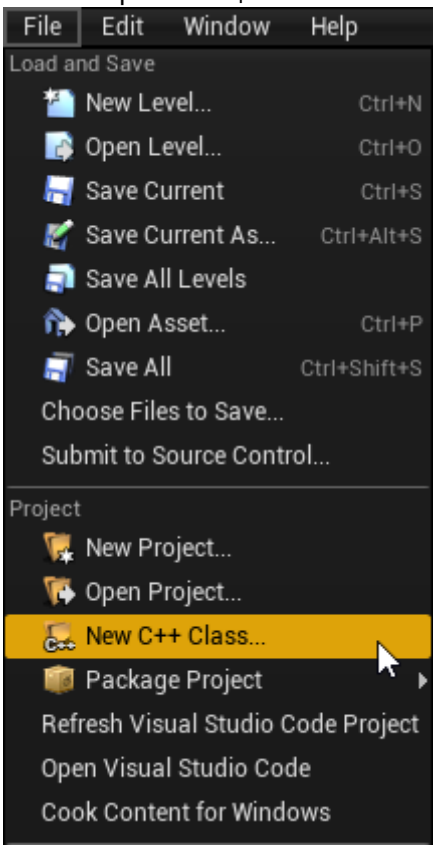
# Тип "механический"

ECS явным образом отделяет данные и логику, оперирующую этими данными. Эта логика, в свою очередь, обычно выполняется покадрово (то есть повторяется). Apparatus имплементирует эту функциональность через концепт, называемый *Mechanical* (механический тип). Механические объекты - как многогранный организм, включают несколько механик (Mechanics), которые выполняются внутри них.

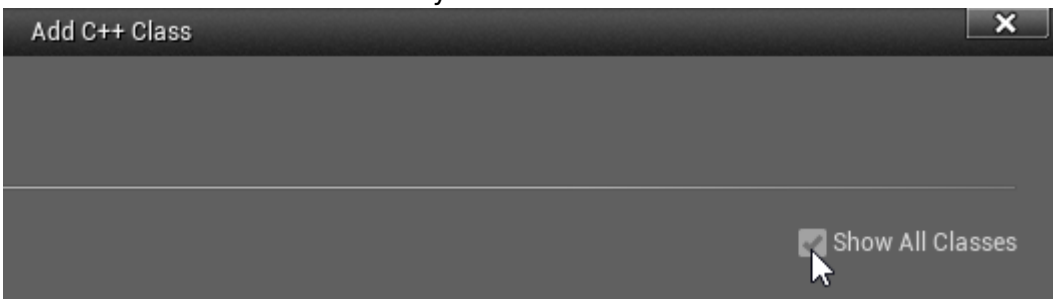
## Организация в C++

Если вы намерены идти C++ путём, создание собственного механического типа проходит примерно так.

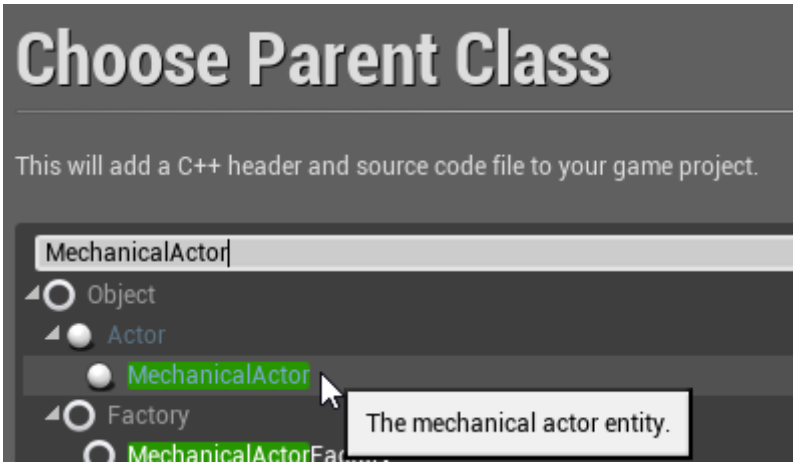
- Откройте главное UE File-меню и выберете опцию “New C++ Class...”:



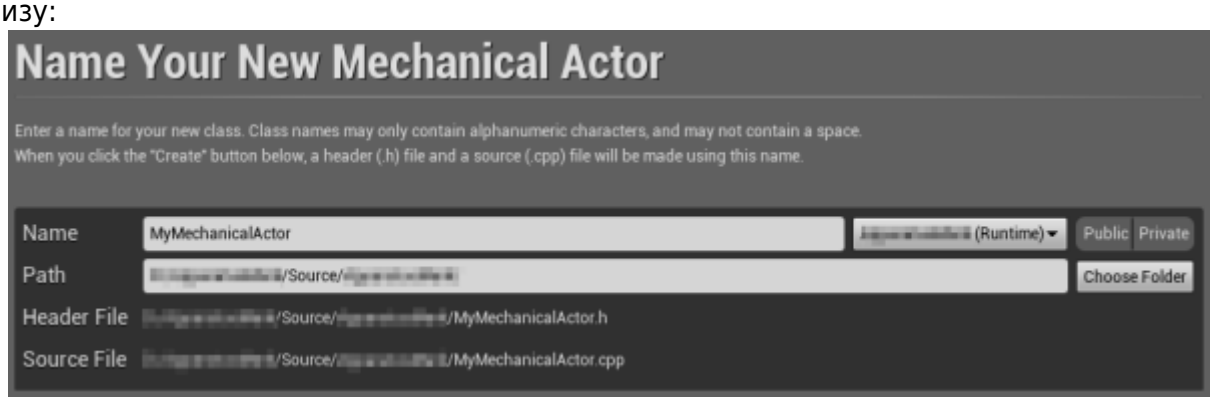

- В открывшемся окне поставьте галочку в чекбоксе “Show All Classes”:



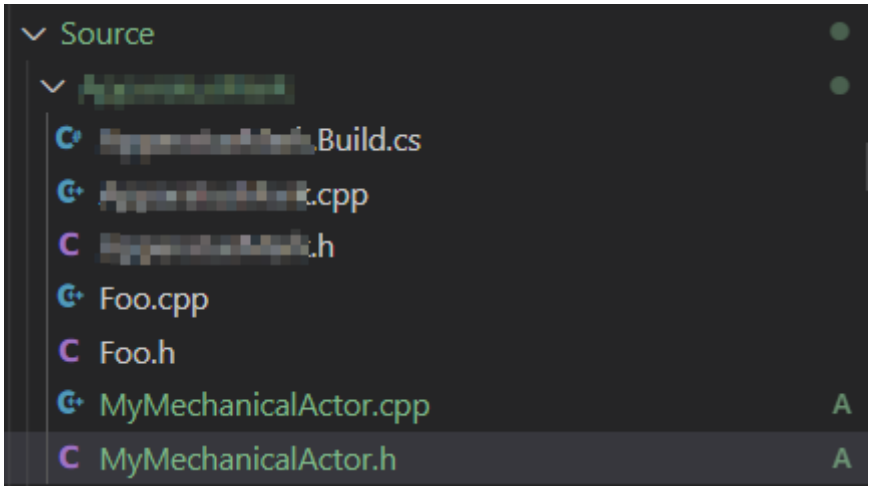

- Теперь вы можете выбрать любой доступный базовый класс, включая классы плагина. Выберете “Mechanical Actor” в качестве базового:



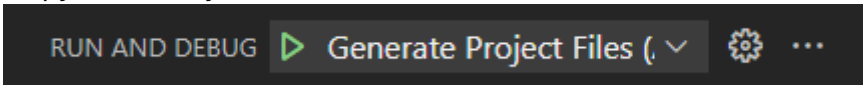

- Нажмите “Next”, и вы должны увидеть диалоговое окно выбора имени. Назовите создаваемый класс, как надо, и продолжите, кликнув зелёную кнопку “Create Class” внизу:




- Новый класс создастся как комбинация хеадера (.h) и файла-ресурса (.cpp). Всё находится в (под)папке Source вашего проекта. Вы должны увидеть файлы в выбранной IDE:




- Заметим, что вам, возможно, надо будет перекомпилировать проект и/или перезапустить редактор после этого. Не пугайтесь некоторых возможных ошибок, ещё раз соберите IDE-проект, скомпилируйте и запустите вновь.




- Наполнение соответствующих файлов должно выглядеть так:
  - `MyMechanicalActor.h`:<code cpp>

// Fill out your copyright notice in the Description page of Project Settings.

#pragma once

#include “CoreMinimal.h” #include “MechanicalActor.h” #include “MyMechanicalActor.generated.h” UCLASS() class MY_API AMyMechanicalActor : public AMechanicalActor {

```
GENERATED_BODY()
```

}; </code>

- MyMechanicalActor.cpp:<code cpp>

// Fill out your copyright notice in the Description page of Project Settings.

#include “MyMechanicalActor.h” </code>

- Теперь вы можете перегрузить один (или несколько) Tick-методов и реализовать вашу механику, как обычно это делается в C++…
  - … в заголовочном файле:<code cpp>

void Tick(float DeltaTime) override; </code>

- … в .cpp-файле:<code cpp>

void AMyMechanicalActor::Tick(float DeltaTime) {

```
// Ваш код механики
```

} </code>

From:
http://turbanov.ru/wiki/ - **Turbopedia**

Permanent link:
**http://turbanov.ru/wiki/ru/toolworks/docs/apparatus/mechanical?rev=1623759036**

Last update: **2021/06/15 12:10**