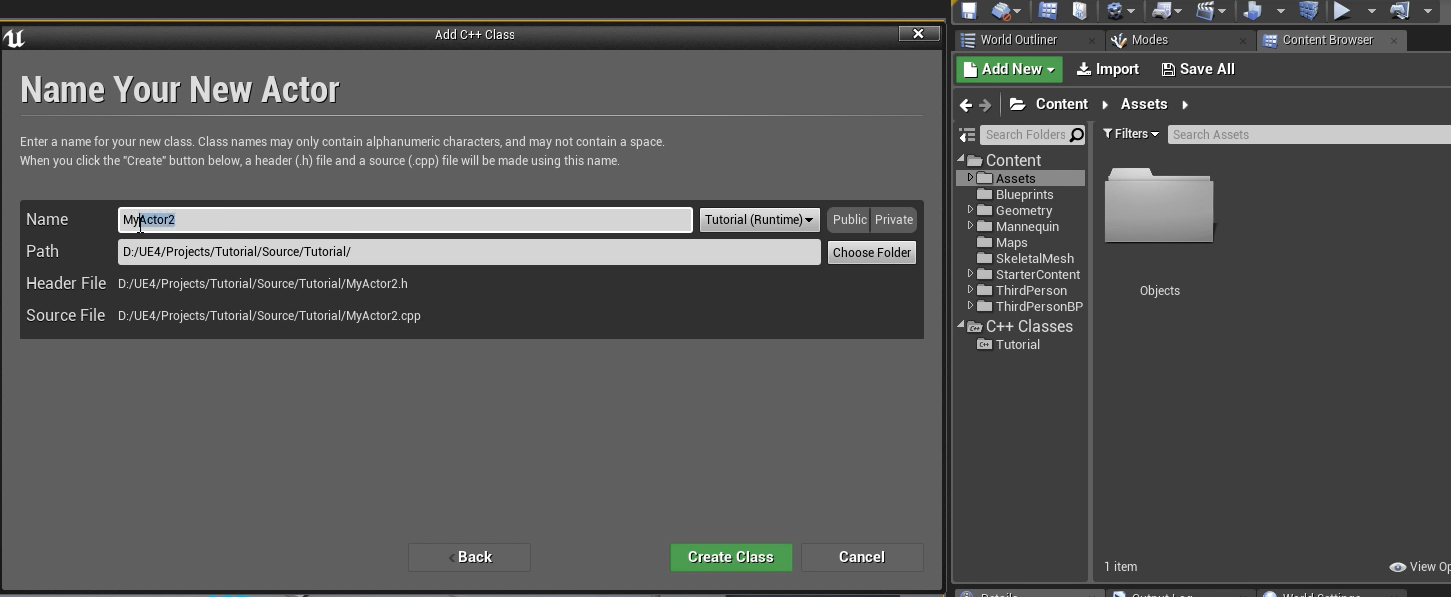
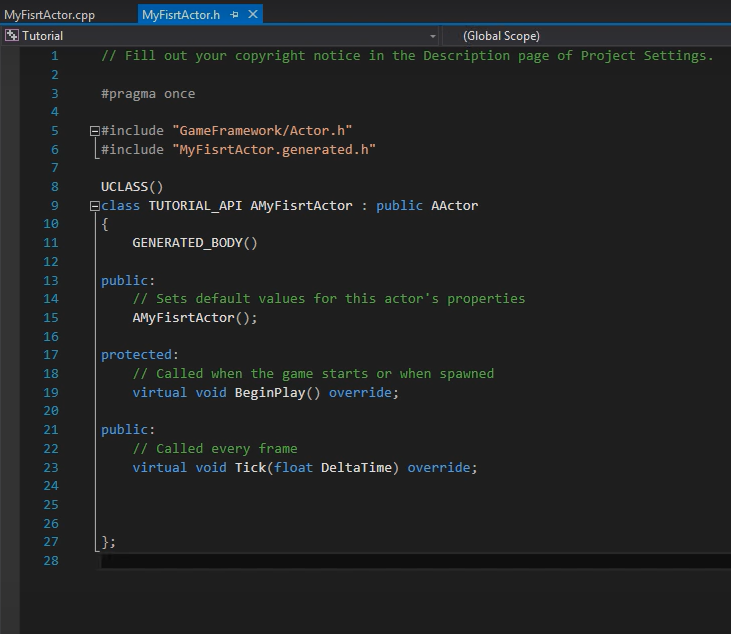
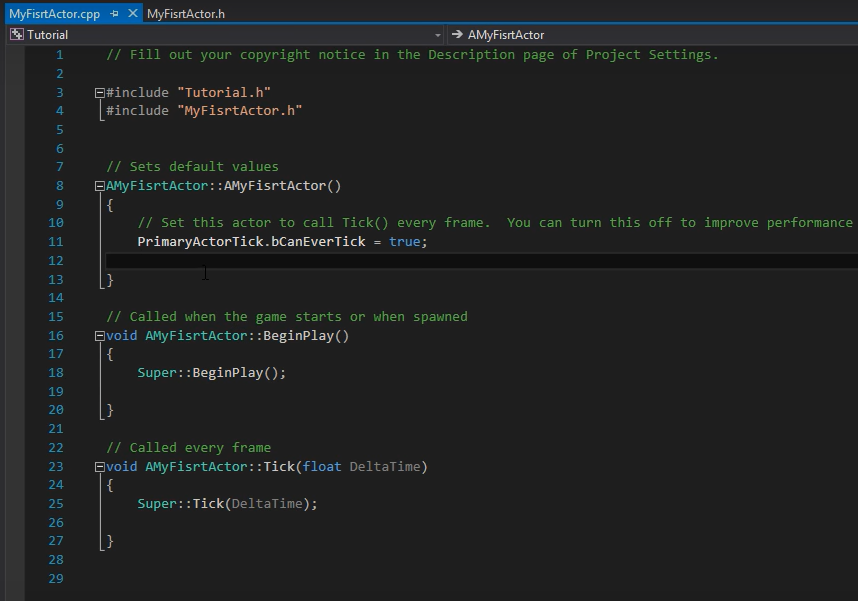
# C++ in UE

<https://www.youtube.com/watch?v=1_FHE7PWOX0&list=PLatwj2blGTZ67xsdMACTBPB8bsqiO5bIY&index=1>

Unreal Engine 4 C++ - 1. Создание класса







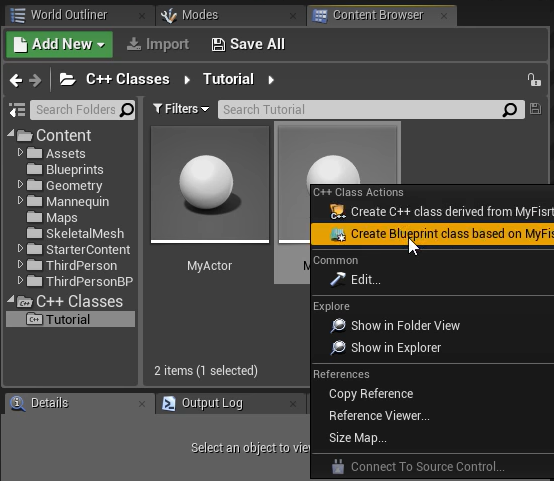
AMyActor::AMyActor() – это конструктор, вызываемый при каждом создании объекта (инстанса).

void AMyActor::BeginPlay() – это функция, вызываемая при начале игры, если актор (объект) уже стоит на сцене, или в момент появления (спауна) объекта. Мб вызвана только после конструктора, из функции SpawnActor.

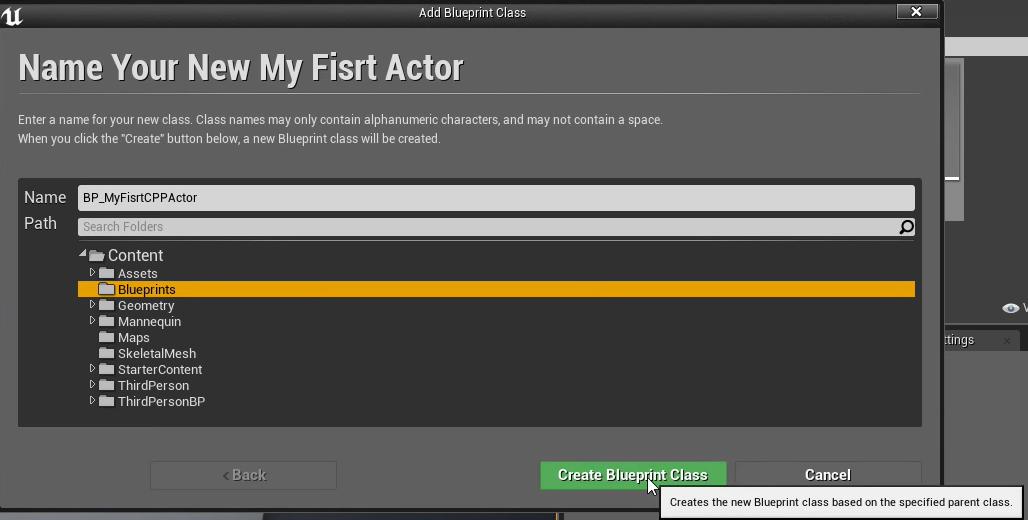
void AMyActor::Tick(float DeltaTime) – это функция, вызываемая каждый кадр, аналог blueprint’ового EventTick. Мб отключен в конструкторе.

Пока перемещение актора на сцену не дает ничего. Его нельзя увидеть, т. к. нет координат и свойств таких как повороти т. д.

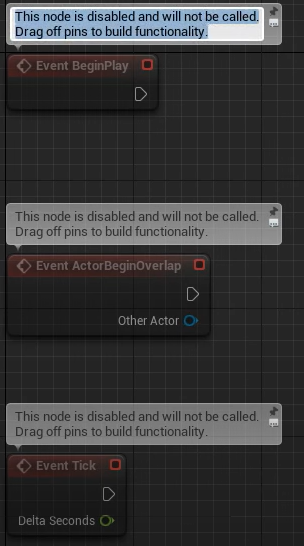
Для BP-вого имеет смысл ставить приставку BP. ОТ cpp-класса можно наследовать такой же или BP-класс.



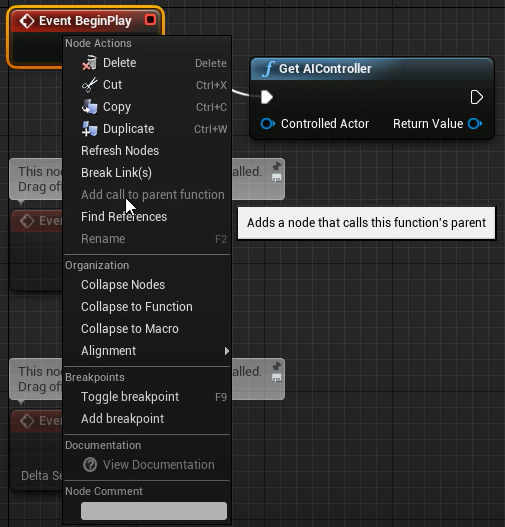
Создание BP-наследника



Те же свойства во вкладке «Event Graph» (они же «чертежи» и «BP»).



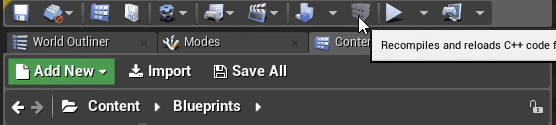
Для использования BeginPlay и Tick нужно использовать супер-логику в BP.



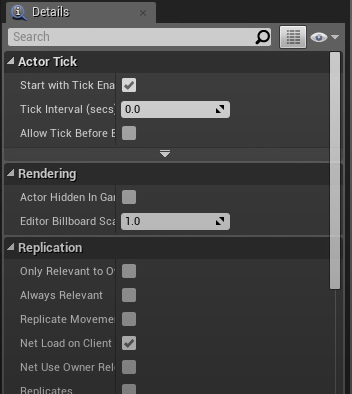
Кое-чего не хватает в макросе, поэтому нельзя вызвать родительскую функцию.

Переменные

Если создать обычную переменную в классе а cpp, то ничего не появися на сцене. Не забыть нажимать HotReload (кубик из кубиков) в Unreal (он же Rebuild в MS VS).

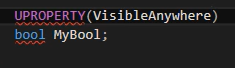


В наследном BP-классе ничего не появилось

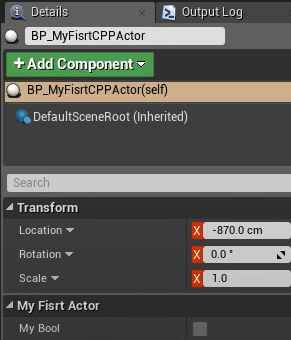


Сообщить редактору о существовании переменной поможет макрос UPROPERTY() 

Этот параметр позволить увидеть переменную в BP в свойствах класса



Однако параметр пока видимый, но не изменяемый



Чтобы можно было редактировать:



Пока эта переменная не инициализирована, однако можно придать ей конкретное значение в наследнике

Чтобы теперь сделать переменную видимой на карте, мб поставлен параметр:



Теперь можно у одного объекта поставить галочку, у другого снять. Желтый цвет – выделение, а не значений переменной true!



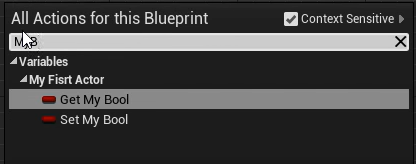
Чтобы мб вызвать переменную в BP только для чтения (защита):



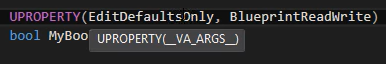
Для изменения значений



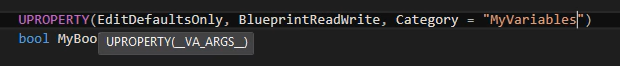
Теперь на чертежах видны методы set и get:



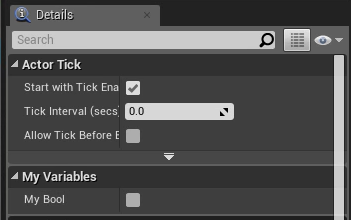
Чтобы в настройках обекта при разработке переменная не была раскрыта в Details, а изменяться могла лишь в Defaults, т. е. на чертеже (в дефолтсах). Т. о. дизайнер уровней обезопашен от запутывания.



Хорошая практика – расставлять категории, т. о. сортировать переменные.



Появление категории в Details и Defaults:



Другие типы данных:



Производная от вектора:



Для локализованного ткста:



В BP-классе тоже легко добавить переменные

