**13. Launcher #1, rocket spawn**

1. Собираемся спавнить ракету – что нам надо добавить для лаунчера? Как это сделали?

2. Нужно создать для самой ракеты что-то – что? Как назвали, от кого наследуется? Как почистили сгенерированный код? Какой добавили компонент и как его настроили? Что мы будем понимать с помощью данного компонента?

3. Определяем функции у лаунчера (две) – какие? Какое добавляем проперти для нашей логики? Какой ЗФ нужен нам для спавна ракеты? Какая функция? Что принимает? Как закончить создание ракеты?

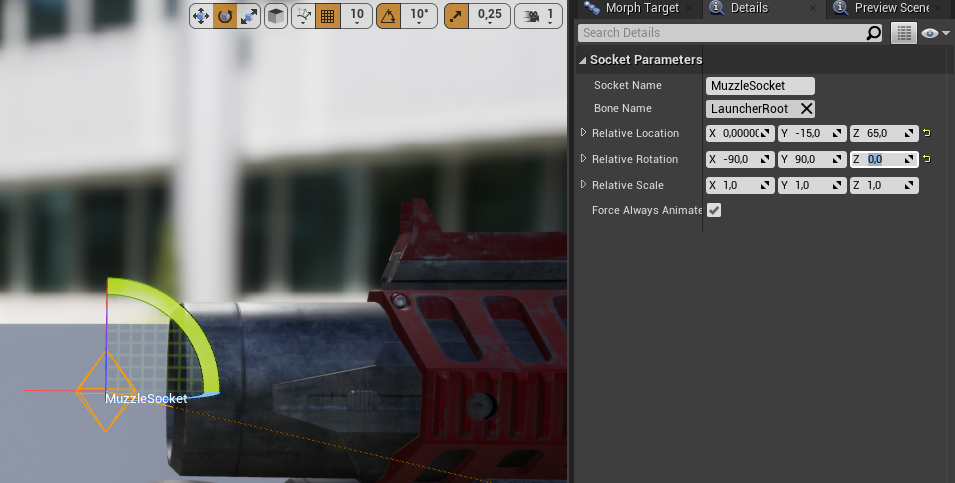
4. Создаем BP-версии и настраиваем кое-что у нашей ракеты – что?

5. Осталось замутить цвет для нашей сферы – как это сделали?

1. Для начала добавляем сокет для места, откуда будет вылетать ракета.

Имя у него будет такое же, поэтому нам не надо будет переопределять его в классе лаунчера.

Устанавливаем ему соответствующую трансформацию.



2. Нам надо создать дополнительный актор, так как лаунчер будет стрелять ракетами, поэтому создаем класс STUProjectile в папке Weapon, наследующийся от Actor.

Убираем ему автообновления.

Для данного класса добавим компонент сферической коллизии – USphereComponent (мы работали с коллизией капсулы, теперь – с коллизией сферы) и создадим его (все как всегда, только добавляется функция InitSphereRadius – устанавливает радиус нашей коллизии):

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

С помощью данной коллизии мы будем понимать, попали мы куда-то или нет. Внешний вид задекорируем в блюпринтах.

3. Теперь динамически при выстреле будем создавать данный класс Projectile в классе лаунчера.

Для начала переопределим поведение функций StartFire и MakeShot и создадим проперти с ракетой, которую мы хотим заспавнить:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

В функции StartFire просто вызываем функцию MakeShot.

Для спавна нашей ракеты нам понадобится класс GameplayStatics, поэтому подключим его ЗФ в cpp.

Для начала нам надо создать начальную трансформацию нашей ракеты.

Далее нам нужен сам актор ракеты, мы его создаем с помощью функции BeginDefferedActorSpawnClass. Она позволяет отложить вызов функции BeginPlay, что позволит нам настроить параметры в данном классе.

Принимает следующие аргументы:

а. Указатель на мир.

б. Класс, который хотим заспавнить.

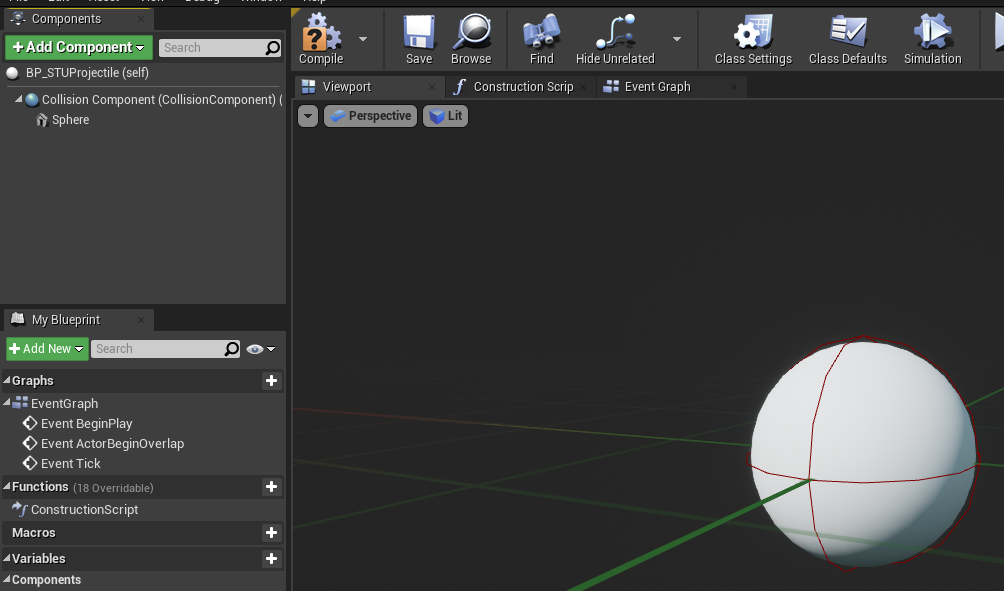
в. Трансформацию.

Остальные по умолчанию.

Далее нам надо будет передать некоторые параметры, которые уже в следующих уроках.

Завершаем спаун с помощью FinishSpawningActor.

4. Разумеется, надо создать BP-версию класса нашей ракеты и выбрать ее у лаунчера. При этом в BP-версии ракеты надо создать меш сферы и смасштабировать до размера коллизии.



5. Изменим цвет сферы. В корневой папке создали новую папку Materials и добавили туда M\_BaseColor материал. Через 4 и ПКМ добавляем 4-мерный вектор для задания цвета. Конвертируем в параметр. Называем Color.

Далее создаем MaterialInstance от него и называем M\_RedColor. Определяем красный цвет для данного материала и добавляем нашей сфере в BP-Projectile.