15. CAMERA INSIDE CHARACTER FIX

1. Какая проблема? Какой сделаем фикс (добавляем проперти, определяем функцию, что в последней делаем – подсказка: два делегата)?

2. Определяем бинд-функции и вспомогательную для них. Как во второй получаем инфу о пересечении?

3. Однако что у нас все равно не пропадает? Как это пофиксили?

1. Проблема – встаем у какого-то объекта и за счет коллизии SpringArm-компонента камера въезжает внутрь персонажа.

Сделаем следующий фикс – добавим сферическую коллизию камере, и как только она будет пересекаться с коллизией капсулы, будем делать меш персонажа невидимым:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Для Overlap-событий существует два делегата – OnComponentBeginOverlap – когда одна коллизия начинает пересекать другую – второй – противоположный.

2. Для наших bind-функций копируем сигнатуры делегатов – для первого это 6 параметров, для второго – 4.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Для них добавили вспомогательную функцию, которая будет проверять, пересекает сферическая коллизия капсулу или нет.

В ней ввели булеву переменную, которая равна true, если компоненты пересекаются, что определяем с помощью функции PrimitimeComponent IsOverlapingComponent, в качестве параметра она принимает другой компонент – мы передаем туда капсулу нашего персонажа (не забываем подключить ЗФ для CapsuleComponent).

А далее в зависимости от нее устанавливаем видимость нашего меша.

3. Однако дочерний меш оружия у нас не пропадает и мы его все равно видим.

Чтобы это подправить, мы можем получить массив всех дочерних компонентов меша с помощью функции GetChildrenComponents (если первый параметр true, то возвращаются потомки со всех уровней, false – только первого уровня, второй параметр – массив по ссылке) и установить видимость в соответствии с HideMesh.

Изображение выглядит как текст

Автоматически созданное описание

После этого нам необходимо пройтись по всем компонентам. Мы получили массив SceneComponent (взяли из сигнатуры функции тип массива) – это один из самых базовых классов в иерархии компонентов и у него не существует функции SetOwnerNoSee, она появляется только начиная с PrimitiveComponent, поэтому кастим к нему.