**2. ABaseWeapon class**

1. Сначала надо перенести контент из проекта исходников (какой контент)?

2. Создаем базовый класс оружия, от кого наследуем?

3. Что убираем из автоматически сгенерированного кода?

4. Добавляем проперти для внешнего вида и создаем его в конструкторе (все как всегда), что есть у этого меша свеху по сравнению с обычным?

5. Теперь создаем блюпринт оружия, далее надо добавить специальную штуку в нашем блюпринте, к которой мы потом будем присоединять наше оружие – где это находится и как это сделать?

6. Смотрим в редакторе предварительный вид – как (в анимациях, просто в меше)?

7. Если запустим игру – что-то будет в руках у нас? Почему? Какую функцию вводим и проперти у нашего персонажа? Где будем функцию вызывать? Функционал самой функции: что создали? С помощью какой функции присоединить нашего актора к компоненту? Что принимает? Как получить объект для первого параметра? Подробно описать второй аргумент – его возможные конструкторы (два, которые мы рассмотрели) и что они принимают? Какие есть варианты перечисления и какой мы используем для конструктора второго параметра?

8. Что осталось настроить в блюпринте?

1. Сначала мы перенесли контент для винтовки (Rifle) и ракетницы (Launcher) – это два SkeletalMesh.

2. Создаем базовый класс оружия, наследуясь от Actor. Назовем STUBaseWeapon и положим в новую папку Weapon (не забываем внести исправления в файл Build.cs).

3. В файлах удаляем функцию Tick(), так как она нам не понадобится, а также ставим в конструкторе false.

4. Начнем с внешнего вида нашего оружия – добавим проперти скелетал меша и все как всегда:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

У скелетного меша, как и у обычного, имеются полигоны (он имеет некий внешний вид), иерархия костей, с помощью которых можно анимировать данный меш.

5. Теперь надо создать блюпринт и установить модель оружия.

Следующая наша задача – заспавнить данное оружие в мире и присоединить к нашему персонажу таким образом, чтобы трансформация оружия высчитывалась относительно персонажа.

Для начала открываем блюпринт с нашим персонажем и через SkeletalMesh-компонент кликаем два раза на меше и попадаем в редактор персонажа.

Слева у нас панель Skeleton Tree, где представлены все кости персонажа.

В Details Panel справа можно увидеть относительные трансформации костей.

У скелета персонажа также имеются виртуальные кости – они никак не представлены графически, однако у них имеются трансформации.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Одну из таких мы и выбираем через ПКМ выбираем Add Socket. Сокеты – это специальные объекты в скелетал меше, мы можем добавить сокет на любую кость и, по сути, данный объект представляет трансформацию.

В Details мы можем видеть название сокета, имя родительской кости и трансформацию.

Изображение выглядит как текст

Автоматически созданное описание

Сокеты в скелете используются по-разному. Данный мы будем использовать для присоединения нашего оружия.

В дальнейшем в нашем оружии мы будем использовать сокет как точку выстрела – то есть это некоторая именованная трансформация в скелетал меше, которую можно изъять в коде или блюринтах.

6. В редакторе можно посмотреть, как будет выглядеть приаттаченный меш к данному сокету через ПКМ и пункт Add Preview Asset и выбрать нам нужный.

Выберем винтовку.

В редакторе появился ее меш. Трансформация сокета единичная и винтовка ориентирована неправильно.

Изображение выглядит как автомат

Автоматически созданное описание

Можем это исправить здесь через трансформацию. Preview asset сохраняется так же и на других вкладках, поэтому можем перейти в анимацию и настроить трансформацию там.

Изображение выглядит как внешний, мужчина, человек, автомат

Автоматически созданное описание

Изменим так, чтобы она была ориентирована правильно:

Изображение выглядит как внешний, автомат

Автоматически созданное описание

Если мы откроем другие анимации, то увидим, что трансформация сокета корректная и оружие зафиксировано в руках нашего персонажа (перемещается соответственно).

7. Однако, если мы запустим игру – никакого оружия мы не увидим, так как это только Preview. Никакого спауна мы еще не делали – этим и займемся.

В классе персонажа создадим функцию SpawnWeapon() – в ней будет происходит спаун оружия и присоединение к сокету нашего персонажа:



Нам потребуется проперти с классом оружия, которое мы хотим заспавнить:

Изображение выглядит как текст, оранжевый, темный

Автоматически созданное описание

Спавнить оружие будем в BeginPlay:

Изображение выглядит как текст

Автоматически созданное описание

В самой функции нам надо создать актора и присоединение к мешу нашего персонажа:

Изображение выглядит как текст, снимок экрана, экран

Автоматически созданное описание

Присоединение делается с помощью функции AttachToComponent().

Ее первый аргумент – указатель на компонент, к которому мы хотим присоединить данного актора. Получаем сюда меш персонажа через функцию GetMesh().

Второй – AttachmentRules – правило присоединения. Она имеет тип FAttachmentTransformRules и имеет несколько конструкторов.

Изображение выглядит как текст

Автоматически созданное описание

Конструкторы принимают enum EAttachmentRule. В первом конструкторе этот параметр применяется ко всем трансформациям.

Во втором – можем указать разные для каждого компонента трансформации.

Есть еще дополнительный параметр bWeldSimulateBodies – он относится к физике и отвечает за то, как будет происходить физическая симуляция когда два тела приаттачены к друг другу. Он нас не интересует.

Изображение выглядит как текст

Автоматически созданное описание

Переходим в определение EAttachmentRule и видим, что это enum и у него есть три элемента. Каждый из них отвечает за то, как будет высчитываться финальная матрица трансформаций.

Мы будем использовать SnapToTarget – в данном случае актор наследует трансформацию компонента к которому мы его присоединяем.

Третий, опциональный – название сокета. Мы его назвали WeaponSocket.

8. Осталось выбрать в блюпринте нашего персонажа выбрать класс оружия, который мы хотим создать (которым мы и создали).