Virtual Production with Unreal Engine

Computer Vision for Visual Effects

Professor Rich Radke

Liu Zhou

Fall 2021

Project Description

The purpose of the project is to bring the whole class to an immersive experience of the actual virtual production process. The actor(s) will stand in front of a big screen and start acting and moving. A camera will be filming behind the actor and have the screen as the background. The vision of the scene from Unreal Engine will change as the camera moves.

The whole movement synchronization of an actual camera and the scene from Unreal Engine is backed by HTC Vive systems (Vive Pro, Vive base stations, Vive tracker). A tracker is hooked to the camera, therefore, Unreal Engine will be able to detect the relative location of the camera within the range of the base station.

This project is a course activity for Computer Vision for Visual Effects taught by Professor Richard J. Radke at Rensselaer Polytechnic Institute.

Table of Content

Hardware and Software Required

Virtual Reality Devices

- HTC Vive PRO Headset (x1)+Controllers (x2)
- HTC Vive Tracker (x1)
- HTC Vive Base Stations 1.0 (x2)
- HTC Vive Link Box (x1)

Software

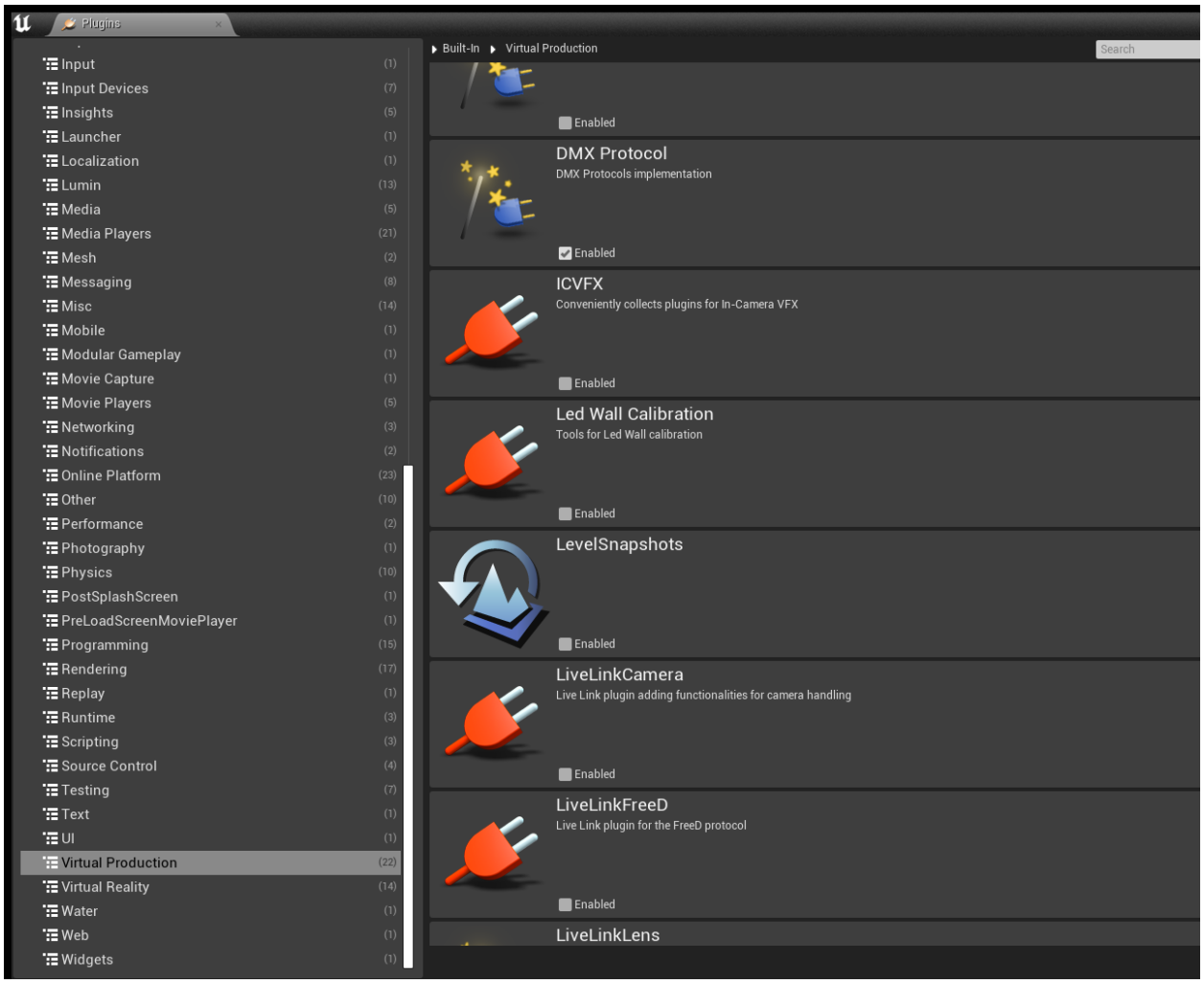
- VIVE Wireless
- Steam VR
- Unreal Engine (Current Version: 4.27)

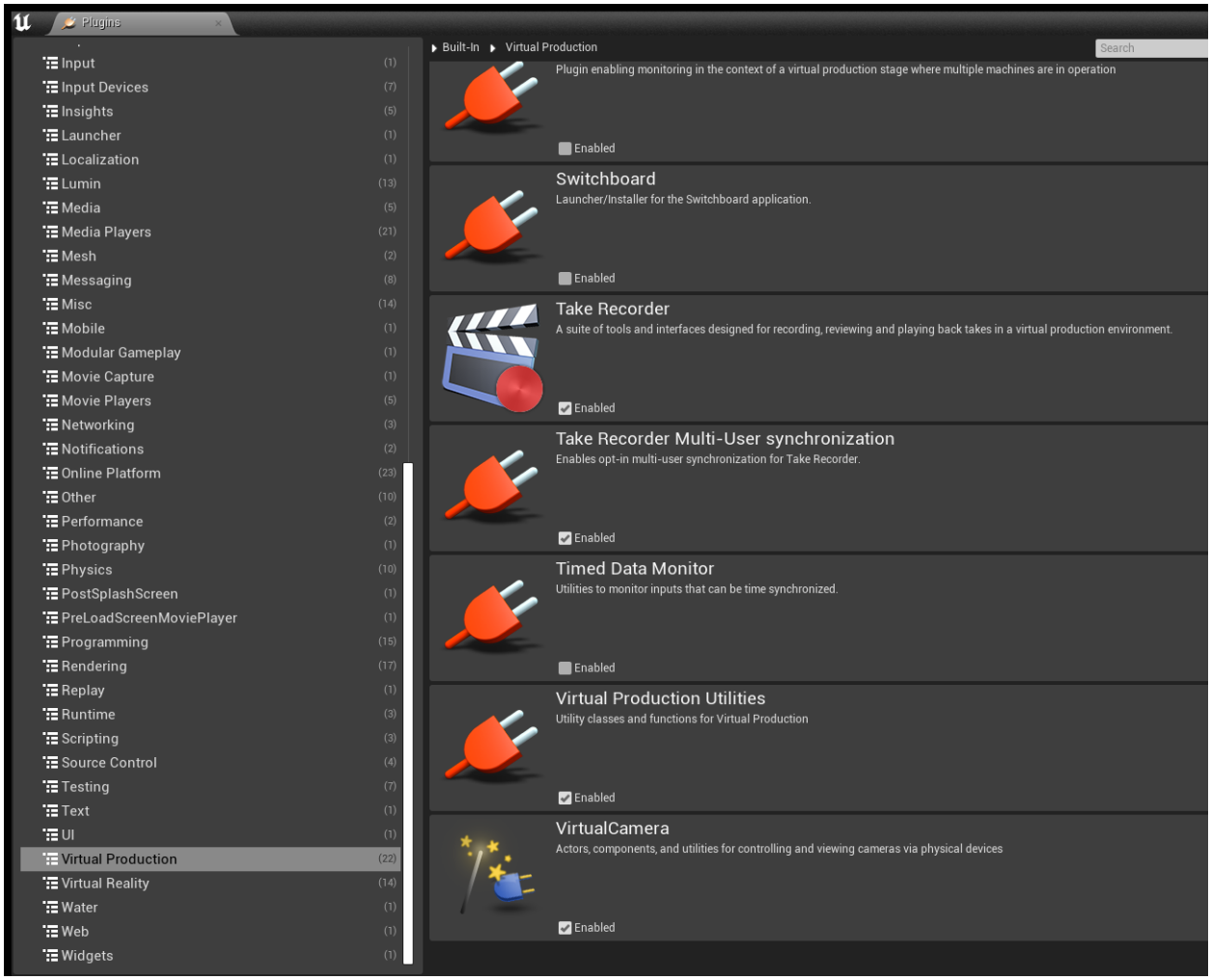
Unreal Engine

Project Setup

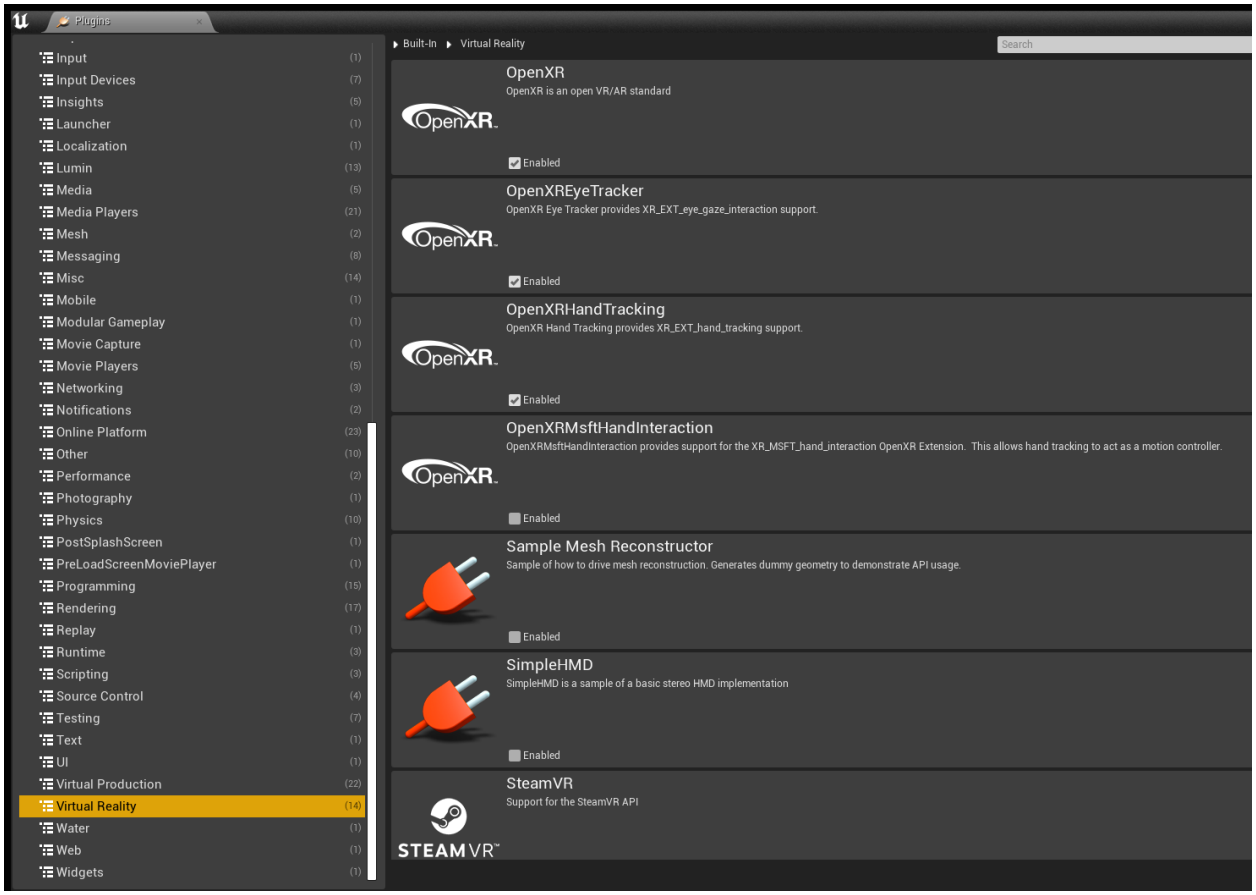
Go to Edit->Plugins

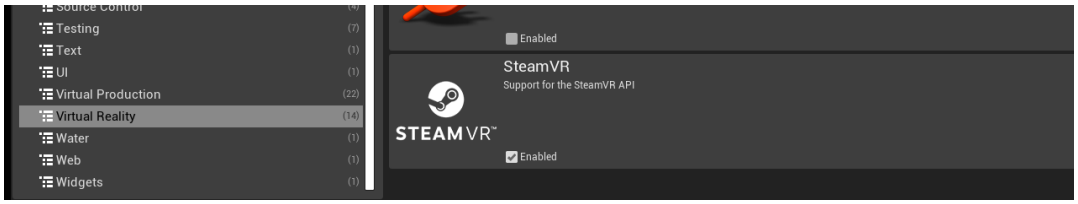
1. Enable Virtual Production Plugins




Plugins
Built-In
Virtual Production
Search
Input (1)
Input Devices (7)
Insights (5)
Launcher (1)
Localization (1)
Lumin (13)
Media (5)
Media Players (21)
Mesh (2)
Messaging (8)
Misc (14)
Mobile (1)
Modular Gameplay (1)
Movie Capture (1)
Movie Players (5)
Networking (3)
Notifications (2)
Online Platform (23)
Other (10)
Performance (2)
Photography (1)
Physics (10)
PostSplashScreen (1)
PreLoadScreenMoviePlayer (1)
Programming (15)
Rendering (17)
Replay (1)
Runtime (3)
Scripting (3)
Source Control (4)
Testing (7)
Text (1)
UI (1)
Virtual Production (22)
Virtual Reality (14)
Water (1)
Web (1)
Widgets (1)
Plugin enabling monitoring in the context of a virtual production stage where multiple machines are in operation
Enabled
Switchboard
Launcher/Installer for the Switchboard application.
Enabled
Take Recorder
A suite of tools and interfaces designed for recording, reviewing and playing back takes in a virtual production environment.
Enabled
Take Recorder Multi-User synchronization
Enables opt-in multi-user synchronization for Take Recorder.
Enabled
Timed Data Monitor
Utilities to monitor inputs that can be time synchronized.
Enabled
Virtual Production Utilities
Utility classes and functions for Virtual Production
Enabled
VirtualCamera
Actors, components, and utilities for controlling and viewing cameras via physical devices
Enabled

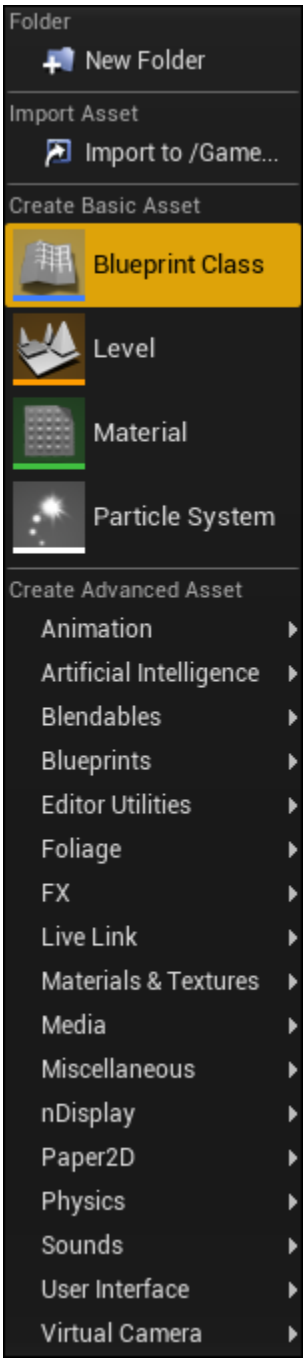
2. Enable Virtual Reality Plugins



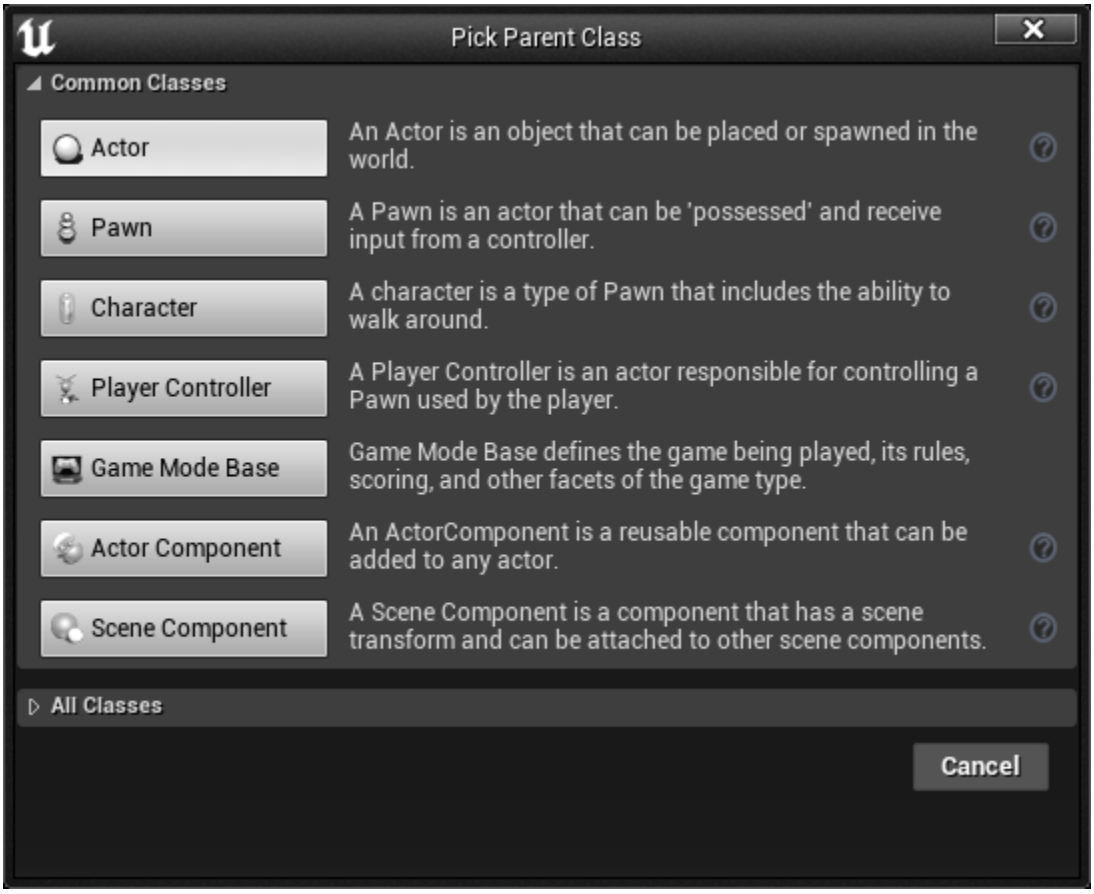


Create a Virtual Camera

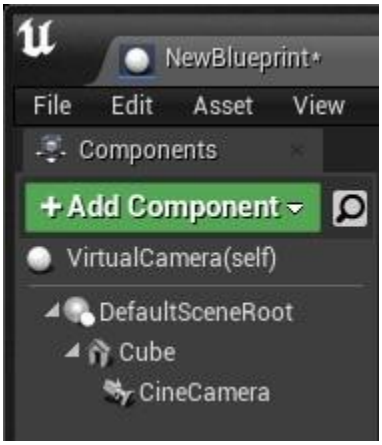
1. Create an “actor”
    a. Right-click inside of the box of Content Browser, Select Blueprint Class



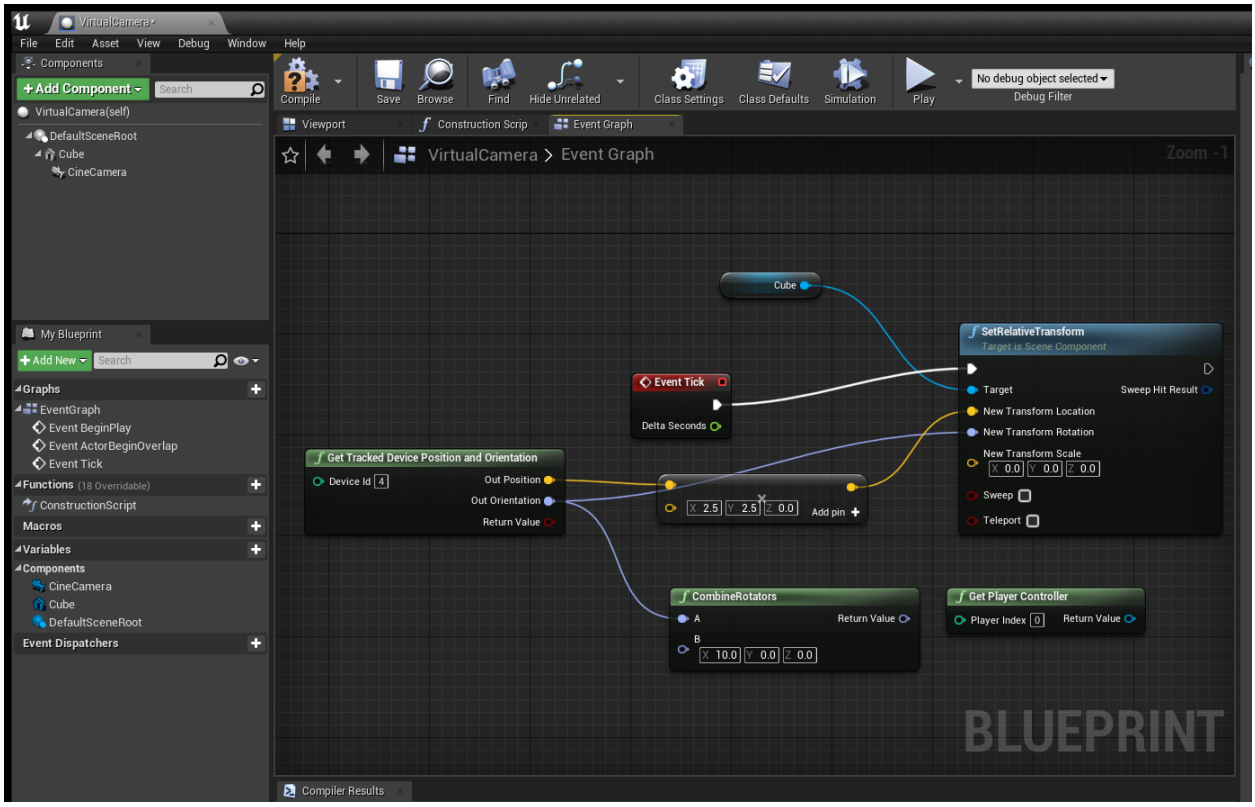
b. Pick Parent Class->Actor



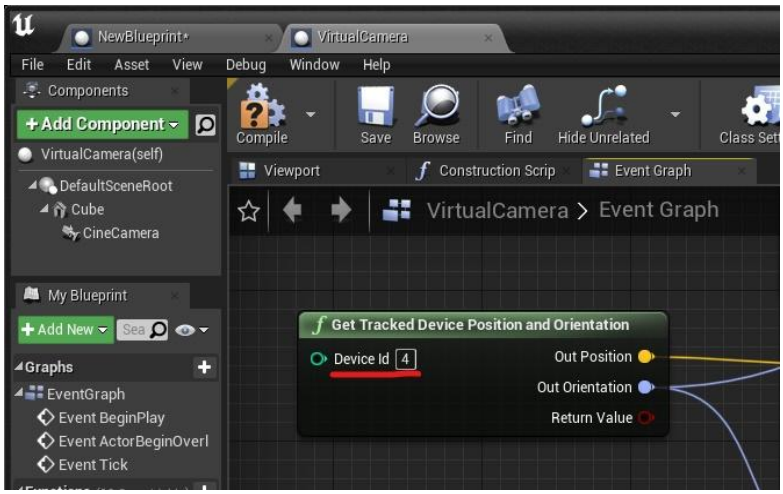
c. Name the project "VirtualCamera" (or something else).
d. Double-click the object to open the blueprint.
e. Add Component->type in "Cube", inside the cube create a cine camera. The nested structure is shown below.



2. VirtualCamera Blueprint (DO NOT FORGET TO COMPILE!)

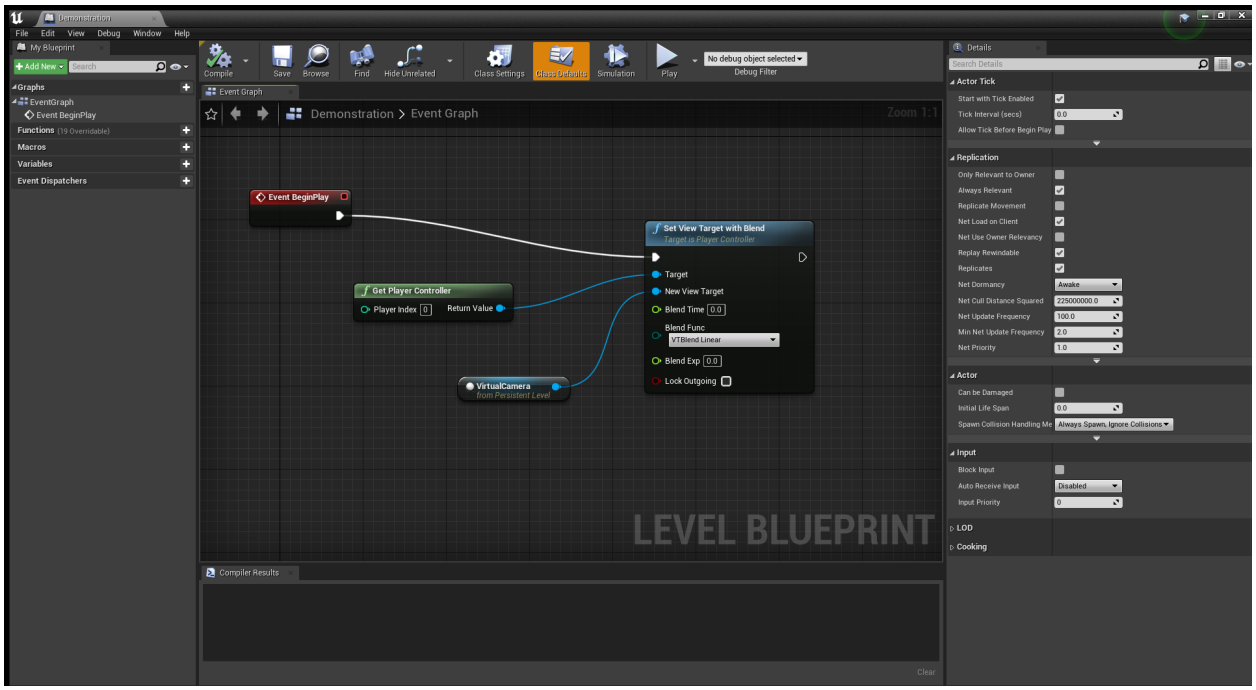


3. Connect VIVE Tracker to the Virtual Camera “actor”
   Modify the number in Device Id.
   Note: The actual Id is the number from the order of how each device is connected to SteamVR. The first device connected is index 0, the second device is index 1, and so on… Also, the set of a headset(x1) and controllers(x2) are counted separately, therefore, we have 3 devices in this set.



Add the Virtual Camera into a Scene

1. Drag the Virtual Camera “actor” from Content Browser into the scene, then adjust the position.
2. Go to Blueprints->Open Level Blueprint (DO NOT FORGET TO COMPILE!)



Normally, nothing else needs to be modified.

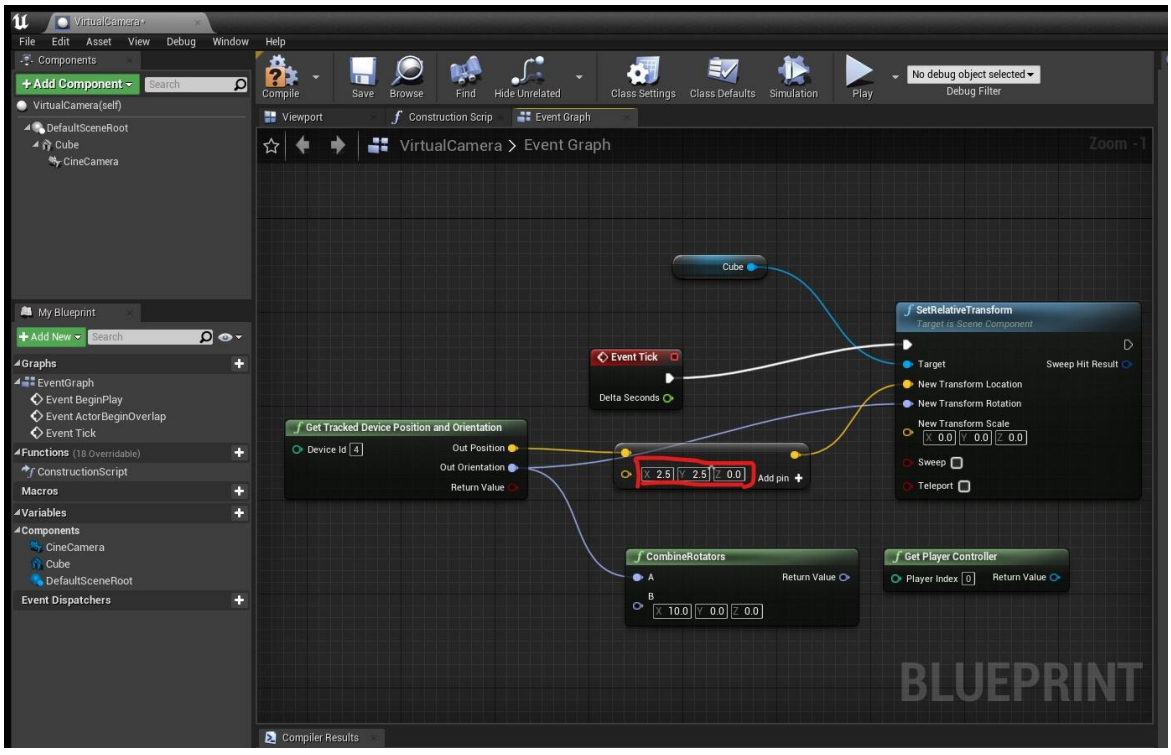
Calibration

VIVE Tracker

Make sure the tracker stays in the below position.

Unreal Engine

Because we are unsure of the scale of a scene. Therefore, we need to adjust the speed of the camera every time we add a virtual camera to a new scene.
Open Virtual Camera Blueprint, and adjust the numbers inside of the marked area (DO NOT FORGET TO COMPILE!):



Troubleshooting

Q1: SteamVR cannot detect VR devices.

Solution:

1) Make sure devices are within the range of the beacons.
2) Redo pairing.

Q2: Unreal Engine cannot detect VR devices.

Solution:

1) Restart every device and complete solution #4 from Q3.

Q3: The scene freezes or goes back to the default view.

Solution:

1) Check if anything blocked the VIVE Link Box.
2) Check device connections.
3) Check if the tracker is out of range.
4) Redo “room setting” on Steam VR.
5) Try different Device Id (Double click Virtual Camera “actor”-> Node “Get Tracked Device Position and Orientation”->” Device Id”)

Reference

Creating a Virtual Camera using HTC Vive in Unreal Engine (4.26) [Video file]. (2020, December 22). In *Creating a Virtual Camera Using HTC Vive in Unreal Engine (4.26)*. Retrieved December 12, 2021, from https://www.youtube.com/watch?v=2W1XuOkN6Kk