (строковый) | маджента | Хранит текст |
| Vector (векторный) | золотой | Хранит три числа с плавающей запятой — X, Y и Z, например, 100,5 , 32,90 , 100,0 |
| Rotator (ротатор) | пурпурный | Является вектором, хранящим три числа с плавающей запятой, где X — тангаж, Y — рыскание и Z — крен |
| Transform (трансформация) | оранжевый | Представляет собой структуру, хранящую вектор для определения местоположения, ротатор для определения ориентации и вектор для определения масштаба |
| Object (объект) | синий | Представляет собой актер на уровне и хранит все свойства в памяти |

ПРИМЕЧАНИЕ

Что подразумевается под словом «структура»?

Структура представляет собой коллекцию переменных любого типа, являющуюся единой переменной. Переменные Vector и Rotator технически являются структурами, поскольку хранят в себе три отдельных переменных типа Float. Вы можете создавать собственные структуры в UE4, но вам потребуется изучить эту сложную тему, чтобы уверенно пользоваться редактором блюпринтов.

Чтобы объявить переменную, щелкните по символу + в графе **Variables** на панели **My Blueprint** и дайте новой переменной имя. Затем на панели **Details** вы можете установить тип переменной и ее значение по умолчанию. Чтобы установить значение по умолчанию, вам необходимо один раз скомпилировать блюпринт сразу после объявления переменной. После того как вы объявили переменную, дали ей имя и присвоили значение, наиболее часто выполняемой операцией является присвоение и получение данных для переменной. Операция получения (Get) извлекает значение, хранящееся в переменной, а операция присвоения (Set) сохраняет значения. На рис. 14.7 показаны ноды Get и Set для основных типов переменных.



Рис. 14.7. Ноды переменных Get и Set

ПРИМЕЧАНИЕ

Списки переменных

Переменная любого типа в блюпринте может хранить одно значение или массив значений. Переменная, конвертированная в массив, хранит список данных ее типа. Вы можете использовать группу функций для управления массивами переменных — присваивать, получать, удалять или добавлять элементы массива.

ПОПРОБУЙТЕ САМИ

Объявите переменные

Используя блюпринт уровня из предыдущей рубрики «Попробуйте сами», выполните следующие шаги, чтобы объявить целочисленную переменную, задать ей имя и исходное значение.

1. На панели **My Blueprint** щелкните по символу + в графе **Variables**, чтобы добавить новую переменную. Назовите ее **MyInteger**. На панели **Details** установите тип переменной **integer**.
2. Чтобы установить значение по умолчанию новой переменной **MyInteger**, щелкните по кнопке **Compile** на панели инструментов редактора блюпринтов. Затем на панели **Details** в разделе **Default Value** установите исходное значение переменной **100**.
3. Чтобы добавить событие **Event Tick**, щелкните правой кнопкой мыши по панели **Event Graph** и в контекстном меню блюпринта выберите вариант **Event Tick**, после чего появится нод события.
4. Нажмите комбинацию клавиш **Ctrl+W**, чтобы копировать и вставить функцию **Print String** из предыдущего упражнения «Попробуйте сами» и соедините ее с ехес-контактом вывода **Event Tick**.
5. На панели **My Blueprint** в графе **Variables** щелкните по переменной **MyInteger** и перетащите ее на панель **Event Graph**. Отпустите кнопку мыши, после чего появится диалоговое окно, предлагающее присвоить переменной значение или получить его. Выберите вариант **Get**, чтобы поместить переменную в граф.
6. Щелкните по контакту данных целочисленной переменной и перетащите его в контакт строковых данных функции **Print String**. Редактор автоматически добавит нод преобразования, чтобы конвертировать целочисленную переменную в строковую. По окончании выполнения задания ваш блюпринт уровня должен выглядеть, как показано на рис. 14.8.
7. Скомпилируйте скрипт и запустите уровень. Вы увидите, что целочисленное значение по умолчанию, повторяясь, отображается в левой части вьюпорта уровня.

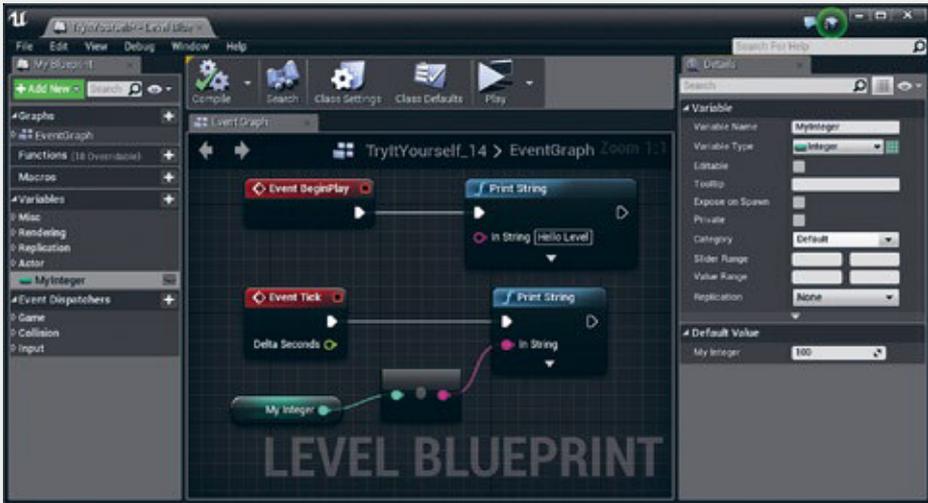


Рис. 14.8. События Event BeginPlay и Event Tick, добавленные на панель Event Graph блюпринта

ПРИМЕЧАНИЕ

Событие Event Tick

По умолчанию событие Event Tick выполняется после каждого рендера кадра (Tick Interval 0). Контакт данных Delta Seconds в ноде Event Tick возвращает количество времени, которое отнял рендер каждого кадра в процессе выполнения этого цикла. Если вы ранее работали в среде программирования, использующей игровые циклы, вы можете узнать в событии Event Tick UE-эквивалент этих циклов. Вы можете изменить интервалы обновлений Event Tick, щелкнув по иконке **Class Settings** на панели инструментов блюпринта. После этого вы сможете изменить настройки на панели **Details** в разделе **Actors Tick/Tick Interval (Sec)**.

Операторы и условия

Операторы и условия можно найти в контекстном меню блюпринта в разделе **Flow Control**. *Операторы* — это математические операции, такие как сложение, вычитание, умножение и деление. Операторы позволяют изменять значение численных переменных, таких как числа с плавающей запятой, целые числа и векторы. *Условные выражения* позволяют проверять или сравнивать выражения переменных и затем реагировать нужным вам образом. Например, вы можете проверить, равна ли одна переменная другой или является ли одна переменная больше другой.

ПОПРОБУЙТЕ САМИ

Используйте условия и операторы и установите переменную

Используя блюпринт уровня из предыдущей рубрики «Попробуйте сами», выполните следующие шаги, чтобы использовать математические операторы и установить переменную для увеличения значения переменной `MyInteger`.

1. Отсоедините переменную `MyInteger` от нода преобразования, зажав клавишу **Alt** и щелкнув по контакту данных **MyInteger**.
2. Щелкните по контакту данных **MyInteger**, перетащите его вправо и отпустите. В поисковой строке контекстного меню блюпринта введите символ **+**. В разделе **Math/Integer** выберите вариант **Integer + Integer**, чтобы добавить нод операции сложения целых чисел. Нод операции **+** будет помещен с соединением проводами. Установите нижний контакт целого числа в значение **1**.
3. Перетащите переменную **MyInteger** из панели **My Blueprint** в граф и выберите вариант **Set**, чтобы поместить функцию присвоения значения переменной.
4. Соедините проводами ехес-контакт вывода **Event Tick** и ехес-контакт ввода функции **Set MyInteger**.
5. Соедините проводами ехес-контакт вывода нода **Set MyInteger** и ехес-контакт ввода функции **Print String**.
6. Соедините проводами контакт данных целого числа нода функции **Set** с нодом **Convert to String**, который уже соединен с функцией **Print String**. Когда вы закончите, ваш блюпринт уровня должен выглядеть, как на рис. 14.9.
7. Скомпилируйте и запустите уровень. Вы увидите, как значение `MyInteger` в левой части выюпорта увеличивается на **1**.

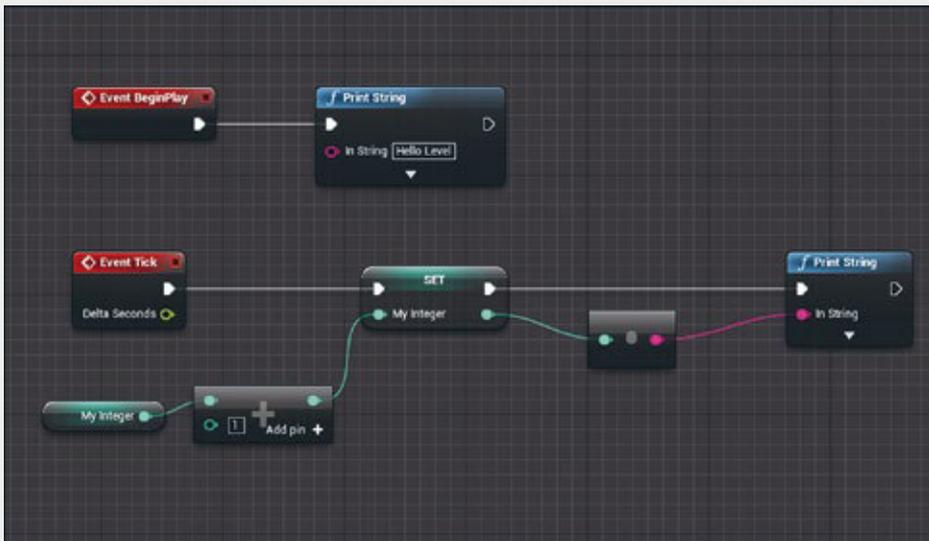


Рис. 14.9. Пример получения в блюпринте уровня целочисленной переменной, добавления значения **1** и сохранения результата в переменной

Организация и комментирование скриптов

В любой скриптовой среде организация и комментирование одинаково важны и при проверке скрипта, написанного вами месяц назад, и при работе над скриптами, написанными вашими коллегами из команды разработки. Хорошо организованные скрипты с комментариями уменьшают время разработки. Как было описано в предыдущих разделах, редактор блюпринтов содержит несколько инструментов, чтобы помочь вам сохранить порядок.

Комментирование нодов

Комментирование нодов позволяет оставлять заметки к любому ноду. Просто щелкните правой кнопкой мыши по названию помещенного нода, и вы увидите всплывающий блок комментирования нода (см. рис. 14.10), или наведите курсор на нод, и через некоторое время появится всплывающий блок комментирования.

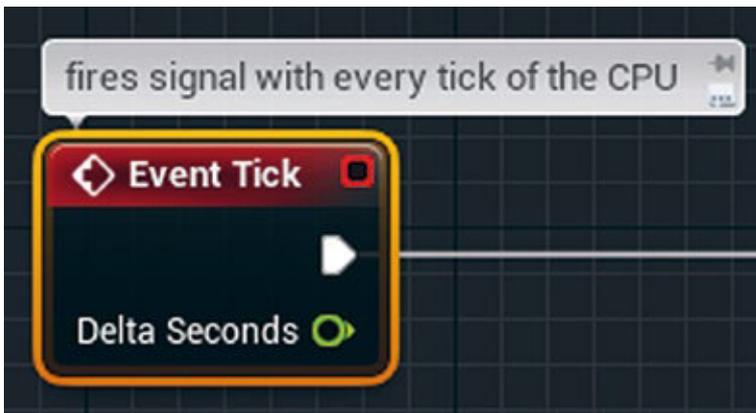


Рис. 14.10. Пример комментирования нода

Блок комментирования

Блоки комментирования (см. рис. 14.11) позволяют помещать выбранные ноды в блок и добавлять текстовые комментарии. Еще одно преимущество блока комментирования заключается в том, что при его перемещении вместе с ним перемещаются все ноды внутри него. Чтобы добавить блок комментирования к выборке нодов, нажмите клавишу С.

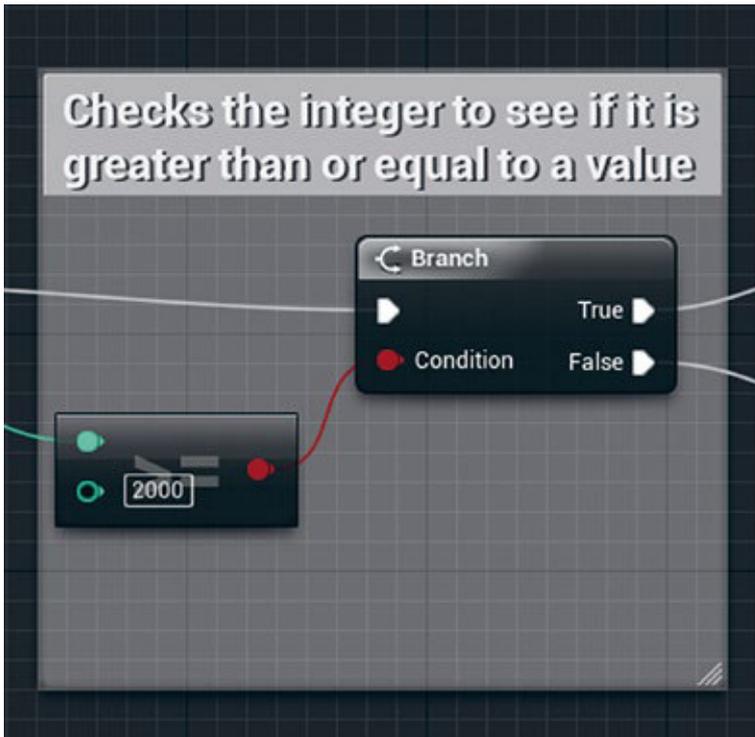


Рис. 14.11. Пример блока комментирования

Ноды перенаправления

По мере усложнения ваших скриптов появляется все больше путаницы в проводах. Нод перенаправления может помочь вам контролировать помещение проводов, как показано на рис. 14.12. Чтобы добавить нод перенаправления, щелкните правой кнопкой мыши по пустому пространству панели **Event Graph**, в поисковой строке контекстного меню напечатайте слово **reroute** и выберите вариант **reroute** из списка, чтобы поместить нод.

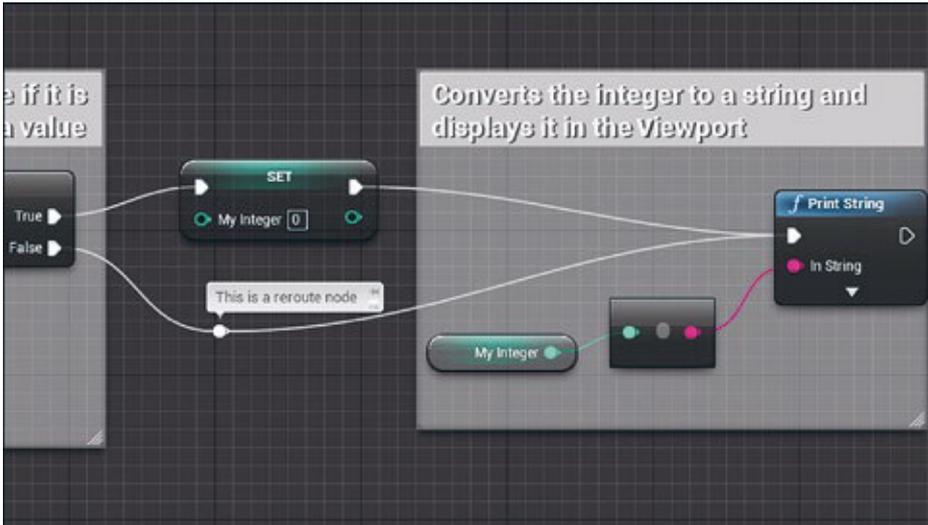


Рис. 14.12. Нод перенаправления для управления проводами

Резюме

В этом часе были представлены два метода программирования в UE4, фундаментальные концепции программирования и интерфейс редактора блюпринтов. Вы научились добавлять события и функции, а также узнали, как объявлять переменные, присваивать и получать значения. Эти ключевые навыки нужны при работе с любым блюпринтом в UE4.

Вопросы и ответы

Вопрос: Когда я пытаюсь добавить второй нод события (BeginPlay), редактор показывает, что первый нод уже помещен на панель Event Graph. Почему это происходит?

Ответ: Некоторые события, такие как Event Tick и BeginPlay, могут иметь только один экземпляр в блюпринте. Использование нескольких панелей Event Graph для одного блюпринта могло бы привести к проблемам, но этого не происходит. Если вы хотите направить событие из одного графа в другой, вы можете создать пользовательское событие, которое вызывает Event Tick или BeginPlay каждый раз, когда оно посылает сигнал.

Вопрос: Как выбрать имя для переменной?

Ответ: Вы можете дать переменной любое имя, которое захотите. Постарайтесь выбрать короткое, описательное имя, которое упростит узнавание

переменной. Лучше всего установить соглашение об именах, которое можно применять ко всем блюпринтам в проекте в целях последовательности.

Вопрос: Можно ли поменять имя переменной, если у переменной уже есть имя?

Ответ: Да, вы можете изменить имя переменной в окне My Blueprint или можете выбрать переменную и изменить ее имя на панели Details блюпринта. Изменение имени обновляет все экземпляры переменной в блюпринте.

Вопрос: Можно ли изменить тип переменной после ее создания и использования в графе событий?

Ответ: Да, вы можете изменить тип уже созданной и используемой переменной, но это повлияет на блюпринт. Помните, что контакты данных в функциях нуждаются в определенных типах данных, и, если вы измените тип переменной, вы можете оборвать какой-либо провод, соединенный с контактами данных переменных. В таком случае вам понадобится вернуться и настроить скрипт вручную.

Вопрос: Могу ли я использовать одни и те же скрипты блюпринта уровня на другом уровне?

Ответ: Нет. Хотя вы можете копировать и вставлять последовательности событий из одного блюпринта в другой, все переменные должны быть заново созданы в новом скрипте блюпринта. Также многие последовательности событий и действия в блюпринте уровня связаны с определенными актерами в этом уровне, которых не существует в новом уровне. В этом заключается преимущество использования актером блюпринт-классов, которые вы изучите в следующих часах.

Семинар

Закончив этот час, попытайтесь ответить на следующие вопросы.

Контрольные вопросы

1. Истинно или ложно высказывание: блюпринт можно использовать для того, чтобы переписать базовый движок рендеринга UE4.
2. Истинно или ложно высказывание: использование функции Print String — не лучший способ коммуникации с игроками.
3. Истинно или ложно высказывание: скрипты блюпринта компилируются в байткод.

4. Истинно или ложно высказывание: вы можете иметь более одного события BeginPlay в одном скрипте блюпринта.
5. Что такое массив?
6. Истинно или ложно высказывание: комментирование скриптов — пустая трата времени.

Ответы

1. Ложь. Если вам нужно модифицировать компоненты основного движка для вашей игры, вам необходимо программировать на C++.
2. Истина. Функция Print String должна использоваться только для отладки.
3. Истина. Блюпринты компилируются в байткод.
4. Ложь. Вы можете иметь только одно событие BeginPlay в блюпринте, но вы можете использовать другое событие, чтобы передать сигнал ноду Sequence для разделения сигнала.
5. Массив — это переменная, хранящая набор значений одного типа.
6. Ложь. Вы всегда должны комментировать свои скрипты.

Упражнение

Продолжая последнее упражнение «Попробуйте сами», добавьте вторую целочисленную переменную, которая меняет добавляемое к целочисленной переменной MyInteger значение с каждым циклом. Затем используйте условие в последовательности Event Tick, проверяющее значение целочисленной переменной MyInteger и приравняющее MyInteger к 0, когда оно достигает значения 2000 или выше. Затем закончите последовательность пользовательским событием, которое при вызове воспроизводит вывод строки. Пример показан на рис. 14.13.

1. Откройте блюпринт уровня, над которым вы работали в этом часе.
2. Объявите новую переменную, назовите ее **MyIntCounter**, смените ее тип на **integer** и дайте ей значение по умолчанию **5**.
3. Добавьте переменную **MyIntCounter** в граф события и соедините ее с нодом **+**.
4. После того как вы установите значение **SET** для переменной **MyInteger**, проверьте, является ли значение бóльшим или равным (**>=**) **2000**.

- В поисковой строке контекстного меню блюпринта введите **integer >=** и выберите вариант **integer >= integer**, чтобы поместить нод. Этот нод возвращает значение **0** (ложь) или **1** (истина).
5. Установите контакт данных целого числа В в ноде **>=** в значение **2000**.
 6. Проверьте, является ли условие истинным или ложным, используя нод Branch. Щелкните и перетащите красный контакт булевых данных в нод **>=**, чтобы вызвать контекстное меню блюпринта. В поисковой строке контекстного меню введите **Branch** и выберите этот вариант из списка, чтобы поместить нод.
 7. Соедините ехес-контакт вывода нода SET MyInteger с ехес-контактом нода Branch.
 8. Перетащите ехес-контакт вывода True из нода Branch, чтобы вызвать контекстное меню блюпринта. В поисковой строке введите **set myinteger** и выберите вариант **Set MyInteger**, чтобы поместить нод.
 9. В ноде SET MyInteger введите **0** в текстовую строку рядом с контактом данных MyInteger.
 10. Теперь создайте пользовательское событие. Под последовательностью в графе события щелкните правой кнопкой мыши по пустой области, чтобы вызывать контекстное меню. В поисковой строке введите **custom** и выберите вариант **Add Custom Event** из списка, чтобы поместить нод события. Переименуйте событие на **MyCustomEvent**.
 11. Перетащите ехес-контакт вывода нода события MyCustomEvent и в поисковой строке контекстного меню блюпринта введите **print**. Выберите вариант **Print String** из списка, чтобы поместить нод.
 12. Из панели **My Blueprint** перетащите переменную **MyInteger** в контакт данных переменной **In String** в ноде Print String. Блюпринт автоматически помещает переменную и добавляет нод конвертации.
 13. Соедините ехес-контакт False с нодом Branch в конце первой последовательности. Перетащите его, чтобы вызвать контекстное меню. В поисковой строке введите **mycustomevent** и выберите вариант **MyCustomEvent** из списка, чтобы поместить функцию.
 14. Перетащите провод из ехес-контакта вывода в нод SET MyInteger, соединенный с нодом Branch, и соедините его с синей функцией

MyCustomEvent. После того как вы закончите, ваш блюпринт уровня должен выглядеть, как на рис. 14.13.

15. Сохраните уровень и запустите его.

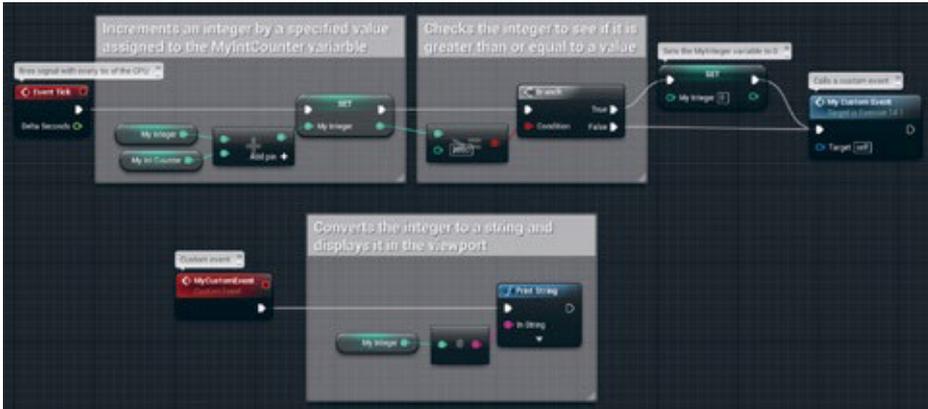


Рис. 14.13. Пример скрипта из упражнения