

# NOLO VR Unity SDK

## 接口说明

北京凌宇智控科技有限公司

2020 年 9 月

## 目录

一、 简介.....	3
1.NOLO VR Unity SDK.....	3
二、 接入准备.....	3
三、 使用说明.....	4
1. 快速入门.....	4
2. 调试说明.....	6
四、 接口说明.....	7
1. Button 事件.....	7
2. Touch 事件.....	8
3. 震动事件.....	9
4. 定位数据.....	9
5. 提交错误信息.....	10
6. 设备连接状态.....	10
7. 设备电量信息.....	10
五、 注意事项.....	11
1.标定原点位置.....	11
2.设置 AppKey.....	11
3.标定方向.....	11
4.射线检测.....	12

# 一、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。

## 二、接入准备

软件需求：Unity5.6 以上版本

NOLO HOME PC 版本下载地址：

[http://download.nolovr.com/download/NOLO\\_home\\_PC.html](http://download.nolovr.com/download/NOLO_home_PC.html)

请联系 [dev@nolovr.com](mailto:dev@nolovr.com) 获取 Appkey，并填写到你的 Unity 项目中，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。

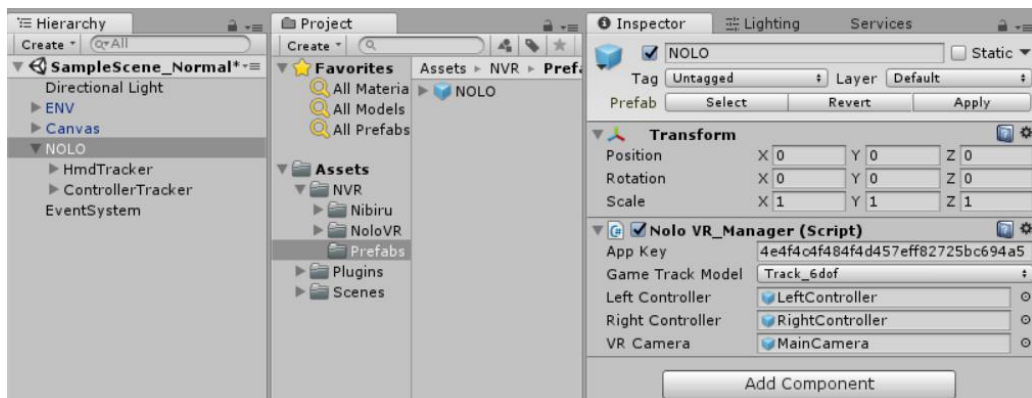
硬件需求：NOLO X1 一体机

## 三、使用说明

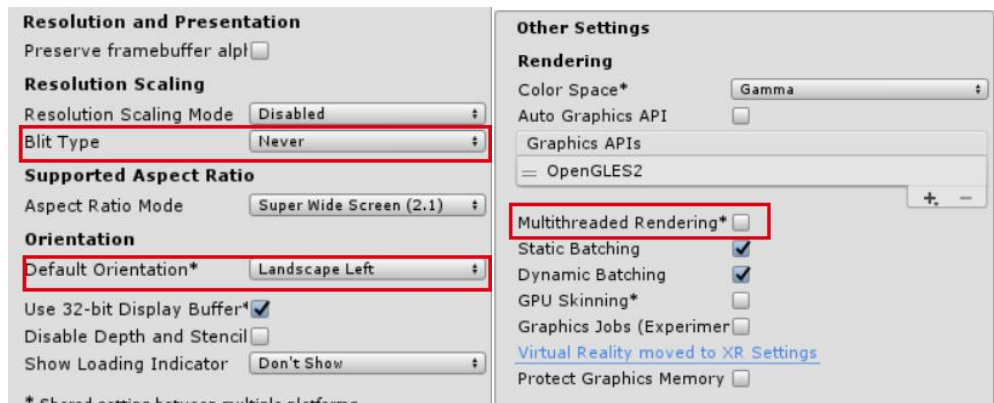
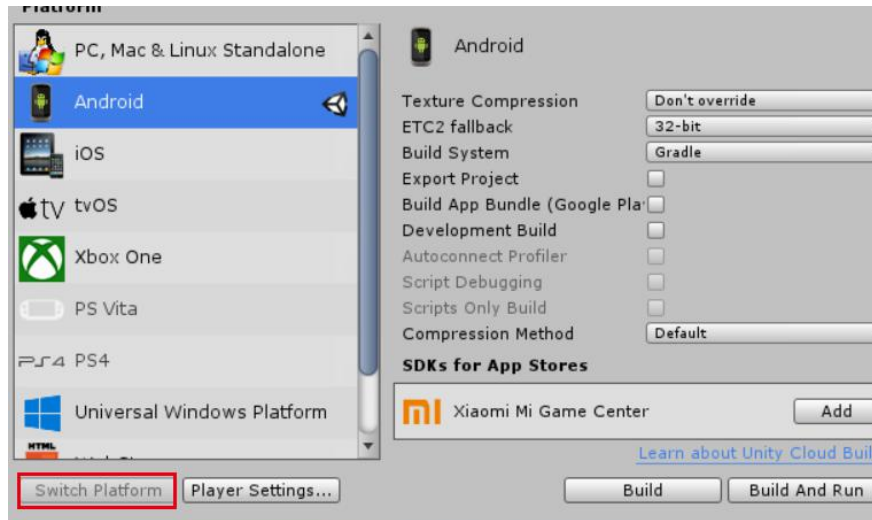
### 1. 快速入门

#### (1) 一体机工程

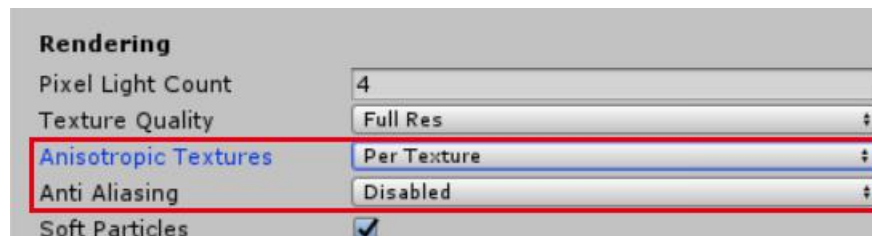
- 1) 新建一个 Unity 工程，将 NOLO VR Unity SDK 导入工程中。
- 2) 新建一个场景，将 NVR/Prefabs/NOLO 放入场景中，并保存。
- 3) 将 Appkey 填写到如下位置

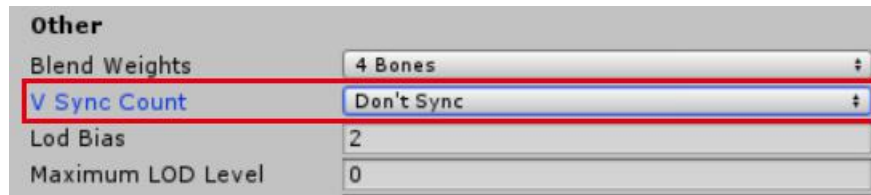


- 4) Player Settings 设置: 'Resolution and Presentation'中, orientation 必须设置为 'Landscape Left', 'Other Settings' 中, 'Multithreaded Rendering'必须设置为不可用。



5) Quality Settings 设置：'Rendering' 中，'Anisotropic Textures' 设置为 'Per Texture'，'Anti Aliasing' 设置为'Disabled'。'Other' 中 'Sync Count' 设置为'Don't Sync'。

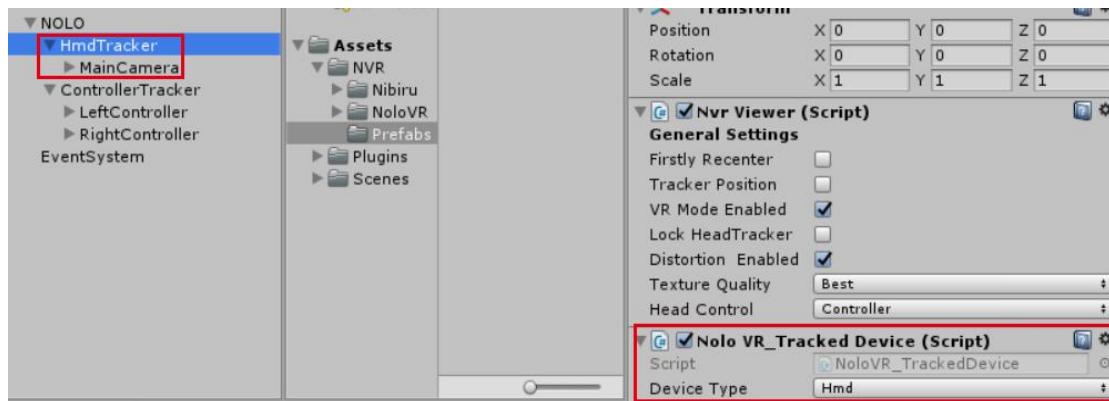




6) 填写正确的包名信息，即可打包到 NOLO X1 一体机上运行。

## 2. NOLO Prefabs

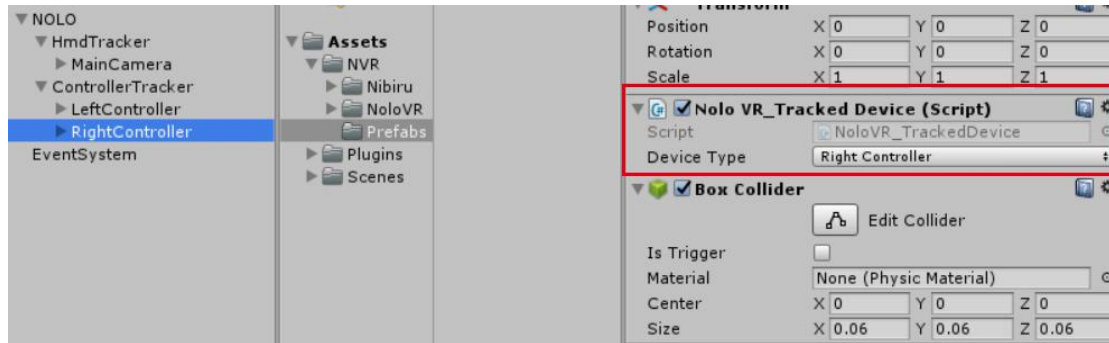
1. 头部组件，包含头部定位信息，摄像机信息



2. 左手组件，包含左手定位信息



3. 右手组件，包含右手定位信息



### 3. 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 HOME PC 版本, 其他 NOLO 设备打开, 待 NOLO HOME 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。

## 四、接口说明

### 1.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

<b>先决条件</b>	NoloVR_Controller.GetDevice()
-------------	-------------------------------

<b>函数名称</b>	bool GetNoloButtonDown()
<b>功能描述</b>	获取按键从抬起到按下状态
<b>参数</b>	Enum NoloButtonID
<b>返回值</b>	bool
<b>先决条件</b>	NoloVR_Controller.GetDevice()

<b>函数名称</b>	bool GetNoloButtonUp()
<b>功能描述</b>	获取按键从按下到抬起状态
<b>参数</b>	Enum NoloButtonID
<b>返回值</b>	bool
<b>先决条件</b>	NoloVR_Controller.GetDevice()

## 2.Touch 事件

<b>函数名称</b>	bool GetNoloTouchPressed()
<b>功能描述</b>	获取触摸板触摸状态
<b>参数</b>	Enum NoloTouchID
<b>返回值</b>	bool
<b>先决条件</b>	NoloVR_Controller.GetDevice()

<b>函数名称</b>	bool GetNoloTouchDown()
<b>功能描述</b>	获取触摸板从非触摸状态到触摸状态
<b>参数</b>	Enum NoloTouchID
<b>返回值</b>	bool
<b>先决条件</b>	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()

### 3. 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()

### 4. 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()

## 5. 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()

## 6. 设备连接状态

函数名称	bool GetNoloConnectStatus()
功能描述	获取 NOLO 设备连接状态信息
参数	int/NoloDeviceType
返回值	bool
先决条件	NoloVR_Plugins.GetNoloConnectStatus()

## 7. 设备电量信息

函数名称	int GetElectricity()
功能描述	获取 NOLO 设备电量信息
参数	int/NoloDeviceType
返回值	int, 范围 (0~5)
先决条件	NoloVR_Plugins.GetElectricity()

## 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“NOLO”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\_Manager 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。

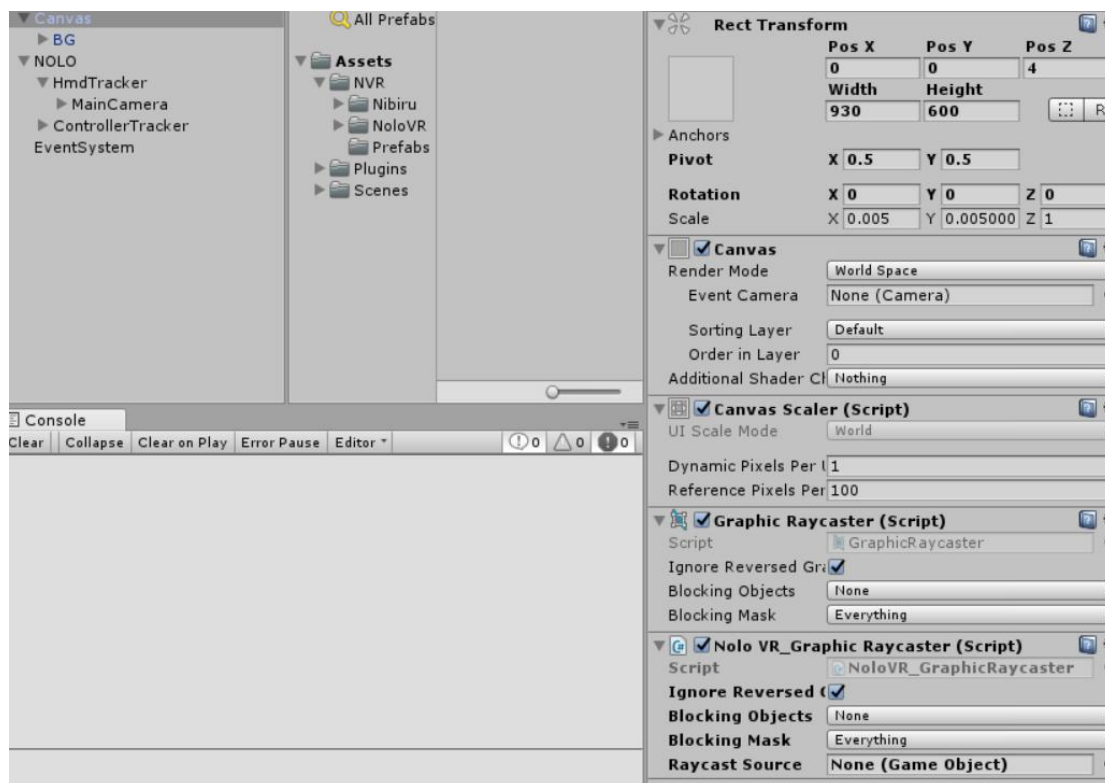
### 3. 标定方向

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。

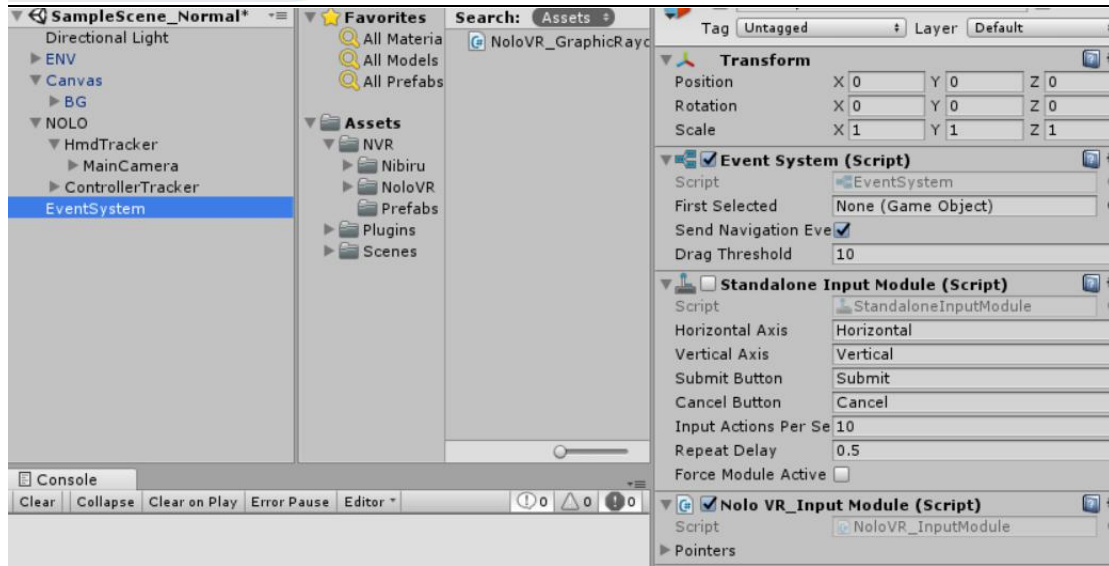
## 4.射线检测

NOLO 提供一套可以 UGUI 射线触发方案:

1. 在 UGUI 的 Canvas 上面添加 NoloVR\_GraphicRaycaster.cs 脚本



2. 在 UGUI 上的 EventSystem 上面添加 NoloVR\_InputModule.cs 脚本



3. 创建一个 GameObject 并添加 NoloVR\_SimplePointer.cs 脚本，通过扳机按键进行切换手柄射线

