**15. LEVEL PRELOADING**

1. Сделаем статический прелоудер – для чего и почему? Какие есть другие варианты решения проблемы? Какая сейчас функция вызывается при клике на кнопку начала игры и что в ней происходит? Как мы теперь переопределим поведение – что есть в базовом классе UserWidget, что мы будем использовать? Добавили проперти и функцию данную и переопределили поведение ранее используемой функции.

2. Как настраиваем анимацию? Что важно установить для нашего Image? Справка – как найти ЗФ любого UI-элемента?

1. Сделаем статический прелоудер. Когда мы загружаем наш уровень, то весь интерфейс подвисает, потому что уровень подгружается синхронно. В UE данную проблему можно решить несколькими способами – можно сделать асинхронную загрузку, стриминг левела, мы решим данную проблему с помощью анимации. Переведем экран в черный цвет и **по завершению** анимации подгрузим уровень.

Когда мы нажимаем на кнопку начала игры, вызывается функция OnStartGame – в ней открывается выбранный уровень. Изменим логику следующим образом: в функции OnStartGame будем запускать анимацию, а когда она завершается, будем вызывать код, который сейчас находится в функции. Чтобы понять, что анимация закончилась, в базовом классе UserWidget есть два нативных ивента:

Изображение выглядит как текст

Автоматически созданное описание

Переопределим функцию OnAnimationFinished в нашем MenuWidget, данная функция будет вызываться каждый раз, когда завершилась какая-нибудь анимация. Какая анимация завершилась мы можем понять по параметру, который имеет данная функция. А так же добавим проперти для анимации:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

У нас еще запускается анимация ShowAnimation при загрузке виджета меню, поэтому, чтобы загрузка уровня происходила только в случае окончания HideAnimation, добавили проверку.

2. Добавляем HideAnimation в MenuWidget. Анимировать мы будем ту же картинку, что анимировали в ShowAnimation. Так же заанимируем проперти ColorAndOpacity, только уже от прозрачного к черному. Дополнительно добавили проперти Visibility и установить его в Visible, чтобы после старта анимации все кнопки стали недоступны – пользователь не сможет ничего нажать в процессе проигрывания анимации. Дополнительно добавили текстовый UI-элемент. Чтобы он не перекрывал наши кнопки (так как как он всегда будет во вьюпорте), выключили ему Visibility с помощью Not Hit-Testable и прозрачность нулевой.

Справка – чтобы узнать, какой C++-класс соответствует UI-элементу, вытаскиваем его на CanvasPanel (или просто Layout) и справа вверху есть гиперссылка Open \*название\_элемента\*.