**3. GAME PAUSE, meta = (BindWidjet)**

1. Что реализуем в данном уроке?

2. Создали мэппинг – какой? Где будем реагировать на нажатие клавиши и почему? Переопределяем одну базовую и вводим одну дополнительную функцию. С помощью какой функции останавливаем игру и как она работает под капотом (на словах)?

3. Что нужно переопределить в гейм моде и почему?

4. Сделаем так, чтобы вместо интерфейса пользователя показывался специальный виджет паузы. Создаем его (BP), что добавляем на сам виджет? Что добавляем худу в плюсах (три проперти)? Как определяем поведение в BeginPlay теперь и колбек-функцию? Что настроить в BP-классе худа?

5. Добавляем кнопку на виджет, как настраиваем? Следующая наша задача – показывать курсор когда мы зашли в состояние паузы и при нажатии на кнопку возвращать тики всем акторам. Создали базовый С++ класс для виджета паузы и всю нужную логику запрограммируем в данном виджете. Добавили проперти (+ особенный параметр в UPROPERTY) и две функции – как определили тут поведение? В гейм моде надо снова переопределить функцию – какую и зачем?

6. Осталось последнее – отобразить курсор мышки в состоянии Pause. Где это сделаем и как (добавляем две функции)?

7. Однако у нас наблюдается проблема – какая? Как ее исправили? Добавили еще на виджет специальный постпроцессинг-элемент – какой?

1. В данном уроке создадим логику паузы. Будем останавливать игру по нажатию какой-либо клавиши. Создадим виджет для режима паузы. И по нажатию кнопки в интерфейсе будем возвращаться к режиму игры.

2. Для начала создаем Action Mapping на клавишу P, назовем PauseGame.

Реагировать на нажатие клавиши будем в STUPlayerController, а не в персонаже как раньше, таким образом, пауза будет доступна, даже если персонаж погиб.

Переопределяем функцию SetupInputComponent из базового класса и private-функцию колбек для биндинга:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Включили ЗФ базового гейм мода, так как каст к нашему не понадобится, потому что функция SetPause (позволяющая поставить нашу игру на паузу) находится в базовом классе. В качестве параметра принимает контроллер, который хочет поставить игру на паузу. Под капотом в данной функции происходит остановка тиков всех акторов.

3. Функцию SetPause нам нужно переопределить (в public-секции) в нашем классе гейм мода, потому что помимо самой паузы мы хотим изменить стейт нашего матча.

Изображение выглядит как текст

Автоматически созданное описание

Функция возвращает булево значение, поэтому мы будем проверять это заранее, чтобы установить новый матч-стейт, только если пауза была установлена.

4. Сделаем так, чтобы вместо интерфейса пользователя показывался специальный виджет паузы. Создаем новый BP виджет WBP\_GamePause. Добавляем ему VerticalBox. Устанавливаем его в середину. Добавляем текст. Сделаем так, чтобы данный виджет показывался, когда мы нажимаем клавишу P.

Изображение выглядит как текст

Автоматически созданное описание

В STUGameHUD создадим виджет, который будет отвечать за паузу. Для хранения всех виджетов будет использовать ассоциативный массив, а так же добавим указатель для текущего виджета.

Изображение выглядит как текст

Автоматически созданное описание

На BeginPlay будем создавать все виджеты. Класс TMap в соответствие каждой паре ставит TPair, у которого есть поля Key и Value. Мы добавляем каждый виджет во вьюпорт, но нам надо сделать их невидимыми при этом с помощью функции SetVisibility.

Изображение выглядит как текст

Автоматически созданное описание

Теперь необходимо при изменении матч-стейта включать нужный виджет для данного стейта.

В блюпринте гейм-худа надо будет указать PauseWidjetClass = WBP\_GamePause.

5. Добавим кнопку на виджет паузы, при нажатии на которую у нас будет пауза убираться. Для этого добавляем элемент Button нашему VerticalBox. Добавили Spacer между текстом и кнопкой.

Изменить размер элемента в Layout’e можно несколькими способами, один из них – обрамить его в специальный контейнер через ПКМ->WrapWith->SizeBox. Тут уже можно настроить размер.

Добавляем еще текстовый элемент на нашу кнопку с текстом Continue.

Настроить стили кнопки при нажатии, наведении и нормальном состоянии в разделе Style.

Следующая наша задача – показывать курсор когда мы зашли в состояние паузы и при нажатии на кнопку возвращать тики всем акторам. Создали базовый С++ класс для виджета паузы и всю нужную логику запрограммируем в данном виджете.

Изображение выглядит как текст

Автоматически созданное описание

Создали проперти для кнопки, функцию Initialize. У проперти кнопки нужно указать специальную мета-информацию, которая ассоциирует кнопку, которая находится в BP, с кнопкой, которую мы будем использовать в C++. А так же создадим функцию колбек, которая будет вызываться по нажатию кнопки. Делегат, на который мы будем подписываться – динамический, поэтому добавляе макрос UFUNCTION.

Изображение выглядит как текст

Автоматически созданное описание

В функции Initialize мы подписываемся на делегат кнопки OnClicked (есть еще множество других). В функции OnClearPause мы вызываем функцию гейм мода ClearPause – она возвращает тики всем акторам. Ее в нашем гейм моде так же нужно переопределить:

Изображение выглядит как текст

Автоматически созданное описание

6. Осталось последнее – отобразить курсор мышки в состоянии Pause. Сделаем это в PlayerController. Для этого переопределим BeginPlay, в котором подпишемся на изменение состояния игры, для чего добавим колбек:

Изображение выглядит как текст

Автоматически созданное описание

Не забываем поменять базовый класс нашему виджету. При этом у нас возникнет ошибка компиляции, связанная с кнопкой. Таким образом, если мы декларируем какой-то UI-элемент в плюсах и указываем meta = (BindWidjet), в виджете у нас должен быть элемент с данным именем. Поэтому переименовали кнопку.

7. Однако у нас наблюдается проблема с тем, что событие наведения мыши не срабатывает, надо явно кликнуть на кнопку. При этом после клика у нас пауза снимается, но надо еще раз кликнуть, чтобы убрать курсор. Для того, чтобы явно указать вид инпута, можно воспользоваться функцией контроллера, которая называется SetInputMode:

Изображение выглядит как текст

Автоматически созданное описание

Добавим еще в виджет декоративный элемент BackgroundBlur, который размывает фон на заднем плане и растягиваем на весь экран. Регулируем параметр BlurStrength для настройки размытия фона.