

# **NOLO VR Unity SDK**

## **Documentation**

NOLO Co., Ltd

February 2020

## 目录

1. Overview.....	3
1.1 About NOLO.....	3
2. Preparation for development.....	3
3. Instructions.....	4
3.1 Quick Start.....	4
3.2 NOLO Prefabs.....	6
3.3 Debugging Instructions.....	7
4. API Description.....	8
4.1 Button Events.....	8
4.2 Touch Events.....	9
4.3 Vibration Events.....	10
4.4 Positional Information.....	10
4.5 Error Report.....	11
4.6 Connection Status of Device.....	11
4.7 Electricity of Device.....	12
5. Notes.....	12
5.1 Set Origin.....	12
5.2 Set AppKey.....	12
5.3 Reset Orientation.....	13
5.4 Raycast Target.....	13

# 1. Overview

## 1.1 About NOLO

NOLO is dedicated to combine desktop-grade VR gaming experience with the convenience of mobile VR devices, redefining a mobile VR gaming experience like never before.

NOLO kit is compatible with some 87,000,000 VR headsets of all kinds currently on the market, indicating huge market potential. In addition, we've partnered with VR headset companies, robotic companies, and drone companies around the globe.

## 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](http://download.nolovr.com/download/NOLO_home_PC.html)

Please contact [dev@nolovr.com](mailto: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.

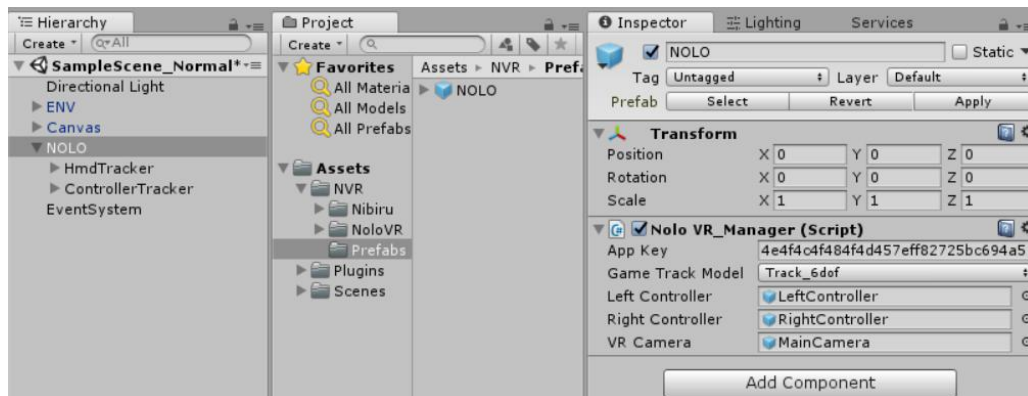
Public Appkey: 4e4f4c4f484f4d457eff82725bc694a5

## 3. Instructions

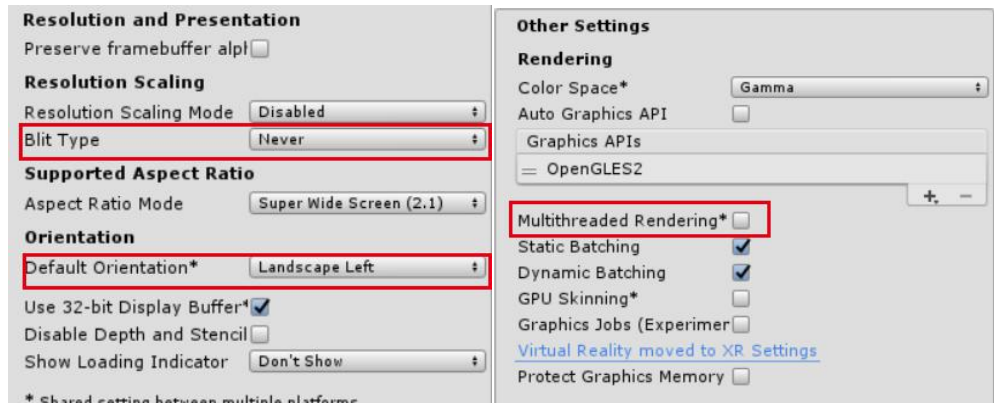
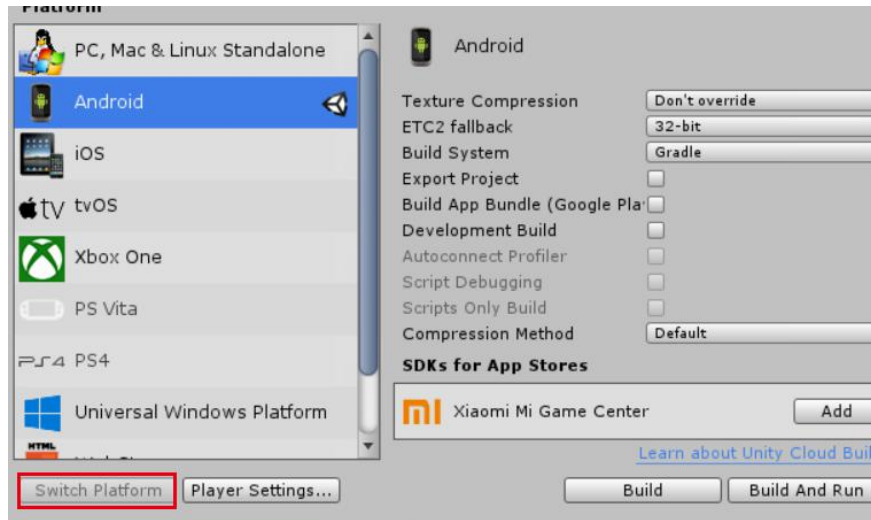
### 3.1 Quick Start

#### (1) All-in-one project

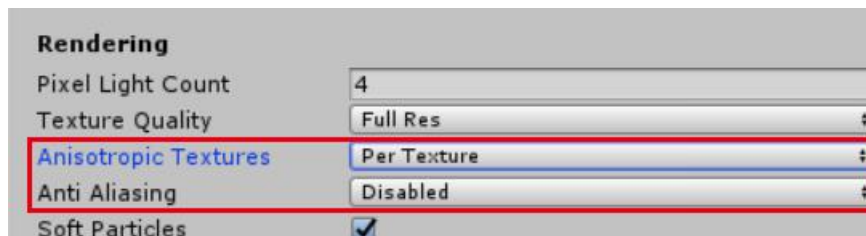
- 1) Create a new Unity project and import the NOLO VR Unity SDK into it.
- 2) Create a new scenario and put NVR/Prefabs/NOLO into it and save.
- 3) Fill the Appkey in the following location

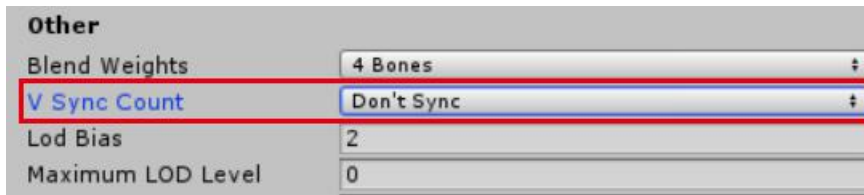


- 4) Player Settings: The orientation must be set to 'Landscape Left' in 'Resolution and Presentation', while the 'Multithreaded Rendering' must be set to unavailable in 'Other Settings'.



5) Quality Settings: In 'Rendering', the 'Anisotropic Textures' is set to 'Per Texture', and 'Anti Aliasing' to 'Disabled'. 'Sync Count' is set to 'Don't Sync' in 'Other'.

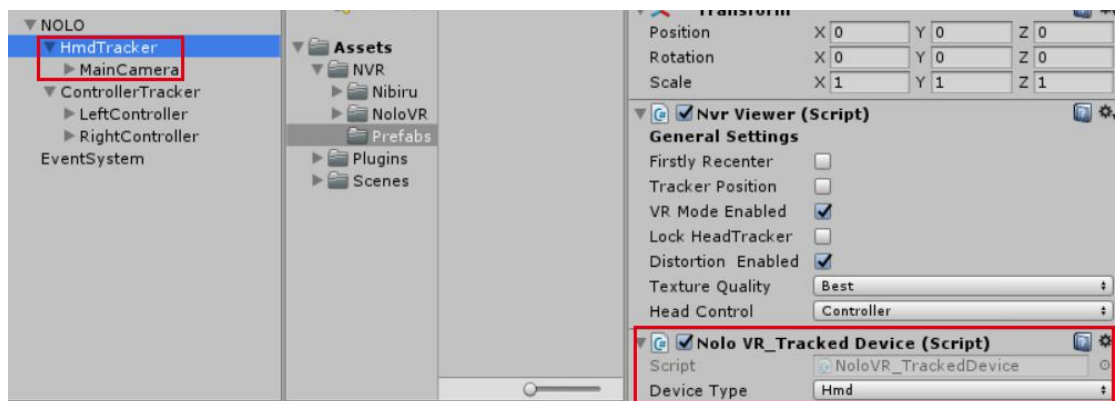




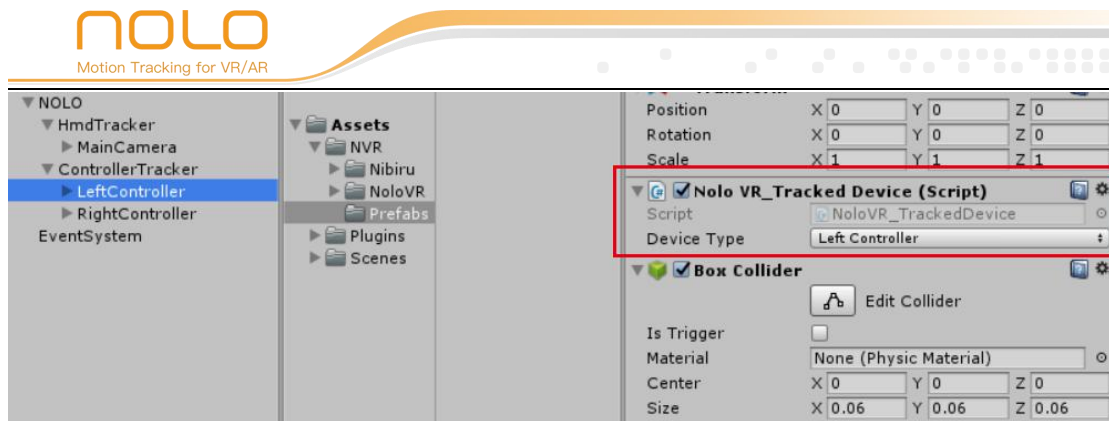
6) Fill in the correct package name information to package the settings and send it to NOLO X1 to run.

## 3.2 NOLO Prefabs

1. Head component, including head positioning information, camera information



2. Left-hand component, contains left-hand positioning information



3. Right-hand component, contains right-hand positioning information



### 3.3 Debugging Instructions

Debugging in Unity Editor: Please only connect the NOLO headset marker to the PC with a USB cable. Open the NOLO HOME windows client and turn on other NOLO devices. Wait until the NOLO HOME displays the battery information of all NOLO devices, then click the Run button to debug in the Unity Editor.

Debug on Android clients: Install NOLO HOME on the mobile device, fill the correct Appkey in the project, and use the test key for debugging before the Appkey is reviewed. Also connect the NOLO headset marker with the OTG cable to the mobile phone or all-in-one. If it is prompted

that "Run NOLO HOME to access USB device?", click OK to get the NOLO data in your APP.

## 4.API Description

### 4.1 Button Events

<b>function name</b>	bool GetNoloButtonPressed()
<b>function description</b>	To check if a button is continuously being pressed down. ( "pressed" status)
<b>input parameters</b>	Enum NoloButtonID
<b>return value</b>	bool
<b>prerequisites</b>	NoloVR_Controller.GetDevice()

<b>function name</b>	bool GetNoloButtonDown()
<b>function description</b>	To check if a button is being pressed from "release" status. ( "press" action)
<b>input parameters</b>	Enum NoloButtonID
<b>return value</b>	bool
<b>prerequisites</b>	NoloVR_Controller.GetDevice()

<b>function name</b>	bool GetNoloButtonUp()
----------------------	------------------------



<b>function description</b>	To check if a button is being released from 'pressed' status. ( "release" action)
<b>input parameters</b>	Enum NoloButtonID
<b>return value</b>	bool
<b>prerequisites</b>	NoloVR_Controller.GetDevice()

## 4.2 Touch Events

<b>function name</b>	bool GetNoloTouchPressed()
<b>function description</b>	To check if the touchpad is touched. ( "touched" status)
<b>input parameters</b>	Enum NoloTouchID
<b>return value</b>	bool
<b>prerequisites</b>	NoloVR_Controller.GetDevice()

<b>function name</b>	bool GetNoloTouchDown()
<b>function description</b>	To check if the touchpad is being touched. ( "touch" action)
<b>input parameters</b>	Enum NoloTouchID
<b>return value</b>	bool
<b>prerequisites</b>	NoloVR_Controller.GetDevice()

<b>function name</b>	bool GetNoloTouchUp()
<b>function description</b>	To check if the touchpad is being released. ( "release" action)

<b>input parameters</b>	Enum NoloTouchID
<b>return value</b>	bool
<b>prerequisites</b>	NoloVR_Controller.GetDevice()

<b>function name</b>	Vector2 GetAxis()
<b>function description</b>	To get the coordinates of the touched spot on the touchpad.
<b>input parameters</b>	Enum NoloTouchID: touchpad(default), other parameters are void (see appendix)
<b>return value</b>	Vector2
<b>prerequisites</b>	NoloVR_Controller.GetDevice()

## 4.3 Vibration Events

<b>function name</b>	void TriggerHapticPulse()
<b>function description</b>	<b>To trigger controller vibration.</b>
<b>input parameters</b>	<b>Vibration intensity: 0~100 (int)</b>
<b>return value</b>	void
<b>prerequisites</b>	NoloVR_Controller.GetDevice()

## 4.4 Positional Information

<b>function name</b>	Nolo_Transform GetPose()
<b>function description</b>	Get device position.

<b>input parameters</b>	Null
<b>return value</b>	Nolo_Transform
<b>prerequisites</b>	NoloVR_Controller.GetDevice()

## 4.5 Error Report

<b>function name</b>	void ReportError ()
<b>function description</b>	Log error messages.
<b>input parameters</b>	string
<b>return value</b>	void
<b>prerequisites</b>	NoloVR_Playform.GetInstance()

## 4.6 Connection Status of Device

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

## 4.7 Electricity of Device

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

## 5. Notes

### 5.1 Set Origin

Turn on all NOLO devices, place the headset marker on the ground, press the 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. An AppKey will be generated automatically when developers apply for their game on NOLO Developer Center. Please add NoloVR\_Manager script to

your project workspace, and fill in the AppKey.

When the game does not upload NOLO HOME, you can use this public Appkey for development testing.

Public Appkey: 4e4f4c4f484f4d457eff82725bc694a5

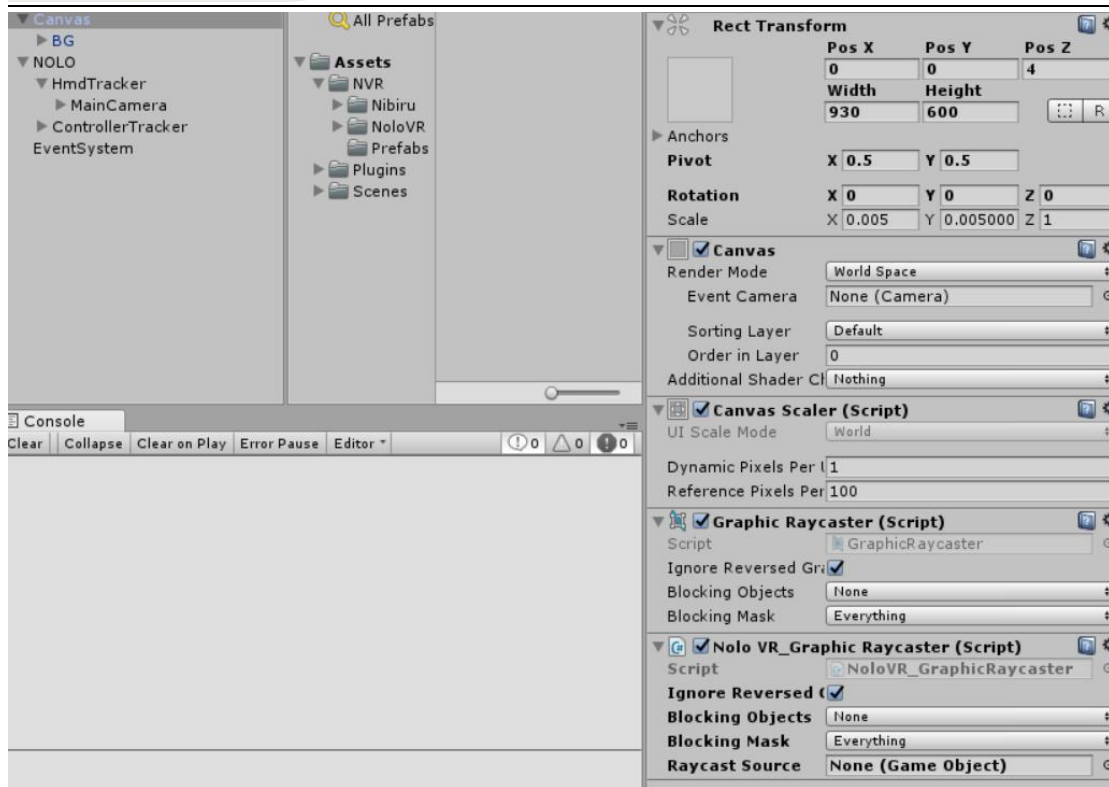
## 5.3 Reset Orientation

Upon starting a game, if the forward direction in the game does not point towards 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 Nolo Base Station, then double click the power button on either controller.

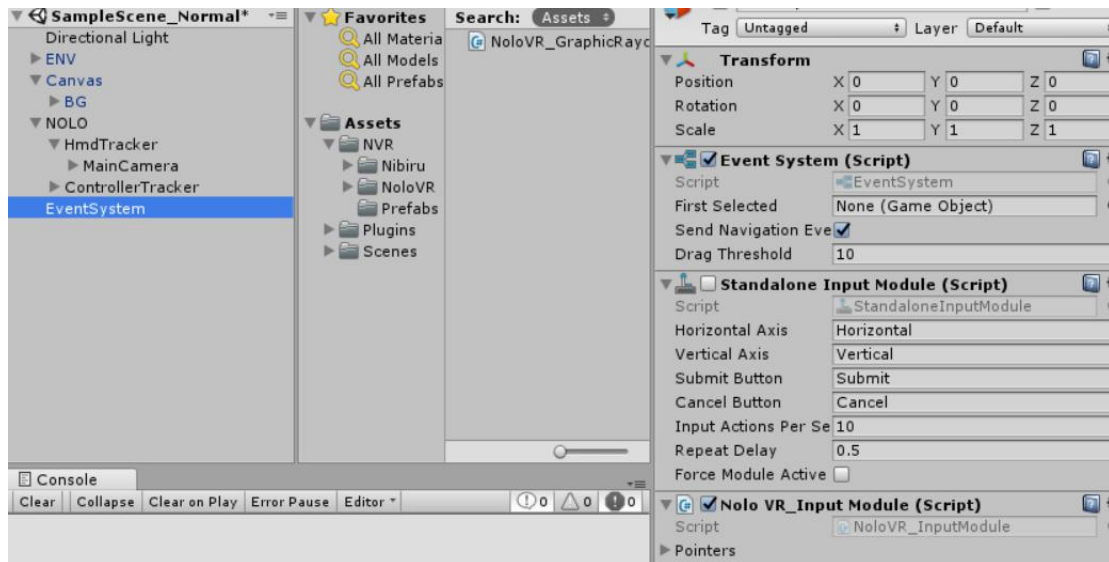
## 5.4 Raycast Target

NOLO provides a set of UGUI recast triggering scheme:

1. Add NoloVR\_GraphicRaycaster.cs script to Canvas in UGUI



## 2. Add NoloVR\_InputModule.cs script to EventSystem in UGUI



## 3. Add the NoloVR\_SimplePointer.cs script to one of the controller to trigger UGUI related components with the secondary controller.

