**NOLO SDK for Unity**

**Documentation**

LYRobotix Co., Ltd

May 2017

**Directory**

[1. Introduction To The NOLO SDK 3](#_Toc482885419)

[2. Development Environment 3](#_Toc482885420)

[3. Release Notes 3](#_Toc482885421)

[4. NOLO SDK Startup Guide 4](#_Toc482885422)

[4.1 Import NOLO SDK 4](#_Toc482885423)

[4.2 Explanation Of The File Folders 4](#_Toc482885424)

[4.2.1 Example File Folder 4](#_Toc482885425)

[4.2.2 Icon File Folder 4](#_Toc482885426)

[4.2.3 Model File Folder 5](#_Toc482885427)

[4.2.4 Prefabs File Folder 5](#_Toc482885428)

[4.2.5 Scripts File Folder 5](#_Toc482885429)

[4.3 Example 5](#_Toc482885430)

[4.4 Reference Design 6](#_Toc482885431)

[4.4.1 General 6](#_Toc482885432)

[4.4.2 Input Test 6](#_Toc482885433)

[4.4.3 Recenter 6](#_Toc482885434)

[4.4.4 Rotate Scene Dome 7](#_Toc482885435)

[4.4.5 Teleport 7](#_Toc482885436)

[4.4.6 TurnAround Demo 7](#_Toc482885437)

[5. SDK Functional Module 8](#_Toc482885438)

[5.1 NoloVR\_Manager.cs 8](#_Toc482885439)

[5.2 NoloVR\_TrackedDevice.cs 8](#_Toc482885440)

[5.3 NoloVR\_Controller.cs 9](#_Toc482885441)

[5.4 NoloVR\_PlayArea.cs 10](#_Toc482885442)

[6. Android Configuration 11](#_Toc482885443)

# Introduction to the NOLO SDK

NOLO SDK for Unity is a development kit which is provided by LYRobotix for the Unity developers. The NOLO SDK fits NOLO CV1, which is also provided by LYRobotix. Using the development kit, developers can get position data for NOLO device headset marker and controllers, rotation data for controllers, all buttons information for controllers, and vibration information，the SDK mainly aims at Android equipment.

# Development Environment

The development environment of NOLO SDK for Unity is Unity5.4.1(Win 64bit). Supported Unity version is 5.4.1 or above, and JDK version is jdk1.8.0\_101.

# Release Notes

|  |  |
| --- | --- |
| Version | Content |
| NoloVR\_SDK\_1.1.0 | 1. Supporting NOLO DK2 and NOLO CV1 2. Processing the positioning data,rotation data and buttons information of NOLO devices 3. Reference design |

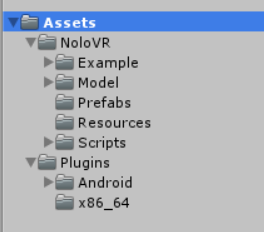
# 4. NOLO SDK Startup Guide

## 4.1 Import NOLO SDK

NOLO SDK for Unity is a kind of .unitypackage, which can be imported from the menu Assets->Import Package->Custom Package into Unity for developing.

## 4.2 Explanation Of The File Folders

The directory of NOLO SDK for Unity file folders shown as below:



### 4.2.1 Example File Folder

Including the reference design of NOLO SDK, as detailed in the next section。

### **4.2.2** **Icon File Folder**

Including the icon material of NOLO. Developers can add the NOLO icon in the upper right corner of their application, showing the application support NOLO equipment.

### **4.2.3** **Model File Folder**

Including the material model of NOLO.

### **4.2.4** **Prefabs File Folder**

Including NoloManager.prefab. It can be used directly for rapid development.

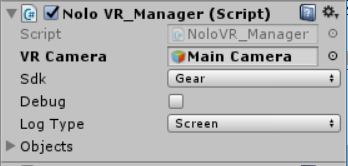
### **4.2.5** **Scripts File Folder**

Including the NOLO SDK Script files.

## 4.3 Example

Create a new scene, then push the NoloManager.prefab under Prefabs file into the scene. Drag the VRCamera you are using into the Hmd(camera) under NoloManager, as a sub-object of Hmd(camera). Clear the position and rotation. Find the NoloVR\_Manager.cs script under the NoloManager. Drag the object whose attitude changed to VR Camera when the game is running. Then you can complete the design, as shown below.





## 4.4 Reference Design

### **4.4.1** **General**

Used to view NOLO basic data.

In the Test scene, the UI\_Test.cs script is used to display the data information provided by the NOLO device to Unity in the UI interface, which facilitate debugging during the development process.

### **4.4.2** **Input Test**

Used to test the NOLO buttons function of controllers.

In the InputTest scene, the Input\_Test.cs script is used to test the buttons state of controllers, the NOLO device, which facilitate debugging during the development process.

### **4.4.3** **Recenter**

Reference design, the realization of the function is to double-click any system button of controllers to reset the camera Yaw value.

Add NoloVR\_Recenter.cs script into NoloManager to reset camera YAW. The official recommendation is to double-click system button (power button) of any controller to reset the camera Yaw value. As Gear an example, in the actual development, the developer should to use a different method to replace the method in 30th line of NoloVR\_Recenter.cs, according to the difference of VRCamera.

### **4.4.4** **Rotate Scene Dome**

Reference design, the realization of the function is pressing the Grip buttons of two controllers. Then you can rotate, zoom, and move the scene.

Add NoloVR\_Recenter.cs script into NoloManager. Place all the objects which need to be changed in the scene under a parent node, and add the parent node into Object Parents under the NoloVR\_RotateScene.cs script. Is change scale means whether you need to modify scale. Is change rotation means whether you need to modify rotation.

### 4.4.5 Teleport

Reference design for implementing NOLO transfer function.

Add NoloVR\_Teleport.cs script into any controller (leftcontroller or rightcontroller). That will do.

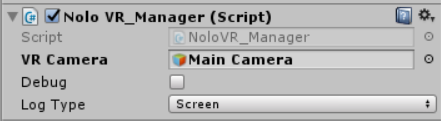
### 4.4.6 TurnAround Demo

Reference design, the realization of the function is to Double-click the menu button of any controller to implement turn around the game scene 180 degrees.

Add NoloVR\_Recenter.cs script into NoloManager. That will do.

# 5. SDK Functional Module

## 5.1 NoloVR\_Manager.cs



VR Camera: Get the VR camera in the scene. The developer can use the third-party SDK, the corresponding VR Camera assigned to it. But note that the VR Camera, must be the game object which have the attitude data, rather than a simple camera.

Debug and Log Type: Used to select whether to open NOLO Debug mode, which the Debug mode is divided into Console and Screen, respectively, the console print and display on the screen. For the official application, do not check Debug.

## 5.2 NoloVR\_TrackedDevice.cs



Device Type：Indicates the type of device, respectively, they are hmd(headset marker), leftcontroller, rightcontroller and base station.

## 5.3 NoloVR\_Controller.cs

NoloVR\_Controller.GetDevice(NoloDeviceType deviceIndex);

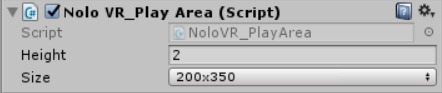
NoloVR\_Controller.GetDevice(NoloVR\_TrackedDevice trackedobject);

Used to get all the information you want to listen on the NOLO devices.

The specific methods are as follow:

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | Parameters | Return Value | Explain |
| GetPose |  | Nolo\_Transform | Return location and attitude of NoloDevice |
| GetNoloButtonPressed | Uint buttonMask  NoloButtonID button | bool | buttonMask:  1<<0 touchpad  1<<1 trigger  1<<2 menu  1<<3 system  1<<4 grip  NoloButtonID:  Touchpad,trigger,menu,system,grip |
| GetNoloButtonDown | Uint buttonMask  NoloButtonID button | bool | buttonMask:  1<<0 touchpad  1<<1 trigger  1<<2 menu  1<<3 system  1<<4 grip  NoloButtonID:  Touchpad,trigger,menu,system,grip |
| GetNoloButtonUp | Uint buttonMask  NoloButtonID button | bool | buttonMask:  1<<0 touchpad  1<<1 trigger  1<<2 menu  1<<3 system  1<<4 grip  NoloButtonID:  Touchpad,trigger,menu,system,grip |
| GetNoloTouchPressed | Uint touchMask  NoloTouchID touch | bool | touchMask:  1<<0 touchpad  NoloTouchID:  Touchpad |
| GetNoloTouchDown | Uint touchMask  NoloTouchID touch | bool | touchMask:  1<<0 touchpad  NoloTouchID:  Touchpad |
| GetNoloTouchUp | Uint touchMask  NoloTouchID touch | bool | touchMask:  1<<0 touchpad  NoloTouchID:  Touchpad |
| GetAxis | NoloTouchID(The default is touchpad, the other is invalid) | Vector2 | X Range (-1~1)  Y Range (-1~1) |
| GetTrackingStaus |  | NoloTrackingStatus | NoloTrackingStatus.NotConnect  NoloTrackingStatus.Normal  NoloTrackingStatus.OutofRange |
| TriggerHapticPulse | Int intensity (means vibration intensity) |  | The parameter range is (0 ~ 100) , larger is more intense |

## 5.4 NoloVR\_PlayArea.cs



Mainly used to suggest the developer the possible activity range of users in the scene , no other role.

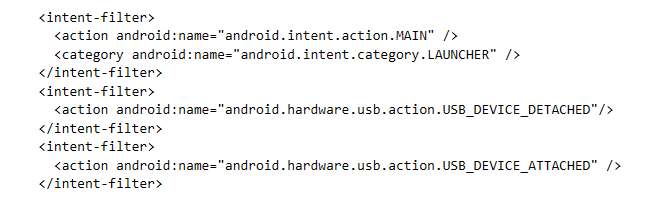
Height：Altitude.

Size：Length × Width.

# Android Configuration

Add the following content in AndroidManifest.xml:

Add the following content in the main activity:



For details, refer to the AndroidManifest.xml file in the SDK.