Streaming Portal Levels Plugin for Unreal Engine 5.2+

Introduction

Video: UE5.2 Streaming Portal Levels - Physics Ball Run

Demo Link: https://teekrugames.com/SPL/StreamingPortalLevels.zip

What are Streaming Portal Levels?

Streaming Portals is a stand alone plugin system to stream levels in and out while using a portal to provide near seamless transition between areas designed to be used with 3rd person view. The system has generic actor support which could be used for first person view but there are a number of excellent first person view solutions out there and our own project is focused on 3rd person.

Think older games, especially 2D. You move around on a map and stand over an icon of a town to enter it. This is the same idea, but in 3D. Your main level map could just have building exteriors only, (dungeons, magic portals, taverns, <insert your own here>. Etc ...) while interiors are streaming levels using portals to walk through into them.

Why was this created?

This is a feature we're using in our own project A.L.A.R.M. and we need to polish each piece of game tech created before release. This is a great way to battle-harden code and make it more generic and reusable.

You said near seamless?

Lumen is a challenge to deal with, but fun. The demo that comes with this plugin shows ways to deal with Lumen (visual noise) transitions between levels. You could also use Post Process volumes to fade out and in as the camera is teleported between levels.

Why not 5.0 or 5.1?

5.0 Scene capture using Lumen was dark and difficult to match. Performance was good, visuals were not.
5.1 has an assert that fires when trying to use Scene capture Component with Lumen.
5.2 Lumen scene capture components look very good and performance in general in 5.2 is awesome.

Does this support World Partition?

Yes. If the Portals detect they are in a world partition level they will look for a world partition streaming source component on themselves. One has already been placed on the default BP_PortalActor and is disabled by default. When the portal renders the scene, ii will turn on it's streaming source.

Why the demo?

Below is a link to a video of our own project using this system. Our levels have lots of set dressing and our characters moderately complex. Performance is good.

The demo dresses up the scenes in such a way, but still a light way, to game-a-fy the demo, creating visual noise (all the stuff in the scenes) to help transition between levels. If you're not using Lumen, stark scenes can be seamless, but with Lumen, make that light bounce!

Why C++?

It's easier to do timing and bind to events within the engine to keep the illusion of passing between worlds smooth. Each C++ class has a corresponding blueprint in the plugins Content folder with examples of how to create a "portal twin" and call backs. This way you can edit this blueprint directly or create a child blueprint and customize it to your liking.

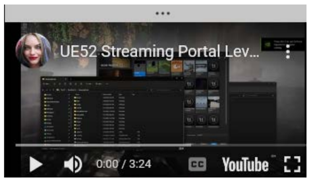
There are 7 classes. One class is just a helper to expose setting Gamma (SP_BlueprintLibrary).

C++	Blueprint
Actors/SP_PortalActor	Content/Blueprints/Actor/BP_PortalActor
Actors/SP_PortalHub	Content/Blueprints/Actor/BP_PortalHub
Actors/SP_PortalLevelStreamManager	Content/Blueprints/Actor/BP_PortalLevelStreamer
Components/SP_PortalActorComponent	Content/Blueprints/Component/AC_PortalActor
Components/SP_PortalCamera ManagerComponent	Content/Blueprints/Component/AC_PortalCameraManager
Components/SP_PortalPlayerComponent	Content/Blueprints/Component/AC_PortalPlayer
Extra/SP_BlueprintLibrary	N/A

Why a section on performance?

We believe for a good portal experience, you need good performance. We've taken what we've learned from our own project, and projects like Lyra Starter Game, and put together a basic place to start improving performance with ranging from settings within the .ini to blueprints that set and test for performance. The project is also set up to generate PSO (Persistent Shader Cache). This is to avoid hitching or chunking in the game. Once generated the levels can be streamed in almost seamlessly (Depending on how you handle your begin play and numbers of objects within the level. But all that is workable and will be covered in the following sections). How deep you want to go is up to you.
We are no experts, but we dive as deep as we need to in order to learn what is required to achieve the goal. If you get into any problems or stuck on how to handle streaming levels, we're here to help!

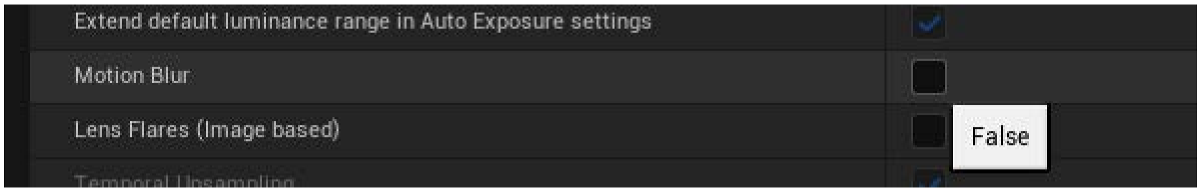
Quick Start



Video Tutorial: UE5.2 Streaming Portal Levels - Tutorial 1: Quick Start

The streaming portals are close to plug and play. We don't like a lot of overhead in setting things up, so with keeping that in mind hopefully this will be fairly easy.

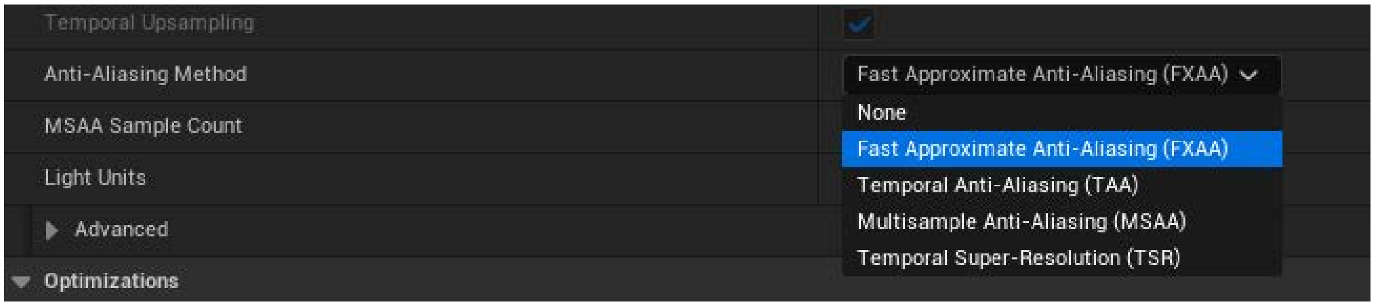
1. Start with the default 3rd Person template in Unreal 5.2
2. Copy the StreamingPortal plugin from the demo in the Plugins folder
3. Create Plugins folder in new project and paste the StreamPortal plugin.
4. Start Project and come to default level. Three settings must be changed in order for the portals to work and look good.
5. First in Project Settings --> Rendering --> Default Settings and disable Motion Blur (r.DefaultFeature.MotionBlur=False), though the scene capture components do not use motion blur, motion blur will happen on the portals and not look great.

   a. 

   Extend default luminance range in Auto Exposure settings ✓
   Motion Blur ☐
   Lens Flares (Image based) ☐ False

6. The second setting to change is the anti aliasing method.

   a.

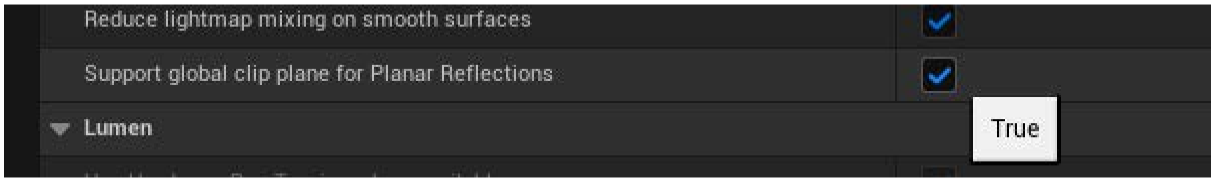


Temporal Upsampling ✓
Anti-Aliasing Method: Fast Approximate Anti-Aliasing (FXAA)
MSAA Sample Count
Light Units
Advanced
Optimizations

None
Fast Approximate Anti-Aliasing (FXAA)
Temporal Anti-Aliasing (TAA)
Multisample Anti-Aliasing (MSAA)
Temporal Super-Resolution (TSR)

   b. Fast Approximate Anti-Aliasing (FXAA or r.AntiAliasingMethod=1) works well with portals
   c. Multisampling Anti-Aliasing (MSAA or r.AntiAliasingMethod=3) works well
   d. Temporal Super-Resolution (TSR default or r.AntiAliasingMethod=4) only looks good if you are using an UpSampler (AMD FX1,2, Nvidia DLSS or Intel XeSS) otherwise can create render artifacts in the portal images.

7. Lastly and most importantly Support for Global Clip Plane for Planar Reflections or the portals cannot clip properly and will create weird images.. (r.AllowGlobalClipPlane=True)

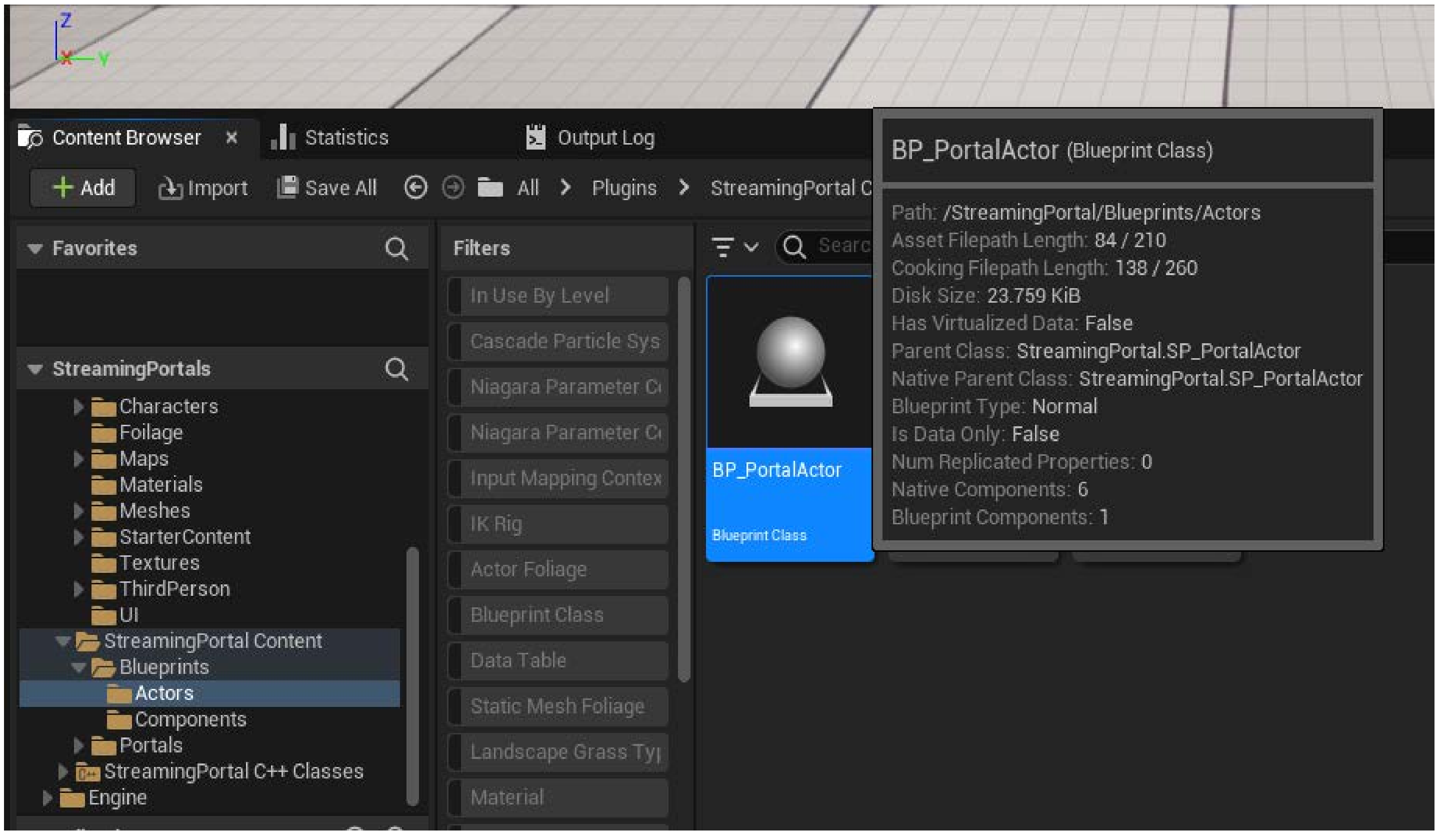
   a. 

   Reduce lightmap mixing on smooth surfaces ✓
   Support global clip plane for Planar Reflections ✓ True
   Lumen

8. This will require a restart and recompile of some shaders. So restart and wait for the shader compilation to finish.

9. Now you can add a portal into the scene by dragging the BP_PortalActor into the scene from the Plugin's content folder.

   a.



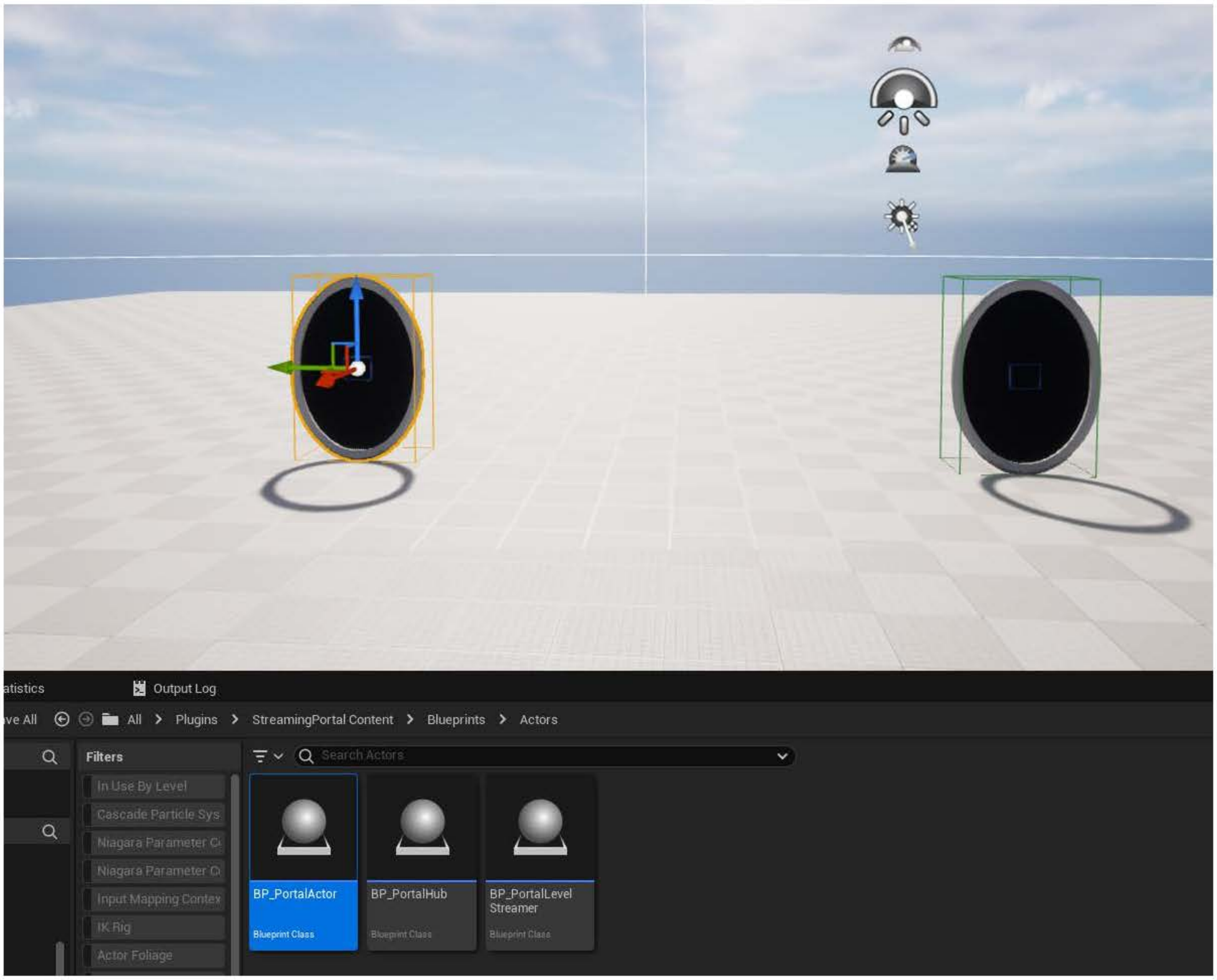
BP_PortalActor (Blueprint Class)

Path: /StreamingPortal/Blueprints/Actors
Asset Filepath Length: 84 / 210
Cooking Filepath Length: 138 / 260
Disk Size: 23.759 KiB
Has Virtualized Data: False
Parent Class: StreamingPortal.SP_PortalActor
Native Parent Class: StreamingPortal.SP_PortalActor
Blueprint Type: Normal
Is Data Only: False
Num Replicated Properties: 0
Native Components: 6
Blueprint Components: 1

   b. Drag the actor into the default scene or create a new blank one your choice.

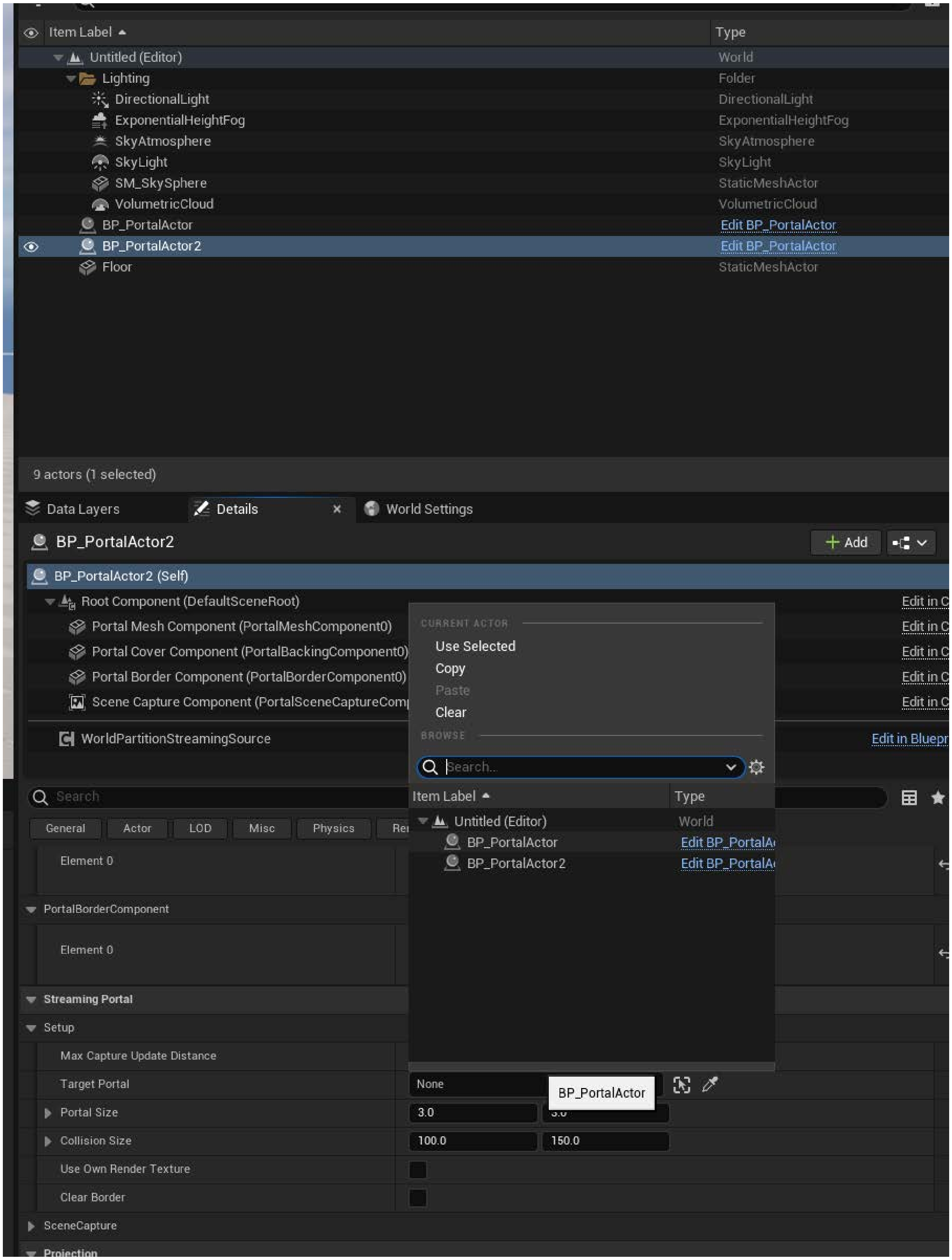
10. Position the portal so it's above the ground. Then hold down ALT and drag to duplicate the actor. Move it at least 1500 units away. This is the default range of the portals will render at. They share the same render texture (Unless told other wise), so only one should be active at a time.

a.


atistics
Output Log
ve All
All
Plugins
StreamingPortal Content
Blueprints
Actors
Filters
Search Actors
In Use By Level
Cascade Particle Sys
Niagara Parameter C
Niagara Parameter C
Input Mapping Contex
IK Rig
Actor Foliage
BP_PortalActor
Blueprint Class
BP_PortalHub
Blueprint Class
BP_PortalLevel Streamer
Blueprint Class

11. Now just set the Target Portal in each Actor to the other Actor.

a.



Item Label	Type
Untitled (Editor)	World
Lighting	Folder
DirectionalLight	DirectionalLight
ExponentialHeightFog	ExponentialHeightFog
SkyAtmosphere	SkyAtmosphere
SkyLight	SkyLight
SM_SkySphere	StaticMeshActor
VolumetricCloud	VolumetricCloud
BP_PortalActor	Edit BP_PortalActor
BP_PortalActor2	Edit BP_PortalActor
Floor	StaticMeshActor

9 actors (1 selected)

BP_PortalActor2

BP_PortalActor2 (Self)
Root Component (DefaultSceneRoot)
Portal Mesh Component (PortalMeshComponent0)
Portal Cover Component (PortalBackingComponent0)
Portal Border Component (PortalBorderComponent0)
Scene Capture Component (PortalSceneCaptureComp
WorldPartitionStreamingSource

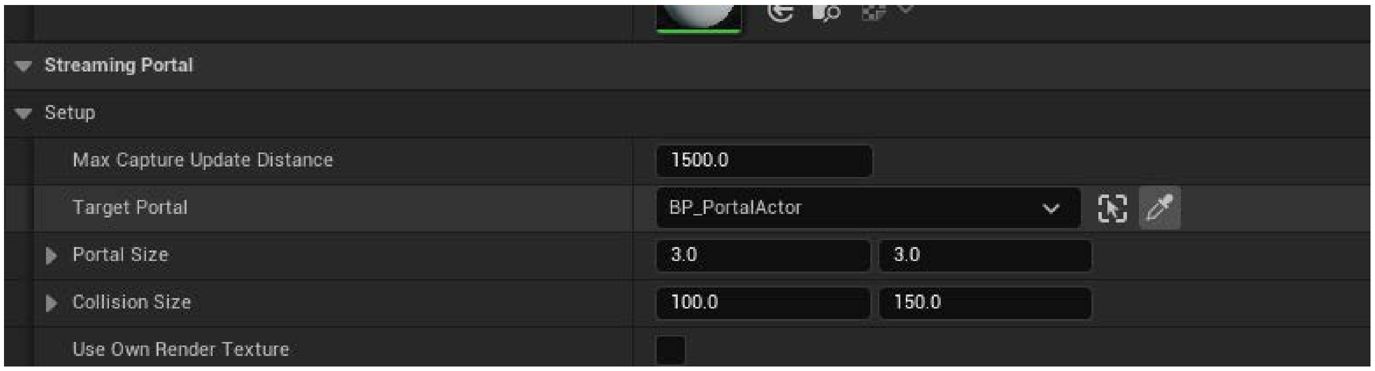
Item Label	Type
Untitled (Editor)	World
BP_PortalActor	Edit BP_PortalA
BP_PortalActor2	Edit BP_PortalA

PortalBorderComponent
Element 0
Streaming Portal
Setup

Max Capture Update Distance	
Target Portal	None
Portal Size	3.0
Collision Size	100.0 / 150.0
Use Own Render Texture	
Clear Border	

b. Actor should have a target to Actor2 and Actor2's target is Actor

c.



Streaming Portal

Setup

Max Capture Update Distance 1500.0

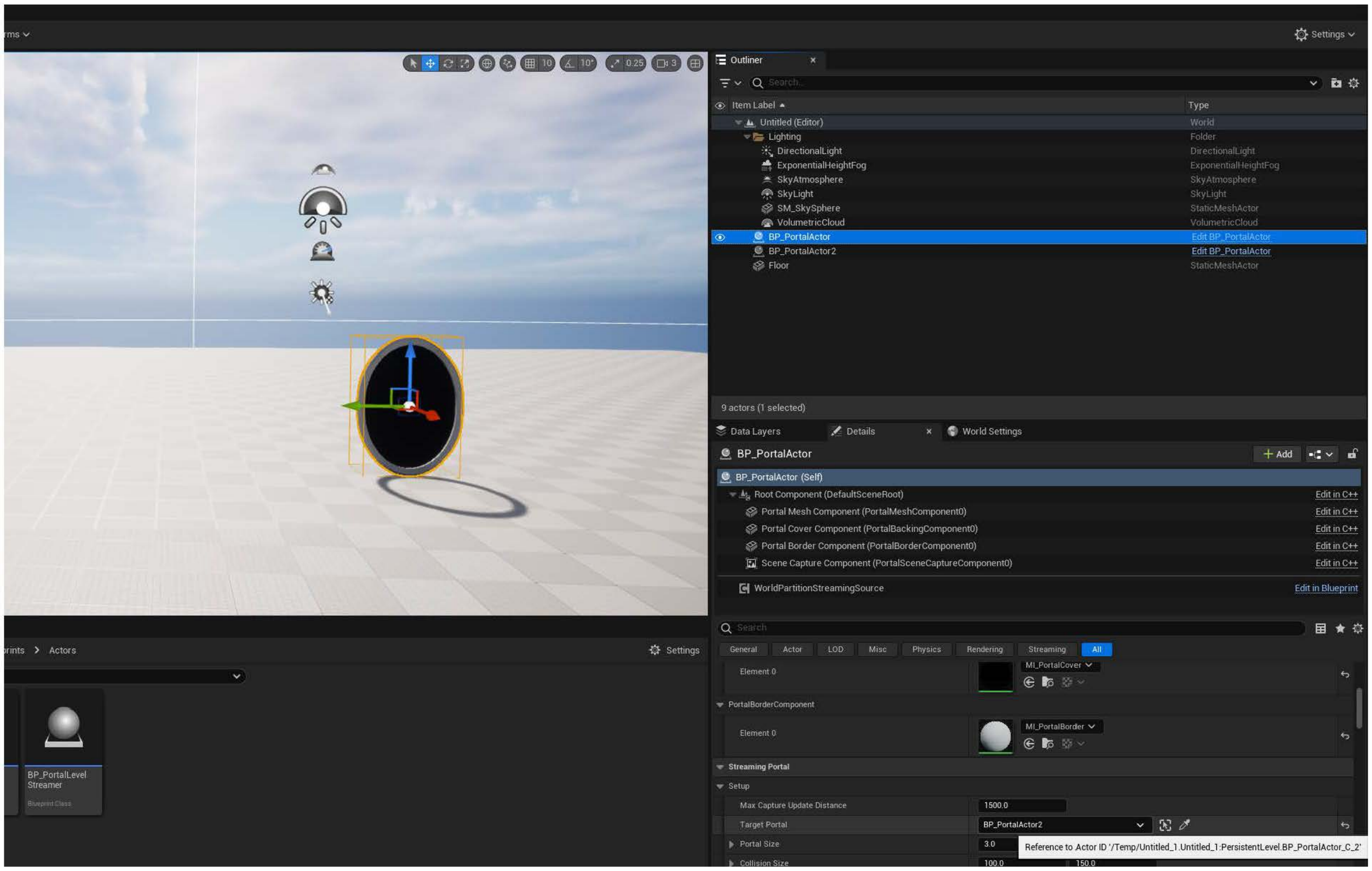
Target Portal BP_PortalActor

Portal Size 3.0 3.0

Collision Size 100.0 150.0

Use Own Render Texture

d.



Item Label	Type
Untitled (Editor)	World
Lighting	Folder
DirectionalLight	DirectionalLight
ExponentialHeightFog	ExponentialHeightFog
SkyAtmosphere	SkyAtmosphere
SkyLight	SkyLight
SM_SkySphere	StaticMeshActor
VolumetricCloud	VolumetricCloud
BP_PortalActor	Edit BP_PortalActor
BP_PortalActor2	Edit BP_PortalActor
Floor	StaticMeshActor

9 actors (1 selected)

BP_PortalActor

- BP_PortalActor (Self)
  - Root Component (DefaultSceneRoot) — Edit in C++
    - Portal Mesh Component (PortalMeshComponent0) — Edit in C++
    - Portal Cover Component (PortalBackingComponent0) — Edit in C++
    - Portal Border Component (PortalBorderComponent0) — Edit in C++
    - Scene Capture Component (PortalSceneCaptureComponent0) — Edit in C++
  - WorldPartitionStreamingSource — Edit in Blueprint

Element 0: MI_PortalCover

PortalBorderComponent
Element 0: MI_PortalBorder

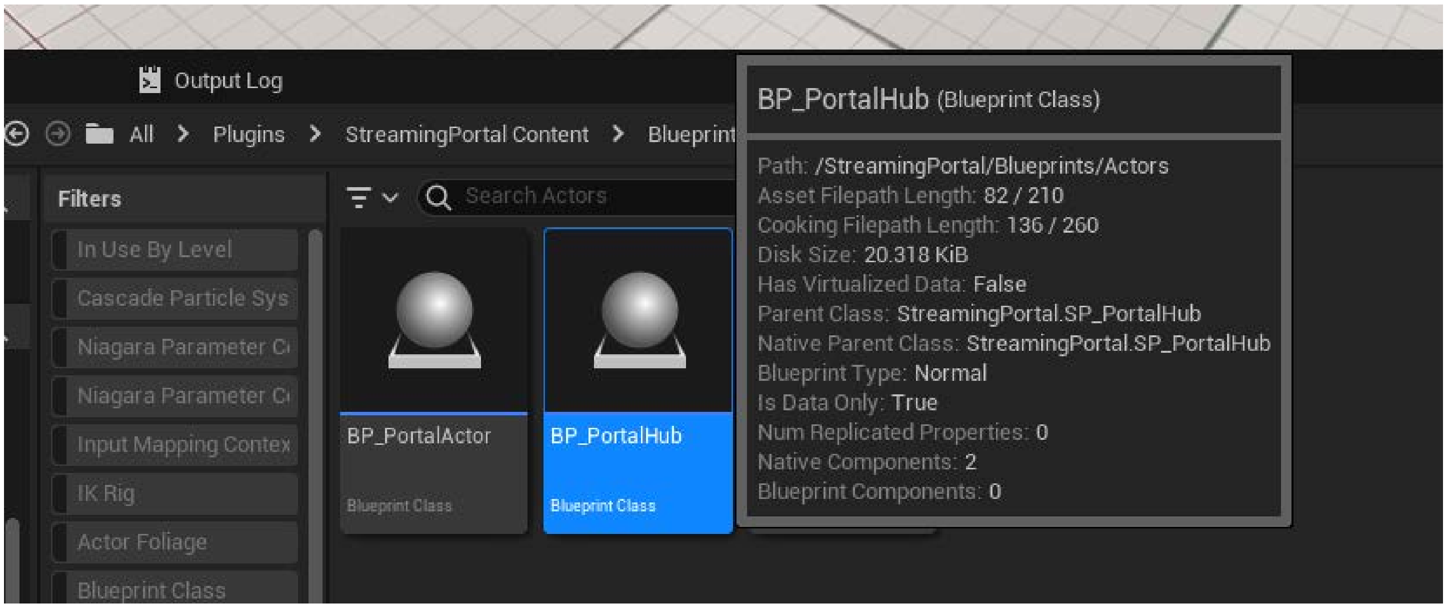
Streaming Portal
Setup
Max Capture Update Distance: 1500.0
Target Portal: BP_PortalActor2
Portal Size: 3.0
Collision Size: 100.0 150.0

Reference to Actor ID '/Temp/Untitled_1.Untitled_1:PersistentLevel.BP_PortalActor_C_2'

12. Press Play and run through the portal.

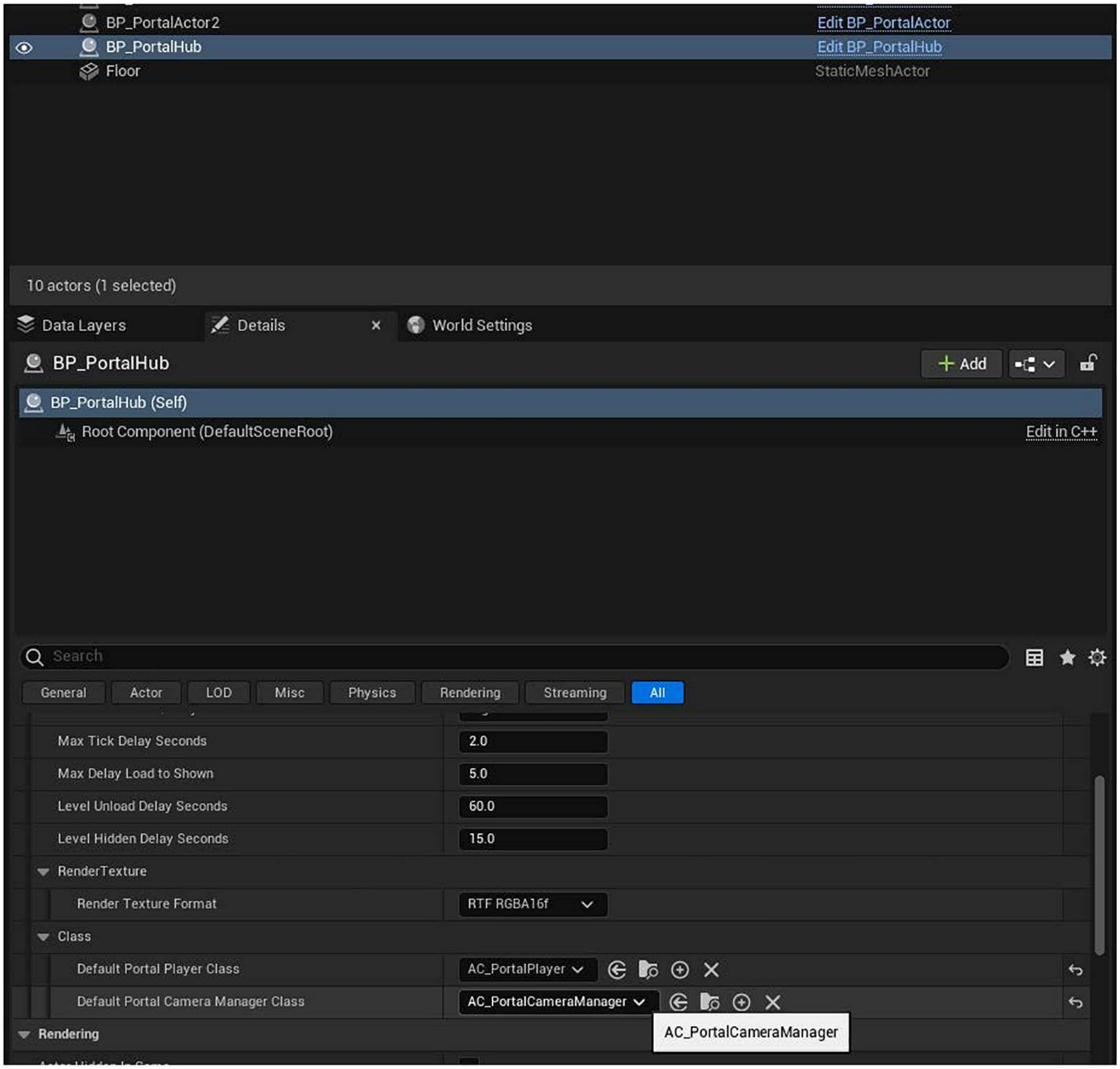
The Portal Hub

By default the system will spawn a PortalHub and also spawn the components and attach them to the Player and Player Camera Manager. This will spawn the C++ versions. In order to use the Blueprint versions drag a PortalHub into the scene and set it's default classes to use the blueprints.



BP_PortalHub (Blueprint Class)

Path: /StreamingPortal/Blueprints/Actors
Asset Filepath Length: 82 / 210
Cooking Filepath Length: 136 / 260
Disk Size: 20.318 KiB
Has Virtualized Data: False
Parent Class: StreamingPortal.SP_PortalHub
Native Parent Class: StreamingPortal.SP_PortalHub
Blueprint Type: Normal
Is Data Only: True
Num Replicated Properties: 0
Native Components: 2
Blueprint Components: 0


BP_PortalActor2 | Edit BP_PortalActor
BP_PortalHub | Edit BP_PortalHub
Floor | StaticMeshActor

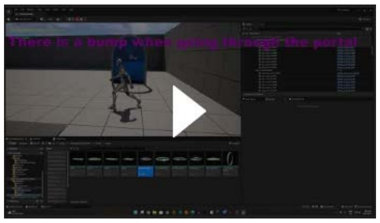
10 actors (1 selected)

Max Tick Delay Seconds	2.0
Max Delay Load to Shown	5.0
Level Unload Delay Seconds	60.0
Level Hidden Delay Seconds	15.0
RenderTexture	
Render Texture Format	RTF RGBA16f
Class	
Default Portal Player Class	AC_PortalPlayer
Default Portal Camera Manager Class	AC_PortalCameraManager
Rendering	

Here you can tweak other settings for the portals such as render texture format and timing for streaming levels in and out.

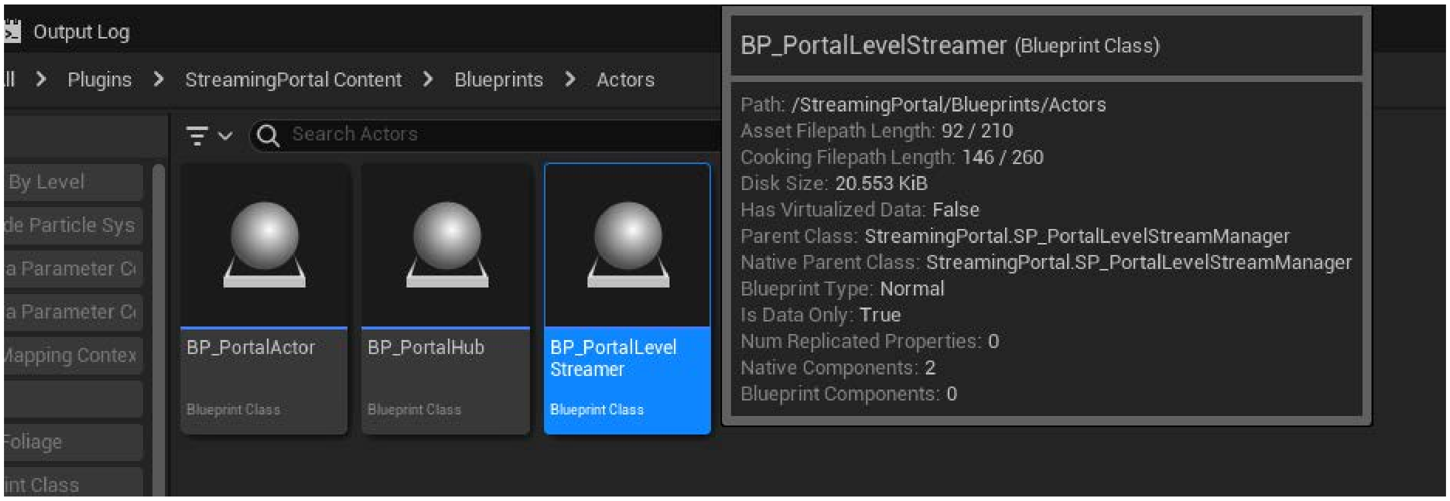
Streaming a Level with the Portals

Video Tutorial: UE5.2 Streaming Portal Levels - Tutorial 2: Stream a Level



Assuming you've gone through the Quick Start section and placed some portals in your level, you want to stream a level in.

First you need a Portal Level Streamer.

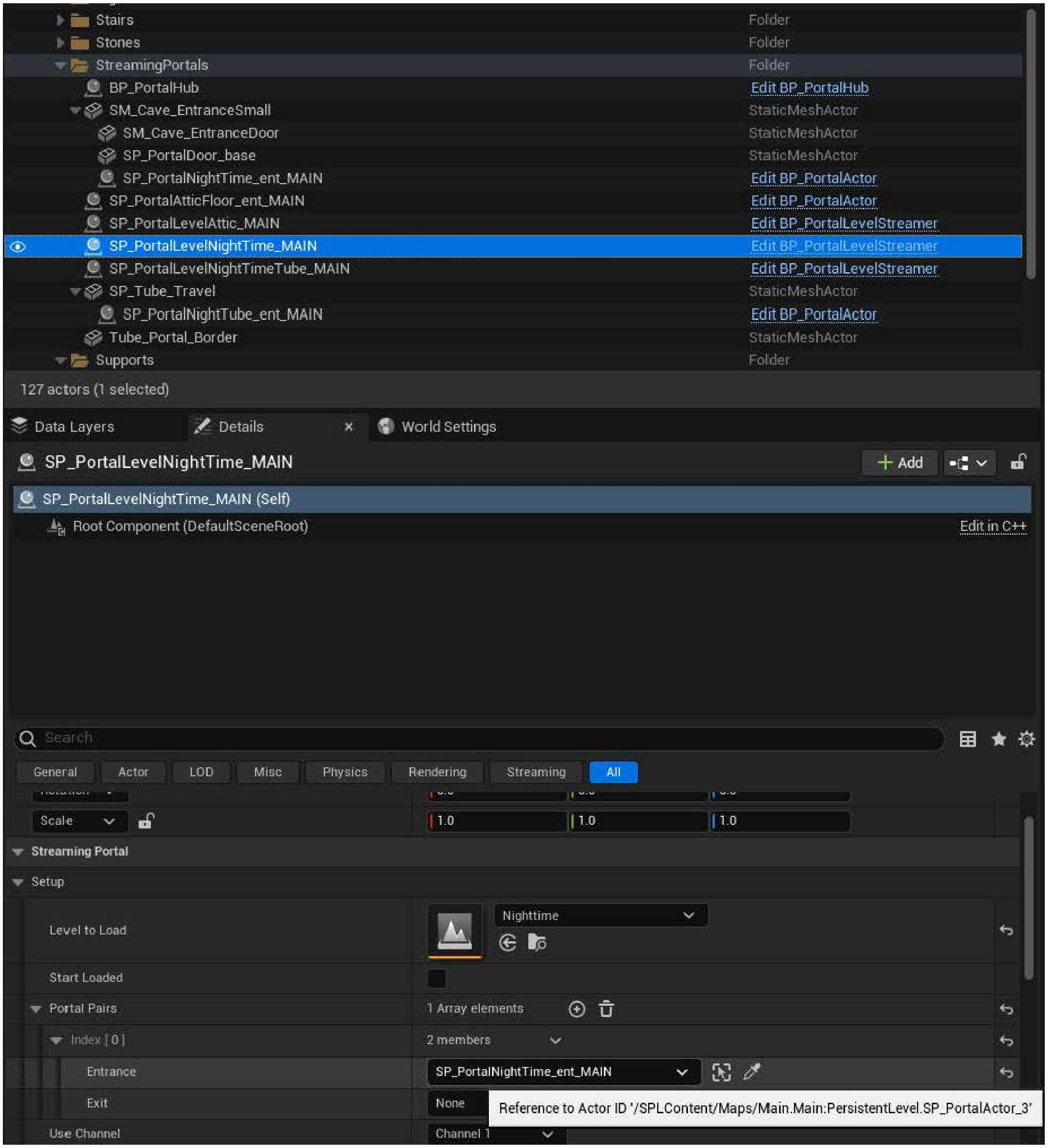


BP_PortalLevelStreamer (Blueprint Class)

Path: /StreamingPortal/Blueprints/Actors
Asset Filepath Length: 92 / 210
Cooking Filepath Length: 146 / 260
Disk Size: 20.553 KiB
Has Virtualized Data: False
Parent Class: StreamingPortal.SP_PortalLevelStreamManager
Native Parent Class: StreamingPortal.SP_PortalLevelStreamManager
Blueprint Type: Normal
Is Data Only: True
Num Replicated Properties: 0
Native Components: 2
Blueprint Components: 0

The Portal Level Streamer has a concept of Portal Pairs (An Entrance and Exit), a soft reference to the level its to manage and the light channels used when switching between levels.

Portal Pairs consist of an Entrance and Exit. The Entrance exists on the same level the Portal Level Streamer does and the exit is always in the streamed level. The two portals are tied together by the Actor Tag. So if you put a tag of "Main" on the Portal Actor in the Persistent Level, then you must put the same tag on the "Exit" portal in the streamed level.



Stairs	Folder
Stones	Folder
StreamingPortals	Folder
BP_PortalHub	Edit BP_PortalHub
SM_Cave_EntranceSmall	StaticMeshActor
SM_Cave_EntranceDoor	StaticMeshActor
SP_PortalDoor_base	StaticMeshActor
SP_PortalNightTime_ent_MAIN	Edit BP_PortalActor
SP_PortalAtticFloor_ent_MAIN	Edit BP_PortalActor
SP_PortalLevelAttic_MAIN	Edit BP_PortalLevelStreamer
SP_PortalLevelNightTime_MAIN	Edit BP_PortalLevelStreamer
SP_PortalLevelNightTimeTube_MAIN	Edit BP_PortalLevelStreamer
SP_Tube_Travel	StaticMeshActor
SP_PortalNightTube_ent_MAIN	Edit BP_PortalActor
Tube_Portal_Border	StaticMeshActor
Supports	Folder

127 actors (1 selected)

SP_PortalLevelNightTime_MAIN

SP_PortalLevelNightTime_MAIN (Self)

Root Component (DefaultSceneRoot) Edit in C++

Scale	1.0	1.0	1.0

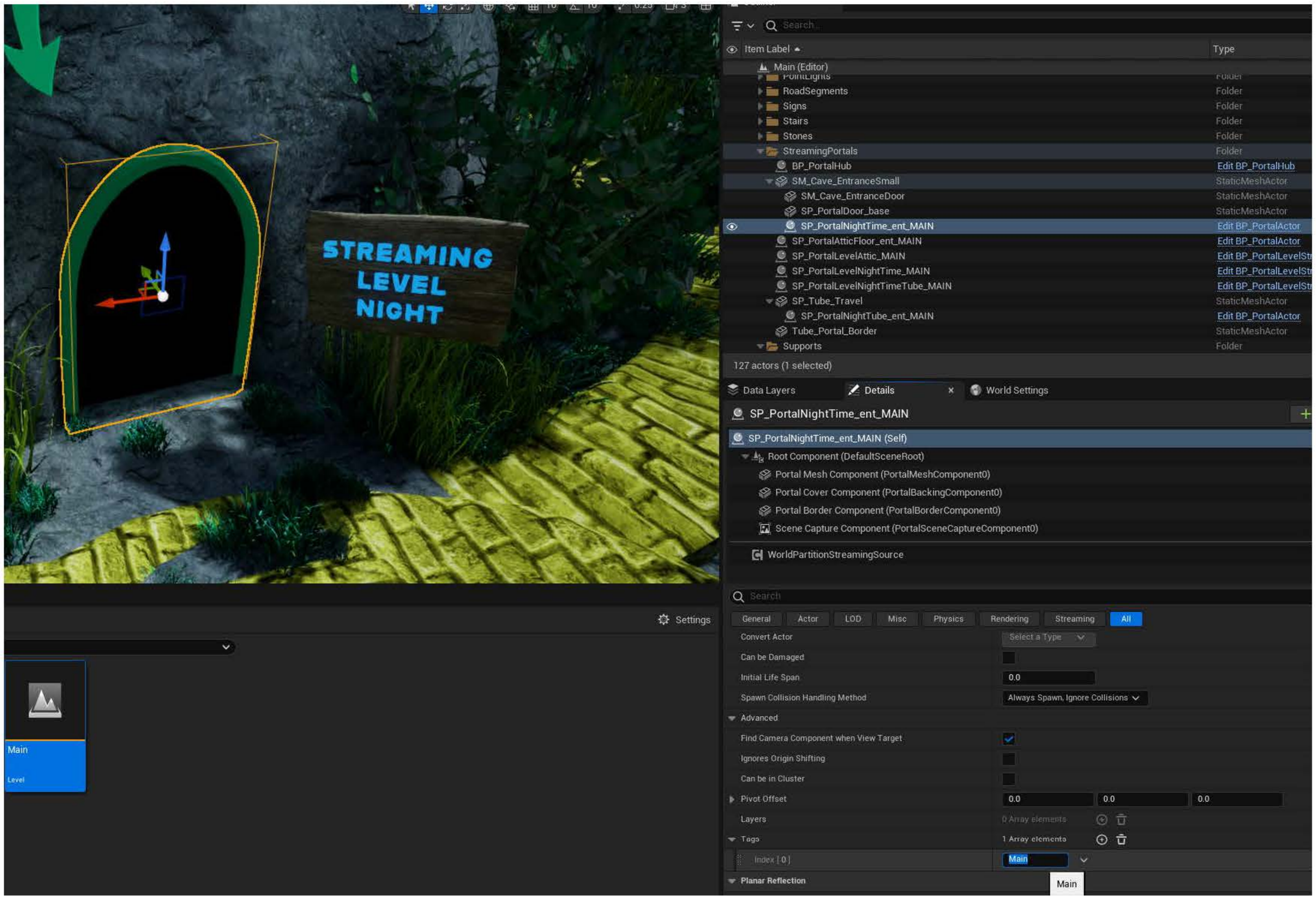
Streaming Portal

Setup

Level to Load	Nighttime
Start Loaded	
Portal Pairs	1 Array elements
Index [0]	2 members
Entrance	SP_PortalNightTime_ent_MAIN
Exit	None
Use Channel	Channel 1

Reference to Actor ID '/SPLContent/Maps/Main.Main:PersistentLevel.SP_PortalActor_3'

The Portal Actor



Item Label	Type
Main (Editor)	
PointLights	Folder
RoadSegments	Folder
Signs	Folder
Stairs	Folder
Stones	Folder
StreamingPortals	Folder
BP_PortalHub	Edit BP_PortalHub
SM_Cave_EntranceSmall	StaticMeshActor
SM_Cave_EntranceDoor	StaticMeshActor
SP_PortalDoor_base	StaticMeshActor
SP_PortalNightTime_ent_MAIN	Edit BP_PortalActor
SP_PortalAtticFloor_ent_MAIN	Edit BP_PortalActor
SP_PortalLevelAttic_MAIN	Edit BP_PortalLevelStr
SP_PortalLevelNightTime_MAIN	Edit BP_PortalLevelStr
SP_PortalLevelNightTimeTube_MAIN	Edit BP_PortalLevelStr
SP_Tube_Travel	StaticMeshActor
SP_PortalNightTube_ent_MAIN	Edit BP_PortalActor
Tube_Portal_Border	StaticMeshActor
Supports	Folder

127 actors (1 selected)

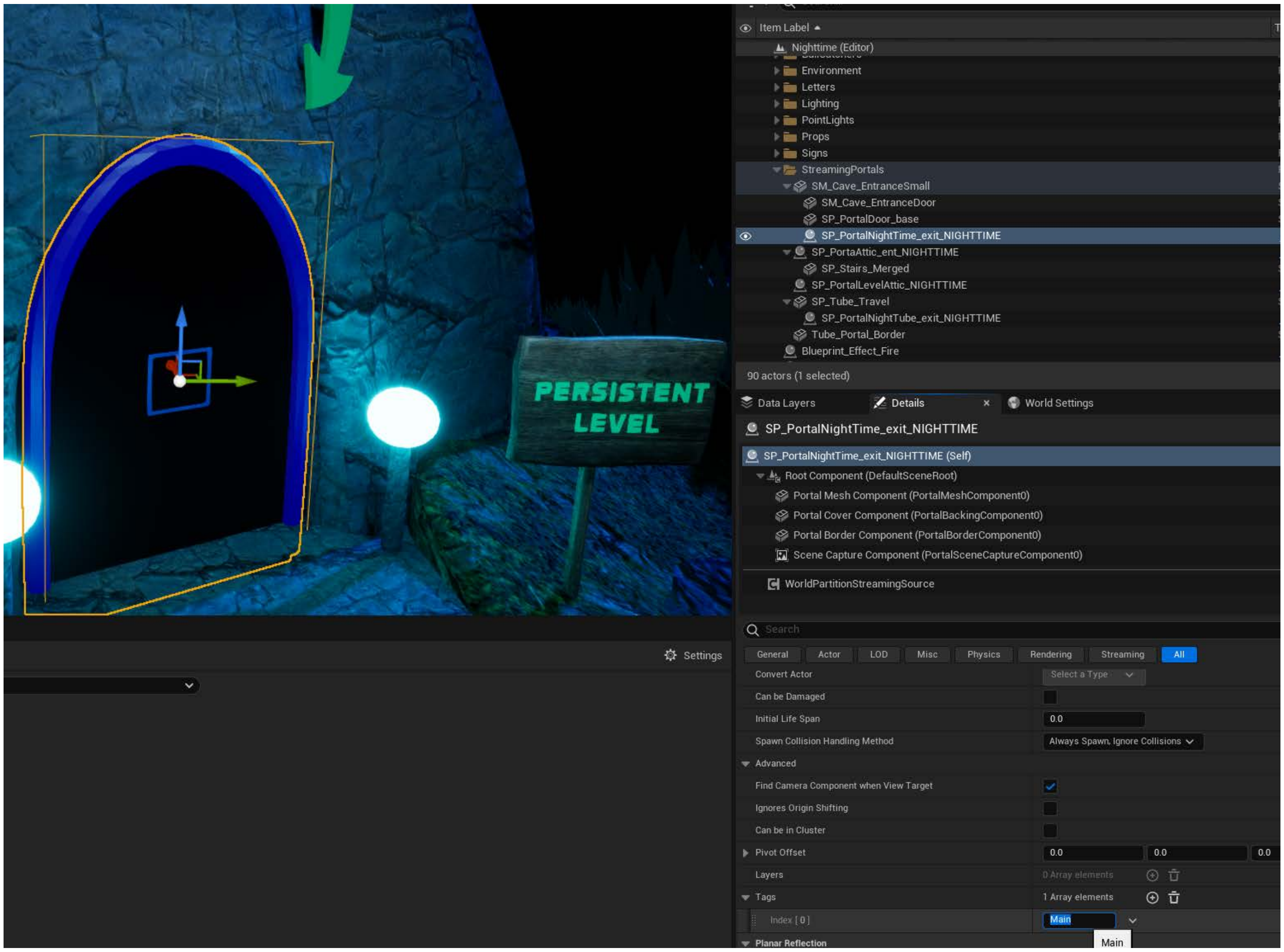
SP_PortalNightTime_ent_MAIN

- SP_PortalNightTime_ent_MAIN (Self)
  - Root Component (DefaultSceneRoot)
    - Portal Mesh Component (PortalMeshComponent0)
    - Portal Cover Component (PortalBackingComponent0)
    - Portal Border Component (PortalBorderComponent0)
    - Scene Capture Component (PortalSceneCaptureComponent0)
  - WorldPartitionStreamingSource

Convert Actor	Select a Type
Can be Damaged	☐
Initial Life Span	0.0
Spawn Collision Handling Method	Always Spawn, Ignore Collisions
Advanced	
Find Camera Component when View Target	✓
Ignores Origin Shifting	☐
Can be in Cluster	☐
Pivot Offset	0.0 0.0 0.0
Layers	0 Array elements
Tags	1 Array elements
Index [0]	Main
Planar Reflection	

Main

The Nighttime Level



Item Label
- Nighttime (Editor)
  - Environment
  - Letters
  - Lighting
  - PointLights
  - Props
  - Signs
  - StreamingPortals
    - SM_Cave_EntranceSmall
      - SM_Cave_EntranceDoor
      - SP_PortalDoor_base
      - SP_PortalNightTime_exit_NIGHTTIME
    - SP_PortaAttic_ent_NIGHTTIME
      - SP_Stairs_Merged
    - SP_PortalLevelAttic_NIGHTTIME
    - SP_Tube_Travel
      - SP_PortalNightTube_exit_NIGHTTIME
    - Tube_Portal_Border
  - Blueprint_Effect_Fire

90 actors (1 selected)

Data Layers | Details | World Settings

SP_PortalNightTime_exit_NIGHTTIME

- SP_PortalNightTime_exit_NIGHTTIME (Self)
  - Root Component (DefaultSceneRoot)
    - Portal Mesh Component (PortalMeshComponent0)
    - Portal Cover Component (PortalBackingComponent0)
    - Portal Border Component (PortalBorderComponent0)
    - Scene Capture Component (PortalSceneCaptureComponent0)
  - WorldPartitionStreamingSource

General | Actor | LOD | Misc | Physics | Rendering | Streaming | All

Convert Actor	Select a Type
Can be Damaged	
Initial Life Span	0.0
Spawn Collision Handling Method	Always Spawn, Ignore Collisions
Advanced	
Find Camera Component when View Target	✓
Ignores Origin Shifting	
Can be in Cluster	
Pivot Offset	0.0 0.0 0.0
Layers	0 Array elements
Tags	1 Array elements
Index [0]	Main
Planar Reflection	Main

When the level is streamed in, the Portal Hub will resolve the portal pairs based on these tags.

You can have many portals leading in and out to the same levels, but each tag must be a one to one relation between Entrance and Exit..

Eg:

Main1

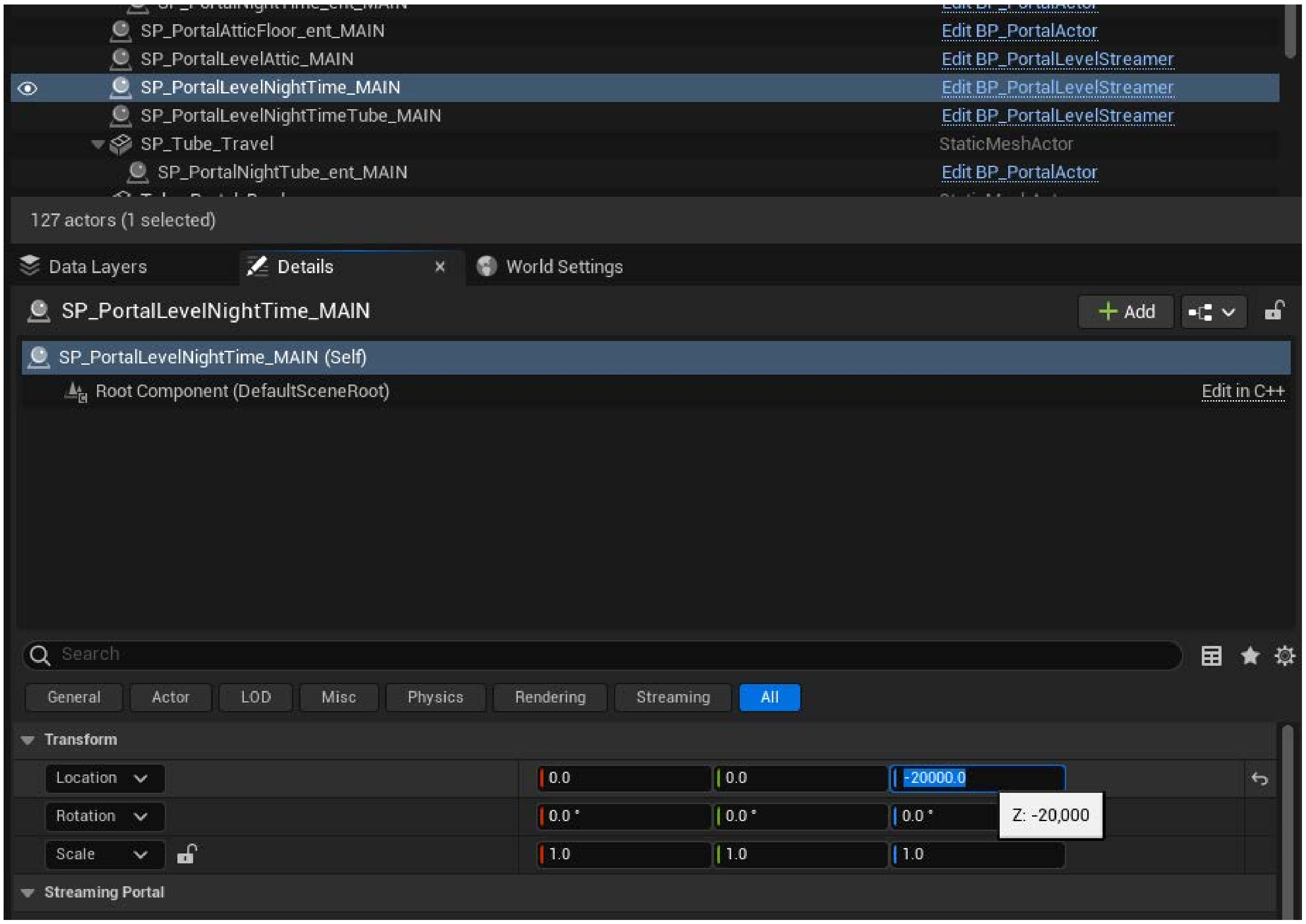
Main2

Main3

Etc..

Placing the Portal Level Streamer is Key!

The level will stream in, with it's origin where the Portal Level Streamer is. So let's change the Z value on the Portal Level Streamer to -20,000



SP_PortalAtticFloor_ent_MAIN	Edit BP_PortalActor
SP_PortalLevelAttic_MAIN	Edit BP_PortalLevelStreamer
SP_PortalLevelNightTime_MAIN	Edit BP_PortalLevelStreamer
SP_PortalLevelNightTimeTube_MAIN	Edit BP_PortalLevelStreamer
SP_Tube_Travel	StaticMeshActor
SP_PortalNightTube_ent_MAIN	Edit BP_PortalActor

127 actors (1 selected)

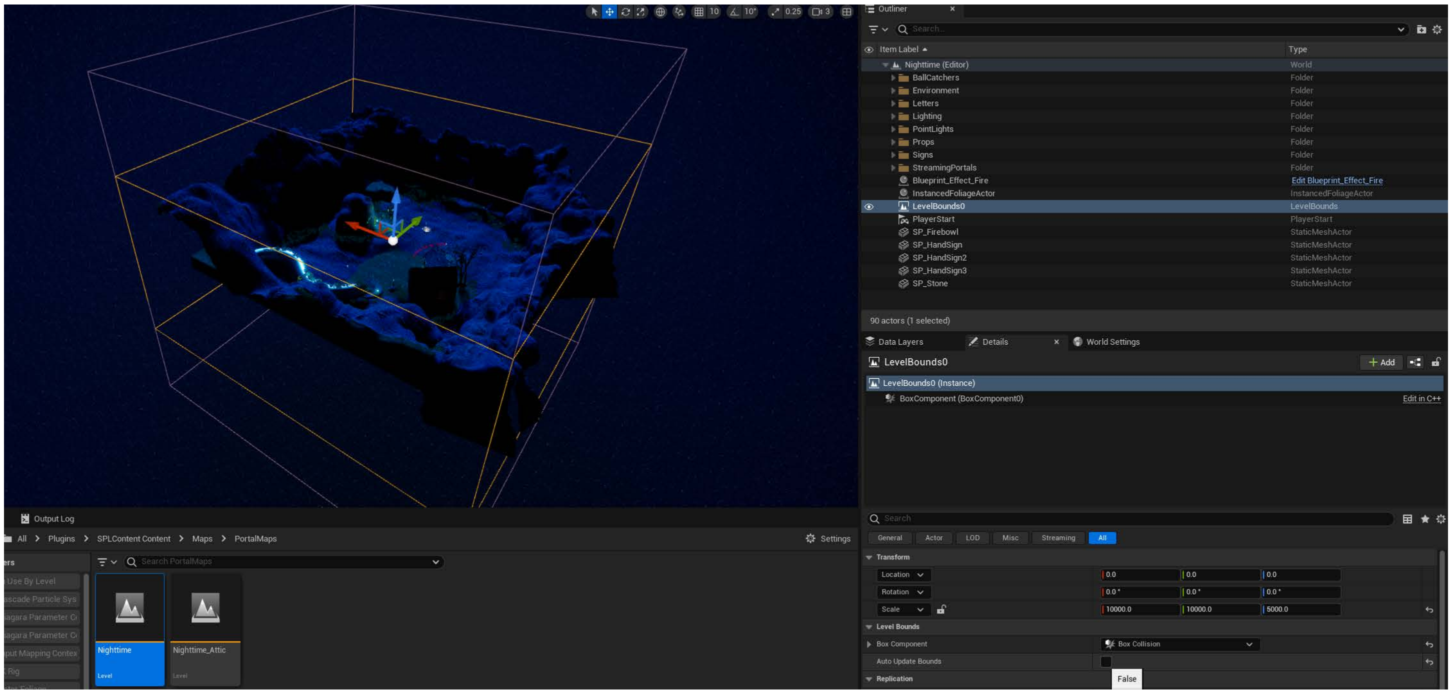
Location	0.0	0.0	-20000.0
Rotation	0.0 °	0.0 °	0.0 °
Scale	1.0	1.0	1.0

Z: -20,000

This will stream in the level -20,000 units below the current level. I do suggest placing streamed levels below the main level or it can cast shadows on other scene capture components you may have running. You can also move them far away on X and Y axis. It's really up to you.

The Stream Level itself **MUST HAVE** a Level Bounds. With out this the system doesn't know how big your level is.

The Level bounds should fit nicely around your level and turn off Auto Update Bounds



Item Label	Type
Nighttime (Editor)	World
BallCatchers	Folder
Environment	Folder
Letters	Folder
Lighting	Folder
PointLights	Folder
Props	Folder
Signs	Folder
StreamingPortals	Folder
Blueprint_Effect_Fire	Edit Blueprint_Effect_Fire
InstancedFoliageActor	InstancedFoliageActor
LevelBounds0	LevelBounds
PlayerStart	PlayerStart
SP_Firebowl	StaticMeshActor
SP_HandSign	StaticMeshActor
SP_HandSign2	StaticMeshActor
SP_HandSign3	StaticMeshActor
SP_Stone	StaticMeshActor

90 actors (1 selected)

LevelBounds0

LevelBounds0 (Instance)
BoxComponent (BoxComponent0)

Transform			
Location	0.0	0.0	0.0
Rotation	0.0 °	0.0 °	0.0 °
Scale	10000.0	10000.0	5000.0

Level Bounds	
Box Component	Box Collision
Auto Update Bounds	False

This is used to determine if the player is in the streamed level space or not.

BP_Portal_Actor

This is the portals themselves. The consist of:

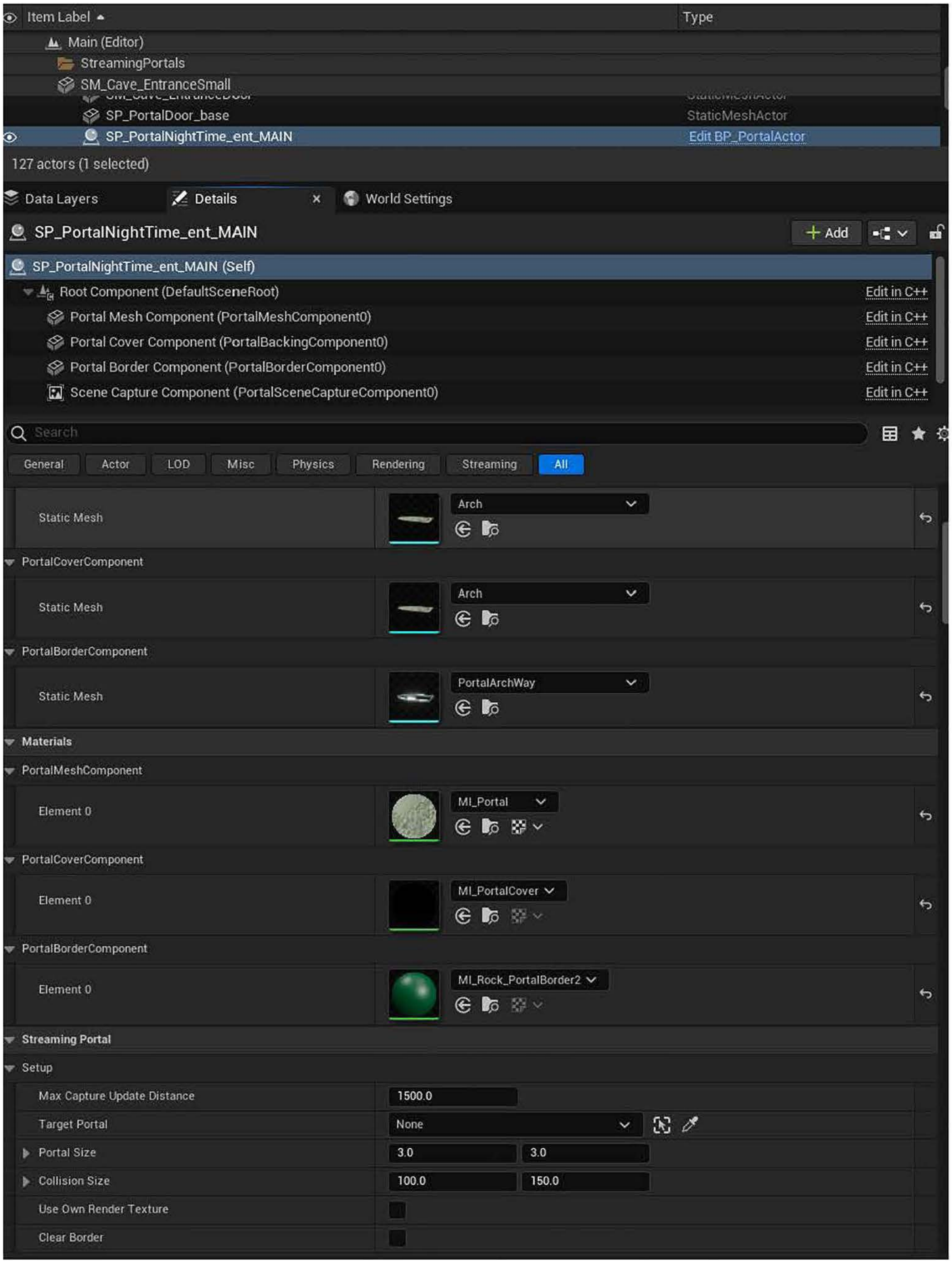
- PortalMesh - The mesh that the portal is on
- PortalCover - Cover for when the portal is not rendering
- PortalBorder - (Optional)
- Collision - Used to start tracking the portal

All other settings can be found in the StreamingPortal --> Setup section

- Max Capture Update Distance - Max distance the portal will render at
- Target Portal - The target portal for this portal.
  Leave blank if the target is in another level.

- Portal Size - Size of portal scale
- Collision Size - size of collision scale
- Use Own Render Texture - Should use its own render texture instead of the shared one

- Clear Border - Do not want a boarder, click this and clear the mesh. It will stay cleared



Item Label ▲ | Type

Main (Editor)
StreamingPortals
SM_Cave_EntranceSmall
SP_PortalDoor_base StaticMeshActor
SP_PortalNightTime_ent_MAIN Edit BP_PortalActor

127 actors (1 selected)

Data Layers | Details | World Settings

SP_PortalNightTime_ent_MAIN + Add

SP_PortalNightTime_ent_MAIN (Self)
Root Component (DefaultSceneRoot) Edit in C++
Portal Mesh Component (PortalMeshComponent0) Edit in C++
Portal Cover Component (PortalBackingComponent0) Edit in C++
Portal Border Component (PortalBorderComponent0) Edit in C++
Scene Capture Component (PortalSceneCaptureComponent0) Edit in C++

General | Actor | LOD | Misc | Physics | Rendering | Streaming | All

Static Mesh: Arch

PortalCoverComponent
Static Mesh: Arch

PortalBorderComponent
Static Mesh: PortalArchWay

Materials
PortalMeshComponent
Element 0: MI_Portal
PortalCoverComponent
Element 0: MI_PortalCover
PortalBorderComponent
Element 0: MI_Rock_PortalBorder2

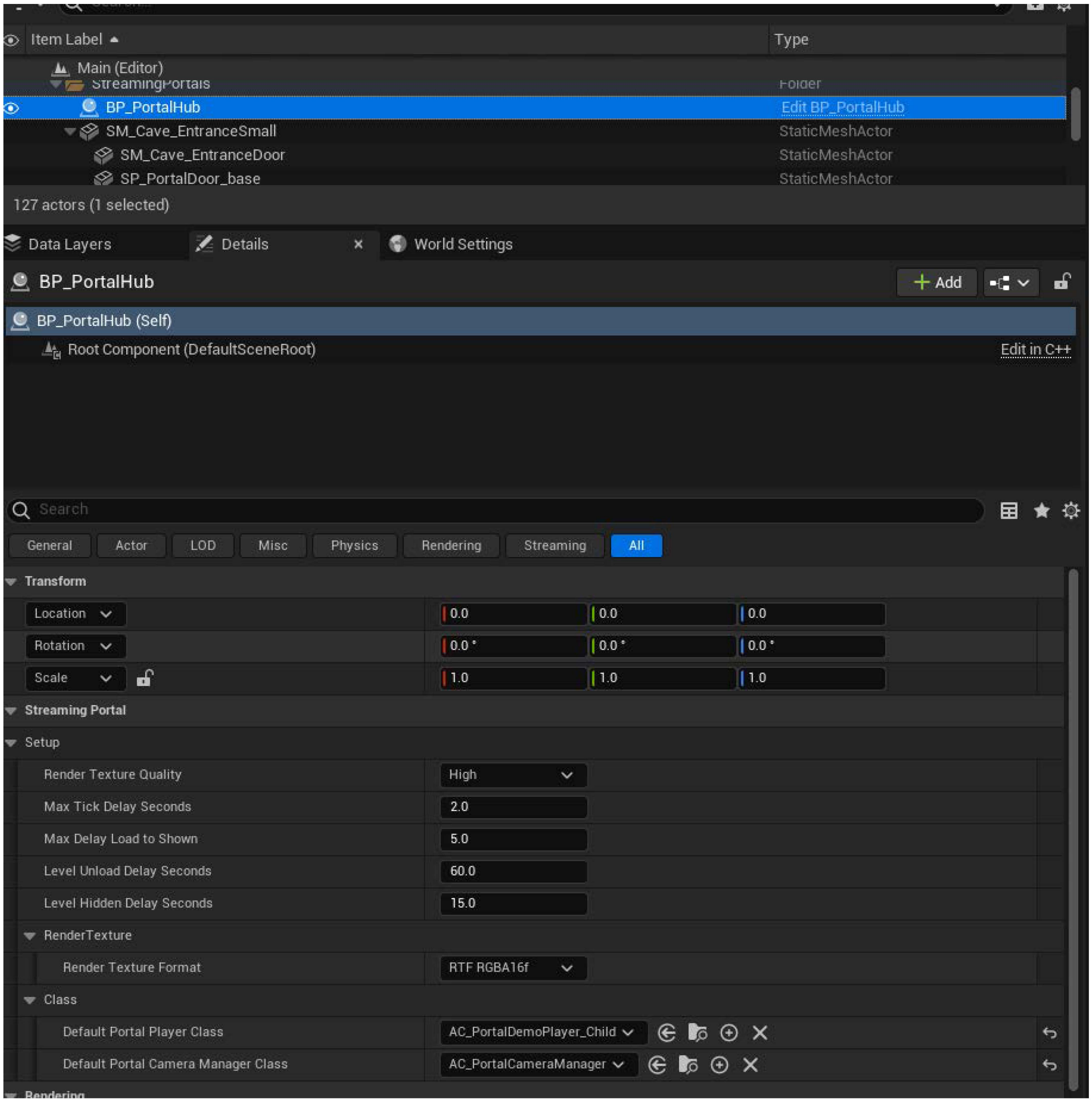
Streaming Portal
Setup

Max Capture Update Distance	1500.0	
Target Portal	None	
Portal Size	3.0	3.0
Collision Size	100.0	150.0
Use Own Render Texture	☐	
Clear Border	☐	

BP_PortalHub

Portal Hub controls the portal system and contains settings for it.
The settings are:

- Render Texture Quality - Low, Medium, High, Very High or Ultra.
  Controls the scale of the render texture compared to the screen
  (50%, 59%, 67%, 77%, 100%)
- Max Tick Delay Seconds - How fast the Portal Level Streamers should tick.
- Max Delay Load to Shown - The time between the level loading and being shown (running BeginPlay on everything). A small delay can help smooth level loading.
- Level Unload Delay Seconds - Delay after a level is hidden until it's unloaded
- Level Hidden Delay Seconds - How long the player is out of range before a level is hidden.
- Render Texture Format - The format you want the render texture in for the portal
- Default Portal Player Class - The class for controlling the players journey through the portal.
- Default Portal Camera Manager Class - The class for controlling the players camera for the journey through the portal.



Item Label | Type

Main (Editor)
StreamingPortals | Folder
BP_PortalHub | Edit BP_PortalHub
SM_Cave_EntranceSmall | StaticMeshActor
SM_Cave_EntranceDoor | StaticMeshActor
SP_PortalDoor_base | StaticMeshActor

127 actors (1 selected)

BP_PortalHub
BP_PortalHub (Self)
Root Component (DefaultSceneRoot)

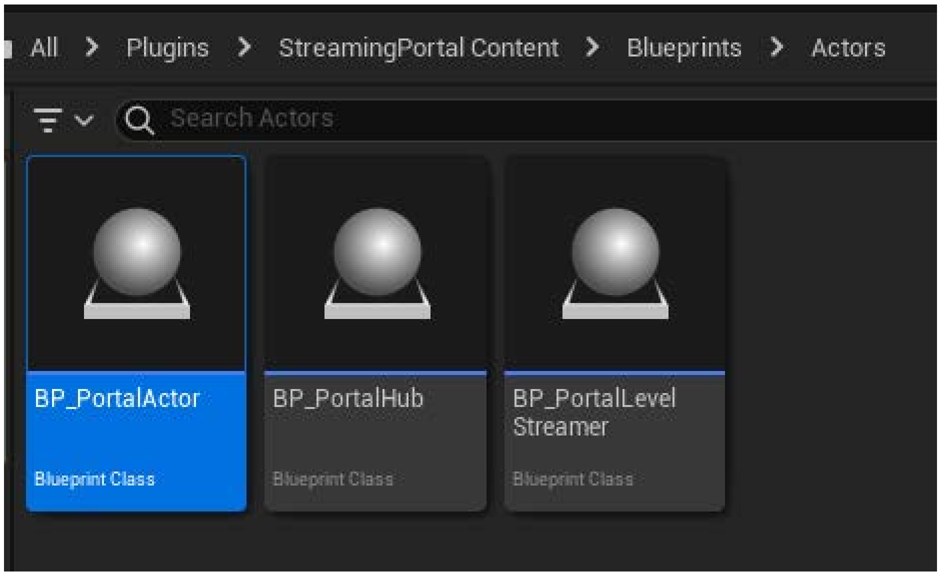
Transform
Location 0.0 0.0 0.0
Rotation 0.0 ° 0.0 ° 0.0 °
Scale 1.0 1.0 1.0

Streaming Portal
Setup
Render Texture Quality: High
Max Tick Delay Seconds: 2.0
Max Delay Load to Shown: 5.0
Level Unload Delay Seconds: 60.0
Level Hidden Delay Seconds: 15.0
RenderTexture
Render Texture Format: RTF RGBA16f
Class
Default Portal Player Class: AC_PortalDemoPlayer_Child
Default Portal Camera Manager Class: AC_PortalCameraManager
Rendering

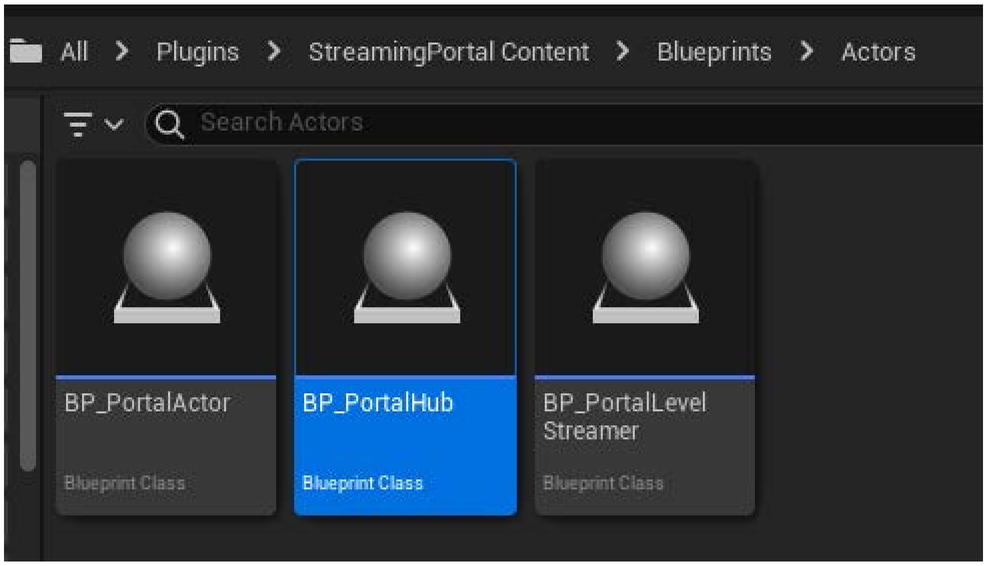
BP_PortalLevelStreamer

A Portal Level Streamer - manages the streaming of the level. It will set the level hidden when the player is away after some time (As set in the Portal Hub). And will unload the level as needed.

The Portal Level Streamer will track the Players proximity to the Portals and load / show the level. This distance can also be adjusted (the Max Capture Update Distance on BP_PortalActor).



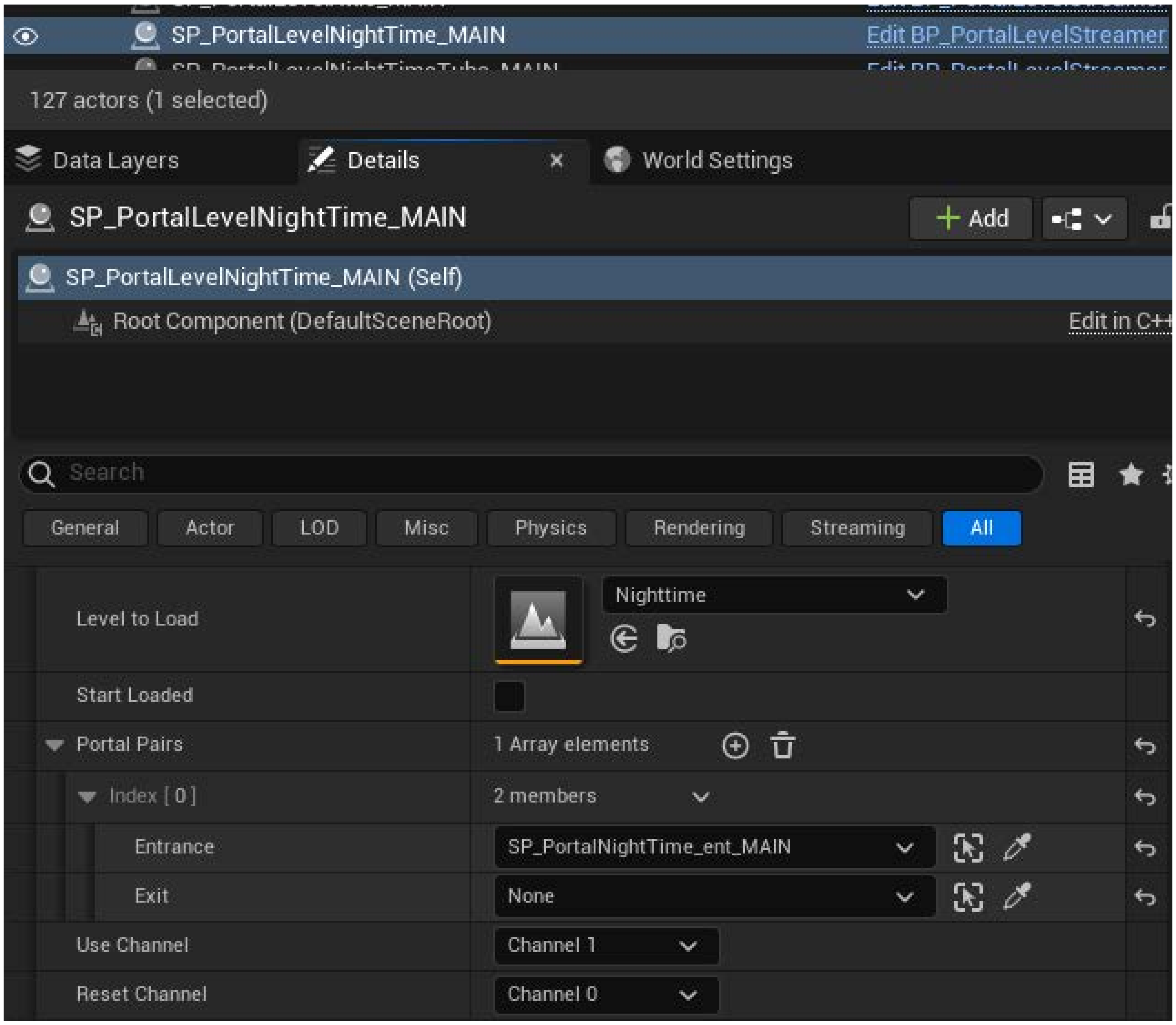
The Portal Level Streamer has a list of Portal Pairs. An Entrance and Exit. The exit is always left as "None" since it will be in the streamed level and resolved by the BP_PortalHub.



The Light Channel settings are used if you want to have different lighting between levels. Going from a day level to night for example. If though, you are just going "inside" a building, you can set both light channels to 0. In the demo we use all 3 for fun.

- Channel 0 - for the main level
- Channel 1 - for the night time level
- Channel 3 - for the red attic level

Start Loaded is used if you have the player spawn first in the level inside the streaming level. We have this in our own project, it's called the Home Room. Click this to true will cause the level to be loaded during BeginPlay of the Portal Level Streamer.



SP_PortalLevelNightTime_MAIN — Edit BP_PortalLevelStreamer

127 actors (1 selected)

SP_PortalLevelNightTime_MAIN (Self)

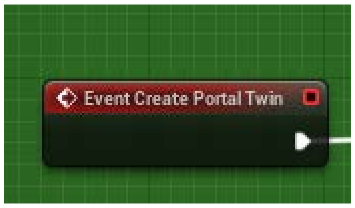
Root Component (DefaultSceneRoot) — Edit in C++

Level to Load	Nighttime
Start Loaded	☐
Portal Pairs	1 Array elements
Index [0]	2 members
Entrance	SP_PortalNightTime_ent_MAIN
Exit	None
Use Channel	Channel 1
Reset Channel	Channel 0

AC_Portal_Actor

The Portal Actor Component s a generic component that could be used for any actor to pass through a portal. The actor needs to have at least one Primitive Component (something render able). In the demo the BP_Ball uses this.

Create Portal Twin



This is called when the component wants the portal twin created. It is called just before the Pre Teleport call back / event, but is only called once and assumed the Portal Twin was created.

Example of creating basic portal twin as been left as blueprint nodes for easier customization.

Bind Events

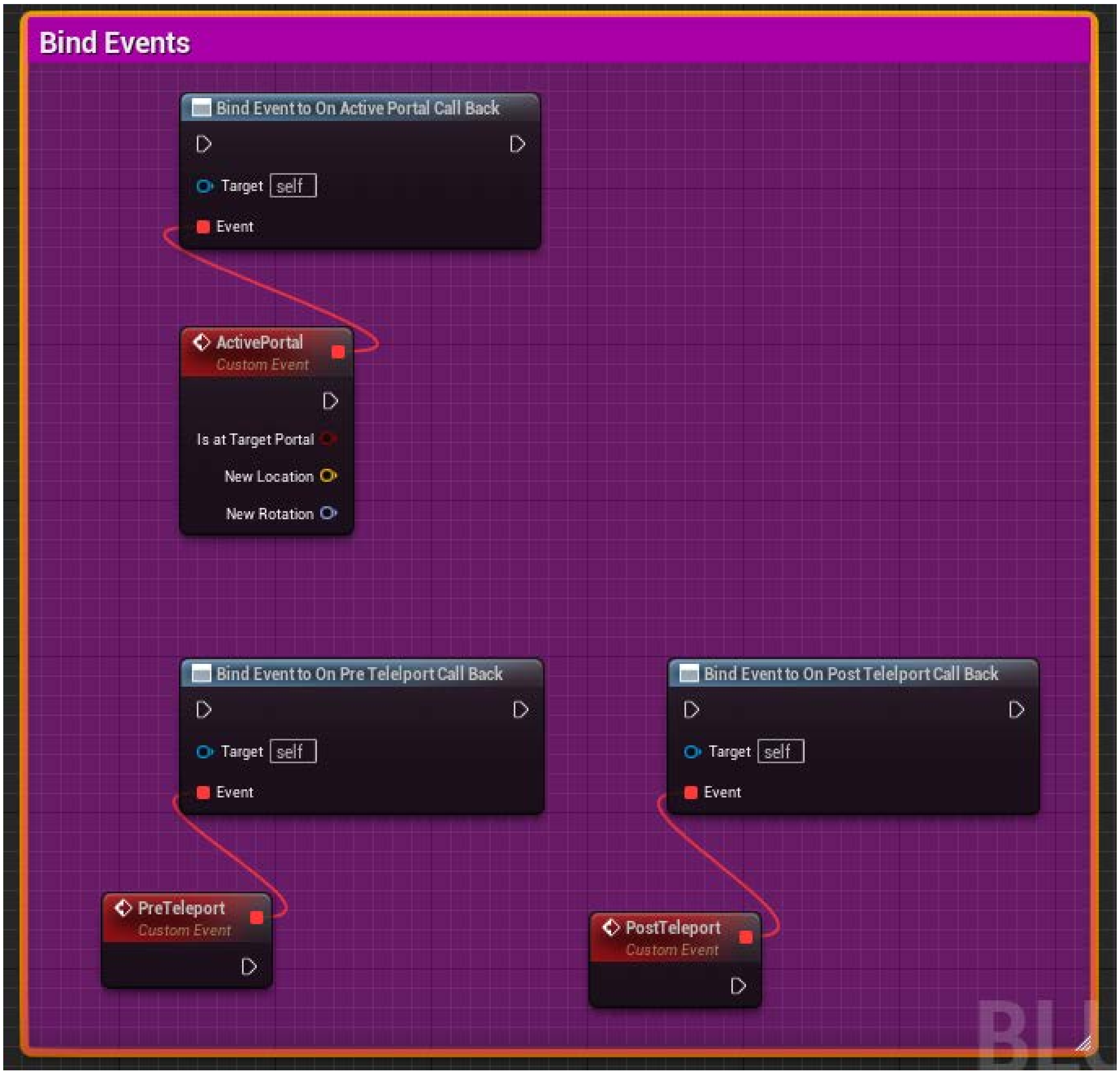
The system contains both Bind Events and regular events. They are called at the same time, you can use either the choice is yours.

On Active Portal Call Back - Called when a portal is teleporting an actor

On Pre Teleport Call Back - Called when the component starts tracking and portal

On Post Teleport Call Back - Called when the component stops tracking a portal

Bind Events contd...



Events

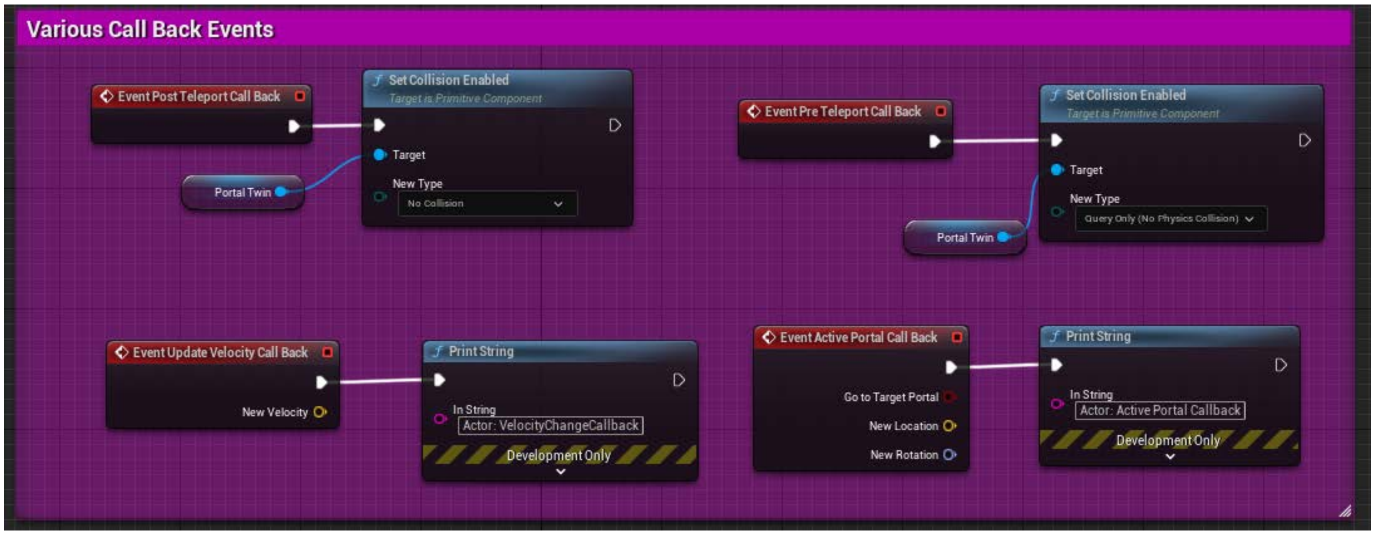
Similar to the bind events the regular events are called at the same time, you can use either the choice is yours.

Active Portal Call Back - Called when a portal is teleporting an actor

Pre Teleport Call Back - Called when the component starts tracking and portal

Post Teleport Call Back - Called when the component stops tracking a portal

Update Velocity Call Back - Called during teleport and returns the new velocity vector change



Various Call Back Events
Event Post Teleport Call Back
Portal Twin
Set Collision Enabled
Target is Primitive Component
Target
New Type
No Collision
Event Pre Teleport Call Back
Set Collision Enabled
Target is Primitive Component
Target
New Type
Query Only (No Physics Collision)
Portal Twin
Event Update Velocity Call Back
New Velocity
Print String
In String
Actor: VelocityChangeCallback
Development Only
Event Active Portal Call Back
Go to Target Portal
New Location
New Rotation
Print String
In String
Actor: Active Portal Callback
Development Only

Settings

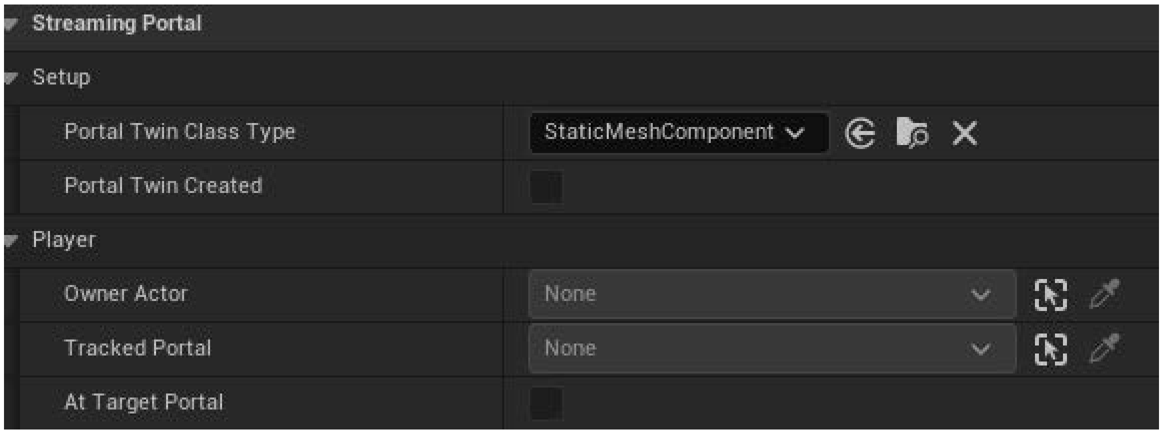
Portal Twin Class Type - Default is Static Mesh Component, but can be any Primitive Component you like.

Portal Twin Created - Read Only bool set after calling Create Portal Twin

Owner Actor - Read Only the actor that owns this component

Tracked Portal - Read Only the portal currently being tracked

At Target Portal - Read Only is the actor at the Target portal or not



Streaming Portal

Setup

Portal Twin Class Type	StaticMeshComponent
Portal Twin Created	☐

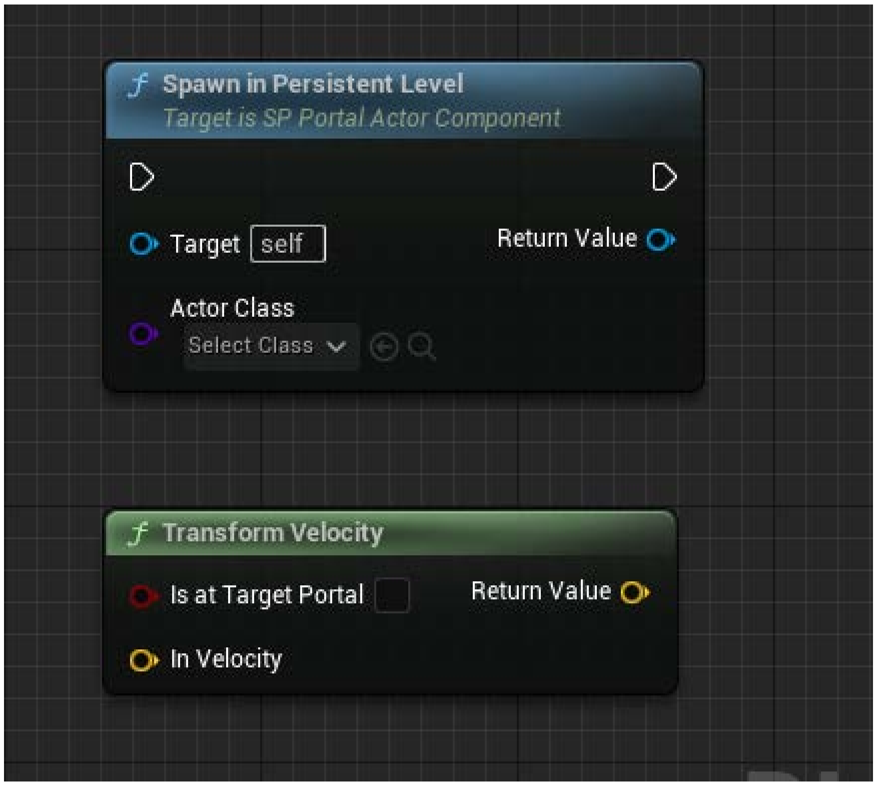
Player

Owner Actor	None
Tracked Portal	None
At Target Portal	☐

Helper Nodes

Spawn in Persistent Level - Will spawn an Actor Class in the Persistent Level. Used on the demo with the Attic Ball

Transform Velocity - Will update the velocity based on the portal transforms. Used in the demo with the physics balls. Sometimes the Player Character is pushing on the Balls Portal Twin. Since the Balls Portal Twin has no physics, the physics must be applied to the actual ball. So the velocity needs to be transformed to the portal the physics ball is at and applied.

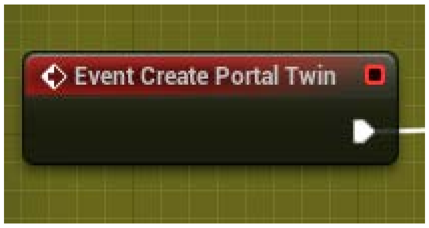


AC_Portal_Player

The Portal Player Component

The Portal Player Component is inherited from the Portal Actor Component, but is designed for the main player character. It must be ACharacter class. In the demo this is automatically attached to the BP_ThirdPersonCharacter player.

Create Portal Twin



This is called when the component wants the portal twin created. It is called just before the Pre Teleport call back / event, but is only called once and assumed the Portal Twin was created.

Example of creating basic portal twin as been left as blueprint nodes for easier customization. Must be Skeletal Mesh Component.

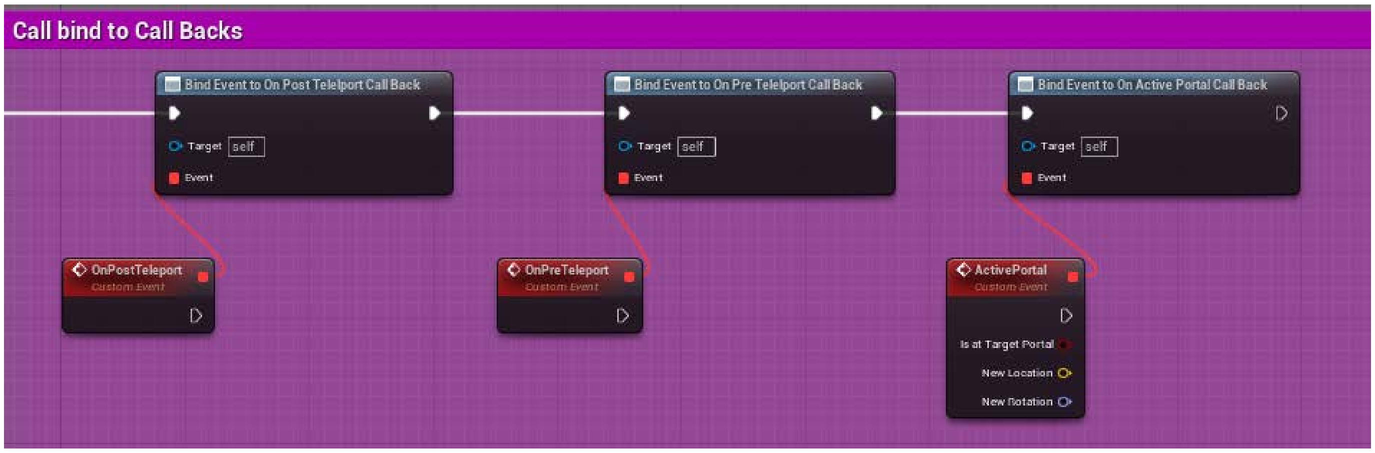
Bind Events

The system contains both Bind Events and regular events. They are called at the same time, you can use either, the choice is yours.

On Active Portal Call Back - Called when a portal is teleporting an the player.

On Pre Teleport Call Back - Called when the component starts tracking and portal

On Post Teleport Call Back - Called when the component stops tracking a portal



Events

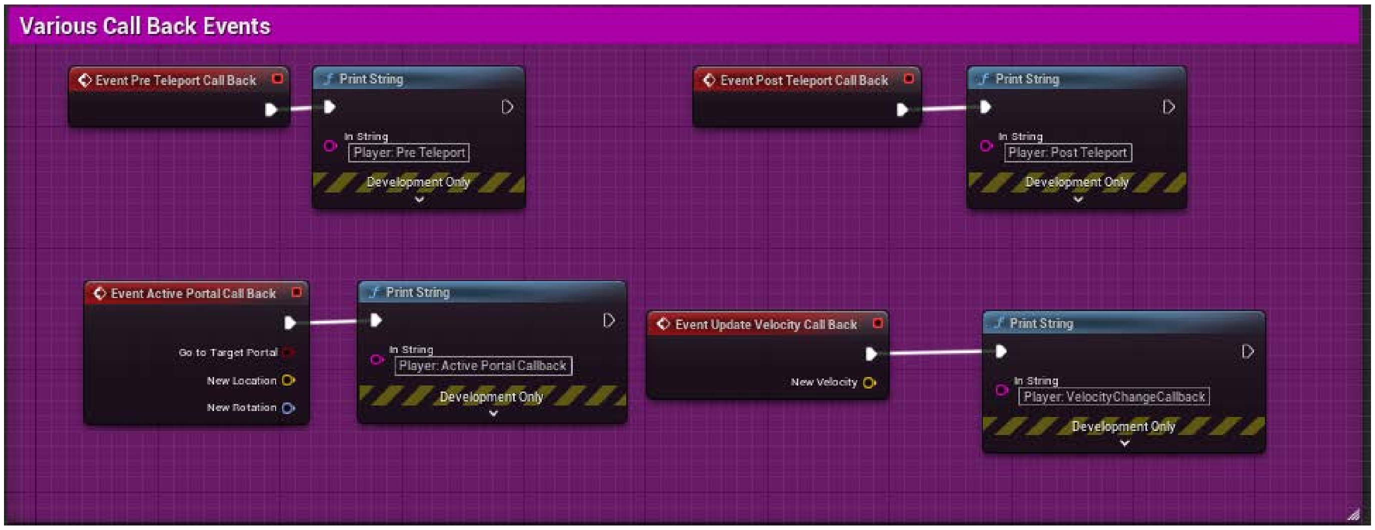
Similar to the bind events the regular events are called at the same time, you can use either, the choice is yours.

Active Portal Call Back - Called when a portal is teleporting an actor

Pre Teleport Call Back - Called when the component starts tracking and portal

Post Teleport Call Back - Called when the component stops tracking a portal

Update Velocity Call Back - Called during teleport and returns the new velocity vector change



Settings

Max Camera to Player Range - How far can the camera get from the player character

Portal Twin Class Type - This is ignored for the Player Component as it must be a Skeletal Mesh Component type, as that is what the Mesh for the Character class is.

Portal Twin Created - Read Only bool set after calling Create Portal Twin

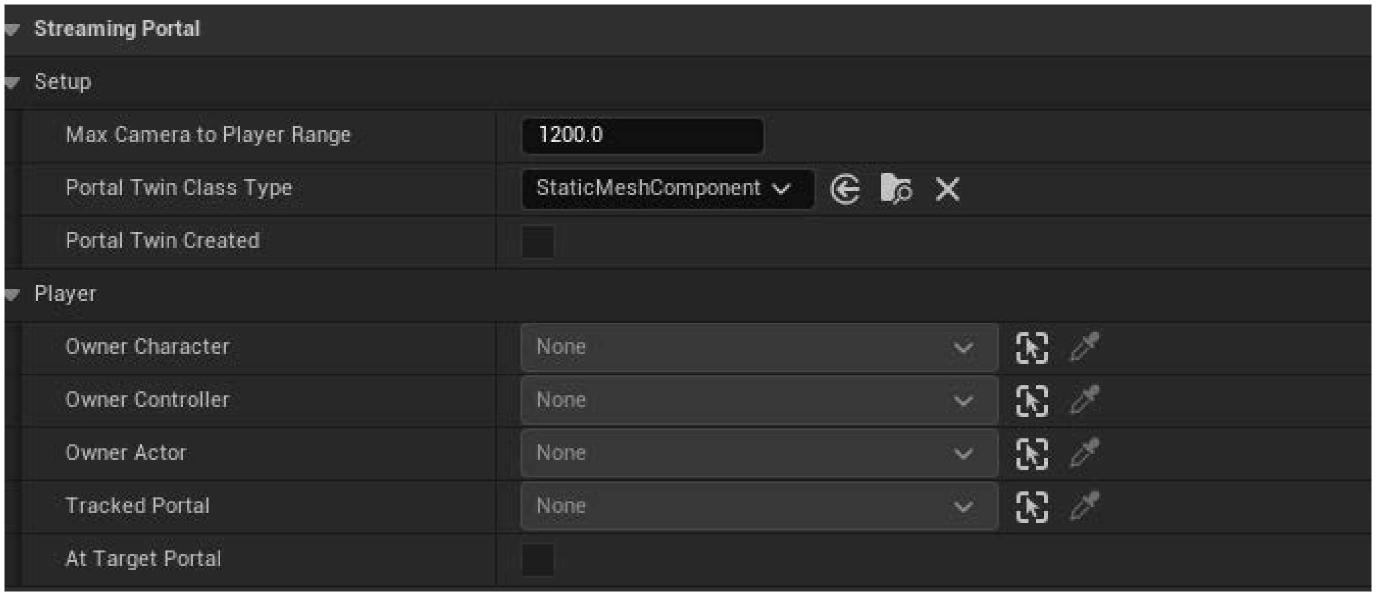
Owner Character - Read Only the Character that owns this component

Owner Controller - Read Only the controller of the Character that owns this component

Owner Actor - Read Only the actor that owns this component

Tracked Portal - Read Only the portal currently being tracked

At Target Portal - Read Only is the actor at the Target portal or not



Streaming Portal

Setup

Max Camera to Player Range	1200.0
Portal Twin Class Type	StaticMeshComponent
Portal Twin Created	☐

Player

Owner Character	None
Owner Controller	None
Owner Actor	None
Tracked Portal	None
At Target Portal	☐

AC_PortalCameraManager

Bind Events

The system contains both Bind Events and regular events. They are called at the same time, you can use either the choice is yours.

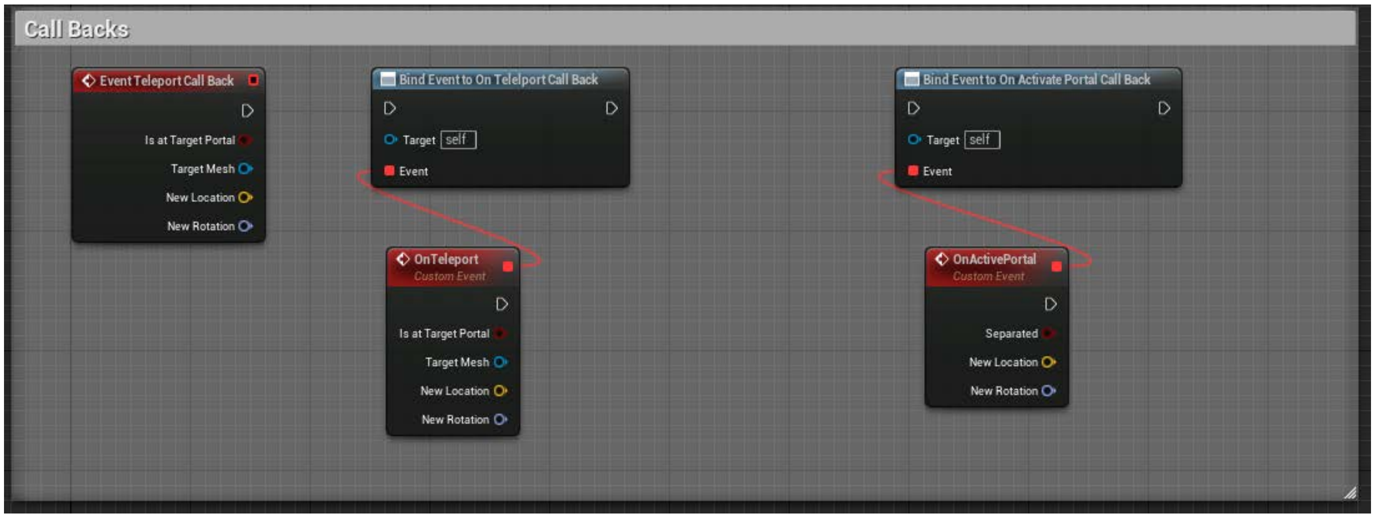
On Active Portal Call Back - Called when a portal becomes active. Weather it's the camera or Player Character that actives it. Tells us if the camera is being separated from the player or not.

On Teleport Call Back - Called when the camera is teleported. Target Mesh is either the Portal Twin or the Player Character Mesh

Events

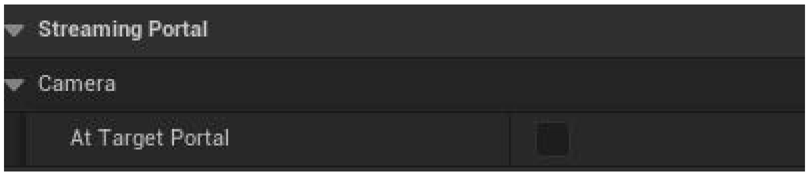
Similar to the bind events the regular events are called at the same time, you can use either the choice is yours.

Teleport Call Back - Called when the camera is teleported. Target Mesh is either the Portal Twin or the Player Character Mesh.



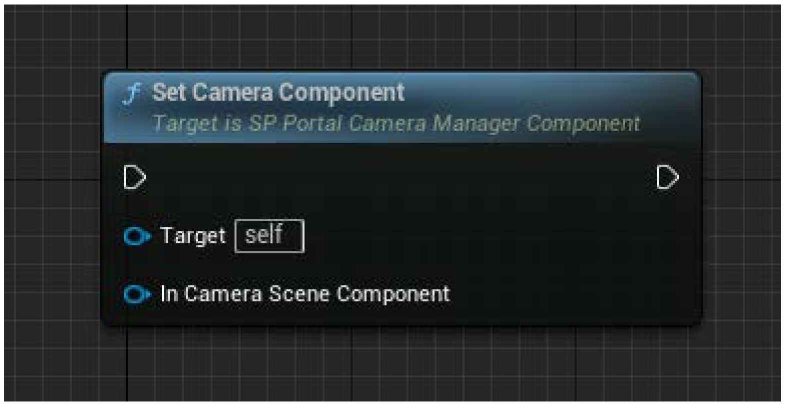
Settings

At Target Portal - Read Only is the actor at the Target portal or not

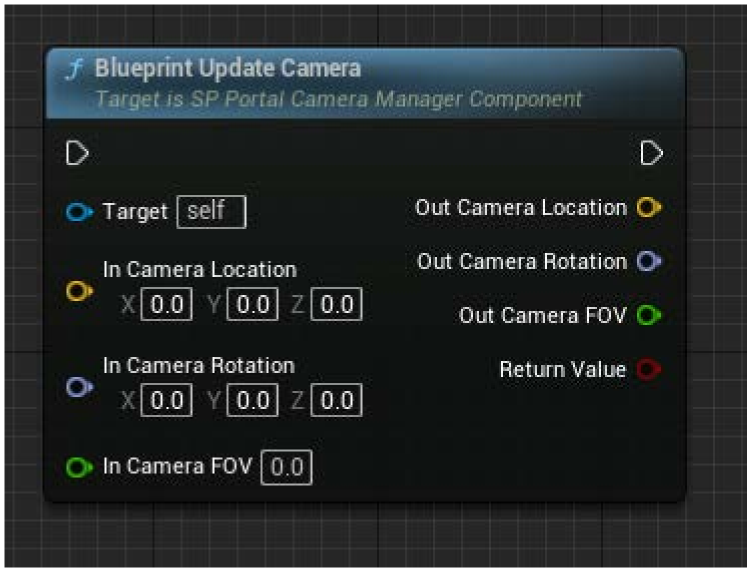


Helper Nodes

Set Camera Component if you want to manually set the camera component. The system will attempt to find the camera component, if it fails to do so you can set it manually with this.

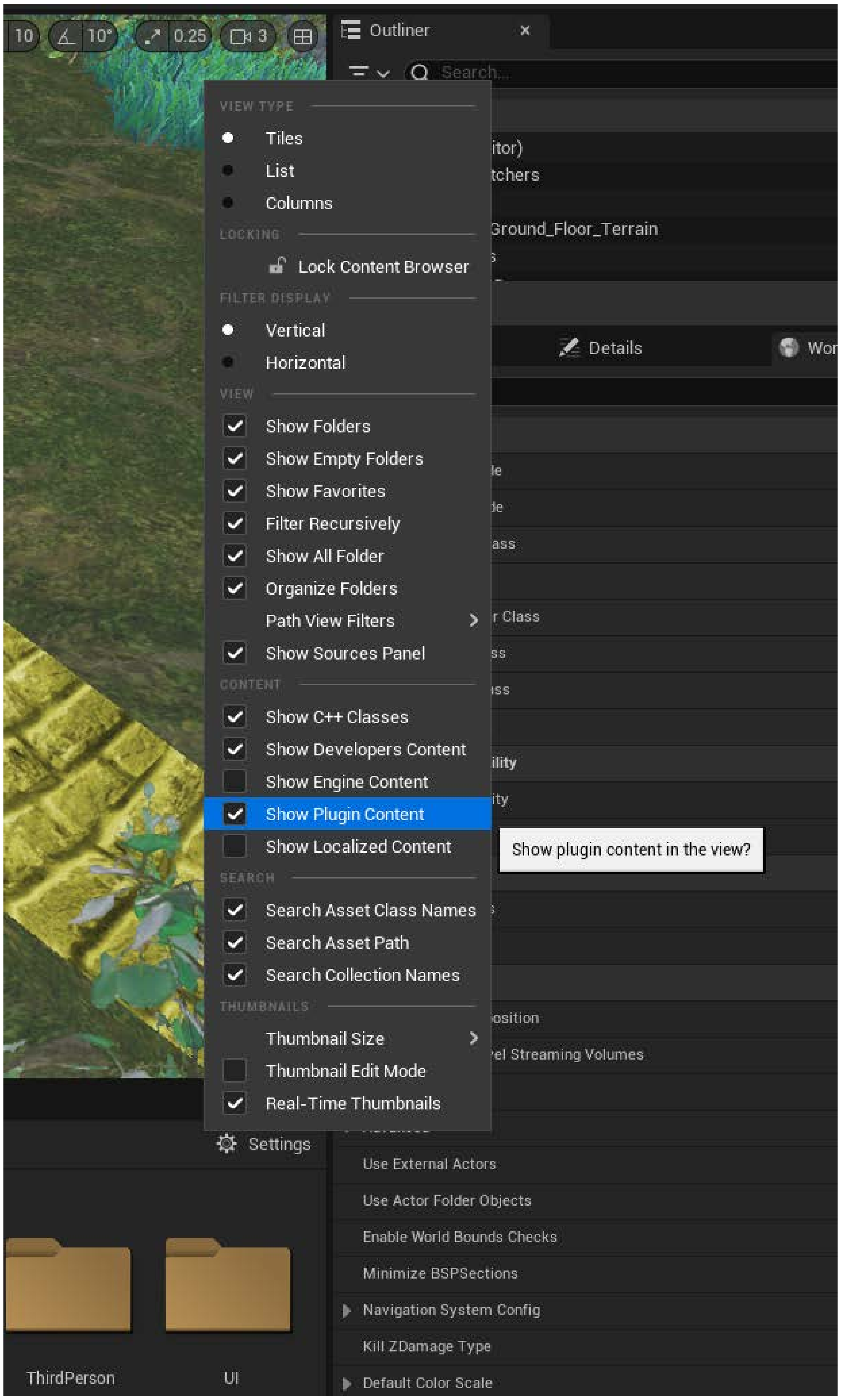

Set Camera Component
Target is SP Portal Camera Manager Component
Target
self
In Camera Scene Component

Can be used in custom player camera managers. If you have a Blueprint Update Camera method use this to pass in your outputs. Normally it will just pass the values through unless tracking a portal and the camera becomes separated from the player character.



Demo Content

The StreamingPortal Plugin Content has a DemoContent folder that contains all the content for the demo. To see plugin content you must enable this in the content browser. Click on the gear on the far right and select "Show Plugin Content"



VIEW TYPE
- ● Tiles
- List
- Columns

LOCKING
- Lock Content Browser

FILTER DISPLAY
- ● Vertical
- Horizontal

VIEW
- ✓ Show Folders
- ✓ Show Empty Folders
- ✓ Show Favorites
- ✓ Filter Recursively
- ✓ Show All Folder
- ✓ Organize Folders
- Path View Filters
- ✓ Show Sources Panel

CONTENT
- ✓ Show C++ Classes
- ✓ Show Developers Content
- Show Engine Content
- ✓ Show Plugin Content
- Show Localized Content

SEARCH
- ✓ Search Asset Class Names
- ✓ Search Asset Path
- ✓ Search Collection Names

THUMBNAILS
- Thumbnail Size
- Thumbnail Edit Mode
- ✓ Real-Time Thumbnails

Index

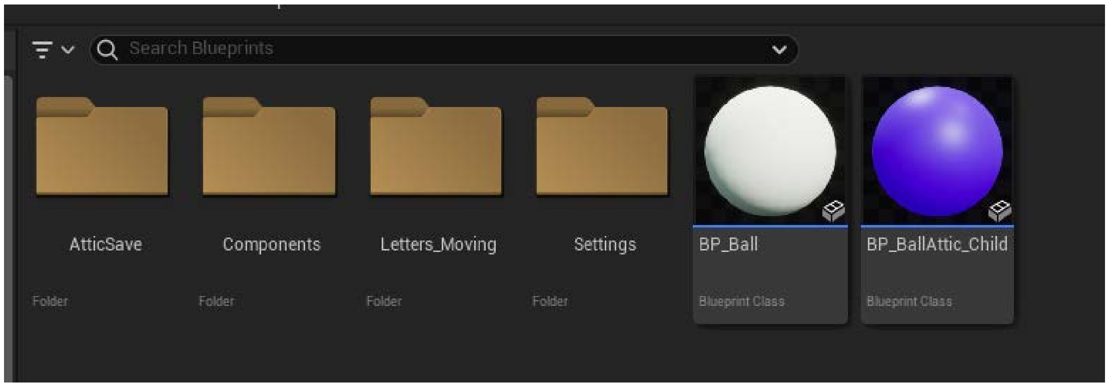
We've put all the content for the demo in a DemoContent folder for you for to help with integration testing. You can copy just the StreamingPortal plugin as seen in Tutorial 1 to your project and away you go.

Whether you start a project based off this project or integrate it with an existing one this should help. You can also just delete the content, once you have your own content and characters in, making cleanup easy.

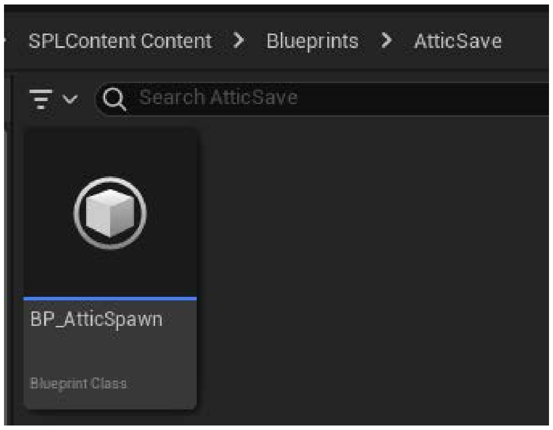
The Demo was generated by starting with the default 3rd Person example. Minor changes to BP_ThirdPersonCharacter and BP_ThirdPersonGameMode have been made purely for the purpose of the demo and is not needed for the StreamingPortal system.

- **DemoContent/Blueprint - Blueprints used in the demo**
    - BP_Ball - The physics ball
    - BP_BallAttic_Child - The ball in the attic. This is done to deal with streaming levels.

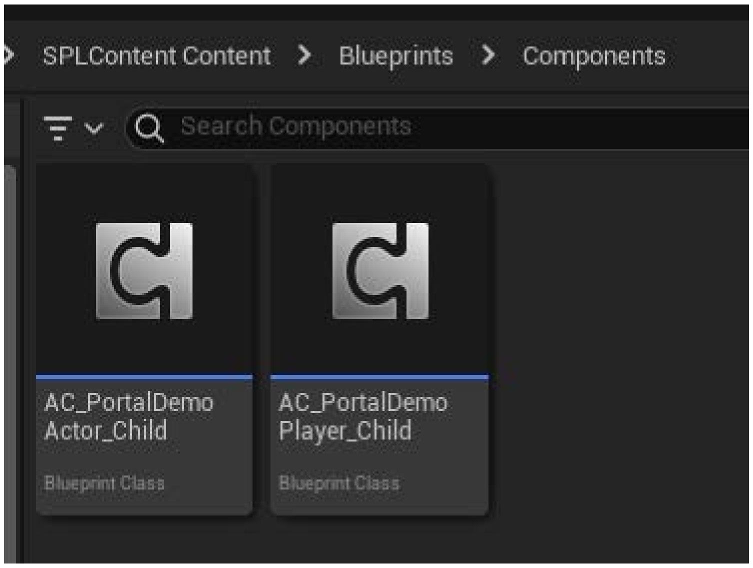
When a level is unloaded and reloaded it is "reset". For this demo we didn't want the ball in the attic to come back if reloaded. - An example of how to deal with level states. Also the ball belongs to the Attic level. When that level is hidden EndPlay gets run on everything in it and would make the ball disappear. To persist the ball and helper function was created to respawn the ball in the persistent level.



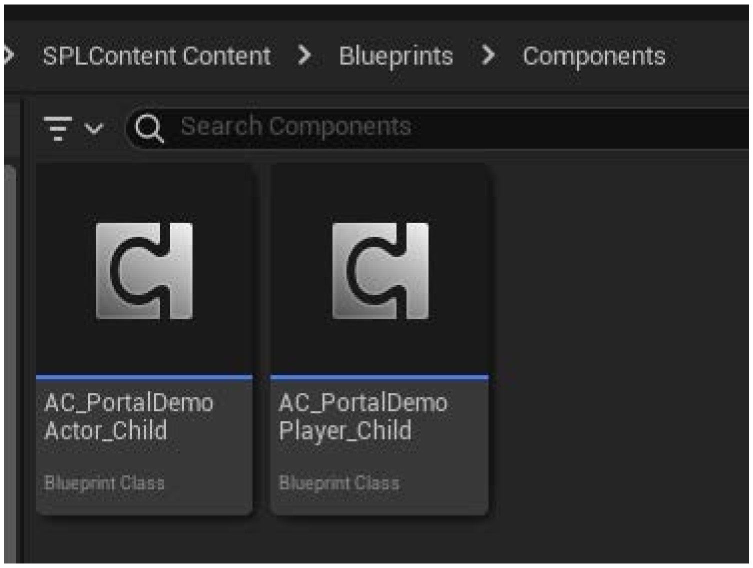
DemoContent/AtticSave/BP_AtticSpawn - The savegame that helps deal with the ball in the attic only spawning once per game session.



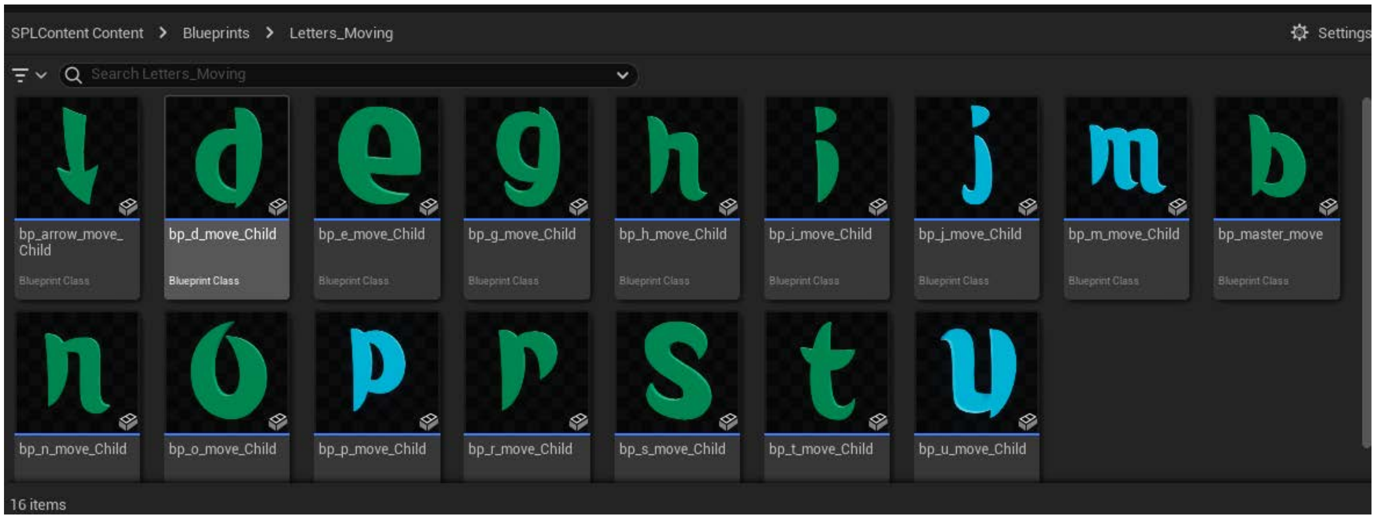
- **DemoContent/Components/AC_PortalDemoActor_Child**
- A child BP of AC_PortalActor modified specifically for the demo



- **DemoContent/Components/AC_PortalDemoPlayer_Child**
- A child BP of AC_PortalPlayer modified specifically for the demo

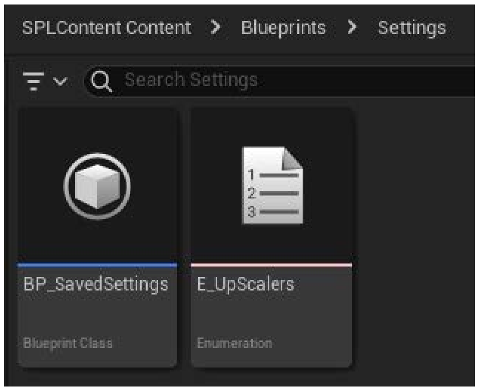


- **DemoContent/Letters_Moving** - the various letters that drift up and down in the demo to draw the player's attention.

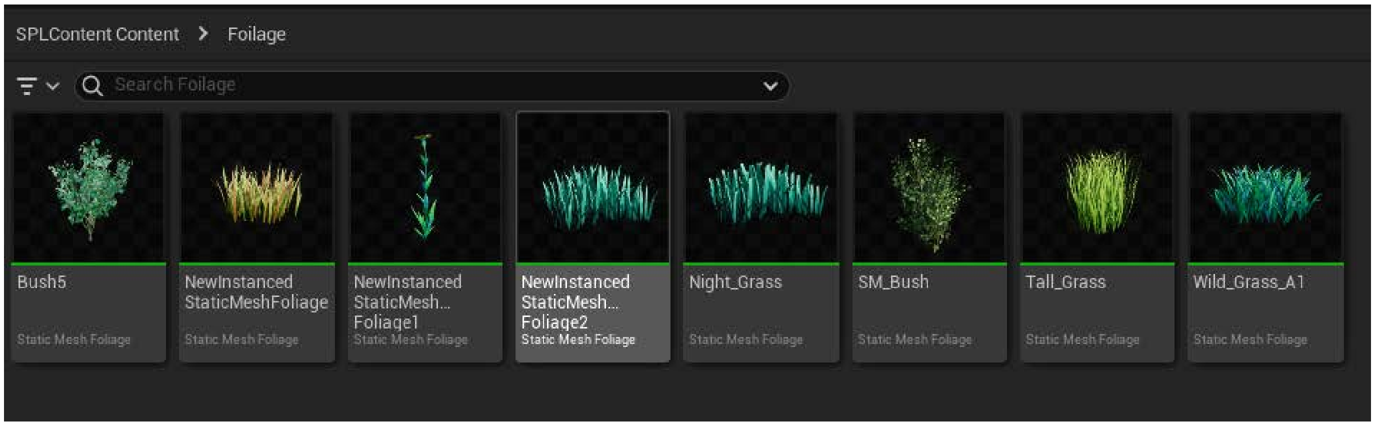


- **DemoContent/Settings/BP_SavedSettings -** SaveGame for the quality, screen scale and upsamplers settings.

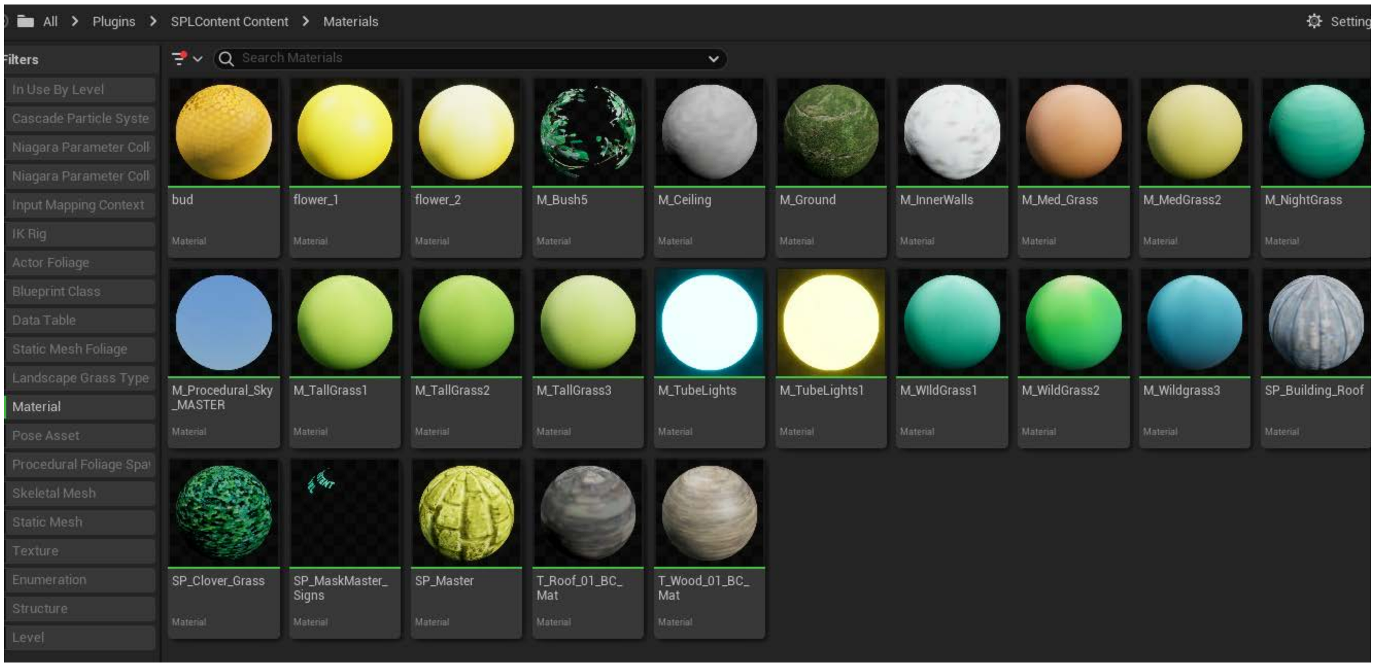
- **DemoContent/Settings/E_UpScalers** - Enums for the UpSamplers

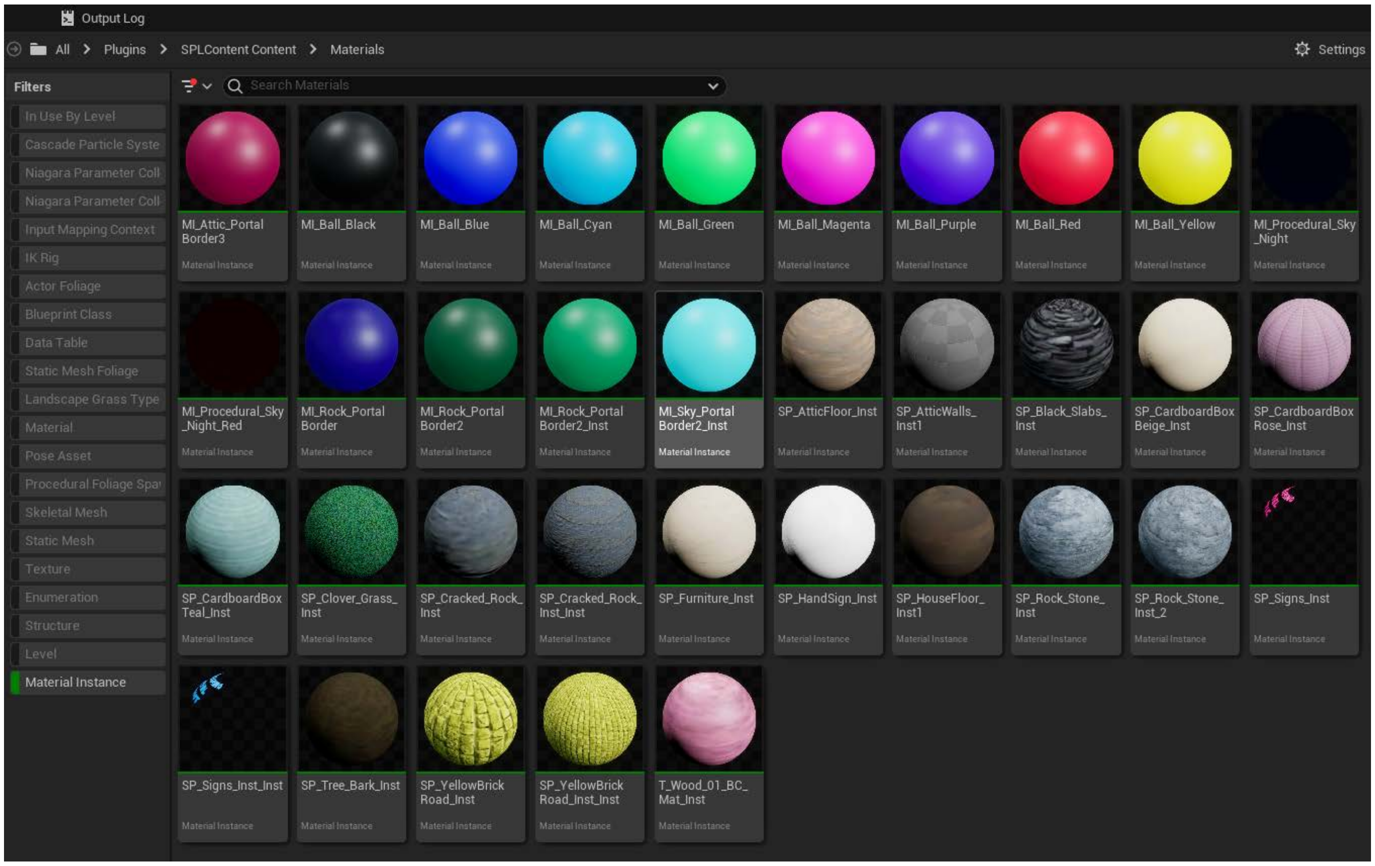


- **DemoContent/Foliage**

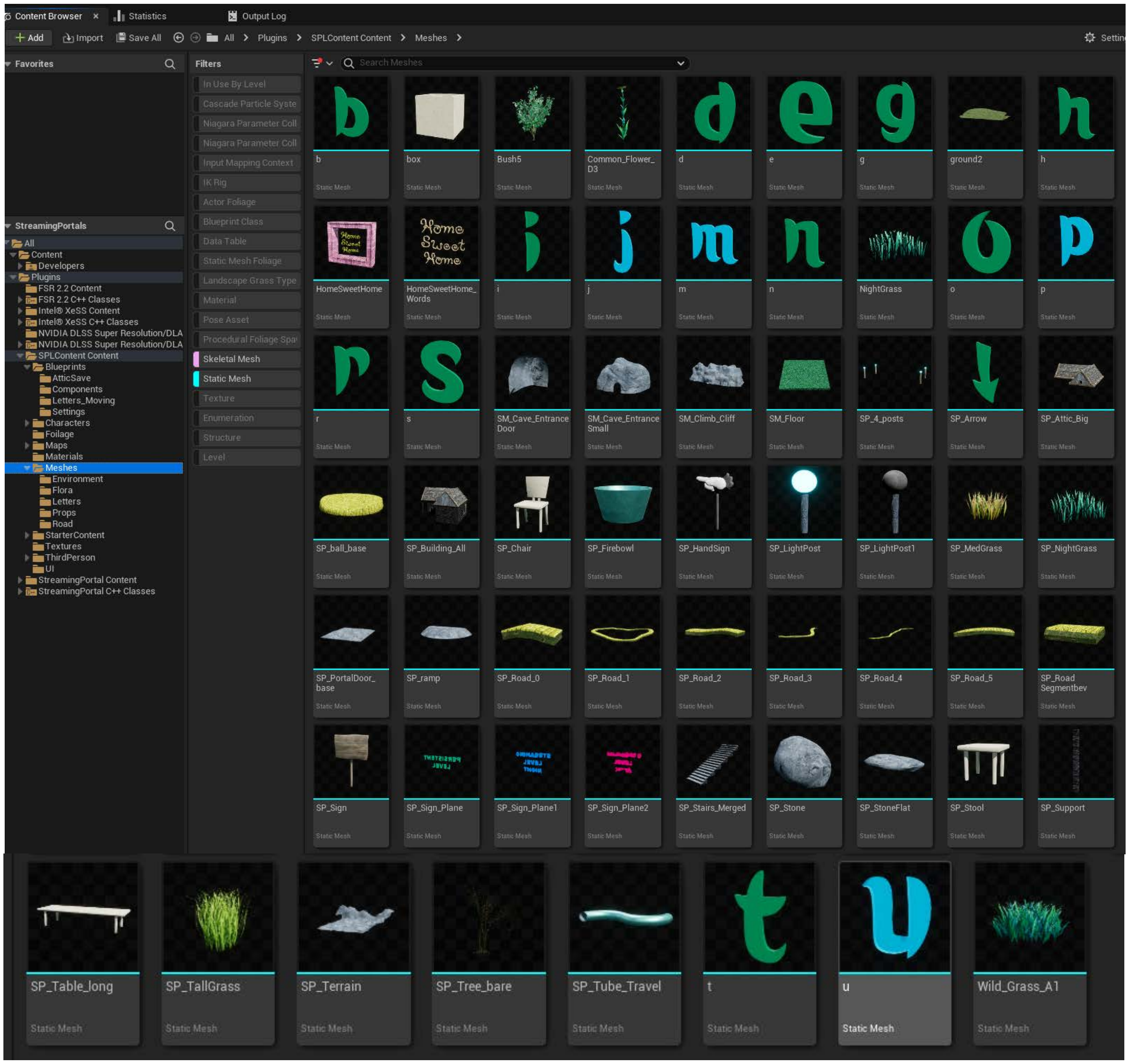

SPLContent Content > Foliage
Search Foliage
Bush5
NewInstanced StaticMeshFoliage
NewInstanced StaticMesh... Foliage1
NewInstanced StaticMesh... Foliage2
Night_Grass
SM_Bush
Tall_Grass
Wild_Grass_A1
Static Mesh Foliage

- **DemoContent/Materials**



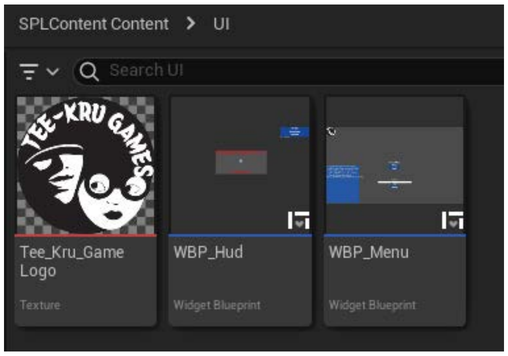


- **DemoContent/Meshes**



- **DemoContent/Textures**
  - Most textures have been virtualized that could.

- **DemoContent/UI**



Performance

As stated in the introduction, we are no experts but have learned from our own project and passion for diving into Unreal. These are the best settings that seem to work well for our project which has very dense environments and moderately complex character models. We do use Clear Coat on the characters themselves and some environment objects that we want to "pop".

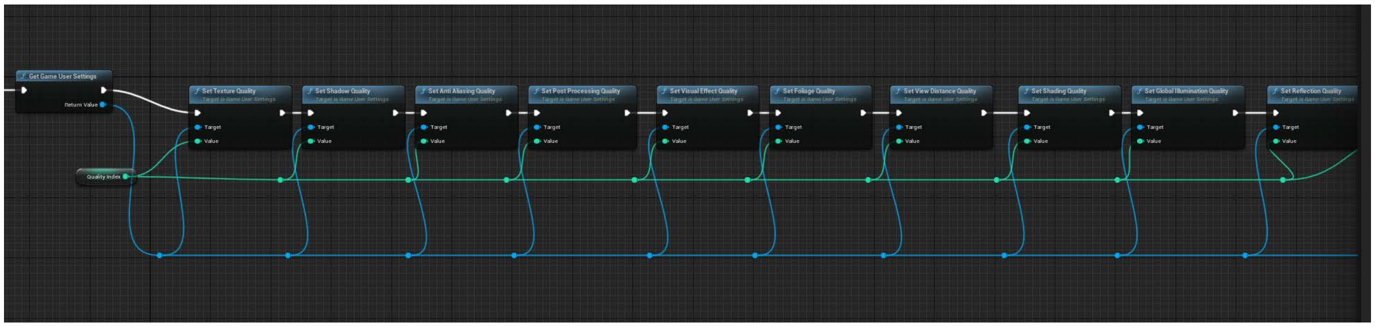
Currently we can run our own project on Steam Deck and Android devices.

First the Config folder

- **DefaultDeviceProfiles.ini**
  - Mostly taken from the Lyra Starter game with some minor texture size changes for Android. And windows max texture sizes set to 1024.
- **DefaultEngine.ini**
  - Changes have been marked with ;TeeKru Change.
  - Changes were made for performance and setting up the UpSamplers so you can switch between them at RunTime. I've seen a few games start doing this which is a great! As PC hardware and android can very.
- **DefaultGameUserSettings.ini**
  - This is what tells the shipping release how to first start up on the players PC
- **And finnly DefaultScalability.ini**
  - The PerfIndexThresholds_* taken from Lyra Start Game.
  - Tweaked Texture pool sizes for a more modern style game (Default settings are fairly low)
  - Lowered r.Streaming.MipBias in each section to help with streaming objects in smoother
  - Changed r.Streaming.MaxNumTexturesToStreamPerFrame in each section for streaming in levels
  - All changes in here are for better performance and have worked well in our project so far.

Game Runtime Settings

All performance code has been put into ThirdPerson/Blueprints/BP_ThirdPersonGameMode blueprint. And has been commented. On Begin Play the blueprint will determine if it needs to run the benchmark test or not. (Only does this once) and will always run the functions SetQuality and EnableUpScaller



The blueprint will save the quality index you select or the one the benchmark selected. You can change this by pressing 1 in the game. It will also set the screen scale which you can change by pressing 2. This uses the PerfIndexThresholds_* settings in the DefaultScalability.ini

Generating your own PSOs

What is a PSO Cache?

Pipeline State Objects. In short, the thing, when missing, causes chunking or hitching in your game.

https://docs.unrealengine.com/5.2/en-US/optimizing-rendering-with-pso-caches-in-unreal-engine/

Observation: I've noticed when testing shipping builds before I configured our own project A.L.A.R.M. to use PSO, it would create a PSO file in the %appdata%..\Local\<Your app> location. As we test shipping builds a lot and things change over time, if you don't delete this file after significant changes, performance can degrade. A stale cache.

Theory: It's always best to package your stuff with a PSO cache

This project has been setup to generate and use a PSO Cache from the above link. This is section will show you how to use this project to generate your own PSO Cache and build it into a shipping release. There is about 10 steps from reading the above Unreal Documentation.

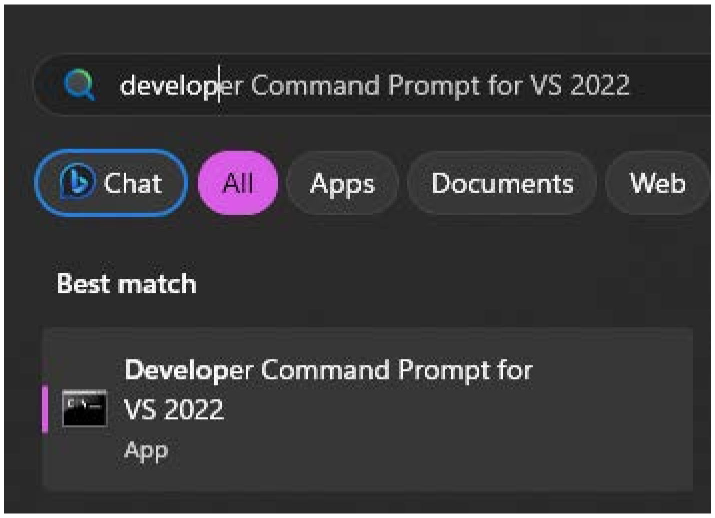
A .bat file has been provided to help you create builds or you can do it through the Unreal Editor UI, which ever your like.

Package.bat

The bat file is called Package.bat and assumes Unreal is installed in it's default location and your project is on C: drive. Open the batch file in your favorite text editor and change the values to your liking.

```
REM ************ Change these Values ************
REM The location of your Unreal project.
set OUTPUT=C:/SP_Build
REM The location where the builds are going to be stored.
set PROJECT="C:/StreamingPortals/StreamingPortals.uproject"
REM Base Engine Folder Location
set ENGINEDIR="C:/Program Files/Epic Games/UE_5.2/Engine"
REM ************ Change these Values ************
```

Then open a Developer Command prompt

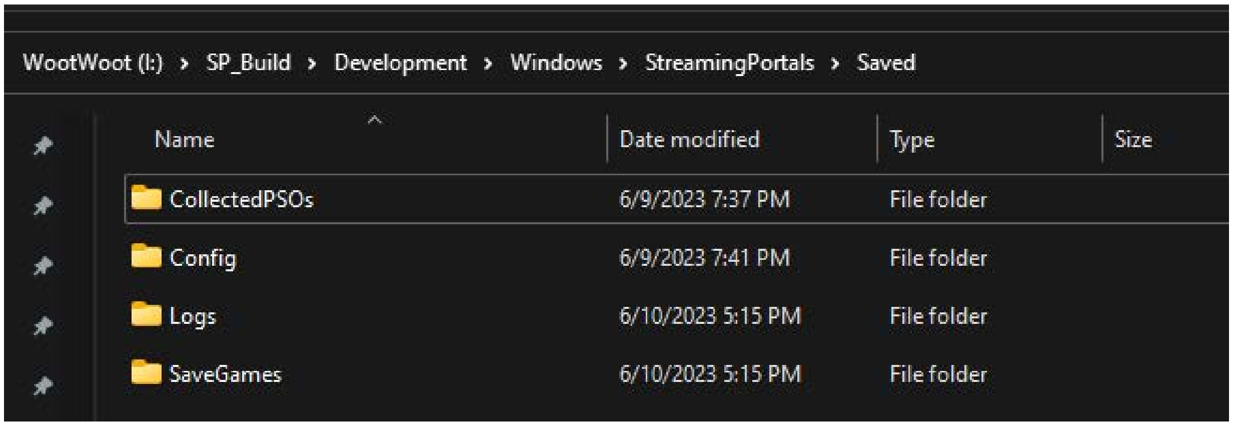

developer Command Prompt for VS 2022
Chat
All
Apps
Documents
Web
Best match
Developer Command Prompt for VS 2022
App

Navigate to project file and you can run

Package - No options will build a development build by default
Package debug - Builds a DebugGame version
Package test - Builds a Test version
Package ship - Builds a shipping version with -distribution so debugging symbols are stripped off
Package debugengine - Only if you really want to

Building the PSO Cache

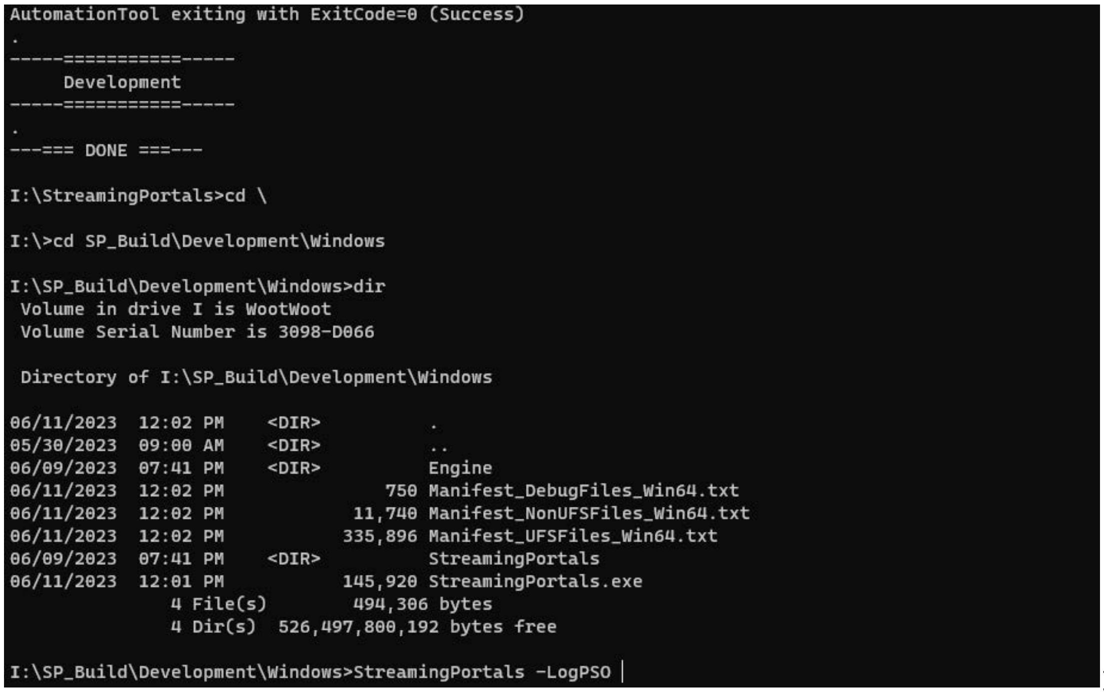
- **Step 1** - do a development build. (Package dev or just Package)
- **Step 2** - Run the game once and exit. This will create the saved game folder for you. You must create the CollectedPSOs folder yourself. The engine will not and you will not get any PSO Cache data.
- **Step 3** - Navigate to the saved game folder
  - C:\SP_Build\Development\Windows\StreamingPortals\Saved from the Package.bat as an example
- Create a folder CollectedPSOs so it looks like this (My drive is I:)



WootWoot (I:) › SP_Build › Development › Windows › StreamingPortals › Saved

Name	Date modified	Type	Size
CollectedPSOs	6/9/2023 7:37 PM	File folder	
Config	6/9/2023 7:41 PM	File folder	
Logs	6/10/2023 5:15 PM	File folder	
SaveGames	6/10/2023 5:15 PM	File folder	

- **Step 4** - start the game from the command prompt or create an alias if you like. It must have the command argument -LogPSO

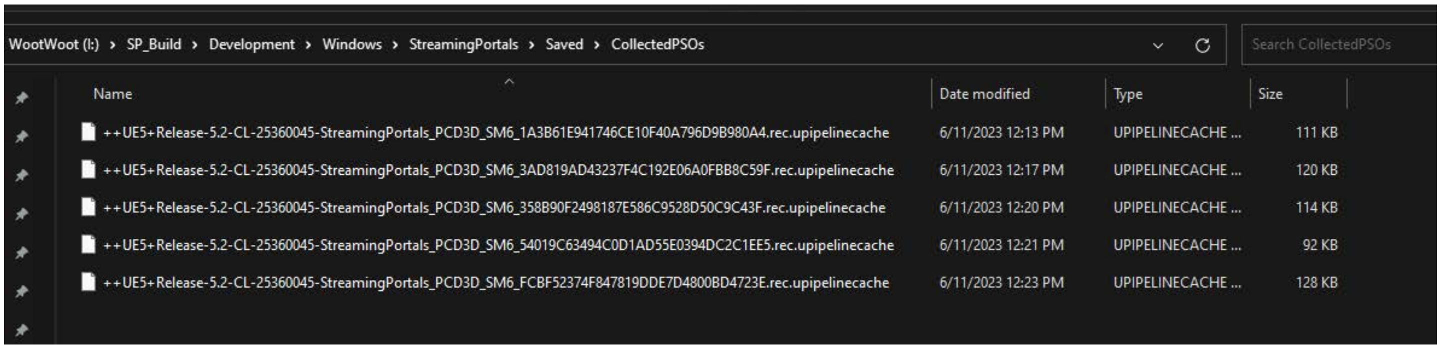


```
AutomationTool exiting with ExitCode=0 (Success)
.
-----==================
     Development
-----==================
.
---=== DONE ===---

I:\StreamingPortals>cd \

I:\>cd SP_Build\Development\Windows

I:\SP_Build\Development\Windows>dir
 Volume in drive I is WootWoot
 Volume Serial Number is 3098-D066

 Directory of I:\SP_Build\Development\Windows

06/11/2023  12:02 PM    <DIR>          .
05/30/2023  09:00 AM    <DIR>          ..
06/09/2023  07:41 PM    <DIR>          Engine
06/11/2023  12:02 PM               750 Manifest_DebugFiles_Win64.txt
06/11/2023  12:02 PM            11,740 Manifest_NonUFSFiles_Win64.txt
06/11/2023  12:02 PM           335,896 Manifest_UFSFiles_Win64.txt
06/09/2023  07:41 PM    <DIR>          StreamingPortals
06/11/2023  12:01 PM           145,920 StreamingPortals.exe
               4 File(s)        494,306 bytes
               4 Dir(s)  526,497,800,192 bytes free

I:\SP_Build\Development\Windows>StreamingPortals -LogPSO
```

Index

- **Step 5** - Now run everywhere you can in the game. Look at everything. Or as the Unreal Documentation says "Execute as many code paths as possible" . You will want to do this once per quality setting and per shader type. I do the quality settings in separate runs. Usually start with Very High, run everywhere and exit the game.
- **Step 6** - (Rinse and Repeat) - Start the game again and run around on the next quality setting, etc (Low, Medium, High, Very High, Ultra)

- Each time you exit you can check for a file in the CollectedPSOs folder. You should see something like below per quality setting.

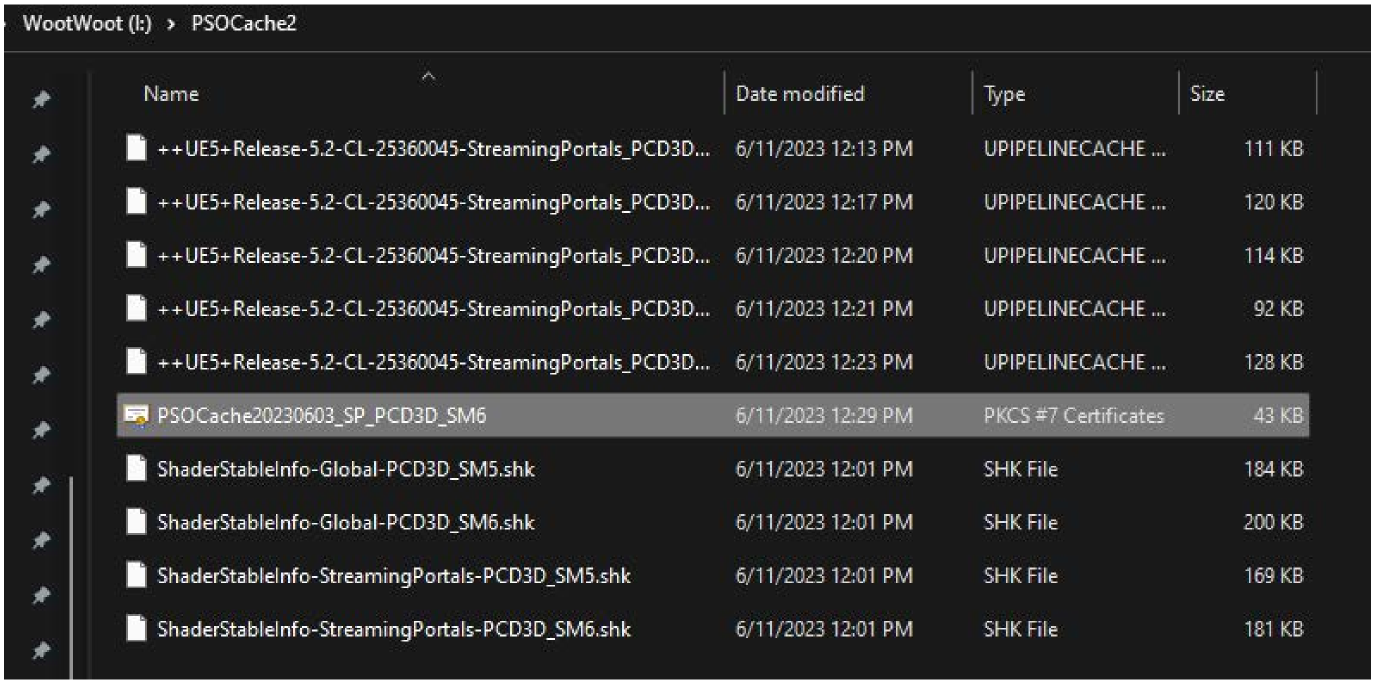


WootWoot (I:) › SP_Build › Development › Windows › StreamingPortals › Saved › CollectedPSOs

Name	Date modified	Type	Size
++UE5+Release-5.2-CL-25360045-StreamingPortals_PCD3D_SM6_1A3B61E941746CE10F40A796D9B980A4.rec.upipelinecache	6/11/2023 12:13 PM	UPIPELINECACHE ...	111 KB
++UE5+Release-5.2-CL-25360045-StreamingPortals_PCD3D_SM6_3AD819AD43237F4C192E06A0FBB8C59F.rec.upipelinecache	6/11/2023 12:17 PM	UPIPELINECACHE ...	120 KB
++UE5+Release-5.2-CL-25360045-StreamingPortals_PCD3D_SM6_358B90F2498187E586C9528D50C9C43F.rec.upipelinecache	6/11/2023 12:20 PM	UPIPELINECACHE ...	114 KB
++UE5+Release-5.2-CL-25360045-StreamingPortals_PCD3D_SM6_54019C63494C0D1AD55E0394DC2C1EE5.rec.upipelinecache	6/11/2023 12:21 PM	UPIPELINECACHE ...	92 KB
++UE5+Release-5.2-CL-25360045-StreamingPortals_PCD3D_SM6_FCBF52374F847819DDE7D4800BD4723E.rec.upipelinecache	6/11/2023 12:23 PM	UPIPELINECACHE ...	128 KB

- **Step 7** - Next Copy these files into a C:\PSOCache2 folder (its what I'm using for this example but you can make it what ever you like)
- **Step 8** - Now go to the project C:\StreamingPortals\Saved\Cooked\Windows\StreamingPortals \Metadata\PipelineCaches and copy the files in there to the C:\PSOCache2 folder
- **Step 9** - Now run the following command at the developer prompt
  - "C:\Program Files\Epic Games\UE_5.2\Engine\Binaries\Win64\UnrealEditor-Cmd.exe" -run=ShaderPipelineCacheTools expand C:\PSOCache2/*.rec.upipelinecache C:\PSOCache2/*.shk C:\PSOCache2/PSOCache20230611_SP_PCD3D_SM6.spc
  - I like to mark my PSO Cache with YYYMMDD.. There is a format to follow see the PSO documentation link at the top of this section.

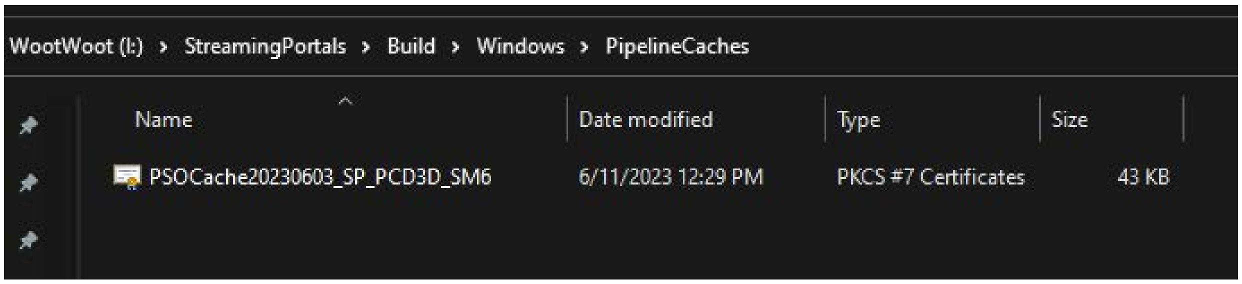
After the command is complete you will see the .spc file



WootWoot (I:) › PSOCache2

Name	Date modified	Type	Size
++UE5+Release-5.2-CL-25360045-StreamingPortals_PCD3D...	6/11/2023 12:13 PM	UPIPELINECACHE ...	111 KB
++UE5+Release-5.2-CL-25360045-StreamingPortals_PCD3D...	6/11/2023 12:17 PM	UPIPELINECACHE ...	120 KB
++UE5+Release-5.2-CL-25360045-StreamingPortals_PCD3D...	6/11/2023 12:20 PM	UPIPELINECACHE ...	114 KB
++UE5+Release-5.2-CL-25360045-StreamingPortals_PCD3D...	6/11/2023 12:21 PM	UPIPELINECACHE ...	92 KB
++UE5+Release-5.2-CL-25360045-StreamingPortals_PCD3D...	6/11/2023 12:23 PM	UPIPELINECACHE ...	128 KB
PSOCache20230603_SP_PCD3D_SM6	6/11/2023 12:29 PM	PKCS #7 Certificates	43 KB
ShaderStableInfo-Global-PCD3D_SM5.shk	6/11/2023 12:01 PM	SHK File	184 KB
ShaderStableInfo-Global-PCD3D_SM6.shk	6/11/2023 12:01 PM	SHK File	200 KB
ShaderStableInfo-StreamingPortals-PCD3D_SM5.shk	6/11/2023 12:01 PM	SHK File	169 KB
ShaderStableInfo-StreamingPortals-PCD3D_SM6.shk	6/11/2023 12:01 PM	SHK File	181 KB

- Copy this file to the Project folder: C:\StreamingPortals\Build\Windows\PipelineCaches



WootWoot (I:) › StreamingPortals › Build › Windows › PipelineCaches

Name	Date modified	Type	Size
PSOCache20230603_SP_PCD3D_SM6	6/11/2023 12:29 PM	PKCS #7 Certificates	43 KB

Step 10 - Now do a shipping build. (Package ship)
- You have now generated PSO Cache for this project.

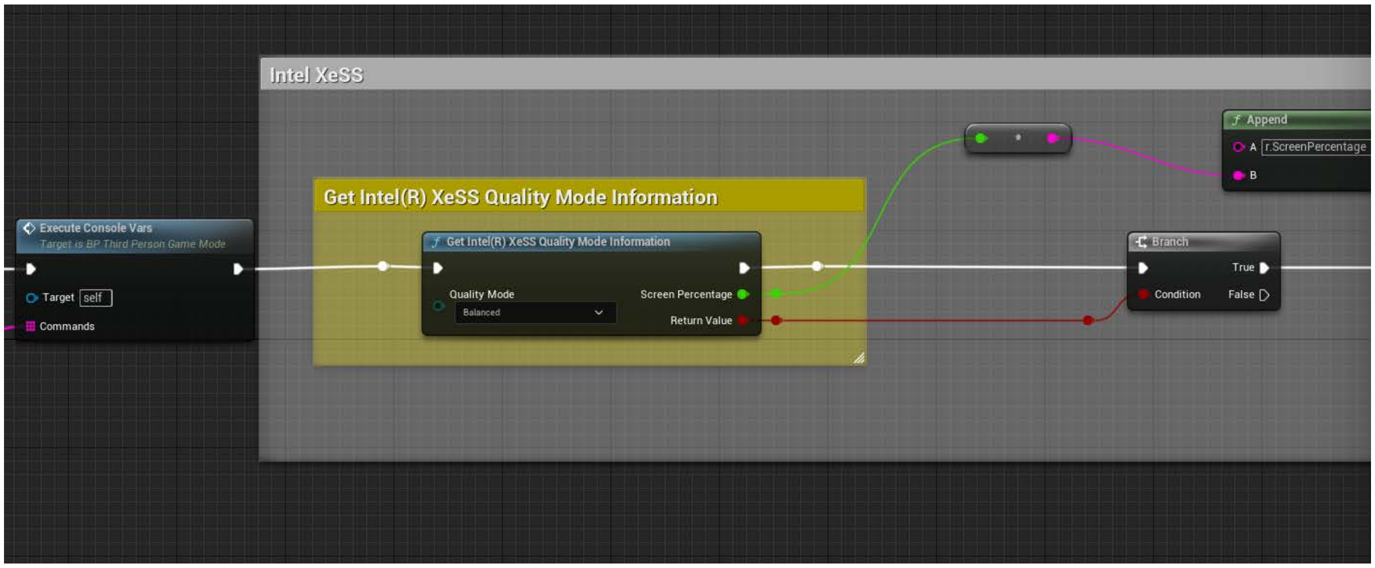
Note if your shipping build fails with Invalid Keys

- If your shipping fails with invalid keys, you may have dirty cache files. Before doing a development build to create the PSO cache ensure the following folders have NO files
    - C:\SP_Build\Development\Windows\StreamingPortals\Saved\CollectedPSOs
    - C:\StreamingPortals\Saved\Cooked\Windows\StreamingPortals\Metadata\PipelineCaches
    - C:\PSOCache2
    - C:\StreamingPortals\Build\Windows\PipelineCaches

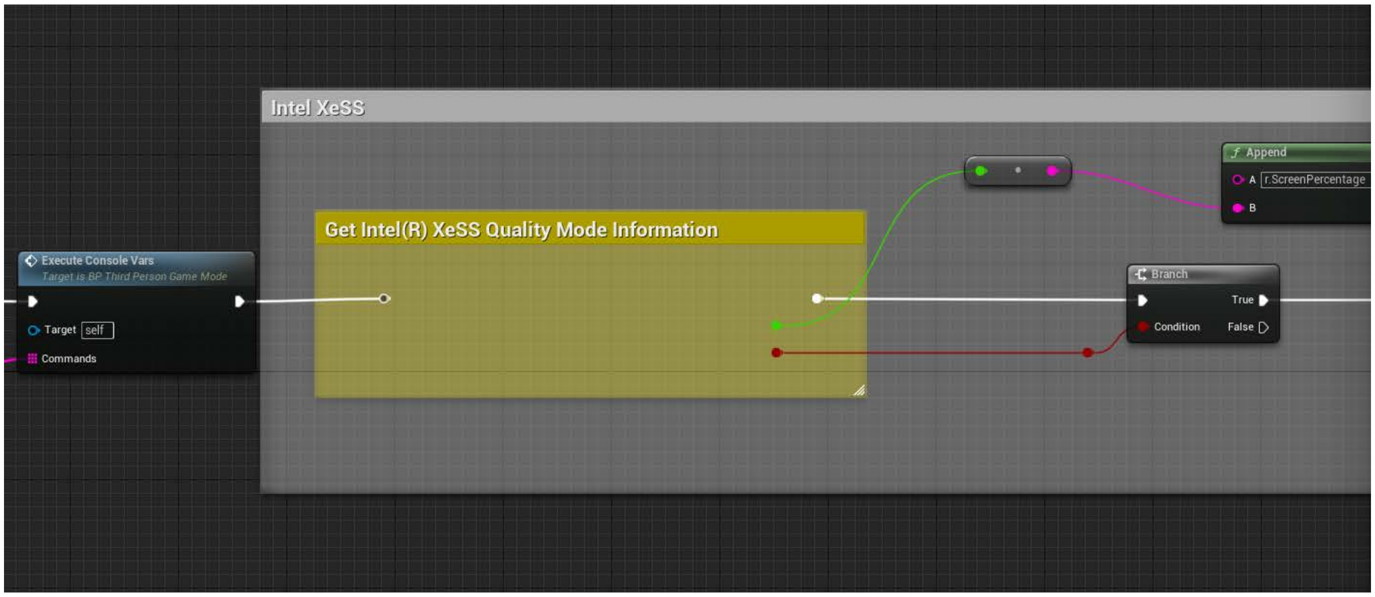
- Do your development build again and generate your PSOs

UpSampler Plugins

The latest Intel XeSS requires a blueprint node. To enable the Intel XeSS, once you've added the plugin to the project, go to the SPLContent/ThirdPerson/Blueprints/ BP_ThirdPersonGameMode blueprint. In the function EnableIntelXeSS add in the node below:



The project ships with it blank like this:



Each UpSampler has it's own function to enable or disable it. Mostly with just console variables.

All of these work with Unreal 5.2

Source AMD Fidelity FX 1: https://github.com/teella/fsr-amd-ue

info: https://gpuopen.com/learn/ue4-fsr/

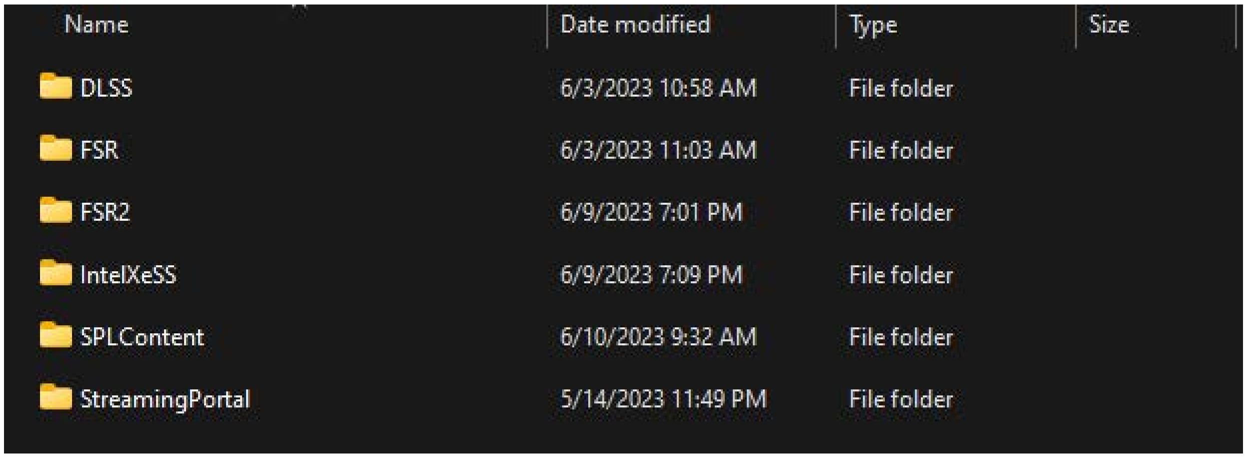
AMD Fidelity FX2: https://gpuopen.com/learn/ue-fsr2/

Intel XeSS: https://github.com/GameTechDev/XeSSUnrealPlugin click Releases towards the bottom

Nvidia DLSS: https://developer.nvidia.com/rtx/dlss/get-started see Download the Unreal Engine Plugin section

We've provided a download link here for all the UpScales. Just unzip it and add it to the StreamingPortals Plugins folder. It looks like this:

https://teekrugames.com/SPL/Plugins.zip



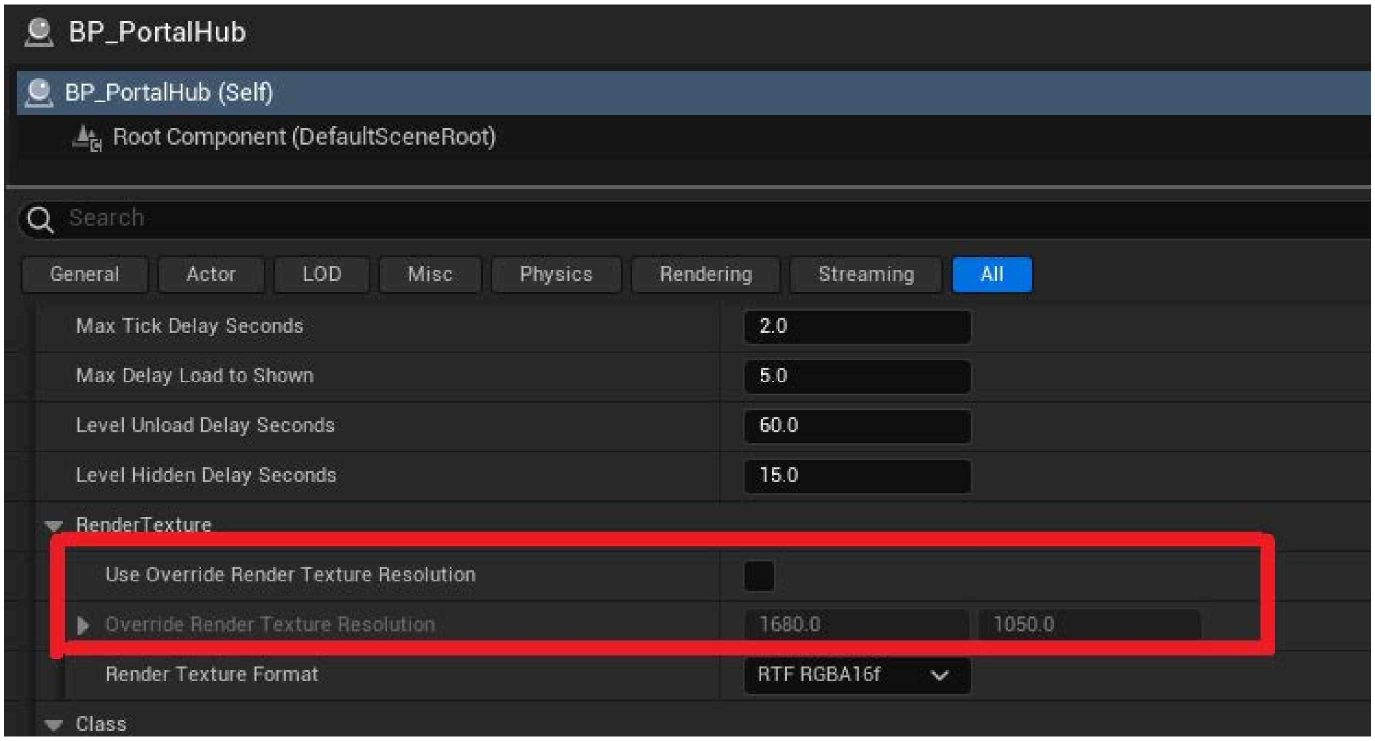
Name	Date modified	Type	Size
DLSS	6/3/2023 10:58 AM	File folder	
FSR	6/3/2023 11:03 AM	File folder	
FSR2	6/9/2023 7:01 PM	File folder	
IntelXeSS	6/9/2023 7:09 PM	File folder	
SPLContent	6/10/2023 9:32 AM	File folder	
StreamingPortal	5/14/2023 11:49 PM	File folder	

Then you can run the project and everything is good to go. The Nvidia DLSS one is not included as Nvidia wants you to agree to their terms to download it. Out of the box, it will not compile for 5.2 but can be fixed up fairly easily.

Version 2.0 New Stuff

Override Render Texture Resolution Settings

PortalHub settings configured to override resolution settings for render textures. This adjustment is typically required only in cases where the automatic resolution check fails to function correctly.


BP_PortalHub

BP_PortalHub (Self)

Root Component (DefaultSceneRoot)

Max Tick Delay Seconds	2.0
Max Delay Load to Shown	5.0
Level Unload Delay Seconds	60.0
Level Hidden Delay Seconds	15.0
RenderTexture	
Use Override Render Texture Resolution	☐
Override Render Texture Resolution	1680.0 1050.0
Render Texture Format	RTF RGBA16f
Class	

Added Shack Example to the demonstration level

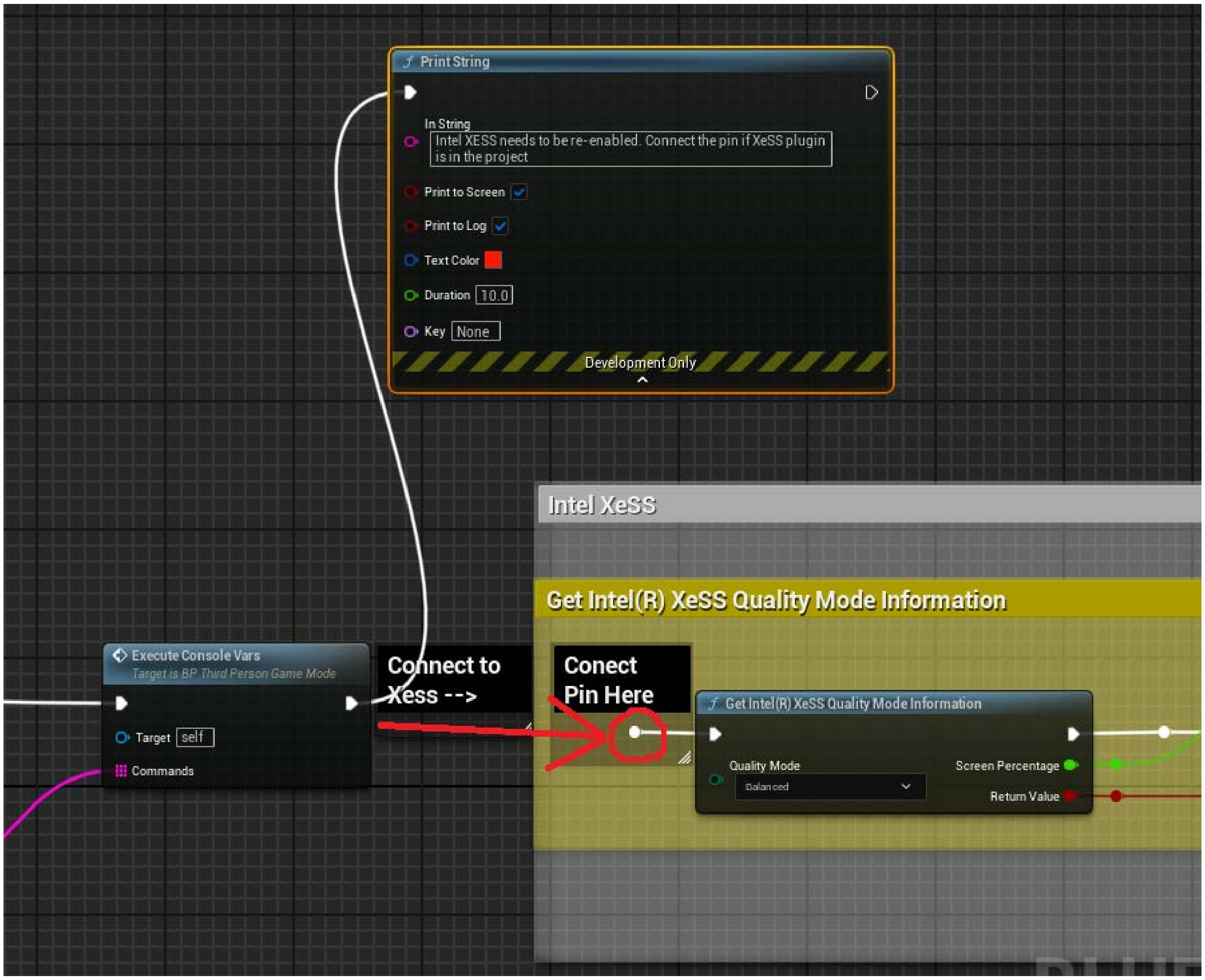
The Shack Example is a unique structure designed to appear small from the outside but spacious within. It seamlessly integrates into the existing level without loading a skysphere. Notably, it features a functional window allowing both interior and exterior views.

It's important to observe that the window portals are configured with the 'Use Own Render Texture' option enabled. This ensures smooth functionality, allowing both the door and windows to remain active simultaneously. Additionally, the interior render distance is set to 3000, facilitating clear views through the door or window from any position within the shack

iggeron the inside

Fix for Example Project

Disconnected the pin to the Intel XeSS Quality Mode Info. If you add the XeSS plugin to your project, simplly hook this back up in BP_ThirdPersonGameMode.

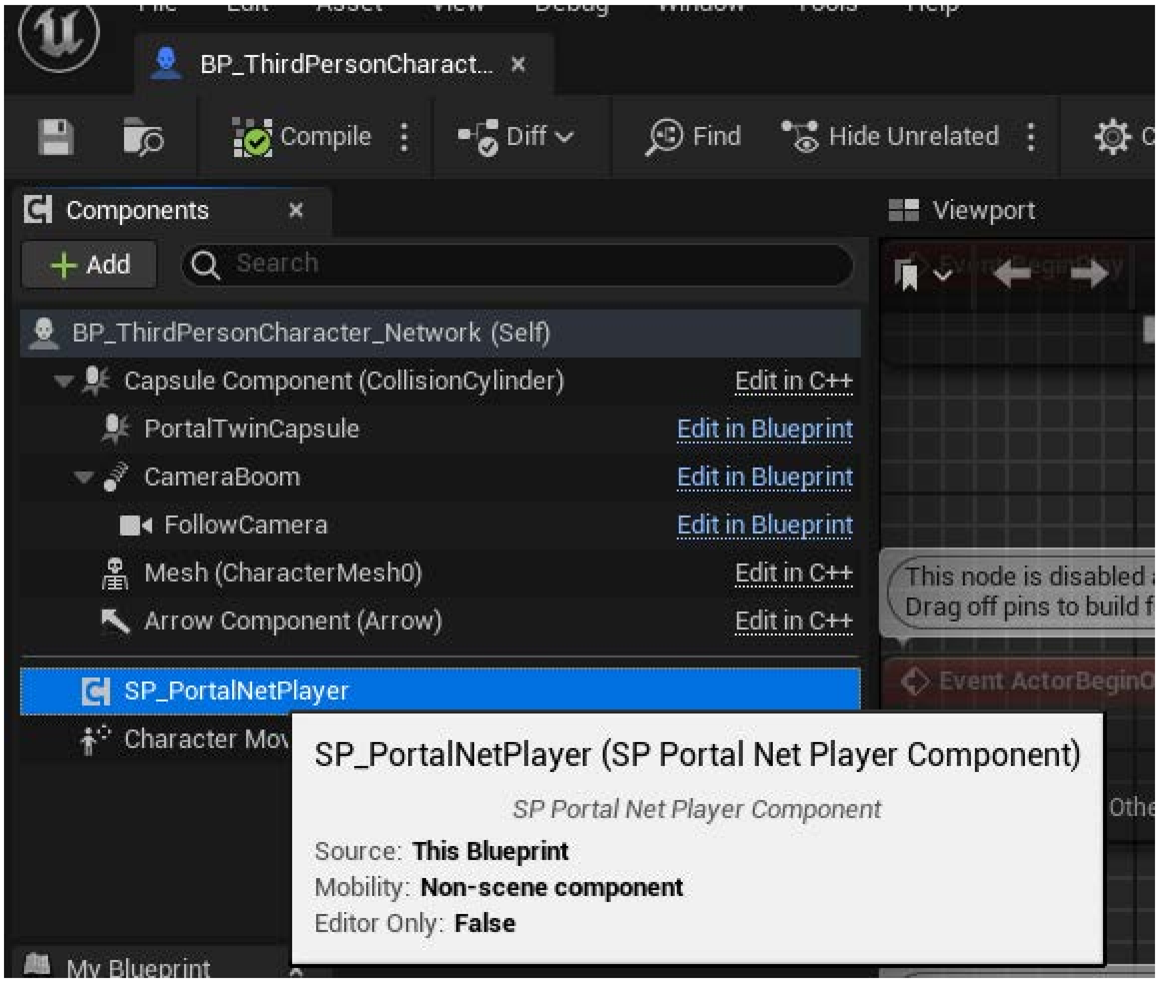


Networked Streaming Level Portals is now supported

Quick Start:

Just add a couple of portals and include the SL_PortalNetPlayerComponent in your main player character blueprint or class make sure Actor Replicates is true. That's all you need!

See Image Below



SP_PortalNetPlayer (SP Portal Net Player Component)

SP Portal Net Player Component

Source: **This Blueprint**
Mobility: **Non-scene component**
Editor Only: **False**

How it all works

Streaming Portal Levels adopts a methodology similar to that of 'Level Streaming Volumes' utilized in Unreal Engine, as documented here:
https://docs.unrealengine.com/5.3/en-US/level-streaming-volumes-reference-in-unreal-engine/

However, unlike level streaming volumes, there's no need to enable 'UseClientSideLevelStreamingVolumes' in your world settings when using Streaming Portal Levels. This feature operates independently of server-side loading. In this system, the server maintains all streaming levels continuously loaded, while clients autonomously manage level streaming based on defined volumes.

In C++, the property 'bUseClientSideLevelStreamingVolumes' is not applicable to Streaming Portal Levels and does not need to be configured in your level's world settings.

Furthermore, all Streaming Portal Actors (such as SP_PortalActor, SP_PortalHub, ASP_PortalLevelStreamManager) have their 'Replicates' set to false and 'OnlyRelevantToOwner' properties set to true, and all components (like SP_PortalActorComponent, SP_PortalCameraManagerComponent, SP_PortalPlayerComponent) are configured not to replicate. This setup is crucial as the entire illusion is handled client-side. In order to keep systems in sync a SP_PortalNetPlayerComponent was added and is the sole component set to replicate, managing network communication for the SPL system and ensuring synchronization among simulation proxies.

Notes about Actors

Understanding how Actors behave in various network settings is crucial. One notable aspect is that when a level is "Hidden," the EndPlay function is triggered for all Actors, regardless of their network settings.

However, the behavior differs for BeginPlay. If your Actor is set to Replicate, it's important to note that BeginPlay executes only once when it is initially loaded as part of a streaming level. This is because the server, being the authority, has already executed BeginPlay. Consequently, when the level is hidden and shown again, BeginPlay will not execute upon being shown, necessitating the replication of any necessary states from the server to your remote actors.

Conversely, if your Actor is set to Not Replicate and is configured as 'OnlyRelevantToOwner,' BeginPlay will execute each time the level is shown, but only on the remote system, as the owner is the remote system.

Network settings added to Portal Hub

1) Dynamically Set Net Relevance:
- This feature toggles 'bAlwaysRelevant' and 'NetCullDistanceSquared' when interacting with a portal.

It ensures that other players can see you on either side of the portal.
- Once out of range of the portal (reaching 'Max Capture Update Distance'), these settings revert.

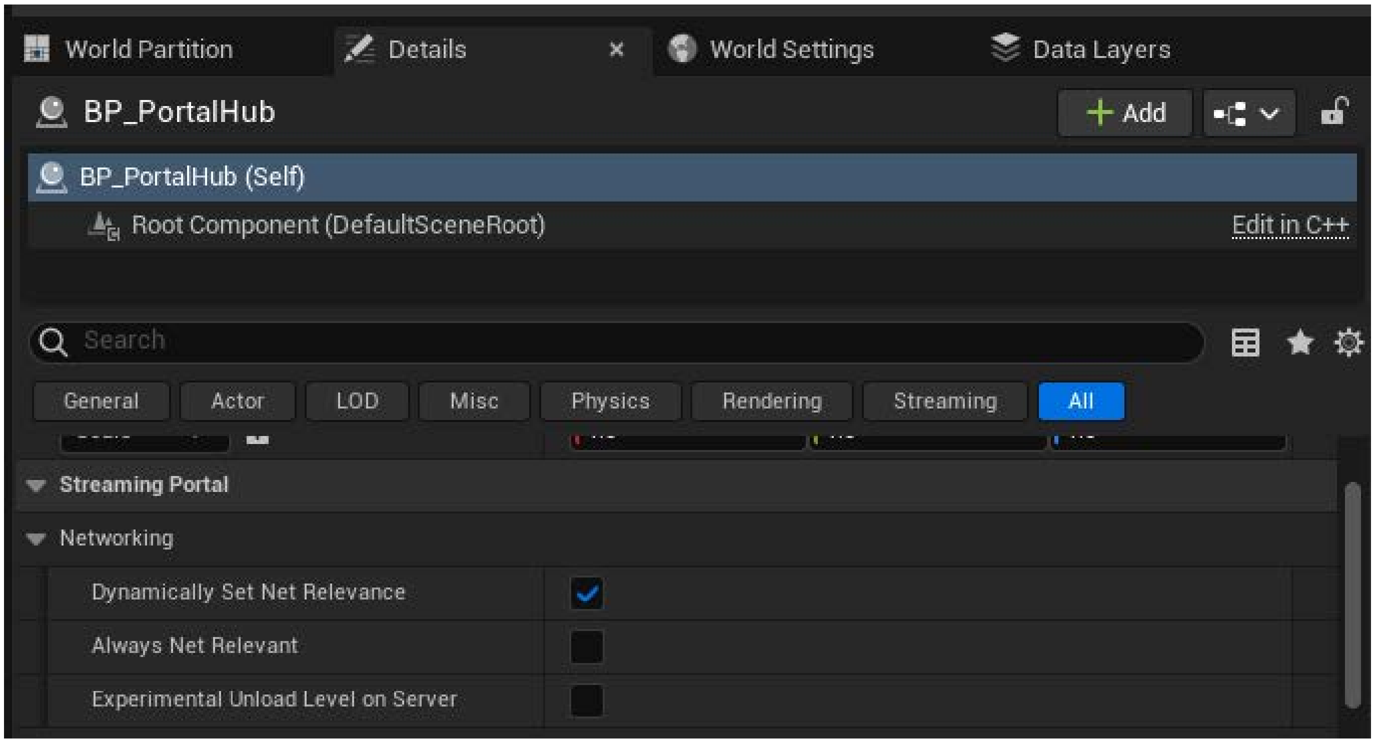
2) Always Net Relevant:
- This setting configures 'bAlwaysRelevant' and 'NetCullDistanceSquared' to zero.

3) Experimental Unload Level on Server:
- This functionality attempts to load and unload streaming levels on the server when they are no longer in use.

- In Unreal Engine 5.2, a warning may appear in the console, but this seems to be an editor-only issue and can be safely ignored. This issue does not occur in version 5.3.

Notably, in Unreal Engine 5.3, when a Level is 'Hidden' in the editor, it disappears from the SceneGraph. However, in version 5.2, a hidden Level remains visible in the SceneGraph, aining the World Settings of the streamed level."

50



Dynamically Set Net Relevance ✓
Always Net Relevant
Experimental Unload Level on Server