**7-8. GAME INSTANCE – STARTUP LEVEL NAME**

1. Когда создается и когда доступен объект данного класса? Что в нем удобно хранить? От кого наследовались при создании своего гейм инстанса? Добавили ему строчку и протестили ее изменение – как получить доступ к гейм инстансу (две перегрузки функции)? Где переопределить базовый класс гейм инстанса?

2. Будем теперь узнавать название стартового уровня из гейм инстанса – добавили проперти (и особенное значение по умолчанию) и функцию. Какое и где определили поведение у STUMenuWidget? Специальная функция для проверки на это особенное значение?

1. Объект данного класса создается во время старта игры и доступен на протяжении всего времени, пока наша игра запущена. В нем удобно хранить настройки, различную инфу, которая не зависит от данного уровня. Мы в любой момент можем либо перезаписать ее, либо прочитать. Создадим свой класс STUGameInstance, унаследовавшись от GameInstance.

Данный класс доступен из любого места нашей программы. Добавим нашему класс переменную:

Изображение выглядит как текст

Автоматически созданное описание

Изменим значение данной переменной при загрузке уровня меню. Сделаем это (произвольно) на BeginPlay нашего контроллера:

Изображение выглядит как текст

Автоматически созданное описание

У функции получения гейм инстанса есть две перегрузки – она либо возвращает указатель на базовый тип (и нам придется потом делать каст), либо на наш, если мы ее шаблонизируем.

Изображение выглядит как текст

Автоматически созданное описание

При запуске TestLevel в гейм моде прочитаем значение данной переменной и выведем в лог на StartPlay. Таким образом мы проверим, что объект данного класса не пересоздался.

Создадим BP-класс нашего инстанса. Теперь переопределим базовый класс гейм инстанса в Project Settings.

2. Почистили предыдущий код весь. Имя уровня, который необходимо нам загрузить, будем хранить в классе GameInstance:

Изображение выглядит как текст, снимок экрана, экран, квартира

Автоматически созданное описание

Конструктор по умолчанию FName инициализирует ее специальным значением None – мы это укажем явно, данное значение хранится в переменной NAME\_None. Перед загрузкой будем проверять значение переменной и, если оно равно None, будем выводить сообщение об ошибке. Для этого еще ввели геттер.

В cpp-файле STUMenuWidget реализуем следующую логику:

Изображение выглядит как текст

Автоматически созданное описание

В классе FName есть специальная функция для проверки на значение None.

Остается в ассете GameInstance значение Startup Level Name установить в TesetLevel (в моем случае).