Getting Started with SRWorks Experience in Unreal

Release version 0.9.0.3

Contents

Prerequisites

To work with this demo project, it is important to have some understanding of Unreal including the use of the **Motion Controller** class, **Blueprint**, **Level Streaming**, and the **Game Instance** class.

To start, create a folder in your Unreal project folder. Then, extract the zip files to the Plugins folders:

(1) Unreal > Plugin
(2) Unreal > Experience

After this process, your Plugins folder will contain three folders including **RuntimeMeshLoader**, **SRWorks** (Figure 1).

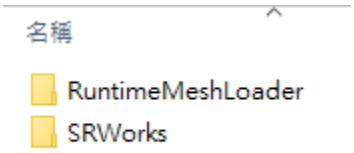

Figure 1. Import

Levels

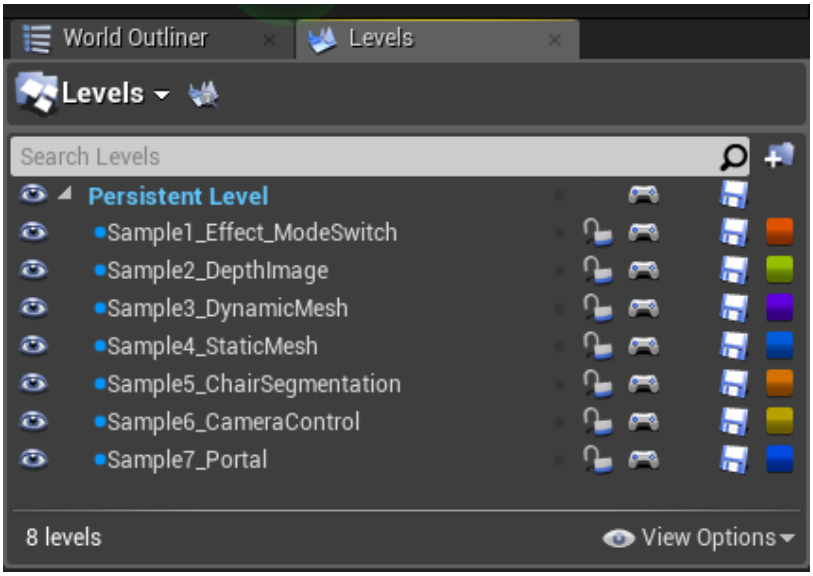

Figure 2. Levels

Open **SRWorks_Experience Content > Maps > Persistent Map**, and then open **Main Menu Bar > Window > Levels**. You can see that all maps have already been added to the Persistent Map (Figure 2).

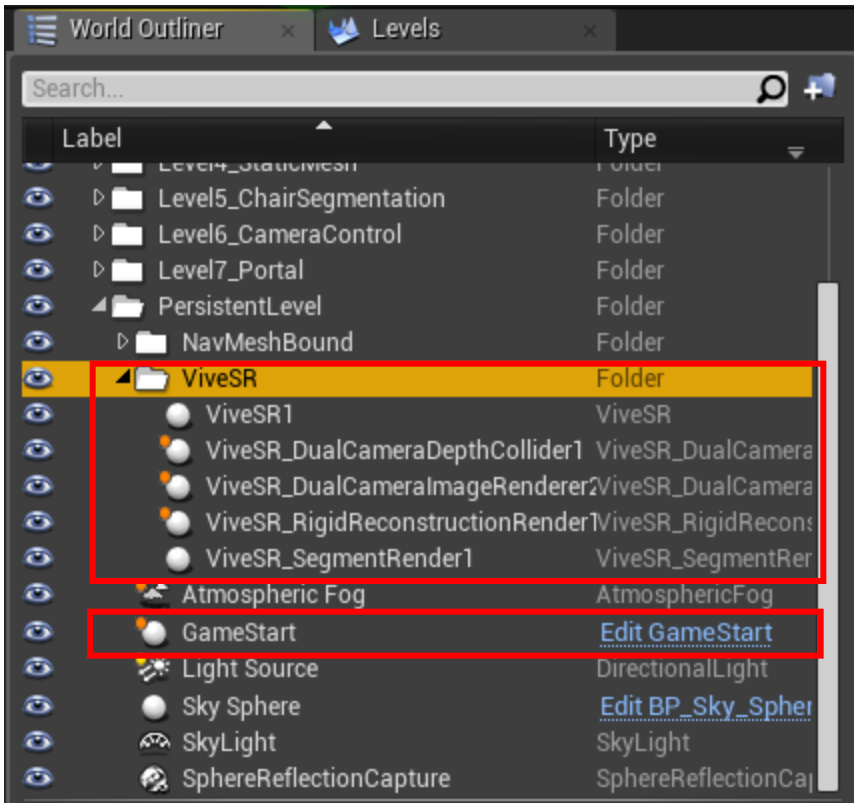

Figure 3. A Set of ViveSR Actors

Persistent Level contains a set of **ViveSR** actors (Figure 3). These actors enable the see-through camera to work with your project in Unreal.

Another Blueprint on Persistent Level is **GameStart**. It serves two purposes:

(1) Stores the actor **ViveSR** as a global reference in the **Experience_SaveGame** to be called cross level.
(2) Loads all the levels upon BeginPlay and sets only the starting level to visible.

Levels and Corresponding Pawns

The seven maps added to the Persistent Map include **Sample1_ Effects_ModeSwitch**, **Sample2_DepthImage**, **Sample3_DynamicMesh**, **Level4_StaticMesh**, **Sample5_ChairSegmentation**, **Sample6_CameraControl** and **Sample7_Portal**. Each map contains a child class of the Pawn Blueprint class **ViveMotionController** and each of the child class demonstrates one to two of the many capabilities of SRWorks.

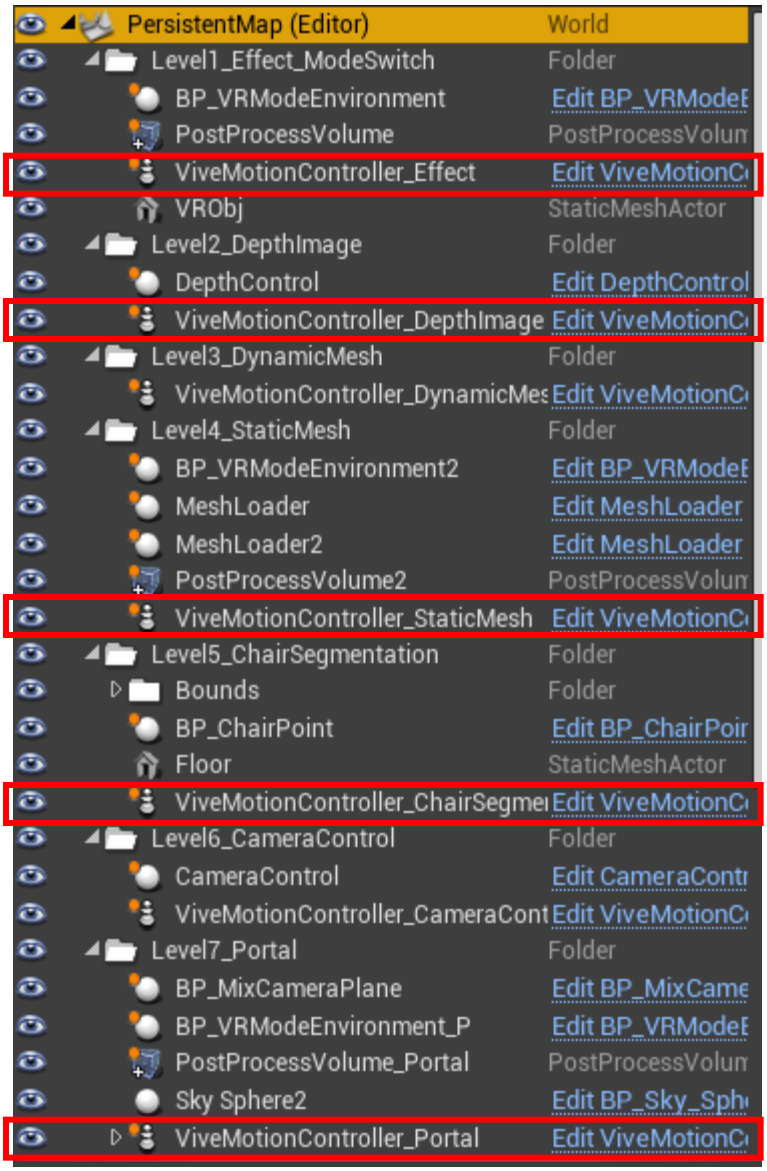

Figure 4. Each Level contains a child Pawn Blueprint class of ViveMotionController.

The **ViveMotionController** Pawn Blueprint serves three purposes:

(1) Initiates tracking of the HMD and the Vive Controllers,
(2) Finds reference of ViveSR from the **Experience_SaveGame**, and
(3) Switches the visibility of levels upon a grip on the controller for level control.

Each of its child class corresponds to the control of only one of the levels (Figure 4). For example, the **Vive_MotionController_Effects** Pawn Blueprint class only exists in **Sample1_Effect_ModeSwitch**. This Blueprint class demonstrates only two of the many capabilities of SRWorks:

(1) Mode Switch - switching camera modes between VR Mode and Mix Mode and
(2) Shader Effects - switching materials on the see-through camera planes.

This pattern applies to the other six levels as well. Each level has a corresponding child Pawn Blueprint class of ViveMotionController that demonstrates part of what SRWorks can do.

Sample1—Effects_ModeSwitch

This sample demonstrates the use of material effects in see-through and the switch between see-through and VR (Figure 5).

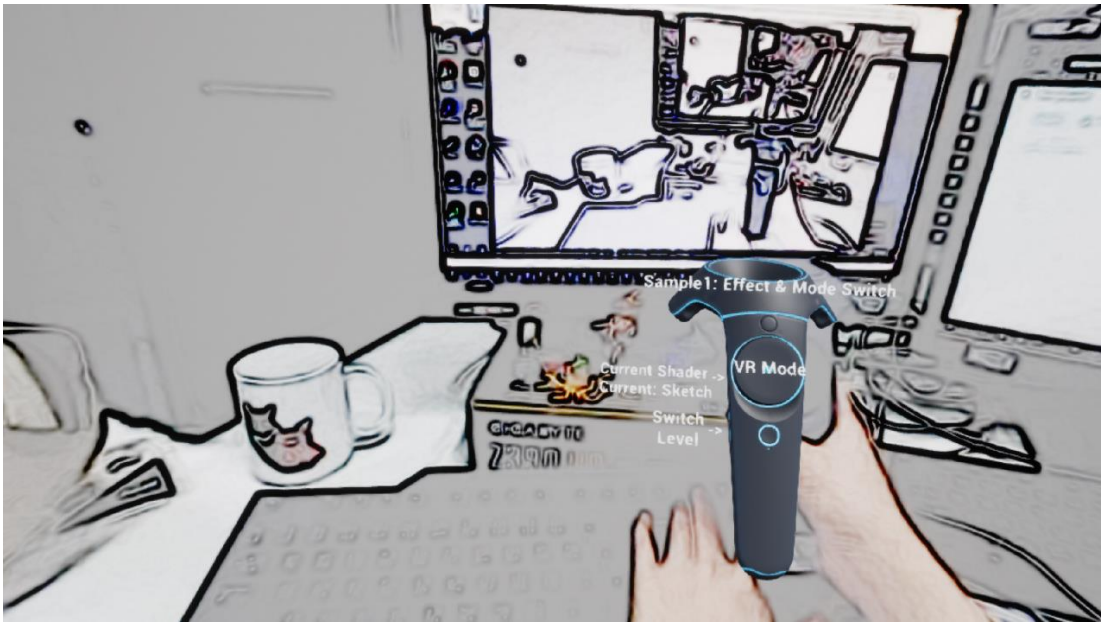

Figure 5. Material Effect (Sample1)

Details

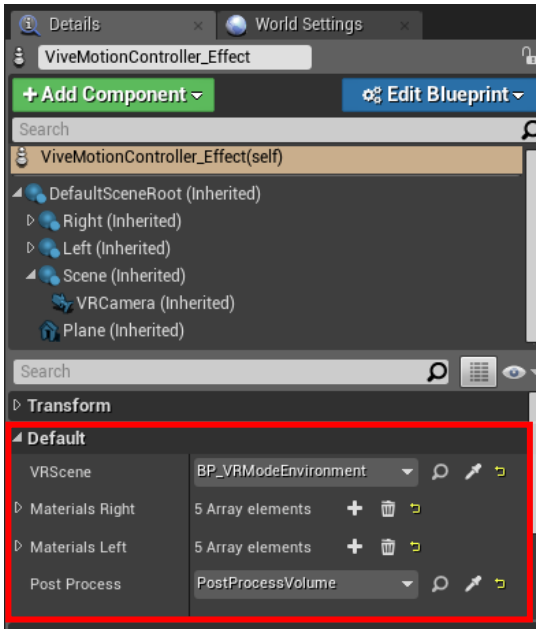

Figure 6. ViveMotionController_Effect (Sample1)

Actor	Setting	Description
ViveMotionController_ Effect	VRScene	Show only Actors in VR Mode.
	Materials Right	Right Eye effect materials.
	Materials Left	Left Eye effect materials.
	Post Process	Post Process in VR Mode.

Table 1. Settings (Sample1)

Controller Input

Input	Description
Thumbstick	Change mode.
Trigger	Change camera plane material (Mix mode).
Grip	Switch to next level.

Table 2. Controller Input (Simple1)

Sample2—Depth Image

This sample demonstrates depth image and its adjustable parameters (Figure 7).

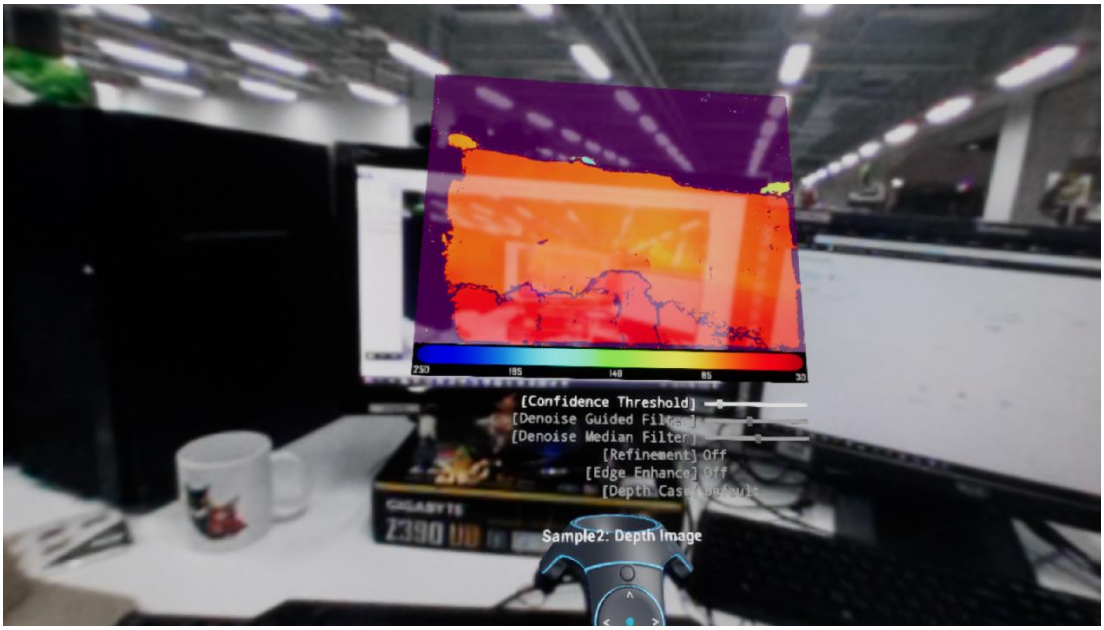

Figure 7. Depth Image (Sample2)

Details

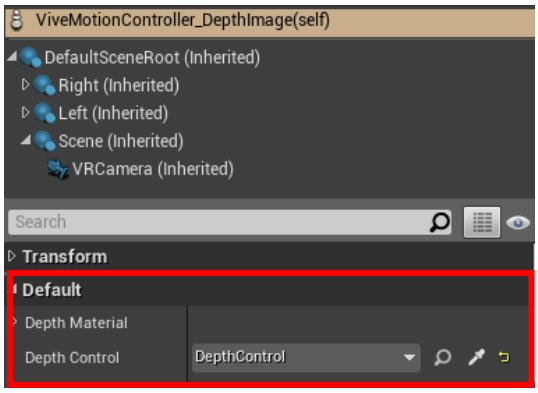

Figure 8. ViveMotionController_DepthImage (Sample2)

Actor	Setting	Description
ViveMotionController_ DepthImage	Depth Material	Depth plane material
	Depth Control	Control Panel value.

Table 3. Settings (Sample2)

Controller Input

Input	Description
Touchpad Up	Select upward (Control panel).
Touchpad Down	Select down (Control panel).
Touchpad Left	Add the selected setting value (Control panel).
Touchpad Right	Decrease the selected setting value (Control panel).
Grip	Switch to next level.

Table 4. Controller Input (Simple2)

Sample3—Dynamic Mesh

This sample demonstrates depth collider that updates every frame (Figure 9).

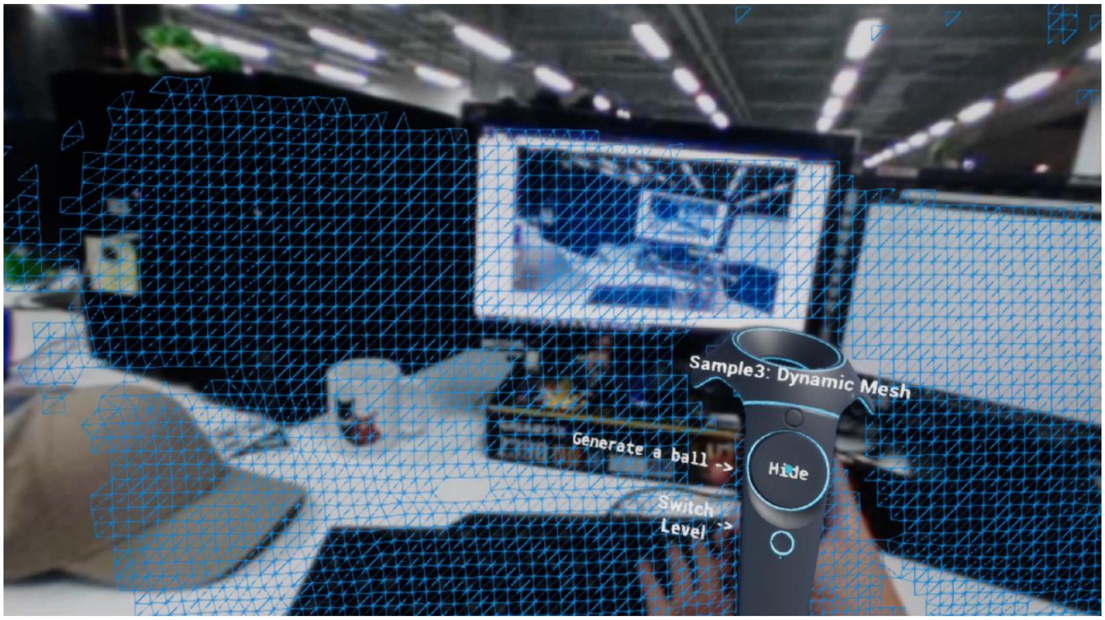

Figure 9. Dynamic mesh(Sample3)

Details

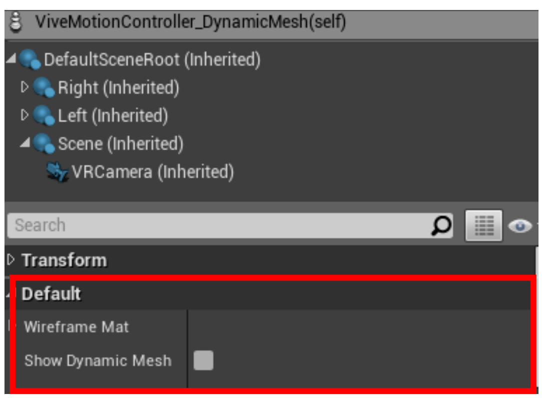

Figure 10. ViveMotionController_DynamicMesh (Sample3)

Actor	Setting	Description
ViveMotionController_ DynamicMesh	Wireframe Mat	Wireframe material on the Mesh.
	Show Dynamic Mesh	Show dynamic mesh at begin or not.

Table 5. Settings (Sample3)

Controller Input

Input	Description
Thumbstick	Hide/Show dynamic mesh.
Trigger	Generate a ball.
Grip	Switch to next level.

Table 6. Controller Input (Simple3)

Sample4—Static Mesh

This sample demonstrates mesh pre-scanning & exportation (Figure 11).

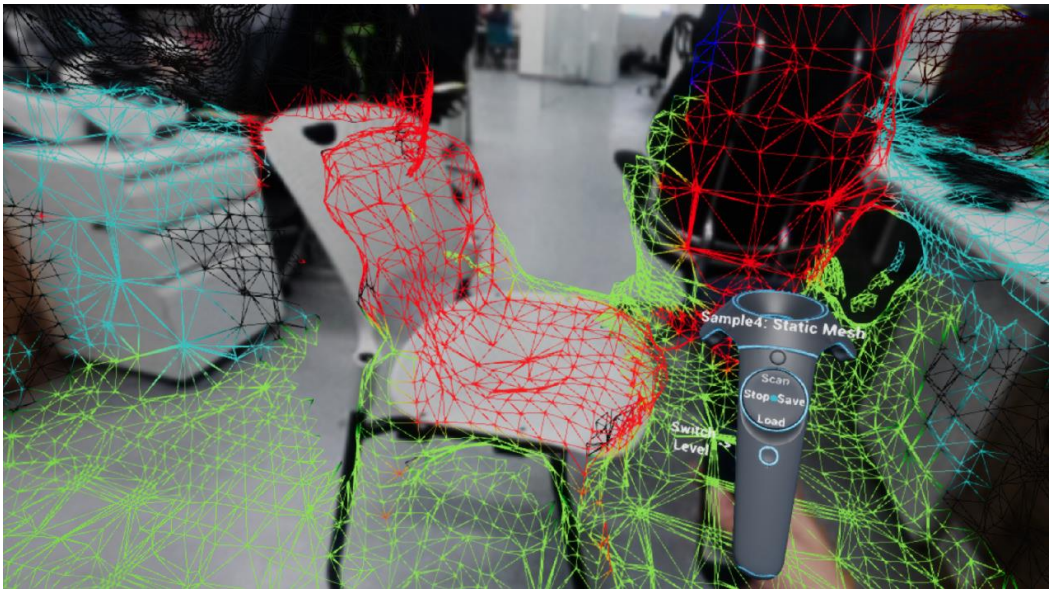

Figure 11. Static Mesh (Sample4)

Details

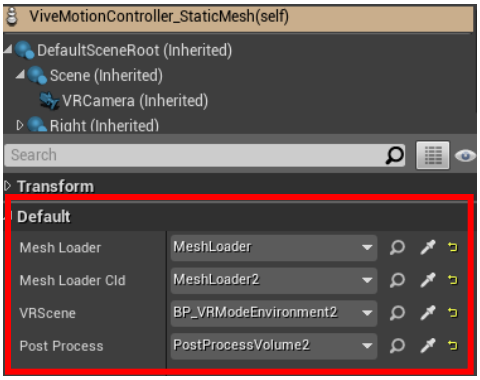

Figure 12. ViveMotionController_StaticMesh (Sample4)

Actor	Setting	Description
ViveMotionController_ StaticMesh	Mesh Loader	Load static mesh Blueprint.
	Mesh Loader Cld	Load static mesh collider Blueprint.
	VRScene	VR Mode Environment.
	Post Process	Post Process for VR Mode.

Table 7. Settings (Sample4)

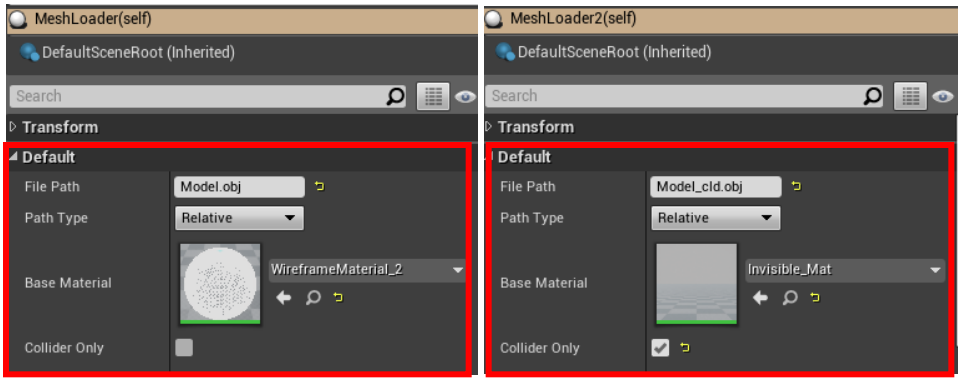

Figure 13. MeshLoader (Sample4)

Actor	Setting	Description
MeshLoader	File Path	Static mesh name (Path).
	Path Type	The path type of the read mesh.
	Base Material	Static mesh's material.
	Collider Only	Is this a mesh collider?

Table 8. Settings (Sample4)

Controller Input

Input	Description
Touchpad Up	Start scanning.
Touchpad Down	Load saved mesh data from the local disc.
Touchpad Left	Stop scanning.
Touchpad Right	Save mesh.
Grip	Switch to next level.

Table 9. Controller Input (Simple4)

Sample5—Chair Segmentation

This sample utilizes segmented chair to demonstrate an application of the segmentation feature (Figure 14).

Figure 14. Chair Segmentation (Sample5)

Details

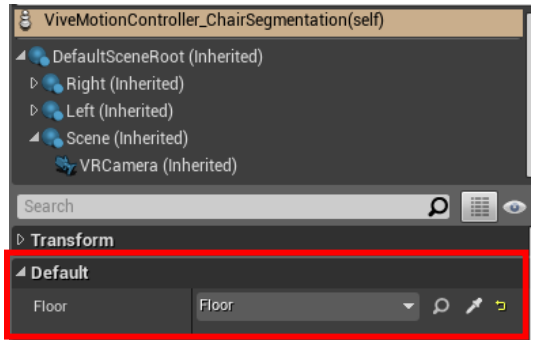

Figure 15. ViveMotionController_Chair Segmentation (Sample5)

Actor	Setting	Description
ViveMotionController_ Chair Segmentation	Floor	For Nav Mesh calculate navigation paths.

Table 10. Settings (Sample5)

Controller Input

Input	Description
Touchpad Up	Start scanning.
Touchpad Left	Stop scanning.
Touchpad Right	Save mesh.
Trigger	Spawn NPC.
Grip	Switch to next level.

Table 11. Controller Input (Simple5)

Sample6—Camera Control

This sample demonstrates the parameters for SRWorks' camera planes (Figure 16).

Figure 16. Camera Control (Sample6)

Details

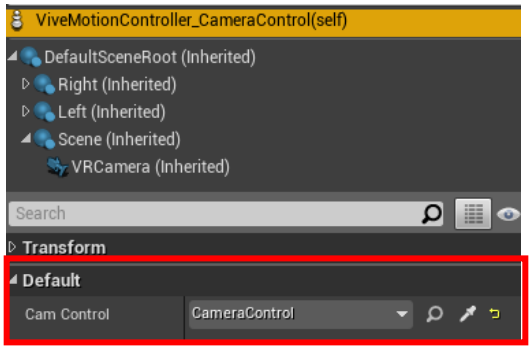

Figure 17. ViveMotionController_CameraControl (Sample6)

Actor	Setting	Description
ViveMotionController_CameraControl	Cam Control	Control panel value.

Table 12. Settings (Sample6)

Controller Input

Input	Description
Touchpad Up	Select upward (Control panel).
Touchpad Down	Select down (Control panel).
Touchpad Left	Add the selected setting value (Control panel).
Touchpad Right	Decrease the selected setting value (Control panel).
Grip	Switch to next level.

Table 11. Controller Input (Simple6)

Sample7—Portal

This sample demonstrates the portal effect (Figure 18).

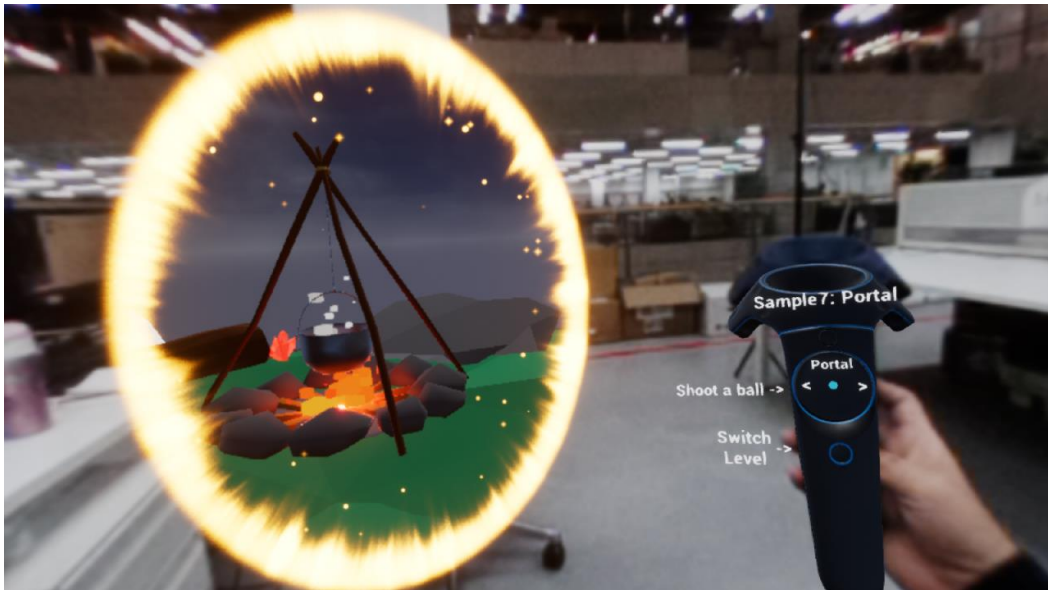

Figure 18. Portal (Sample7)

Details

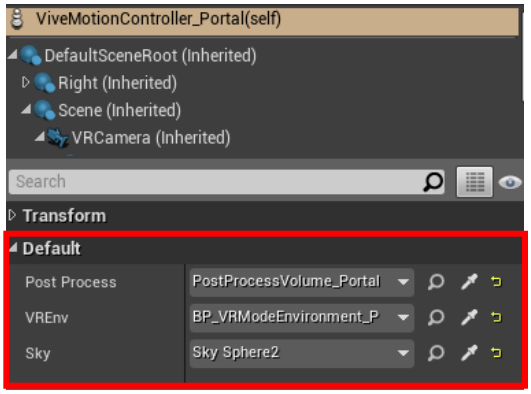

Figure 19. ViveMotionController_Portal (Sample7)

Actor	Setting	Description
ViveMotionController_Portal	Post Process	PostProcess for VR environment.
	VREnv	VR environment Blueprint.
	Sky	VR environment's Sky sphere.

Table 12. Settings (Sample7)

Controller Input

Input	Description
Touchpad Up	Spawn Portal in front of camera.
Touchpad Left	Select to the left (Trigger spawn object type).
Touchpad Right	Select to the right (Trigger spawn object type).
Trigger	Spawn a Ball/Vive Doll.
Grip	Switch to next level.

Table 13. Controller Input (Simple7)