**11. LEVEL SELECTION**

1. Добавим возможность выбора уровня в главном меню. Создали BP-виджет и настроили его – как?

2. Эти виджеты будем добавлять динамически на виджет меню. Что добавили для этого виджету меню (куда будем помещать)? Где опишем логику добавления LevelItem’ов? Но где ее лучше будет создавать, если у нас разрастется код виджета меню? Создаем C++-виджет STULevelItemWidget. Что определили в STUCoreTypes (две вещи - какие)?

3. Что добавляем в ЗФ гейм инстанса (проперти, одно проперти поменяли и перенесли, добавили две функции и одну переопределили)?

4. Добавляем 6 проперти в LevelItemWidget и 5 функций – для чего? Подсказка – 3 проперти для переменных из FLevelData, еще 3 связаны с ними, касательно функций – функция-колбек, сеттер, геттер, функция для видимости определенного объекта и одна базовая функция. Какое в них определили поведение?

5. Осталось написать код создания LevelItem в MenuWidget. Добавляем три проперти и три функции – какие и для чего? Так же понадобилось подправить одну ранее реализованную функцию – какую и как?

6. Как настраиваем WBP\_LevelItem? Как WBP\_Menu? Какие создали уровни? Как настроили гейм инстанс? Как можно добавить подсказку к какому-то проперти в эдиторе?

1. Добавим возможность выбора уровня в главном меню.

Изображение выглядит как текст, снимок экрана, электроника, компьютер

Автоматически созданное описание

Создаем новый BP-виджет WBP\_LevelItem – он будет представлять одну плитку для выбора уровня. В данном виджете будет несколько элементов, которые надо сгруппировать по слоям, для чего используем Overlay. Так же добавили кнопку, чтобы реагировать на нажатие. Добавляем ее в самый верхний слой, выравнивание заполнения по ширине и высоте и свойство Render Opacity устанавливаем в 0. У картинки для рамки ставим значение Visibility в Hidden по умолчанию.

2. Эти виджеты будем добавлять динамически на виджет меню. Добавляем последнему HorizontalBox, в который будем добавлять наши виджеты.

Логику добавления LevelItem’ов мы создадим в меню виджете, так как у нас данный виджет достаточно простой. Но если функциональность меню виджета будет увеличиваться, лучше добавлять данные айтемы не в HorizontalBox, а создать для этого специальный подвиджет, в котором инкапсулировать логику и просто вставлять его целиком.

Создаем C++-виджет STULevelItemWidget. Для начала в STUCoreTypes.h добавляем структуру для данных наших уровней (LevelItem). Она будет содержать имя уровня, имя уровня для отображения и картинку для отображения (без рамки, короче). А так же добавляем делегат, который будет передавать данные от LevelItem в MainMenu – мы будем понимать, какой уровень выбран.

Изображение выглядит как текст, снимок экрана, экран

Автоматически созданное описание

3. В ЗФ файле гейм инстанса нужно добавить проперти, в котором мы будем настраивать все данные по уровню нашей игры – массив с LevelData’ми (вручную мы теперь не будем выбирать StartUpLevelName – это будет происходить автоматически, в зависимости от того, какой элемент массива выбран, поэтому переместили данную переменную в секцию private + поменяли тип данной переменной и геттер для нее + сеттер), а еще геттер для всего массива уровней:

Изображение выглядит как текст

Автоматически созданное описание

4. В LevelItemWidget добавляем кнопку, текст блок и картинку (которые мы и будем устанавливать из LevelData), а также для рамки:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Так же у нас есть переменная FLevelData, которая будет относиться непосредственно к данному LevelItem. Функция колбек, которая будет срабатывать при нажатии на кнопку LevelSelectButton. Так же добавили делегат из STUCoreTypes и сеттер для FLevelData, с помощью которого при создании данного виджета мы будем устанавливать соответствующие ему данные. И геттер для него же. А так же функцию SetSelected, с помощью параметра которой мы будем понимать, выбран виджет или нет – в зависимости от этого делать видимой или невидимой рамку.

5. Осталось написать код создания LevelItem в MenuWidget. Нам нужен будет HorizontalBox, в который будем добавлять LevelItem. А так же GameInstance и LevelItemWidget. Массив указателей на созданные объекты виджетов. В InitLevelItems находится основная логика создания плиток наших уровней. А так же функция-колбек для делегата, который мы создали в LevelItemWidget – будет вызываться, когда мы нажали на одну из кнопок выбора уровня. И вспомогательная функция для получения GameInstance.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Функция InitLevelItems будем вызывать один раз в функции NativeOnInitialized. В цикле идет по всем элементам из гейм инстанса (в которым мы описали названия и картинку), создаем виджеты, устанавливаем им LevelData, подписываем на делегат из виджета нашу функцию в меню виджете, добавляем в бокс и массив виджетов.

Изображение выглядит как текст

Автоматически созданное описание

В функцию OnLevelSelected мы попадаем, когда пользователь нажимает на одну из плиток. Для начала надо сразу поменять переменную StartupLevel в соответствии с данными, которые пришли в колбек делегата. После этого идем по всему массиву виджетов и определить, какой из них был выбран. Это нужно, чтобы сделать видимой или нет рамку вокруг картинки.

Изображение выглядит как текст

Автоматически созданное описание

В конце проверяем, пустой у нас стартап левел или нет. Если да – вызываем напрямую LevelSelected для нулевого элемента (безопасно, потому что у нас уже точно что-то там есть), если нет – данная ситуация возможна, когда мы выбрали уровень, поиграли в игру и вернулись в MainMenu. В данном виджете снова вызвалась функция инициализации, но стартап левел уже не None, поэтому вызываем функцию OnLevelSelected со стартап левелом, чтобы просто подсветить данную плитку.

Изображение выглядит как текст

Автоматически созданное описание

Подправим функцию OnStartGame в соответствии с нашими изменениями. Создадим гейм инстанс и в функцию передадим имя стартап уровня. Проверку на None убрали, потому что код в InitLevelItems в конце предотвращает такую возможность.

6. Переопределяем родителя нашему WBP\_LevelItem. Настраиваем имена. В блюпринте меню надо настроить название для HorizontalBox и указать класс виджета, который мы будем создавать. Создали уровни для теста с названиями SphereLevel и CubeLevel, не забыв указать гейм мод. Переходим в гейм инстанс и настраиваем наш массив LevelData.

Если у нас два уровня с одинаковым именем, то у нас выделяются два уровня сразу. Фиксить это не будем, но можно указывать дополнительную мета-информацию для проперти, которая умеет выводить подсказку в эдитор:

