**6. Spectator Widjet + Animation**

1. Что будем делать при смерти персонажа с UI элементами? Объявили две функции-проверки, для чего и какой функционал? А так же получение компонента еще одного, для чего и какой функционал?

2. Как настраиваем пропадание UI-элементов?

3. Что создаем для режима наблюдателя? Какие элементы добавляем? Как настраиваем?

4. Куда добавляем виджет из предыдущего пункта? Как для него настроили видимость?

5. Добавляем анимацию нашему элементу текста – как? Как ее заставляем включаться в ивент графе?

1. Будем делать невидимыми все UI-элементы при смерти персонажа и добавлять UI-виджет, который будет отображаться в режиме наблюдателя.

Для начала объявим две функции для проверки того, жив ли персонаж или нет, и для проверки того, в режиме наблюдателя сейчас мы или нет.

А так же добавим функцию GetHealthComponent, так как мы будем в функции IsPlayerAlive использовать HealthComponent:

Изображение выглядит как текст, снимок экрана, экран

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Не забываем поменять код функции GetHealthPercent из-за новой функции:

Изображение выглядит как текст

Автоматически созданное описание

После смерти персонажа контроллер захватывает SpectatorPawn поэтому функция isDead() возвращать true никогда не будет, то есть условие того, что персонаж умер, будет определяться по нулевому указателю на HealthComponent:

Изображение выглядит как текст

Автоматически созданное описание

Функция получения текущего контроллера игрока называется GetOwningPlayer:

Изображение выглядит как текст

Автоматически созданное описание

2. Разберемся с видимостью наших элементов. Находим, например, у ProgressBar в Details->Behavior категорию Visibility и биндим сюда функцию IsPlayerAlive:

Изображение выглядит как текст, внутренний, снимок экрана

Автоматически созданное описание

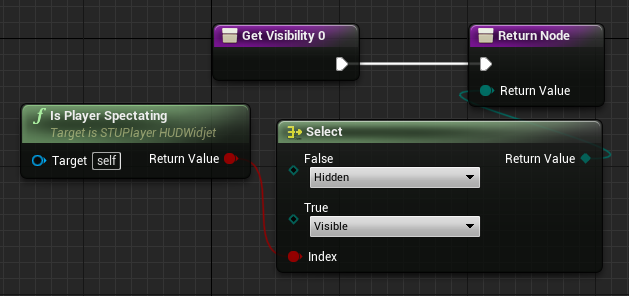
У остальных элементов (прицела и арсенала) указываем эту же функцию.

3. Теперь создадим еще один виджет WBP\_SpectatorHUD. Добавляем элемент Circular Throbber – бегающие по окружности кружочки (значок загрузки), и текст.

Воспользуемся элементом Overlay. Он помогает группировать элементы и располагать их слоями – один над другим. Устанавливаем SizeX и SizeY = 200 и ставим в центр. Отцентровали текст в самом контейнере Overlay. У троббера настроили выравнивание так, чтобы он занимал все пространство, а так же можно настроить количество кружочков, период их обращения.

4. Переходим в PlayerHUD. Каждый виджет может содержать другие виджеты. Поэтому в Palette->UserCreated найдем наш WBP\_SpectatorHUD, который может добаить в наш WBP\_PlayerHUD. Прикрепляем к центру и делаем выравнивание как у прицела.

Далее для него так же задаем видимость через биндинг-функцию:



5. Еще раз зайдем в SpectatorHUD, нажмем на текст. В левом нижнем углу есть раздел Animations, в котором можно добавить анимацию элементу. Создадим TextBlinking.

Далее Track->Text. У нас создался элемент на таймлайне. Для анимации нажимаем на +Track и выбираем Color and Opacity. По умолчанию у нас имеется один фрейм. Двигая ползунок и нажимая кнопочку +, мы получаем новые ключевые кадры. Делаем цвет белым-красным-белым.

Изображение выглядит как текст

Автоматически созданное описание

Последнее, что осталось – запустить проигрывание данной анимации при старте игры. В панели слева у нас появилась переменная, отвечающая за анимацию. Мы достаем данный объект на ивент граф и вызываем у него функцию PlayAnimation. Меняем Num Loops to Play – при установке 0 будет проигрываться бесконечно.

У виджетов нет знакомой функции BeginPlay. Здесь имеется функция Event Construct. В целом можно запустить на этот ивент, но если мы будем динамически удалять и добавлять виджет во вьюпорт, данный ивент будет генерироваться каждый раз при добавлении. Есть другое, более подходящий ивент EventOnInitialized – он будет вызван все один раз при старте игры (как BeginPlay).

