**TextColor – не трогать**

**TextColor – ожидает создания**

**TextColor – под вопросом**

**BP\_MyCharacter –** главный 3D персонаж, которым управляет игрок**.**

**Методы:**

**Category [Default]:**

**public:**

* **GrabSingleObject** – подобрать один объект и поместить в **GrabPlace**.
* **ReleaseSingleObject** – бросить подобранный объект
* **ReceiveDamage** – получить урон
* **EnableFullMovement** – включить полное движение персонажа.
* **DisableFullMovement** – выключить полное движение персонажа.

**private:**

* **GrabImmediately** – моментально подобрать объект (без анимации)

**Переменные:**

**Category [Components]:**

**public:**

* **GrabPlace** – место, где появляется взятый объект
* **MainCamera** – камера персонажа

**Category [Default]:**

**public:**

* **IsRayBlocked** – блокирует луч, который выходит из камеры для распознавания объектов, с которыми можно взаимодействовать. Так как переменная **TriggerObject**, которая находится в **BP\_MainGameMode** перезаписывается (в ней хранится объект типа BP\_TriggerBase, который одновременно является и объектом, с которым взаимодействует только луч (BP\_TriggerRayOnlyBase), и только персонаж (BP\_TriggerOverlapOnlyBase), а если луч столкнулся с объектом, с которым можно взаимодействовать и одновременно персонаж зашел в триггер, то переменная перезаписывает на последний объект, который столкнулся с триггером. Чтоб не было ошибок в дальнейшем, нужно блокировать луч. **(НУЖНО СДЕЛАТЬ ТАК ЧТОБ ЛУЧ НЕ НУЖНО БЫЛО БЛОКИРОВАТЬ, НАПРИМЕР В BP\_MAINGAMEMODE РАЗДЕЛИТЬ ЭТИ ПЕРЕМЕННЫЕ ТРИГГЕРА, ТО ЕСТЬ, TriggerOverlappedObject – КОТОРЫЙ ХРАНИТ ОБЪЕКТЫ ТРИГГЕРОВ С КОТОРЫМИ ПЕРСОНАЖ СТОЛКНУЛСЯ, TriggerRayObject – КОТОРЫЙ ХРАНИТ ОБЪЕКТЫ ТРИГГЕРОВ С КОТОРЫМИ ЛУЧ ПЕРСОНАЖА СТОЛКНУЛСЯ).**
* **CanGrabSingleObjects** – может ли игрок поднимать один объект
* **CanGrabMultipleObjects – может ли игрок поднимать несколько объектов**
* **CanDropGrabbedObject** – может ли игрок бросить поднятый объект
* **CanMove** – может ли игрок двигаться
* **CanJump** – может ли игрок прыгать
* **CanSprint** – может ли игрок бежать
* **CanCrouch** – может ли игрок приседать
* **CanRotate** – может ли игрок поворачиваться
* **GrabbedObject** – поднятый игроком объект

**private:**

* **MainGameMode** – ссылка на главный игровой режим
* **TriggerHitted** – триггер с которым луч игрока столкнулся
* **PreviousActorHitted** – предыдущий объект с которым луч игрока столкнулся (для того чтоб понимать луч был наведен на другой объект или на тот же самый. Если на тот же самый, то не нужно много раз выполнять одинаковые вычисления, так как выполнение луча срабатывает каждый кадр).
* **DefaultCrouchedSpeed** – скорость движения игрока по умолчанию, когда он присел
* **DefaultWalkSpeed** – скорость движения игрока по умолчанию, когда он ходит
* **RayLength** – длинна луча
* **Health** – количество здоровья игрока
* **IsDead** – мертв ли игрок

**Event Dispatchers:**

**Category [Default]:**

* **OnPlayerDead** – вызывается когда игрок мертв

**Процесс создания**

Добавить в игровой режим, в поле **Default Pawn Class**.

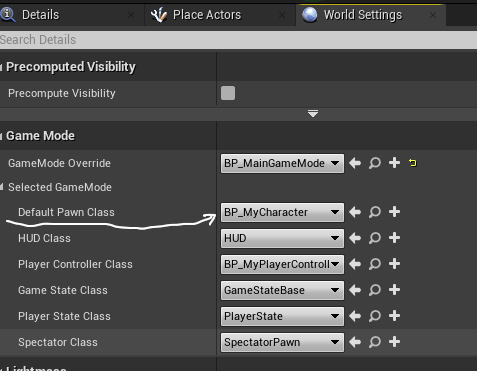
****

Рис.1.Назначение BP\_MyCharacter