

ГЛАВА 1

ЗНАКОМСТВО С BLUEPRINT EDITOR

Добро пожаловать в удивительный мир создания игр на Unreal Engine 5 при помощи языка визуального программирования Blueprints, который компания Epic Games разработала специально для Unreal Engine.

Прежде чем приступить к обсуждению блюпринтов, необходимо подготовить среду разработки. Unreal Engine 5 можно скачать бесплатно. Мы установим его и создадим новый проект, после чего поговорим об основных понятиях, принятых в языке Blueprints, а также познакомимся с панелями Blueprint Editor (Редактор блюпринтов).

В этой главе мы рассмотрим следующие темы:

- Установка Unreal Engine
- Создание новых проектов и работа с шаблонами
- Визуальное программирование на языке Blueprints
- Интерфейс Blueprint Class Editor (Редактор блюпринтов-классов)
- Добавление компонентов в блюпринт

Установка Unreal Engine

Чтобы использовать Unreal Engine, сначала необходимо установить приложение Epic Games Launcher (лончер Epic Games):

1. Перейдите на сайт <https://www.unrealengine.com>.
2. Зарегистрируйтесь и скачайте приложение Epic Games Launcher.
3. Установите и запустите его.

4. Перейдите на вкладку **Unreal Engine** в левой части окна.
5. Перейдите на вкладку **Library** (Библиотека) в верхней части окна.
6. Нажмите на кнопку **+** рядом с надписью **ENGINE VERSIONS** (Версии движка), чтобы добавить в **Epic Games Launcher** одну из версий Unreal Engine. Можно добавить любую, например самую новую версию.
7. Нажмите на кнопку **Install** (Установить). Приложение начнет скачивать нужные файлы, что может занять довольно много времени.
8. Нажмите на кнопку **Launch** (Запустить), чтобы запустить установленную версию Unreal Engine. На одном компьютере можно установить сразу несколько версий движка и сделать одну из них текущей. Кнопка **Launch**, расположенная в правом верхнем углу окна, запускает текущую версию.

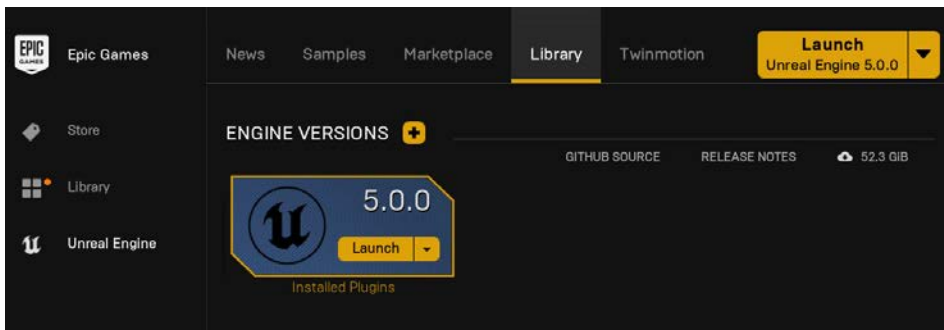


Рис. 1.1. Запуск Unreal Engine

Система визуального программирования Blueprints — давно существующая и стабильно работающая технология. В этой книге обсуждается версия 5.0.0, но все примеры будут отлично работать и на более поздних версиях.

Новые проекты и работа с шаблонами

При запуске редактора Unreal Engine на экране появляется **Unreal Project Browser** (Просмотр проектов Unreal). В левой его части находятся кнопки. Самая верхняя из них — **RECENT PROJECTS** (Недавние проекты) — позволяет открыть список уже начатых проектов, а остальные — списки шаблонов из соответствующих категорий. Шаблоны используются для создания новых проектов.

Например, если выбрать категорию **GAMES** (Игры), окно будет выглядеть так, как показано ниже.

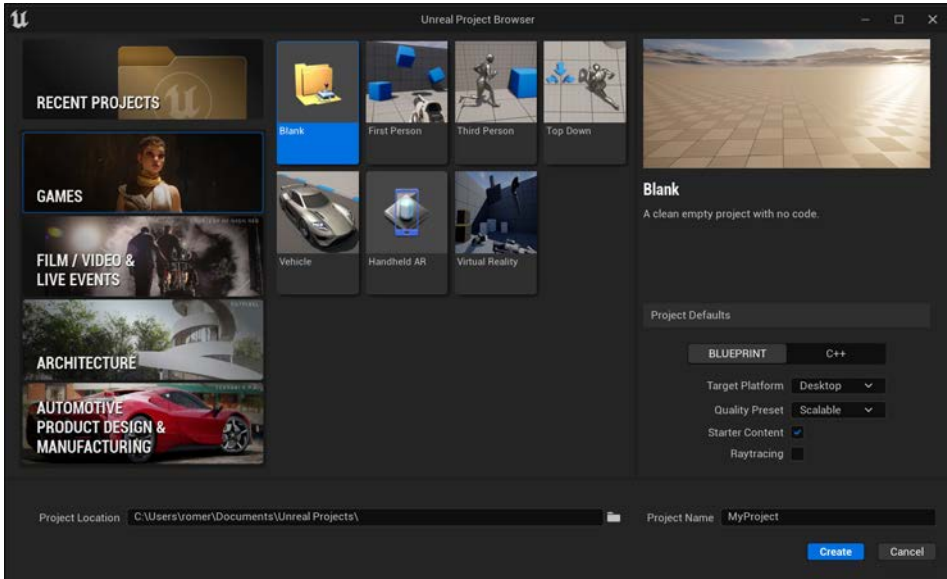


Рис. 1.2. Шаблоны в категории Games

Каждый шаблон представляет собой контейнер с базовым набором файлов и заготовкой уровня, с которой удобно начать работу над проектом того или иного типа. Кроме того, шаблоны могут быть полезны для быстрого создания прототипов, а также изучения стандартной механики проекта. Однако можно начать и с пустого шаблона, то есть с нуля. В категории **GAMES** имеются следующие шаблоны:

- **First Person** (Первое лицо). Предназначен для создания игр от первого лица, в которых на экране видны только руки персонажа с каким-нибудь оружием (в данном случае с автоматом, который стреляет небольшими шариками). Управлять движением персонажа можно при помощи клавиатуры, геймпада, джойстика или сенсорного устройства.
- **Handheld AR** (Дополненная реальность). Предназначен для создания игр с дополненной реальностью для устройств с iOS и Android. В шаблоне уже заложена логика для включения и выключения дополненной реальности, а также имеется несколько примеров кода для обнаружения попаданий и оценки освещенности.
- **Third Person** (Третье лицо). Предназначен для создания игр от третьего лица, в которых камера находится сзади и немного выше персонажа и следует за ним, то есть игрок видит анимацию бега, ходьбы, прыжков и т. д. Управлять

движением персонажа по уровню можно при помощи клавиатуры, геймпада, джойстика или сенсорного устройства.

- **Top Down (Вид сверху)**. Предназначен для создания игр, в которых камера находится высоко над персонажем, а для управления используется мышь или сенсорное устройство. Игрок показывает, куда нужно перемещаться, а также применяет различные средства навигации для обхода препятствий. Такое расположение камеры часто используется в ролевых играх.
- **Virtual Reality (Виртуальная реальность)**. Предназначен для создания игр на основе виртуальной реальности и содержит функции, необходимые для телепортации, работы с интерактивными и захватываемыми объектами, а также управления камерой. Шаблон имеет уровень, по которому игрок может перемещаться, и содержит объекты, которые можно брать в руки и что-то с ними делать.
- **Vehicle (Автомобиль)**. Предназначен для создания игр, в которых персонаж может выбирать тип подвески автомобиля и управлять им. На уровне уже есть простой трек и препятствия.

В нижнем правом углу окна **Project Browser** находится область **Project Defaults** (Настройки по умолчанию), в которой перечислены настройки, доступные для выбранного шаблона. В примерах из этой книги мы будем использовать настройки, которые показаны на скриншоте и перечислены ниже. В ходе работы над проектом любые из них можно изменить.

- **Blueprint/C++**. Шаблон может быть написан на языке C++ или только с помощью Blueprints, как мы и сделаем в этой книге. При создании игр на Unreal Engine 5 можно использовать Blueprints, C++ или и то и другое вместе. Вы можете добавлять код на C++ в проекты на Blueprints и Blueprints в проекты на C++.
- **Target Platform (Целевая платформа)**. Если игра предназначена для настольных компьютеров и игровых консолей, следует выбрать вариант **Desktop** (Настольный компьютер), а если для мобильных устройств — вариант **Mobile** (Мобильное устройство). В этой книге мы будем рассматривать только настольные компьютеры.
- **Quality Preset (Пресет качества)**. Влияет на производительность проекта. При выборе варианта **Scalable** (Масштабируемое) некоторые сложные функции недоступны, тогда как вариант **Maximum** (Максимальное) обеспечивает доступ ко всем функциям, которые поддерживаются целевой платформой. В этой книге мы будем использовать вариант **Scalable**.
- **Starter Content (Стартовый контент)**. Если поставить этот флажок, в проект будет входить стартовый контент, то есть некоторый набор простых мешей, материалов и систем частиц. Во всех примерах из этой книги стартовый контент понадобится.

- **Raytracing** (Трассировка лучей). Если поставить этот флажок, в проекте будет использоваться трассировка лучей в реальном времени. Эта функция сильно влияет на производительность, и в нашем проекте мы не будем ее включать.

Выберите шаблон **Third Person**, укажите папку для хранения проекта и введите его имя в поле **Name** (Имя). В области **Project Defaults** выберите указанные выше значения настроек, после чего нажмите на кнопку **Create** (Создать). После загрузки нового проекта открывается **Unreal Engine Level Editor** (Редактор уровней Unreal Engine), который показан на скриншоте ниже.

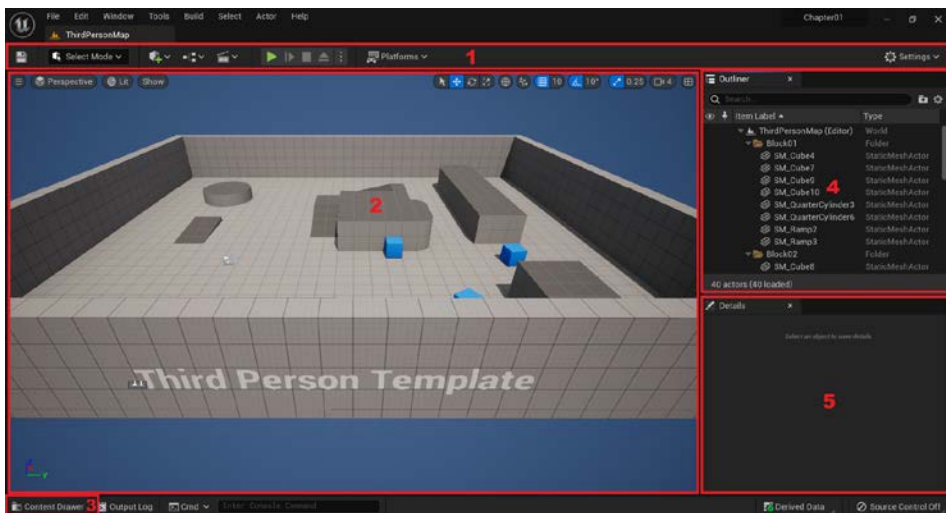


Рис. 1.3. Level Editor (Редактор уровней)

В этом окне важно обратить внимание на следующие панели:

1. **Toolbar** (Панель инструментов). Находится в верхней части окна **Level Editor** и предназначена для вызова часто выполняемых функций. Для этого на ней имеются кнопки, которые разделены на четыре группы. Первая слева группа позволяет сохранить проект и добавить к нему различные объекты и код, вторая служит для переключения между режимами редактирования, третья используется для доступа к настройкам целевой платформы, а также позволяет проиграть текущий уровень. И наконец, четвертая группа, которая состоит из одной кнопки **Settings** (Настройки) и расположена у правого края окна, необходима для работы с настройками проекта.
2. **Viewport** (Область просмотра). Находится в центральной части окна **Level Editor** и предназначена для просмотра текущего уровня и добавления на

него объектов. Перемещая мышь с нажатой правой кнопкой, можно поворачивать изображение относительно камеры, а при помощи клавиш **W**, **A**, **S** и **D** — перемещаться по уровню.

3. **Content Browser** (Просмотр контента). Используется для добавления в проект **ассетов**, как в Unreal Engine называются элементы контента: **материалы**, **статические меши**, **блюпринты** и т. п. Чтобы открыть эту панель, нужно нажать на кнопку **Content Drawer** (Источник контента) в нижнем левом углу окна **Level Editor**. Если затем перетащить какой-либо из ассетов с панели **Content Browser** на панель **Viewport**, копия выбранного ассета возникнет в соответствующем месте на создаваемом уровне.
4. **Outliner** (Каталог). Находится в правой части окна **Level Editor**, содержит список объектов, размещенных на создаваемом уровне.
5. **Details** (Свойства). Находится в правой части окна **Level Editor**, под панелью **Outliner**. Здесь отображаются редактируемые свойства объекта, выбранного на панели **Viewport**.

Кратко познакомившись с **Level Editor**, поговорим о визуальном программировании с **Blueprints**.

Визуальное программирование на языке **Blueprints**

Прежде всего нужно немного рассказать о том, что такое блюпринт.

Слово **блюпринт** (**Blueprint**) имеет в мире Unreal Engine два значения. Во-первых, **Blueprints** — это язык визуального программирования, созданный компанией Epic Games специально для Unreal Engine, во-вторых — это новый тип игровых объектов, которые можно создать при помощи этого языка.

Чаще всего используются блюпринты двух типов: **блюпринты уровней** (**Level Blueprint**) и **блюпринты-классы** (**Blueprint Class**). Для каждого уровня в игре создается блюпринт уровня. Нельзя создать блюпринт этого типа отдельно от игры. Блюпринты-классы, напротив, служат для программирования интерактивных объектов, которые можно поместить на любой уровень в любом количестве.

Открытие **Level Blueprint Editor**

Чтобы открыть **Level Blueprint Editor** (Редактор блюпринтов уровней), нажмите на кнопку **Blueprints** (Блюпринты), которая находится в крайней левой группе кнопок на панели инструментов окна **Level Editor**. В открывшемся меню выберите пункт **Open Level Blueprint** (Открыть блюпринт уровня), как показано ниже.

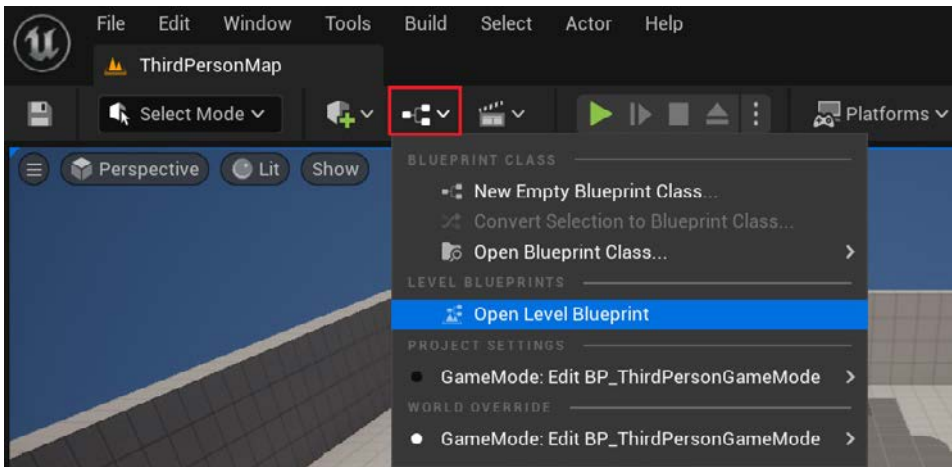


Рис. 1.4. Открытие блюпринта уровня

Как видно на следующем скриншоте, откроется окно Level Blueprint Editor, а в нем — блюпринт текущего уровня. Структура этого окна проще, чем у Blueprint Class Editor (Редактор блюпринтов-классов). Здесь есть три панели: My Blueprint (Мой блюпринт), Details (Свойства), а также Event Graph (Граф событий).

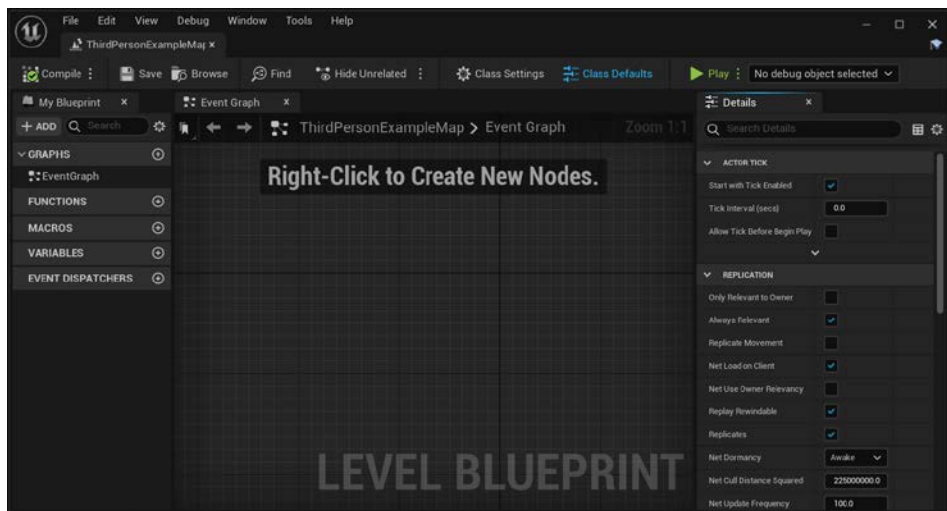


Рис. 1.5. Окно Level Blueprint Editor

Сейчас мы не будем работать в этом окне, а только посмотрим на интерфейс. Закройте его и вернитесь в Level Editor. Сначала поговорим о том, как создаются блюпринты-классы.

Создание блюпринтов-классов

Создать блюпринт-класс можно тремя способами.

1. Нажмите на кнопку **Blueprints** (Блюпринты) на панели инструментов, как и в прошлом пункте, но выберите пункт меню **New Empty Blueprint Class** (Новый пустой блюпринт-класс).
2. Нажмите на кнопку **Content Drawer**, чтобы открыть **Content Browser**, а затем нажмите на кнопку **ADD** (Добавить) и выберите **Blueprint Class** (Блюпринт-класс) в категории **Create Basic Asset** (Создать базовый ассет).
3. Щелкните правой кнопкой мыши на пустом поле панели **Content Browser** и выберите пункт **Blueprint Class** контекстного меню.

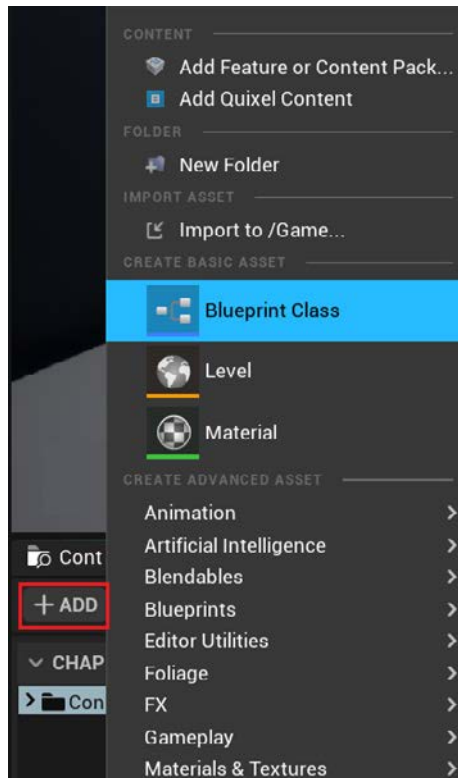


Рис. 1.6. Создание блюпринта-класса

В следующем окне (**Pick Parent Class**) выберите **родительский класс** для нового блюпринта, то есть, как мы на данный момент будем думать, его тип. По умолчанию в окне развернут список классов **COMMON** (Общие), то есть перечень самых распространенных классов, полный же перечень находится в списке

ALL CLASSES (Все классы), который при необходимости можно раскрыть. Как только родительский класс будет выбран, окно со списком закроется, а новый блюпринт появится на панели **Content Browser**, где можно дать ему новое имя. Сейчас пока нажмите на кнопку **Cancel** (Отмена): ведь мы только знакомимся с процессом.

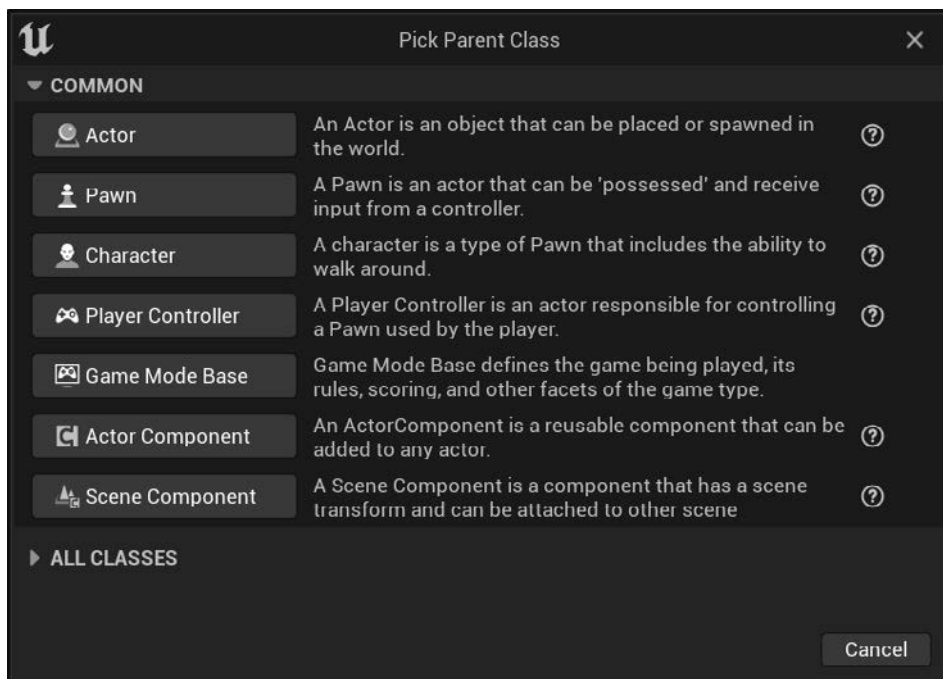


Рис. 1.7. Родительский класс блюпринта

Мы научились открывать **Level Blueprint Editor** и создавать новые блюпринты-классы. Настало время поговорить о том, как устроен **Blueprint Class Editor**, который можно открыть, щелкнув правой кнопкой мыши по блюпринту на панели **Content Browser** и выбрав пункт **Edit** (Правка) контекстного меню либо дважды щелкнув по этому блюпринту.

Интерфейс **Blueprint Class Editor**

Blueprint Class Editor часто называют просто **Blueprint Editor**. Для изменения свойств блюпринта служит несколько панелей. Перечислим самые важные из них.

1. **Toolbar** (Панель инструментов)
2. **Components** (Компоненты)

3. My Blueprint (Мой блюпринт)
4. Details (Свойства)
5. Viewport (Область просмотра)
6. Event Graph (Граф событий)

Все эти панели видны на скриншоте ниже. В качестве примера рассмотрим блюпринт BP_ThirdPersonCharacter (Персонаж для игры от третьего лица) из шаблона Third Person. Откройте папку ThirdPerson/Blueprints и дважды щелкните по этому блюпринту, чтобы открыть окно Blueprint Editor.

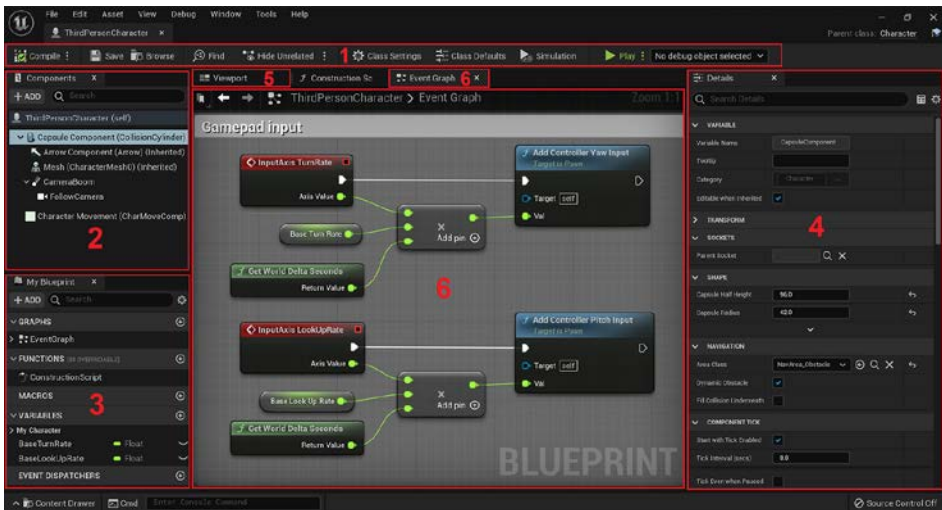


Рис. 1.8. Панели Blueprint Editor

Панель инструментов

Панель инструментов находится в верхней части окна Blueprint Editor и состоит из кнопок для вызова часто используемых функций.

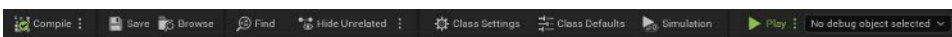


Рис. 1.9. Панель инструментов

Вот эти кнопки:

- **Compile** (Компилировать). Преобразование текущего блюпринта из скрипта в пригодный для исполнения формат низкого уровня. Перед запуском игры все блюпринты должны быть скомпилированы, иначе сделанные изменения не вступают в силу. В случае успешной компиляции на этой кнопке появляется зеленая галочка.

- **Save (Сохранить)**. Сохранение всех изменений, внесенных в текущий блюпринт.
- **Browse (Обзор)**. Показывает текущий блюпринт на панели Content Browser.
- **Find (Поиск)**. Поиск в текущем блюпринте.
- **Hide Unrelated (Скрыть несвязанное)**. Если эта кнопка нажата, все узлы, не связанные с выбранными узлами, удаляются с экрана.
- **Class Settings (Настройки класса)**. Изменение настроек текущего класса при помощи панели Details. Среди таких настроек — **Description (Описание)**, **Category (Категория)** и **Parent Class (Родительский класс)**.
- **Class Defaults (Принятые по умолчанию значения свойств класса)**. Изменение принятых по умолчанию значений свойств блюпринта при помощи панели Details.
- **Simulation (Моделирование)**. Выполнение блюпринта без выхода из Blueprint Editor.
- **Play (Играть)**. Прохождение текущего уровня.
- **Debug Object (Отладка объекта)**. Меню, при помощи которого можно выбрать объект для отладки. Если ничего не выбрано, начнется отладка случайного объекта из текущего блюпринта.

Панель Components

На панели Components перечислены все компоненты текущего блюпринта.

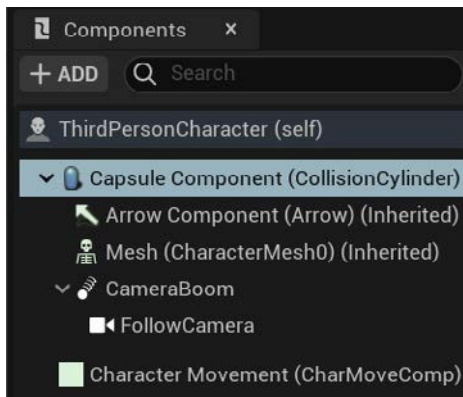


Рис. 1.10. Панель Components

Компоненты — это готовые к применению объекты, которые можно добавить в блюпринт. К ним относятся, например, статические меши, источники освещения, звуки, коллизии, системы частиц и камеры. Можно создать достаточно функциональный блюпринт, просто добавляя в него компоненты.