

### 3. Игровые объекты.

#### 3.1. Сцена.

Сцены содержат объекты вашей игры. Они могут использоваться для создания главного меню, отдельных уровней и для других целей. Можно считать каждый файл сцены отдельным игровым уровнем. В каждой сцене можно разместить объекты окружения, заграждения, декорации, по кусочкам создавая дизайн и саму игру.

Когда вы создаете новый проект Unity, в представлении сцены будет отображаться новая сцена. Это безымянная и несохраненная сцена. Сцена будет пустой, за исключением объектов по умолчанию — либо ортогональной камеры, либо перспективной камеры и направленного света, в зависимости от того, в каком режиме вы начали проект: 2D или 3D.

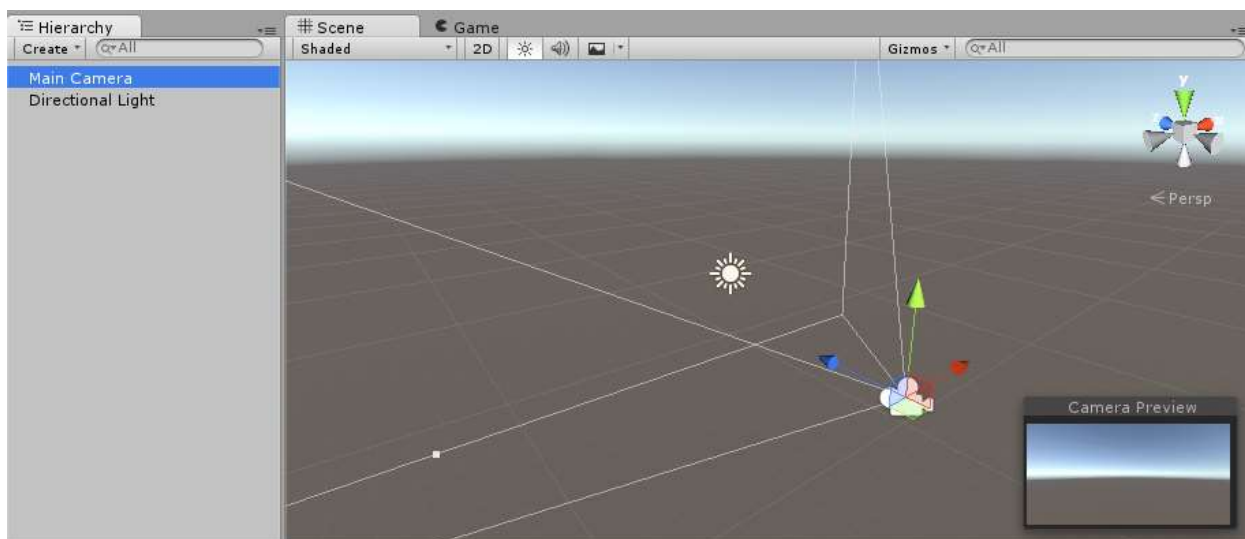


Рисунок 21 – Пример сцены с камерой и источником освещения

Чтобы сохранить сцену, над которой вы сейчас работаете, выберите в меню «Файл» > «Сохранить сцену» или нажмите Ctrl/Cmd + S. Сцены сохраняются как активы в папке Assets вашего проекта. Поэтому они появляются в окне проекта, как и любой другой актив.

#### 3.2. GameObjects.

Каждый объект в игре — это GameObject. Однако, GameObject'ы ничего не делают сами по себе. Они требуют специальной настройки, прежде чем стать персонажами, предметами окружения или специальными эффектами. Сами по себе они мало что делают, но действуют как контейнеры для Компонентов, реализующих реальную

функциональность. Например, объект Light создается путем присоединения компонента Light к GameObject.

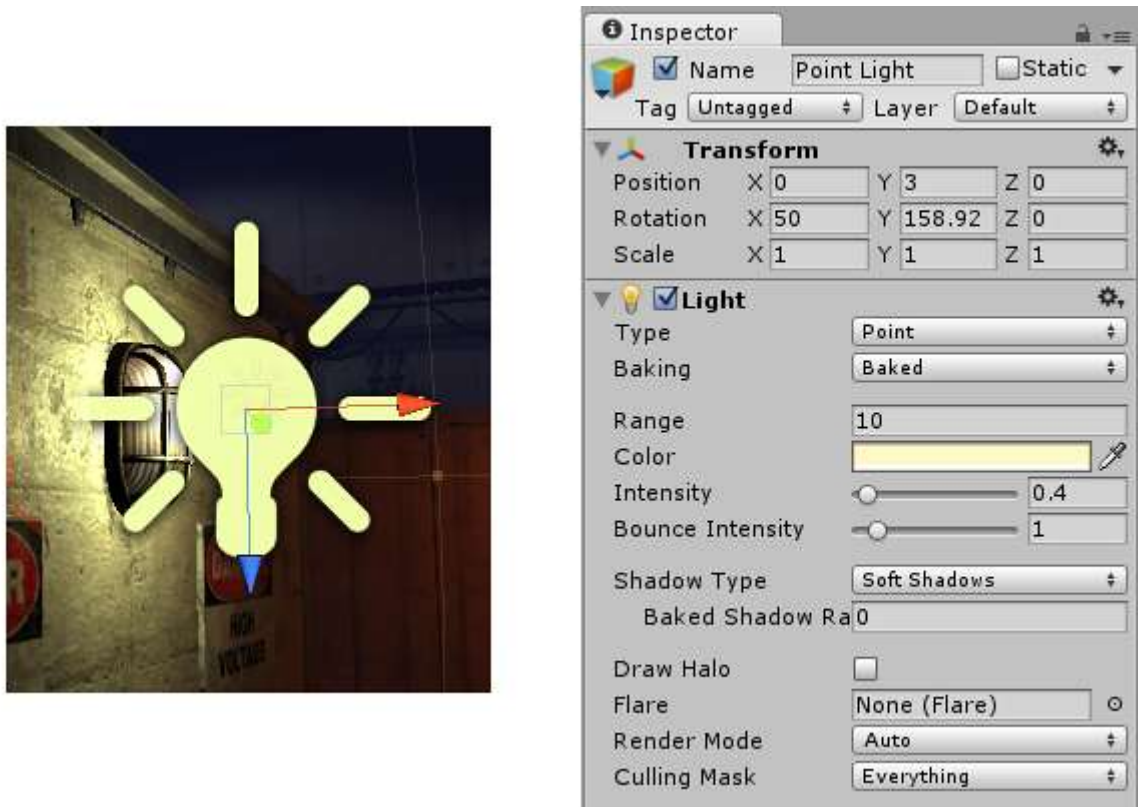


Рисунок 22 – Присоединение компонента Light к объекту лампы

### 3.3. Компоненты.

GameObject содержит компоненты. Мы исследуем эту взаимосвязь на примере GameObject'a и его наиболее распространенного компонента Transform.

В Unity невозможно создать GameObject без компонента Transform. Компонент Transform – один из самых важных компонентов, так как все свойства GameObject'a связанные с трансформациями используют этот компонент. Он определяет положение, вращение и масштаб GameObject'a в игровом мире/окне Scene. Если GameObject не будет иметь компонента Transform, он будет не более чем некоторой информацией в памяти компьютера. Он не сможет эффективно существовать в игровом мире. Компонент Transform также вводит концепцию, называемую наследование, которая используется редактором Unity и является критической частью работы с GameObject'ами.

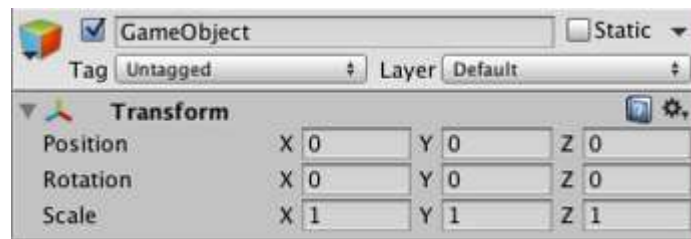


Рисунок 23 – GameObject с компонентом Transform

Вы можете добавить компоненты к выбранному игровому объекту через меню Components. Сейчас мы попробуем это, добавив Rigidbody на пустой игровой объект, который только что создали. Выделите его и выберите в меню Component->Physics->Rigidbody. После этого, вы увидите, что в инспекторе отобразился компонент Rigidbody и его свойства. Если нажать Play в то время как пустой игровой объект все еще выбран, вы можете получить небольшой сюрприз. Попробуйте, и обратите внимание, как Rigidbody добавил функциональность пустому игровому объекту (Y-компонент игрового объекта начинает уменьшаться, потому что физический движок в Unity заставляет игровой объект падать под действием силы тяжести).

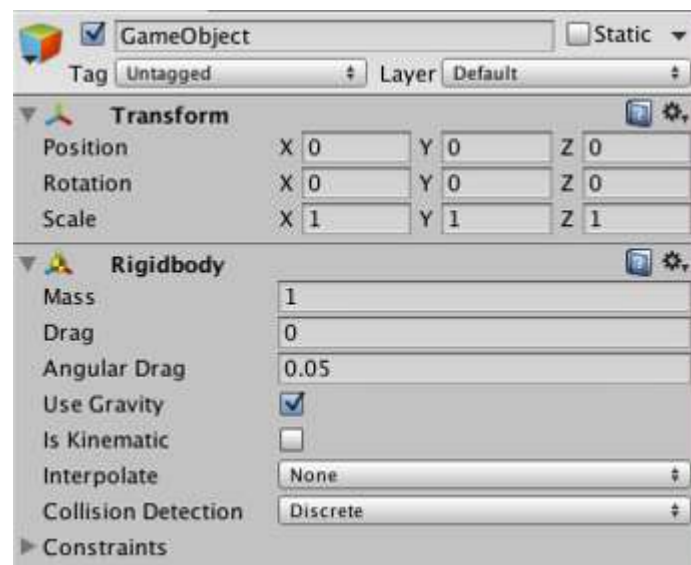


Рисунок 23 – GameObject с компонентами Transform и Rigidbody

Другой вариант заключается в использовании браузера компонентов (Component Browser), который может быть активирована с помощью кнопки Add Component в инспекторе объекта.

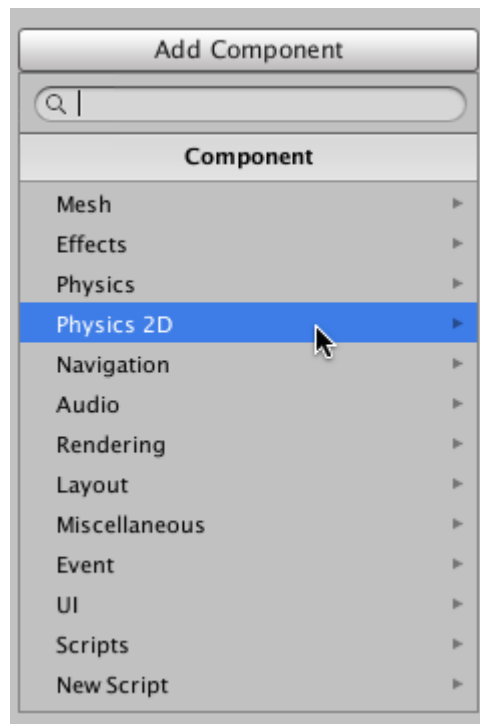


Рисунок 24 – Браузер компонентов

Браузер обеспечивает удобную навигацию по компонентам с помощью категорий, а также имеет окно поиска, которое можно использовать, чтобы найти компоненты по имени.

Вы можете прикрепить любое количество или комбинацию компонентов к одному игровому объекту. Некоторые компоненты работают лучше в сочетании с другими. Например, Rigidbody работает с любым коллайдером. Rigidbody контролирует Transform через физический движок NVIDIA PhysX, а коллайдер позволяет Rigidbody сталкиваться и взаимодействовать с другими коллайдерами.

Одной из замечательных особенностей компонентов является гибкость. При подключении компонента к игровому объекту, существуют различные значения или свойства (Properties) в компоненте, которые могут изменяться в редакторе при создании игры или через скрипты в запущенной игре. Есть два основных типа свойств: значения (Values) и ссылки (References). Взгляните на изображение ниже. Это пустой Игровой Объект с компонентом Audio Source. Все параметры компонента Audio Source в Инспекторе выставлены по умолчанию.

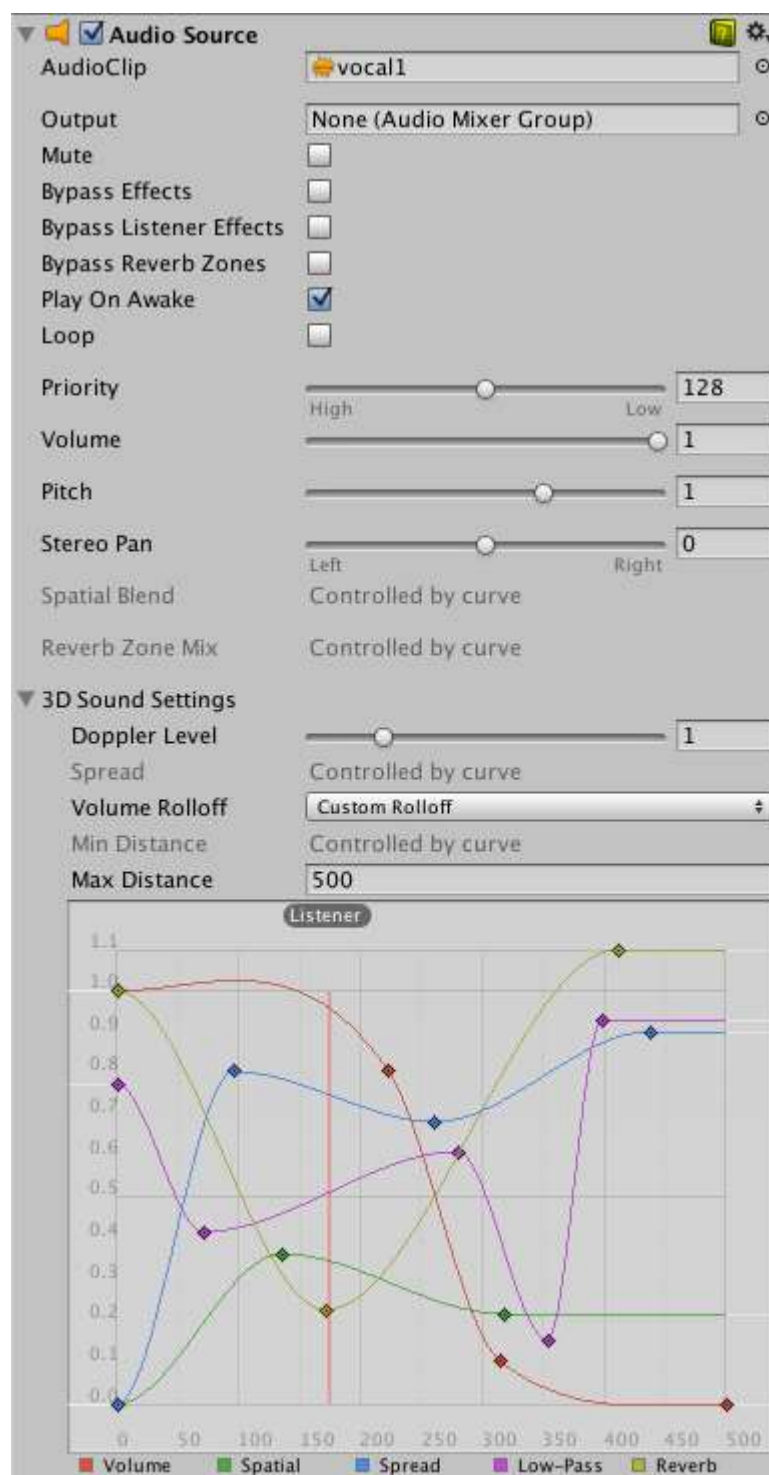


Рисунок 25 – Параметры компонента Audio Source

Компонент содержит одно свойство-ссылку и семь свойств-значений. Audio Clip - это свойство-ссылка. Когда этот аудио источник начинает играть, он будет пытаться проиграть файл, на который ссылается свойство Audio Clip. Если такой ссылки не окажется, то возникнет ошибка, так как никакое аудио не будет проиграно.

Компоненты могут включать ссылки на любые другие типы компонентов, игровых объектов или ассетов.

### 3.4. Теги.

Тег — это слово, которое вы связываете с одним или несколькими игровыми объектами. Например, вы можете определить теги «Игрок» и «Враг» для персонажей, контролируемых игроком, и неигровых персонажей соответственно; Тег «Собираемый» может быть определен для предметов, которые игрок может собирать в Сцене; и так далее. Очевидно, что теги предназначены для идентификации игровых объектов в целях написания сценариев. Мы можем использовать их для написания кода сценария для поиска GameObject путем поиска любого объекта, содержащего нужный нам тег. Это достигается с помощью функции `GameObject.FindWithTag()` используемой в скриптах.

Для добавления тега используйте окно инспектора, в котором есть раскрывающееся меню «Тег» и «Слой-> Слои» сразу под именем любого игрового объекта. Чтобы применить тег к игровому объекту, просто откройте раскрывающийся список «Теги» и выберите нужный тег:

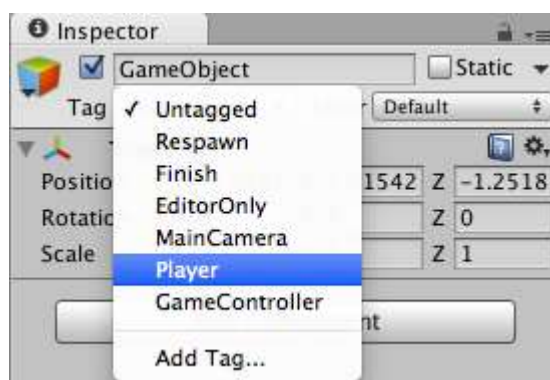


Рисунок 25 – Добавление тега из инспектора

Чтобы создать новый тег, нажмите «Добавить тег...» в конце раскрывающегося меню. Это откроет Диспетчер тегов в Инспекторе.

### 3.5. Трансформации.

Как уже было отмечено компонент Transform (трансформация) используется для хранения значений позиции, вращения, размеров и состояния наследования GameObject'а, потому он очень важен. К GameObject'у всегда добавлен компонент Transform - его невозможно удалить или создать GameObject без него.

Компоненты Transform управляются в 3D пространстве по осям X, Y, и Z, или в 2D пространстве просто по X и Y. В Unity эти оси представлены красным, зелёным и синим цветами соответственно.

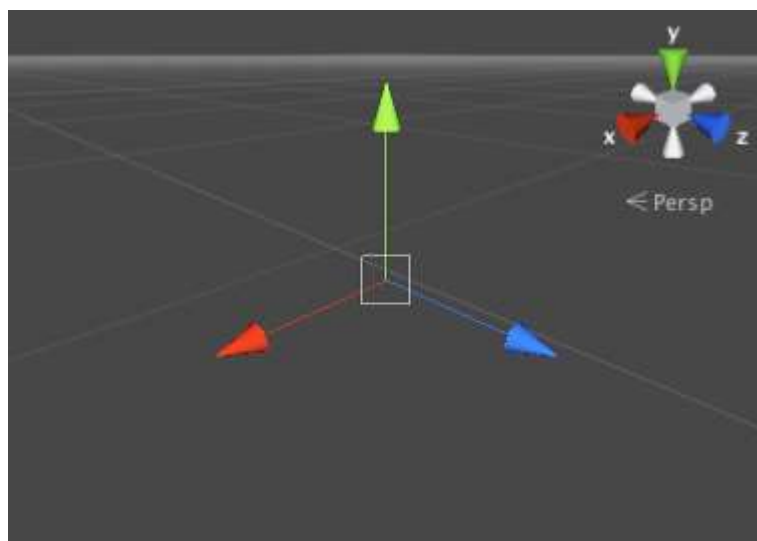


Рисунок 26 – Компонент Transform с цветными осями

Transform может быть изменён в окне Scene или путём изменения параметров в инспекторе. В сцене вы можете изменять Transform используя инструменты Translate, Rotate и Scale (двигать, вращать и масштабировать). Эти инструменты расположены в верхнем левом углу редактора Unity. Эти инструменты можно применить к любому объекту в сцене. Когда вы кликнете на объект, вы увидите, что у объекта появится гизмо инструмента. Вид гизмо зависит от выбранного инструмента.

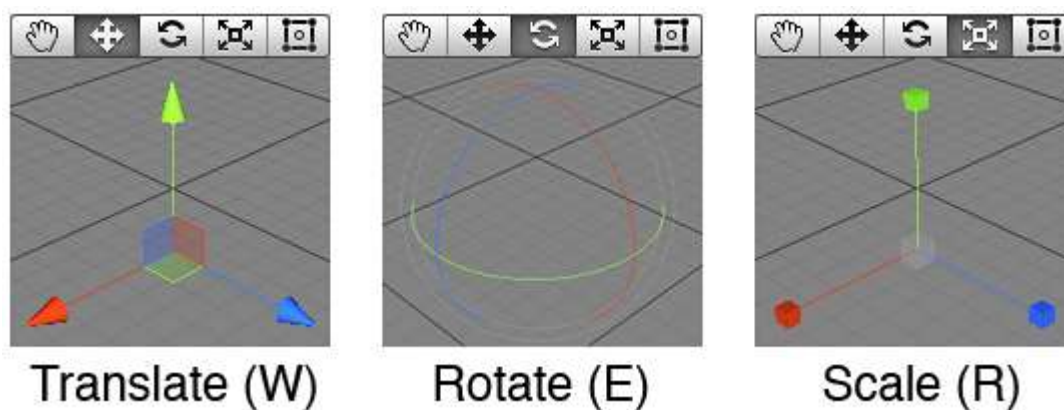


Рисунок 27 – Гизмо компонента Transform

Когда вы нажмёте на одной из трёх осей гизмо и потянете, вы заметите, что её цвет изменится на жёлтый. По мере движения мыши, вы увидите, как объект будет двигаться, вращаться или менять размер соответственно выбранной оси. Когда вы отпускаете кнопку мыши, ось остаётся выделенной. Если вы впоследствии будете двигать мышью с зажатым колёсиком, то будет использована последняя выбранная ось, независимо от позиции курсора мыши.

Для режима перемещения есть дополнительная опция - перемещение объекта в отдельной плоскости (другими словами, позволяет перемещать объект в двух осях сразу, не затрагивая третью). Три маленьких цветных квадрата вокруг центра гизмо перемещения активируют фиксацию для каждой из плоскостей; цвета соответствуют оси, которая будет зафиксирована, если нажать на квадрате (например, синий квадрат фиксирует ось Z).