**5. RESET ALL PLAYERS**

1. В данном уроке сделаем респаун всех персонажей. Какие две вещи нам надо сделать в начале каждого раунда? Вводим для этого две функции. Что происходит с персонажами и контроллерами во время игры? Когда вызываем функцию перезагрузки персонажей? Как получить информацию о всех контроллерах, которые находятся в мире (заодно определяем логику функции перезагрузки)? Что будем вызывать во второй функции? Вспомним, что в ней вызывается за функция и как она работает?

2. Нам надо делать \_ для всех паунов, которые остались в игре, и тогда функция рестарта сработает. Через какую встроенную функцию можно реализовать данную логику? Как она работает внутри?

3. Что будет, если мы сейчас запустим нашу игру (хаос в определенный момент)? Как это починили (а так же как используется данное проперти в одной функции, из-за чего и наблюдался хаос)? Какая теперь возникает проблема после фикса (связанная с рестартом NPC) и почему? Как нам ее уже исправить?

1. В данном уроке сделаем респаун всех персонажей. Каждый раз, когда начинается новый раунд, нам надо **перезагрузить** персонажей и **заспаунить** их в начальных позициях. Создадим две функции в режиме игры: ResetPlayers и ResetOnePlayer. Персонажи могут быть уничтожены во время игры, нам нужно будет их респаунить, а с контроллерами ничего не происходит – мы заспаунили их на StartPlay и они находятся на сцене на протяжении всей игры:

Изображение выглядит как текст

Автоматически созданное описание

Когда начинается новый раунд, вызываем функцию ResetPlayers:

Изображение выглядит как текст

Автоматически созданное описание

Информацию о всех контроллерах, которые находятся в мире, можно получить через специальный итератор из объекта UWorld. Получив данный итератор, мы можем создать цикл обхода по всем имеющимся в мире контроллерам. И для каждого контроллера на сцене вызовем функцию ResetOnePlayer, для получения сырого указателя на контроллер надо вызвать функцию Get:

Изображение выглядит как текст

Автоматически созданное описание

В функции ResetOnePlayer же будем вызывать уже знакомую функцию RestartPlayer.

В данной функции в конце вызывается функция RestartPlayerAtPlayerStart:

Изображение выглядит как текст

Автоматически созданное описание

Если Pawn не нулевой, то вычисляется текущее вращение пауна и записывается в SpawnRotation, и далее просто обновиться вращение контроллера – то есть ничего особенного не произойдет и паун останется на своем месте.

Если же паун нулевой, то персонаж во время игры был убит, и он будет заспаунен заново, то есть сработает логика, которая у нас выполняется на StartPlay (из предыдущих уроков).

2. По-хорошему, нам надо делать Destroy для всех паунов, которые остались в игре, и тогда функция рестарта сработает. Данную логику можно реализовать через специальную функцию актора Reset:

Изображение выглядит как текст

Автоматически созданное описание

Проверим, что контроллер и паун, которым он владеет, не нулевые (если у контроллера паун нулевой, то функция RestartPlayer сработает так, как нам надо). Если же условие выполнено, то вызываем у пауна функцию Reset.

Перейдем в данную функцию:

Изображение выглядит как текст

Автоматически созданное описание

Функция Destroy пауна вызовется, если определенно одно из условий: либо контроллер нулевой (что мы проверяем до входа в функцию), либо указатель на PlayerState контроллера **НЕ** нулевой. Таким образом, будет выполняться только второе условие. Что такое PlayerState – далее.

3. Если мы сейчас запустим нашу игру, то после окончания первого раунда у нас создастся куча NPC на сцене. Чтобы это исправить, надо в AICharacter исправить проперти Auto Possessed AI с Placed in World or Spawned на Disabled.

Теперь все хорошо. Кроме одного – когда начинается новый раунд, наш персонаж респаунится на стартовой позиции, а с NPC ничего не происходит. В дереве объектов мира поищем PlayerState и найдется только один такой – нашего персонажа. У NPC его нет, поэтому в функции Reset пауна Destroy вызвана не будет. Исправим это и заспауним PlayerState для NPC.

Для начала установим в плюсах настройку для AutoPossessedAI у нашего AICharacter:

Изображение выглядит как текст

Автоматически созданное описание

Теперь переместимся в код базового APawn и в cpp-файле посмотрим, как используется переменная AutoPossess. Найдем функцию PostInitializeComponents:

Изображение выглядит как текст

Автоматически созданное описание

Если AutoPossessAI имеет значение PlacedInWorldOrSpawned, то вызовется SpawnDefaultController. Таким образом, при старте игры создастся один лишний AIController. При начале следующего раунда во время рестарта данного контроллера заспаунится дополнительный паун, который в свою очередь заспаунит еще один контроллер и так далее. Все это в цикле итератора, поэтому у нас и заспаунилось гигантское количество персонажей.

Переходим в GameModeBase в функцию Reset. Destroy произойдет, если указатель на PlayerState не нулевой. У PlayerController PlayerState спаунится автоматически, а для AIController это надо явно указать – в классе AIController установим переменную bWantsPlayerState:

Изображение выглядит как текст

Автоматически созданное описание

Готово!