



UNREAL
ENGINE

ЛЕКЦИЯ 2

Основы программирования на Блюпринтах

ЦЕЛИ И ИТОГИ ЛЕКЦИИ

Goals

Цели этой лекции

- Продемонстрировать как создавать переменные и менять их свойства
- Показать, как использовать арифметические, родственные и логические операторы
- Показать функции, пользовательские события, макросы и их отличия
- Показать поток программы

Outcomes

К концу этой лекции вы сможете

- Создавать переменные, используя правильный тип для каждой ситуации
- Создавать выражения с помощью операторов
- Понимать, когда использовать функции, настраиваемые события или макросы
- Использовать некоторые базовые ноды, которые управляют ходом программы



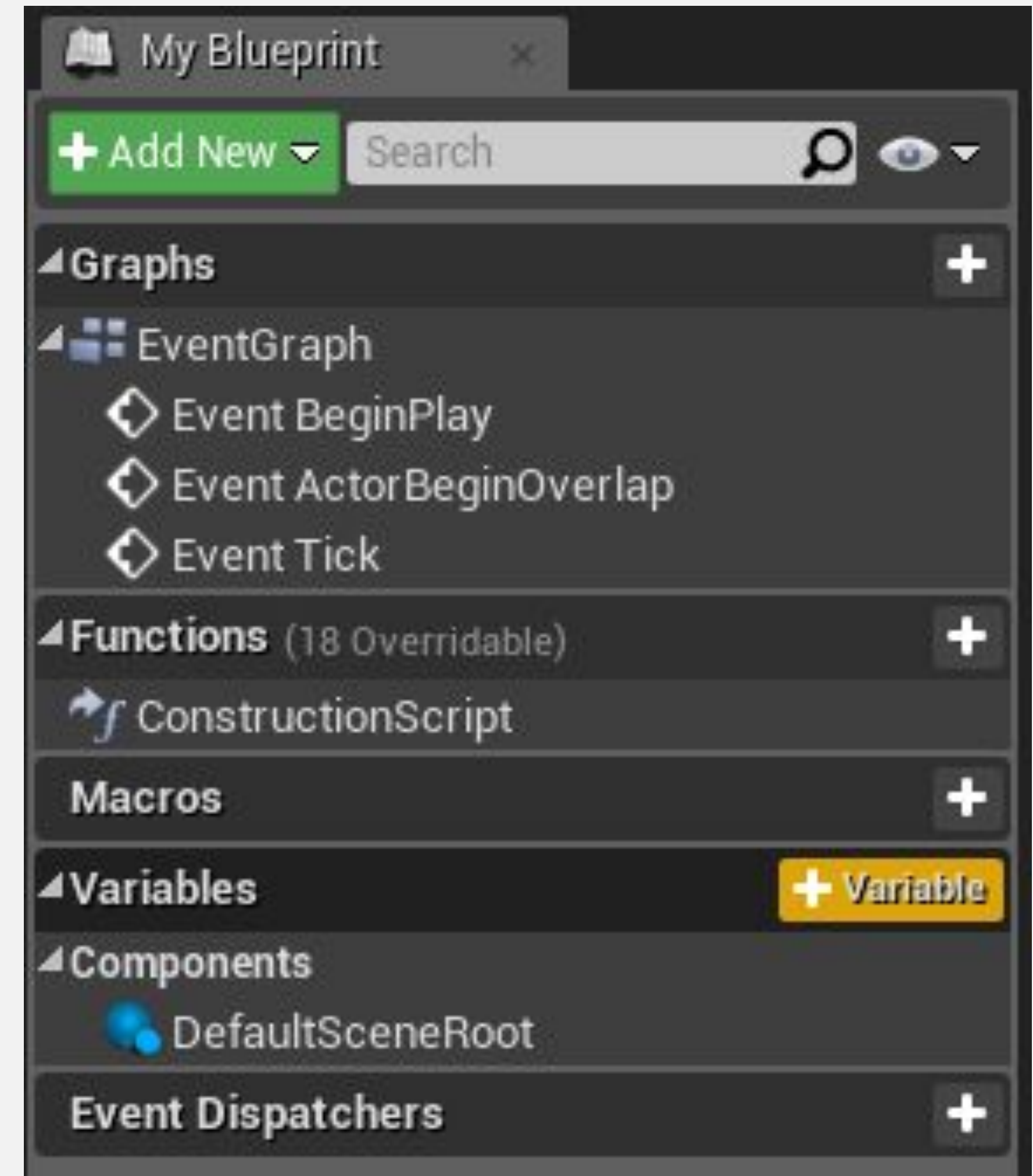
ПЕРЕМЕННЫЕ



СОЗДАНИЕ ПЕРЕМЕННЫХ

Переменные используются для хранения значений и свойств в Блюпринтах, которые можно изменять во время выполнения игры. Переменные могут быть разных типов.

Чтобы создать переменную, переключитесь на панель **My Blueprint** в редакторе **Blueprint Editor** и нажмите кнопку “+” в категории **Variables**.

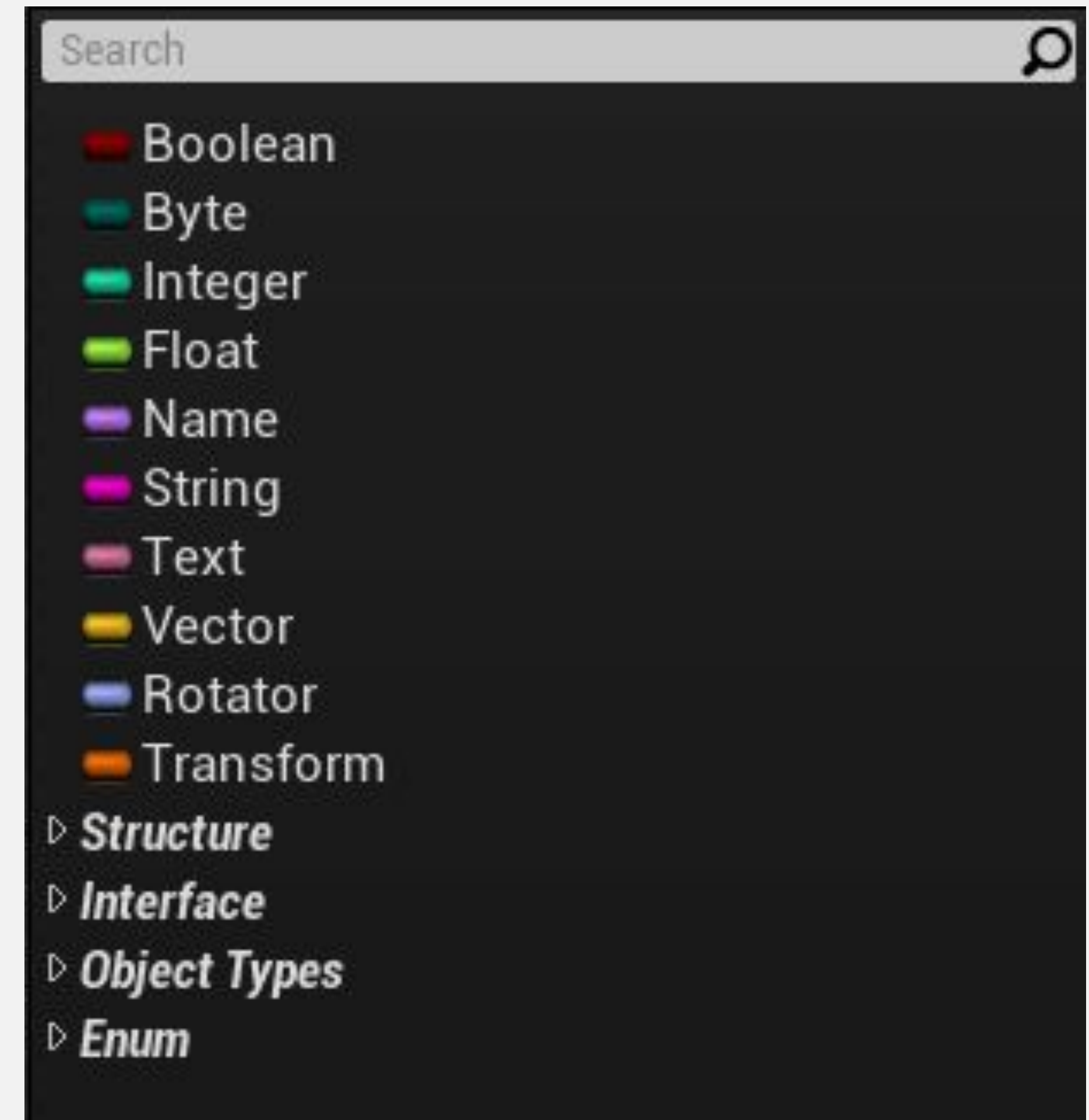




ТИПЫ ДАННЫХ ПЕРЕМЕННЫХ

Ниже представлены основные типы переменных:

- **Boolean:** Может содержать только значение «ИСТИНА» или «ЛОЖЬ».
- **Integer:** Используется для хранения целочисленных значений.
- **Float:** Используется для хранения десятичных значений.
- **String / Text:** Используется для хранения текста. Переменная Text предпочтительнее, поскольку она поддерживает локализацию.
- **Vector:** Содержит значения с плавающей запятой X, Y и Z.
- **Transform:** Используется для хранения местоположения, поворота и масштаба.



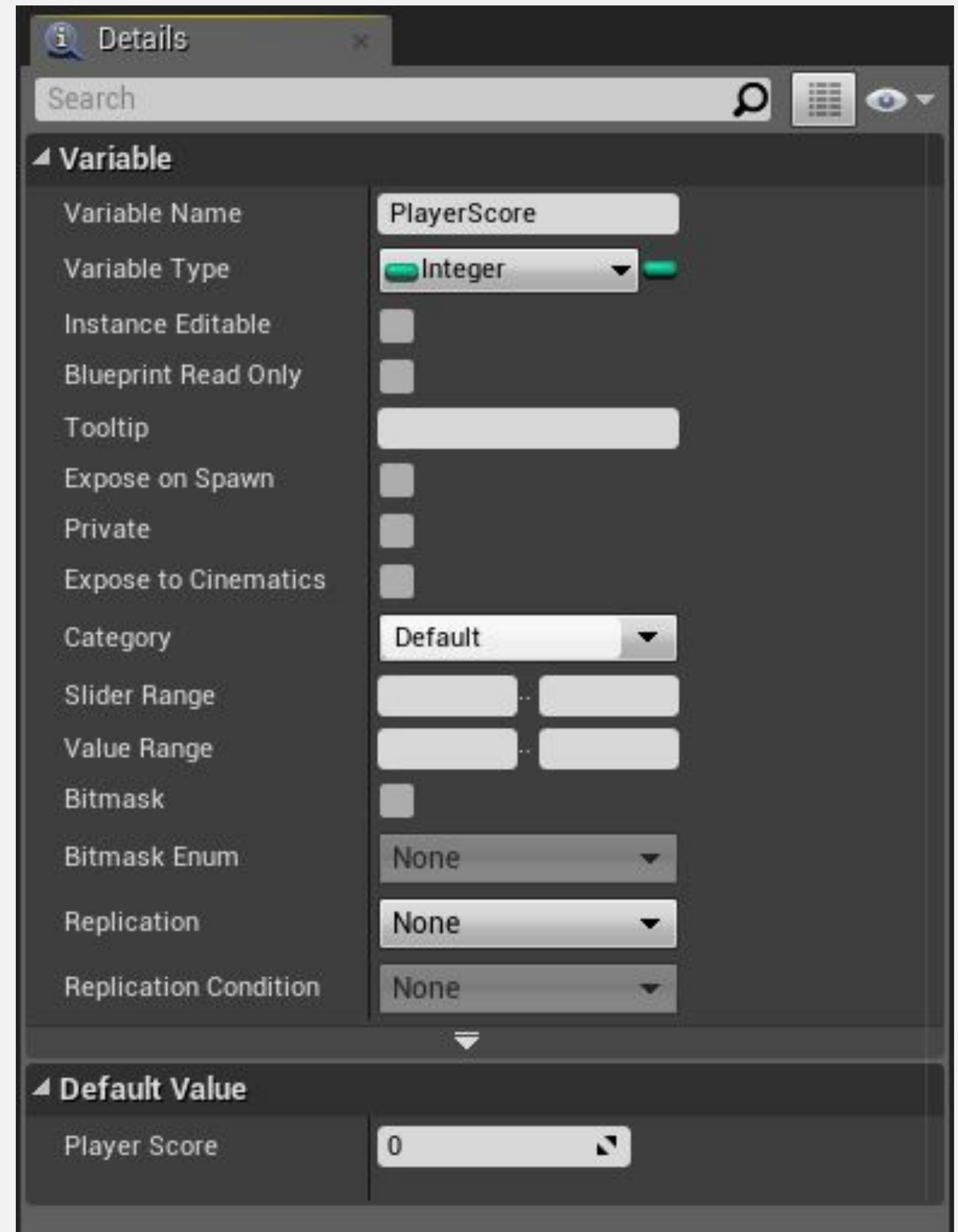


ПАНЕЛЬ DETAILS

Когда переменная выбрана, ее свойства отображаются на панели **Details**. Здесь можно изменить имя и тип переменной.

Другие свойства на панели **Details** включают следующее:

- **Instance Editable:** Если эта галочка поставлена, переменная может быть изменена в экземплярах, которые находятся на Уровне.
- **Blueprint Read Only:** Если отмечено, переменная не может быть изменена нодами Блюпринта.
- **Tooltip:** Содержит информацию, отображаемую при наведении курсора на переменную..
- **Expose on Spawn:** Если эта галочка поставлена, переменная может быть установлена при спавне Блюпринта.



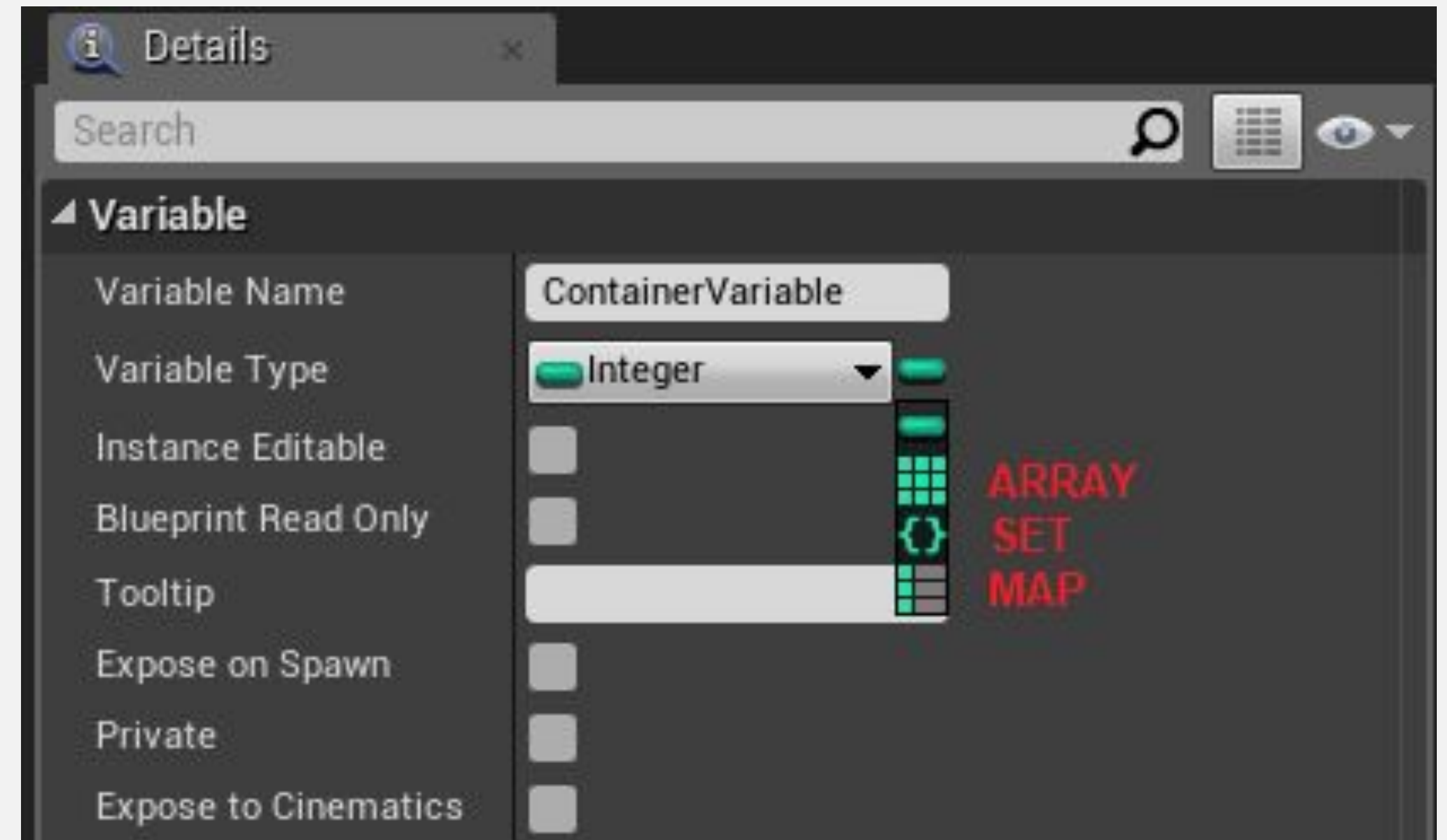


МАССИВ, НАБОР И КАРТА

Свойство **Variable Type** включает кнопку, которая используется для преобразования переменной в **контейнер**.

В контейнере может храниться несколько элементов одного типа. Ниже перечислены типы доступных контейнеров.

- **Array:** Упорядоченный список значений, доступ к которым осуществляется с помощью значения индекса.
- **Set:** Неупорядоченный набор значений. Повторяющиеся значения не допускаются.
- **Map:** Список, в котором для определения каждой записи используется пара "ключ-значение". Повторяющиеся пары "ключ-значение" не допускаются.





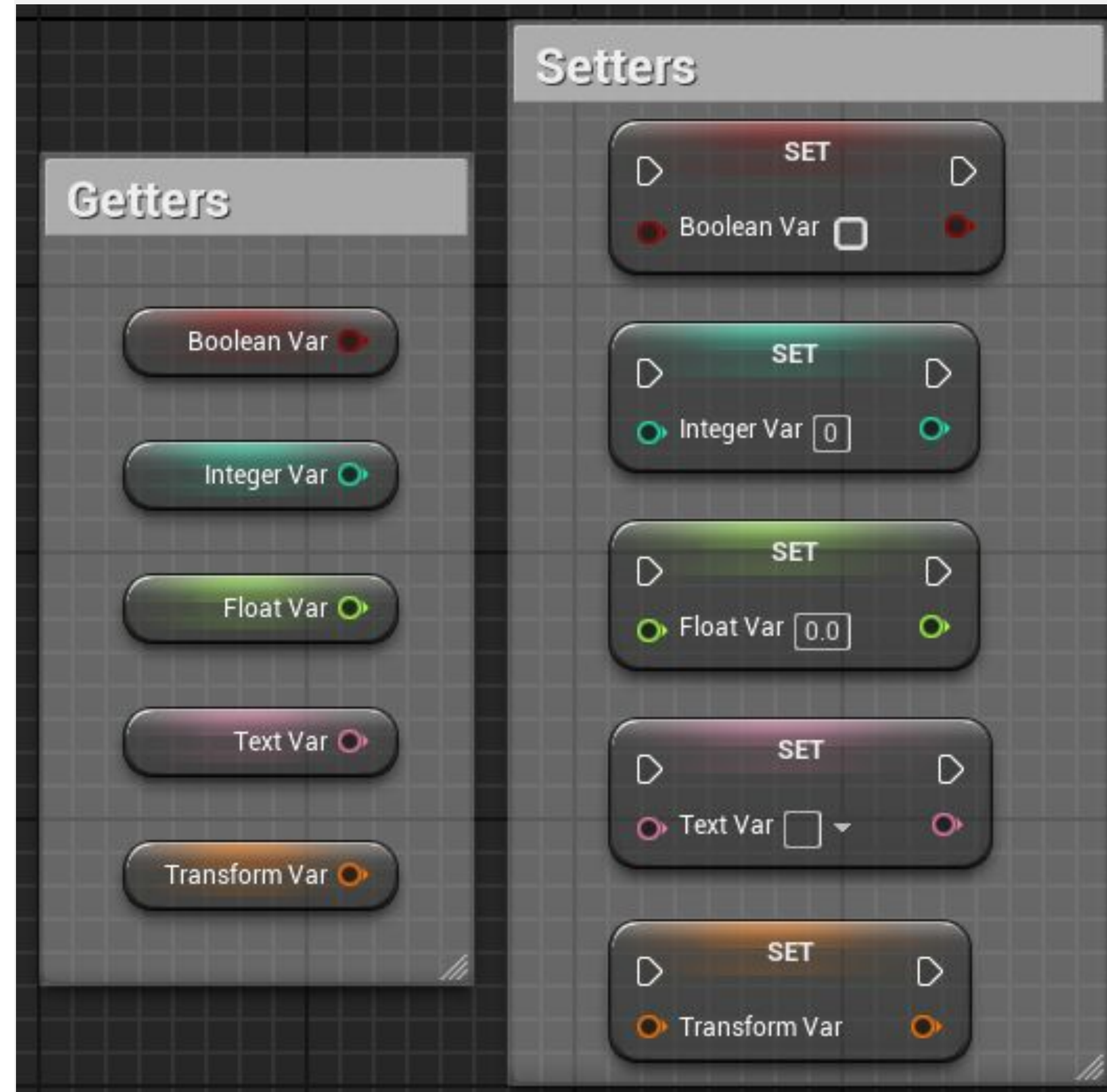
ГЕТТЕРЫ И СЕТТЕРЫ

Когда переменная перетаскивается в Event Graph, появляется контекстное меню с опциями «Получить» (**Get**) и «Установить» (**Set**).

Get ноды используются для чтения значения переменной.

Set ноды используются для хранения нового значения в переменной.

Есть полезные горячие клавиши для создания нод **Get** и **Set**. Чтобы создать ноду **Get**, нажмите клавишу **Ctrl** при перетаскивании переменной. Нода **Set** создается с помощью клавиши **Alt**.



ОПЕРАТОРЫ



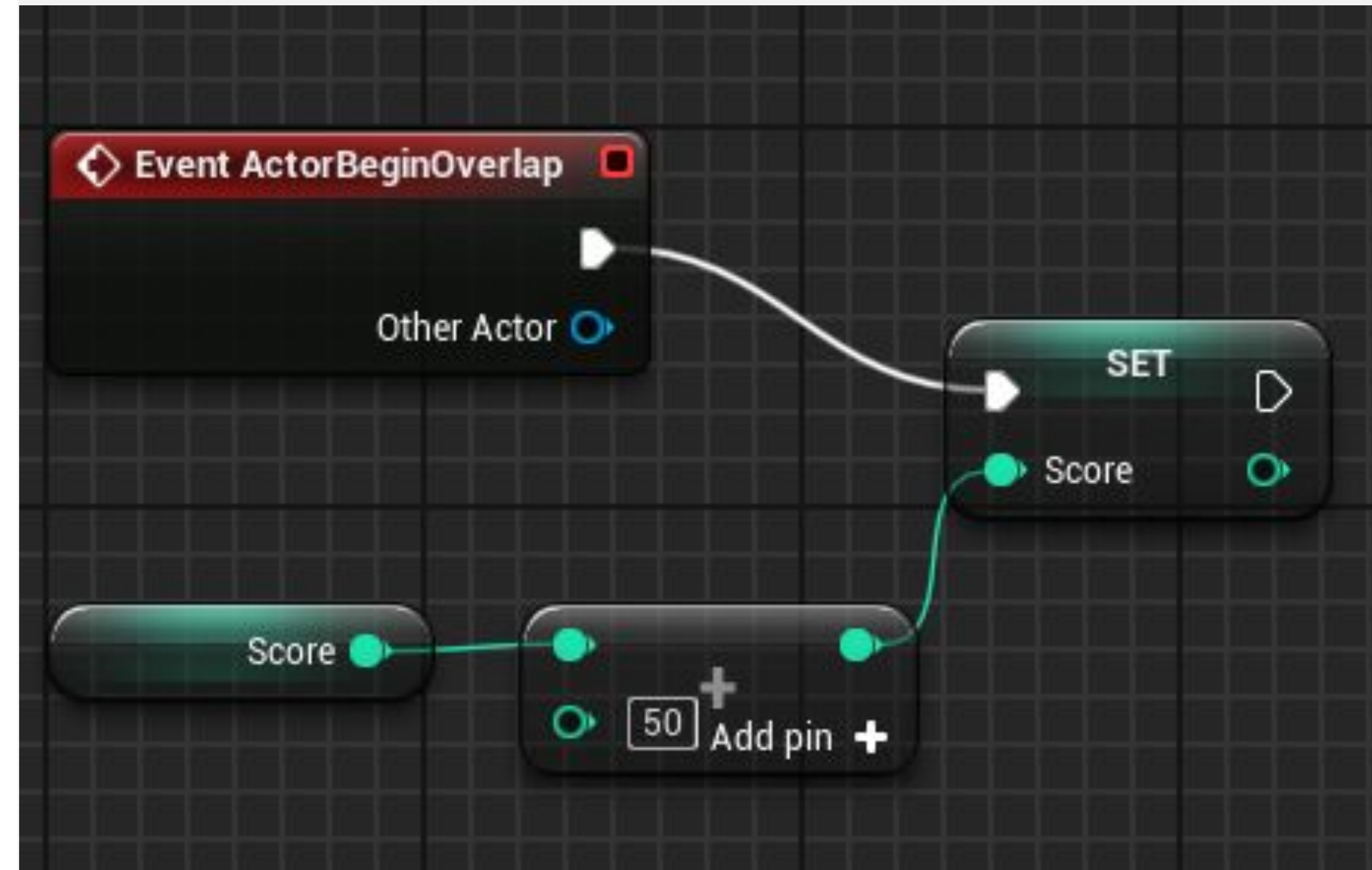
АРИФМЕТИЧЕСКИЕ ОПЕРАТОРЫ

Арифметические операторы (+, -, *, /) могут использоваться для создания математических выражений в Блюпринтах.

На изображении справа показано простое выражение, которое добавляет значение «50» к текущей переменной **Score**, а затем устанавливает новое вычисленное значение в переменной **Score**.

Оператор «+» получает два входных значения слева и дает результат операции справа. Чтобы использовать более двух входных значений, просто нажмите на опцию **Add pin**.

Входные значения могут быть введены непосредственно в ноды или могут быть получены из переменных.

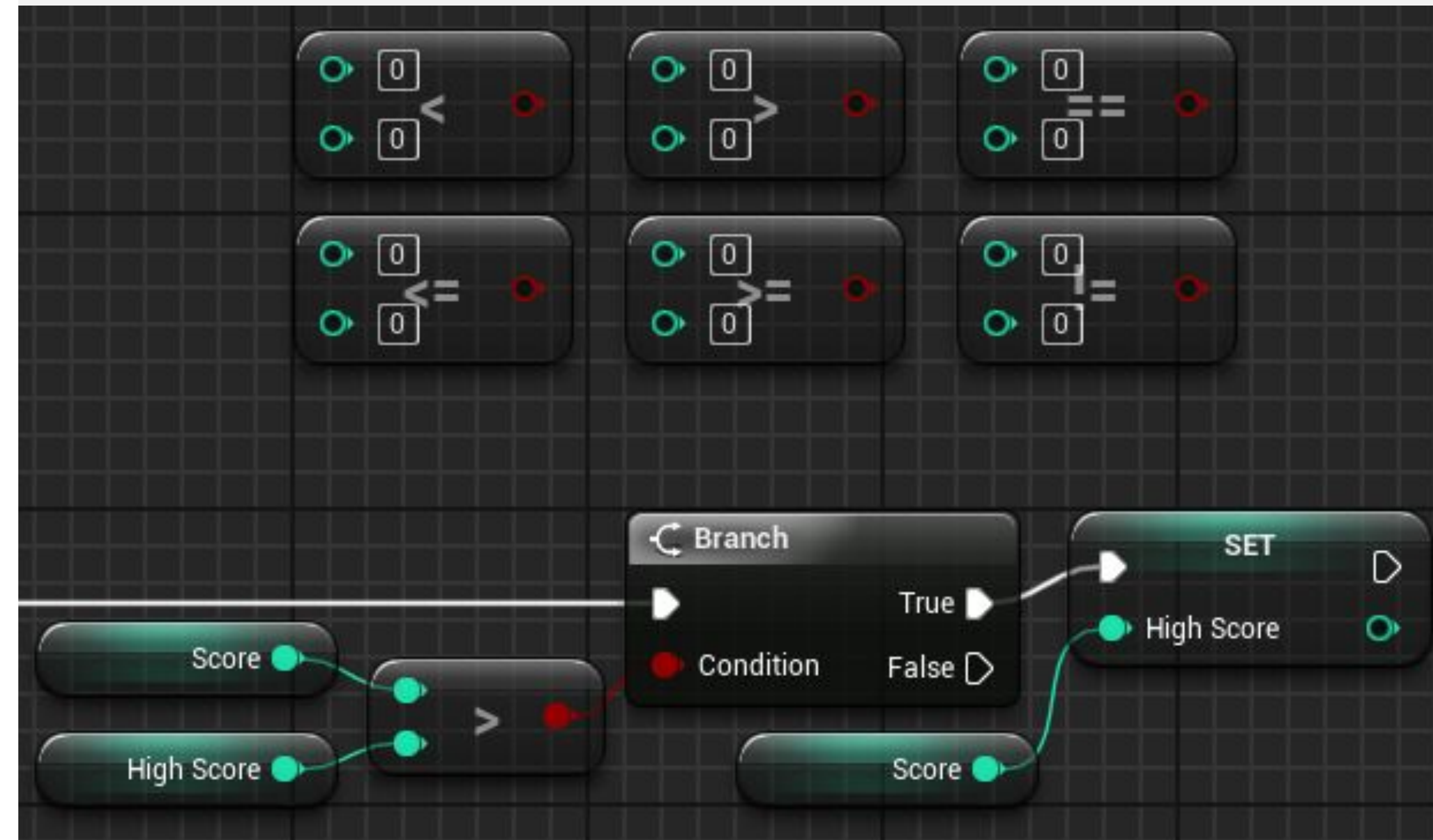




ОПЕРАТОРЫ СООТНОШЕНИЯ

Операторы соотношения выполнить сравнение двух значений и вернуть логическое значение (“true” or “false”) в результате сравнения.

На изображении справа показаны операторы соотношения и пример использования ноды **Branch**. В конце игры текущий счет игрока (переменная **Score**) сравнивается с наивысшим зарегистрированным счетом игры (переменная **High Score**). Если результат игрока выше, значение переменной **Score** будет сохранено в переменной **High Score**.



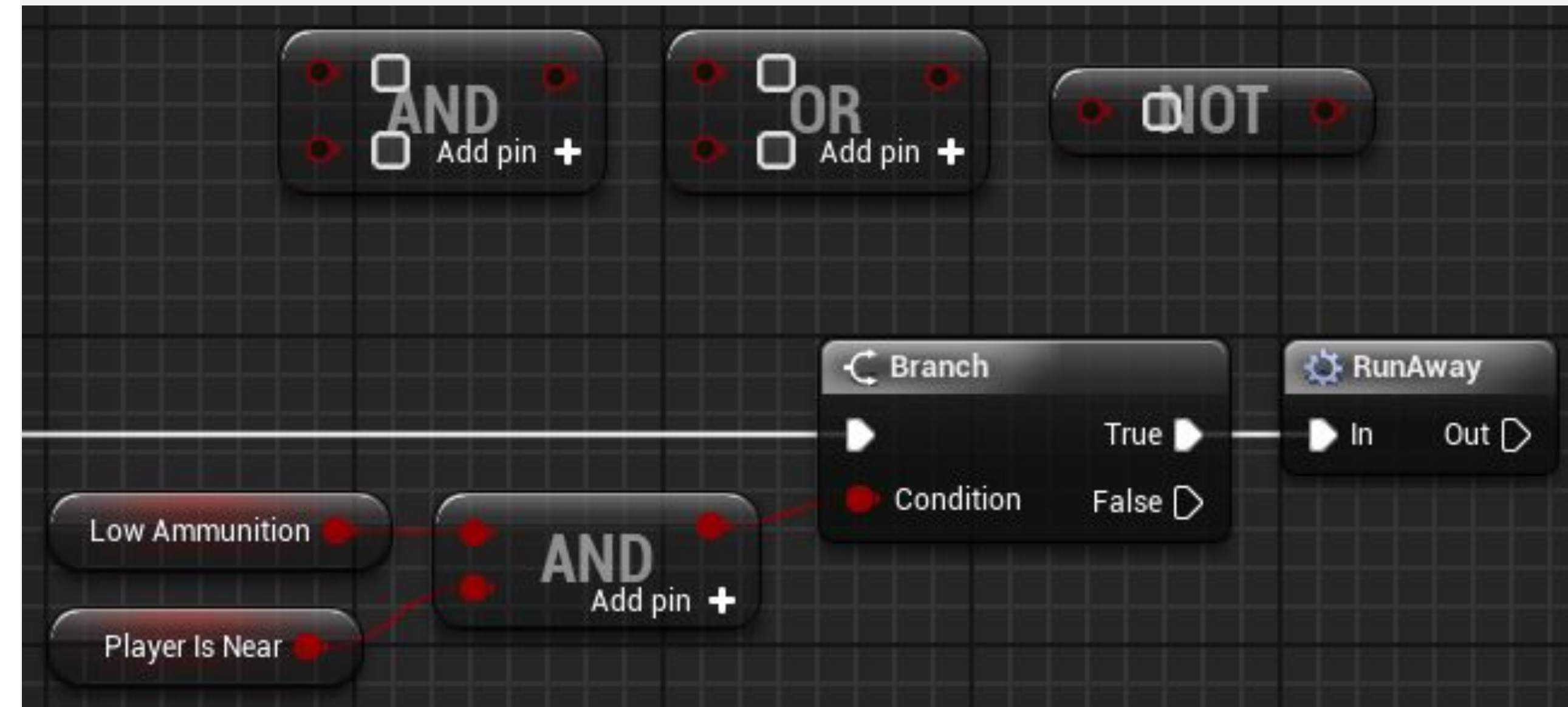


ЛОГИЧЕСКИЕ ОПЕРАТОРЫ

Логические операторы выполняют операцию между логическими значениями и возвращают логическое значение («true» or «false») в результате операции. Основные логические операторы следующие:

- **OR**: Возвращает значение «true», если любое из входных значений «true».
- **AND**: Возвращает значение «true», только если все входные значения равны «true».
- **NOT**: Получает только одно входное значение, и результатом будет обратное значение.

Пример справа имитирует простое решение противника в игре. Если у врага мало боеприпасов (переменная **Low Ammunition**) и игрок находится поблизости (переменная **Player Is Near**), то враг решает убежать.



ФУНКЦИИ, СОБЫТИЯ И МАКРОСЫ

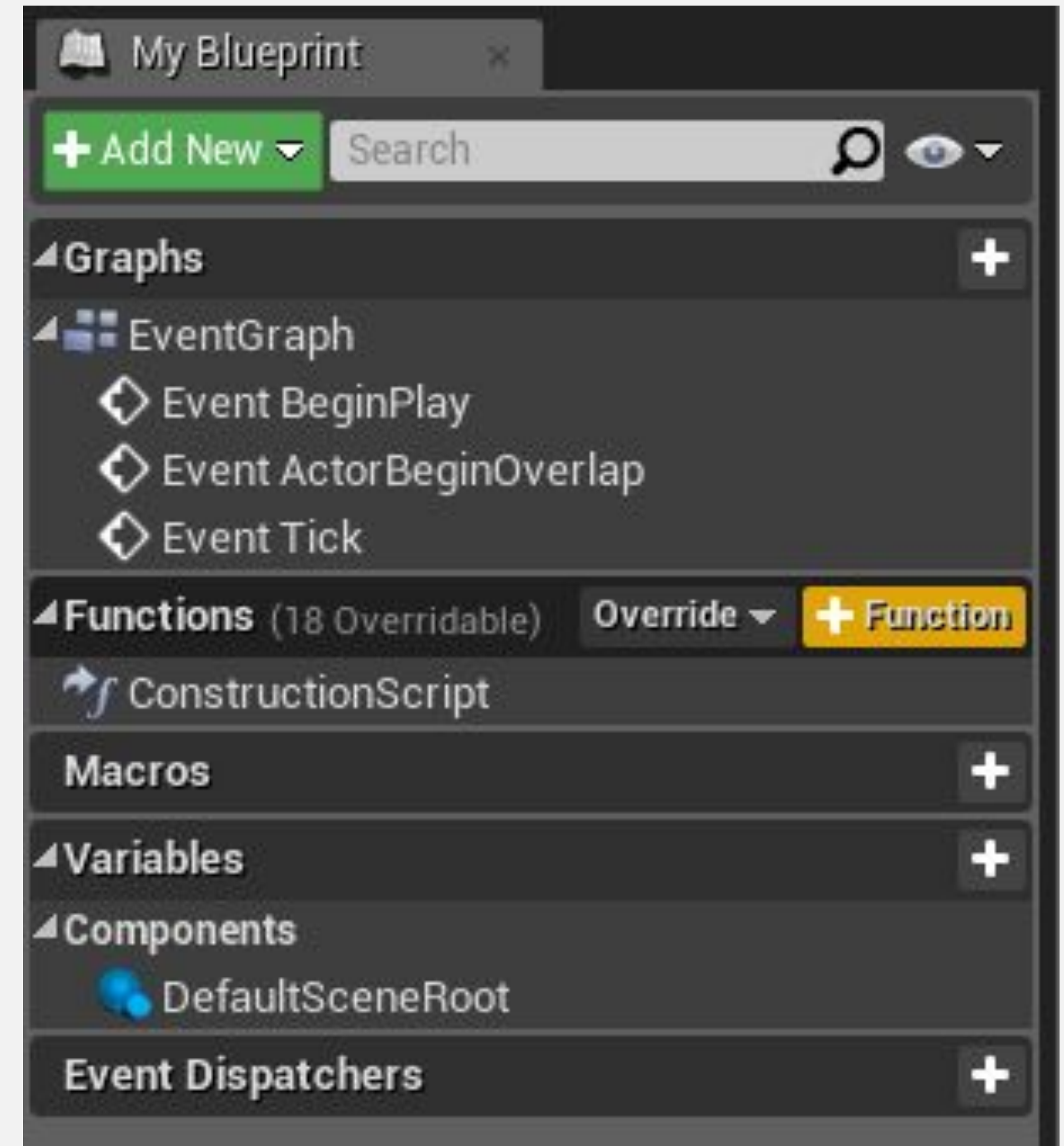


ФУНКЦИИ

Функции позволяют собрать в одном месте набор действий, выполняемых в различных частях Блюпринта, для упрощения организации и обслуживания скрипта.

Функции могут быть вызваны из других Блюпринтов и позволяют использовать параметры ввода и вывода.

Чтобы создать функции, перейдите на панель **My Blueprint** в редакторе Блюпринтов и щелкните символ «+» в категории «**Functions**».





ФУНКЦИИ: ВХОДЫ И ВЫХОДЫ

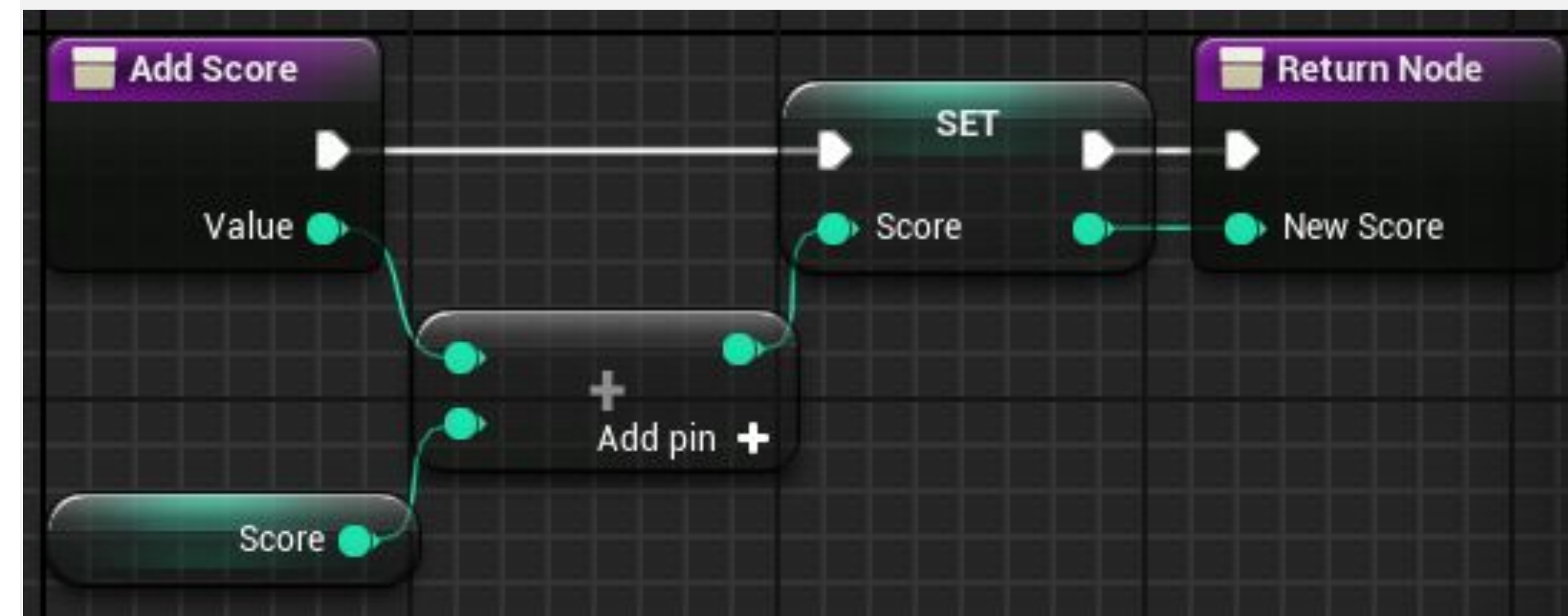
Входные параметры - это значения, которые можно передать в функцию.

Выходные параметры - это значения, которые могут быть возвращены функцией.

Чтобы добавить параметры ввода или вывода, выберите функцию на панели **My Blueprint** и используйте панель **Details**.

На изображениях справа показана функция с входным параметром «**Value**», который добавляется к переменной **Score**.

Результат суммы устанавливается в переменной **Score**, а затем возвращается с выходным параметром с именем «**New Score**».

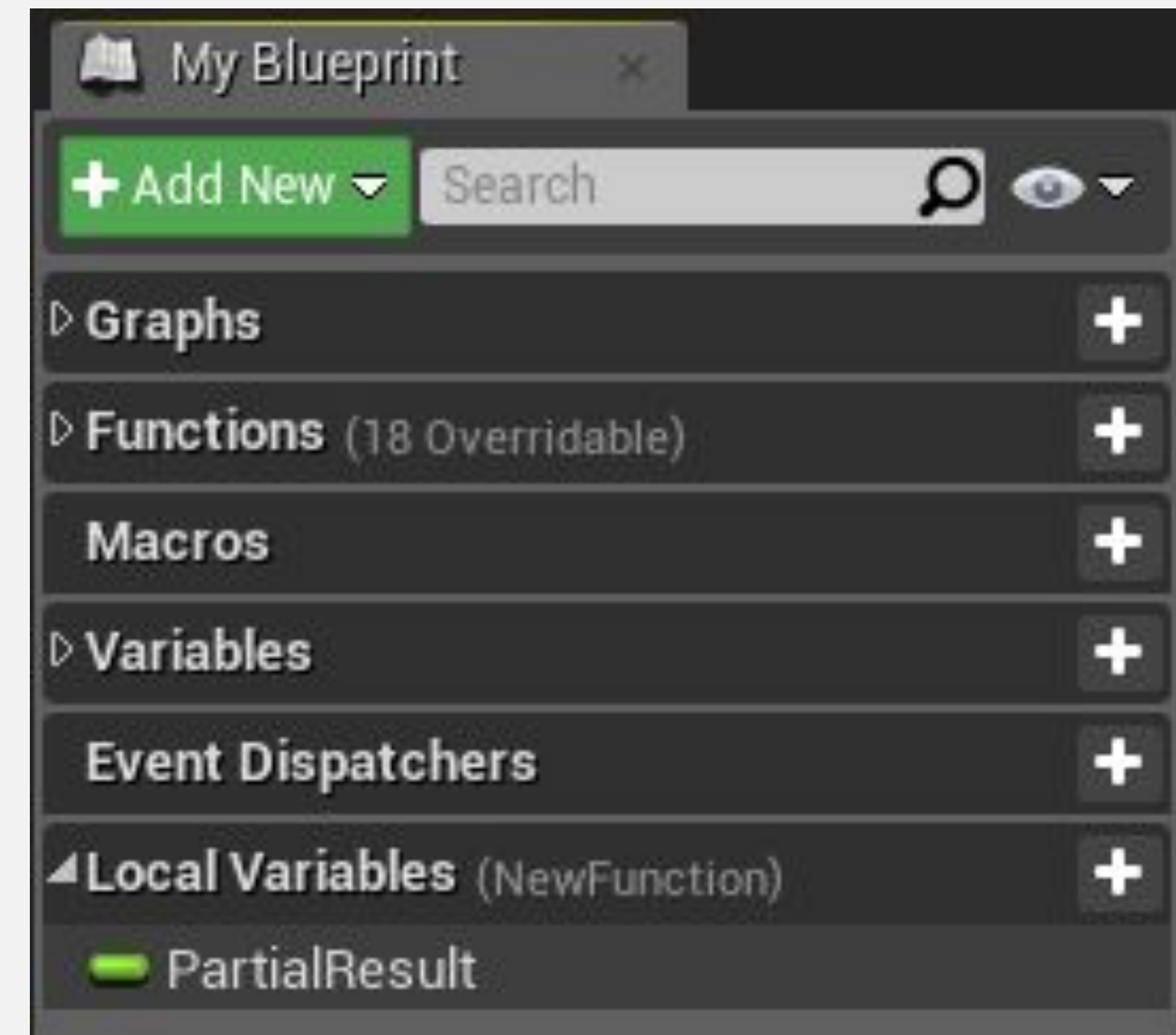




ФУНКЦИИ: ЛОКАЛЬНЫЕ ПЕРЕМЕННЫЕ

Функции позволяют использовать **локальные переменные**, которые видны только внутри функции. Они очень эффективны при выполнении сложных функций и не смешиваются с другими переменными Блюпринта.

Чтобы создать локальную переменную, дважды щелкните по функции, чтобы отредактировать ее, затем посмотрите на панель **My Blueprint**. Внизу панели вы найдете категорию с именем **Local Variables** с именем функции в круглых скобках. Нажмите кнопку «+» в категории **Local Variables**.



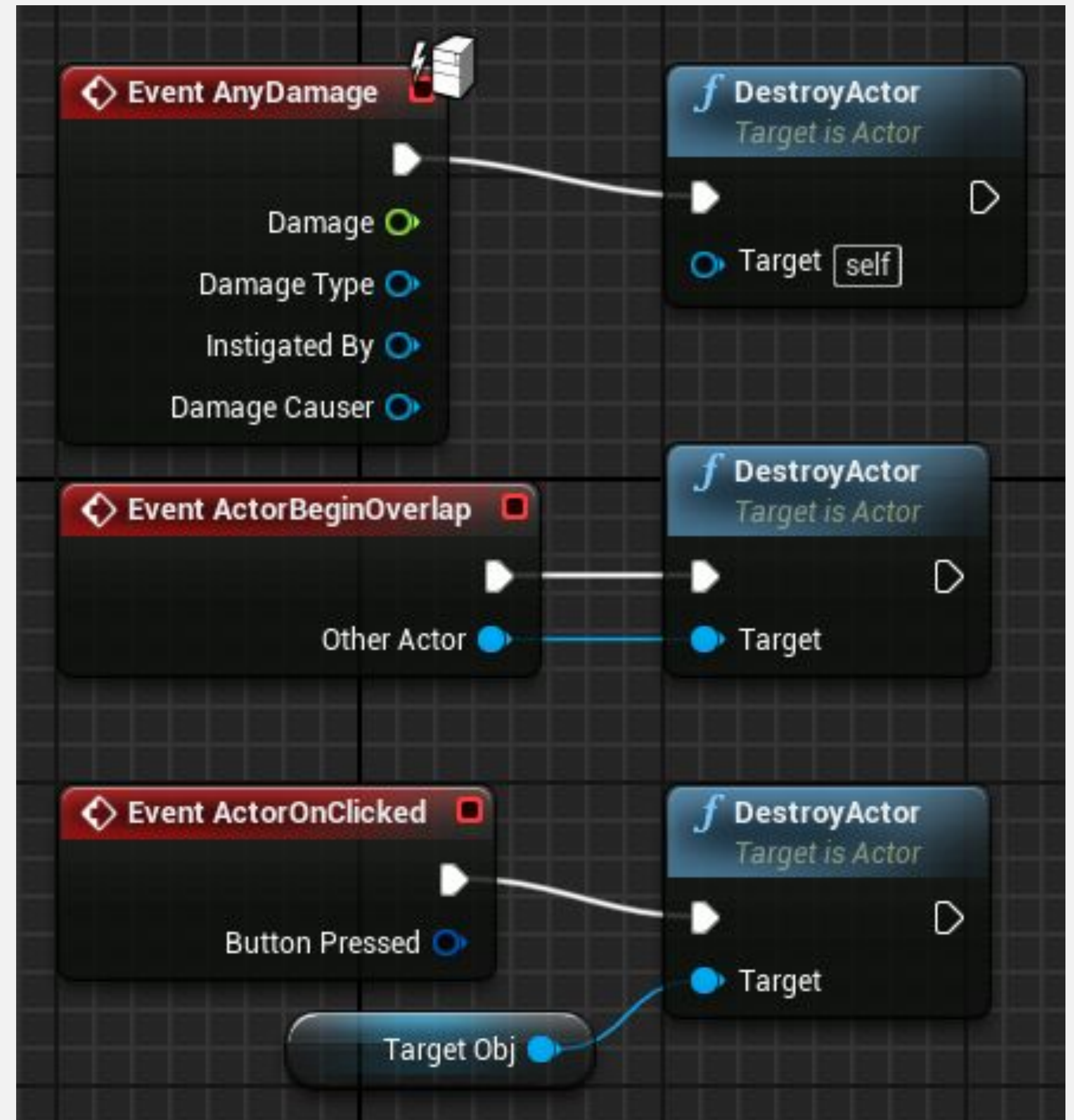


ФУНКЦИИ: ПАРАМЕТР TARGET

Параметр **Target** является общим для нескольких функций и указывает объект, который будет изменен при вызове функции.

Значение по умолчанию для этого параметра - «**self**», которое представляет собой специальную ссылку на Актор или экземпляр объекта, которому принадлежит выполняемый скрипт.

На изображении справа показаны различные способы использования параметра **Target** функции **DestroyActor**.

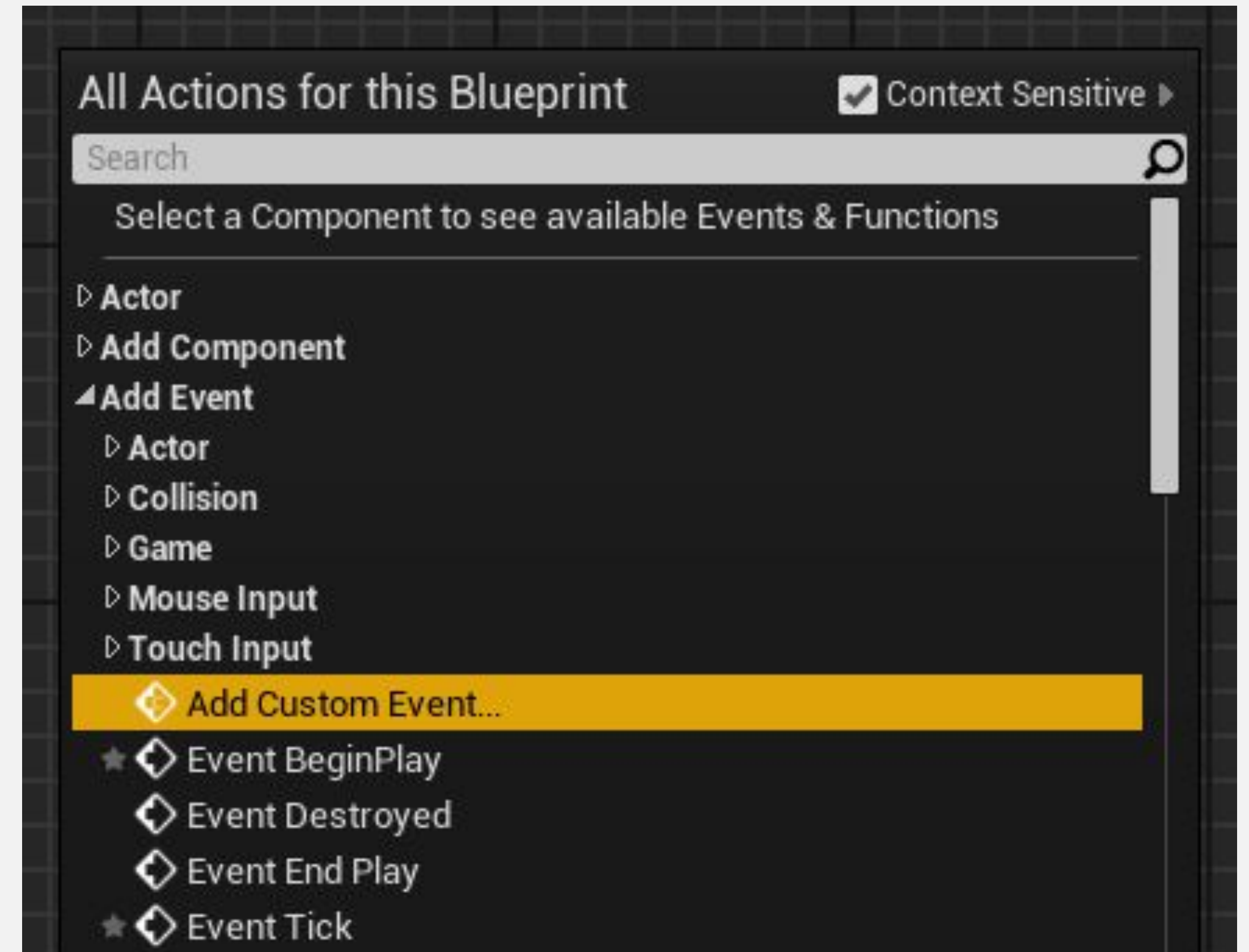




НАСТРАИВАЕМЫЕ СОБЫТИЯ

Unreal Engine предоставляет ряд predefined событий, но можно создавать новые для использования в Блюпринте. Эти события могут быть вызваны как из Блюпринта, в котором они определены, так и из других Блюпринтов.

Чтобы создать **custom event** нажмите ПКМ по **Event Graph** разверните категорию **Add Event** и выберите “**Add Custom Event...**”.



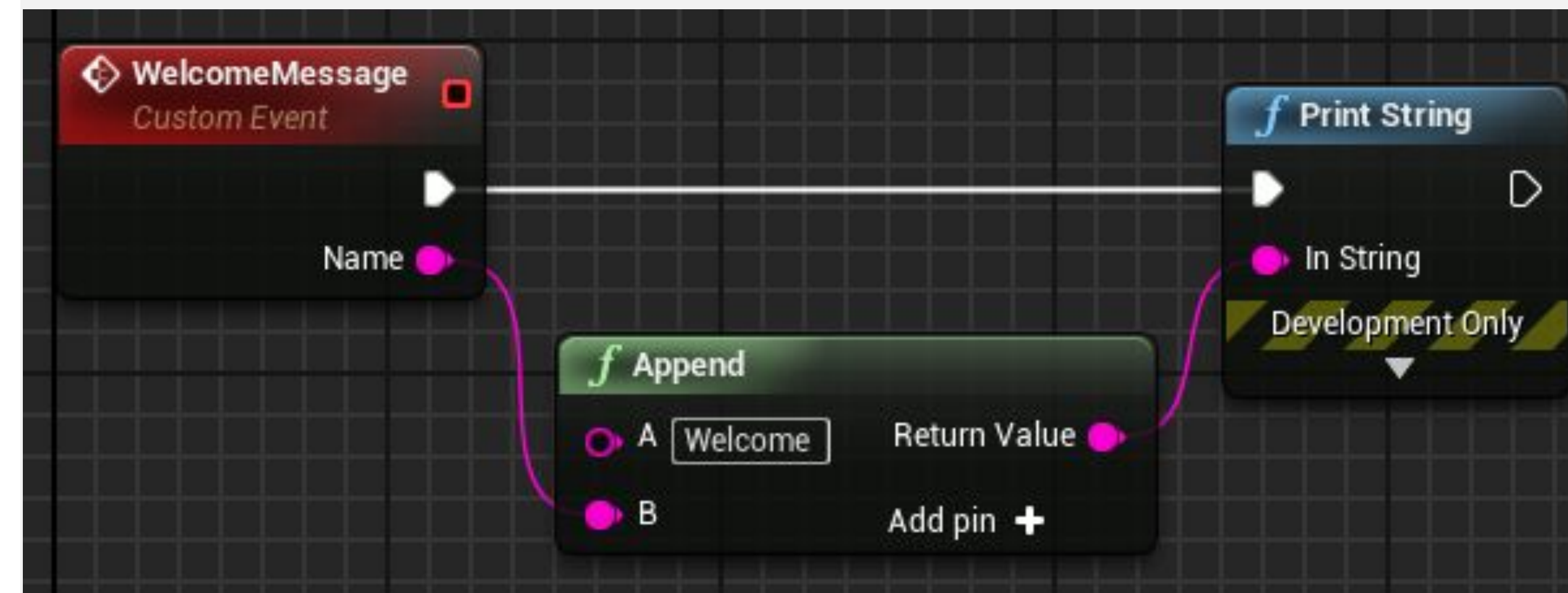
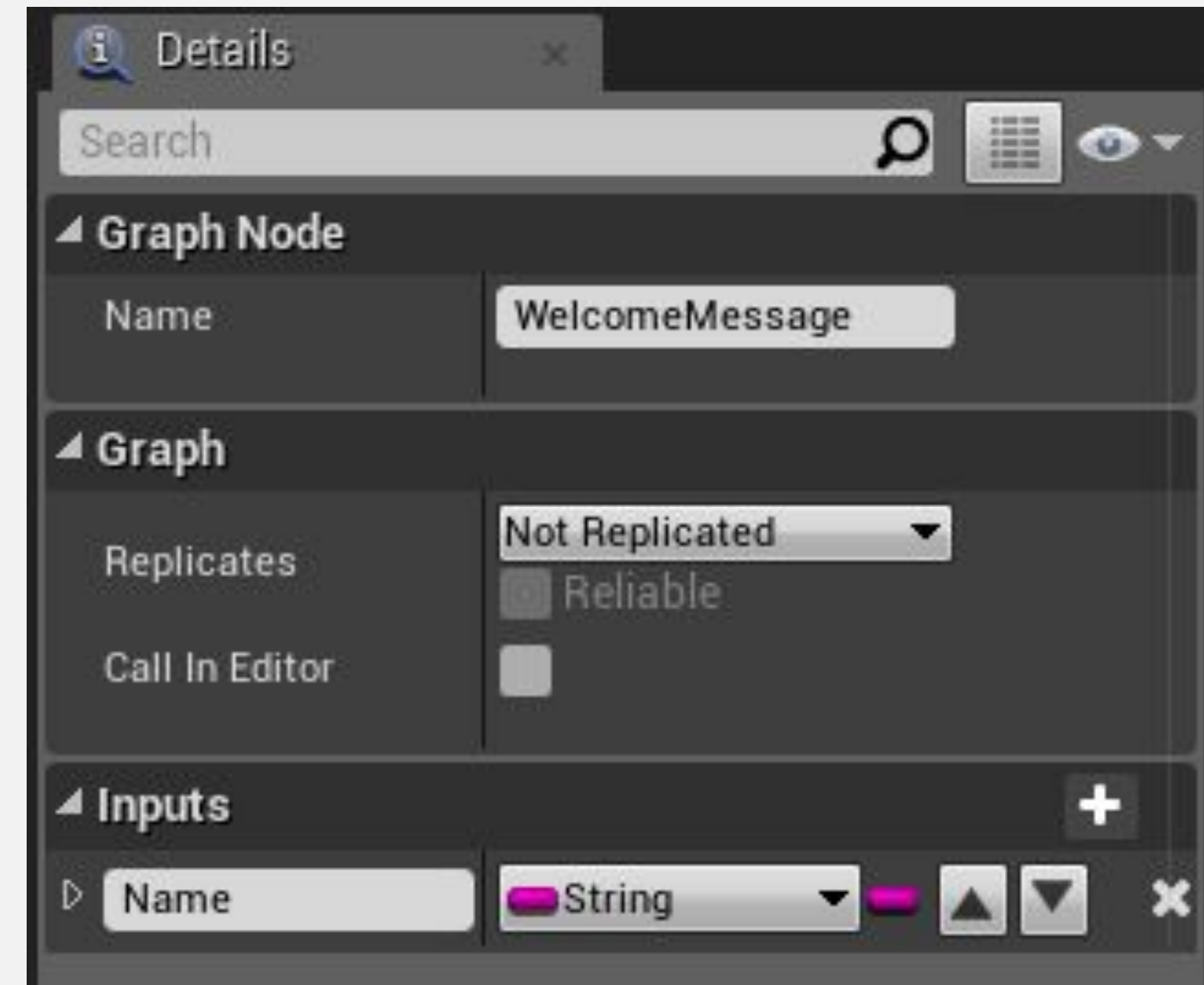


НАСТРАИВАЕМЫЕ СОБЫТИЯ: ВХОДНЫЕ ПАРАМЕТРЫ

Выбор ноды **Custom Event** позволяет управлять именем события и входными параметрами. События не имеют выходных параметров.

На изображениях справа показано настраиваемое событие с именем «**WelcomeMessage**» с входным параметром «**Name**».

Событие создаст и напечатает на экране настраиваемое сообщение, используя имя, переданное в качестве параметра..

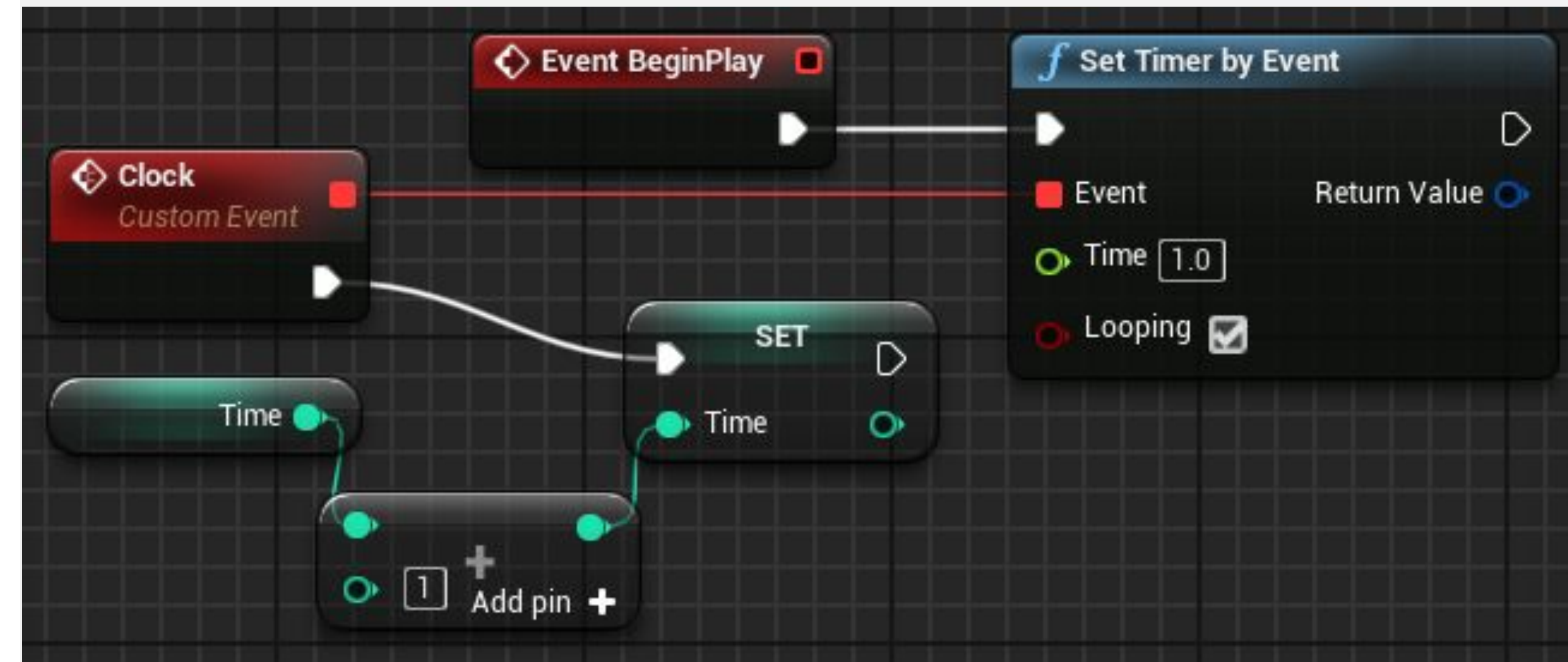




НАСТРАИВАЕМЫЕ СОБЫТИЯ: DELEGATES

В правом углу события отображается небольшой красный квадрат, который называется **delegate**. Это просто отсылка к событию. Некоторые действия получают событие в качестве параметра, и использование делегата делает это возможным.

На изображении справа делегат настраиваемого события с именем «**Clock**» подключен к **Set Timer** с помощью входного контакта **Event** ноды **Event**, поэтому событие **Clock** будет вызываться каждую секунду.

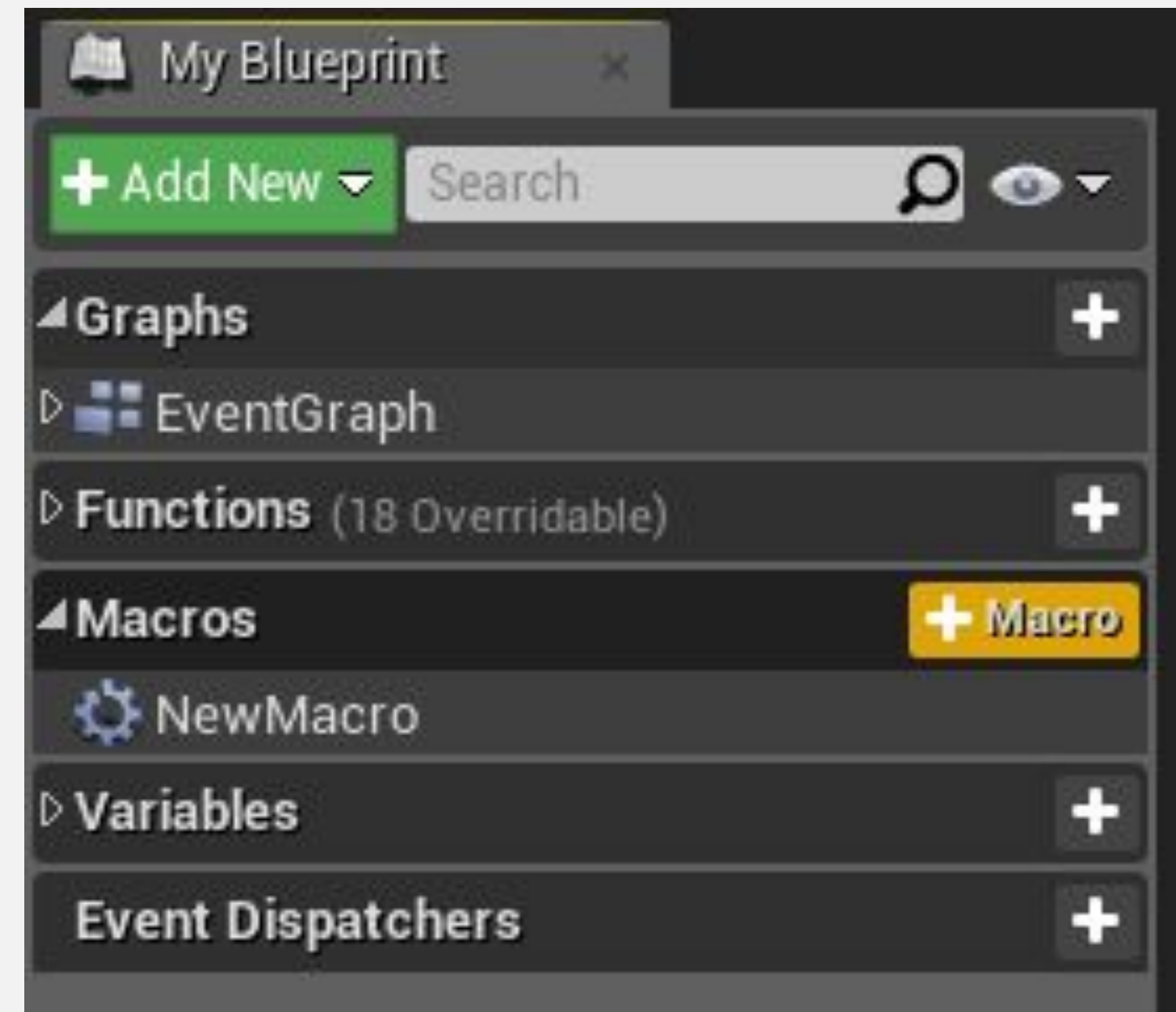




МАКРОСЫ

Другой способ собрать действия в одном месте - использовать **макросы**. Макрос похож на свернутый граф нод. Во время компиляции действия макроса раскрываются в тех местах, где макрос используется.

Макросы могут иметь параметры ввода и вывода, а также несколько выводов выполнения ввода и вывода.





МАКРОСЫ VS. СОБЫТИЯ VS. ФУНКЦИИ

Макросы, настраиваемые события и функции предоставляют различные способы организации сценария. У каждого из них есть свои преимущества и недостатки. Их объединяет то, что все они имеют входные параметры.

- **Макросы** имеют выходные параметры и могут иметь много путей выполнения. Их нельзя вызвать из другого Блюпринта.
- **События** не имеют выходных параметров. Их можно вызывать из других Блюпринтов и иметь ссылку «делегат». Они поддерживают Таймлайны.
- **Функции** может вызываться из других Блюпринтов и иметь выходные параметры. Функции не поддерживают скрытые действия, такие как действие "Задержка» (Delay).

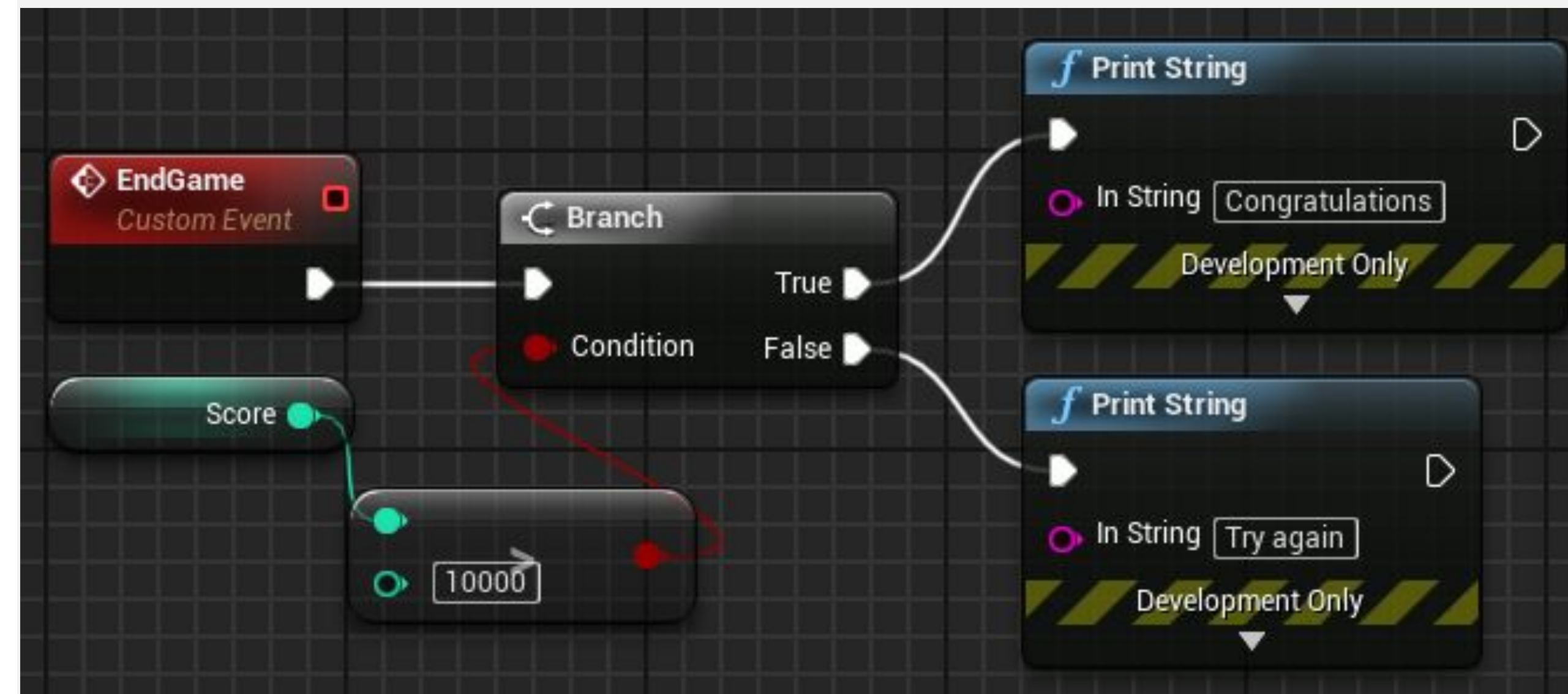
ТЕЧЕНИЕ ПРОГРАММЫ



НОДА BRANCH

Нода **Branch** управляет потоком выполнения Блюпринта на основе значения логического входа «**Condition**», которое может быть «true» или «false».

На изображении справа есть пользовательское событие, которое вызывается в конце игры. Нода **Branch** используется для проверки того, превышает ли результат «**10000**». В зависимости от результата будет показано другое сообщение.





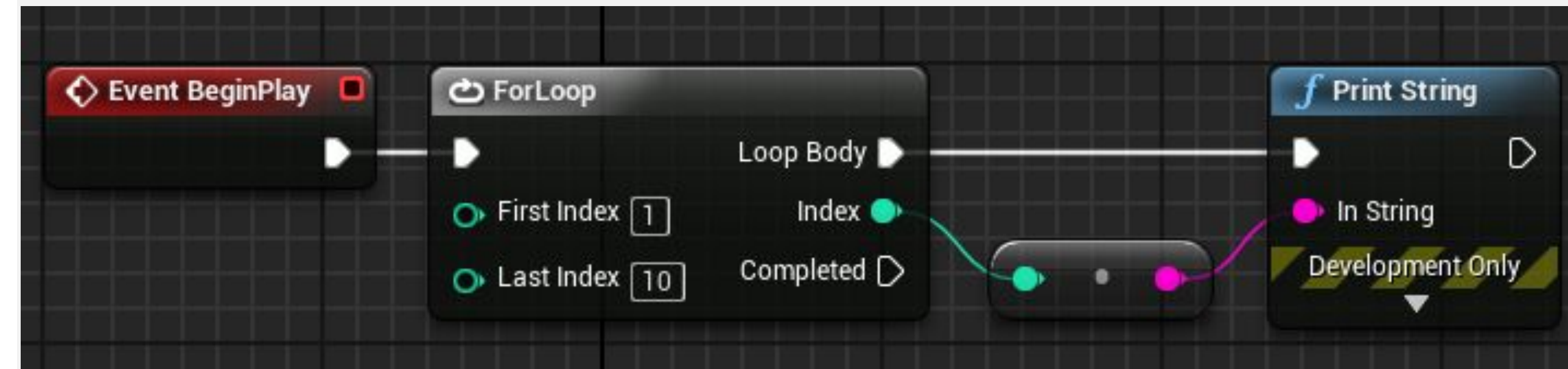
НОДА FOR LOOP

Нода **ForLoop** выполняет набор действий, связанных с выходным контактом **Loop Body** для каждого индекса.

Когда нода **ForLoop** завершает свое выполнение, срабатывает выходной контакт **Complete**.

Справа нода **ForLoop** используется для выполнения ноды **Print String** десять раз. Значение выходного контакта **Index** ноды **ForLoop** используется в качестве входного для ноды **Print String**.

Нода преобразования автоматически создается редактором, когда целочисленное значение подключается к строковому входу.



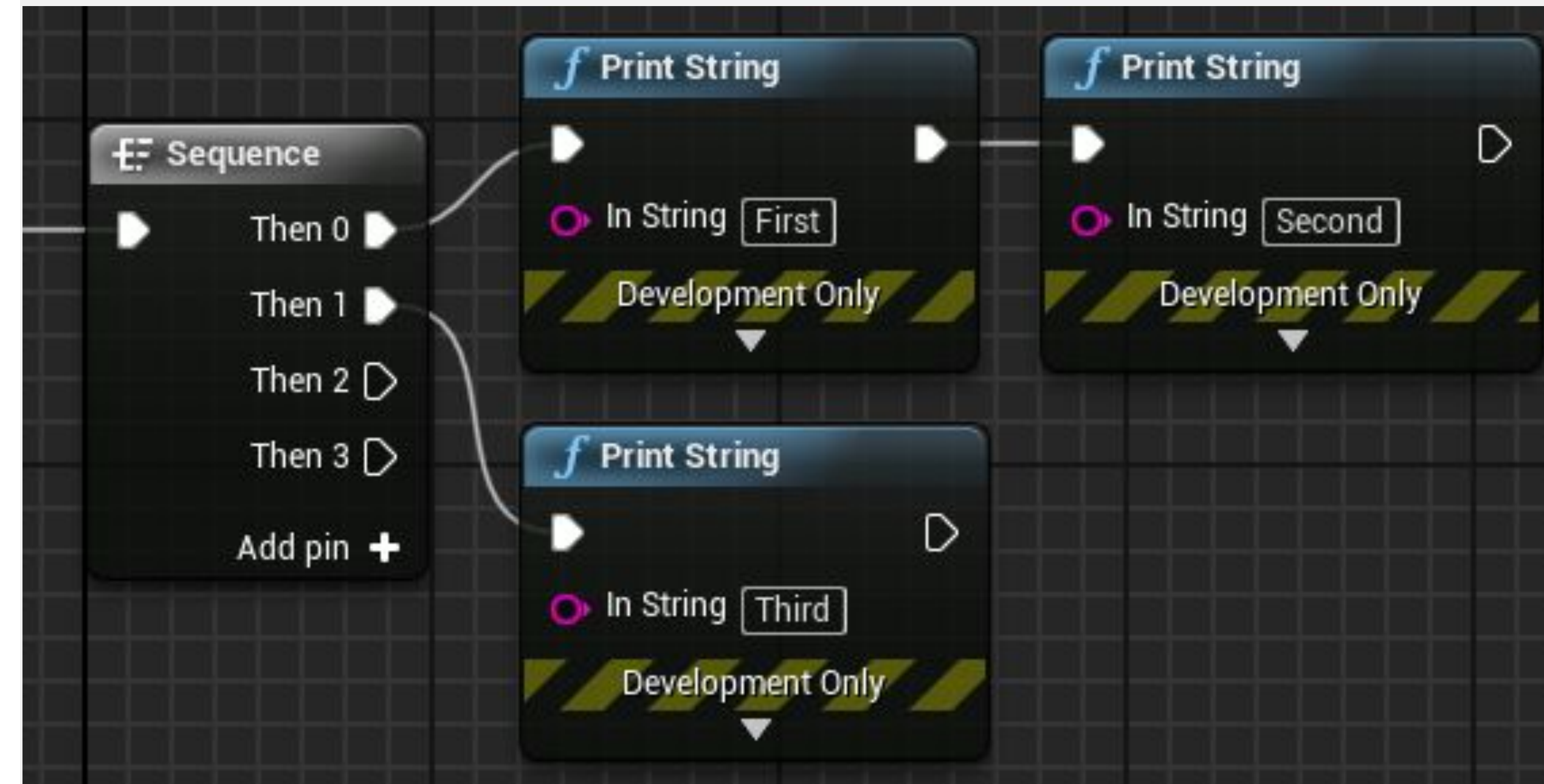
```
10
9
8
7
6
5
4
3
2
1
```




НОДА SEQUENCE

Нода **Sequence** может использоваться для помощи в организации других действий Блюпринтов. При срабатывании триггера он выполняет все ноды, подключенные к выходным контактам, в последовательном порядке, то есть выполняет все действия контакта **Then 0**, затем все действия контакта **Then 1** и так далее.

Выходные контакты могут быть добавлены с помощью опции **Add pin +**. Чтобы удалить пин, щелкните ее ПКМ и выберите параметр **Remove execution pin**.



ИТОГИ

В этой лекции было показано, как создавать переменные и как использовать операторы для создания выражений.

Были представлены функции, пользовательские события и макросы, а в лекции были представлены некоторые базовые ноды, которые управляют ходом выполнения программы.

