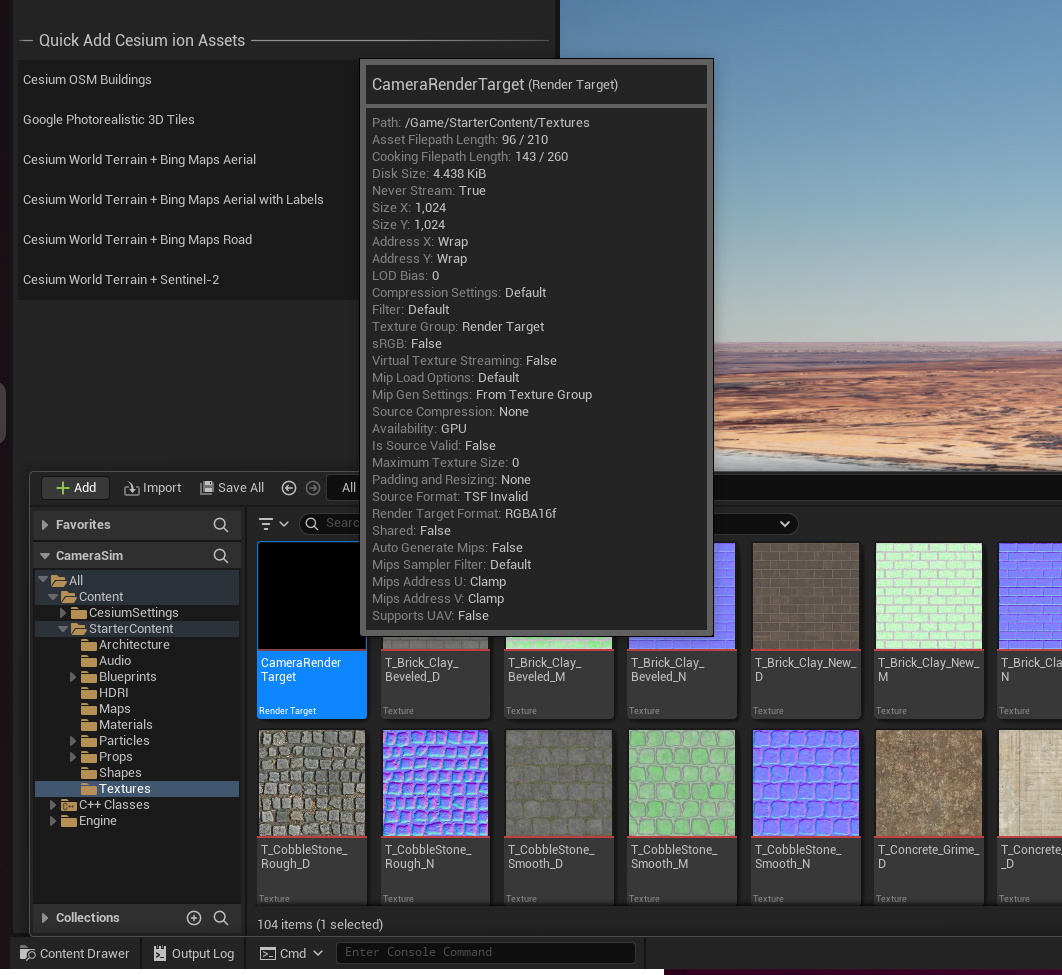
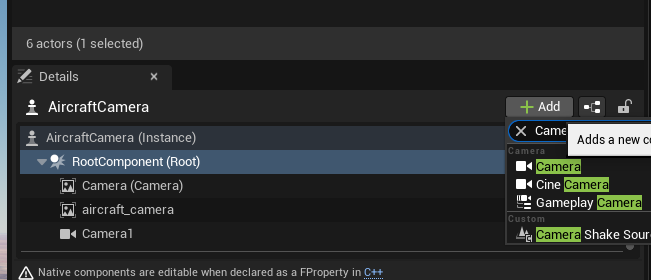
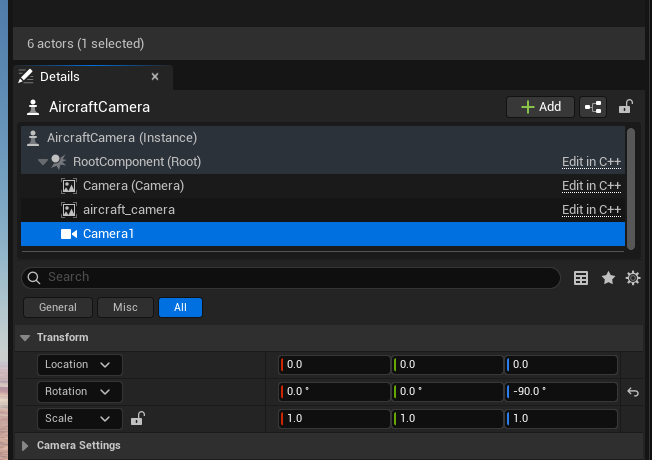
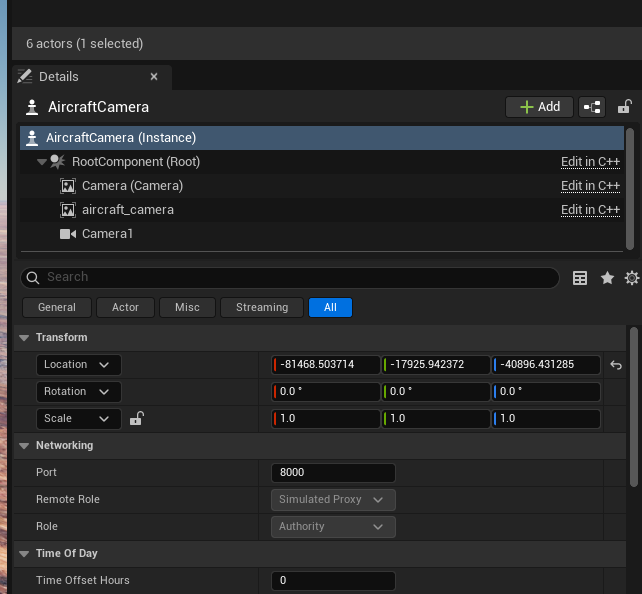
**Things to Do Once**

1. [Install Unreal Engine 5](https://dev.epicgames.com/documentation/en-us/unreal-engine/installing-unreal-engine)
2. [Download Cesium](https://github.com/CesiumGS/cesium-unreal/releases) and extract within the UE5 installation, Linux\_Unreal\_Engine\_5.5.1/Engine/Plugins/Marketplace

* Done correctly, there should be a Engine/Plugins/Marketplace/CesiumForUnreal directory
* Note that you may need to create the Marketplace Directory

1. Create a new project (blank game, c++, make sure ‘starter content’ is ticked)
2. Ensure that the Cesium Plugin is enabled (Edit -> Plugins, tick Cesium for Unreal), probably need to restart UE5 afterwar
3. [follow the guide](https://cesium.com/learn/unreal/unreal-quickstart/) to setup cesium ion account
4. Create a new Empty level, add the ‘Google Photorealistic 3D Tiles’ and ‘Cesium SunSky’, and set the origin location to the location of your flight simulation. **Note you will need to create a new level for each location that you want to test.**
5. Adjust the date and time settings so that there is light
6. Ensure that the origin height is set to 0
7. Add a TextureRenderTarget within the Unreal Engine project (in All/Content/StarterContent/Textures). Set the size of this target to 1920x1080. Call it ‘CameraRenderTarget’. Ensure the format is RGBA16f
8. Create a bogus C++ Actor (this is to get UnrealEngine to recognise that you have custom c++ code, not sure how else to do this) Tools->New c++ class -> Actor -> Call it whatever)
9. Copy “AircraftCamera.h, AircraftCamera.cpp, JSBSim.h, JSBSim.cpp, Geo.h, Geo.cpp” into the project source code directory (Project/Source/Project/\*)
10. In AircraftCamera.cpp, replace #import “CameraSim.h” with your project name eg. #import “TimProject.h”
11. In AircraftCamera.h, replace CAMERASIM\_API with your project name eg. class TIMPROJECT\_API AAircraftCamera : public APawn
12. Compile the code
13. Drag and drop ‘AircraftCamera’ into the scene
14. In the ‘AircraftCamera’ actor, add a Camera component
15. Set the rotation of **all** Camera sub-components to (0, 0, -90)
16. Set the port for the JSBSim interface to whatever is coming out of JSBSim



1. Create an empty file sudo touch /usr/share/ue5camsim.data change its user permissions to 666 (R/W)
2. Within the JSBSim aircraft model (eg. Wanderer.xml) add the following lines

<output name="127.0.0.1" type="SOCKET" protocol="UDP" port="8000" rate="30" precision="9">

<position> ON </position>

<property> position/lat-geod-deg </property>

</output>

**Things to Do Every Time**

1. Launch SITL with this JSBSim model (ensure that the UDP output has been added to the aircraft descriptor)
2. Within Unreal Engine, Press play
3. Run UnrealEngineCamera.py as an example for how to pipe the data into Python