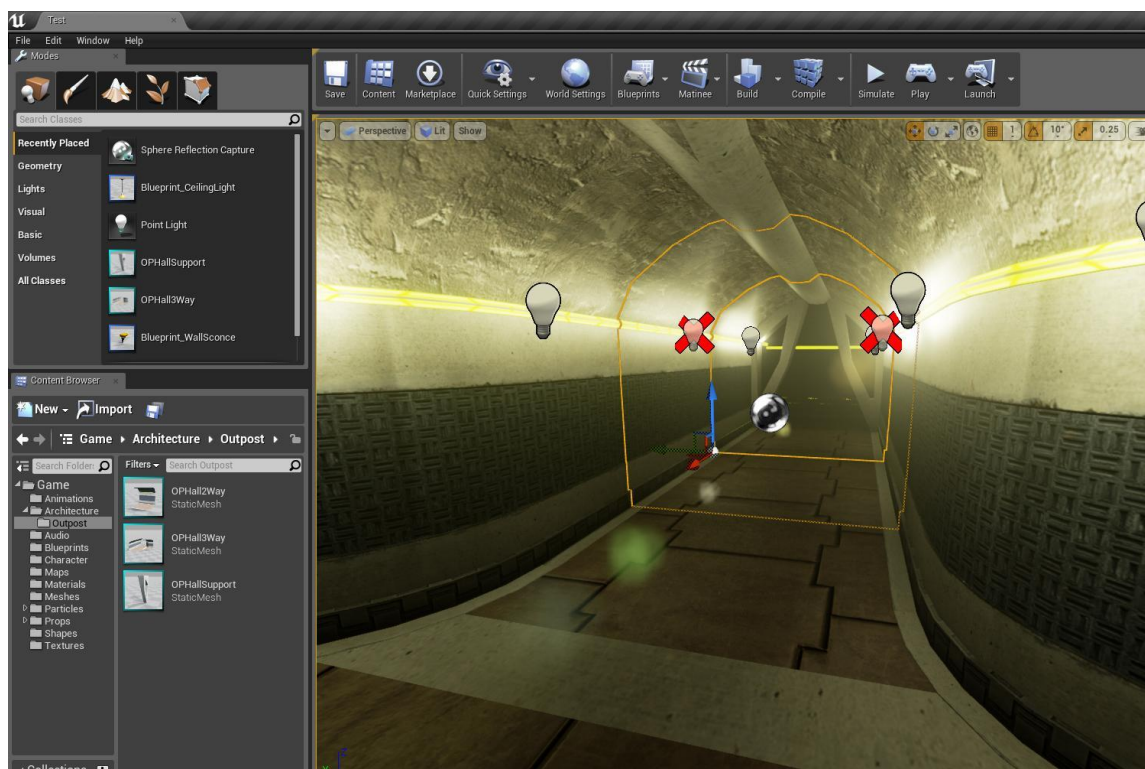


# Тutorial по Unreal Engine. Часть 1:

## ЗНАКОМСТВО С ДВИЖКОМ



Unreal Engine 4 — это набор инструментов для разработки игр, имеющий широкие возможности: от создания двухмерных игр на мобильные до AAA-проектов для консолей. Этот движок использовался при разработке таких игр, как *ARK: Survival Evolved*, *Tekken 7* и *Kingdom Hearts III*.

Разработка в Unreal Engine 4 очень проста для начинающих. С помощью системы визуального создания скриптов Blueprints Visual Scripting можно создавать готовые игры, не написав ни строчки кода! В сочетании с удобным интерфейсом это позволяет быстро изготавливать рабочие прототипы.

В этой части tutorials по Unreal Engine 4 мы ознакомимся с основными возможностями программы. Вот основные темы, которые будут в нём рассмотрены:

- Установка движка
- Импорт ассетов
- Создание материалов
- Использование Blueprints для создания объектов с простейшими функциями

# Установка Unreal Engine 4

Для установки Unreal Engine 4 используется *Epic Games Launcher*. Перейдите на [сайт Unreal Engine](#) и нажмите на кнопку *Get Unreal* в правом верхнем углу.



GET UNREAL

Перед загрузкой программы запуска необходимо будет создать учётную запись. После её создания скачайте программу запуска, соответствующую вашей операционной системе.

Thank you for downloading

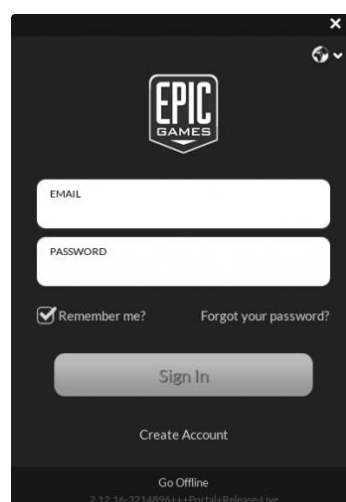


Download for:

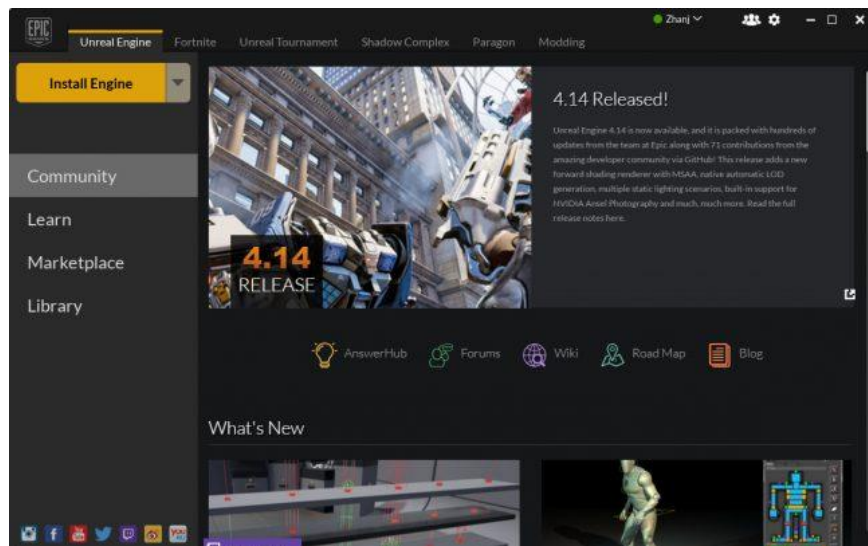
WINDOWS

MAC

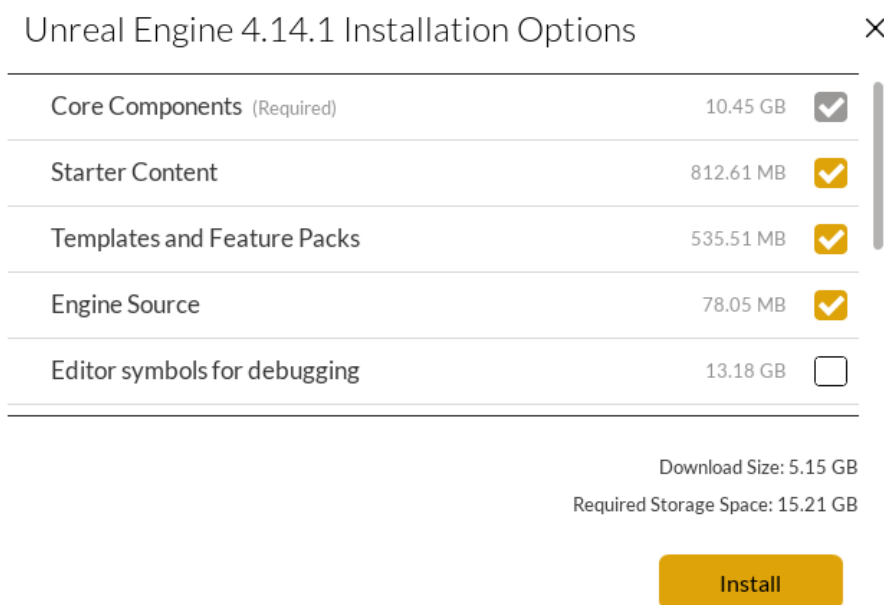
После скачивания и установки программы запуска откройте её. Появится следующее окно:



Введите адрес электронной почты и пароль, использованный для скачивания программы загрузки и нажмите на *Sign In*. После выполнения входа откроется такое окно:



Нажмите на *Install Engine* в левом верхнем углу. Программа запуска перейдёт к экрану, на котором можно будет выбрать устанавливаемые компоненты.



*Примечание:* Epic Games постоянно обновляет Unreal Engine, поэтому ваша версия движка может слегка отличаться от представленной на скриншотах. Например, после написания первого черновика этого туториала версия уже обновилась до 4.14.3! Туториал подойдёт вам, если у вас есть версия не ниже 4.14.

По умолчанию выбраны *Starter Content*, *Templates and Feature Packs* и *Engine Source*. Лучше так всё и оставить. Они будут полезны по следующим причинам:

- *Starter Content*: это коллекция ассетов (ресурсов), которые можно бесплатно использовать в собственных проектах. В том числе это модели и материалы. Их можно использовать как временные ресурсы или в уже готовой игре.
- *Templates and Feature Packs*: шаблоны (Templates) задают базовые возможности выбранного жанра игры. Например, при выборе шаблона *Side Scroller* будет создан

проект с персонажем, простыми движениями и камерой на фиксированной плоскости.

- *Engine Source*: Unreal — это движок с открытым исходным кодом, то есть вносить в него изменения может кто угодно. Если вам понадобится добавить к редактору дополнительные кнопки, то это можно сделать, изменив исходный код.

Под списком есть список возможных платформ. Если не планируете разрабатывать игру под конкретную платформу, то можете спокойно снять все флажки.

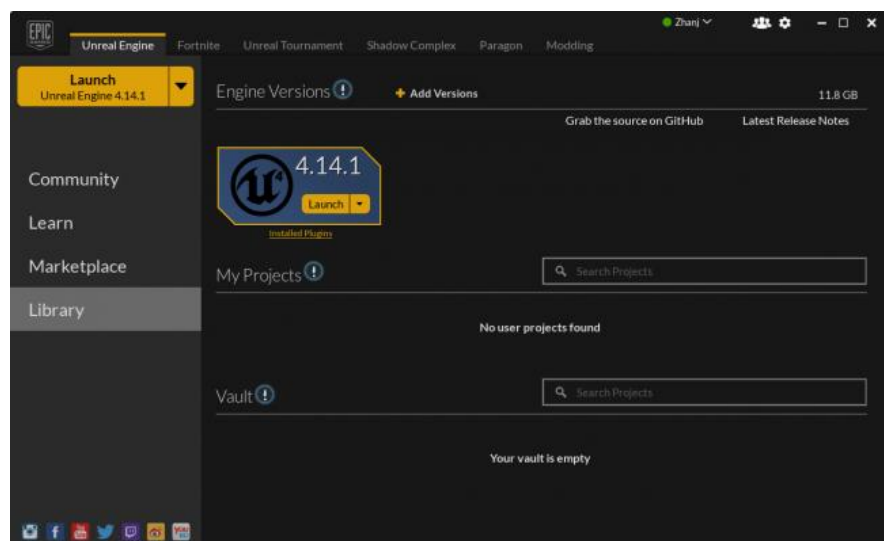
Unreal Engine 4.14.1 Installation Options ×

IOS	869.58 MB	<input checked="" type="checkbox"/>
Android	1.43 GB	<input checked="" type="checkbox"/>
HTML5	562.08 MB	<input checked="" type="checkbox"/>
Linux	545.59 MB	<input checked="" type="checkbox"/>
TVOS	689.96 MB	<input type="checkbox"/>

Download Size: 5.15 GB  
Required Storage Space: 15.21 GB

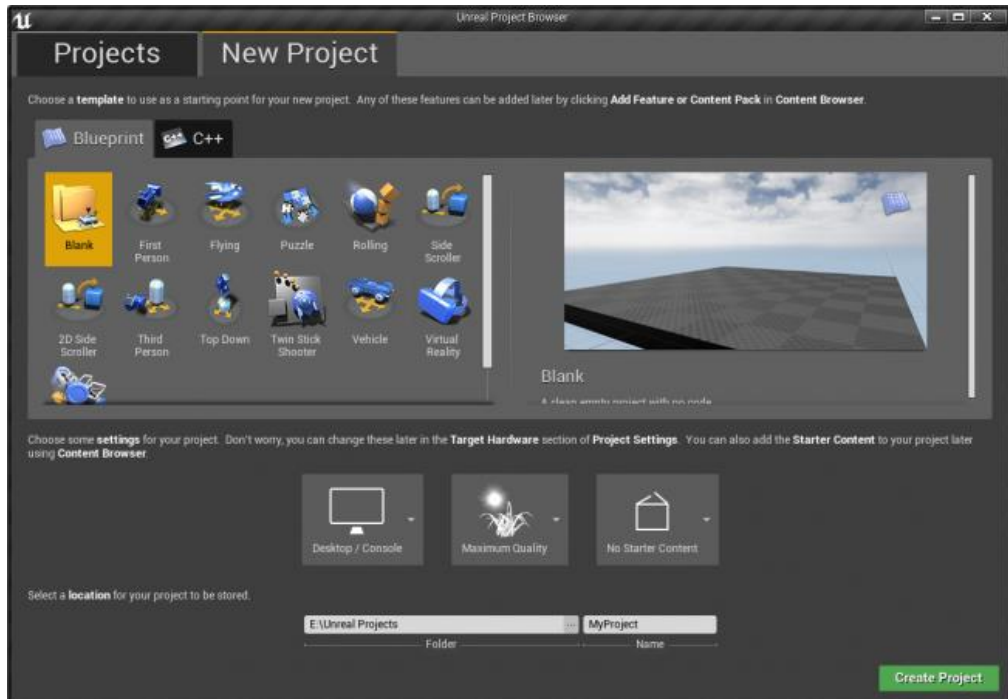
Install

Выбрав нужные компоненты, нажмите на *Install*. После завершения установки движок появится в библиотеке. Теперь настало время создать проект.



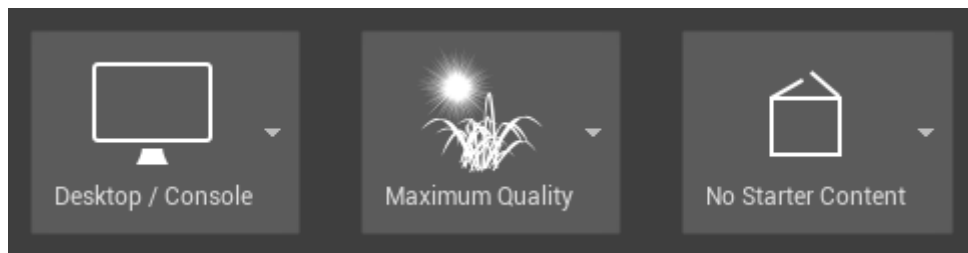
# Создание проекта

Нажмите на одну из кнопок *Launch*, чтобы открыть браузер проектов (Project Browser). После его открытия нажмите на вкладку *New Project*.



Нажмите на вкладку *Blueprint*. Здесь можно выбрать один из шаблонов. Однако, поскольку мы начинаем с нуля, то выберем шаблон *Blank*.

Ниже будут перечисленные дополнительные параметры.



Вот, за что отвечает каждая опция:

- *Target Hardware*: при выборе *Mobile/Tablet* будут отключены некоторые эффекты постобработки. Также можно будет использовать мышь для сенсорного ввода. Выберите здесь опцию *Desktop/Console*.
- *Graphical Target*: при выборе *Scalable 3D or 2D* будут отключены некоторые эффекты постобработки. Выберите здесь *Maximum Quality*.
- *Starter Content*: можно включить эту опцию, чтобы добавить базовый контент (Starter Content). Чтобы нам было проще, выберем *No Starter Content*.

Кроме того, здесь есть раздел для выбора местоположения папки проекта и имени проекта.



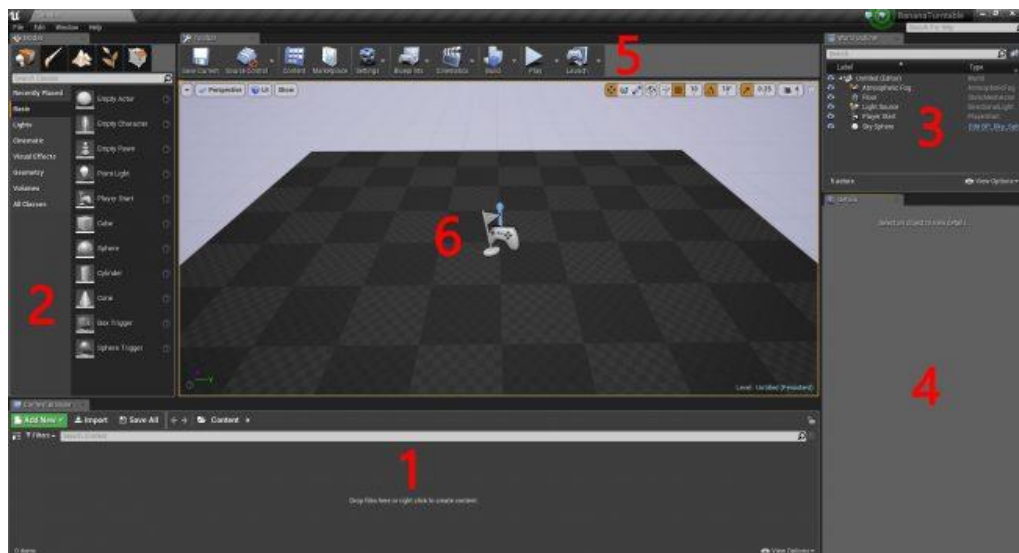
Сменить папку хранения проекта можно, нажав на *многоточие* в конце поля *Folder*.

Имя проекта не является названием игры, так что не волнуйтесь, если хотите название позже. Выберите текст в поле *Name* и введите *BananaTurntable*.

И наконец нажмите на *Create Project*.

## Навигация по интерфейсу

После создания проекта откроется редактор. Он разделён на несколько панелей:



1. *Content Browser*: в этой панели отображаются все файлы проекта. Её можно использовать для создания папок и упорядочивания файлов. Здесь также можно выполнять поиск по файлам с помощью поисковой строки или фильтров.
2. *Modes*: в этой панели можно переключаться между инструментами, например *Landscape Tool* и *Foliage Tool*. Инструментом по умолчанию является *Place Tool*. Он позволяет располагать на уровне различные типы объектов, такие как источники освещения и камеры.
3. *World Outliner*: отображает все объекты на текущем уровне. Можно упорядочить список, распределив связанные объекты по папкам, а также искать и фильтровать их по типам.
4. *Details*: здесь отображаются все свойства выбранного объекта. Эта панель используется для изменения параметров объекта. Внесённые изменения повлияют только на выбранный экземпляр объекта. Например, если в сцене есть две сферы, то при изменении размера одной изменения коснутся только неё.

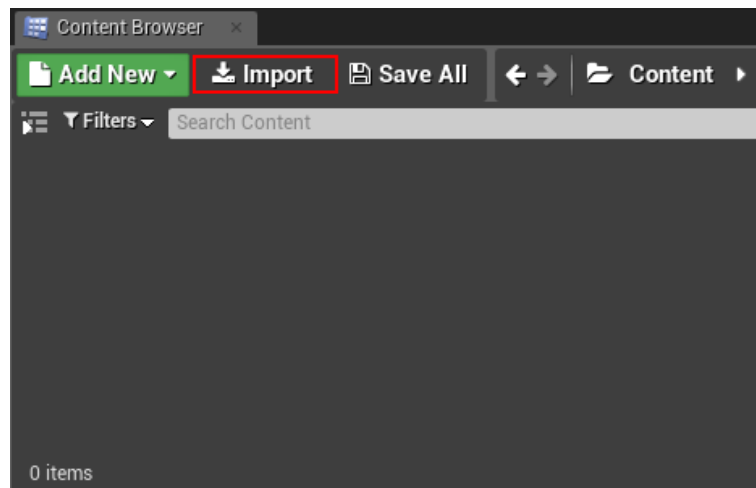


5. *Toolbar*: содержит множество различных функций. Чаще всего мы будем пользоваться *Play*.
6. *Viewport*: это обзор уровня. Осматриваться по сторонам можно, удерживая *правую клавишу мыши* и *перемещая её*. Для перемещения по уровню нужно зажать *right-click* and use the *WASD* keys.

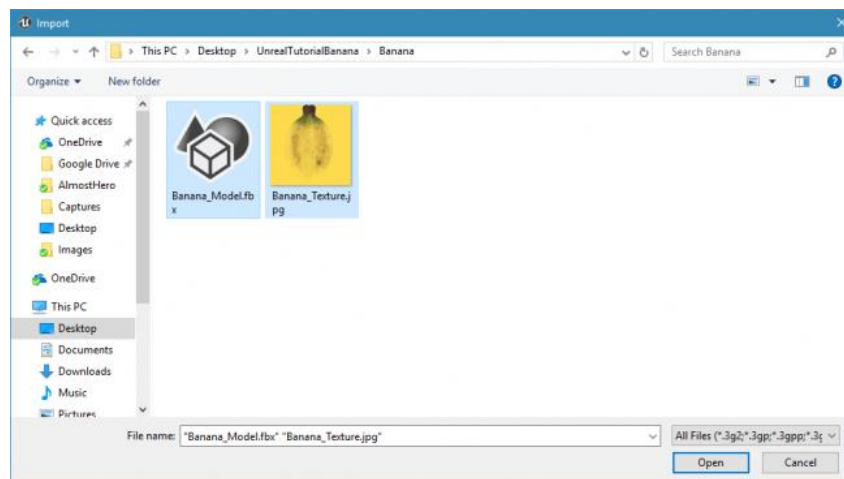
## Импорт ассетов

Какой смысл в поворотном столе, если на нём нечего показывать? <https://koenig-media.raywenderlich.com/uploads/2016/12/UnrealTutorialBanana-1.zip> Внутри находятся два файла: *Banana\_Model.fbx* и *Banana\_Texture.jpg*. Можете также использовать собственную модель, но зачем, если есть такой потрясающий банан?

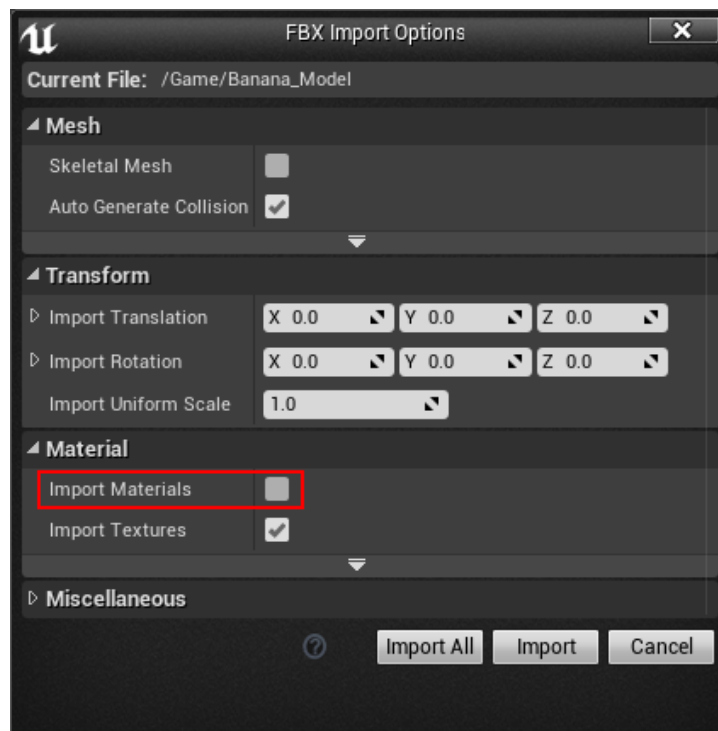
Чтобы Unreal мог использовать файлы, их нужно импортировать. Перейдите в Content Browser и нажмите на *Import*.



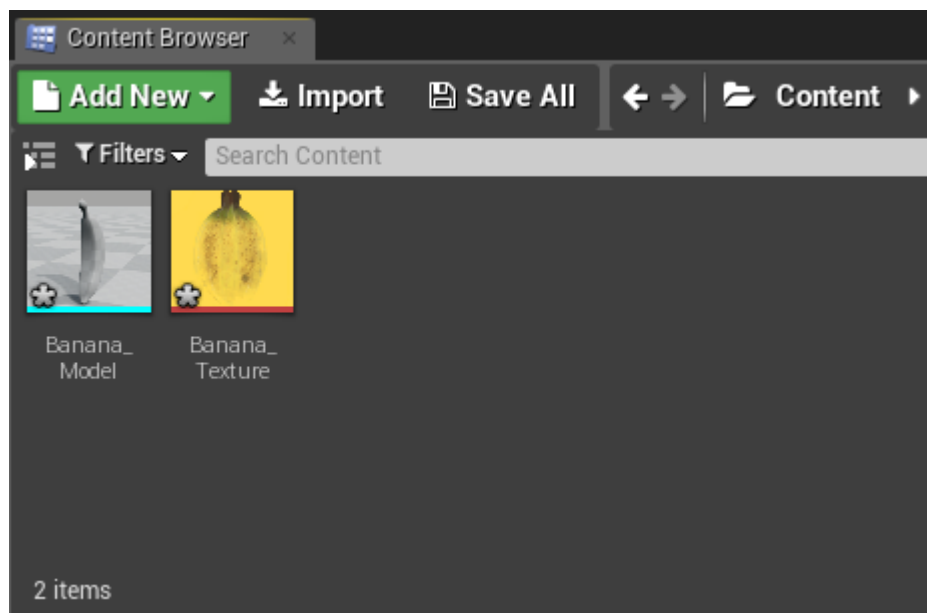
С помощью диспетчера файлов найдите папку, в которой находятся *Banana\_Model.fbx* и *Banana\_Texture.jpg*. Выделите оба файла и нажмите *Open*.



Unreal предложит несколько вариантов импорта файла .fbx. *Снимите* флажок *Import Materials*, потому что мы будем создавать собственный материал. Остальные параметры можно не менять.



Нажмите на *Import*. Два файла появятся в Content Browser.



При импорте файла на самом деле он не сохраняется в проект, если не указать этого явным образом. Сохранять файлы можно, нажав на файл *правой клавишей мыши* и выбрав *Save*. Также можно сохранить все файлы за раз, выбрав *File\Save All*. Старайтесь сохраняться почаще!

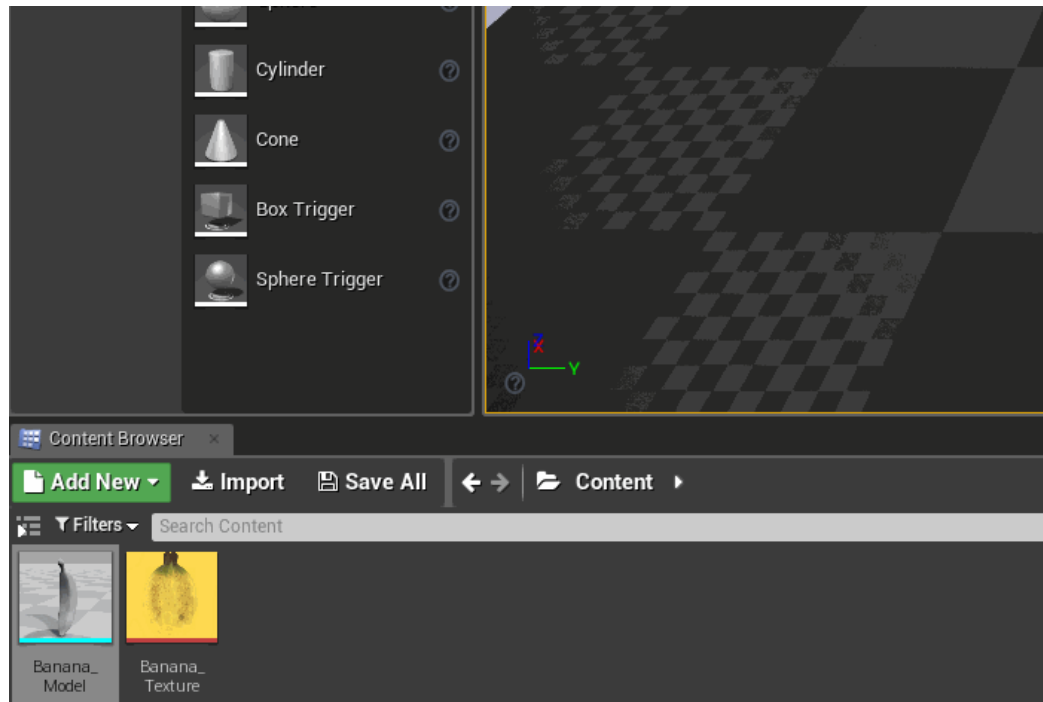


Учтите, что в Unreal модели называются «мешами» (*meshes*). Теперь у нас есть меш банана, настало время добавить его на уровень.

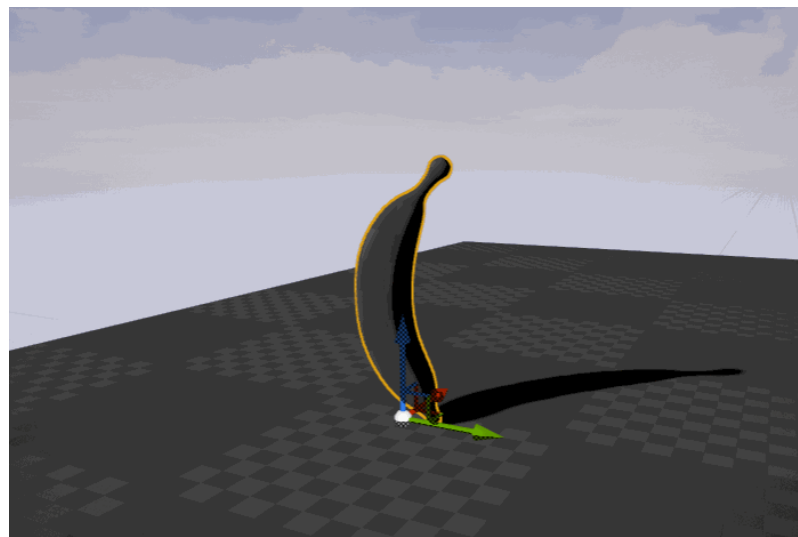
## Добавление мешей на уровень

Пока уровень выглядит довольно пустым, давайте сделаем его интереснее.

Чтобы добавить на уровень меш, *нажмите правой клавишей мыши и перетащите Banana\_Model* из Content Browser во Viewport. Отпустите *левую клавишу мыши* и меш добавится на уровень.



Объекты на уровне можно *перемещать*, *поворачивать* и *масштабировать*. Горячие клавиши для этих действий — *W*, *E* и *R*. После нажатия на них можно использовать манипулятор:



## О материалах

Если внимательно посмотреть на банан, то можно увидеть, что он не жёлтый! На самом деле, он выглядит почти полностью серым.

Чтобы придать банану цвет и детали, необходимо создать *материал*.

### Что такое «материал»?

Материал задаёт внешний вид поверхности. На базовом уровне материал определяет четыре аспекта:

- *Base Color*: цвет или текстуру поверхности. Используется для добавления детализации и вариаций цвета.
- *Metallic*: насколько металлическим выглядит материал. В общем случае, чисто металлический объект имеет максимальное значение *Metallic*, а ткань — значение 0.
- *Specular*: управляет блеском неметаллических поверхностей. Например, керамика будет обладать высоким значением *Specular*, в отличие от глины.
- *Roughness*: поверхность с максимальной *roughness* (шероховатостью) не будет иметь никакого блеска. Используется для таких поверхностей, как камень и дерево.

Ниже представлен пример трёх различных материалов. Они имеют одинаковый цвет, но разные атрибуты. Каждый материал имеет высокое значение соответствующего атрибута. Значение остальных атрибутов равно нулю.



Metallic



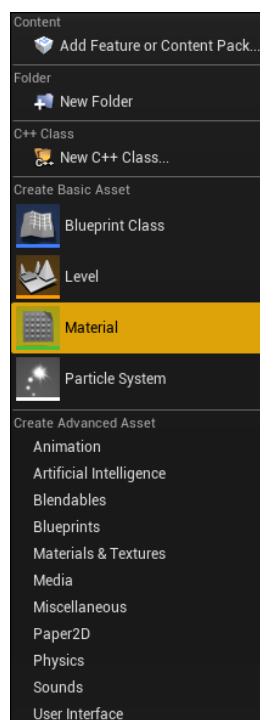
Specular



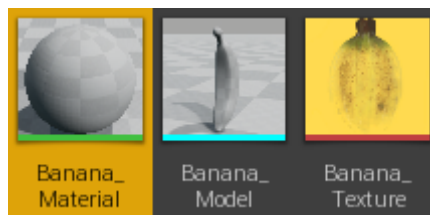
Roughness

## Создание материала

Для создания материала нужно перейти в Content Browser и нажать на зелёную кнопку *Add New*. Появится меню со списком ассетов, которые можно создать. Выберите *Material*.

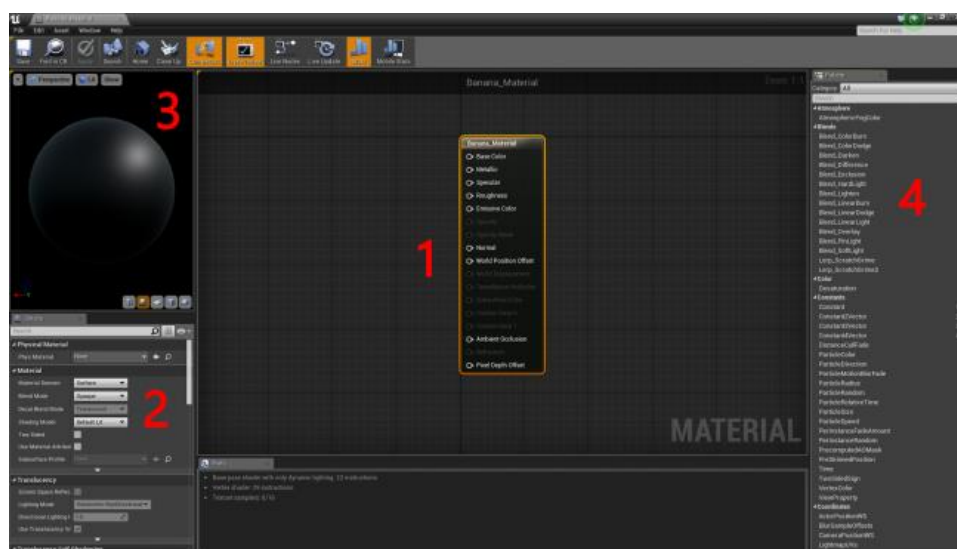


Назовите материал *Banana\_Material*, а затем *дважды нажмите* левой клавишей мыши на файле, чтобы открыть его в редакторе материалов.



## Редактор материалов (Material Editor)

Редактор материалов состоит из пяти основных панелей:



1. *Graph*: в этой панели содержатся все ноды (узлы) и нод *Result*. Перемещаться по панели можно, удерживая *правую клавишу мыши* и *двигая мышью*. Масштабирование выполняется *прокруткой колёсика мыши*.
2. *Details*: здесь отображаются свойства выбранного нода. Если нод не выбран, то здесь отображаются свойства материала.
3. *Viewport*: содержит меш предварительного просмотра, представляющий материал. Поворачивать камеру можно, удерживая *левую клавишу* и *двигая мышью*. Масштабирование выполняется *прокруткой колёсика мыши*.
4. *Palette*: список всех нодов, доступных для вашего материала.

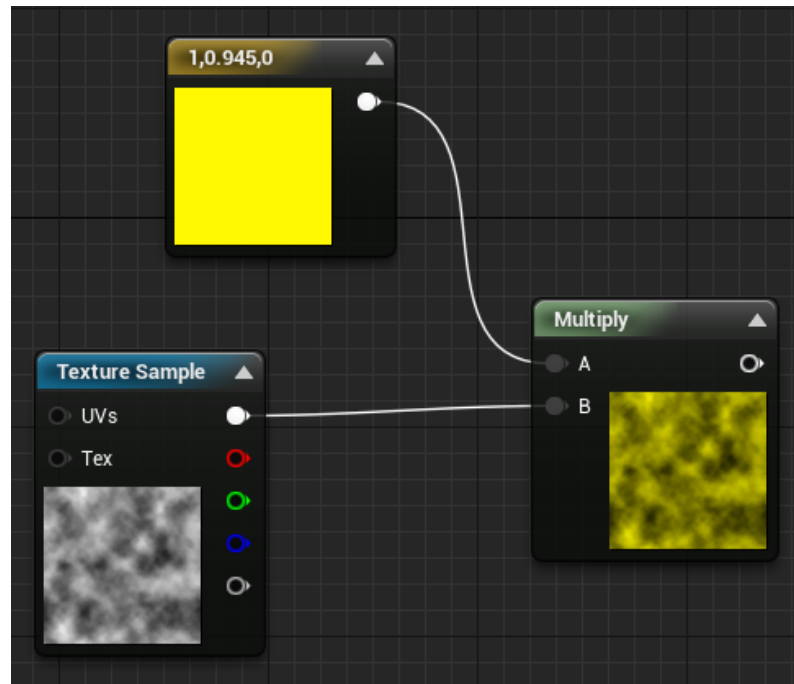
## Что такое нод (узел)?

Прежде чем приступать к созданию своего материала, нужно узнать про объекты, которые используются для его создания: *ноды*.

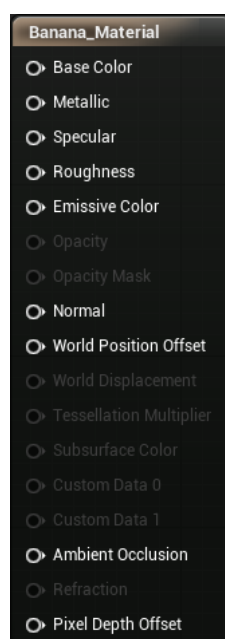
Ноды составляют большую часть материала. Существует множество типов нодов, имеющих различные функции.

У нодов есть входы и выходы, представленные кругом со стрелкой. Входы расположены слева, а выходы — справа.

Вот пример с использованием нодов *Multiply* и *Constant3Vector*, добавляющих текстуре жёлтого цвета:



У материалов есть особый нод, называемый нодом *Result*, который в нашем случае уже создан как *Banana\_Material*. Здесь заканчиваются со временем все ноды. То, что подключено к этому ноду, определяет внешний вид конечного материала.

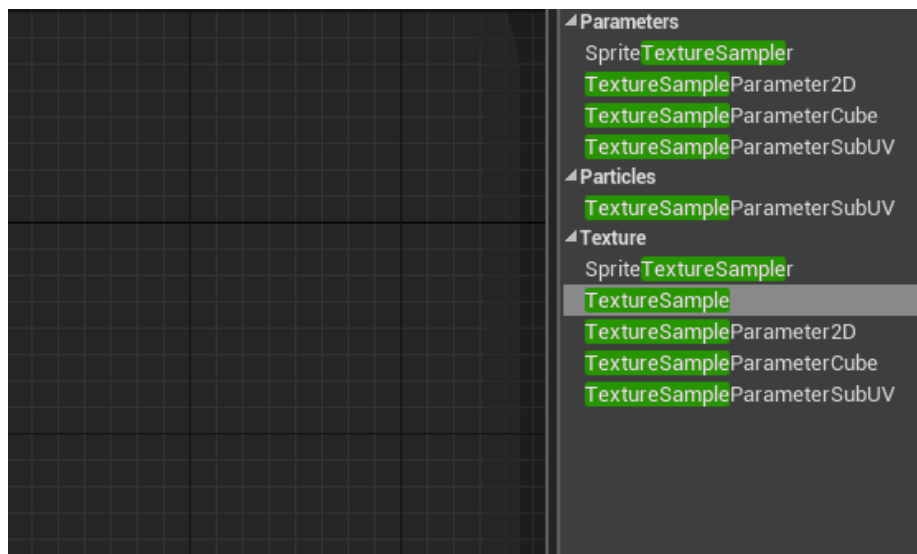


## Добавление текстур

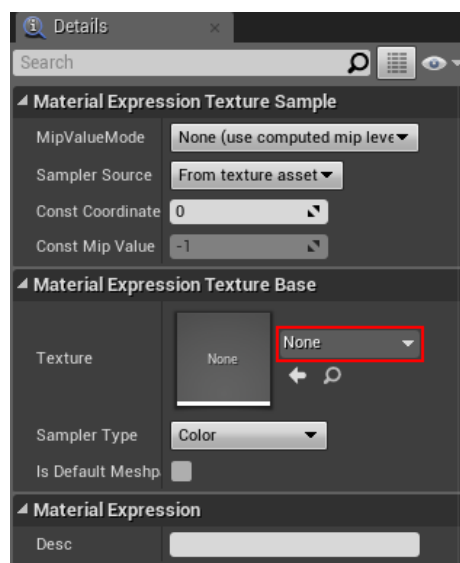
Для добавления модели цвета и деталей нам необходима *текстура*. Текстура — это просто двухмерное изображение. Обычно они проецируются на трёхмерные модели, придавая им цвет и детали.

Для текстурирования банана мы используем *Banana\_Texture.jpg*. Применить в материале текстуру позволяет нод *TextureSample*.

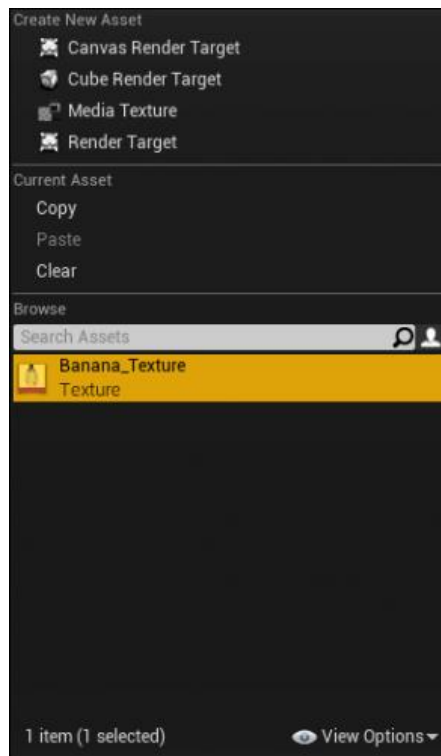
Перейдите к панели *Palette* и найдите *TextureSample*. Добавьте нод, удерживая *левую клавишу мыши* и *перетаскив* его на схему.



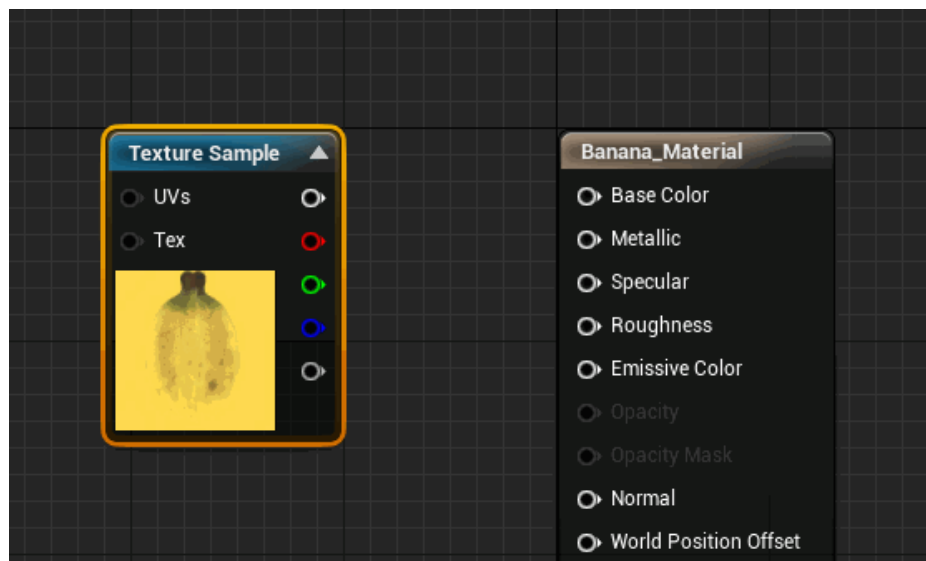
Для выбора текстуры необходимо сначала выделить нод *TextureSample*. Перейдите в панель *Details* и нажмите на *раскрывающийся список*, расположенный справа от *Texture*.



Откроется меню, в котором перечислены все текстуры проекта. Выберите *Banana\_Texture*.

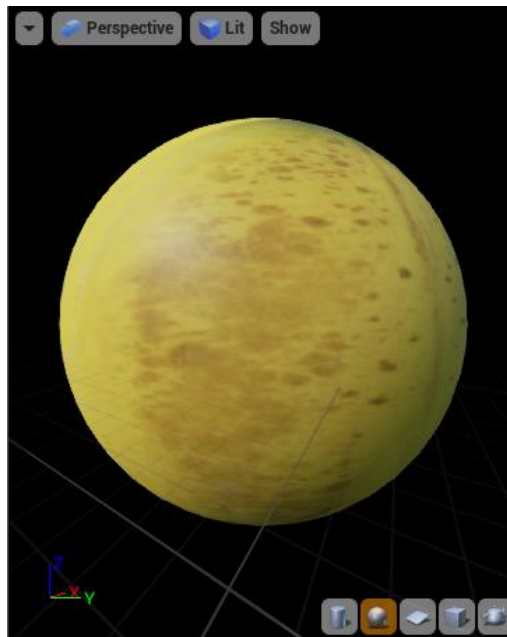


Чтобы увидеть текстуру на меше предварительного просмотра, нужно подключить её к ноду *Result*. Удерживайте левую клавишу мыши на белом контакте выхода ноды *TextureSample*. Перетащите его на входной контакт *Base Color* ноды *Result*.



Вернитесь во Viewport, чтобы увидеть текстуру на меше предварительного просмотра. Можно поворачивать его (удерживая левую клавишу мыши и перемещая мышь), чтобы рассмотреть другие детали.

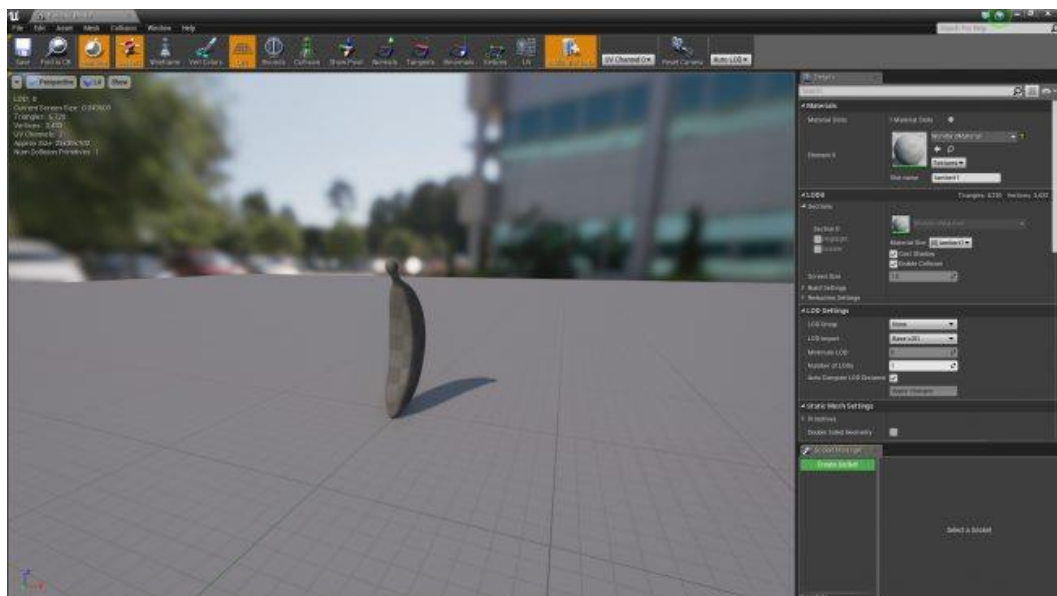




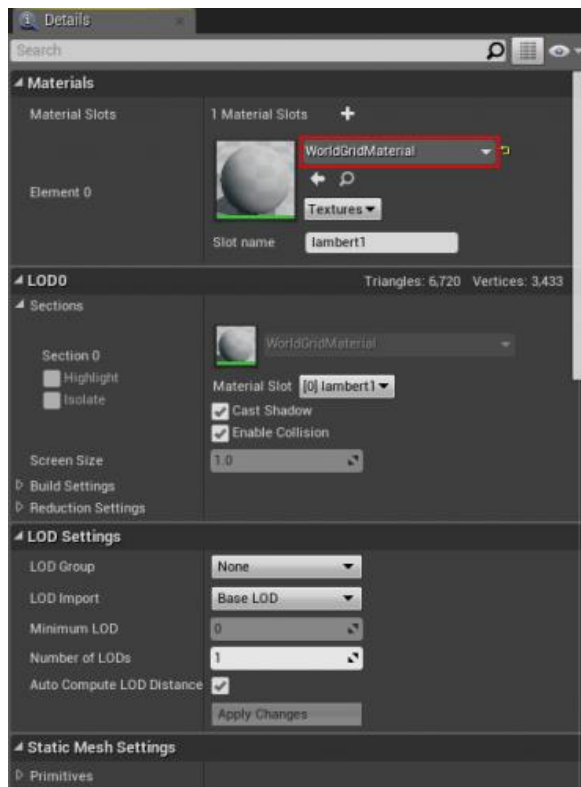
Нажмите на *Apply* в Toolbar, чтобы обновить материал, и закройте редактор материалов после завершения.

## Использование материалов

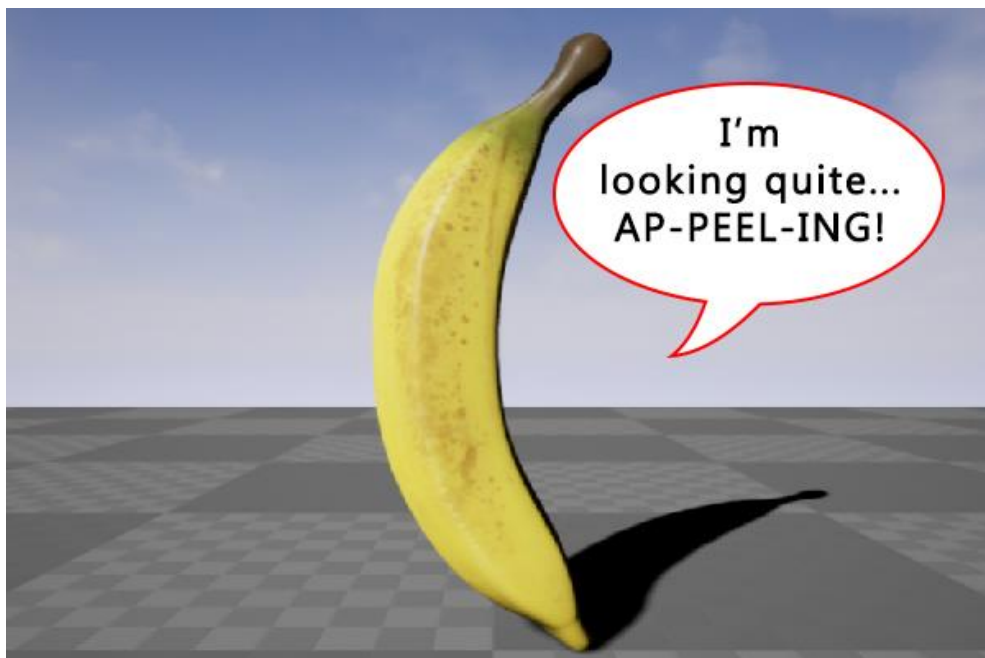
Чтобы применить материал на банан, нужно его назначить. Вернитесь к Content Browser и *дважды нажмите* на *Banana\_Model*, чтобы открыть его. Появится следующее окно редактора:



Перейдите в панель Details и найдите раздел *Materials*. Нажмите на *раскрывающееся меню*, расположенное справа от *Element 0*, и выберите *Banana\_Material*.



Закройте редактор мешей, вернитесь к основному редактору и посмотрите на Viewport. Вы увидите, что теперь на банане есть текстура. Поздравляю, вы теперь знаете всё необходимое, чтобы стать дизайнером уровней!



*Примечание:* если освещение слишком тёмное, можно изменить его, зайдя в World Outliner и нажав на *Light Source*. В панели Details найдите параметр *Intensity* и увеличьте его значение.

# Про Blueprints

Даже несмотря на то, что банан выглядит отлично, будет ещё лучше, если он начнёт вращаться на поворотном столе. Проще всего создать его с помощью «чертежей» *Blueprints*.

В простейшем случае Blueprint представляет собой «вещь». Blueprints позволяют создавать свои поведения для объектов. Объект может быть чем-то физическим (типа поворотного стола) или чем-то абстрактным, например, системой здоровья.

Хотите создать движущийся автомобиль? *Используйте Blueprint*. А как насчёт летающей свинки? *Используйте Blueprints*. А если нужен взрывающийся при касании котик? *Blueprints*.

Как и в материалах, в Blueprints используется система на основе нодов. Это значит, что достаточно создать ноды и соединить их — никакого кода не требуется!

*Примечание:* если вы предпочитаете писать код, то используйте вместо этого C++.

Blueprints просты в использовании, однако не так быстры, как код на C++. То есть если вам нужно создать что-то «тяжёлое» с точки зрения вычислений, например, сложный алгоритм, то лучше воспользоваться C++.

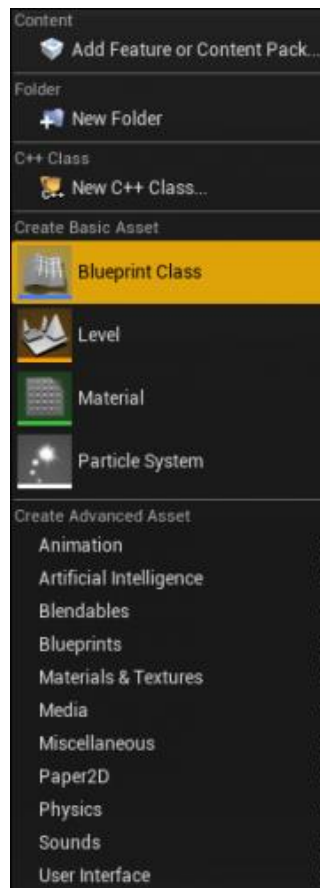
Но даже если вы предпочитаете C++, то бывают случаи, когда оптимальнее использовать Blueprints. Вот некоторые из преимуществ Blueprints:

- Обычно разработка на Blueprints быстрее, чем на C++.
- Простота упорядочивания. Можно разделять ноды на разные области, например, на функции и графы.
- Если вы работаете с людьми, не знающими программирование, то изменение Blueprint проще благодаря их наглядности и интуитивной понятности.

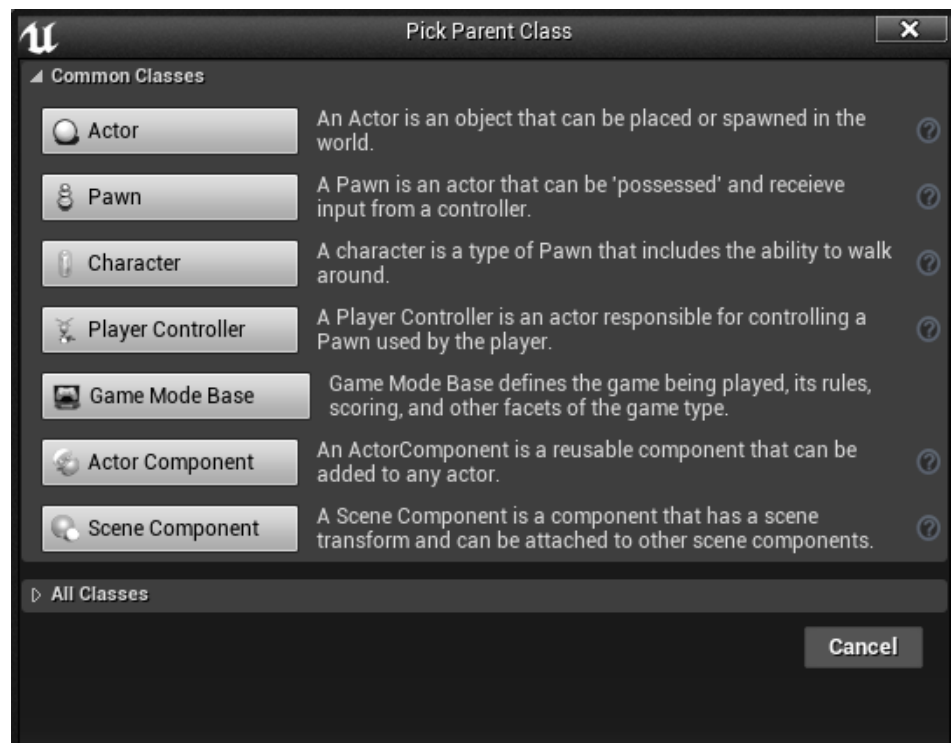
Хорошим подходом будет создание объектов с помощью Blueprints. А когда требуются дополнительные возможности, преобразование их в C++.

## Создание Blueprint

Перейдите в Content Browser и нажмите на *Add New*. Выберите в списке *Blueprint Class*.



Откроется окно с запросом выбора родительского класса. Ваш Blueprint будет наследовать все переменные, функции и компоненты из выбранного родительского класса. Уделите время на изучение возможностей каждого класса.

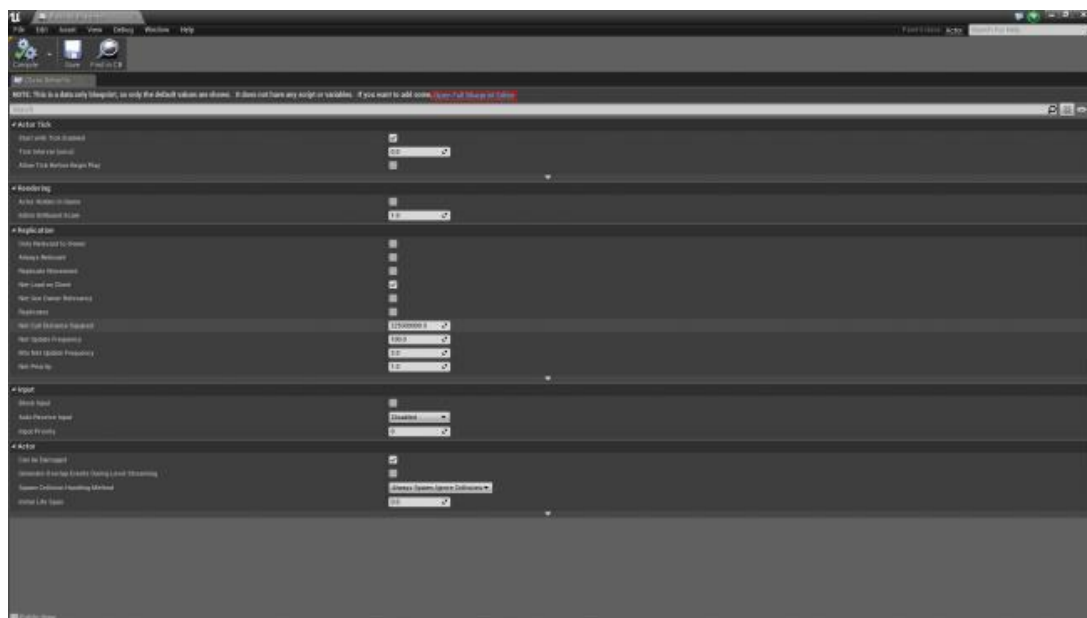


*Примечание:* поскольку мы можем расположить классы *Pawn* и *Character*, они также являются актёрами (*Actors*).

Поворотный стол будет находиться на месте, поэтому самым подходящим будет класс *Actor*. Выберите *Actor* и назовите новый файл *Banana\_Blueprint*.



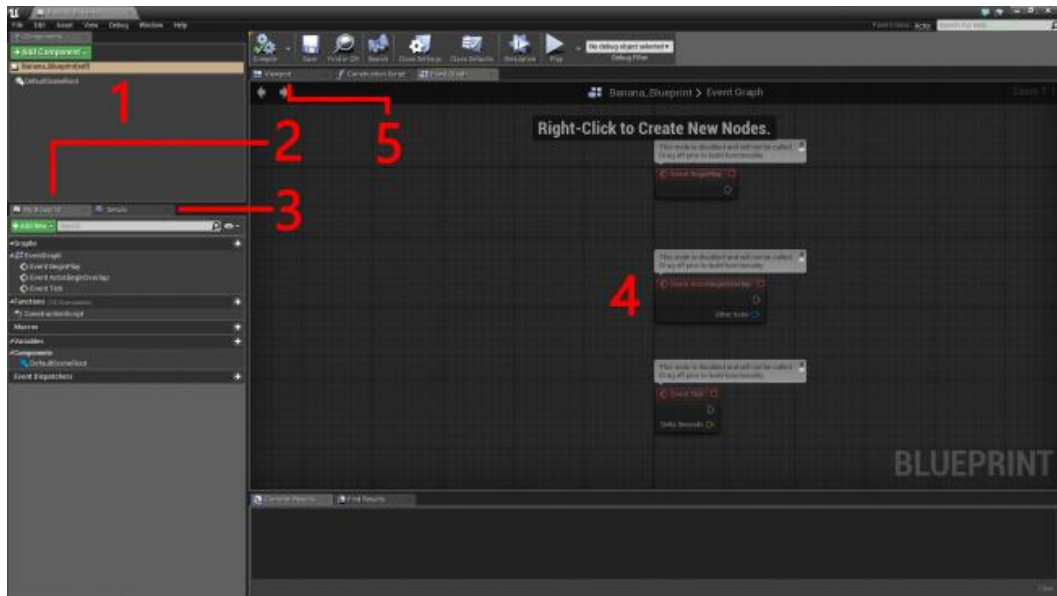
Дважды нажмите на *Banana\_Blueprint*, чтобы открыть его. Нажмите на *Open Full Blueprint Editor*, если появится подобное окно:



## Blueprint Editor

Во-первых, выберите в редакторе Blueprint editor вкладку Event Graph.

Blueprint editor состоит из четырёх основных панелей:



1. *Components*: содержит список текущих компонентов.
2. *My Blueprint*: этот раздел обычно используется для управления графами, функциями и переменными.
3. *Details*: здесь отображаются свойства текущего выбранного элемента.
4. *Graph*: именно здесь происходит магия. Все ноды и логика находятся здесь. Перемещаться по панели можно, зажав *правую клавишу мыши* и *перемещая* мышь. Масштабирование выполняется *прокруткой колёсика мыши*.
5. *Viewport*: здесь отображаются все компоненты, имеющие визуальные элементы. Перемещение и обзор выполняются так же, как и во Viewport основного редактора.

## Создание поворотного стола

Для создания стола нам нужно две вещи — основание и подставка. Их можно создать с помощью *компонентов*.

### Что такое «компоненты»?

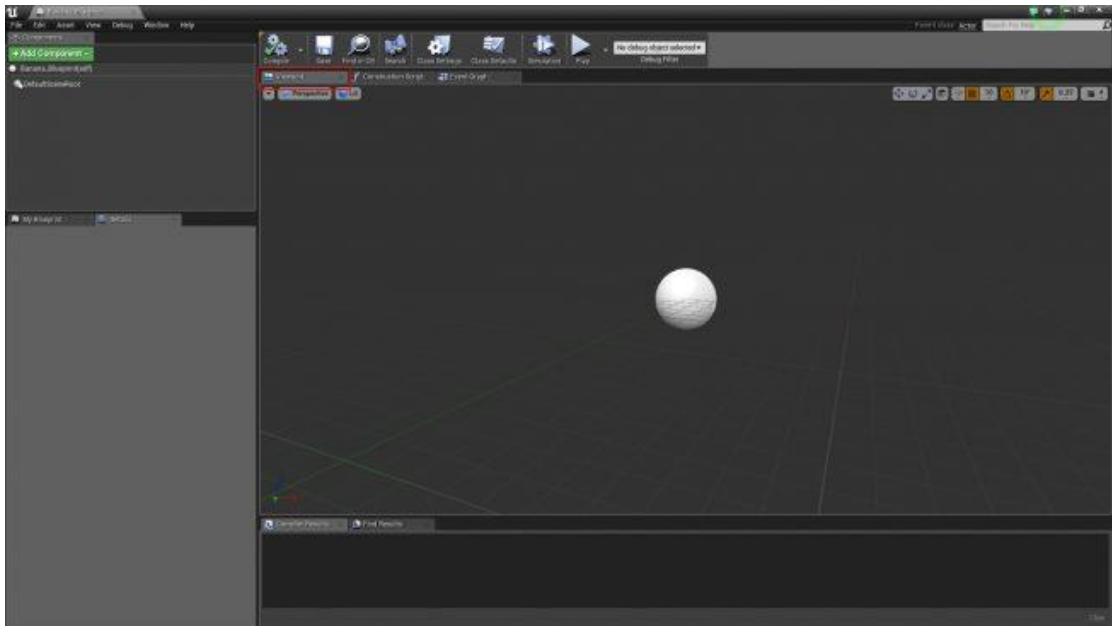
Если Blueprint — это автомобиль, то компоненты — это строительные элементы, из которых он состоит. Примерами компонентов могут быть двери, колёса и двигатель.

Однако компоненты могут быть не только физическими объектами.

Например, чтобы автомобиль мог двигаться, можно добавить компонент движения. Можно даже заставить машину летать, если добавить компонент полёта.

### Добавление компонентов

Чтобы увидеть компоненты, необходимо переключиться в режим Viewport. Нажмите на вкладку *Viewport*, чтобы переключиться на неё. Вот как это выглядит:

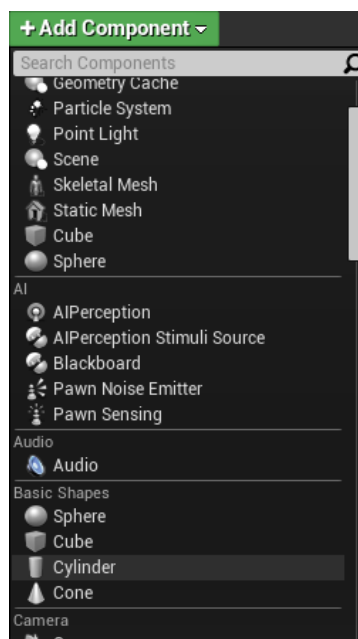


*Примечание:* компонент *DefaultSceneRoot* при запуске приложения не отображается, он виден только в редакторе.

Поворотный стол будет использовать два компонента:

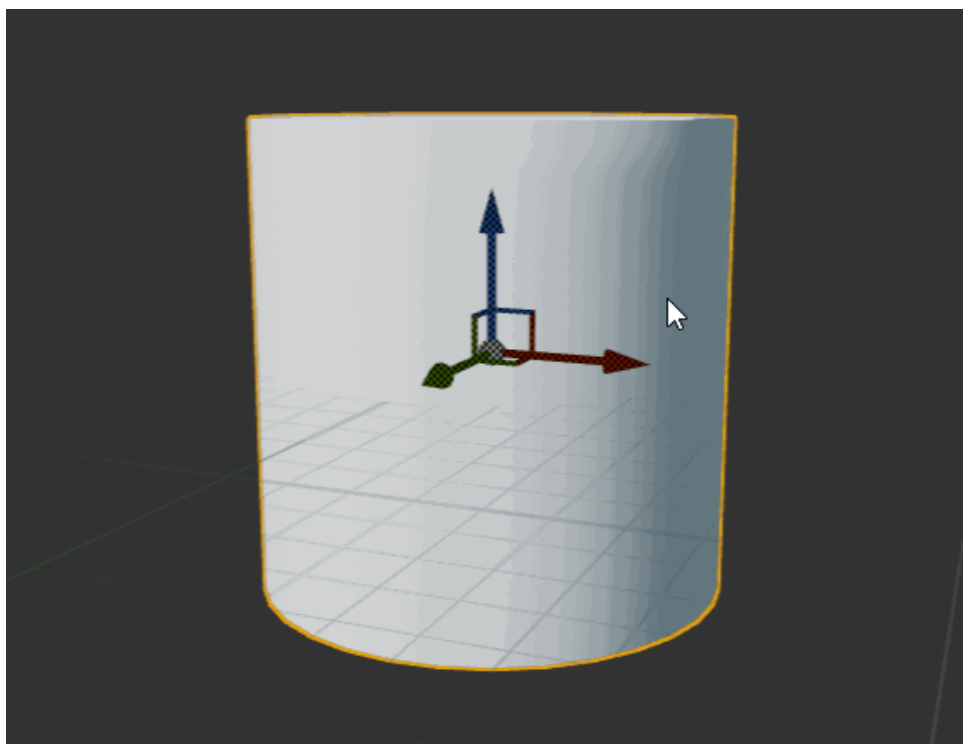
- *Cylinder*: простой белый цилиндр. Это будет основанием, на котором лежит банан.
- *Static Mesh*: — это компонент будет отображать меш банана.

Чтобы добавить основание, перейдите в панель Components. Нажмите на *Add Component* и выберите *Cylinder*.

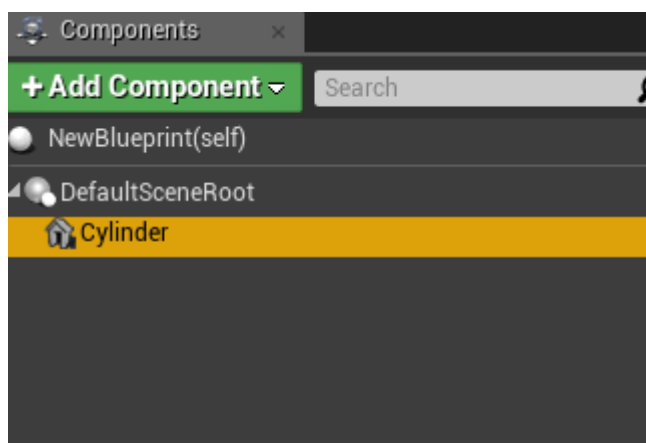




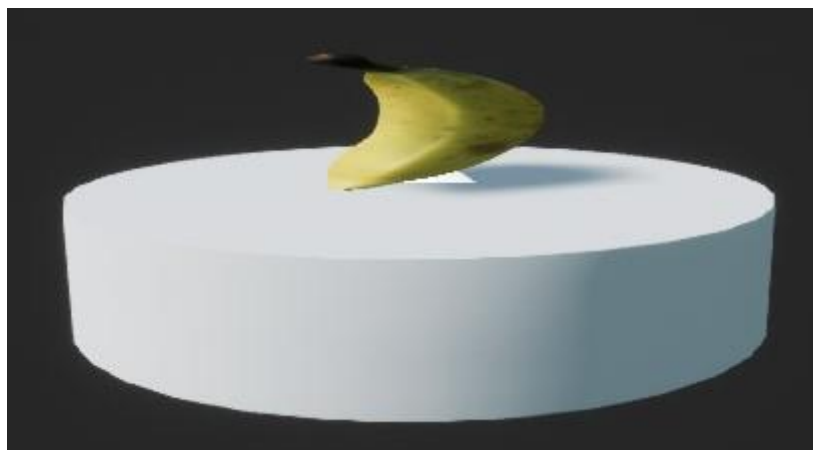
Неплохо было бы сделать основание чуть короче. Активируйте *манипулятор масштаба*, нажав *R*, а затем уменьшите масштаб (точный размер неважен, можно будет изменить его позже).



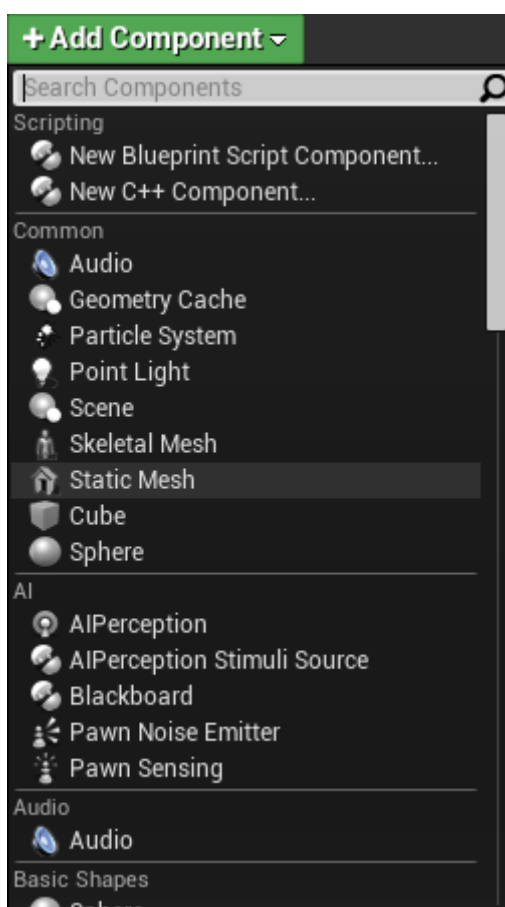
Теперь пора добавить меш. Вернитесь в панель компонентов Components и *нажмите* левой клавишей на пустой области, чтобы снять выделение с компонента *Cylinder*. Благодаря этому следующий добавляемый компонент не будет прикреплён к компоненту *Cylinder*.



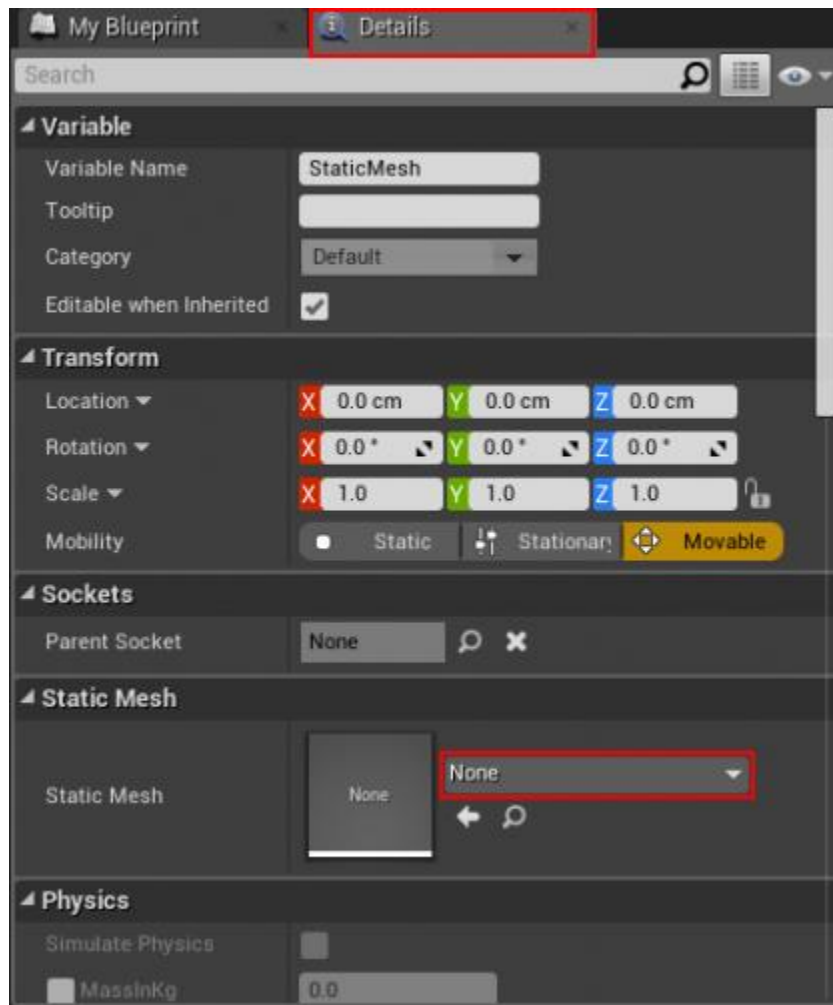
*Примечание:* если этого не сделать, то следующий компонент будет прикреплён к компоненту *Cylinder*. Это значит, что он также унаследует *масштаб* компонента *Cylinder*. Поскольку мы уменьшили масштаб цилиндра, следующий компонент тоже будет уменьшен.



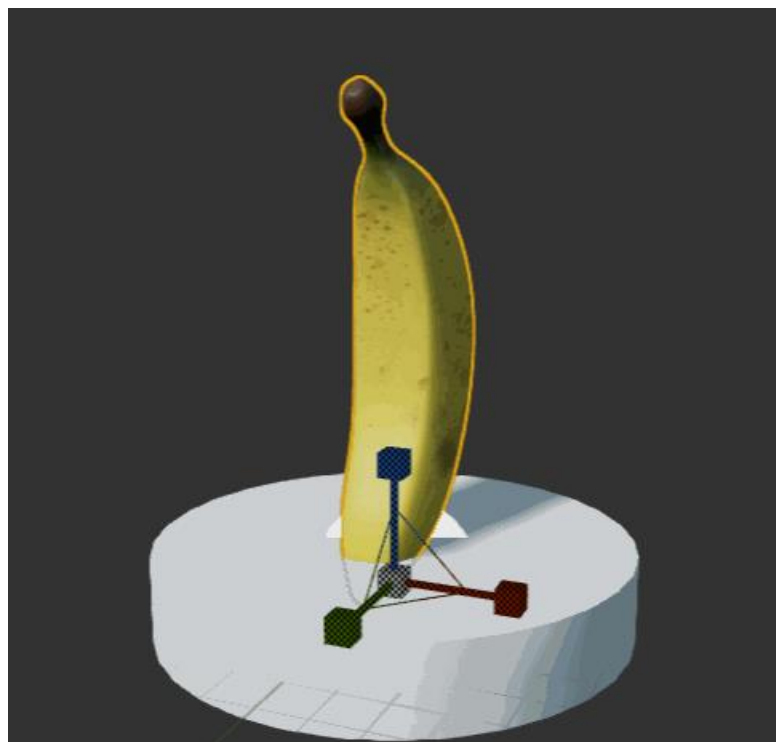
Затем нажмите на *Add Component* и выберите из списка *Static Mesh*.



Для отображения банана выберите компонент *Static Mesh*, а затем нажмите на вкладку *Details*. Нажмите на *раскрывающийся список* в правой части *Static Mesh* и выберите *Banana\_Model*.



Переместите банан, если он находится в неправильном положении. Для этого активируйте *манипулятор перемещения*, нажав *W*, а затем переместите его вверх.



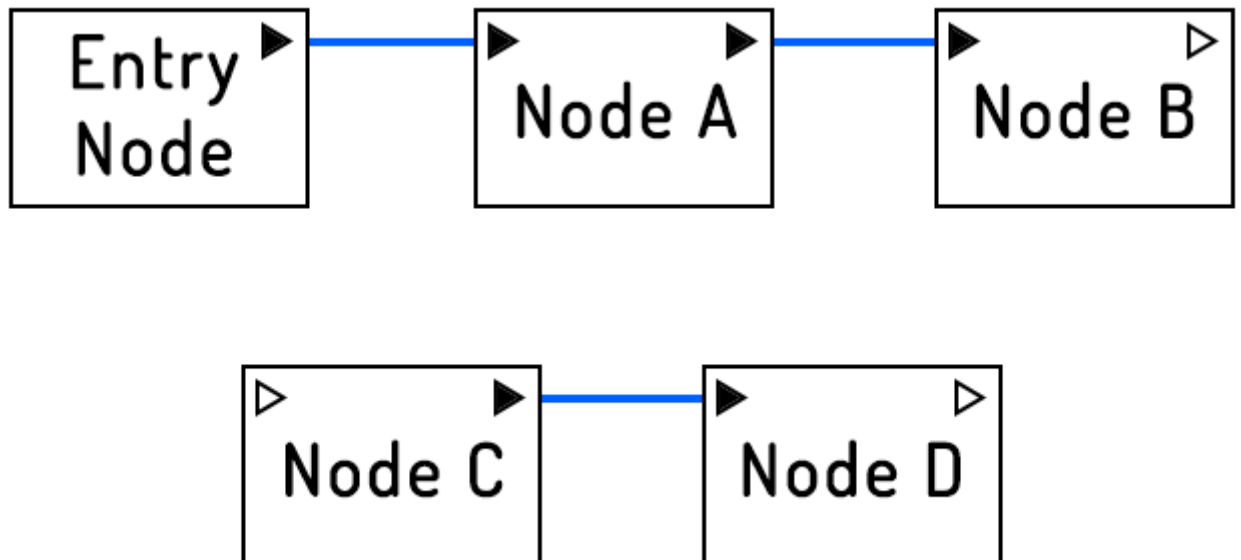
## Про ноды Blueprint

Теперь нужно сделать так, чтобы поворотный стол вращался. И здесь нам потребуются *ноды Blueprint*.

В отличие от своих близких родственников — нодов материалов — ноды Blueprint имеют особые контакты, называемые контактами *Execution*. Контакт слева — это вход, контакт справа — выход. У всех нодов есть хотя бы по одному входу и выходу.

Если нод имеет контакт входа, то его нужно подключить, чтобы он заработал. Если нод не подключен, все последующие ноды не будут выполняться.

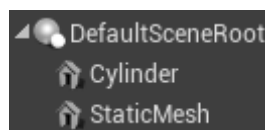
Вот пример:



*Node A* и *Node B* будут выполняться, потому что у их входных контактов есть подключение. *Node C* и *Node D* никогда не выполняются, потому что входной контакт *Node C* не имеет подключения.

## Вращение поворотного стола

Прежде чем начать, давайте посмотрим на панель Components. Можно заметить, что у *Cylinder* и *Static Mesh* есть отступ, а у *DefaultSceneRoot* — нет, потому что они подключены к *DefaultSceneRoot*.



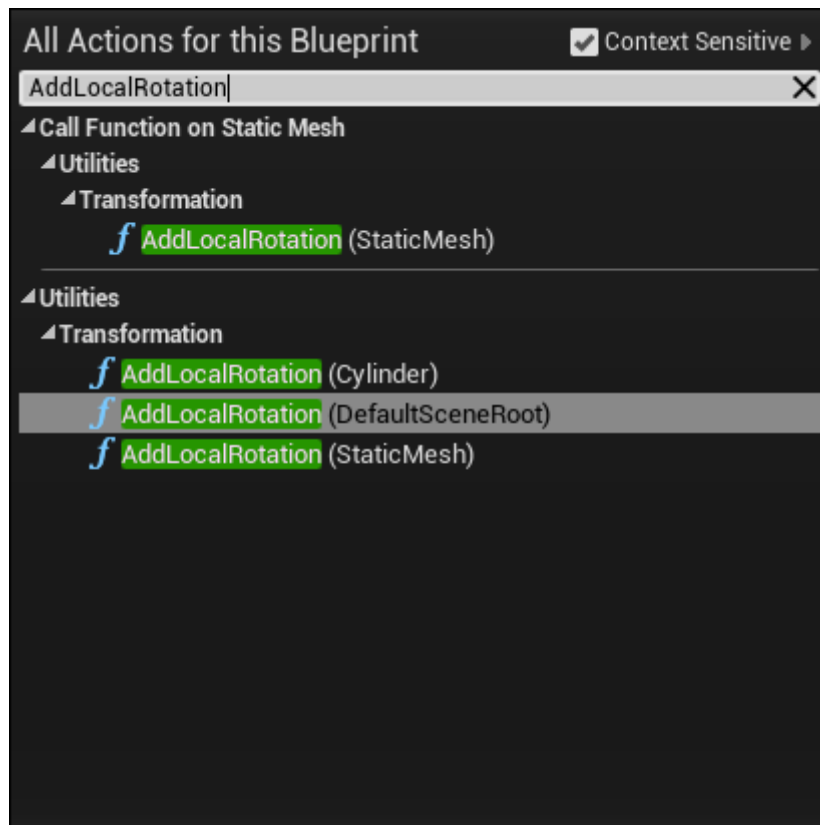
Если переместить, повернуть или отмасштабировать корневой компонент, то тоже самое произойдёт и с прикреплёнными к нему компонентами. Благодаря этому поведению можно поворачивать *Cylinder* и *Static Mesh* одновременно, а не по отдельности.

## Создание нода

Чтобы приступить к созданию скриптов, переключитесь назад на вкладку *Event Graph*.

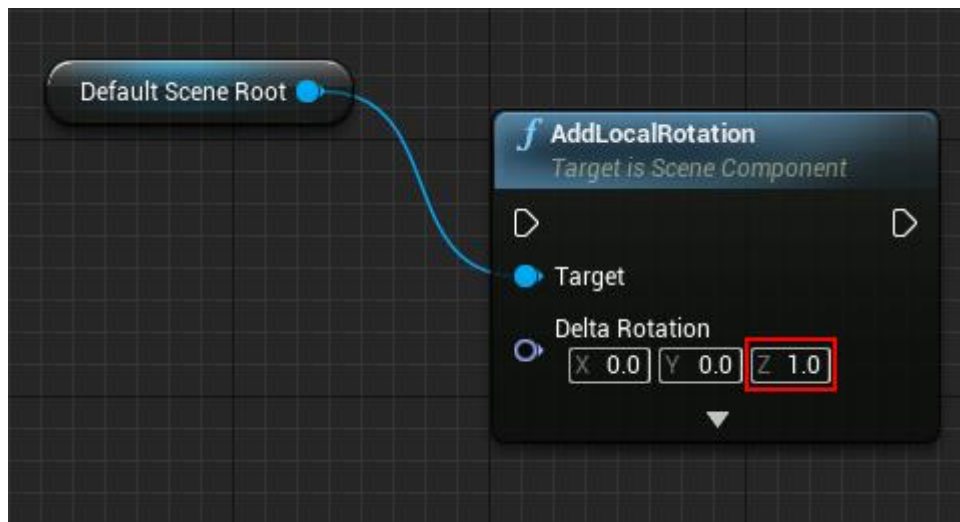
Реализация вращения объекта настолько проста, что требует всего одного нода. *Нажмите правой клавишей* на пустое пространство в графе, чтобы открыть меню доступных нодов. Найдите *AddLocalRotation*. Нам нужно поворачивать основание и банан, поэтому мы просто будем вращать корневой компонент. Выберите *AddLocalRotation (DefaultSceneRoot)*.

*Примечание:* если нода нет в списке, снимите флажок *Context Sensitive* в правом верхней части меню.



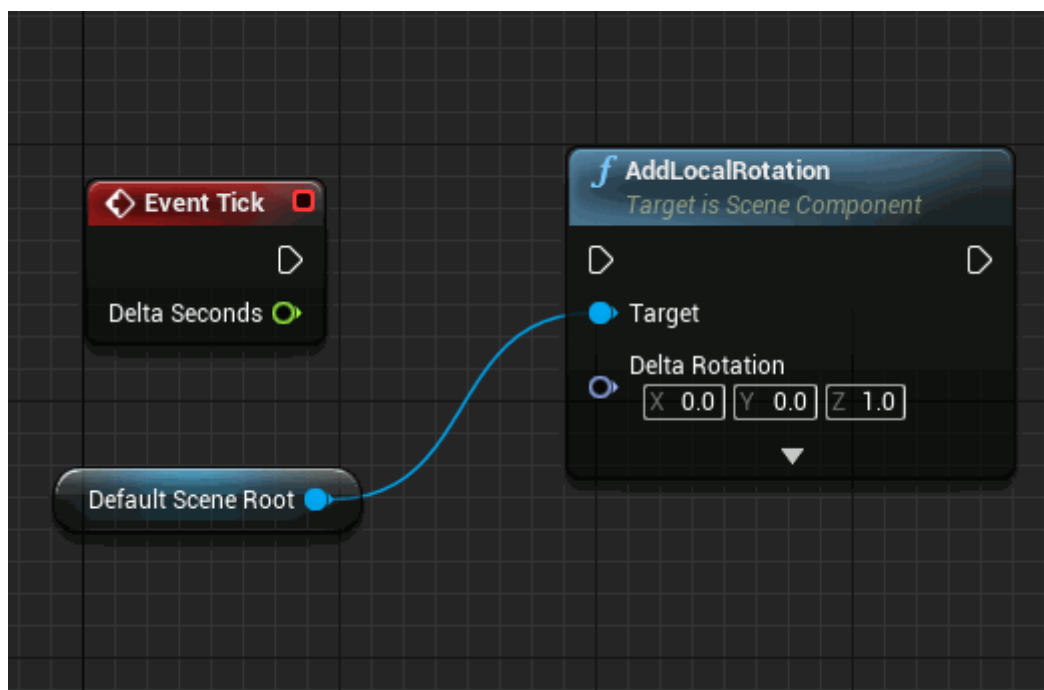
В вашем графе теперь появится новый нод *AddLocalRotation*. Вход *Target* автоматически подключится к выбранному компоненту.

Чтобы задать значение вращения, перейдите к входу *Delta Rotation* и измените значение *Z* на *1.0*. Благодаря этому Blueprint сможет выполнять вращение относительно оси *Z*. Чем выше значения, тем быстрее будет вращаться стол.



Чтобы поворотный стол вращался постоянно, нужно вызывать *AddLocalRotation* в каждом кадре. Для выполнения нода в каждом кадре воспользуемся нодом *Event Tick*. Он уже находится в графе. Если его нет, то создайте его тем же способом, что и ранее.

Перетащите выходной контакт нода *Event Tick* ко входному контакту нода *AddLocalRotation*.



*Примечание:* в этой реализации скорость вращения зависит от частоты кадров. Это значит, что поворотный стол на медленных компьютерах будет вращаться с меньшей скоростью, и наоборот. Для tutorials это нас вполне устраивает, потому что я не хочу ничего усложнять, но в будущем я покажу, как это исправить.

Наконец, перейдите в Toolbar и нажмите на *Compile*, чтобы обновить Blueprint, а затем закройте Blueprint editor.

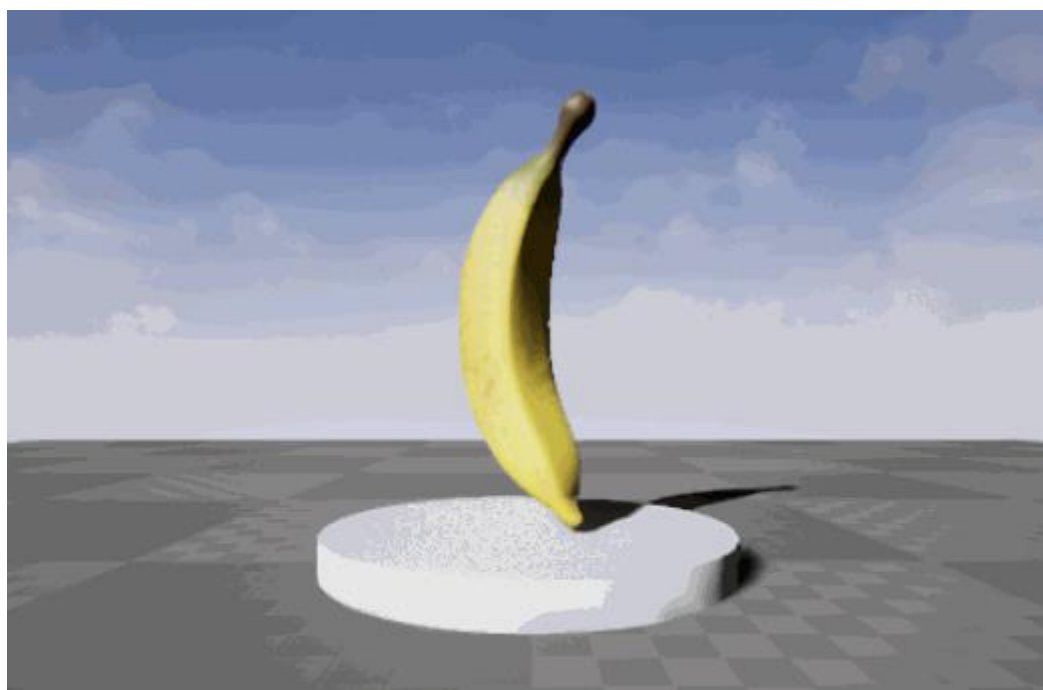


## Добавление Blueprints на уровень

Прежде чем добавлять Blueprint, вернитесь ко Viewport в основном редакторе и удалите модель банана. Для этого выберите модель, а затем выберите *Edit\Delete* или нажмите клавишу *Delete*.

Добавление Blueprint — это тот же процесс, что и добавление меша. Удерживайте *левую клавишу мыши* на файле и *перетащите* его во Viewport.

Перейдите к Toolbar и нажмите *Play*, чтобы увидеть результаты своих трудов!



*Примечание:* если вы не удалите исходную модель банана, то можете получить предупреждение о необходимости перестройки освещения. Если удалить модель, то ошибка больше не будет проявляться.