UNREAL
ENGINE

ЛЕКЦИЯ 1

Введение в Блюпринты

ЦЕЛИ И ВЫВОДЫ ЛЕКЦИИ

Goals

Цели этой лекции

- Представить Блюпринты
- Разобраться в интерфейсе Blueprint Editor
- Показать как создавать Блюпринт Классы
- Представить что такое родительски-дочерняя связь
- Показать как размещать ноды в графе Event Graph
- Объяснить исполнение Блюпринтов в графе

Outcomes

В конце этой лекции вы сможете

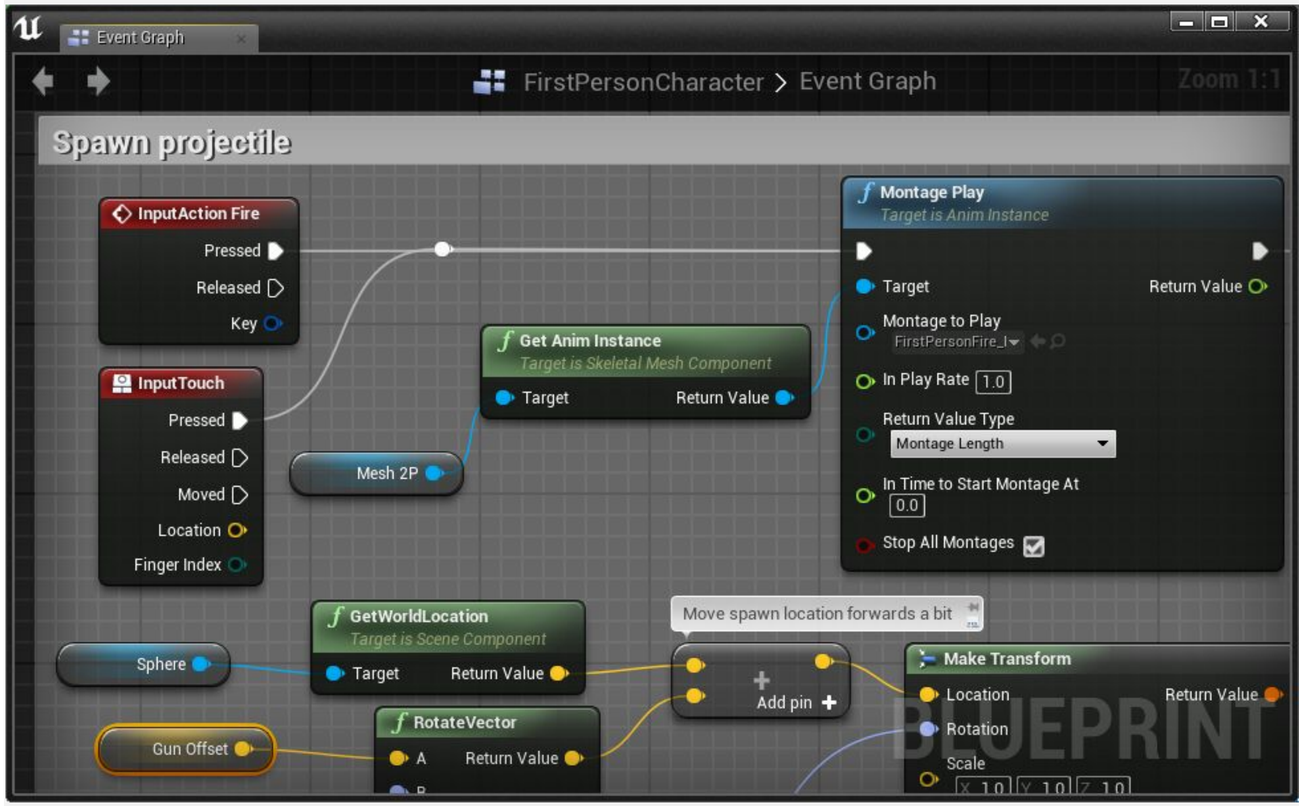
- Создавать Блюпринт классы в Редакторе
- Отличать Level Blueprints (Блюпринты Уровня) и Блюпринт классы
- Размещать ноды в графе Event Graph
- Понимать основную идею родительски-дочерней связи
- Понимать последовательность исполнения Блюпринта

ЧТО ТАКОЕ БЛЮПРИНТ?

Блюпринт это язык визуального программирования созданный Epic Games для Unreal Engine 4. Он используется в пределах редактора Unreal Editor для создания новых классов и функционала геймплея.

Код в Блюпринтах представлен в виде графиков нод соединенных проводами которые отображают порядок исполнения.

Слово “Блюпринт” также используется для определения игрового объекта созданного с использованием Блюпринта.



ОСНОВНЫЕ ТИПЫ БЛЮПРИНТОВ

Level Blueprint

Блюпринт Уровня это особый тип Блюпринта который принадлежит Уровню. Он используется для определения особых событий и действий в Уровне.

Блюпринт Уровня может быть использован для взаимодействия с Акторами Блюпринт Классов и для управления некоторыми системами, такими как кинематика и потоковая передача уровня.

Blueprint Class

Класс - это определение данных и поведения, которые будут использоваться конкретным типом объекта. Блюпринт класс может быть основан на C++ классе или на другом Блюпринт классе.

Блюпринт класс используется для создания интерактивных объектов для игры и может быть использован много раз с любом Уровне.

OBJECT, ACTOR, И ACTOR COMPONENT КЛАССЫ

При создании нового Блюпринт класса вы должны определить **родительский** класс. Все переменные и действия родительского класса будут частью нового класса, который известен как **дочерний класс** или **подкласс**. Это понятие называется **наследованием**.

Ниже представлены некоторые родительские классы:

- **Object**: Основной класс для объектов в Unreal Engine. Все остальные классы это подклассы для Object класса.
- **Actor**: Базовый класс, используемый для объектов, которые могут быть размещены или заспавнены на Уровне.
- **Actor Component**: Базовый класс для компонентов, определяющих многократно используемое поведение, которое может быть добавлено к Акторам.

Следовательно, каждый Актор - это Объект, но не все Объекты - Акторы. Например, компонент-актор - это объект, но и не Актор.

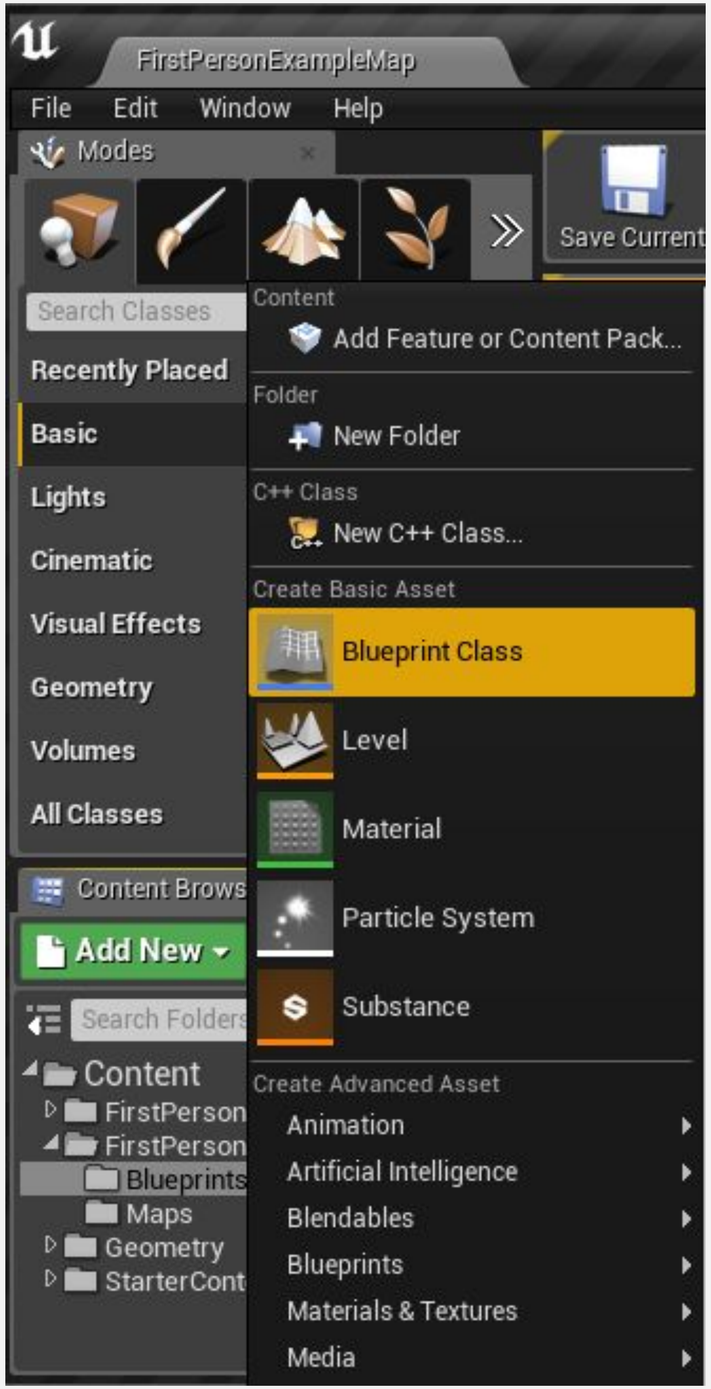
ЭКЗЕМПЛЯРЫ КЛАССА (INSTANCES)

“**Instance**” это термин, используемый для ссылки на объект класса.

Пример можно увидеть на изображении справа. Предполагая, что существует класс под названием **«Blueprint_Chair»**, который представляет стул, изображение показывает, что четыре **экземпляра** класса **Blueprint_Chair** были добавлены на Уровень.

СОЗДАНИЕ БЛЮПРИНТ КЛАССОВ: КОНТЕНТ БРАУЗЕР

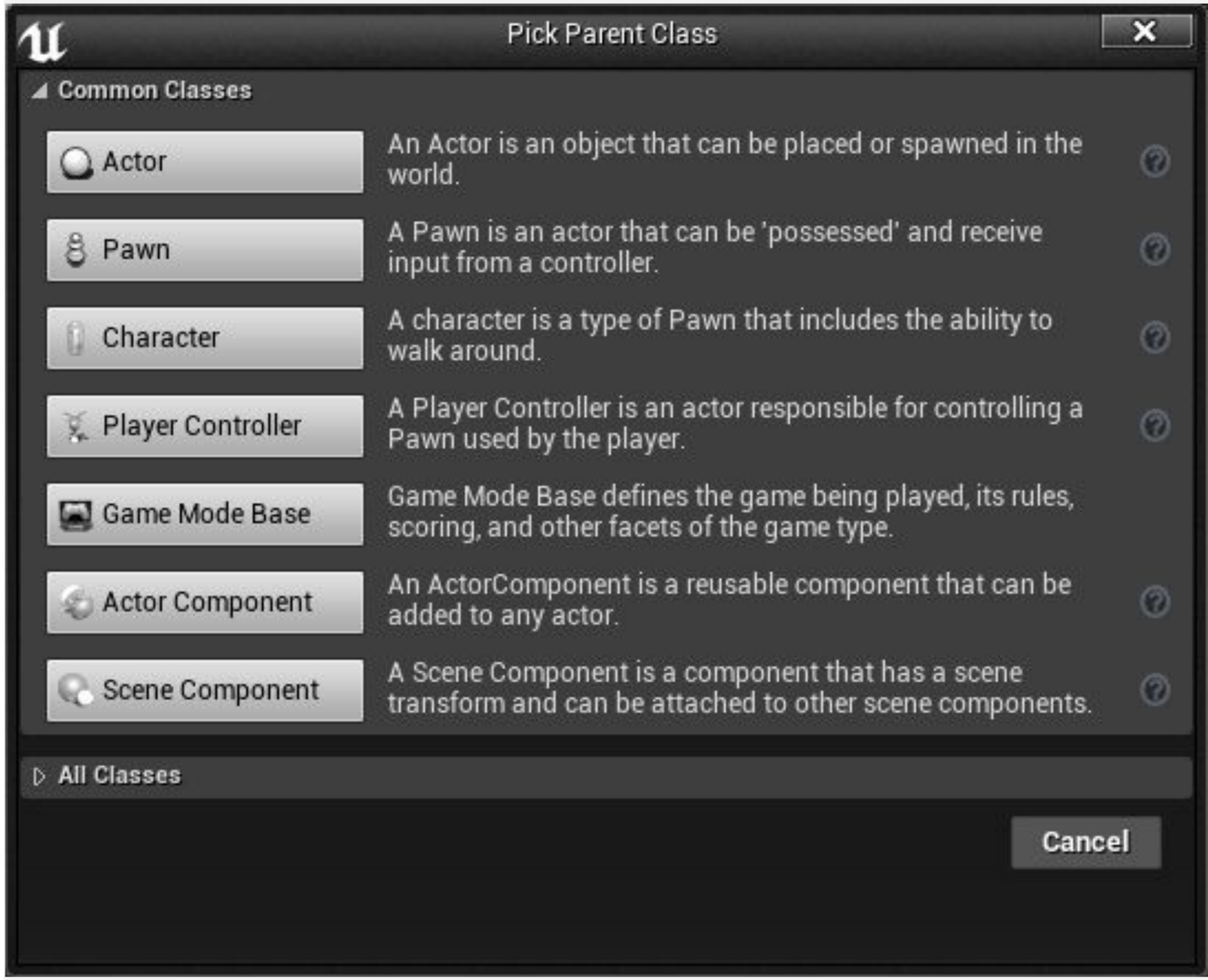
Чтобы создать новый Блюпринт класс в **Content Browser**, нажмите зеленую кнопку **Add New** и выберите “**Blueprint Class**”.



СОЗДАНИЕ БЛЮПРИНТ КЛАССОВ: ОКНО PICK PARENT CLASS

Следующим шагом является выбор родительского класса для нового Blueprint.

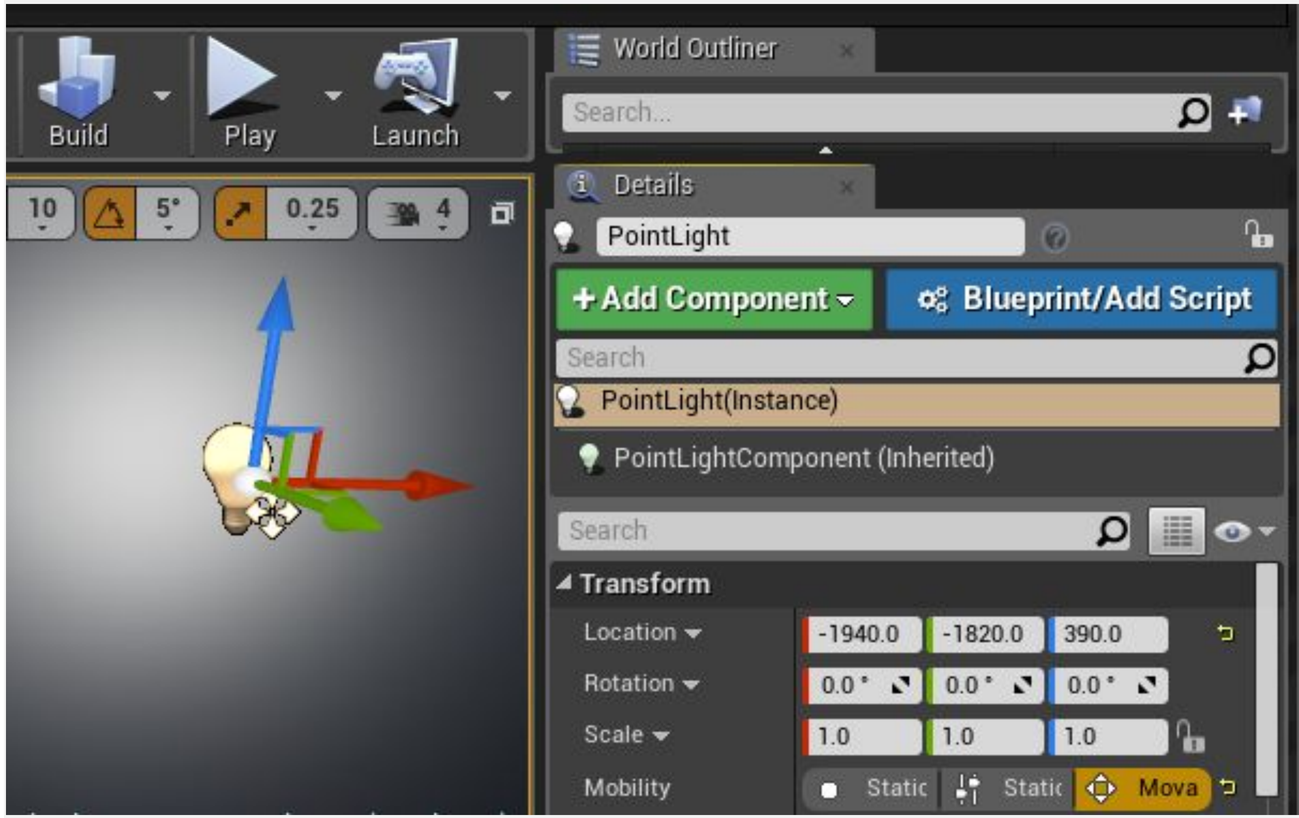
На изображении справа показаны наиболее часто используемые классы, перечисленные в окне **Pick Parent Class.** Другие родительские классы можно найти, развернув параметр **All Classes.**



СОЗДАНИЕ БЛЮПРИНТ КЛАССОВ: АКТОРЫ НА УРОВНЕ

Можно создавать Блюпринт классы на основе Акторов, размещенных на Уровне.

Для этого выберите Актера на уровне и нажмите синюю кнопку **Blueprint/Add Script** на панели **Details** редактора уровней **Level Editor**. Класс Актора будет использоваться как родительский класс нового Блюпринта.



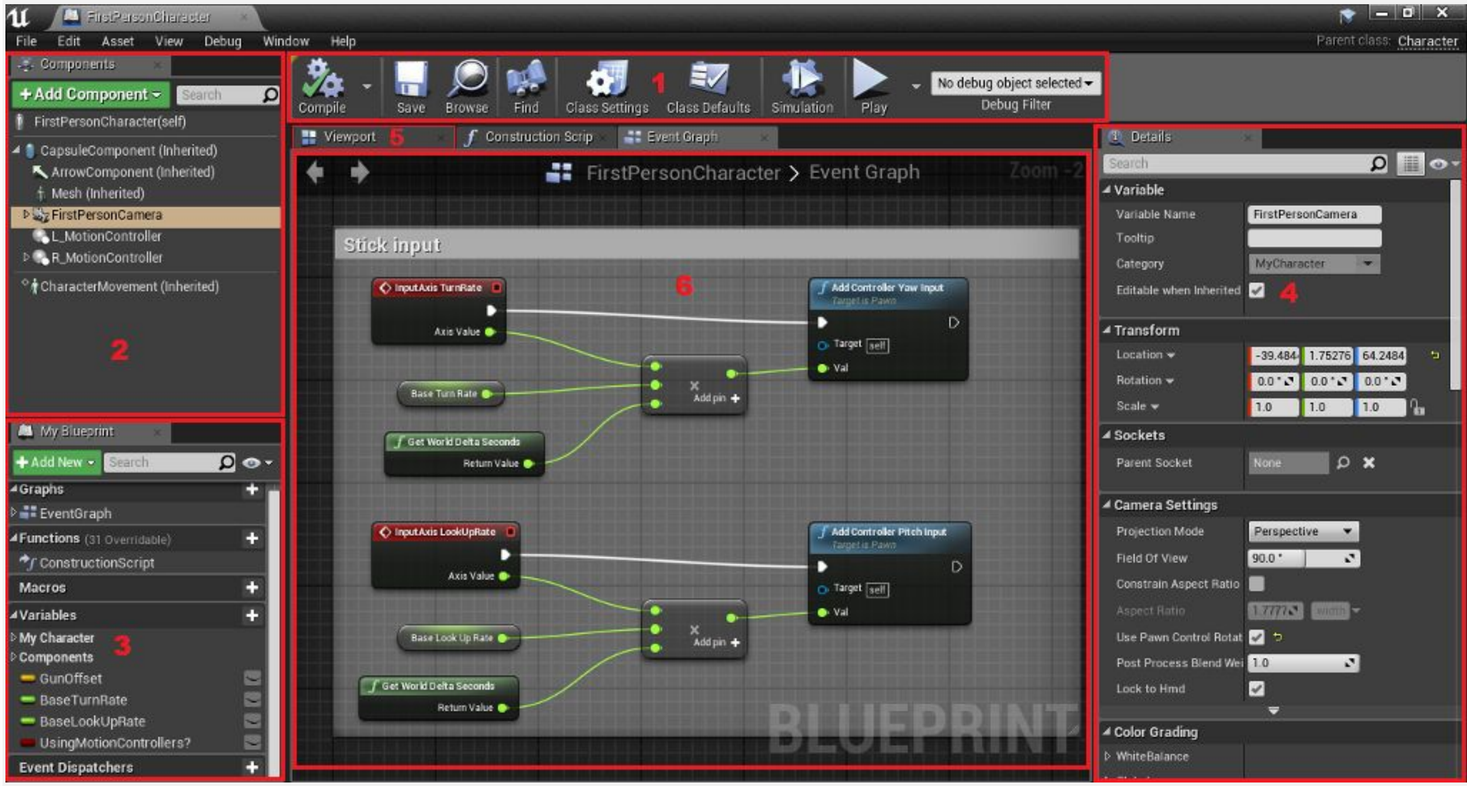
BLUEPRINT EDITOR ИНТЕРФЕЙС

РЕДАКТОР БЛЮПРИНТОВ: ОБЗОР

Чтобы открыть редактор **Blueprint Editor** дважды кликните по Блюпринту или один раз ПКМ и выберите “**Edit...**”.

Основные части редактора Blueprint Editor, выделенные на изображении справа, следующие:

1. Toolbar (панель инструментов)
2. Components panel (компоненты)
3. My Blueprint panel
4. Details panel
5. Viewport
6. Event Graph



РЕДАКТОР БЛЮПРИНТОВ: TOOLBAR

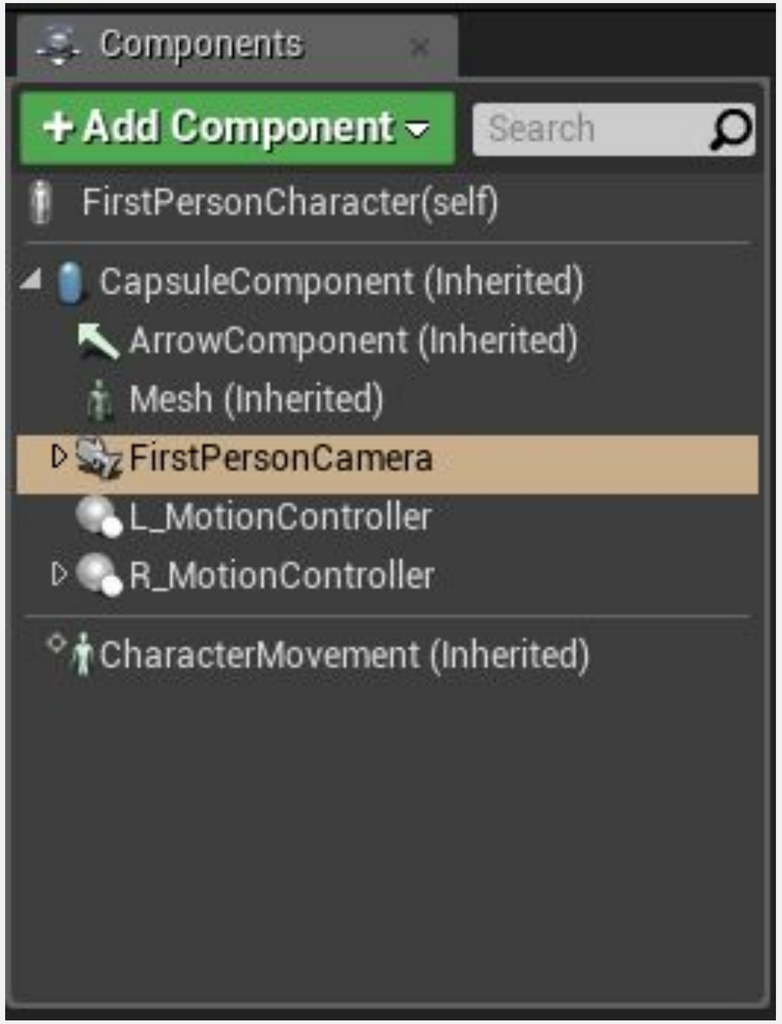
Панель инструментов, расположенная в верхней части редактора, имеет несколько важных кнопок для редактирования Блюпринтов:

- **Compile**: «Компилирует» Блюпринт, важно для проверки кода и применения модификаций.
- **Save**: Сохраняет все изменения, внесенные в текущий Блюпринт, на диск.
- **Browse**: Показывает текущий Блюпринт класс в Content Browser.
- **Find**: Ищет ноды внутри Блюпринта.
- **Class Settings**: Открывает свойства Блюпринта.
- **Class Defaults**: Позволяет изменять начальные значения переменных Блюпринта.

РЕДАКТОР БЛЮПРИНТОВ: COMPONENTS PANEL

На панели **Components** к текущему Блюпринту могут быть добавлены различные типы компонентов.

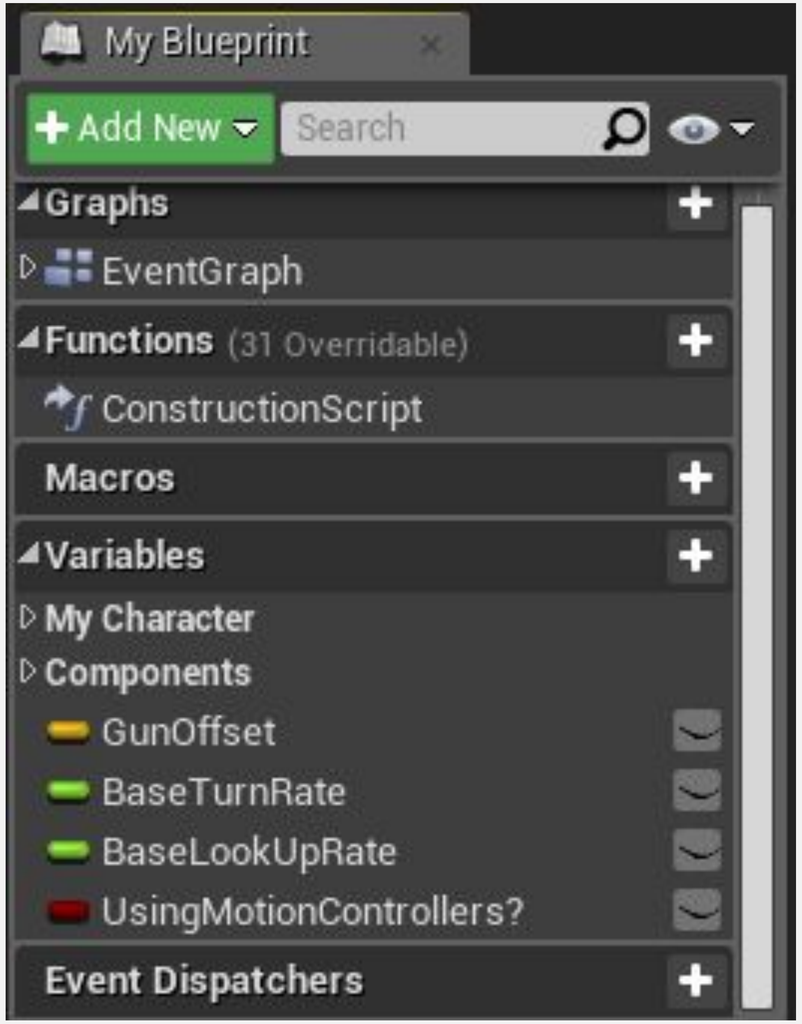
Примерами компонентов являются Статик Меши, источники света, звуки и геометрические объемы, используемые в тестах на столкновения.



РЕДАКТОР БЛЮПРИНТОВ: MY BLUEPRINT PANEL

Панель **My Blueprint** используется для управления переменными, макросами, функциями и графиками Блюпринта.

Она разделена на категории, и каждая категория имеет кнопку “**+**” для добавления новых элементов.

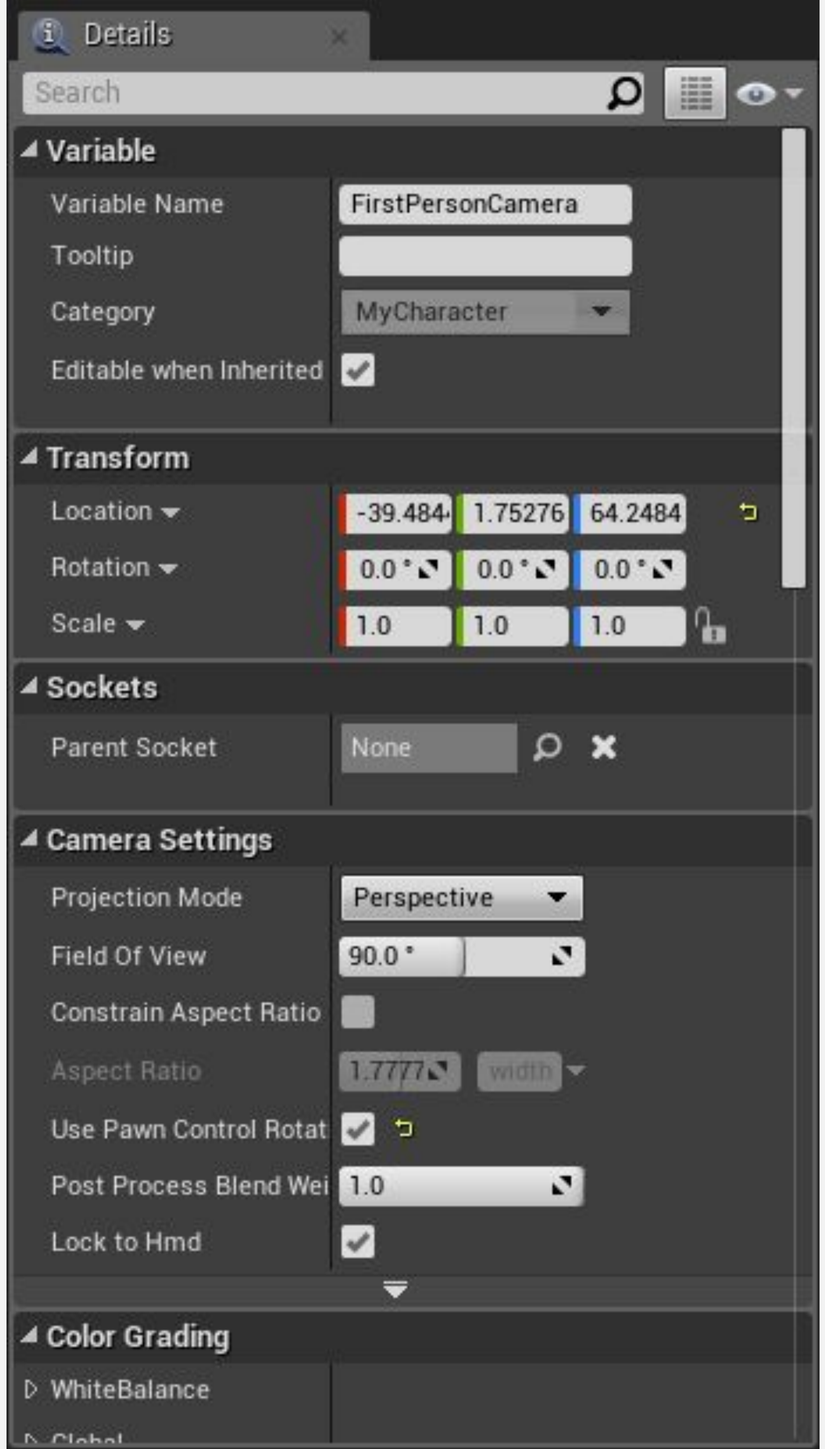


РЕДАКТОР БЛЮПРИНТОВ: DETAILS PANEL

На панели **Details** отображаются свойства текущего выбранного элемента Блюпринт класса, который может быть переменной, функцией или компонентом.

Эти свойства сгруппированы по категориям, и их значения можно изменять.

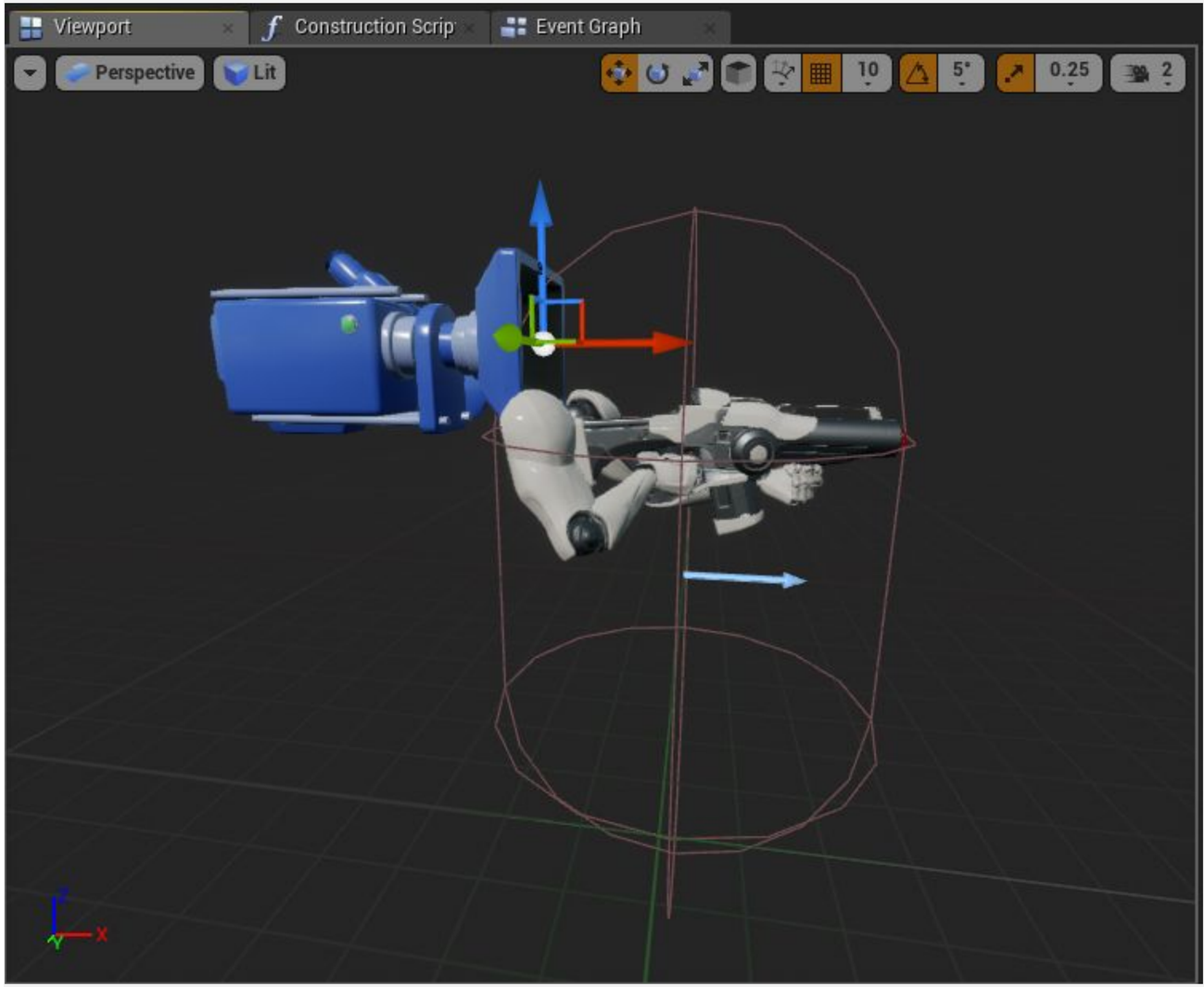
В верхней части панели есть поле поиска, которое можно использовать для фильтрации свойств.



РЕДАКТОР БЛЮПРИНТОВ: VIEWPORT

Viewport содержит визуальное представление компонентов, которые являются частью Блюпринта.

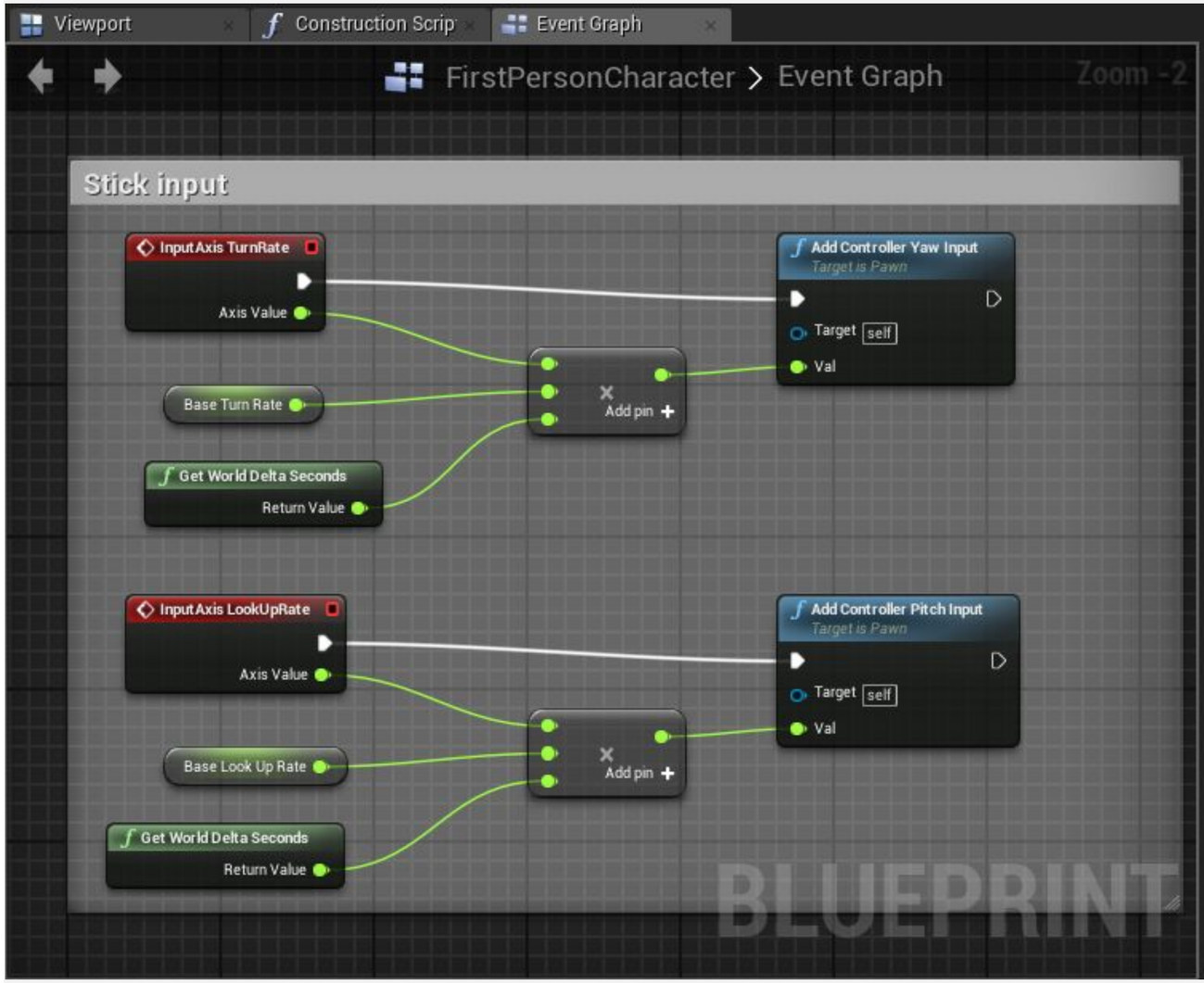
Компонентами можно управлять во Вьюпорте с помощью виджетов преобразования так же, как в редакторе уровней Level Editor.



РЕДАКТОР БЛЮПРИНТОВ: EVENT GRAPH

Event Graph это место, где появляется логика игрового процесса, которая определяет поведение Блюпринт класса во время игры.

Event Graph содержит события и действия, представленные графом из нод.

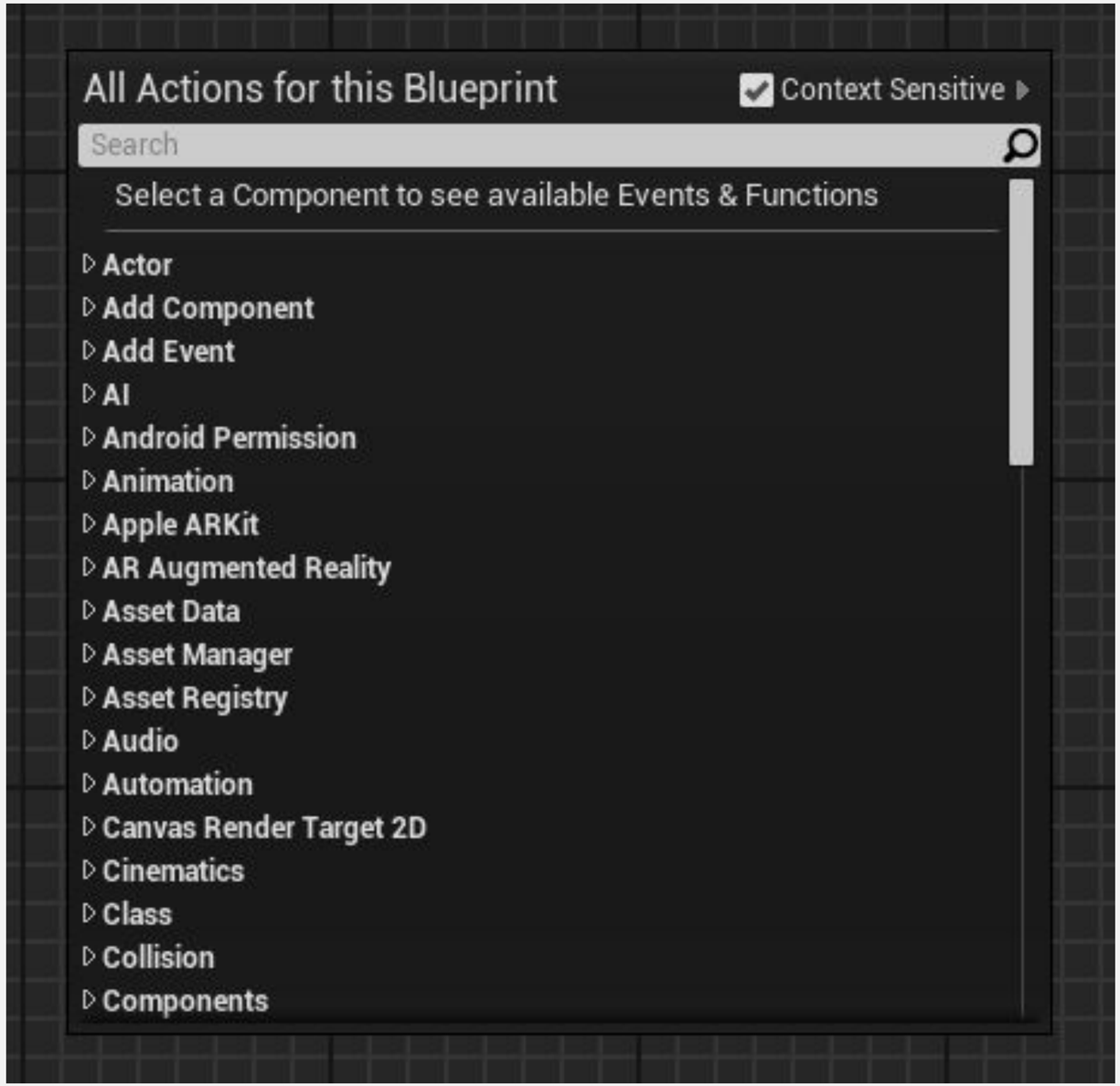


РАЗМЕЩЕНИЕ НОД

РАЗМЕЩЕНИЕ НОД: CONTEXT MENU

Context menu используется для добавления нод в граф. Ноды могут представлять переменные, операторы, функции, и события.

Чтобы открыть **context menu** нажмите ПКМ по пустому месту в **Event Graph** или перетащите провод из пина у ноды и отпустите кнопку мыши.

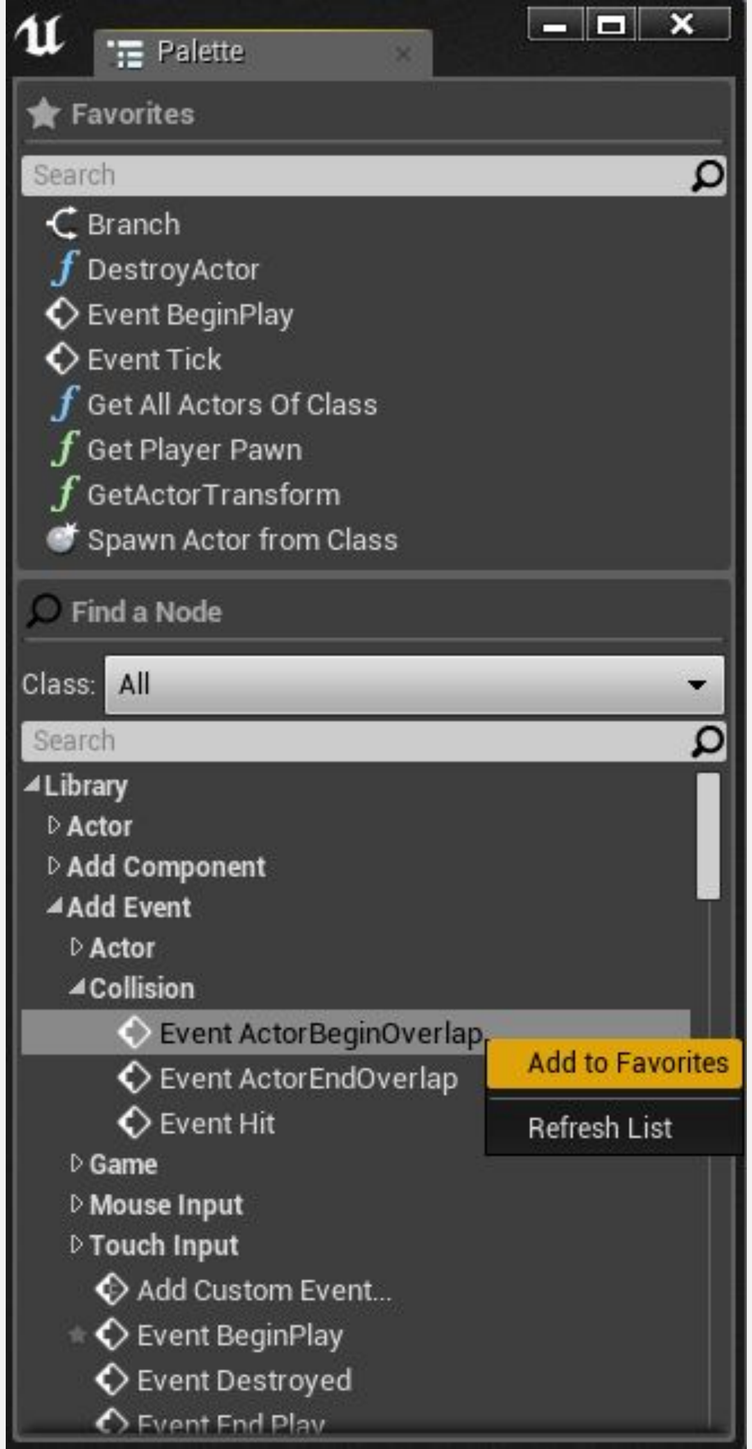


РАЗМЕЩЕНИЕ НОД: PALETTE PANEL

Это панель **Palette** которую можно открыть через **Window > Palette** в редакторе **Blueprint Editor**.

Панель Palette содержит список всех нод, которые можно использовать в Блюпринтах.

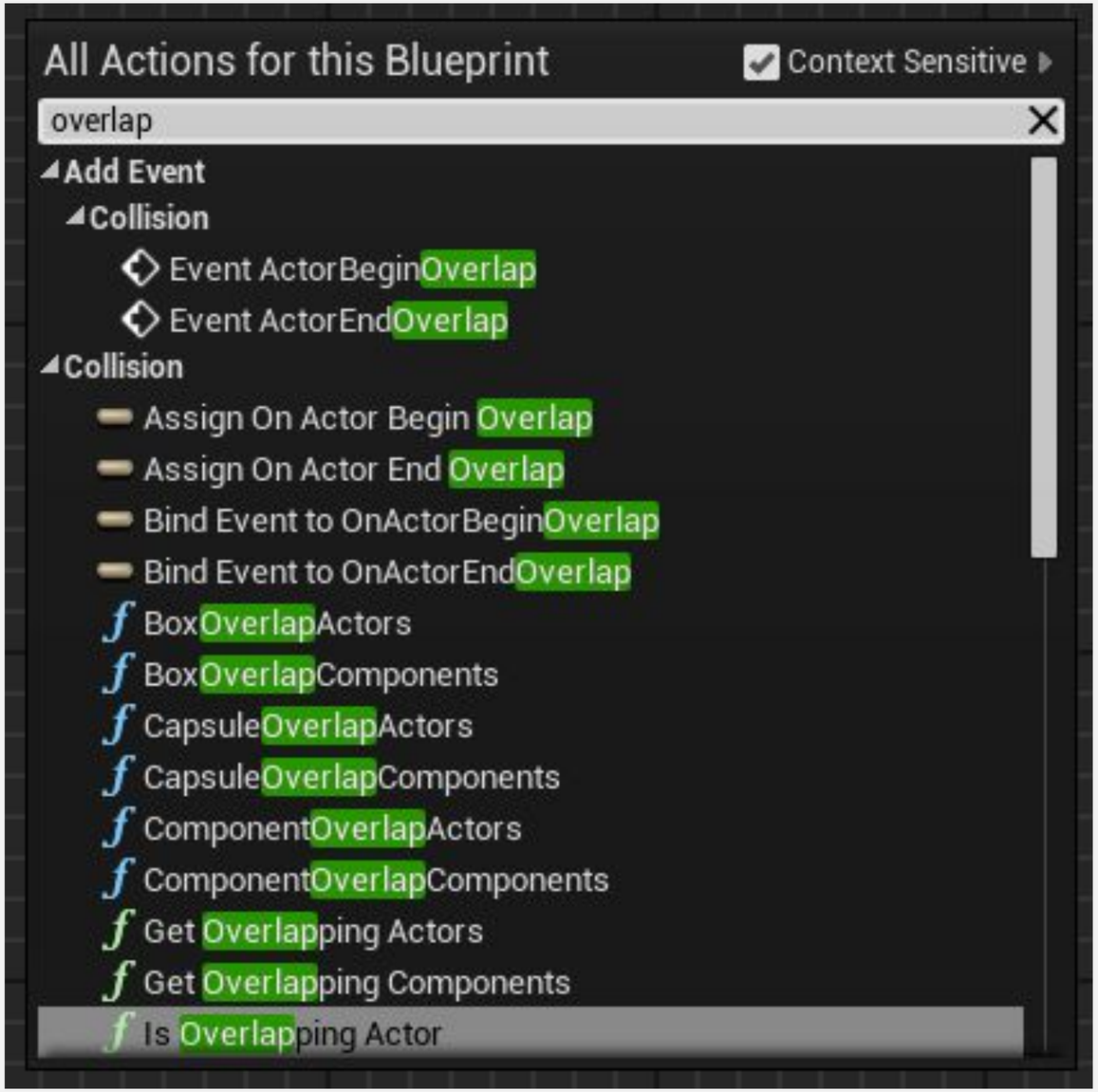
В верхней части панели находится раздел **Favorites** в котором показаны избранные и наиболее часто используемые ноды. Чтобы добавить ноду в избранное, щелкните его ПКМ на панели **Palette** и выберите “**Add to Favorites**”.



РАЗМЕЩЕНИЕ НОД: SEARCH BARS

В **контекстном меню (context menu)** есть **панель поиска (search bar)**, которая фильтрует список нод, когда пользователь вводит текст.

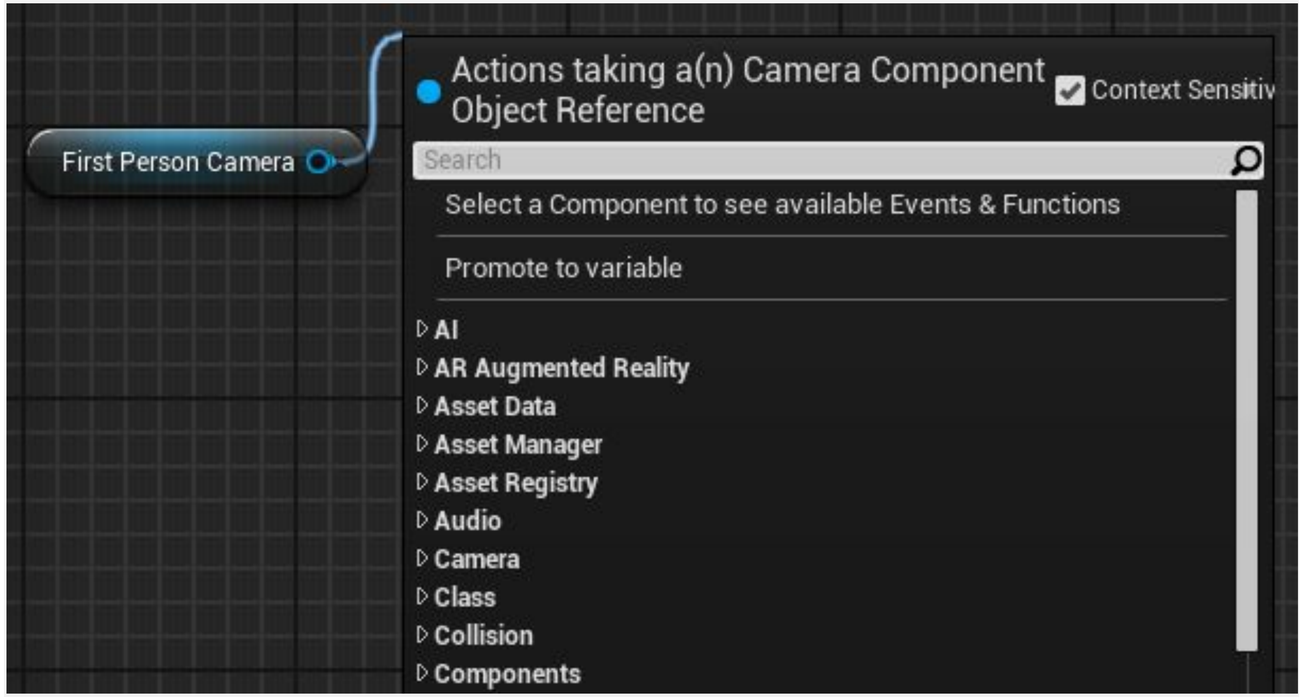
Панель **Palette** имеет **две панели поиска**, одна для раздела **Favorites** а другая для списка **Palette**.



РАЗМЕЩЕНИЕ НОД: CONTEXT SENSITIVE

В верхней части context menu есть галочка с названием “**Context Sensitive**”. Если эта галочка установлена, список нод будет отфильтрован до действий, которые можно использовать в текущем контексте.

Если context menu было открыто при щелчке ПКМ на **Event Graph**, контекстом будет текущий Блюпринт класс. Если он был открыт путем перетаскивания провода от вывода узла, контекстом будет тип вывода.



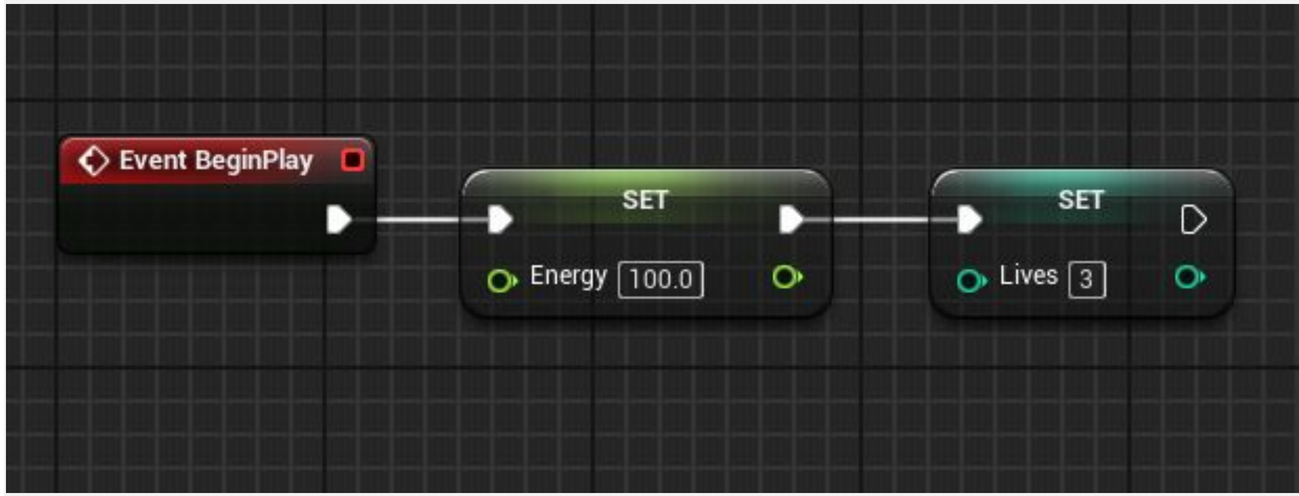
НОДЫ, ПИНЫ, И ПРОВОДА

ВЫПОЛНЕНИЕ BLUEPRINT GRAPH

Выполнение нод Блюпринта начинается с красной ноды **Event** и следует по **белому проводу** слева направо, пока не достигнет последнего узла. После этого он переходит к другому инициированному событию Блюпринта.

Белые пины нод называются **исполнительными пинами**. Пины другого цвета - это **пины данных**.

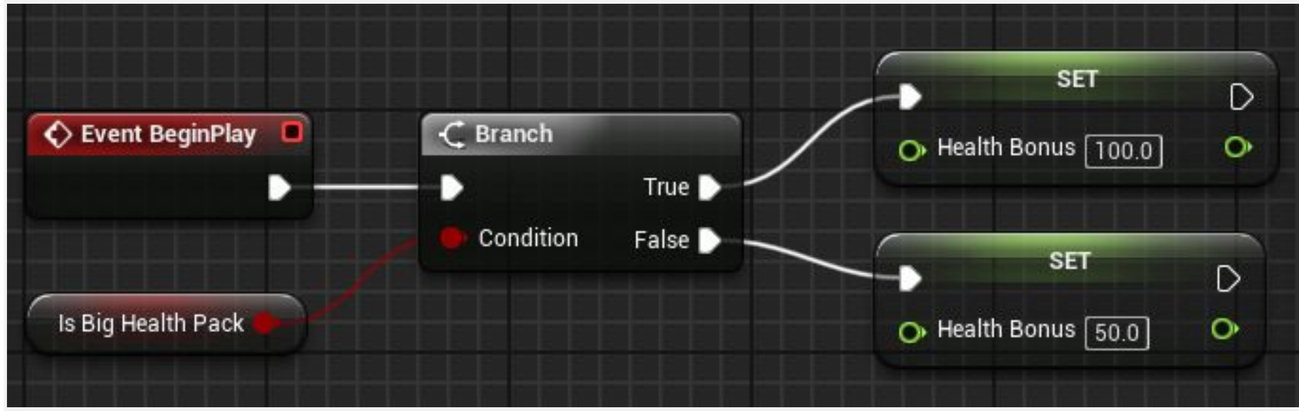
На изображении справа значения присваиваются переменным **Energy** и **Lives** при запуске события **BeginPlay**.



ПУТИ ВЫПОЛНЕНИЯ

Есть несколько нод, которые контролируют поток выполнения Блюпринта. Эти ноды определяют **путь выполнения** на основе условий.

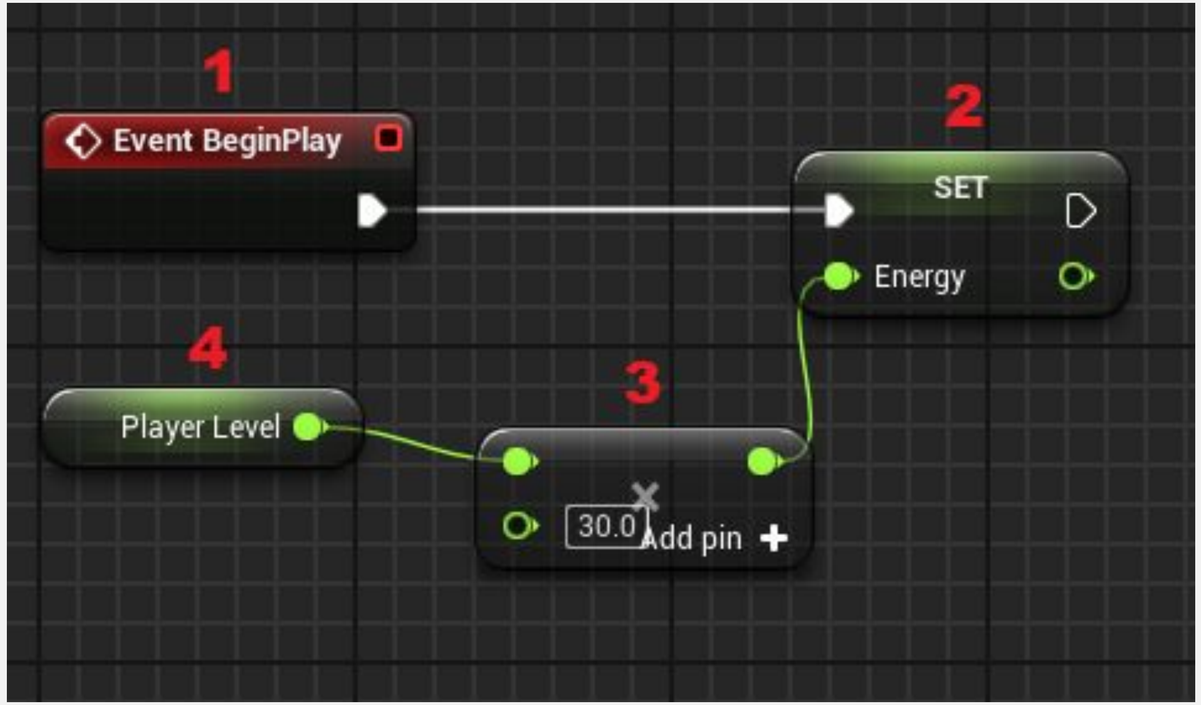
Например, изображение справа показывает ноду **Branch**, который использует в качестве условия значение логической переменной **Is Big Health Pack**. Если значение “**true**”, то выполнение будет продолжено на выводе **True**. Если значение равно «**false**», выполнение будет продолжено на выводе **False**.



ПРОВОДА ПЕРЕДАЧИ ДАННЫХ

Когда нода с выводами данных работает, она извлекает необходимые данные с помощью **проводов данных** перед завершением своего выполнения.

На изображении справа выполнение начинается с события **BeginPlay**. Нода **Set** присваивает новое значение переменной **Energy**, но это значение должно быть получено с помощью провода данных, подключенного к ноде **умножения**, которой нужно будет получить значение переменной уровня игрока с помощью другого провода данных.



ИТОГ

В этой лекции были представлены основные концепции, используемые в Blueprint. Он объяснил разницу между классами Level Blueprints и Blueprint и концепцию наследования родитель-потомок.

Он показал различные части редактора Blueprint и объяснил поток выполнения на графике Blueprint.