**3. AICharacterSpawner**

1. В данном уроке научимся динамически создавать NPC в GameMode. Для начало создаем структуру – какую и какое наполнение (пока не богатое)? Что будем спаунить (вопрос с подвохом) и почему? Добавляем два проперти – куда и какие? Какую переопределим функцию из базового класса и добавим нашу? Какое определяем поведение в обеих?

2. Настраиваем блюпринт (что именно, смотри предыдущий пункт) и запускаем игру – что случилось? Что нужно, чтобы заспаунить персонажей?

3. Как работает функция для спауна персонажей (вопрос объемный, поэтому вот подсказки: ищем точку -> спауним там актора -> проверяем пауна -> если нет, возвращаем пауна по умолчанию -> и устанавливает контроллеру пауна -> последняя что-то устанавливает и вызывает функцию спауна в точке -> а последняя уже вызывает спаун просто)?

4. Что произойдет теперь, если запустим игру? Как это исправили? Однако наблюдается еще одна проблема – все персонажи \_ - почему так происходит (ответ кроется в одной из функций, которая вызывается в цепочке предыдущих)? Как переопределить ее поведение (тут необычный способ переопределения – какой и как работает)? Какое проперти добавляем для переопределения? Какое поведение определяем в функции? Что останется настроить в BP нашего режима игры?

1. В данном уроке научимся динамически создавать NPC в GameMode. Для начала создадим структуру данных, в которой будут храниться все настройки игры, по мере продвижения по данному разделу будем ее дополнять:

Изображение выглядит как текст

Автоматически созданное описание

В ЗФ нашего режима игры подключаем ЗФ с нашей структурой и спавнить будем не самих персонажей, а их контроллеры. У персонажа вызывается Destroy в случае гибели персонажа, контроллеры же при этом со сцены не удаляются и они и будут отвечать за респаун персонажа – поэтому предпочтительнее спаунить именно контроллеры – добавляем для них проперти и для нашей новой структуры:

Изображение выглядит как текст

Автоматически созданное описание

Перейдем в GameModeBase и найдем функцию StartPlay – данная функция вызывается до BeginPlay GameMode и всех остальных акторов. В ней мы будем инициализировать все необходимые данные для нашей игры и спаунить NPC персонажей. Переопределяем эту функцию в нашем режиме и вводим функцию для спауна контроллеров:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Нам нужно дополнительная переменная типа FActorSpawnParameters, в ней определяем, как должны обрабатываться коллизии при спауне наших игроков. Для этого в этой структуре есть поле SpawnCollisionHandlingOverride, значения которого представляют собой enum, элементы которого определяют, каким образом будут обрабатываться коллизии. Мы выбрали AlwaysSpawn. И далее заспаунили контроллерра.

2. В BP-классе нашего игрового режима устанавливаем проперти AIController’a и запускаем игру. В итоге мы на сцене одни, создались только контроллеры. Чтобы заспаунить AICharacter для каждого AIController, получим возвращаемый указатель из функции SpawnActor. Функция GameMode, которая создает пауна, называется RestartPlayer и принимает один параметр – базовый класс контроллера:

Изображение выглядит как текст

Автоматически созданное описание

3. В данной функции находится точка, в которой необходимо заспаунить пауна (FindPlayerStart), после этого вызывается RestartPlayerAtPlayerStart:

Изображение выглядит как текст

Автоматически созданное описание

В которой проверяется, существует ли паун у контроллера. Если нет, то вызывается функция GetDefaultPawnClassForController, которая возвращает класс пауна, и после этого контроллеру устанавливается паун, которого возвращает функция SpawnDefaultPawnFor:

Изображение выглядит как текст

Автоматически созданное описание

Последняя определяет трансформация, с которой надо заспаунить пауна, и вызывает функцию SpawnDefaultPawnAtTransform:

Изображение выглядит как текст

Автоматически созданное описание

Здесь уже вызывается функция SpawnActor.

Изображение выглядит как текст

Автоматически созданное описание

4. Если мы теперь запустим нашу игру, все персонажи появятся в одной и той же точке. Все это время у нас на сцене существовал специальный актор PlayerStart. С помощью акторов данного типа в GameMode вычисляется точка старта для нового пауна. Продублируем данные PlayerStart. Теперь GameMode при запуске выбрал подходящие PlayerStart для каждого персонажа и заспаунил их с нужными трансформациями.

Однако если мы посмотрим, какие создались персонажи, мы увидим, что создались не AICharacter, а STUBaseCharacter.

Изображение выглядит как текст, табло, снимок экрана

Автоматически созданное описание

Почему так происходит? Зайдем в функцию RestartPlayer в базовом GameMode. Мы уже упоминали функцию GetDefaultPawnClassForController – именно она выдает класс пауна, который необходимо создать. Она выглядит следующим образом:

Изображение выглядит как текст

Автоматически созданное описание

Код под макросом нас не интересует. А происходит здесь одно – возвращается DefaultPawnClass – то есть STUBaseCharacter.

Однако данную функцию можно переопределить. Обратим внимание, что у нее есть приписка \_Implementation – находим данную функцию в ЗФ и обратим внимание на спецификатор BlueprintNativeEvent:

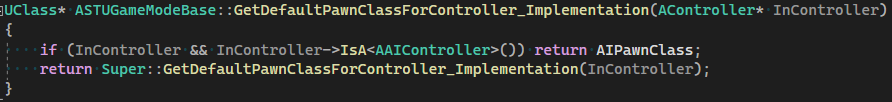
Изображение выглядит как текст

Автоматически созданное описание

В документации написано, что функции данного типа можно переопределять в блюпринтах и C++ - необходимо добавить окончание \_Implementation: в результате автоматической кодогенерации UE, если нет BP-функции, то будет вызываться данная функция с \_Implementation. Поэтому переопределим данную функцию в нашем классе GameMode, добавив проперти для указания нужного класса AICharacter для спауна (при этом укажем APawn):

Изображение выглядит как текст

Автоматически созданное описание



У функции есть параметр контроллер, который мы проверяем на то, является ли он AIController или PlayerController, и в зависимости от этого выдаем нужный класс пауна. Останется только настроить в BP GameMode установить BP\_STUAICharacter.