**6. PLAYER STATE**

1. Создали свой PlayerState, от кого наследовались? Как работает PlayerState в плане существования на протяжении игры и для чего используется? Что в данном уроке сделаем с игроками? Добавляем два проперти и четыре функции – какие?

2. Добавляем два проперти в GameData-структуру – какие?

3. Добавляем 3 функции в ЗФ гейм мода – какие, для чего, какая логика в них реализована? Где их вызываем? Переопределяем базовый класс в конструкторе. Остается создать функцию у нашего персонажа – какую?

1. Создали класс PlayerState в директории с нашим персонажем и назвали STUPlayerState, наследуясь от PlayerState (ОГО!).

PlayerState создается для каждого игрока, он, как и контроллер, существует на протяжении всей игры, он не удаляется вместе с персонажем при гибели, поэтому в нем удобно хранить различные данные, которые не относятся непосредственно к персонажу – чаще всего это статистические данные: количество смертей, убийств, имя игрока, номер команды и т.д.

В данном уроке распределим игроков по командам и каждой команде присвоим свой уникальный цвет. Добавляем в PlayerState два проперти и сеттеры и геттеры для них:

Изображение выглядит как текст

Автоматически созданное описание

2. Так же добавим два поля в GameData – для дефолтного цвета команды (будет использоваться в крайнем случае, если в BP\_STUGameModeBase не установят цвета для команд) и массив цветов, из которого мы будем выбирать цвет в зависимости от TeamID:

Изображение выглядит как текст

Автоматически созданное описание

3. В ЗФ гейм мода добавляем функции для распределения игроков по командам: первая будет вызываться один раз при старте игры и в ней мы будем распределять игроков по командам, вторая – возвращать цвет в зависимости от TeamID, третья – будет передавать классу персонажа его цвет:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

В конструкторе гейм мода задаем базовый класс PlayerState, функцию CreateTeamsInfo вызываем на StartPlay.

**Указатель на PlayerState хранится в контроллере**. Поэтому мы сможем как раньше проитерироваться по всем контроллерам и установить в PlayerState каждого все необходимые данные. Так как у нас две команды только, то определили их ID через тернарный оператор:

Изображение выглядит как текст, снимок экрана, экран

Автоматически созданное описание

В функции определения цвета проверяем, чтобы мы не выбежали за границы. В противном случае выводим Warning и дефолтный цвет (массив не был заполнен):

Изображение выглядит как текст

Автоматически созданное описание

Мы установили цвет команде в CreateTeamsInfo, после этого надо вызвать SetPlayerColor и передать контроллер. А так же нам надо добавить вызов этой функции, когда происходит респаун – только что созданный паун ничего не знает о том, какой у него должен быть цвет, поэтому добавим вызов и в ResetOnePlayer:

Изображение выглядит как текст

Автоматически созданное описание

В функции SetPlayerColor определяет такую логику:

Изображение выглядит как текст

Автоматически созданное описание

Остается создать функцию для нашего персонажа, не забыв добавить проперти для имени параметра материала (заранее преподаватель узнал это имя):

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

В эдиторе у нашего гейм мода надо заполнить массив цветов и все.