**7. ASpectatorPawn**

1. Что происходит на данном этапе после смерти? Специальный класс наблюдатель в UE, чем является

2. Осуществим переключение на данный класс после смерти – какой ЗФ, где вызовем новую функцию, что принимает

3. Попадем в функцию смены состояния, что там происходит, провалимся в функцию включения именно наблюдателя – что там происходит

1. В последнем уроке после смерти персонажа он падал на 5 секунд и пропадал, а мы не могли перемешать камеру. Контроллер остается без персонажа, то есть он больше не владеет никаким пауном.

В архитектуре движка существует специальный класс SpectatorClass.

Фактически, это просто паун с камерой с инпутом на клавиши и мышку.

2. Сделаем так, чтобы, когда наш персонаж умирал, мы переключались на данный SpectatorClass.

Нам нет необходимости создавать свой собственный класс, будем пользоваться тем, который имеется по умолчанию.

Функция, которая отвечает за переключение пауном находится в контроллере, поэтому подключаем ЗФ у персонажа и в функции OnDeath проверяем на валидность Controller и вызываем функцию ChangeState().

У нее есть один параметр типа FName. Мы воспользуемся константой NAME\_Spectating:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

3. Через ЗФ нашего контроллера (STUPlayerController – он, оказывается, у нас был) проваливамся в базовый контроллер и находим там функцию ChangeState:

Изображение выглядит как текст

Автоматически созданное описание

В данной функции происходит смена управления пауном. Здесь проверяется смена двух состояний – NAME\_Playing и NAME\_Spectating – отвечают за паун обычного пауна и пауна наблюдателя.

Если мы проваливамся в функцию BeginSpectatingState(), то увидим, что вначале проверяется, был паун или нет, если был – то вызываем UnPossess(). Далее уничтожается предыдущий наблюдатель и в конце спаунится новый и контроллер получает над ним управление.

Изображение выглядит как текст

Автоматически созданное описание