NOLO SDK for Unity Documentation

LYRobotix Co., Ltd

May 2017

Directory

1.Introduction

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 is mainly suitable for Android equipment.

2.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.

3.Release Notes

Version	Content
NoloVR_SDK_1.1.6	1.Supporting NOLO DK2 and NOLO CV1 2.Processing the positioning data and buttons information of NOLO devices 3.Reference design

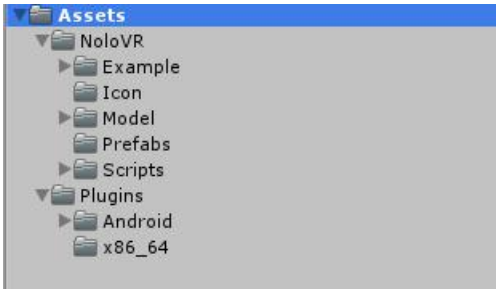
4. NOLO SDK Startup Guide

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 section 4.4.

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.

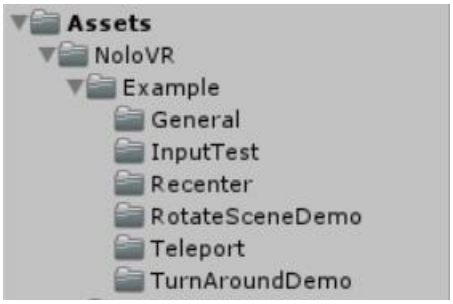
4.3 Example

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 by Example

The directory Example file folders shown as below:



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 Rotate Scene demo

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.

4.4.4 Teleport

Reference design for implementing NOLO transfer function

Add NoloVR_Teleport.cs script into any controller (leftcontroller or rightcontroller). That will do.

5. SDK Functional Module

5.1 NoloVR_Manager.cs

The NoloVR_Manager.cs module shown as below:

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.

Automatic Connection：NOLO devices will be connected automaticly if you choose this

option, or you can build the connnection manually anywhere using the methods below.

NOLO_Events.Send(NOLO_Events.EventsType.ConnectNoloDevice);

Turn Around Button Type:Customize the hot-key for 180 degree turn around function. You can choose different hotkeys for the function or disable it.

5.2 NoloVR_TrackedDevice.cs

The NoloVR_TrackedDevice.cs module shown as below:

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

5.3 NoloVR_Controller.cs

The NoloVR_Controller.cs module include two interface functions.

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.

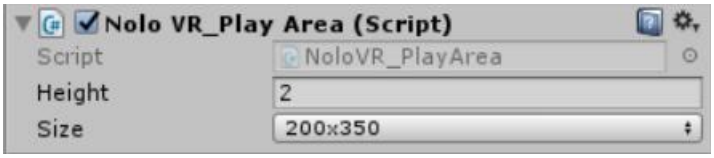
The specific methods are as follow:

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

The NoloVR_PlayArea.cs module mainly used to suggest the developer the possible activity range of users in the scene, no other role. And it is shown as below.



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

Height：Altitude

Size：Length × Width

6. Android Configuration

Add the following content in AndroidManifest.xml:

```
<uses-permission android:name="android.hardware.usb.host" />
<uses-feature android:name="android.hardware.usb.host" android:required="true"/>
```

Add the following content in the main activity:

```
<intent-filter>
  <action android:name="android.intent.action.MAIN" />
  <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
<intent-filter>
  <action android:name="android.hardware.usb.action.USB_DEVICE_DETACHED"/>
</intent-filter>
<intent-filter>
  <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
</intent-filter>
```

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