

# A real short tutorial on how to begin making a basic stage mod for Soulcalibur 6

## Requirements:

Unreal Engine 4.17.2 available from the Epic Games launcher

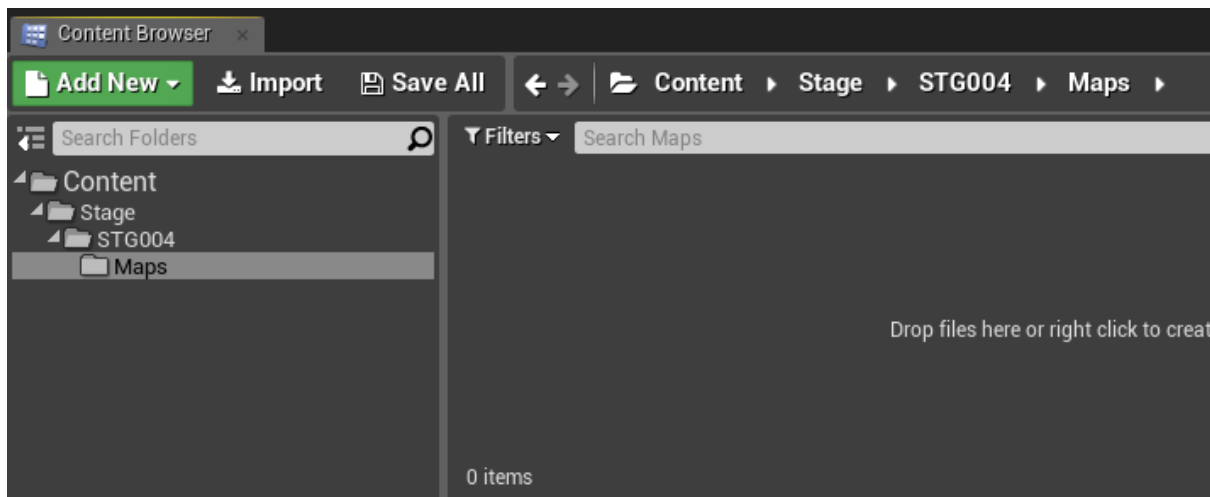
UE Viewer aka Umodel available at <https://www.gildor.org/downloads>

Any type of 3D modelling tool, examples are Blender, 3DSMax, Maya (This tutorial does not cover modelling but a note that .FBX is the optimal format for importing into unreal)

Okay so, im gonna use astral chaos as the example here as it covers the things i need to show how to replace.

so say i wanted to replace astral chaos with my own stage stuff first thing im gonna need to do is replicate the folder structure in a similar way to how we normally do texture or replacer mods

So in our case, it would be Stage > STG004 > Maps

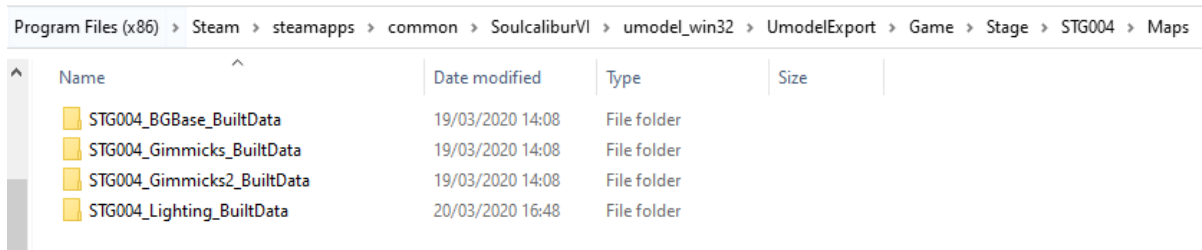


Once thats done, use Umodel and navigate to that directory we have set up here, the names of the folders within the Maps folder is important, as they're the things we're gonna use and replace to set up our own stage.

=====

This directory in particular.

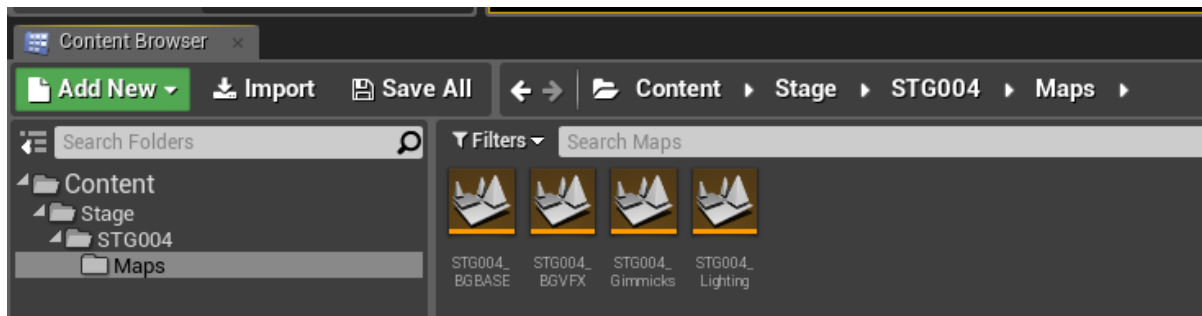
Note: STGXXX\_BGBASE with the XXX being the number ID of the stage you wish to replace



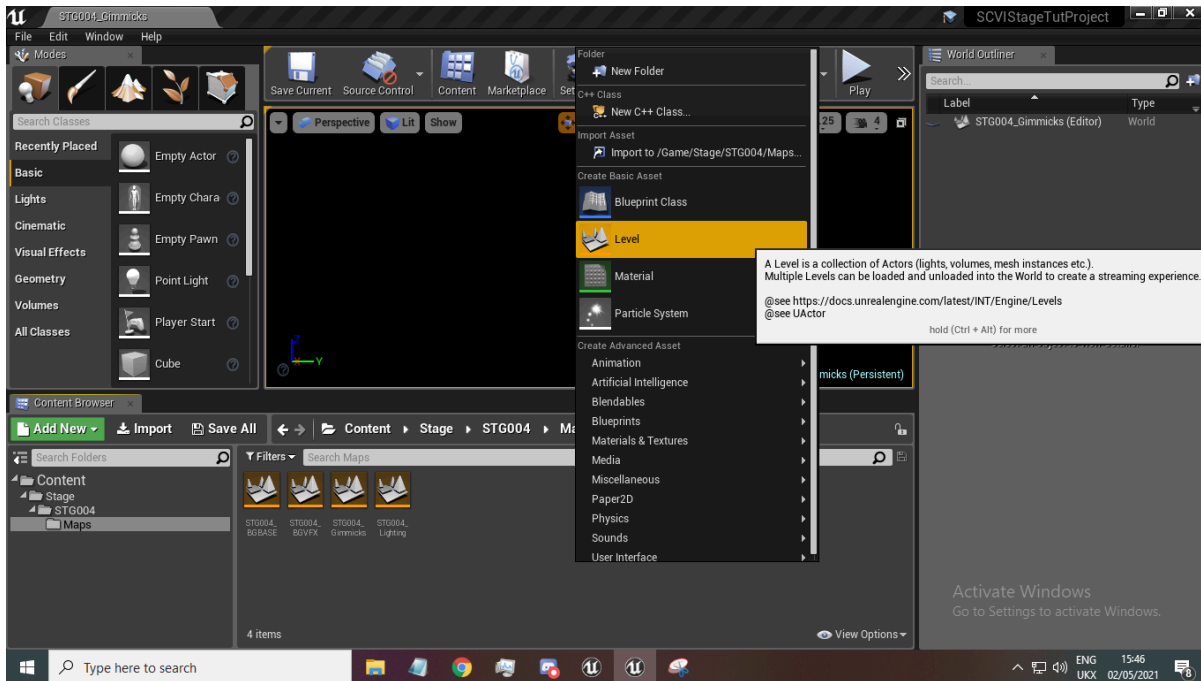
Name	Date modified	Type	Size
STG004_BGBase_BuiltData	19/03/2020 14:08	File folder	
STG004_Gimmicks_BuiltData	19/03/2020 14:08	File folder	
STG004_Gimmicks2_BuiltData	19/03/2020 14:08	File folder	
STG004_Lighting_BuiltData	20/03/2020 16:48	File folder	

we're gonna right click the active folder and create a level with those exact names / EDIT: You will also need to make a STG004\_BGVFX and STG004\_Gimmicks and STG004\_Gimmicks2

If you did this correctly, you should have this.



To create a level within UE, right click the content browser and you should get this dropdown menu.



With those done, in order to edit our stage, we're gonna be using STG004\_BGBASE which will serve as the model basis for the stage.

STG004\_BGVFX pertains to the effects in the stage, (needs more clarification, but the giant red and blue portals are alongside this if my memory serves correct)

STG004\_Gimmicks and Gimmicks2 pertain to the floating debris on the astral chaos level and the astral wall that gets broken when hit enough.

STG004\_Lighting is used if you want to override the default stage lighting with your own. If you wish to keep the original stage lighting, you don't need to make a level for it.

This also extends to the Gimmicks and Gimmicks2, if you wish to keep those, you don't need to create a level for them.

=====

With this all done, we can now open our BGBASE by doubleclicking, and effectively, we have a blank canvas that the game will load regardless in the place of the original stage.

Note that any sort of stage collision will persist, even if the model does not exist, the original walls will, this makes the stage fit for online and it won't cause a desync.

My example of a stage fitted over:



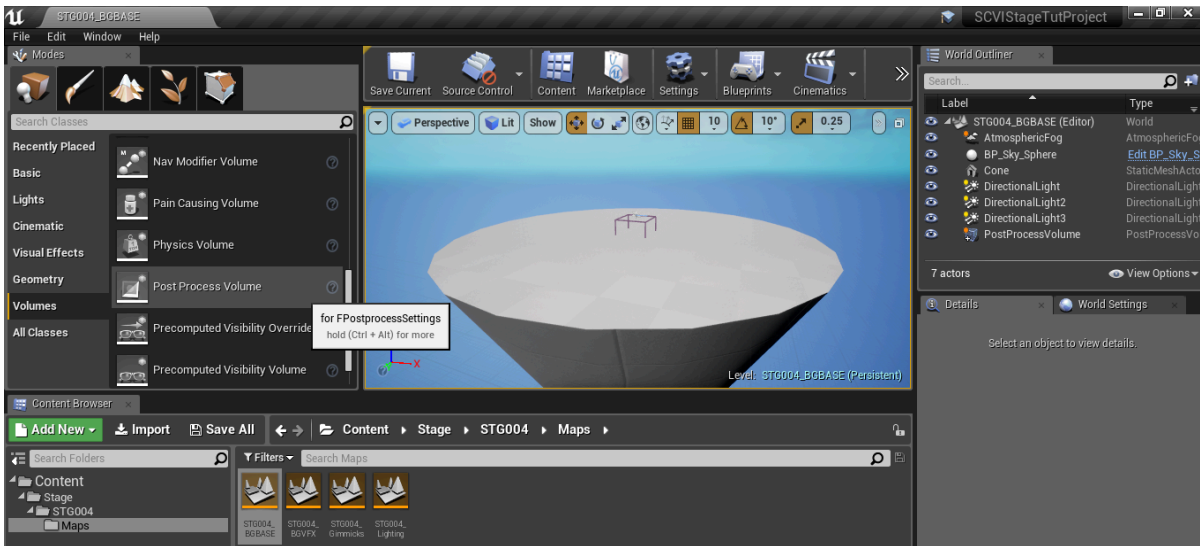
(Astral chaos replaced with a mesh from Persona 5)

=====

There are two things to note about this:

One: you will most likely have to build a stage around the original collisions, otherwise this will desync online and force a disconnect.

Two: Any unreal engine level has automatic built in motion blur, so you will need to include a post processing volume (This can be placed in any of the levels, and has options in its settings to disable this natural motion blur)



The super cool thing about bringing anything into the level, is the fact that it just loads. The picture above this one was a model with nothing renamed from its original model.

Unlike having replacer mods, you will not need to rename the assets of the models you bring in. No tedious name replacing or I forgot to name this material as it should be in the game files.

I won't go into lighting the stage but for the most part, directional lights and sky lights can help the basis of getting stuff lit up, at that point you're crossing into UE4 Level design, which is far above what i can provide.

That aside, once you're done, you create the mod in the original way, packaging the content folder, using u4pak to package and then putting your pak file into the ~mods folder.

Your mod is complete! I can't wait to see the stages people make.

=====