**5. Decals**

1. Что такое Decal и где можно его взять в UE (базовый, а не создать)?

2. Как создать Decal на основе изображения – в два этапа (создание \_, а затем его настройка – получим Decal)? Что можно создавать с помощью Decal, что будем создавать мы?

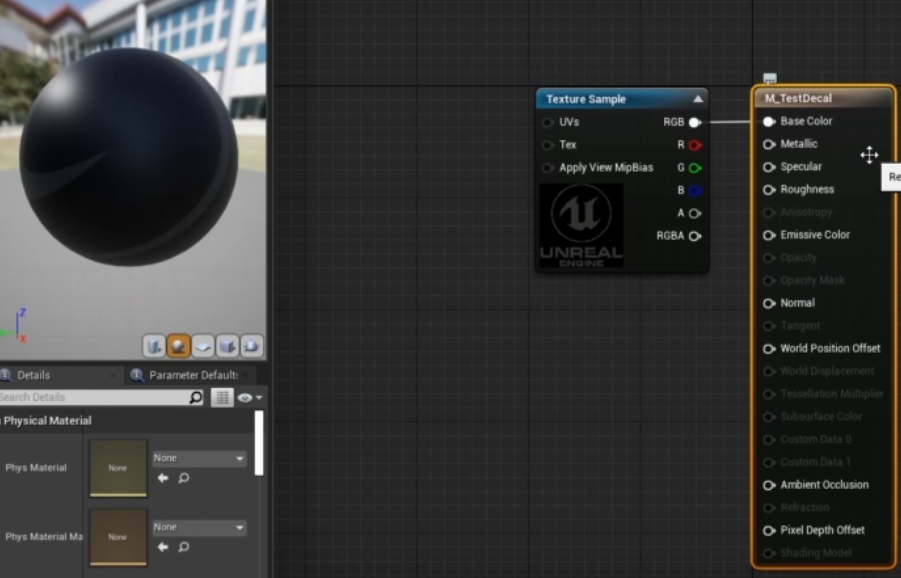
3. Когда будет происходить спаун декаля? Какие добавляем две структуры?

4. Как теперь будем в WeaponFXComponent реализовывать функцию PlayImpactFX и как мы изменили уже имеющиеся проперти (два)?.

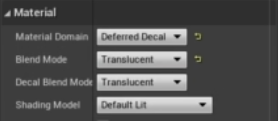
5. Как теперь настраиваем наши Projectile и Rifle? Какая осталась проблема, связанная с декалями? Как пофиксили?

1. В окне Place Actors (откуда берем всю геометрию и т.д.) в категории Visual Effects есть ассет Decal. Decal – это специальный вид материалов, который проецируется на меши (как StaticMesh, так и SkeletalMesh).

2. Чтобы создать Decal на основе изображения, кликаем ПКМ на изображение и затем Create Material. Текстура нашего изображения автоматически подставляется в Base Color.



Для того чтобы использовать наш материал как Decal, нам необходимо выделить основную ноду материала и выбрать в Material Domain значение Deferred Decal, а в Blend Mode – Translucent.



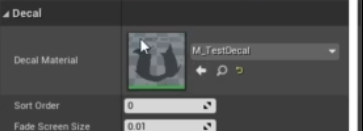
Теперь мы можем присвоить наш Decal Material ранее добавленному на сцену Decal’у, и у нас будет отображаться наш материал.

Изображение выглядит как спорт, с плиткой

Автоматически созданное описание

Изображение выглядит как текст, земля

Автоматически созданное описание



С помощью данной техники можно делать различные граффити, создавать эффект следов персонажа. Мы будем использовать ее для создания следов от пуль.

В настройках так же можно изменить размер декаля, тем самым изменив объем, в котором он будет проецироваться на меши.

Добавим специальный материал M\_Impact\_Decal для отображения следа от пули.

3. Спаун декаля будет происходить в тот же самый момент, что и спаун импакт эффекта.

Создадим дополнительные структуры в STUCoreTypes.h:

Изображение выглядит как текст

Автоматически созданное описание

В первой структуре содержатся необходимые настройки для текущей декали.

Первое поле – материал декали UMaterialInterface.

Второе – размер декали.

Третье – время жизни декаля.

Четвертый – будет отвечать за время анимации альфа-компонента декаля (за данный промежуток времени декаль будет растворяться).

Вторая структура объединит ниагара-эффект и декаль.

4. Переходим в WeaponFXComponent. Теперь у нас будет не отдельная ниагара-система, а новый объект структуры, а так же нам надо будет поменять и нашу словарь материалов и импактов:

Изображение выглядит как текст

Автоматически созданное описание

В cpp-файле подключаем два ЗФ и переопределяем поведение нашей функции:

Изображение выглядит как текст

Автоматически созданное описание

Стало:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Для спауна декаля воспользовались функцией из GameplayStatics под названием SpawnDecalAtLocation. Четвертый параметр – место, где хотим создать декаль, пятый параметр – вращение декаля.

Функция возвращает указатель на DecalComponent, нам необходим данный указатель, чтобы установить параметры жизни декаля через функцию SetFadeOut. Первый параметр отвечает за то, через сколько запустится анимация выцветания, а второй – сколько она будет длиться. Есть еще третья переменная, отвечающая за удаления Owner’а декаля. Но в данном случае нам это не нужно, так как данная переменная имеет смысл, только если мы использовали SpawnDecalAttached, а не SpawnDecalAtLocation.

5. Теперь нужно настроить все визуальные ассеты для Rifle и Projectile (наши словари). Материал для Decal будем использовать везде один и тот же, остальное настраиваем так же.

Осталась одна проблема – декали действительно исчезают через пять секунд, но мгновенно, без анимации альфа-канала. Чтобы это исправить, в нашем материале декаля добавляем специальный параметр, в котором хранится значение переменной прозрачности декаля – Decal Lifetime Opacity. В тот момент, когда декаль начинает анимацию выцветания, данный параметр начинает изменяться от 1 до 0. Мы можем домножить данный параметр на значение альфа-канала нашей текстуры и подать итоговое значение в ноду Opacity:

