

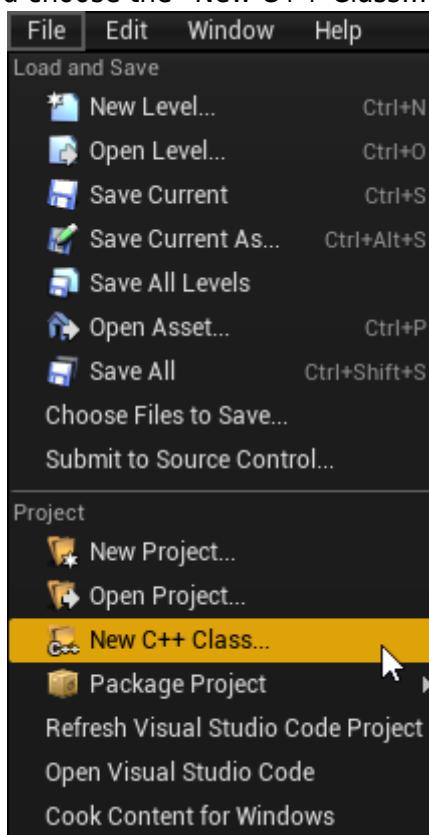
Mechanical

ECS clearly separates the data and the logic operating on that data. This logic in turn is usually executed on an iterative per-frame basis. Apparatus implements this animation-like functionality via a concept called *Mechanical*. Mechanicals are complex in nature and comprise multiple Mechanics that are executed inside of them.

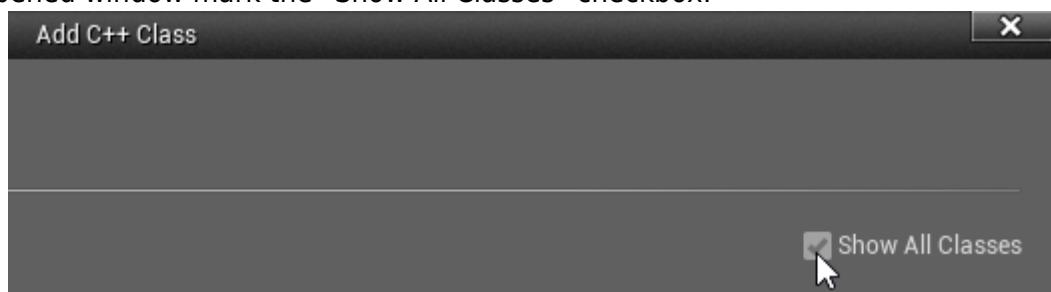
C++ Workflow

If you're going the C++ way, creating your Mechanicals goes like this.

1. Open the main UE File menu and choose the “New C++ Class...” option:



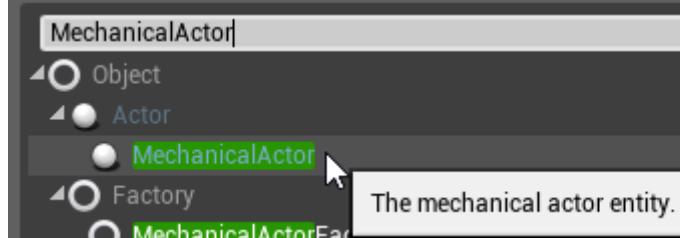
2. In the opened window mark the “Show All Classes” checkbox:



3. Now you can select any of the base classes available including the Apparatus ones. Choose the Mechanical Actor as a base class:

Choose Parent Class

This will add a C++ header and source code file to your game project.



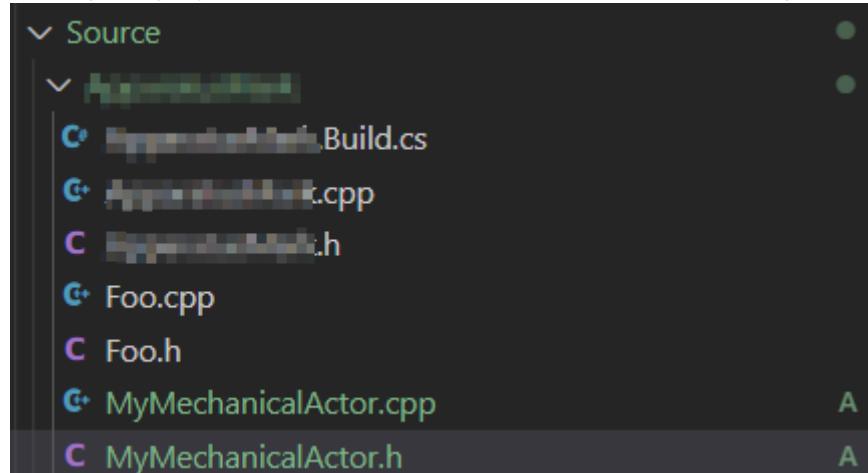
4. Click “Next” and you should see a name choosing dialog. Adjust the name of the class as needed and proceed by pressing the green “Create Class” button at the bottom:

Name Your New Mechanical Actor

Enter a name for your new class. Class names may only contain alphanumeric characters, and may not contain a space.
 When you click the “Create” button below, a header (.h) file and a source (.cpp) file will be made using this name.

Name	MyMechanicalActor	(Runtime) ▾	Public	Private
Path	.../Source/	Choose Folder		
Header File	.../Source/MyMechanicalActor.h			
Source File	.../Source/MyMechanicalActor.cpp			

5. The new class gets created as a combo of its header (.h) and a source file (.cpp). All in the Source (sub)folder of your project. You should now see them in the IDE of your choice:



6. Note that you may have to recompile and/or restart the Editor after that.

From:

<http://turbanov.ru/wiki/> - Turbopedia

Permanent link:

<http://turbanov.ru/wiki/en/toolworks/docs/apparatus/mechanical?rev=1623153574>

Last update: **2021/06/08 11:59**

