Build Unreal Engine 4.27 / 5.0 OpenXR applications on VIVE Focus3

July. 2022

Revision History

Revision	Date	Description
1.0.0	May 2022	Formal release
1.0.1	July 2022	Support UE5.0 Added Hand Interaction and Wrist Tracker

Contents

1. Introduction

VIVE Wave runtime had already supported OpenXR standard on VIVE Focus3. Welcome to join OpenXR of Wave runtime.

Since Unreal Engine 4.27, Epic Games claimed that the OpenXR Plug-in which allowed to develop the cross-platform applications for OpenXR is Production-Ready. In this guide, you will learn how to build an OpenXR app running on Focus3 through the following step-by-step sections.

What you will learn in this guide:

- Download pre-built Unreal Engine 4.27 / 5.0 through Epic Games Launcher
- Package, install and run Wave OpenXR application on VIVE Focus3
- Porting scene to use Wave OpenXR

What Wave support on Focus3 in this release version:

- HMD tracking pose and XR rendering
- Controller tracking pose, key input, and haptics
- Hand tracking pose and Hand Interaction
- Wrist Tracker

Okay. Let’s get started to create your VIVE Wave OpenXR content on Focus3!

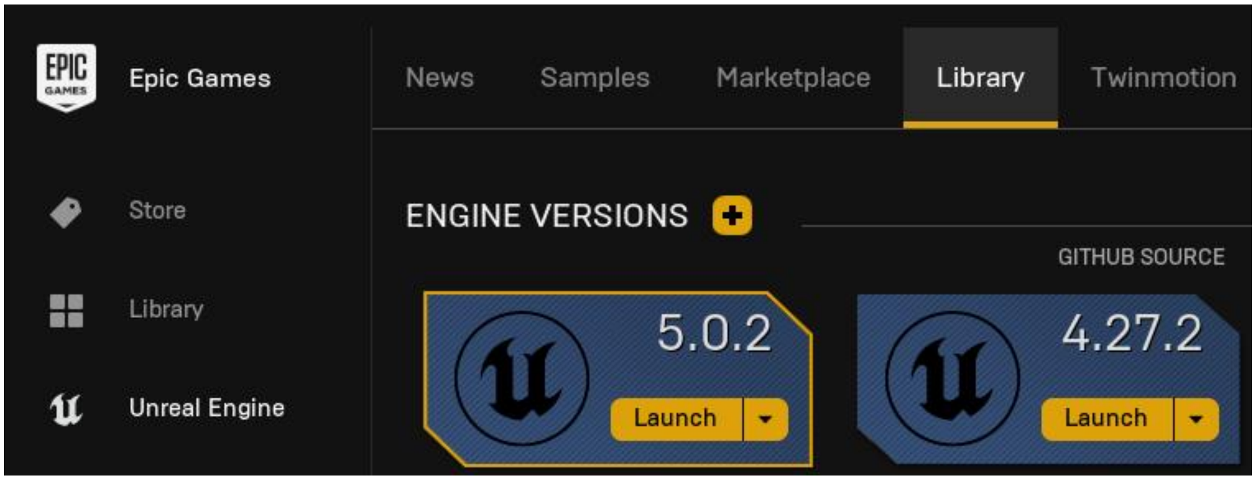
2. Download Pre-built Unreal Engine

2.1 Download Epic Games Launcher.

Download Epic Games Launcher here.

2.2 Download Unreal Engine by Epic Games Launcher

Navigate to ***Unreal Engine > Library*** and add ENGINE VERSIONS **4.27.2** or **5.0.2**.



3 Wave OpenXR SDK

3.1 Wave OpenXR Plug-In

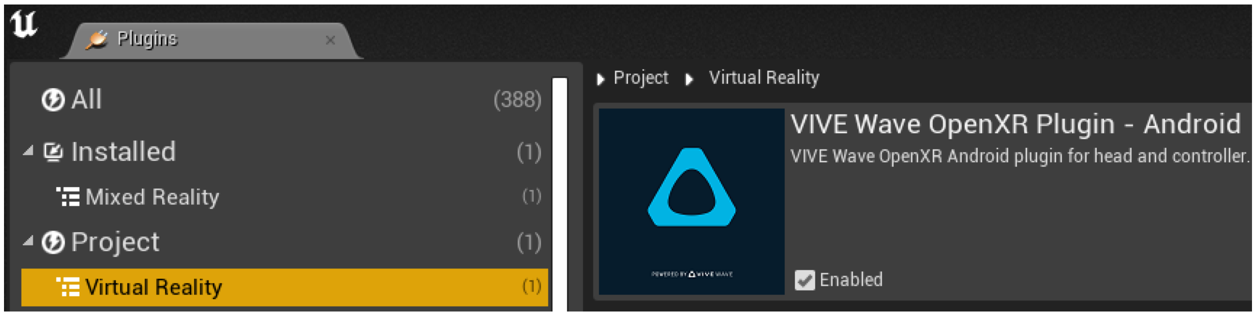
The **Wave OpenXR Plug-In** is at **<Where you unzip wave_openxr_unreal_plugin_<version>.zip>\WaveOpenXR**.

3.2 How to import Wave OpenXR Plug-In

Steps:

1. Copy and paste **WaveOpenXR** to *<Your Project Folder>\Plugins\* such as *<Your Project Folder>\Plugins\***WaveOpenXR** (Create a directory named "Plugins" if the project doesn't have.)
2. Double-click **<Your project>.uproject** to launch the project.

Then click *Unreal Engine Editor > Edit > Plugins*, the **WaveOpenXR** should be in the category *Project > VirtualReality*. Please relaunch the editor if the WaveOpenXR Plug-In did not include in the Plugins list.



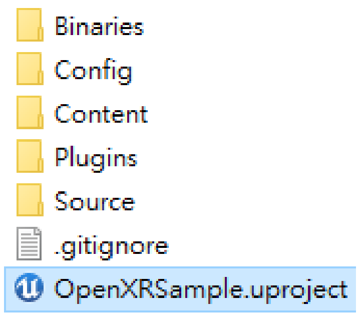
3.3 OpenXRSample

OpenXRSample is a sample project which includes scenes which let you experience our features such as input, hand tracking or other advanced features.

Unzip wave_openxr_unreal_sample_<version>.zip to get it. Please note that, you should import Wave OpenXR Plug-In before packaging the android apps.

4 Package the Android Sample

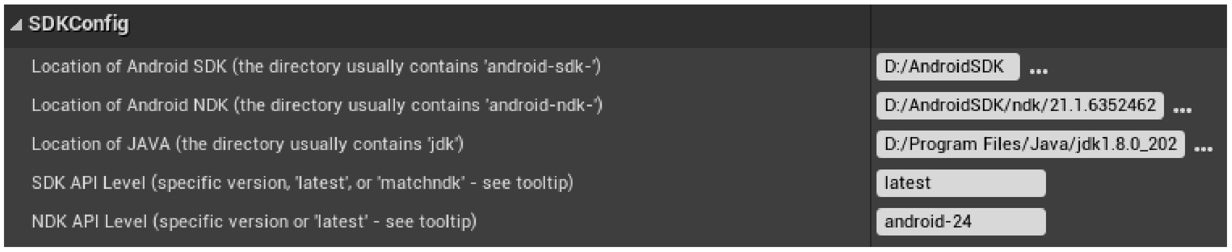
Please make sure you already import the WaveOpen XR Plug-In. Then launch OpenXRSample**OpenXRSample.uproject**.



Binaries
Config
Content
Plugins
Source
.gitignore
OpenXRSample.uproject

4.1 Project Settings of Android Environments

Click **Edit > Project Settings** to open the Project Settings window and navigate to **Platforms > Android SDK**.



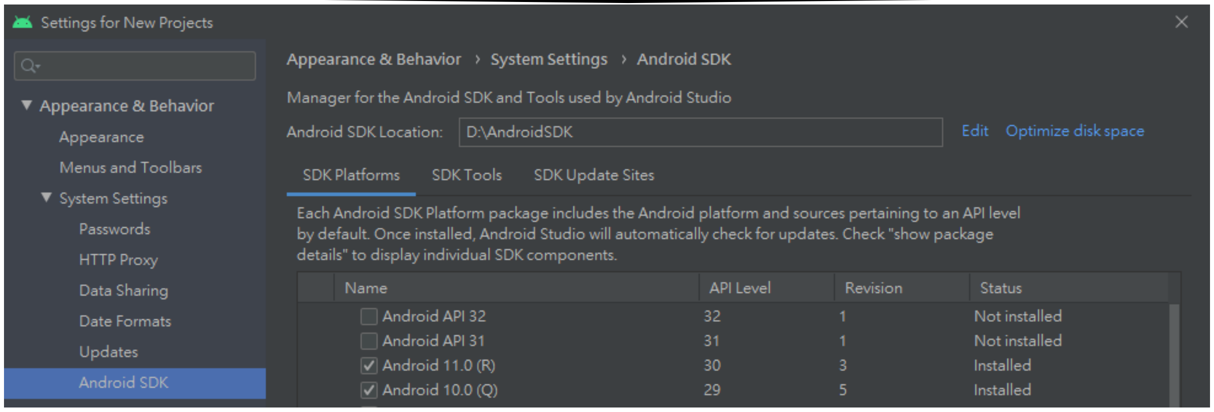
SDKConfig

Location of Android SDK (the directory usually contains 'android-sdk-')	D:/AndroidSDK
Location of Android NDK (the directory usually contains 'android-ndk-')	D:/AndroidSDK/ndk/21.1.6352462
Location of JAVA (the directory usually contains 'jdk')	D:/Program Files/Java/jdk1.8.0_202
SDK API Level (specific version, 'latest', or 'matchndk' - see tooltip)	latest
NDK API Level (specific version or 'latest' - see tooltip)	android-24

You could refer [Setting up Android SDK and NDK for Unreal]() to set up.

4.1.1 Location of Android SDK

Download Android Studio if you didn't have yet. In Android Studio Editor, click *Tools > SDK Manager* and navigate to *Appearance & Behavior > System Settings > Android SDK*. The **Android SDK Location** is the path to be filled.



Settings for New Projects

Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: D:\AndroidSDK

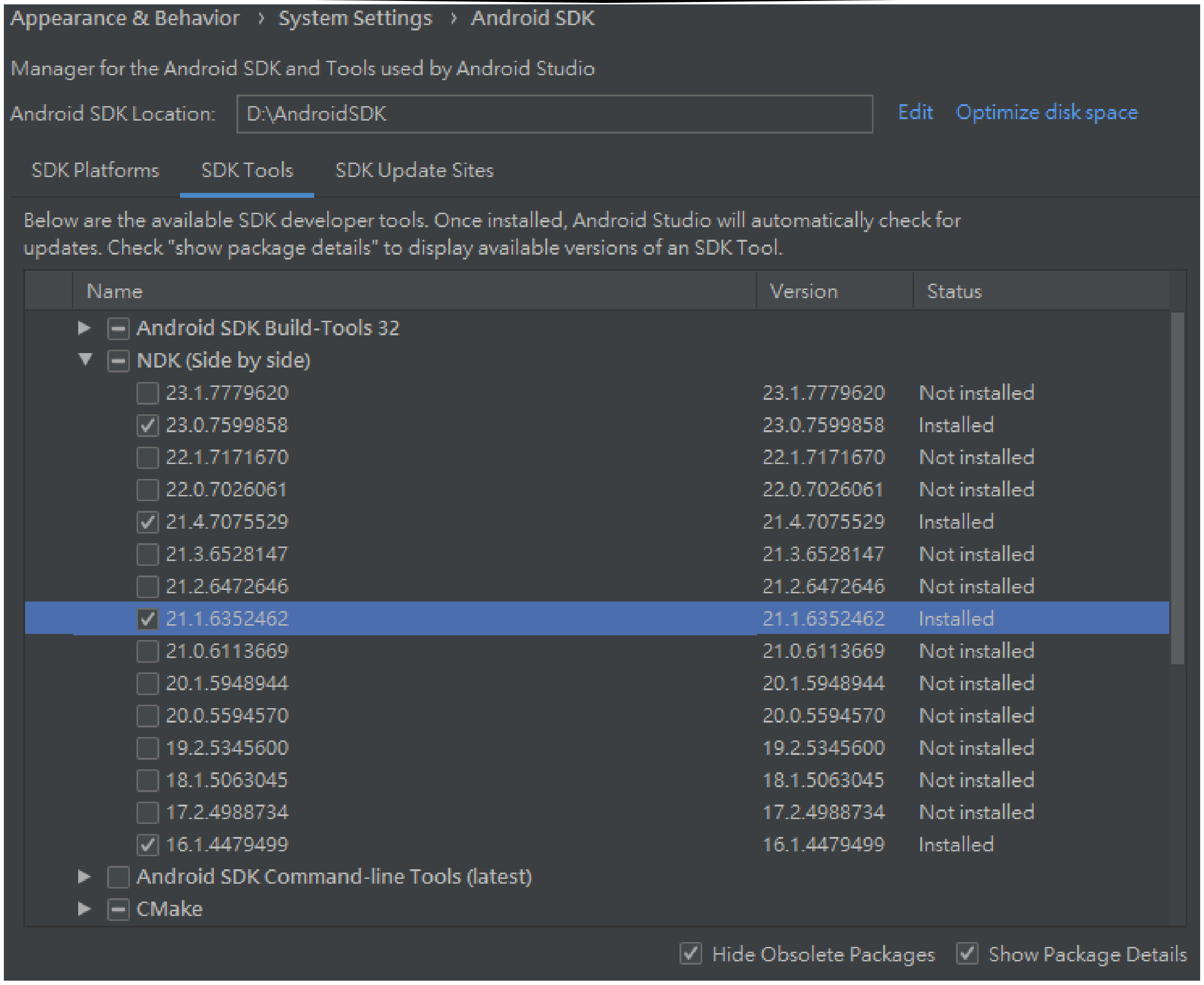
SDK Platforms | SDK Tools | SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

Name	API Level	Revision	Status
☐ Android API 32	32	1	Not installed
☐ Android API 31	31	1	Not installed
✓ Android 11.0 (R)	30	3	Installed
✓ Android 10.0 (Q)	29	5	Installed

4.1.2 Location of Android NDK

Please fill your Android NDK(**21.1.6352462**) path to it. The NDK could be downloaded in the Android SDK. Click the tab **SDK Tools** and select **Show Package Details** at bottom right of the page to see the options.



Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: D:\AndroidSDK

Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

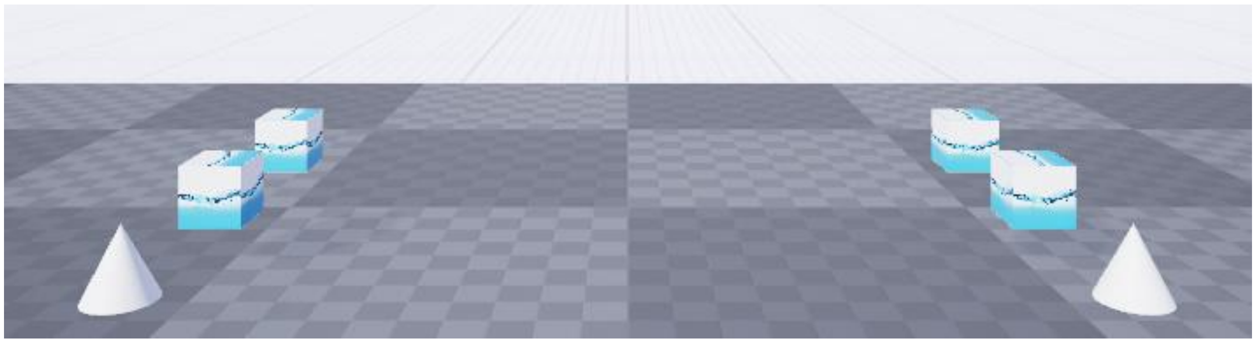
Name	Version	Status
Android SDK Build-Tools 32		
NDK (Side by side)		
☐ 23.1.7779620	23.1.7779620	Not installed
✓ 23.0.7599858	23.0.7599858	Installed
☐ 22.1.7171670	22.1.7171670	Not installed
☐ 22.0.7026061	22.0.7026061	Not installed
✓ 21.4.7075529	21.4.7075529	Installed
☐ 21.3.6528147	21.3.6528147	Not installed
☐ 21.2.6472646	21.2.6472646	Not installed
✓ 21.1.6352462	21.1.6352462	Installed
☐ 21.0.6113669	21.0.6113669	Not installed
☐ 20.1.5948944	20.1.5948944	Not installed
☐ 20.0.5594570	20.0.5594570	Not installed
☐ 19.2.5345600	19.2.5345600	Not installed
☐ 18.1.5063045	18.1.5063045	Not installed
☐ 17.2.4988734	17.2.4988734	Not installed
✓ 16.1.4479499	16.1.4479499	Installed
☐ Android SDK Command-line Tools (latest)		
CMake		

4.1.3 Location of JAVA

It is recommended to install jdk1.8.0.

4.2 Package the ButtonTest to an Android APK

You will see the current level is **ButtonTest** which is located at *Content > ButtonTest > ButtonTest.umap*.



In UE4.27, click **File > Package Project > Android > Android(ASTC)** to package the android application. In UE5.0, click **icon Platforms > Android > Package Project** to package it.

Then you will see the OpenXRSample-arm64.apk in the path *<path to OpenXRSample>/Android_ASTC/*.

Then you can execute **Install_OpenXRSample-arm64.bat** to install the android application automatically. For more information, refer to Android Getting Started.

Launch the sample **OpenXRSample** in Focus3, you can use right controller to control right objects and left controller to control left objects.

The far cube presents the button "touch" and "axis" actions. The far cube will change color when touching a button and rotate when the button has axis. Only **Trigger** and **Grip** button have axis.

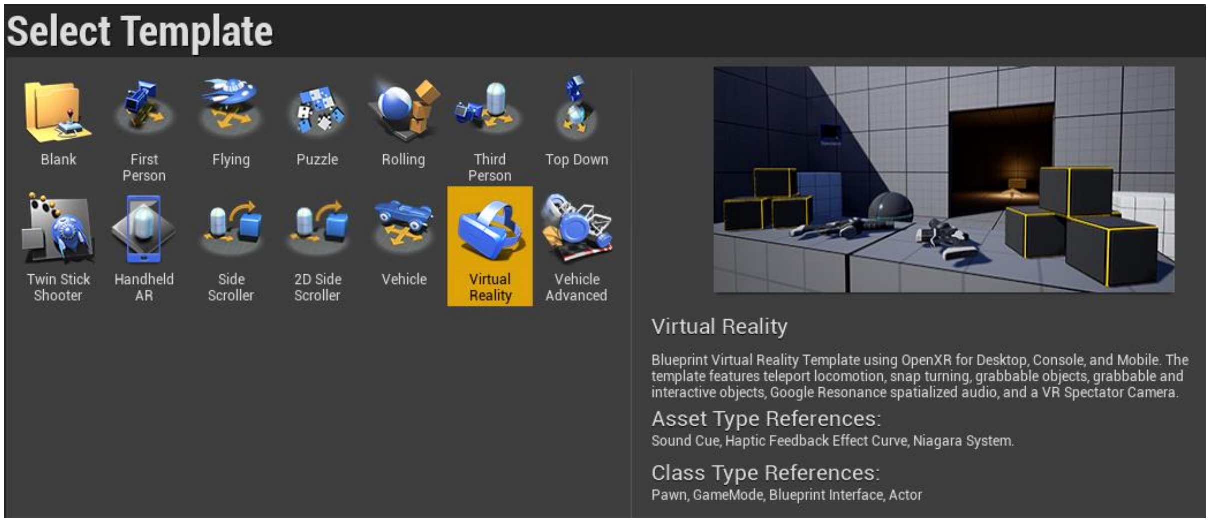
The near cube presents the button "press" action and will change color when pressing a button.

The cone presents the **Thumbstick** axis and will move when you operate the **Thumbstick** button.

5 Porting scene to use Wave OpenXR

Unreal offered a Virtual Reality template project for starter. You can migrate your scene to the project and refer the video below to get started.

In UE4.27, select Virtual Reality project in the template list.



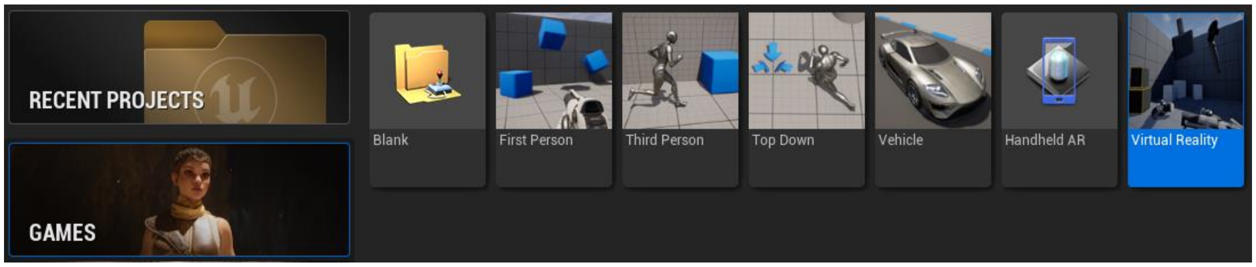
Virtual Reality

Blueprint Virtual Reality Template using OpenXR for Desktop, Console, and Mobile. The template features teleport locomotion, snap turning, grabbable objects, grabbable and interactive objects, Google Resonance spatialized audio, and a VR Spectator Camera.

Asset Type References:
Sound Cue, Haptic Feedback Effect Curve, Niagara System.

Class Type References:
Pawn, GameMode, Blueprint Interface, Actor

In UE5.0, click **GAMES** then select Virtual Reality project.



[Creating AR and VR Projects with Unreal Engine | Webinar]

5.1 Import Wave OpenXR Plug-In

Please refer chapter 3.2 above to see how to import the Wave OpenXR Plug-In.

5.2 Disable other Virtual Reality Plugin-Ins

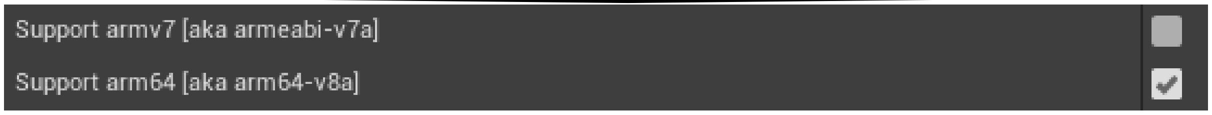
Please keep **Oculus OpenXR**, **Oculus VR** and **SteamVR** disabled and keep **OpenXR**, **OpenXREyeTracker**, **OpenXRHandTracking**, **WaveOpenXR** enabled.

5.3 Project Settings

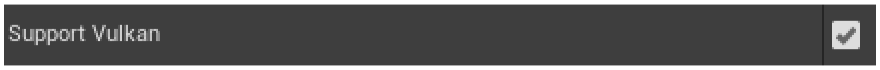
5.3.1 Project Settings of Unreal Engine

We assumed you migrate your scene to Virtual Reality project which Epic Games offered. You can refer other settings in Virtual Reality project that we didn't mention below.

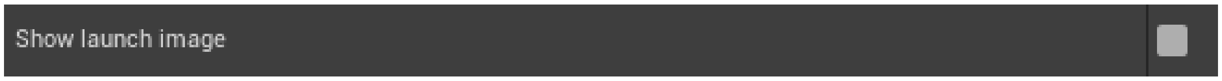
Enable **Support arm64** and disable **Support armv7**. (Skip this in UE5 because UE5 did not support armv7 anymore.)


Support armv7 [aka armeabi-v7a] ☐
Support arm64 [aka arm64-v8a] ✓

Enable **Support Vulkan**. **(MUST)**

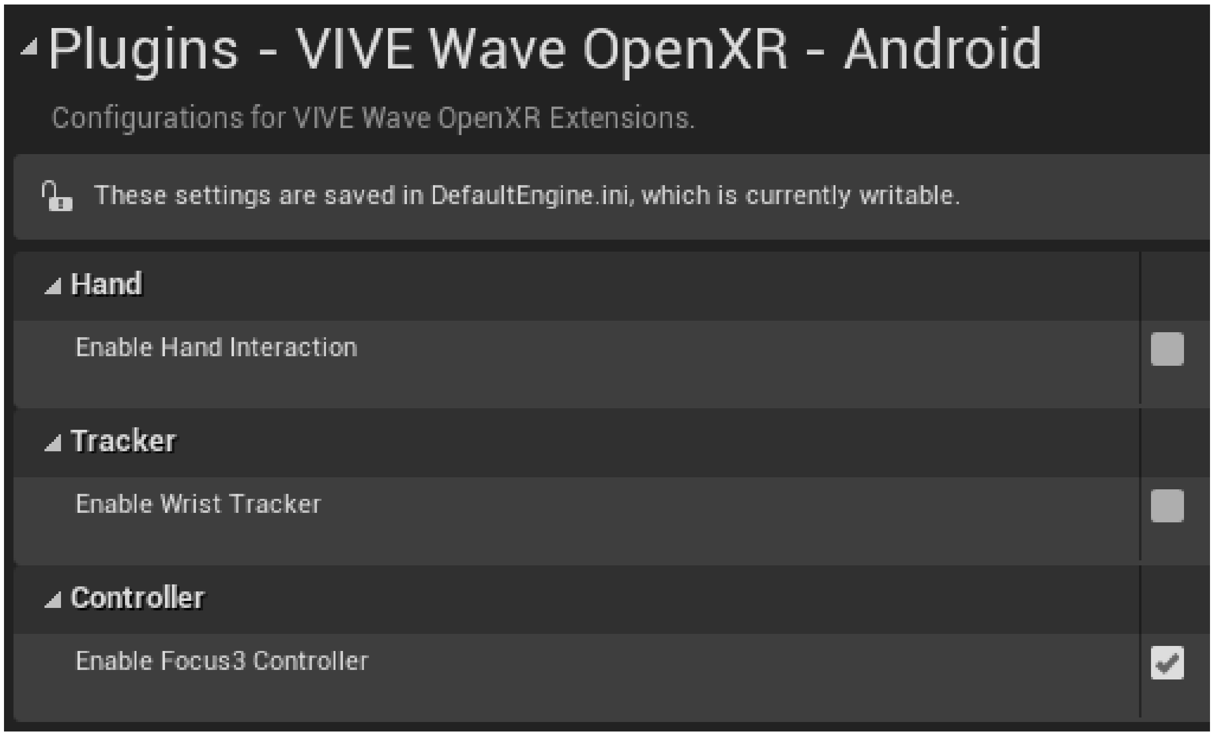

Support Vulkan ✓

Disable **Show launch image** to get rid of the non-stereo splash while launching app.


Show launch image ☐

5.3.2 Project Settings of WaveOpenXR Plug-In

Click **Edit > Project Settings** to open the Project Settings window and navigate to **Plugins > VIVE Wave OpenXR - Android**.


Plugins - VIVE Wave OpenXR - Android

Configurations for VIVE Wave OpenXR Extensions.

These settings are saved in DefaultEngine.ini, which is currently writable.

Hand

Enable Hand Interaction ☐

Tracker

Enable Wrist Tracker ☐

Controller

Enable Focus3 Controller ✓

5.3.2.1 Enable Hand Interaction

Select this option to enable Hand Interaction feature. For more information, see chapter 6.2.

5.3.2.2 Enable Wrist Tracker

Select this option to enable Wrist Tracker feature. For more information, see chapter 6.3.

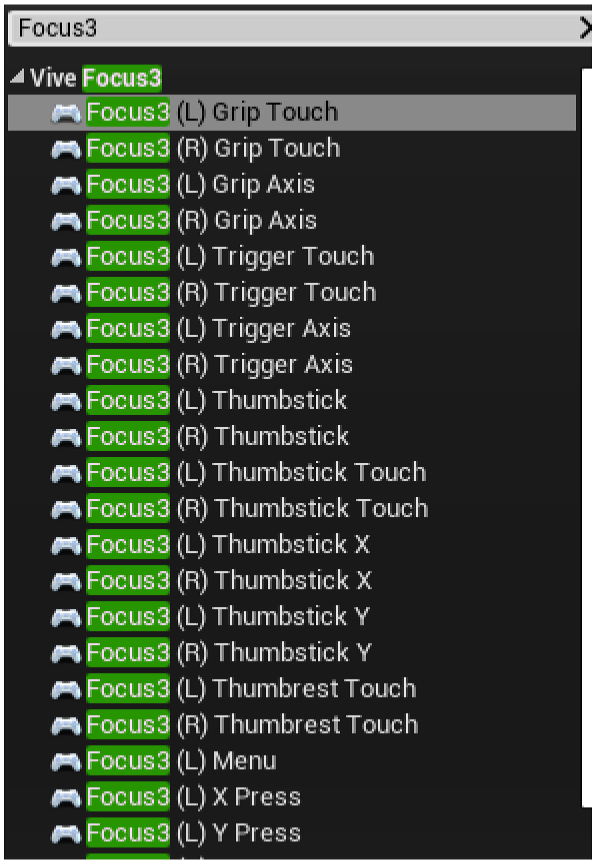
5.3.2.3 Enable Focus3 Controller

Select this option to support Focus3 controller.

5.4 Input

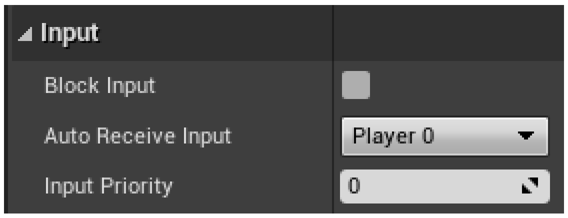
To use the Focus3 controller input, you have to apply the **Input** action and axis mappings. Refer to this guide for detailed information.

Go to **Project Settings > Engine > Input** to configure the **Action Mappings** and **Axis Mappings**. You can see the inputs of **VIVE Focus3** by typing "Focus3" in the action specifying window.



```
Focus3
◢ Vive Focus3
   Focus3 (L) Grip Touch
   Focus3 (R) Grip Touch
   Focus3 (L) Grip Axis
   Focus3 (R) Grip Axis
   Focus3 (L) Trigger Touch
   Focus3 (R) Trigger Touch
   Focus3 (L) Trigger Axis
   Focus3 (R) Trigger Axis
   Focus3 (L) Thumbstick
   Focus3 (R) Thumbstick
   Focus3 (L) Thumbstick Touch
   Focus3 (R) Thumbstick Touch
   Focus3 (L) Thumbstick X
   Focus3 (R) Thumbstick X
   Focus3 (L) Thumbstick Y
   Focus3 (R) Thumbstick Y
   Focus3 (L) Thumbrest Touch
   Focus3 (R) Thumbrest Touch
   Focus3 (L) Menu
   Focus3 (L) X Press
   Focus3 (L) Y Press
```

Note: you have to enable the **Auto Receive Input** option in your blueprints to retrieve inputs.

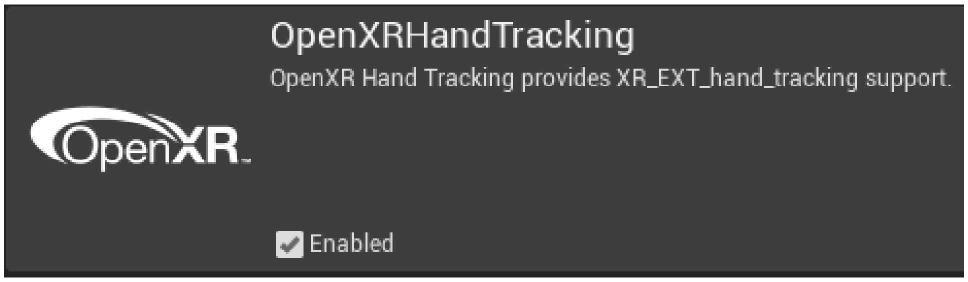


Input

Block Input	
Auto Receive Input	Player 0
Input Priority	0

6 Advanced Features

6.1 Hand Tracking


OpenXRHandTracking

OpenXR Hand Tracking provides XR_EXT_hand_tracking support.

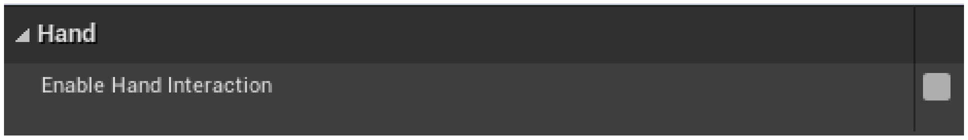
Enabled

Please make sure the OpenXRHandTracking is enabled and refer to OpenXRSample/Content/NaturalHand/NaturalHand.umap

6.2 Hand Interaction

Use pinch motion to interact with Actors.

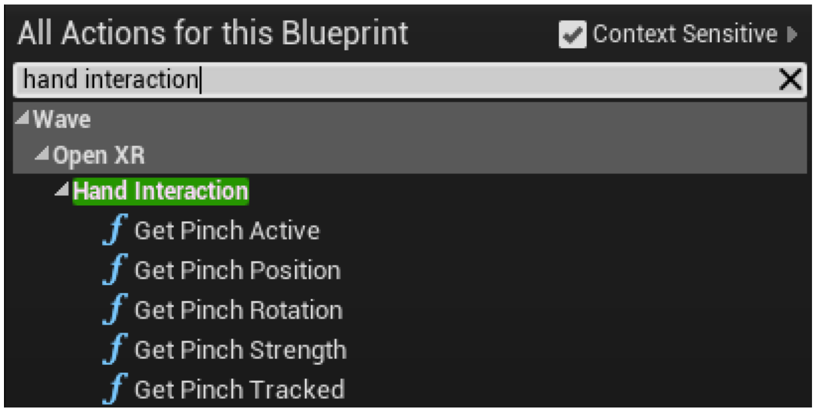
6.2.1Project Setting


Hand

Enable Hand Interaction

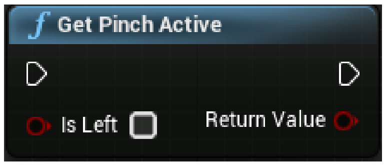
Click **Edit > Project Settings** to open the Project Settings window and navigate to **Plugins > VIVE Wave OpenXR - Android**. Select **Enable Hand Interaction** to enable this feature.

6.2.2Blueprint Function Libraries



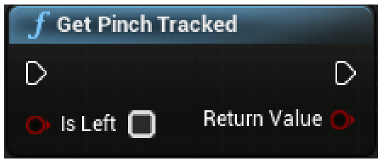
You can retrieve these pinch status of left hand if **Is Left** is true and the status of right hand if **Is Left** is false.

6.2.2.1 Get Pinch Active



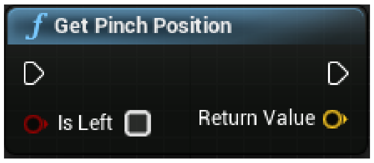
Checks if the pinch motion is active.

6.2.2.2 Get Pinch Tracked



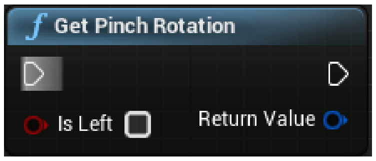
Checks if the pinch motion is tracked.

6.2.2.3 Get Pinch Position



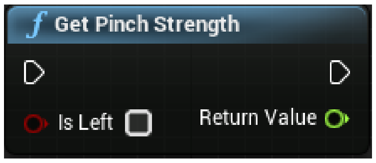
Retrieves the position of pinch origin of left or right hand.

6.2.2.4 Get Pinch Rotation



Retrieves the rotation of pinch origin of left or right hand.

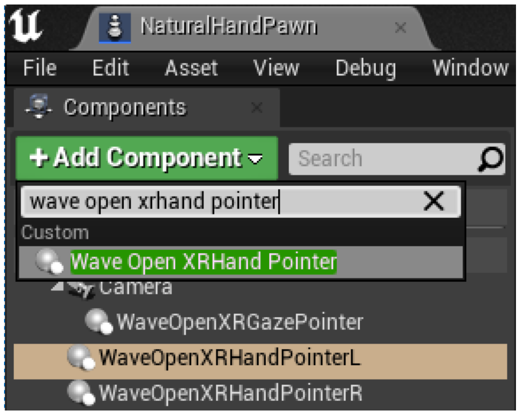
6.2.2.5 Get Pinch Strength



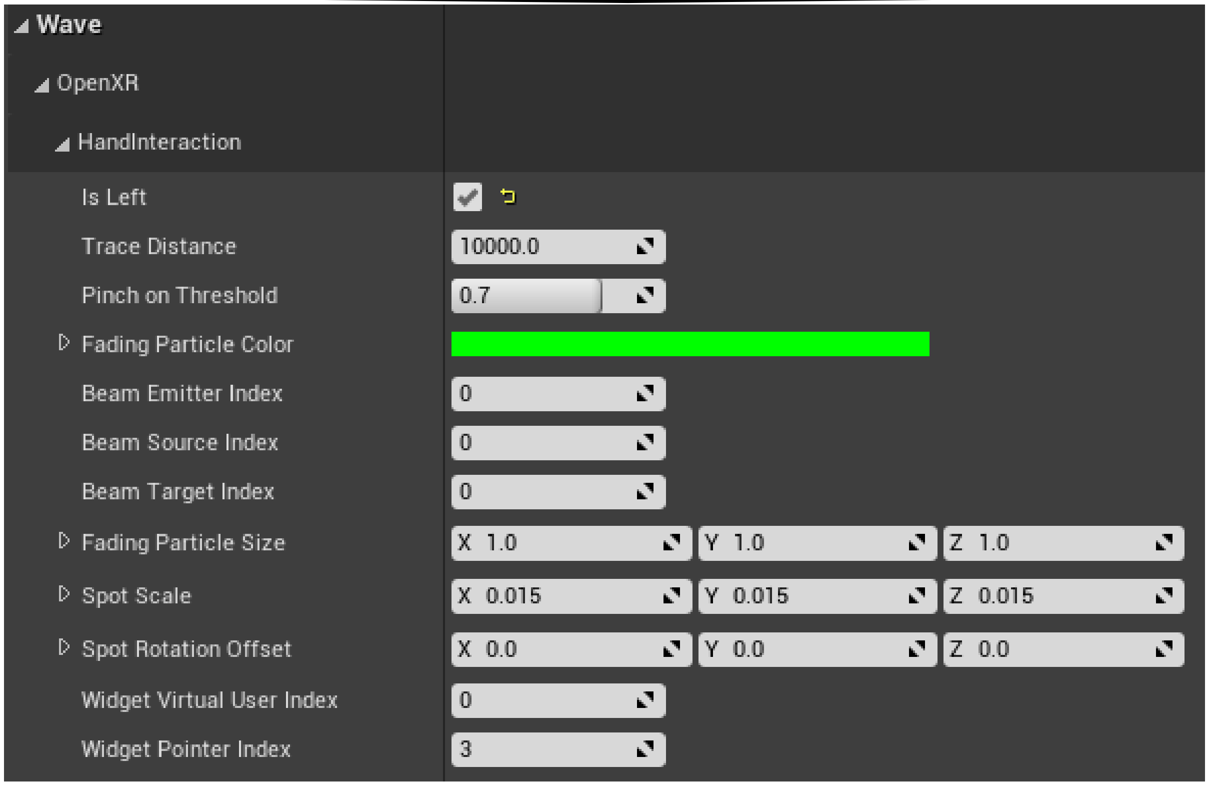
Retrieves the pinch strength of left or right hand. The return value should be [0, 1]. 1 means the index finger is closed to thumb very much.

6.2.3Wave OpenXR Hand Pointer

Wave OpenXR Hand Pointer is a SceneComponent which use the Hand Interaction feature. Wave OpenXR Hand Pointer make you easy to interact with Actors by hand.



The detail of WaveOpenXRHandPointer:



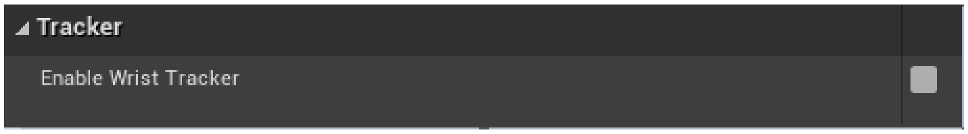
Wave	
OpenXR	
HandInteraction	
Is Left	✓
Trace Distance	10000.0
Pinch on Threshold	0.7
Fading Particle Color	🟩
Beam Emitter Index	0
Beam Source Index	0
Beam Target Index	0
Fading Particle Size	X 1.0 Y 1.0 Z 1.0
Spot Scale	X 0.015 Y 0.015 Z 0.015
Spot Rotation Offset	X 0.0 Y 0.0 Z 0.0
Widget Virtual User Index	0
Widget Pointer Index	3

Please refer OpenXRSample/Content/NaturalHand/NaturalHand.umap to see the sample.

6.3 Wrist Tracker

Retrieve the status of Wrist Tracker.

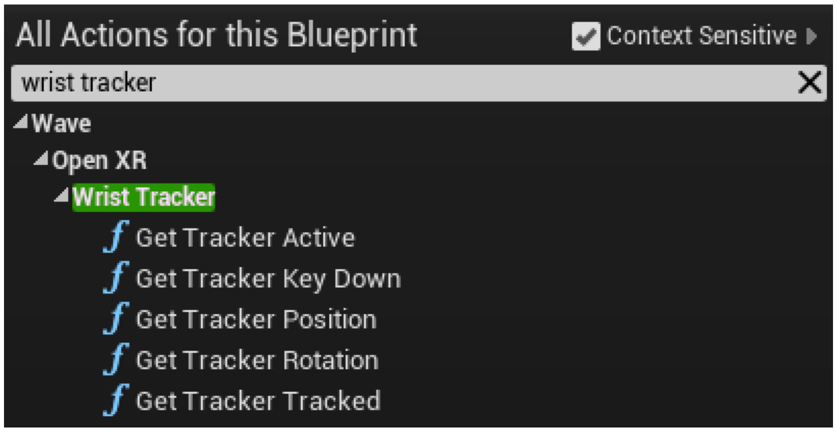
6.3.1Project Settings


Tracker

Enable Wrist Tracker

Click **Edit > Project Settings** to open the Project Settings window and navigate to **Plugins > VIVE Wave OpenXR - Android**. Select **Enable Wrist Tracker** to enable this feature.

6.3.2Blueprint Function Libraries

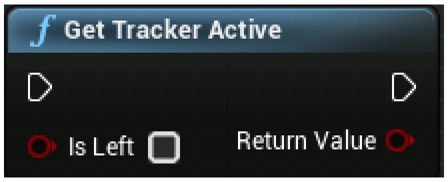

All Actions for this Blueprint ✓ Context Sensitive

wrist tracker

- Wave
  - Open XR
    - Wrist Tracker
      - Get Tracker Active
      - Get Tracker Key Down
      - Get Tracker Position
      - Get Tracker Rotation
      - Get Tracker Tracked

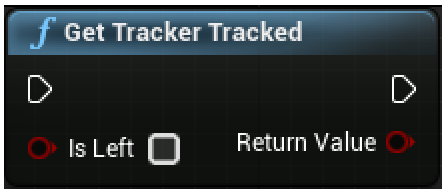
You can retrieve these tracker status of left or right device by setting **Is Left**. The **L** / **R** light mark the device is left or right. The L light will be on if the tracker is the left device.

6.3.2.1 Get Tracker Active



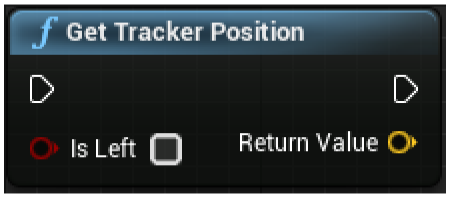
Checks if the tracker is active. (The device is connected or not.)

6.3.2.2 Get Tracker Tracked



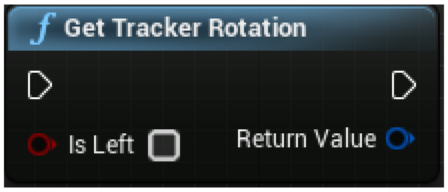
Checks if the tracker is tracked.

6.3.2.3 Get Tracker Position



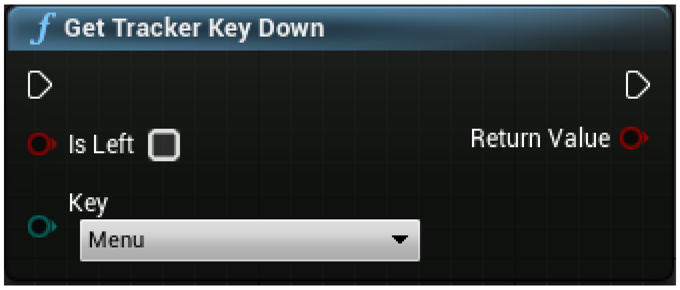
Retrieves the position of the left or right tracker.

6.3.2.4 Get Tracker Rotation



Retrieves the rotation of the left or right tracker.

6.3.2.5 Get Tracker Key Down



There are two buttons for each tracker. They are **Menu** and **Primary**.

Checks if these buttons pressing or not (releasing).

Note: You cannot retrieve the key status of Menu on the right device since it's a reserved key.

6.3.3Key event of Tracker input

The Get Tracker Key Down Blueprint Function is a polling way to get key status. If you want to listen one key event when the button had been pressed or released. You can register your input action event and add Tracker inputs in it:

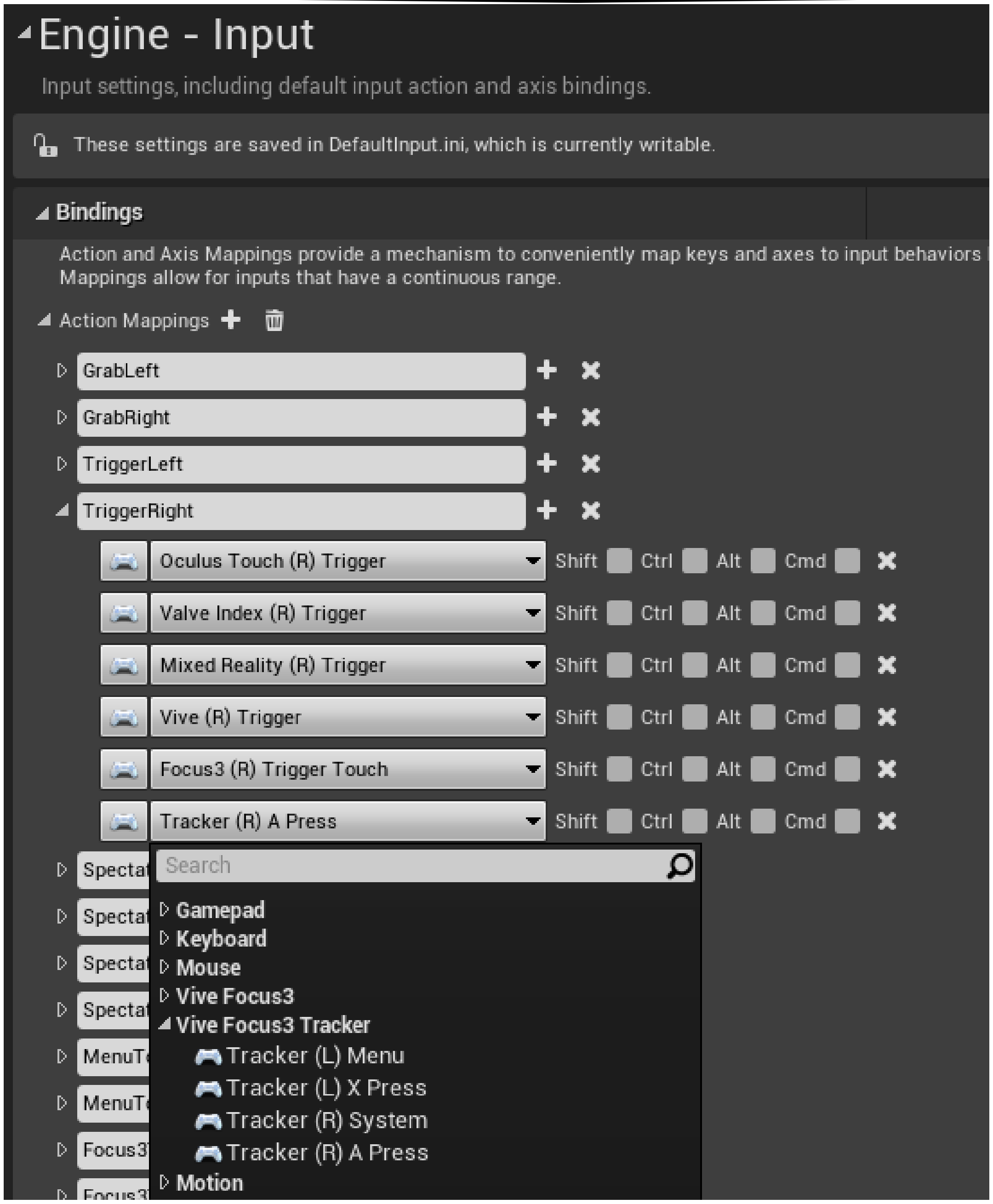
Click **Edit > Project Settings** to open the Project Settings window and navigate to **Engine > Input**.

There are four inputs belongs to tracker, two for each tracker:

Tracker (L) Menu, Tracker (L) X Press, Tracker (R) System and Tracker (R) A Press.

Note: You cannot retrieve the Tracker (R) System key status since it's a reserved key.

For example, I register the Input Action TriggerRight and add the **Tracker (R) A Press** in it, then I can listen the key event by InputAction TriggerRight.



Engine - Input

Input settings, including default input action and axis bindings.

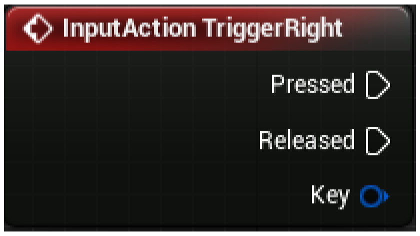
These settings are saved in DefaultInput.ini, which is currently writable.

Bindings

Action and Axis Mappings provide a mechanism to conveniently map keys and axes to input behaviors
Mappings allow for inputs that have a continuous range.

Action Mappings

- GrabLeft
- GrabRight
- TriggerLeft
- TriggerRight
  - Oculus Touch (R) Trigger — Shift Ctrl Alt Cmd
  - Valve Index (R) Trigger — Shift Ctrl Alt Cmd
  - Mixed Reality (R) Trigger — Shift Ctrl Alt Cmd
  - Vive (R) Trigger — Shift Ctrl Alt Cmd
  - Focus3 (R) Trigger Touch — Shift Ctrl Alt Cmd
  - Tracker (R) A Press — Shift Ctrl Alt Cmd
- Specta
- Specta
- Specta
- Specta
- MenuT
- MenuT
- Focus3
- Focus3



For more information, refer to InteractiveExperiences-Input.

7 Known Issues

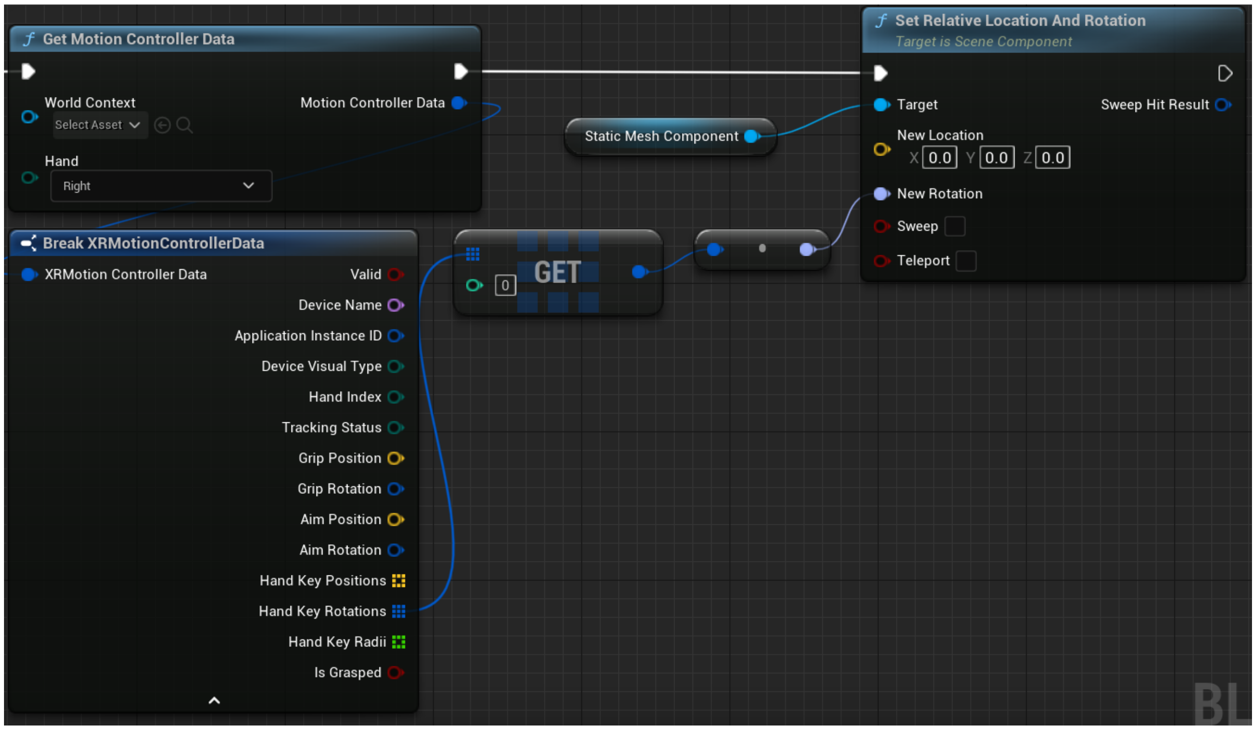
None.

8 Troubleshooting

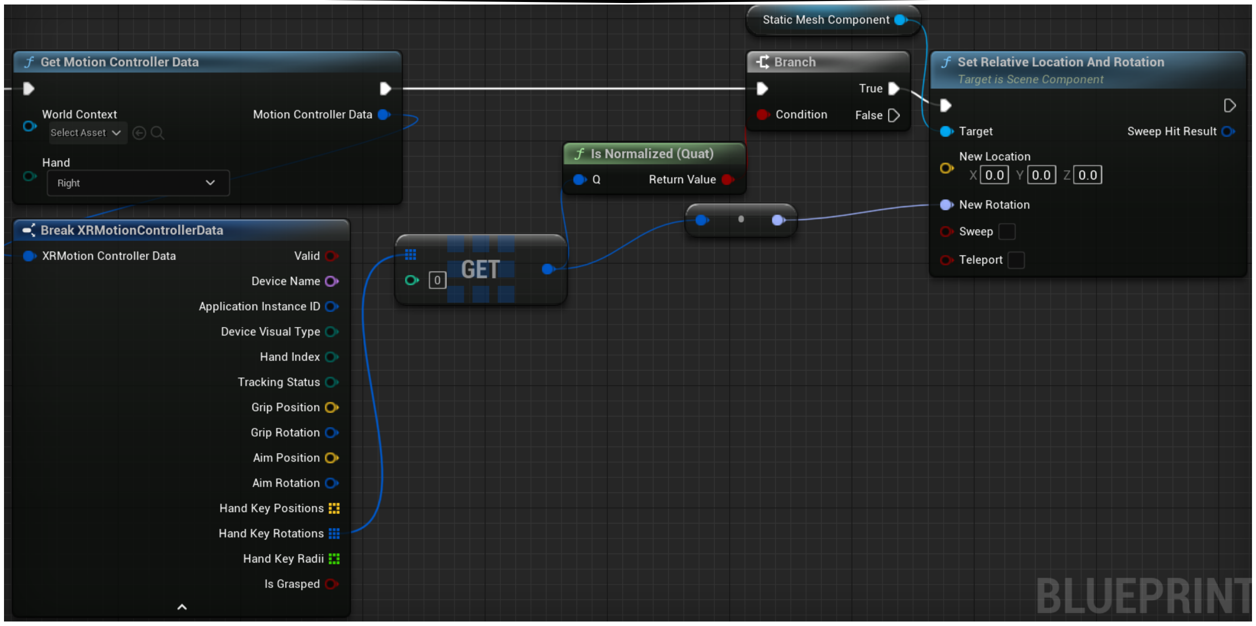
8.1 Apps with development build may crash after turning off HMD by pressing power bank or take off for a while and put it on.

[How to fix] Select *project settings > Project > Packaging > Project > Build Configuration* as Shipping.

8.2 [UE5][HandTracking][ShippingBuild] The Actor may disappear when you convert Hand Key Rotation (Quat) to Rotator and set the rotation to the Actor.



[How to fix] Make sure the Quat **IsNormalized(Quat)** first.


Static Mesh Component
Get Motion Controller Data
World Context
Select Asset
Motion Controller Data
Hand
Right
Branch
Condition
True
False
Set Relative Location And Rotation
Target is Scene Component
Target
Sweep Hit Result
New Location
X 0.0 Y 0.0 Z 0.0
New Rotation
Sweep
Teleport
Is Normalized (Quat)
Q
Return Value
Break XRMotionControllerData
XRMotion Controller Data
Valid
Device Name
Application Instance ID
Device Visual Type
Hand Index
Tracking Status
Grip Position
Grip Rotation
Aim Position
Aim Rotation
Hand Key Positions
Hand Key Rotations
Hand Key Radii
Is Grasped
GET
0
BLUEPRINT