NOLO VR Unity SDK

Documentation

NOLO Co., Ltd

October 2020

北京凌宇智控科技有限公司 NOLO Co.,Ltd
电话：010-53630964
邮箱：hr@nolovr.com
地址：北京市海淀区永泰庄北路 1 号中关村东升国际创业园 1 号楼 2 层优投空间 U101
邮编：100192
www.nolovr.com

目录

1. Overview

1. 1 NOLO VR Unity SDK

NOLO VR Unity SDK is developed by NOLO Co., Ltd. for the NOLO X1, use for Unity developers to develop 3-DoF, 6-DoF Mobile VR games.

2. Preparation for development

Requirement for software: Unity 5.6 or above.

NOLO HOME PC version address:
http://download.nolovr.com/download/NOLO_home_PC.html

Please contact dev@nolovr.com to apply Appkey and fill it in your Unity project.

You can use the public Appkey in the debugging phase, and change it to official Appkey when it is launched officially.

Public Appkey: 4e4f4c4f484f4d457eff82725bc694a5

The name of the project test build: com.nolo.xr.unitydemo. The apk package name should be bound with NibiruSDKKey.bin file. If you want to modify the package name, please refer to the Nolo Unity SDK package

name modification instruction 3.1 document.

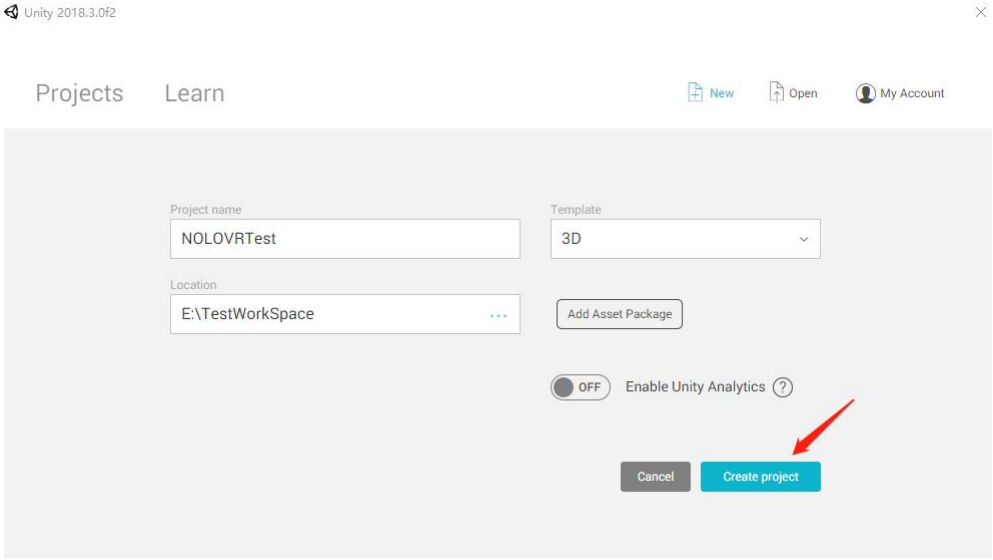
Requirement for Hardware: NOLO X1 standalone VR headset

3. Instructions

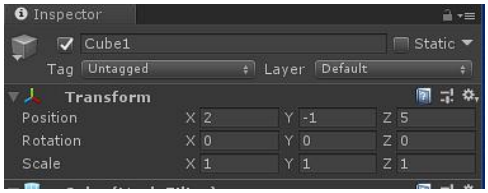
3. 1 Quick Start

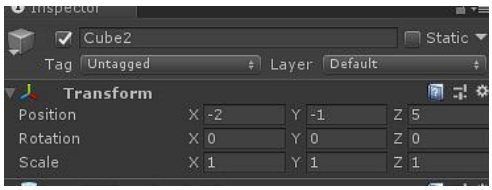
Step 1: Create a project

1) Open Unity, and create a new Unity project.

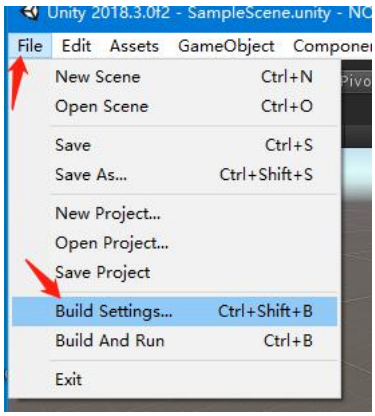


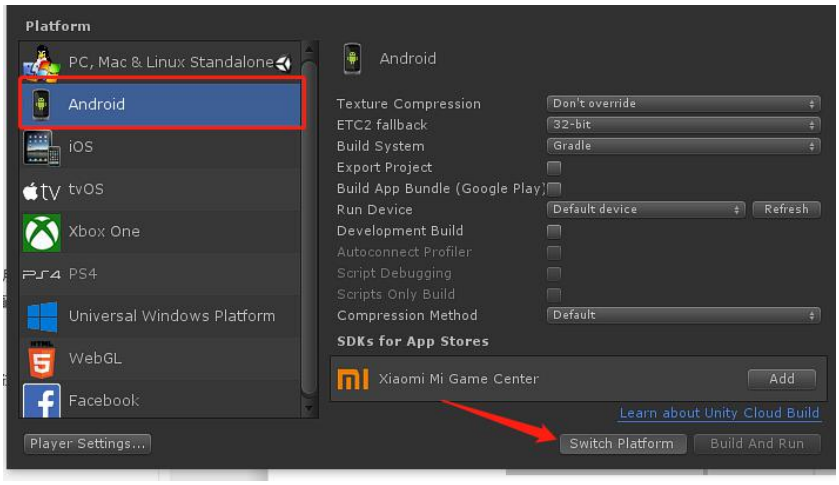
2) Create two Cubes, namely Cube1 and Cube2. The positions are shown in the following figure.





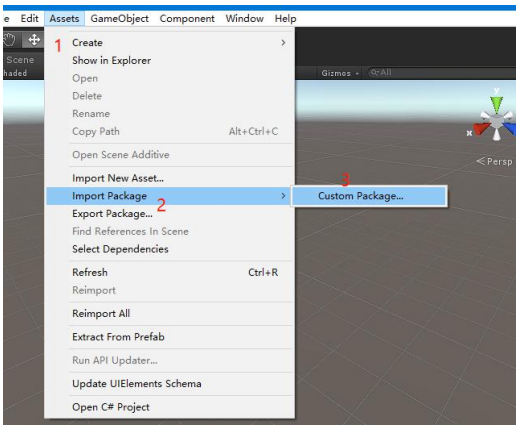
3) Select File->Build Settings, then select Android platform. Click Switch Platform to convert the project to Android project.



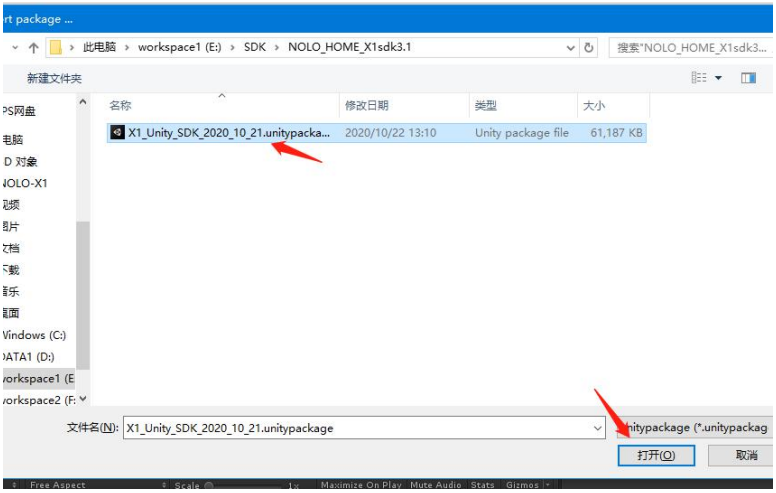


Step 2: Import SDK

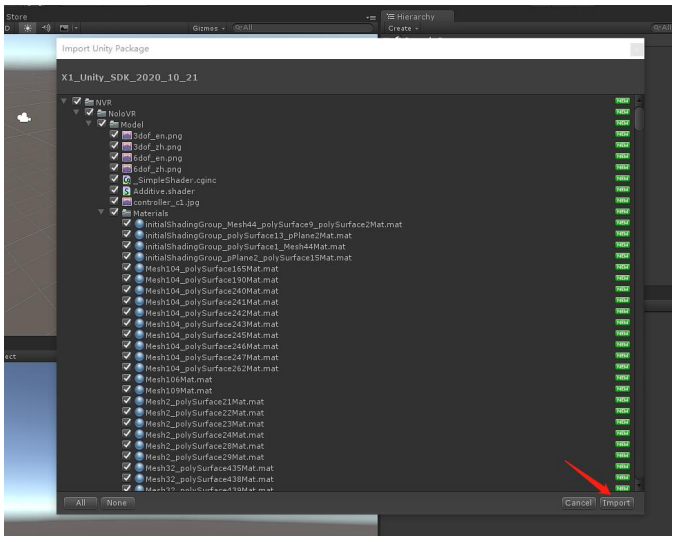
1) Select Assets -> Import Package -> Custem Package…

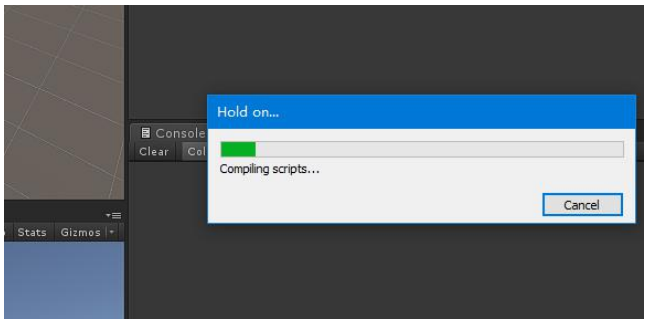


2） According to the pop-up, navigate to the corresponding directory, select SDK, click Open.



3) After opening the SDK, the import resource dialog box pops up, select Import.



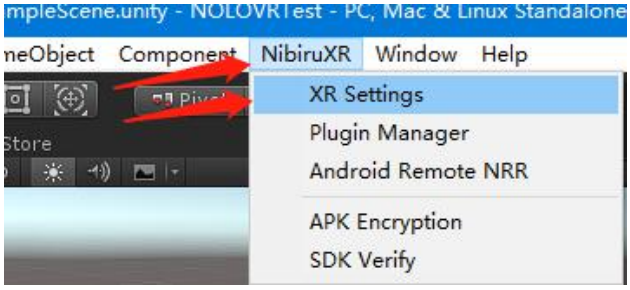


Step 3: Use SDK

1) Put NVR/Prefabs/NOLO into the scenario, delete the Main Camera comes with the scenario, and save.

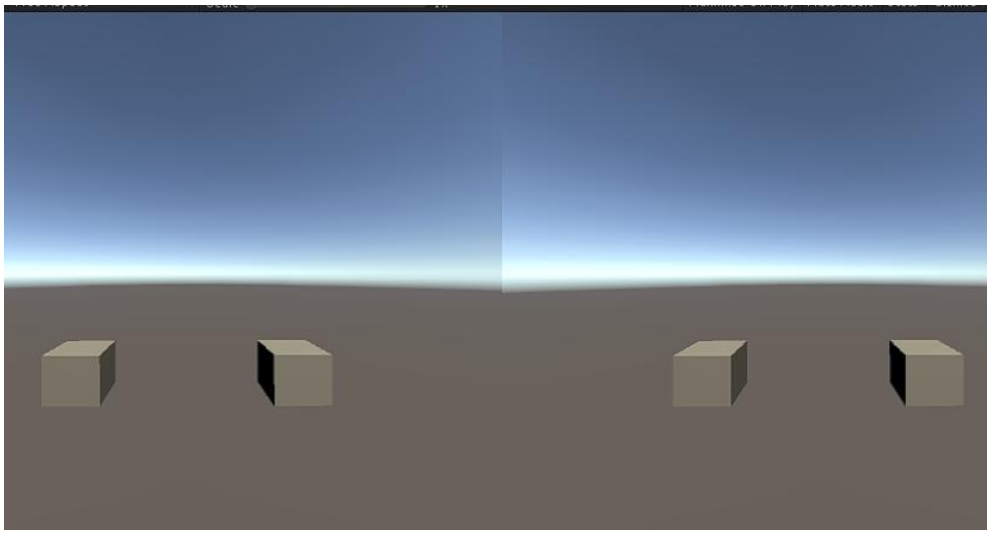
2) Select Nibiru->XR Settings, the project settings dialog box will pop

up. Select Middle in the Quality Setting. Select VR in the HMD type. Select ARMV7 in the Target Architectures. Click Confirm, the project environment configuration will automatically completed.



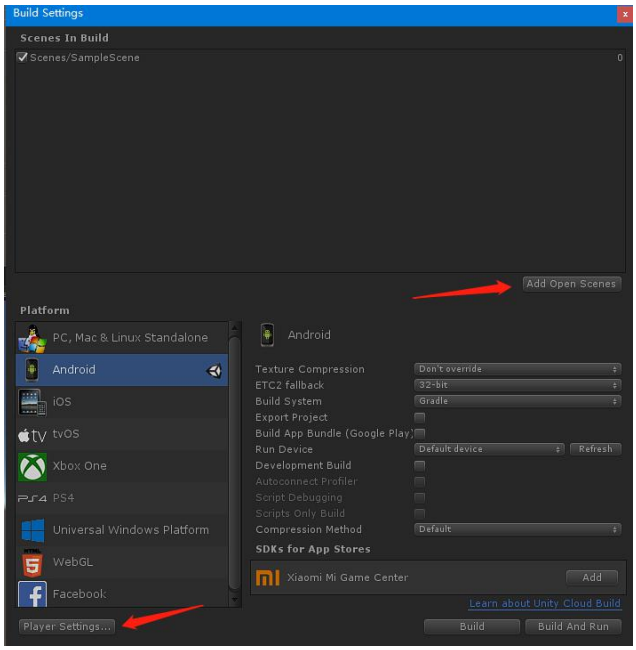
Step 4: Run a simulation

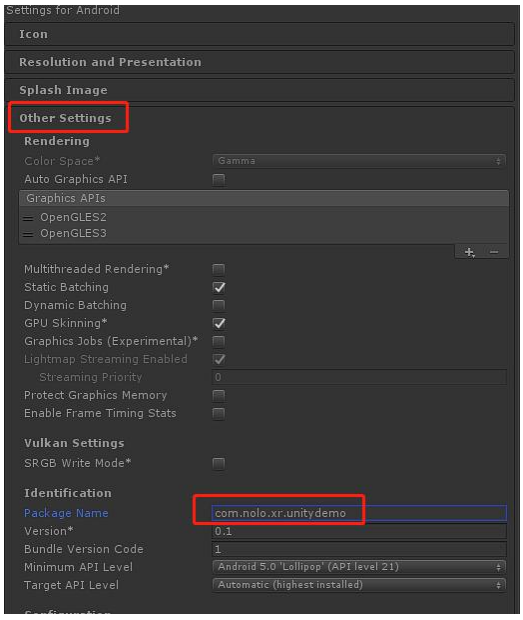
Click Run and observe it.



Step 5: Packaging

1) Select File->Build Settings. In the pop up panel, click Add Open Scenes, select Player Settings. Make sure the name of the package is com.nolo.xr.unitydemo. (If you want to amend the name of the package, please refer to the NOLO Unity SDK package name modification instruction document.) **Do not tick Development Build**





2) Create keystore.

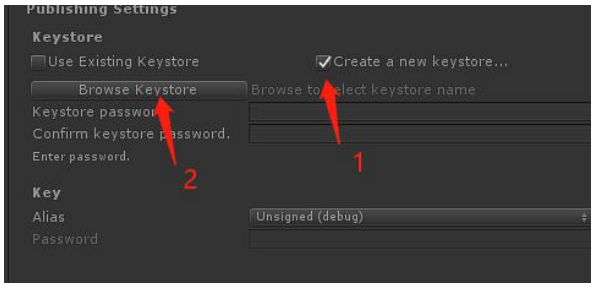
a) On the PlayerSettings interface, under the Publishing Settings page, check Create a new keystore, then click Browse password, and click save in the pop up window.

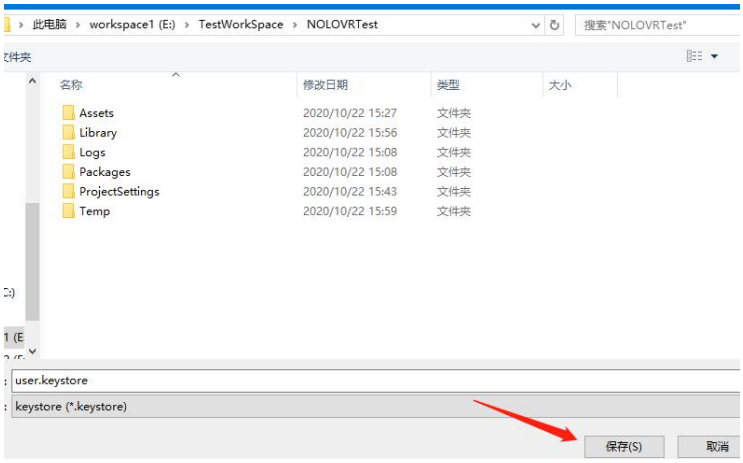
b) On the Publishing Settings page, fil in Keystore password:111111, confirm keystore password: 111111, then click Alias option, and

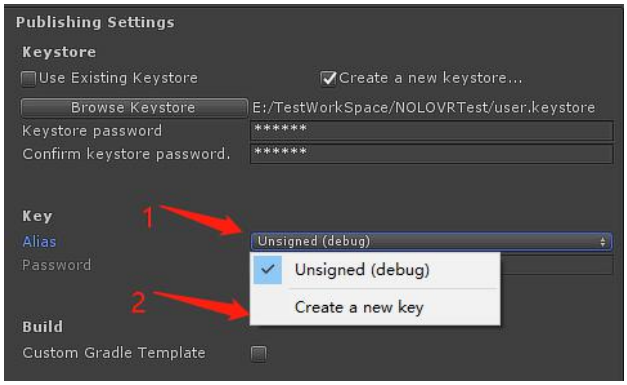
select Create a new key.

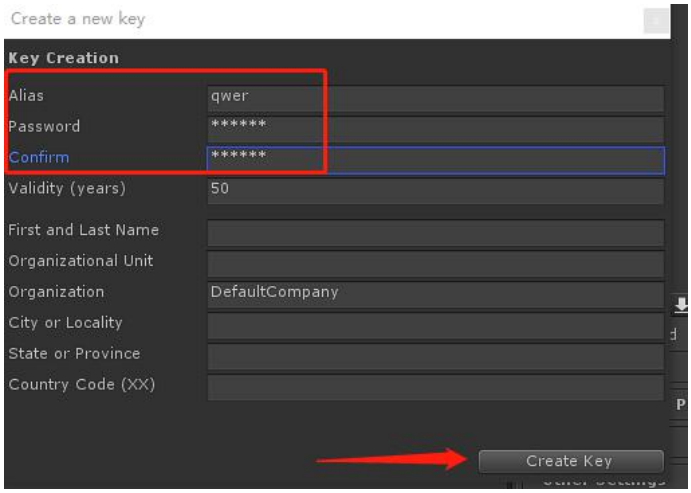
c) In the pop up dialog box, fill in qwer in Alias, 111111 in Password and 111111 in Confirm, then click Create key.

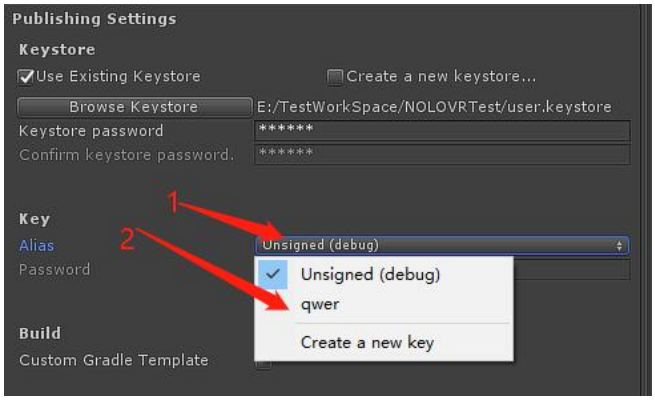
d) On the Publishing Settings page, select qwer in Alias, fill in 111111 in Password, then press Ctrl+s to save.





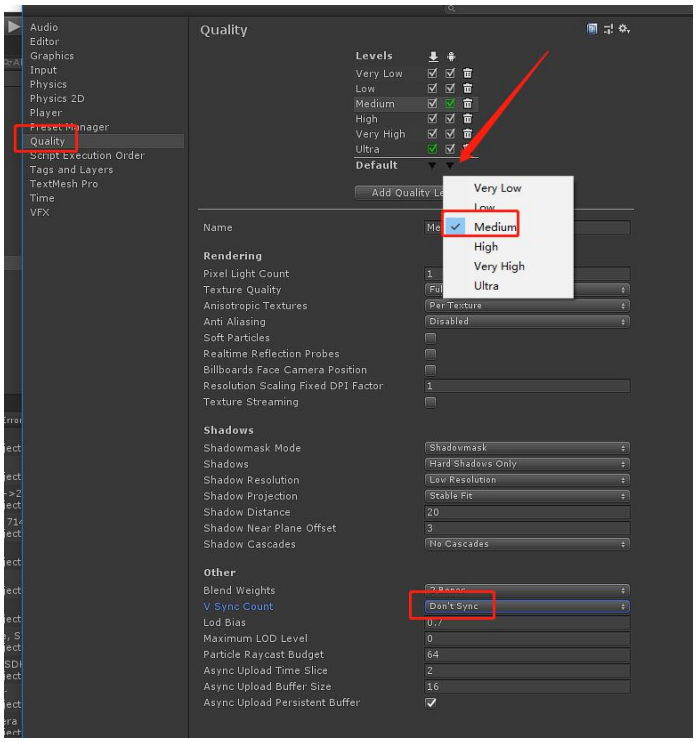






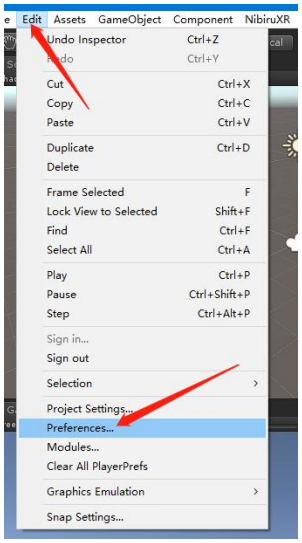
3) Set Quality

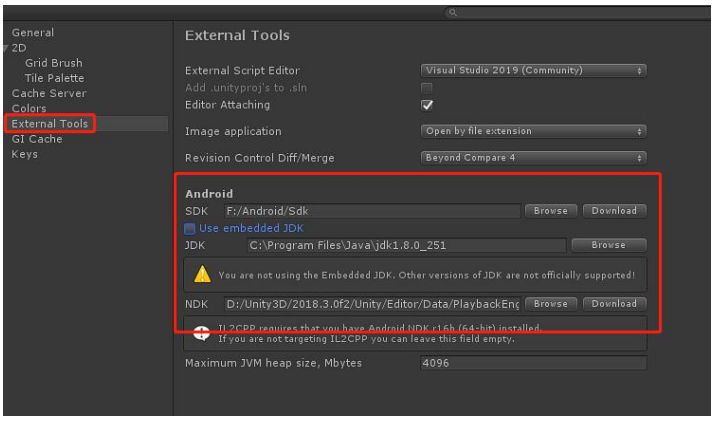
Select Edit->Project Settings from the menu, open Project Settings interface, select Quality item, set the Android platform default Levels to Medium and V Sync count to Don’ t Sync.



4) Set SDK, NDK, JDK

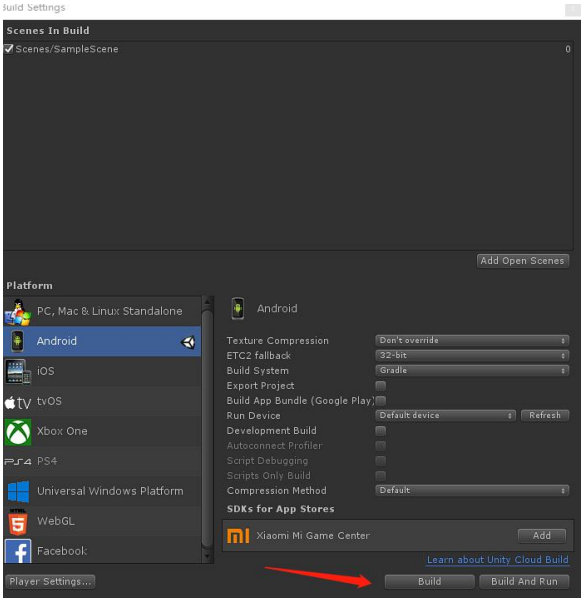
Select Edit->Preferences..., select External Tools, configure the appropriate SDK, JDK, NDK. If the configuration is missing, please click Download to download the corresponding version.

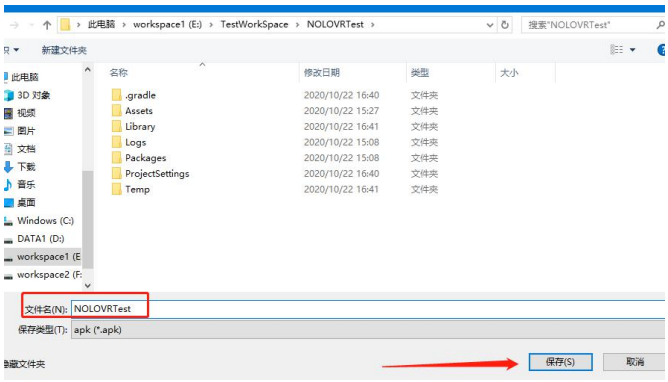




5) Packaging

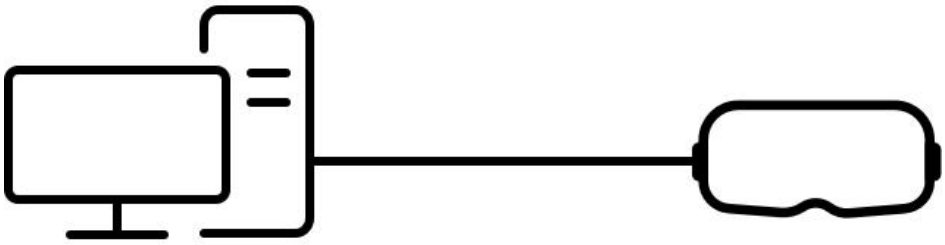
Open the project setting from the menu File->Build Settings, click Build. Enter NOLOVRTest (or any other names) as the file name, click Save.



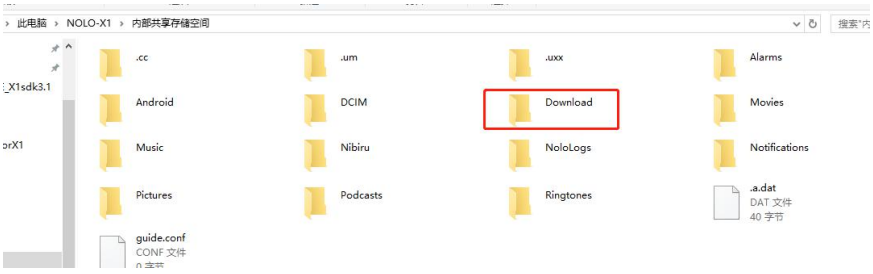


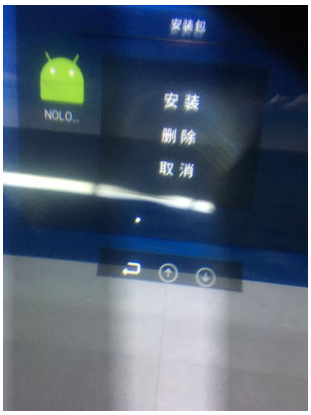
Step 6: Install it on X1 to test

1. Connect the NOLO X1 to the PC via data cable.



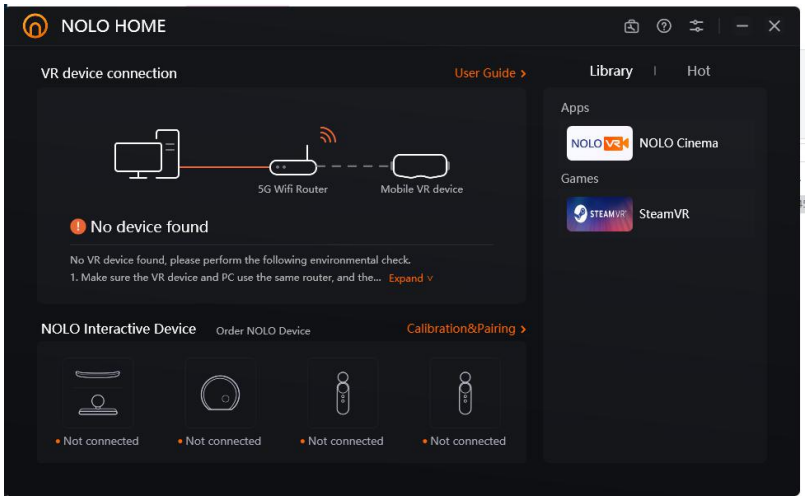
2. Install the apk into the X1
   a) Find the Download folder in the NOLO-X1 catalog, and copy the apk to the Download folder.
   b) In the NOLO X1, use the OK button to find the File Manager->installation package->NOLOVRTest.apk. Install it and start the apk to test the effect.





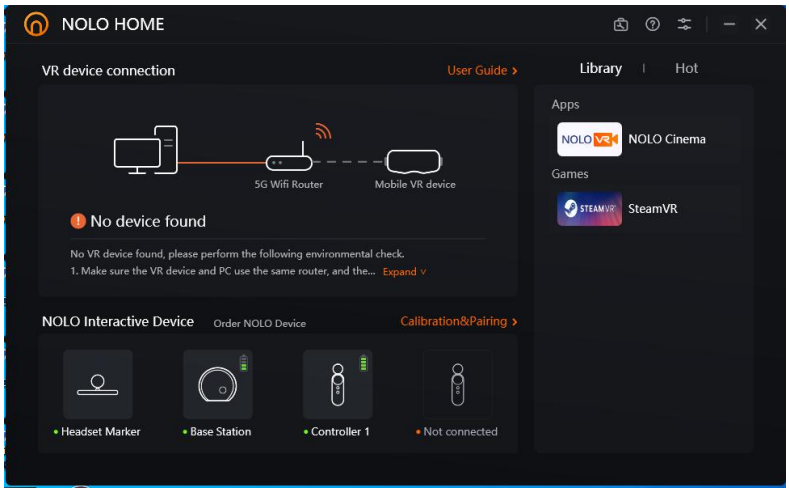
3. 2 Debugging Instructions

1. Open NOLO HOME windows client



2. Only connect the NOLO Headset Marker to the PC via USB cable. Turn on the Controllers and the Base Station.

3. After the NOLO HOME windows client displays the battery information of all NOLO devices, click the Run button on Unity to debug in the Unity Editor.



3. 3 Controller Button Instructions

4. API Description

4. 1 Button Events

function name	bool GetNoloButtonPressed()
function description	To check if a button is continuously being pressed down. (“pressed” status)
parameters	Enum NoloButtonID
return value	bool
prerequisites	NoloVR_Controller.GetDevice()

function name	bool GetNoloButtonDown()
function description	To check if a button is being pressed from “release” status. (“press” action)
parameters	Enum NoloButtonID

return value	bool
prerequisites	NoloVR_Controller.GetDevice()

function name	bool GetNoloButtonUp()
function description	To check if a button is being released from ‘pressed’ status. (“release” action)
parameters	Enum NoloButtonID
return value	bool
prerequisites	NoloVR_Controller.GetDevice()

Example:

Call of press the CV1 left handed controller Trigger Button:

```
NoloVR_Controller.GetDevice(NoloDeviceType.LeftController).GetNoloButtonDown(NoloButtonID.Trigger);
```

Call of press the C1 controller Trigger Button. It needs to use NoloC1ButtonID and convert to NoloButtonID, and the C1 controller only has the left handed controller. Example:

```
NoloVR_Controller.GetDevice(NoloDeviceType.LeftController).GetNoloButtonDown((NoloButtonID)NoloC1ButtonID.Trigger);
```

4.2 Standalone VR headset Button

Button Icon	Unity Key Assignments
OK	*KeyCode.JoystickButton0 or KeyCode.Joystick2Button0*
⮌	*KeyCode.Escape*

4. 3 Touch Events

function name	bool GetNoloTouchPressed()
function description	To check if the touchpad is touched. (“touched” status)
parameters	Enum NoloTouchID
return value	bool
prerequisites	NoloVR_Controller.GetDevice()

function name	bool GetNoloTouchDown()
function description	To check if the touchpad is being touched. (“touch” action)
parameters	Enum NoloTouchID
return value	bool
prerequisites	NoloVR_Controller.GetDevice()

function name	bool GetNoloTouchUp()
function description	To check if the touchpad is being released. (“release” action)
parameters	Enum NoloTouchID
return value	bool
prerequisites	NoloVR_Controller.GetDevice()

function name	Vector2 GetAxis()
function description	To get the coordinates of the touched spot on the touchpad.
parameters	Enum NoloTouchID: touchpad(default), other parameters are void

	(see appendix)
return value	Vector2
prerequisites	NoloVR_Controller.GetDevice()

Example:

```
NoloVR_Controller.GetDevice(NoloDeviceType.LeftController).GetNoloTouchDown(NoloTouchID.TouchPad);
```

4.4 Vibration Events

function name	void TriggerHapticPulse()
function description	To trigger controller vibration.
parameters	Vibration intensity: 50~100 (int)
return value	void
prerequisites	NoloVR_Controller.GetDevice()

4.5 Positional Information

function name	Nolo_Transform GetPose()
function description	Get device position.
parameters	Null
return value	Nolo_Transform
prerequisites	NoloVR_Controller.GetDevice()

4.6 Error Report

function name	void ReportError ()

function description	Log error messages.
parameters	string
return value	void
prerequisites	NoloVR_Playform.GetInstance()

4. 7 Connection Status of Device

function name	bool GetNoloConnectStatus()
function description	Get connection status of NOLO device
parameters	int/NoloDeviceType
return value	bool
prerequisites	NoloVR_Plugins.GetNoloConnectStatus()

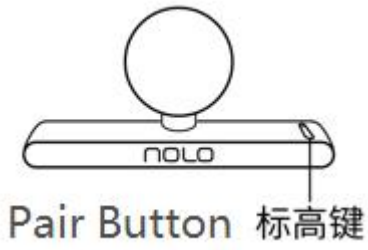
4. 8 Electricity of Device

function name	int GetElectricity()
function description	Get GetElectricity of NOLO device
parameters	int/NoloDeviceType
return value	int，Range（0~5）
prerequisites	NoloVR_Plugins.GetElectricity()

5. Notes

5. 1 Set Origin

Turn on all NOLO devices, place the Headset Marker on the ground, press the Pair Button on the Headset Marker. The Headset Marker’ s current position will be the origin in the game, aka the position of “NOLO” in the game engine. The origin’ s coordinates will be saved. This process only needs to be repeated if the Base Station has been moved.



5. 2 Set AppKey

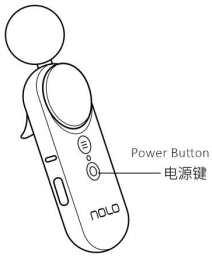
A game must acquire an AppKey to run properly with NOLO CV1. Please fill in the AppKey correctly in the NoloVR_Manager script on the NOLO GameObject.

Please contact dev@nolovr.com to apply Appkey and fill it in your Unity project.

Public Appkey：4e4f4c4f484f4d457eff82725bc694a5

5. 3 Reset Orientation

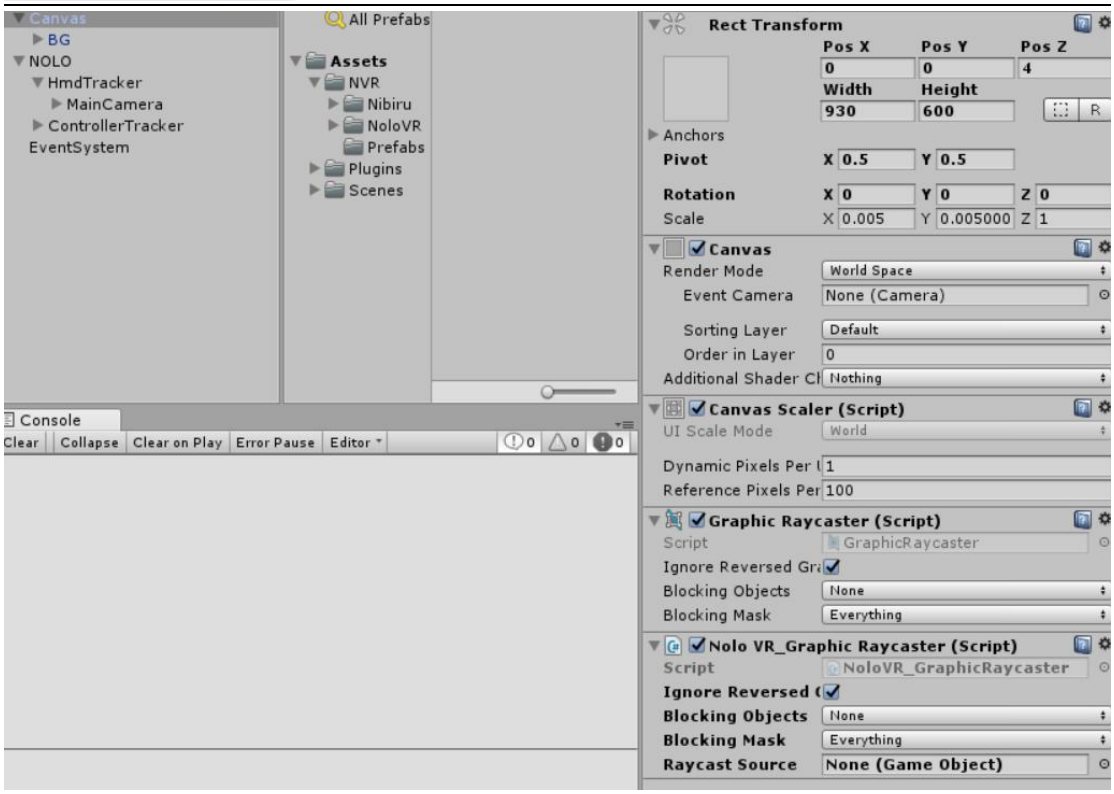
Upon starting a game, if the forward direction in the game does not point towards the NOLO Base Station, or the controller orientation seems a little odd, you may need to reset orientation by doing the following: Put on your headset, face the NOLO Base Station, point both controllers towards the Base Station, then double click the Power Button on each controller.



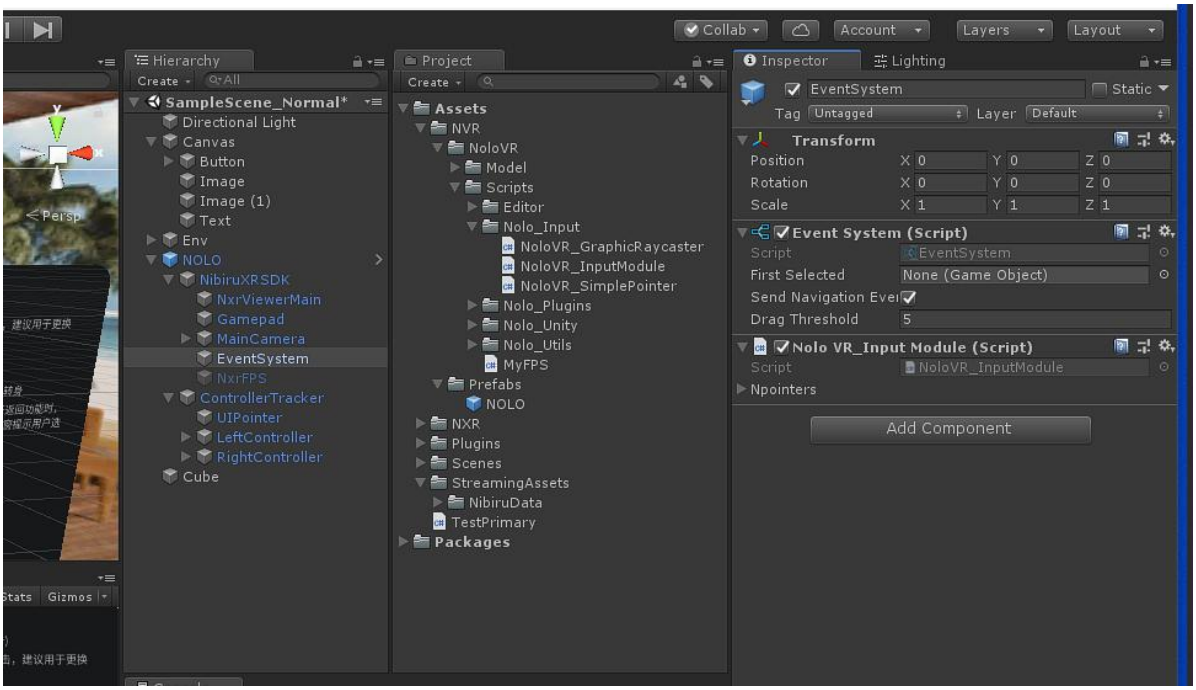
5. 4 Raycast Target

NOLO provides a set of UGUI raycast triggering scheme:

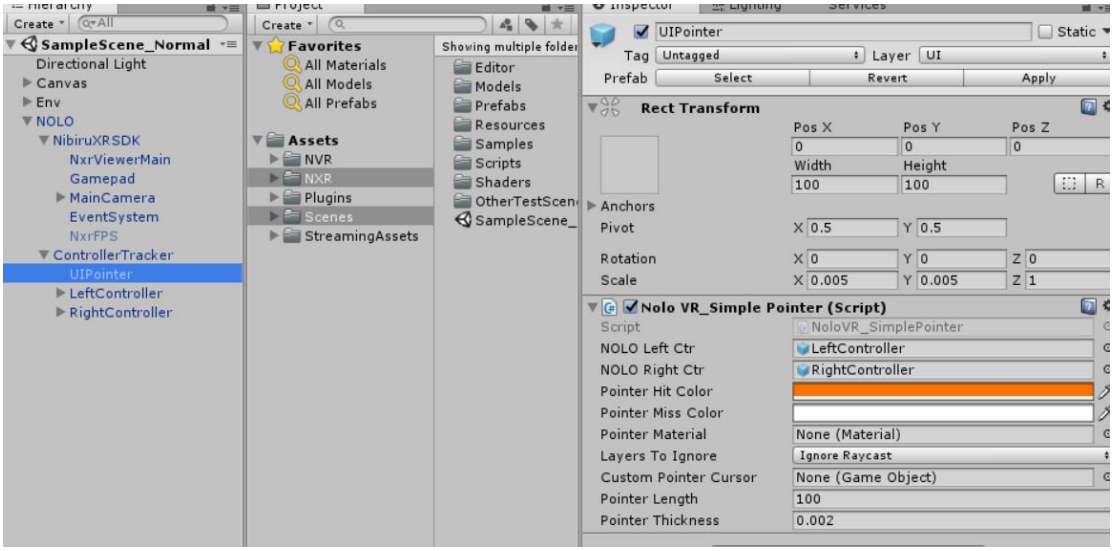
1. Add NoloVR_GraphicRaycaster.cs script to Canvas on UGUI



2. Add NoloVR_InputModule.cs script to EventSystem GameObject, For an example of interacting with 3D objects, see SampleScene_Normal scene.



3. Add the NoloVR_SimplePointer.cs script, switch the controller ray by the Trigger Button.



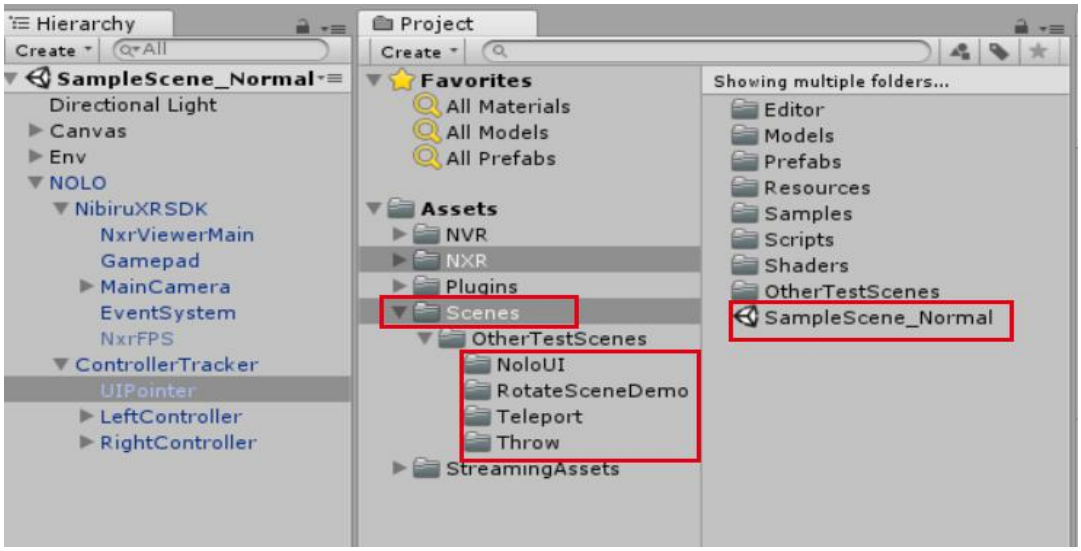
5. 5 Demo scene description provided by NOLO

1. NOLO Unity SDK provides different Demo Scenes for understanding the use of the NOLO sdk.

   SampleScene_Normal Scane: This scene is used to describe how to use the NOLO controller to trigger the UI, display the positioning information of the NOLO device, key information and keymapping function introduction.

   TelePort Scene: This scene shows how to use the NOLO controller to teleport.

   RotateSceneDemo: This scene shows how to use the NOLO controller to rotate and zoom in or out the entire game scene.

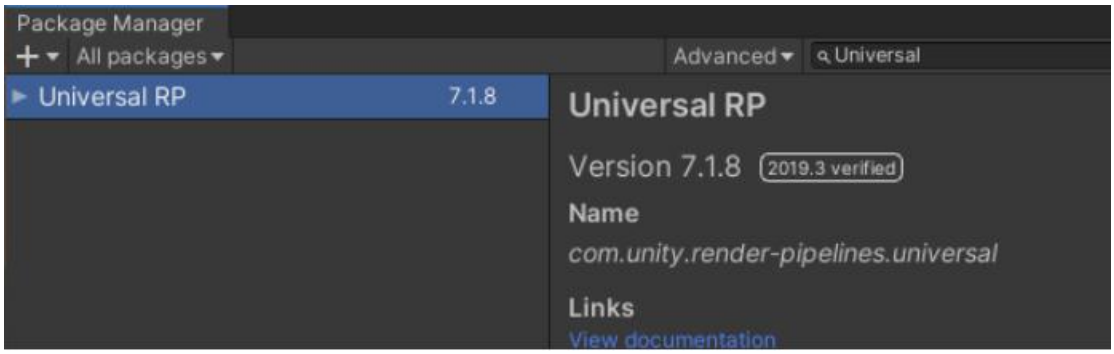


5. 6 Universal RP (Universal Rendering Pipeline) configuration instructions

Unity version: Unity2019.3.6

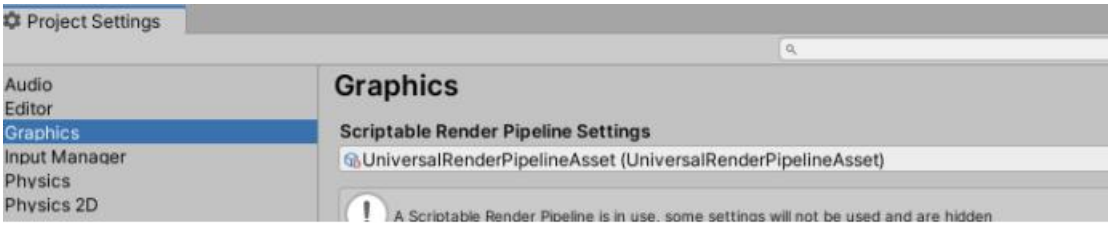
URP version: V7.1.8

1) Open Window/Package Manager, search Universal RP, then click Install.



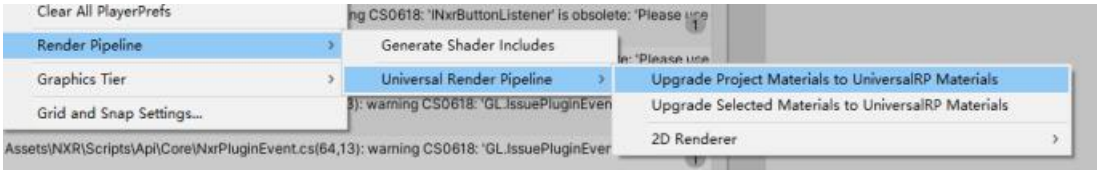
2) After installation, create UniversalRenderPipelineAsset, Assets/Create/Remdering/Universal Render Pipeline/Pipeline Asset

3) Open Edit/Project Settings/Graphics/, select the created PipelineAsset in the Scriptable Render Pipeline settings.



4） Open Edit/Project Settings/Quality interface, select Android category, select the created PipelineAsset in the Rendering.

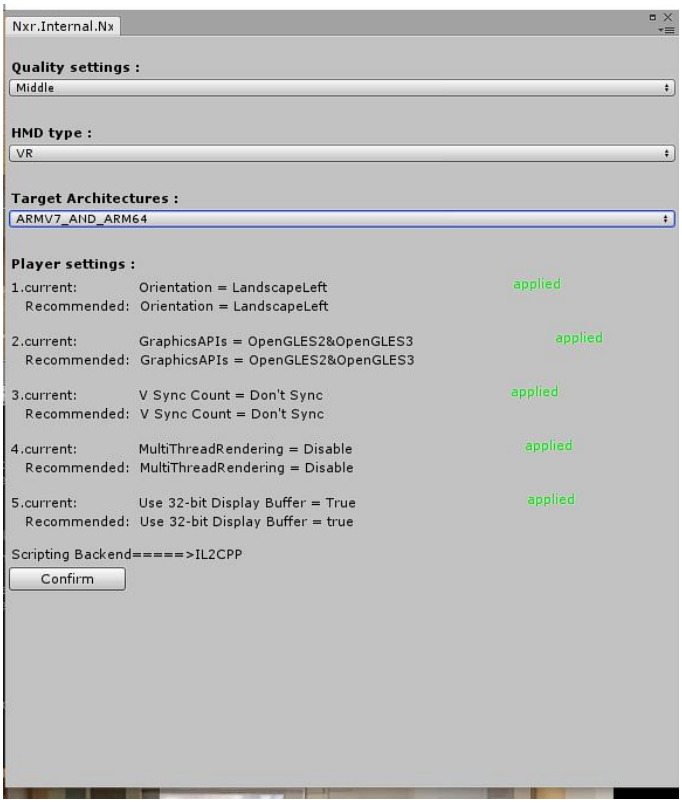
5) Upgrade engineering materials



Note: Cancel SPR Batcher/Dynamic Batching in the UniversalRenderPipelineAsset, otherwise memory leaks will occur.

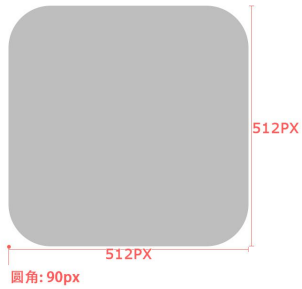
5. 7 NOLO SDK supports Armv8

1) Package setting: Click NibiruXR->XR Settings in the menu, the setting panel will pop up. Then operate as shown below.



5. 8 Packaging specification

Game icon, 512 x 512, png format with rounded corners 90px, less than 200kb. The icon needs to add “NOLO_HOME” corner mark, see following template.



5.9 Contact details

If you have any question or suggestion, please feel free to contact us at dev@nolovr.com.