**NOLO VR Unity SDK**

**接口说明**

北京凌宇智控科技有限公司

2018年3月

目录

**[一、 简介](#_Toc4873_WPSOffice_Level1)** **[3](#_Toc4873_WPSOffice_Level1)**

[1.NOLO VR Unity SDK](#_Toc29840_WPSOffice_Level2) [3](#_Toc29840_WPSOffice_Level2)

**[二、 接入准备](#_Toc29840_WPSOffice_Level1)** **[3](#_Toc29840_WPSOffice_Level1)**

**[三、 使用说明](#_Toc21290_WPSOffice_Level1)** **[4](#_Toc21290_WPSOffice_Level1)**

[1. 快速入门](#_Toc21290_WPSOffice_Level2) [4](#_Toc21290_WPSOffice_Level2)

[2. 调试说明](#_Toc14641_WPSOffice_Level2) [4](#_Toc14641_WPSOffice_Level2)

**[四、 接口说明](#_Toc14641_WPSOffice_Level1)** **[5](#_Toc14641_WPSOffice_Level1)**

[1. Button事件](#_Toc4873_WPSOffice_Level2) [5](#_Toc4873_WPSOffice_Level2)

[2. Touch事件](#_Toc28005_WPSOffice_Level2) [6](#_Toc28005_WPSOffice_Level2)

[3. 震动事件](#_Toc17304_WPSOffice_Level2) [7](#_Toc17304_WPSOffice_Level2)

[4. 定位数据](#_Toc31319_WPSOffice_Level2) [7](#_Toc31319_WPSOffice_Level2)

[5. 提交错误信息](#_Toc11910_WPSOffice_Level2) [7](#_Toc11910_WPSOffice_Level2)

[6. 设备连接状态](#_Toc15204_WPSOffice_Level2) [8](#_Toc15204_WPSOffice_Level2)

**[五、注意事项](#_Toc28005_WPSOffice_Level1)** **[8](#_Toc28005_WPSOffice_Level1)**

[1.标定原点位置](#_Toc16510_WPSOffice_Level2) [8](#_Toc16510_WPSOffice_Level2)

[2.设置AppKey](#_Toc30274_WPSOffice_Level2) [8](#_Toc30274_WPSOffice_Level2)

[3.配置AndroidManifest.xml](#_Toc27950_WPSOffice_Level2) [8](#_Toc27950_WPSOffice_Level2)

[4.标定方向](#_Toc18805_WPSOffice_Level2) [9](#_Toc18805_WPSOffice_Level2)

[5.一键转身](#_Toc5229_WPSOffice_Level2) [9](#_Toc5229_WPSOffice_Level2)

# 简介

## 1.NOLO VR Unity SDK

NOLO VR Unity SDK是由NOLO Inc. 开发，方便Unity开发者开发移动端、6-DoF的VR游戏。

# 接入准备

开发者需要准备Unity5.6以上版本，NOLO VR Unity SDK。如果在安卓手机上调试需要先安装NOLO HOME，同时需要在NOLO开发者平台申请Appkey，并填写到你的Unity项目中。

Unity下载地址：

<https://unity3d.com/>

NOLO HOME下载地址：

<http://download.nolovr.com/download/nolohome.html>

调试阶段可以使用公用Appkey，正式上线的时候请修改为正式的Appkey，公用Appkey：4e4f4c4f484f4d457eff82725bc694a5。

# 使用说明

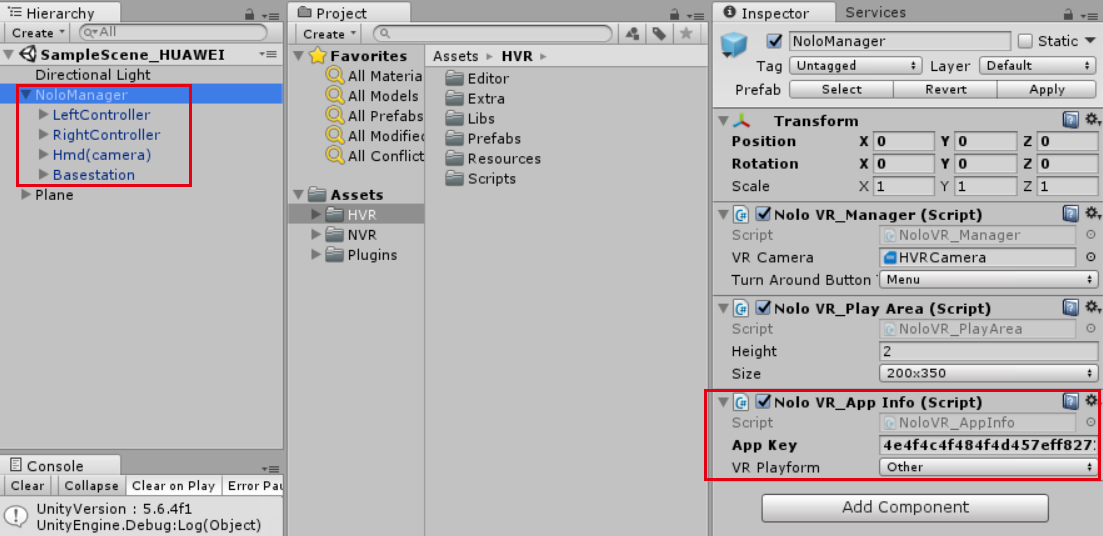
## 快速入门

#### （1）华为工程

1）新建一个Unity工程，将NOLO VR Unity SDK导入工程中。

2）新建一个场景，将NVR/Prefabs/NoloManager放入场景中。并保存。

3）将Appkey填写到如下位置



4) 填写正确的包名信息，即可打包到手机或一体机上运行。

## 2. 调试说明

Unity Editor中调试：请将且仅将NOLO头盔定位器用usb线连接至电脑，开启NOLO Assistant，其他NOLO设备打开，待NOLO Assitant上显示NOLO所有设备的电量信息后，点击Unity的Run按钮，即可在Unity Editor进行调试。

安卓端调试：请在移动设备上安装NOLO HOME，在项目中填写正确的Appkey，在未审核之前，可以使用测试key进行调试。同样用OTG线，将NOLO头盔定位器连接到手机或一体机上，如果提示“是否运行NOLO HOME访问USB设备”时，点击确定，即可在你的APP中获取到NOLO数据。

注意：目前只支持高通芯片的一体机，如爱奇艺奇遇Ⅱ，PICO G2。

# 接口说明

## Button事件

|  |  |
| --- | --- |
| **函数名称** | bool GetNoloButtonPressed() |
| **功能描述** | 获取按键长按状态 |
| **参数** | Enum NoloButtonID |
| **返回值** | bool |
| **先决条件** | NoloVR\_Controller.GetDevice() |

|  |  |
| --- | --- |
| **函数名称** | bool GetNoloButtonDown() |
| **功能描述** | 获取按键从抬起到按下状态 |
| **参数** | Enum NoloButtonID |
| **返回值** | bool |
| **先决条件** | NoloVR\_Controller.GetDevice() |

|  |  |
| --- | --- |
| **函数名称** | bool GetNoloButtonUp() |
| **功能描述** | 获取按键从按下到抬起状态 |
| **参数** | Enum NoloButtonID |
| **返回值** | bool |
| **先决条件** | NoloVR\_Controller.GetDevice() |

## Touch事件

|  |  |
| --- | --- |
| **函数名称** | bool GetNoloTouchPressed() |
| **功能描述** | 获取触摸板触摸状态 |
| **参数** | Enum NoloTouchID |
| **返回值** | bool |
| **先决条件** | NoloVR\_Controller.GetDevice() |

|  |  |
| --- | --- |
| **函数名称** | bool GetNoloTouchDown() |
| **功能描述** | 获取触摸板从非触摸状态到触摸状态 |
| **参数** | Enum NoloTouchID |
| **返回值** | bool |
| **先决条件** | NoloVR\_Controller.GetDevice() |

|  |  |
| --- | --- |
| **函数名称** | bool GetNoloTouchUp() |
| **功能描述** | 获取触摸板从触摸状态到非触摸状态 |
| **参数** | Enum NoloTouchID |
| **返回值** | bool |
| **先决条件** | NoloVR\_Controller.GetDevice() |

|  |  |
| --- | --- |
| **函数名称** | Vector2 GetAxis() |
| **功能描述** | 获取触摸板触摸点的坐标数据 |
| **参数** | Enum NoloTouchID:默认值为触摸板，其他参数无效。（见附录） |
| **返回值** | Vector2 |
| **先决条件** | NoloVR\_Controller.GetDevice() |

## 震动事件

|  |  |
| --- | --- |
| **函数名称** | void TriggerHapticPulse() |
| **功能描述** | 调用手柄震动 |
| **参数** | int:震动强度，范围0~100 |
| **返回值** | void |
| **先决条件** | NoloVR\_Controller.GetDevice() |

## 定位数据

|  |  |
| --- | --- |
| **函数名称** | Nolo\_Transform GetPose() |
| **功能描述** | 获取设备定位信息 |
| **参数** | Null |
| **返回值** | Nolo\_Transform |
| **先决条件** | NoloVR\_Controller.GetDevice() |

## 提交错误信息

|  |  |
| --- | --- |
| **函数名称** | void ReportError () |
| **功能描述** | 收集游戏错误信息 |
| **参数** | string |
| **返回值** | void |
| **先决条件** | NoloVR\_Playform.GetInstance() |

## 设备连接状态

NoloVR\_Plugins.API.GetPoseByDeviceType(0).bDeviceIsConnected

其中参数0代表头盔定位器，1代表左手柄，2代表右是吧，3代表基站，此方法仅在Android平台上有效。

# 五、注意事项

## 1.标定原点位置

NOLO正常运行时，将头盔定位器放置在地面上，按一下头盔定位器上的按钮，此时，这个位置就会成为游戏中的初始位置，即引擎中“NoloManager”所在的位置。标定原点操作是会被记录到设备中的，只有在基站位置发生很大变化时，重新执行一次标定操作即可。

## 2.设置AppKey

游戏必须设置APPKey，才能正常运行，AppKey是在NOLO开发者平台创建应用时自动生成的，请在游戏中添加NoloVR\_AppInfo脚本，并正确填写AppKey。

游戏未上传NOLO HOME时，可以使用此公用Appkey进行开发测试。

公用Appkey：4e4f4c4f484f4d457eff82725bc694a5。

## 3.配置AndroidManifest.xml

请在AndroidManifest.xml中配置如下权限：

<uses-permission android:name="android.permission.BROADCAST\_STICKY" />

<uses-permission android:name="nolo.permission.ACCESS\_SERVER" />

<uses-permission android:name="android.permission.PACKAGE\_USAGE\_STATS" />

<uses-permission android:name="android.permission.SYSTEM\_ALERT\_WINDOW" />

<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission android:name="com.android.launcher.permission.WRITE\_SETTINGS" />

<uses-permission android:name="android.permission.WRITE\_APN\_SETTINGS" />

请在Application中配置如下参数：

<meta-data android:name="com.picovr.type" android:value="vr"/>

请在Activity中配置如下参数：

<intent-filter>

<action android:name="android.intent.action.NOLOHOME" />

<category android:name="android.intent.category.DEFAULT" />

</intent-filter>

## 4.标定方向

游戏启动之后，如果游戏正方向不是基站所在方向，需要面对基站双击其中一个手柄的电源键来标定正方向，同样，如果手柄的方向不对，也请面对基站，双击这个手柄的电源键标定正方向。

## 5.一键转身

修改下图值，可以将NOLO CV1某个按键设置为转身热键，可以在双击这个按键的情况下，实现原地旋转180度的操作，将游戏中背后的场景，转到身前，如下。

