htc

VIVE WAVE

Build Unreal Engine 4.27 OpenXR applications on VIVE Focus3

Mar. 2022

Revision History

Revision	Date	Description
1.0	January 2022	Initial release
1.1	March 2022	Support Vulkan with pre-built Unreal Engine

Contents

1. Introduction

VIVE Wave runtime is going to support OpenXR standard on VIVE Focus3. If you are reading this development guide now, it means that you are the close partners with Wave, and welcome to join OpenXR Beta trial of Wave runtime.

In 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 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

Okay. Let's get started to create your 1st VIVE Wave OpenXR content on Focus3!

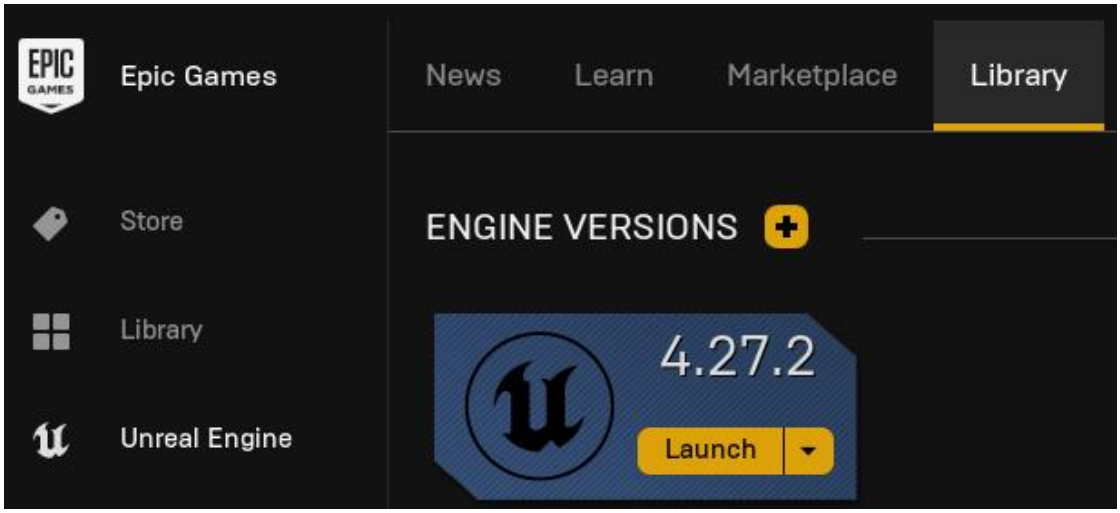
2. Download Pre-built Unreal Engine 4.27

2.1 Download Epic Games Launcher.

Download Epic Games Launcher here.

2.2 Download Unreal Engine 4.27 by Epic Games Launcher

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



3 Wave OpenXR SDK

3.1 OpenXRSample

OpenXRSample is a project which included Wave OpenXR Plug-In. Unzip OpenXRSample.zip to get it.

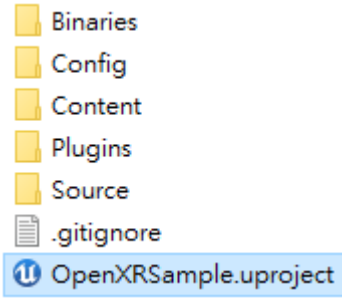
3.2 Wave OpenXR SDK

The **Wave OpenXR SDK** is at **<Where you unzip OpenXRSample.zip>\OpenXRSample\Plugins\WaveOpenXR**.

Note
You can refer chapter 5.1 below to see how to import the Wave OpenXR SDK.

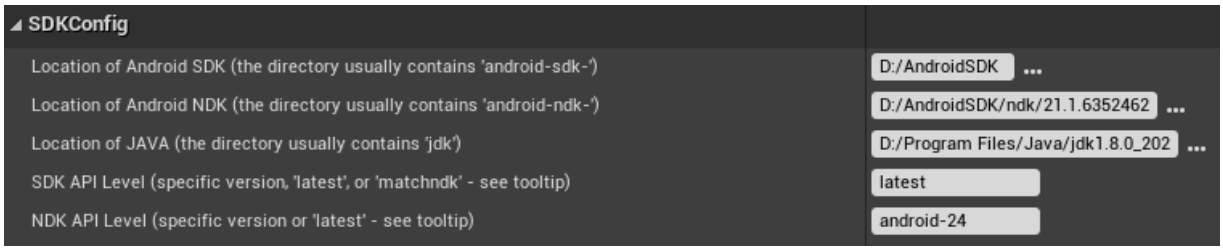
4 Package the Android Sample

Launch OpenXRSample**OpenXRSample.uproject**.



4.1 Project settings of Android environments

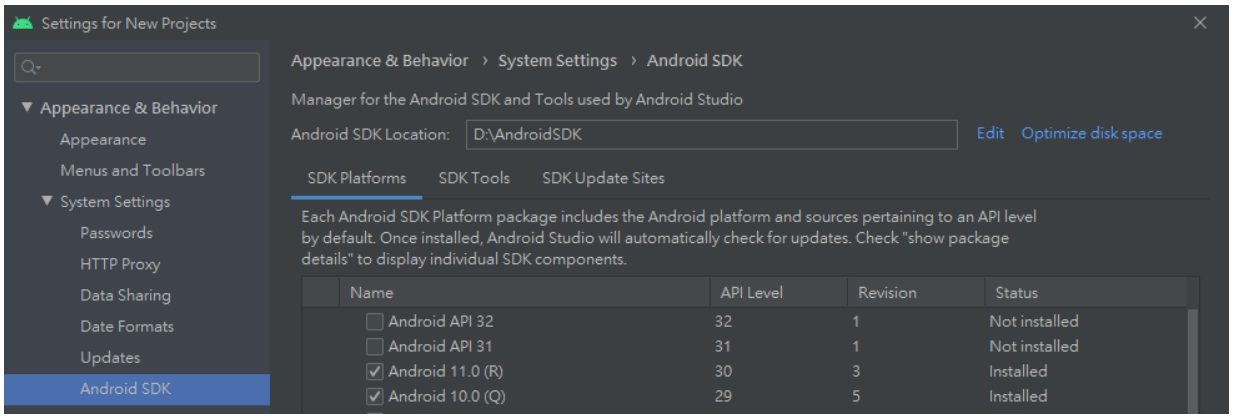
Click **Edit > Project Settings** to open the Project Settings window and navigate to **Platforms > Android SDK**.



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

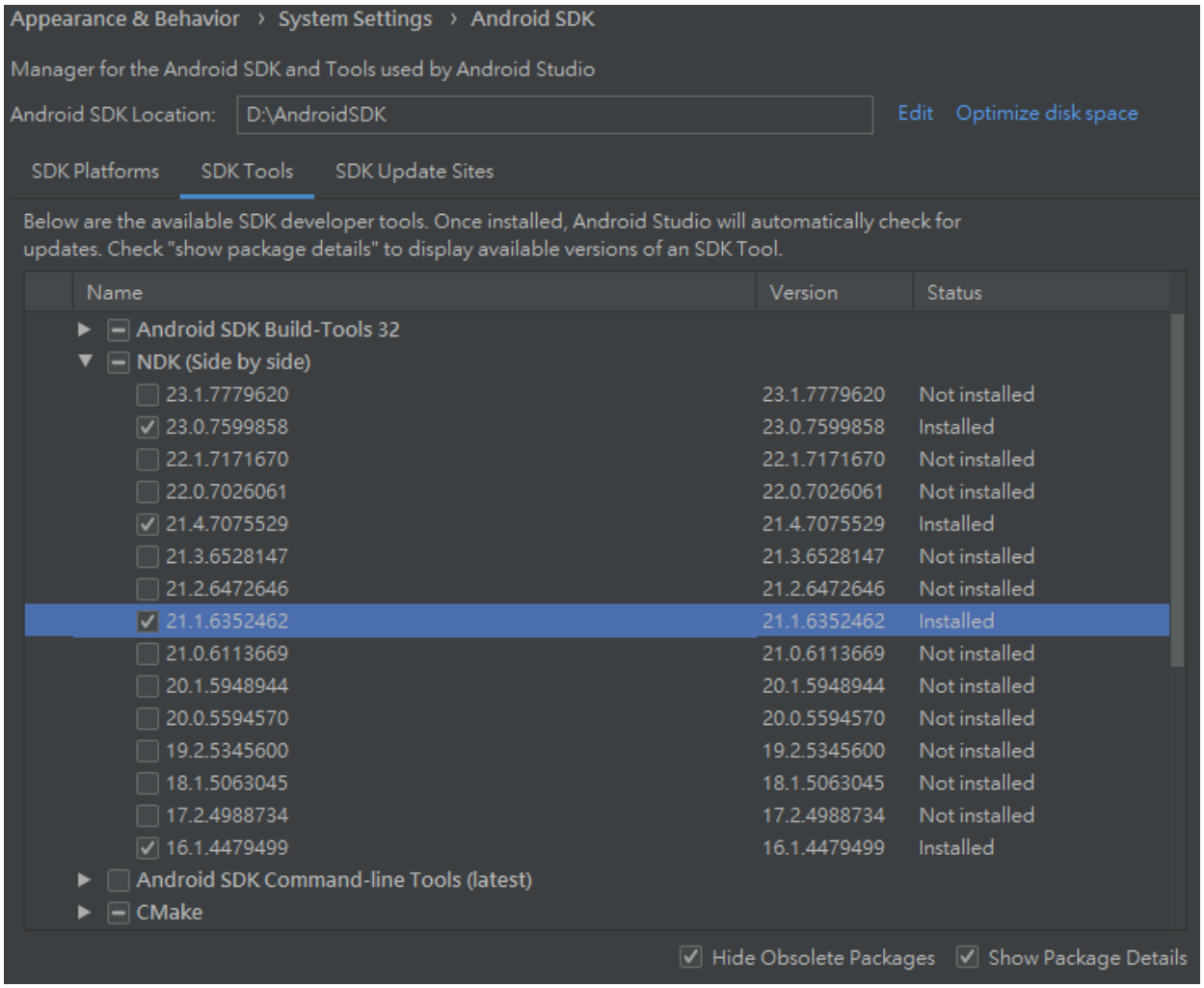
4.1.1 Location of Android SDK

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.



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.

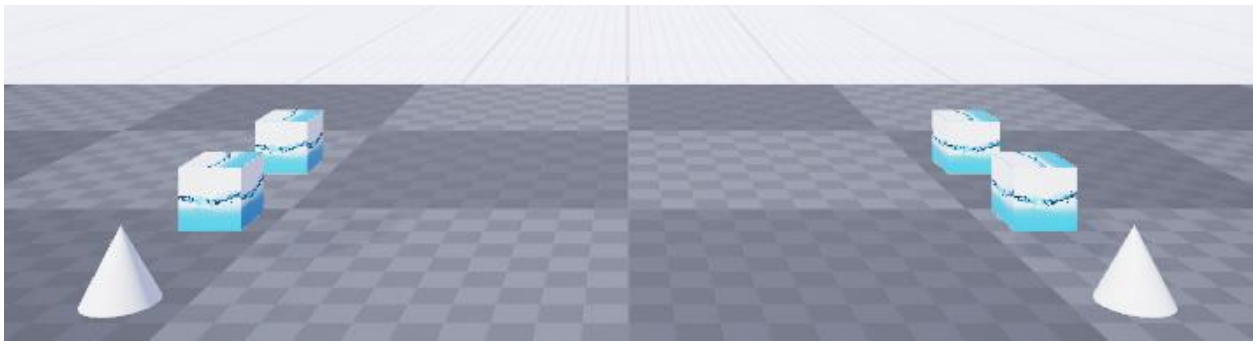


4.1.3 Location of JAVA

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*.



Click **File > Package Project > Android > Android(ASTC)** to package the android application 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.

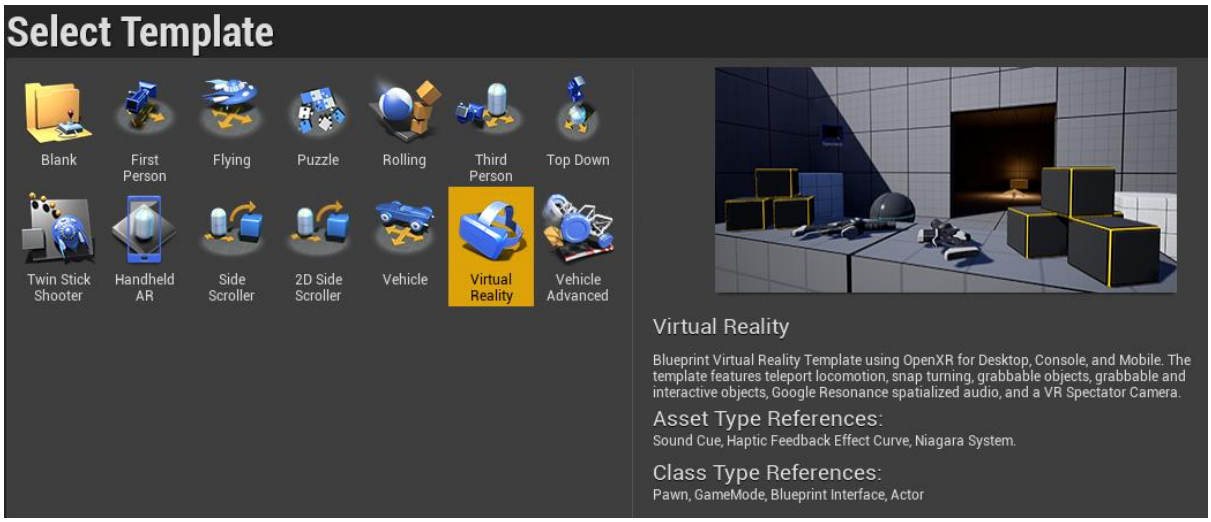
Launch the sample **OpenXRSample** in Focus3, you can use right controller to control right objects and left controller to control left objects.

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. You can migrate your scene to the project and refer the video to get started.

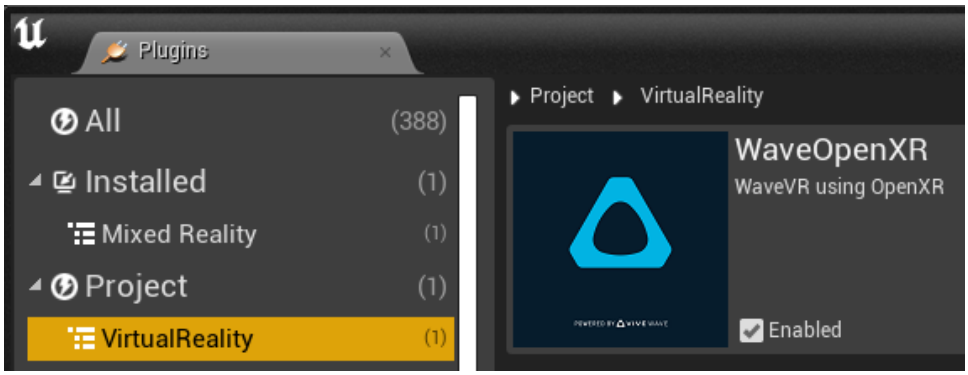
[Creating AR and VR Projects with Unreal Engine | Webinar]

5.1 Import Wave OpenXR SDK

Steps:

1. Copy and paste OpenXRSample*Plugins***WaveOpenVR** 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*.



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

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**.

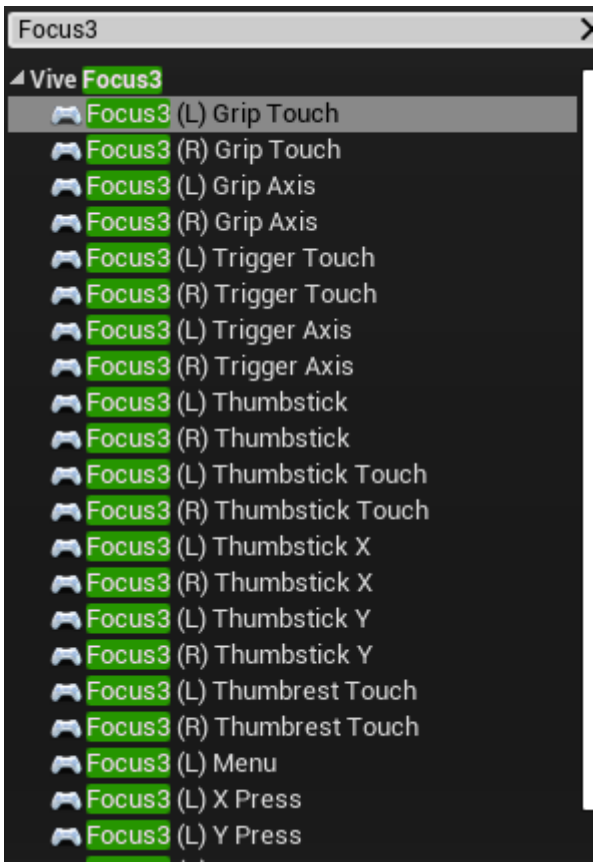
Enable **Support Vulkan**. **(MUST)**

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

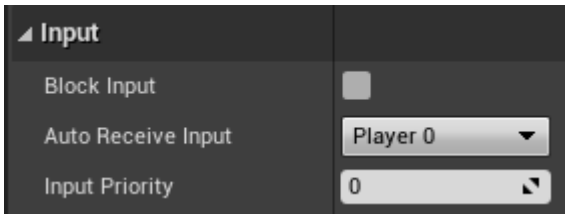
5.4 Input

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.



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



6 Advanced Features

6.1 Hand Tracking

Please refer to
OpenXRSample/Content/NaturalHand/NaturalHand.umap

7 Known Issues

1. Vulkan render consumes more GPU usage than our expectation. Clarifying.

8 Troubleshooting

None