**NOLO VR UE4 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 3](#_Toc1907)**

[1.关于NOLO 3](#_Toc15666)

[2. 关于NOLO CV1 3](#_Toc14732)

[3.关于NOLO HOME 3](#_Toc16439)

[4.关于NOLO VR UE4 SDK 4](#_Toc9907)

[5.关于移动端SDK架构 4](#_Toc1372)

**[二、 接入准备 5](#_Toc20024)**

**[三、 接口说明 6](#_Toc666)**

[1. 蓝图接口总览 6](#_Toc32324)

[2. 接口详细说明 7](#_Toc12232)

[3. 手柄按键说明 10](#_Toc8419)

**[四、注意事项 13](#_Toc8975)**

[1.标定原点位置 13](#_Toc10440)

[2.设置AppKey 13](#_Toc3283)

[3.标定方向 14](#_Toc29503)

[4.UE4 摄像机设置 14](#_Toc29128)

[5.搭配第三方VR SDK 14](#_Toc25716)

# 简介

## 1.关于NOLO

NOLO 致力于将桌面级 VR 的优秀沉浸式体验与移动 VR 的便捷性之间架起一座桥梁，重新定义下一代移动 VR 的交互方式。

目前，NOLO 可兼容全球将近 8700 万市场存量的各类移动 VR 头盔，市场潜力巨大。不仅如此，NOLO 也已大规模展开与海内外众多 VR 一体机、PC VR、AR、服务机器人、无人机公司的合作。

## 关于NOLO CV1

NOLO CV1 是 NOLO Inc. 自主研发的全球首款全沉浸式 VR/AR 交互产品，包含 1 个定位基站、1 个头盔定位器及 2 个交互手柄。

NOLO CV1 能适配目前市面上主流的移动 VR 头盔、部分PC VR 和 AR 头盔，以及基于NibiruOS的一体机，为其提供空间定位与交互功能。用户首次使用时，仅需简单设置，即可实现在虚拟世界中真实的移动，以及通过交互手柄与虚拟世界中的事物进行多元化的互动。

## 3.关于NOLO HOME

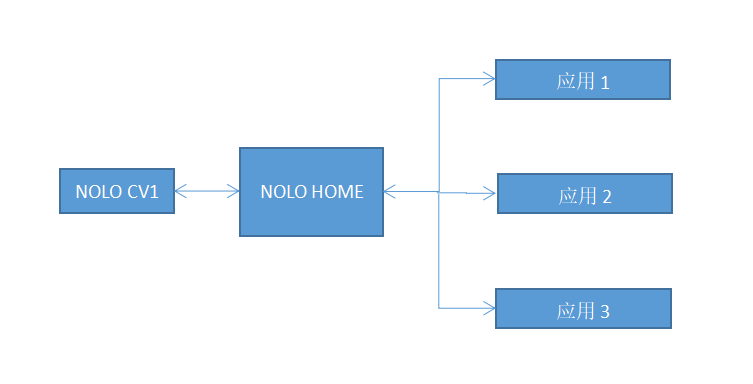
NOLO HOME 是 NOLO 品牌旗下的移动 VR 生态平台，也是全球首个基于手机的 6-DoF 移动 VR 应用平台。NOLO HOME + NOLO CV1 + 手机，用户可以随时随地、无线便捷得享受到全沉浸式 VR 体验，从而为普通消费者创造一种全新的娱乐游戏方式。目前，Android 用户率先可以下载安装，iOS 版将后续开放。

## 4.关于NOLO VR UE4 SDK

NOLO VR UE4 SDK是由NOLO Inc. 开发，便于UE4 开发者获取NOLO设备数据，搭配Cardboard和GearVR 的SDK，开发移动端6-DoF VR游戏。

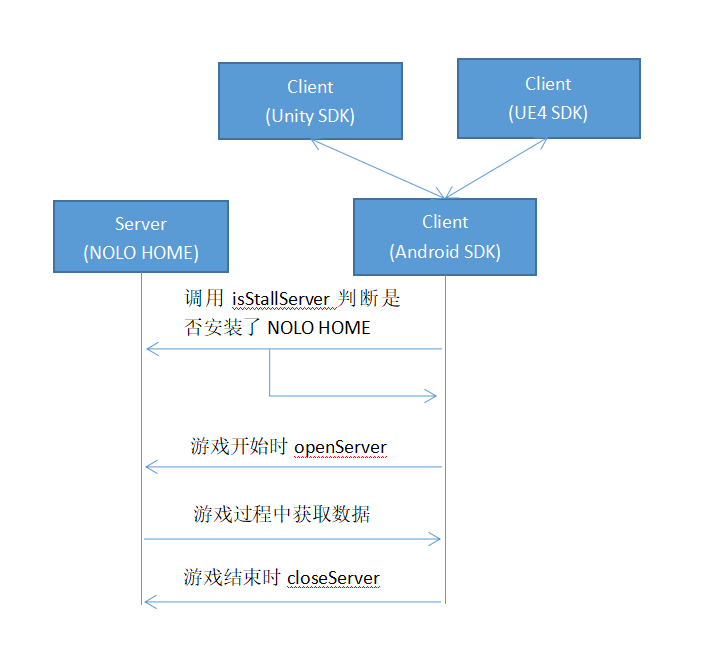
## 5.关于移动端SDK架构

NOLO SDK采用C/S架构设计，NOLO HOME为Server端，负责读取NOLO CV1设备数据，并进行处理。游戏为Client端，SDK通过AIDL将游戏与NOLO HOME建立通信，获取NOLO CV1设备数据。游戏本身不需要去读取USB设备数据，只需要与NOLO HOME建立连接来实现数据收发，如（图一）。



（图一）

NOLO VR Android SDK通过AIDL与NOLO HOME建立进程间通信，来获取设备数据。首先要通过getInstance()来获取NOLOVR的实例，游戏启动前调用isStallServer()判断用户是否安装了NOLO HOME，确认安装之后，调用openServer()绑定Server，在游戏结束后调用closeServer()结束绑定。NOLO VR Unity SDK和NOLO VR UE4 SDK是对NOLO VR Android SDK进行部分接口封装，开发者只需要通过对应接口获取定位和按键数据进行游戏开发即可，如（图二）。



（图二）

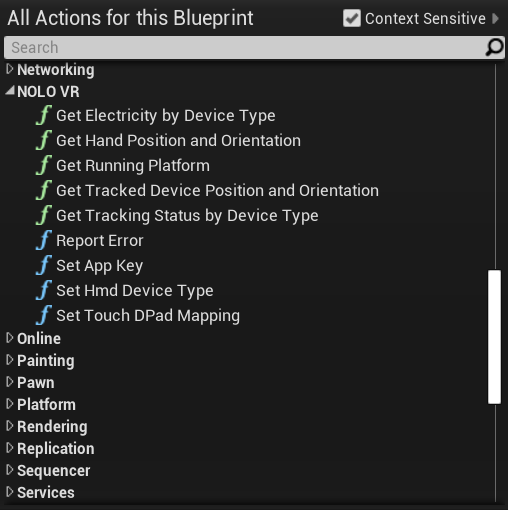
# 接入准备

开发和调试需要准备：UnrealEngine4.18.2/4.18.3，手机上运行需要先安装NOLO HOME。

# 接口说明

## 蓝图接口总览

如（图三）。



（图三）

1. GetRunningPlatform：获取游戏运行平台。
2. GetTrackedDevicePositionAndOrientation：获取头部定位器的位置和姿态信息。
3. GetHandPositionAndOrientation：获取手柄控制器的位置和姿态信息。
4. GetTrackingStatusByDeviceType：获取设备的跟踪状态。
5. GetElectricityByDeviceType：获取设备的电量信息。
6. SetHmdDeviceType：设置游戏开发所使用的头盔类型。
7. SetAppKey：设置APPKey。
8. ReportError：捕获异常信息。
9. SetTouchDPadMapping：定义触摸板上下左右四个按键关联事件。

## 2. 接口详细说明

|  |  |
| --- | --- |
| **函数名称** | **FString GetRunningPlatform（）** |
| **功能描述** | 获取游戏运行平台 |
| **参数** | **NULL** |
| **返回值** | **FString : Android or Default** |
| **先决条件** |  |

|  |  |
| --- | --- |
| **函数名称** | **bool GetTrackedDevicePositionAndOrientation（FVector & OutPosition, FRotator & OutOrientation）** |
| **功能描述** | 获取头部定位器的位置和姿态信息 |
| **参数** | **OutPosition : (out) Current position of the device**  **OutOrientation (out) Current orientation of the device** |
| **返回值** | **Bool : True if the specified device id had a valid tracking pose this frame, false otherwise** |
| **先决条件** |  |

|  |  |
| --- | --- |
| **函数名称** | **Bool GetHandPositionAndOrientation（EControllerHand Hand, FVector& OutPosition, FRotator& OutOrientation）** |
| **功能描述** | 获取手柄控制器的位置和姿态信息 |
| **参数** | **Hand : Which hand's controller to get the position and orientation for**  **OutPosition : (out) Current position of the device**  **OutOrientation (out) Current orientation of the device** |
| **返回值** | **Bool : True if the specified controller is Available** |
| **先决条件** |  |

|  |  |
| --- | --- |
| **函数名称** | **bool GetTrackingStatusByDeviceType(const ENoloDeviceType DeviceType)** |
| **功能描述** | 获取设备的跟踪状态 |
| **参数** | **ENoloDeviceType : Hmd,Left,Right,Basestation** |
| **返回值** | **True : Tracked ,false : not Tracked** |
| **先决条件** |  |

|  |  |
| --- | --- |
| **函数名称** | **Bool bool GetElectricityByDeviceType(const ENoloDeviceType DeviceType,int & battery)** |
| **功能描述** | 根据设备类型获取设备电量 |
| **参数** | **ENoloDeviceType : Hmd,Left,Right,Basestation**  **Battery : (out)0:shut down 、1:low 、2:middle 、3:high** |
| **返回值** | **Bool : True if the battery info is Available** |
| **先决条件** |  |

|  |  |
| --- | --- |
| **函数名称** | **void SetHmdDeviceType(const EMobileVR\_HMDDeviceType DeviceType)** |
| **功能描述** | 设置游戏开发所使用的头盔类型 |
| **参数** | **EMobileVR\_HMDDeviceType ：DT\_GearVR、 DT\_GoogleVR\_Cardboard、DT\_GoogleVR\_Daydream、DT\_Other** |
| **返回值** | **Void** |
| **先决条件** |  |

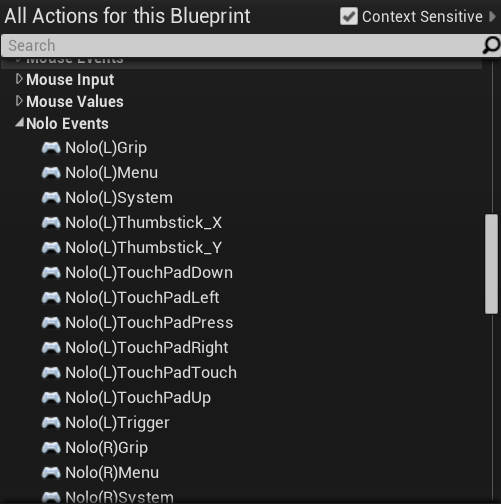
|  |  |
| --- | --- |
| **函数名称** | **void SetAppKey (FString appKey)** |
| **功能描述** | 设置AppKey，决定app是否有权限获取定位数据 |
| **参数** | **FString** |
| **返回值** | **Void** |
| **先决条件** |  |

|  |  |
| --- | --- |
| **函数名称** | **void ReportError (FString appKey)** |
| **功能描述** | 捕获异常信息 |
| **参数** | **FString** |
| **返回值** | **Void** |
| **先决条件** |  |

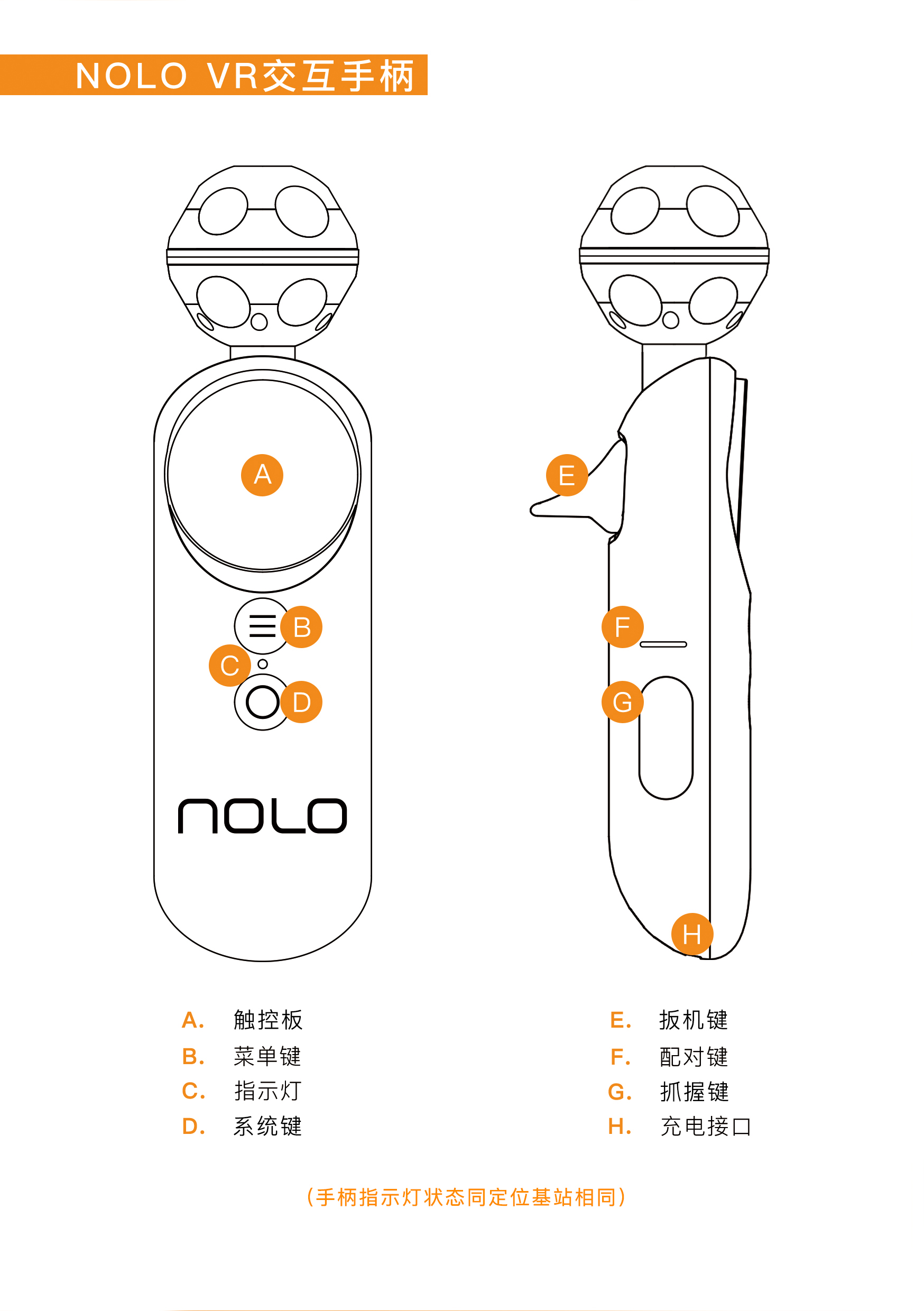
|  |  |
| --- | --- |
| **函数名称** | **void SetTouchDPadMapping (ENoloVRTouchDPadMapping NewMapping)** |
| **功能描述** | **定义触摸板上下左右四个按键关联事件** |
| **参数** | **ENoloVRTouchDPadMapping** |
| **返回值** | **Void** |
| **先决条件** |  |

## 手柄按键说明

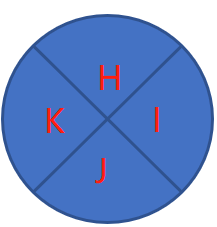
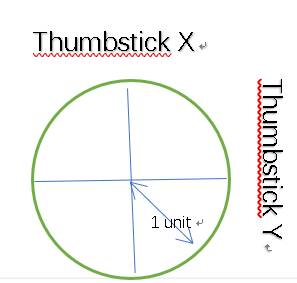
如（图四）、（图五）、（图六）。



（图四）



（图五）



（图六）

按键对照表如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **图例按键** | **描述** | **NOLO Events** | **Gamepad Events** | **SteamEvents** |
| A | TouchPadPress | Nolo\_Left\_TouchPadPress | MotionController\_Left\_Thumbstick |  |
| A | TouchPadTouch | Nolo\_Left\_TouchPadTouch |  | Steam\_Touch\_0 |
| B | Menu | Nolo\_Left\_Menu | MotionController\_Left\_Shoulder |  |
| D | System | Nolo\_Left\_System | SpecialLeft |  |
| E: | Trigger | Nolo\_Left\_Trigger | MotionController\_Left\_Trigger |  |
| G | Grip | Nolo\_Left\_Grip | MotionController\_Left\_Grip1 |  |

触摸板按键值对照表如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **图例按键** | **描述** | **NOLO Events** | **Face Buttos**  **(Gamepad Events)** | **ThumbstickDirections**  **(Gamepad Events)** |
| H | Up | Nolo\_Left\_TouchPadUp | MotionController\_Left\_FaceButton1 | MotionController\_Left\_Thumbstick\_Up |
| I | Right | Nolo\_Left\_TouchPadRight | MotionController\_Left\_FaceButton2 | MotionController\_Left\_Thumbstick\_Right |
| J | Down | Nolo\_Left\_TouchPadDown | MotionController\_Left\_FaceButton3 | MotionController\_Left\_Thumbstick\_Down |
| K | Left | Nolo\_Left\_TouchPadLeft | MotionController\_Left\_FaceButton4 | MotionController\_Left\_Thumbstick\_Left |

# 四、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。标定原点操作是会被记录到设备中的，当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。

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.UE4 摄像机设置

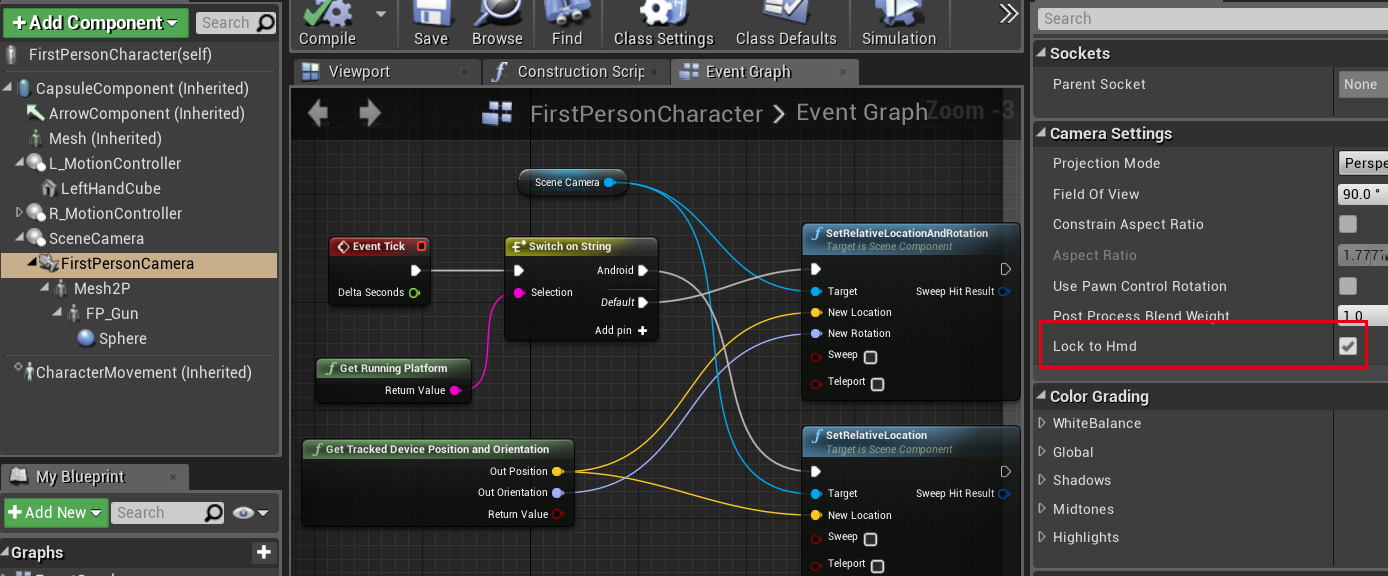
NOLO VR提供了6DOF的定位数据，但是对于NOLO VR插件并没有接入UE4 HMD接口（IHeadMountedDisplay），

方案一：当您选择只使用NOLO VR头部定位器的位置信息时，可以按下图设计，即改变摄像机的父类节点的位置信息。即使用手机IMU控制摄像机姿态（Rotation）和NoloVR的位置（position）信息进行定位。

方案二：当您选择使用NOLO VR头部定位器的位置和姿态数据时，请不要勾选Lock to Hmd,然后使用GetTrackedDevicePositionAndOrientation蓝图接口控制摄像机的position and rotation 。

在此：我们建议您使用方案一。

注意：该部分与原中文版相比改动较大，英文版已对应翻译，日文版需做相应改动。



（图七）

## 5.搭配第三方VR SDK

NOLO VR插件完全实现UE4引擎的手柄控制接口，在头部定位跟踪这块提供了蓝图接口，获取定位数据，所以完全可以配合Google VR、Gear VR进行游戏开发，解决移动端只有3DOF数据问题。