

# Null Touch 交互套件

## Unity 插件开发手册

V2.0



广州幻境科技有限公司



修订记录

日期	修订版本	描述
2018/1/22	V1.0	起草
2018/1/29	V1.1	编写
2018/2/1	V1.2	修改 UI
2018/11/6	V1.5	修复手朝向
2019/2/14	V2.0	修改添加手势、震动、Tracker 等功能



## 目录

Null Touch 交互套件 .....	
Unity 插件开发手册.....	
目录.....	
1. 基本使用和功能测试 .....	
1.1. 链接设备 .....	
1.2. 确认设备功能正常 .....	
1.3. 使用 Null Touch 校准程序 .....	
1.4. 测试与 VIVE 定位器配合使用.....	
2. 示范场景(与 VIVE 定位器配合使用).....	
2.1. 手.....	
2.2. 手+校准 .....	
2.3. 手+震动 .....	
2.4. 手+手势 .....	
2.5. 手+虚拟现实+校准 .....	
2.6. 手+手势+虚拟现实+校准+震动.....	
3. 开发说明 .....	
3.1.1 校准.....	
3.1.2 震动.....	
3.1.3 手势.....	
3.1.4 虚拟现实.....	
3.1.5 抓取.....	



- 4. 自定义场景.....
- 5. 报错说明.....



## 1. 基本使用和功能测试

### 1.1. 链接设备

安装好 Deamon 软件(官网提供下载 <http://huantek.com>), 插上接收器后

把配对的蓝牙接收器插到电脑设备 (蓝牙会闪烁)



把 NullTouch 手套电源打开 (绿灯为正常, 红灯为电量不足)



## 1.2. 确认设备功能正常

观看手背 LED 灯（**闪烁时表示正在连接；当不闪烁时表示已近成功连接**）



注：




- 1.硬件在成功连接过设备，以后会将设备作为默认连接设备自动进行连接，无需重复配对设备操作；
- 2.当在开发过程中发现有手指无法弯曲时，请插拔蓝牙接收器重新连接

### 1.3. 使用 Null Touch 校准程序

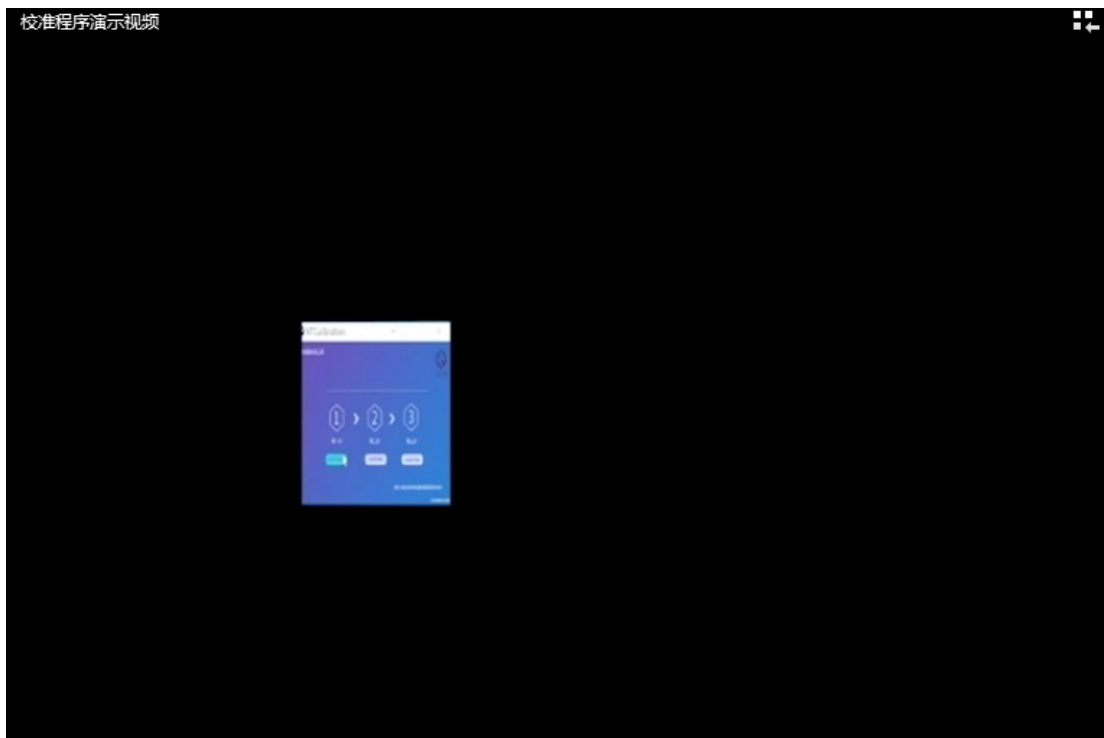
到幻境官网下载校准程序(只有在校准完成后，手套才能正常使用)

下载地址：<http://www.huantek.com/h-col-127.html>

下载后解压：

 NT_Calibration	2017/8/31 星期...	文件夹	
 安装说明.txt	2017/8/8 星期二 ...	文本文档	1 KB
 校准程序演示视频.avi	2017/8/9 星期三 ...	视频剪辑	110,832 KB

使用教程请参考“校准程序演示视频”



第一步：重复张手握拳动作，获取手部弯曲度系数（过程中右手套震动）



第二步：保持握拳动作，校验手部弯曲度最大值（过程中右手套震动）



第三步：保持张手动作，校验手部弯曲度最小值（过程中右手套震动）





校准失败提示:



如果校准失败即重复以上动作, 如果每次都出现校准不通过的情况, 尝试将蓝牙接收器取出然后重新接入, 进行设备重连。再进行校准。最后提示校准成功后, 此时可开始使用插件进行开发。

注:

1. 必须提前插入蓝牙接收设备
2. 此程序在第一次使用以及开发 NullTouch 前必须校准一次
3. 在后续开发中需要校准调整也是使用此程序(例如手指无法弯曲等)



## 1.4.测试与 VIVE 定位器配合使用

首先，你得有两个 VIVE 定位器，使用前检查设备固件是否需要更新，尽可能保持最新版本



注：VIVE 定位器的朝向与上图显示方向保持一致

## 2. 示范场景

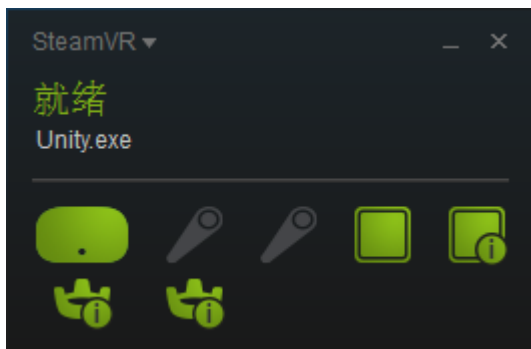
首先配对设备（详见文档目录），配对设备成功

此 VR 示例场景默认使用 HTC VIVE 设备进行演示(请用户提前做好设备并按官方提示连接好设备)。

特别注意：在 Steam 窗口中查看定位器图标是否有闪烁，尽可能避免出现闪烁的情况因为有可能丢失定位信息；有提示固件需要更新请按官方教程跟新尽



可能保持固件为最新版本



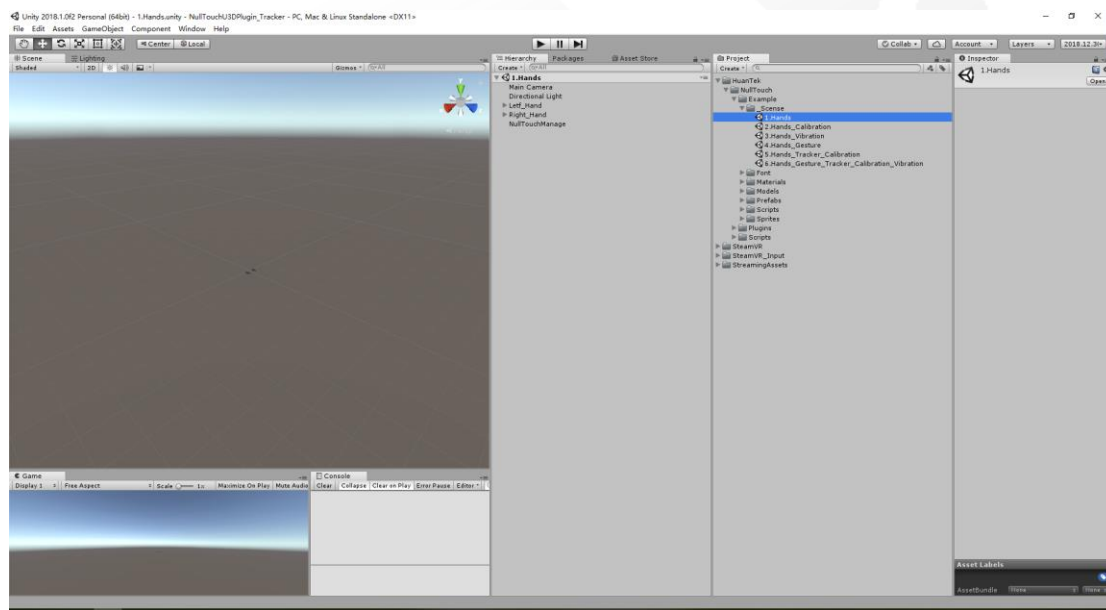
(上图为需要升级的状态)

## 2.1. 只有双手的示范场景

插件请使用 **Unity2018.1.0** 以上版本导入

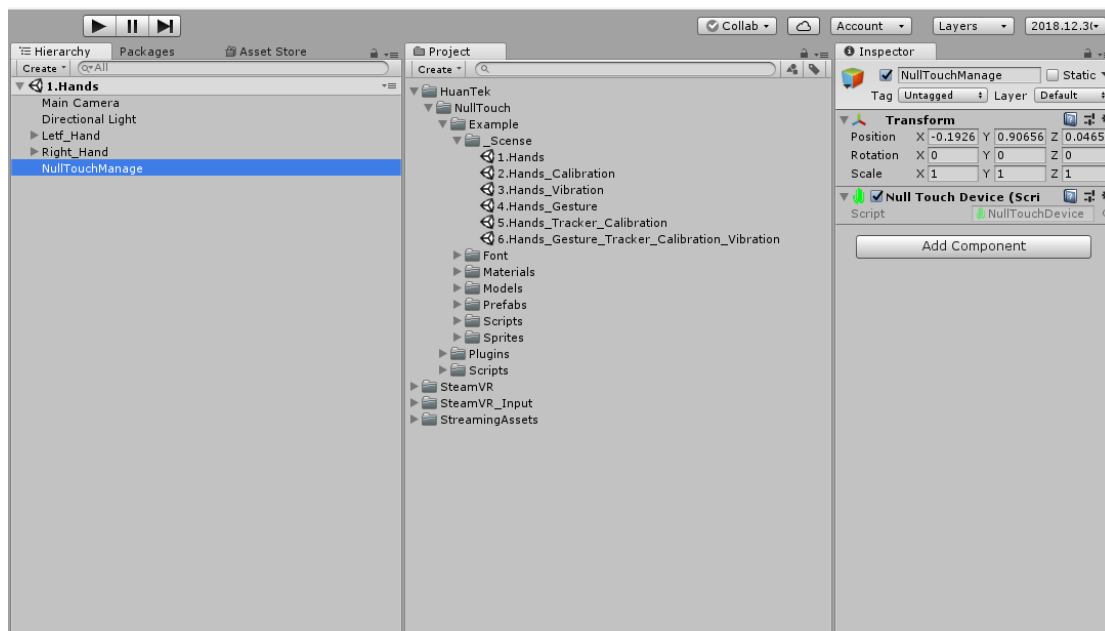
导入 NullTouch Unity 插件后, 打开 HuanTek>Examples>\_Scenes

>1.Hands

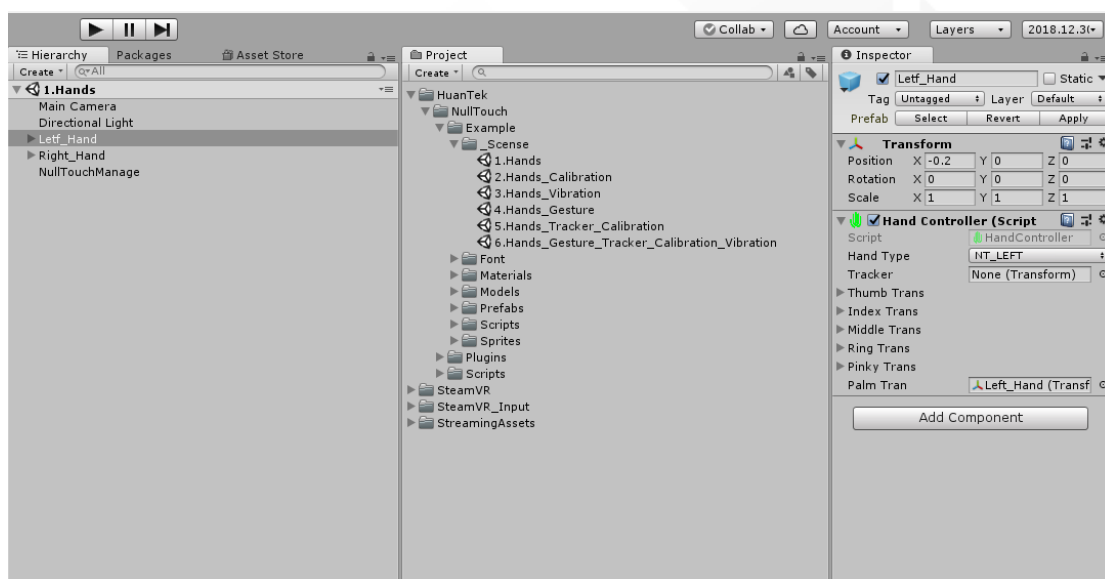


场景中包含核心的几个脚本容器对象分别是

1). NullTouchEvent.cs (设备数据和状态)



## 2). HandController.cs (手部对象数据)



注：当用户需要自定义手部模型的时候，一定要注意手模型的骨骼需要与本插件提供的手模型骨骼保持一致；此脚本通过静态赋值获取手模型的骨骼数据。

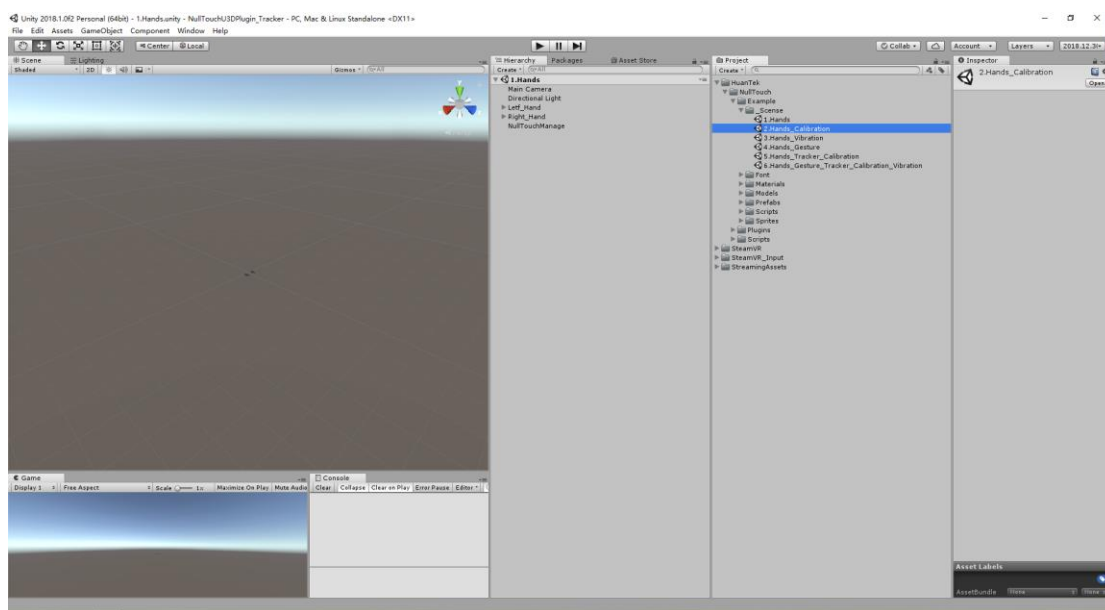
## 2.2. 有双手并且可以手动点击按钮进行校准的示范场景

注：未校准与校准后的区别，未校准手腕位置不可旋转，校准后手腕位置可以旋转



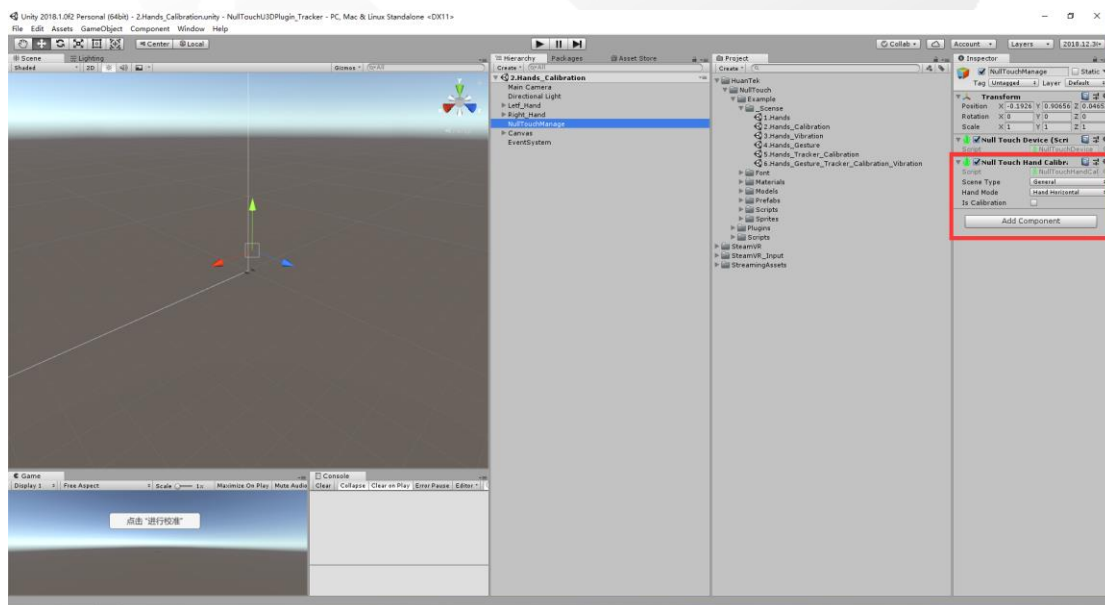
导入 NullTouch Unity 插件后, 打开 HuanTek>Examples>\_Scenes

## >2.Hands\_Calibration



场景中包含核心的几个脚本容器对象分别是

1). NullTouchHandCalibrationController.cs (针对不同的应用场景, 采用不同的双手校准方式)

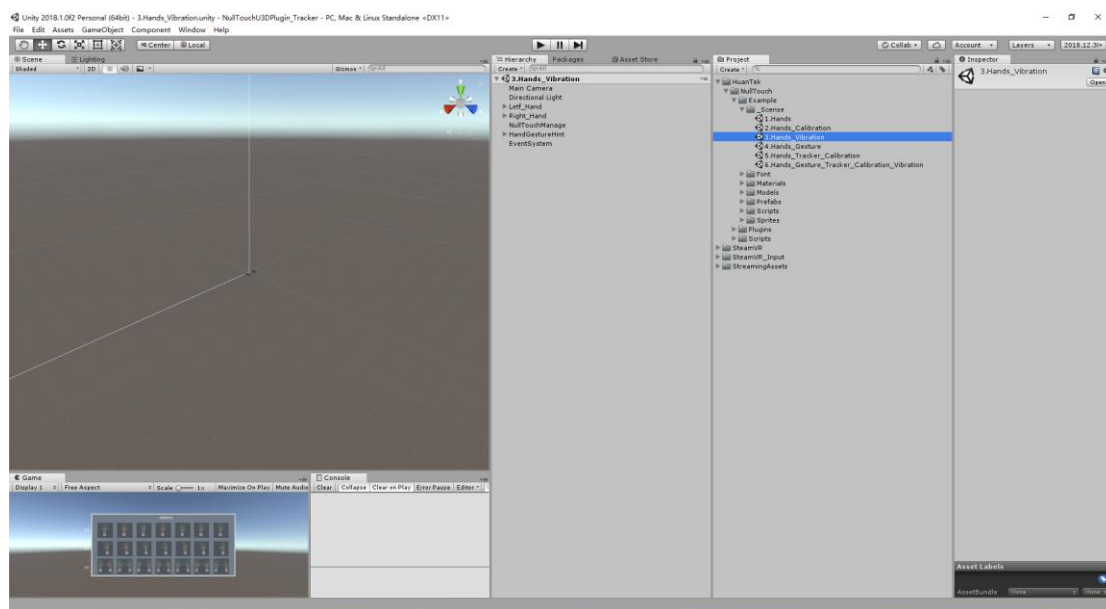




## 2.3. 有双手并且可以查看手套震动功能的示范场景

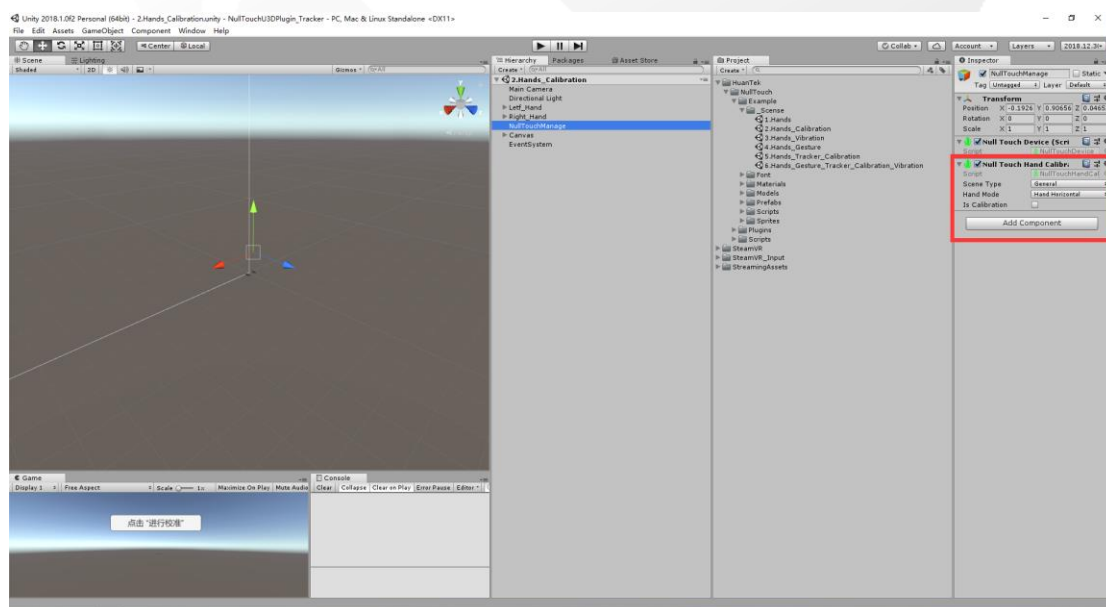
导入 NullTouch Unity 插件后，打开 HuanTek>Examples>\_Scenes

### >3.Hands\_Vibration



场景中包含核心的几个脚本容器对象分别是

#### 1). NullTouchAllVibration.cs (可获取双手震动事件脚本)



点击 Play 逐一查看

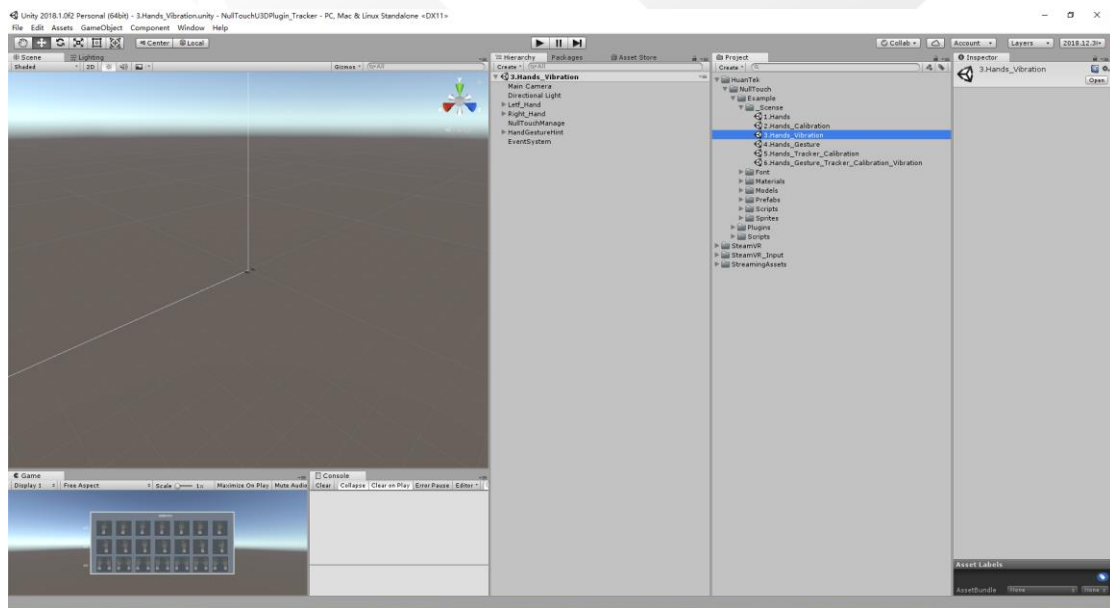




## 2.4. 有双手并且可以查看手套手势功能的示范场景（查看 NullTouch 可支持手势）

导入 NullTouch Unity 插件后，打开 HuanTek>Examples>\_Scenes

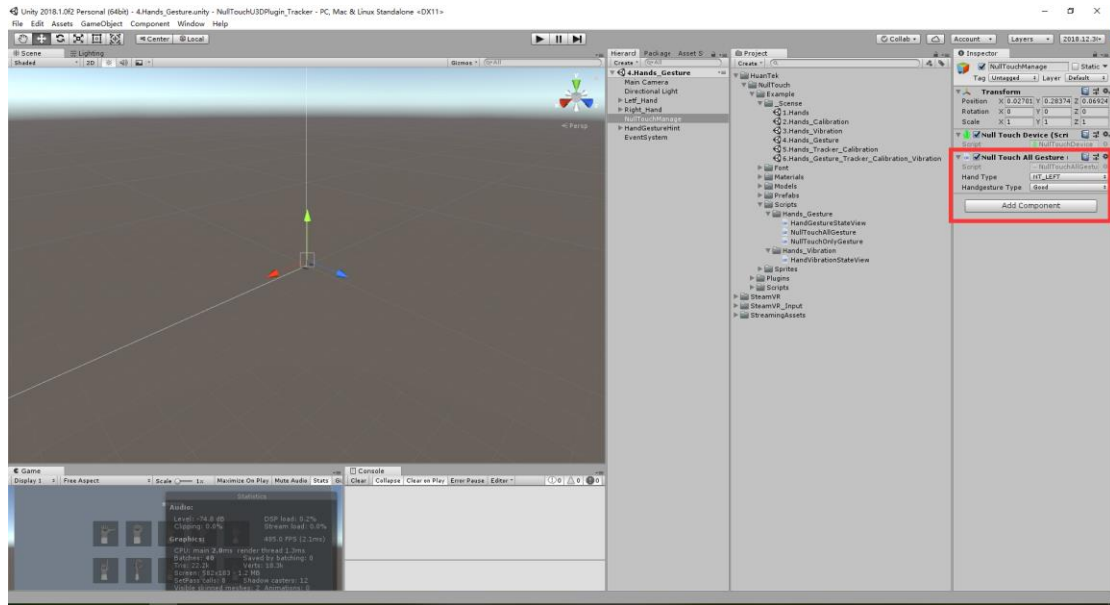
>4.Hands\_Gesture



场景中包含核心的几个脚本容器对象分别是



## 1). NullTouchAllGesture.cs (可获取双手手势事件脚本, 合适简单功能项目)

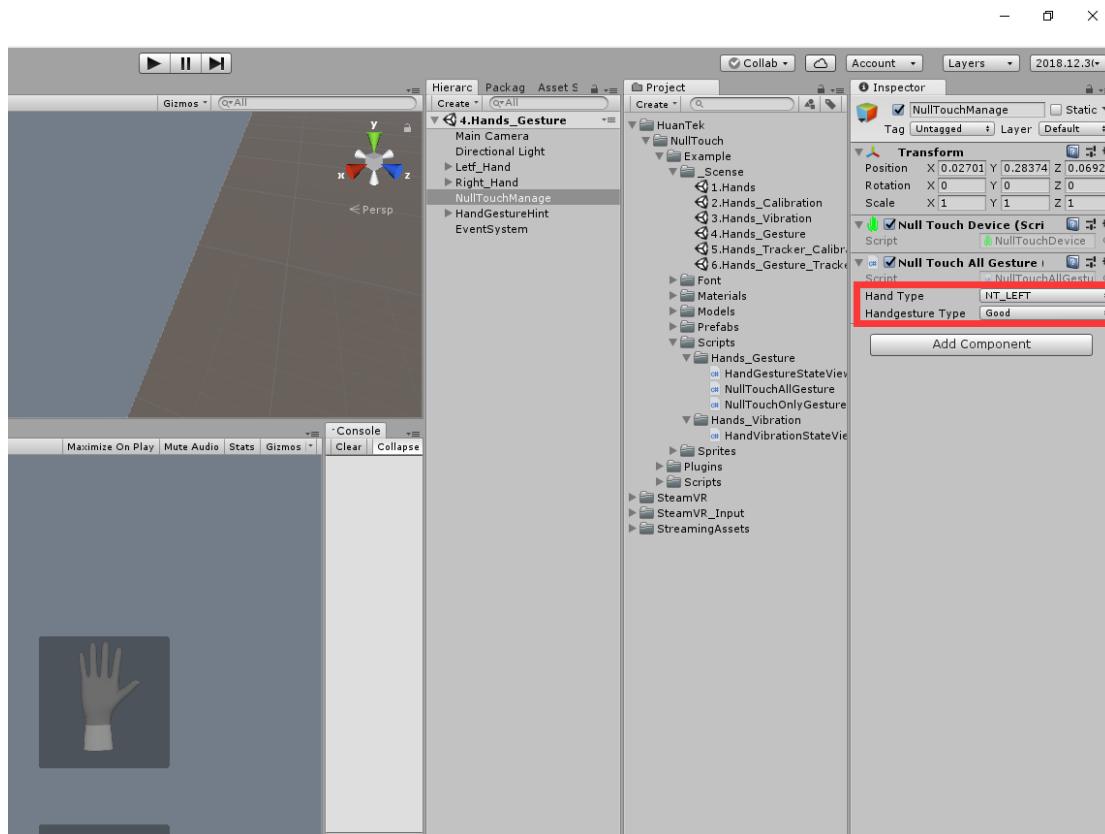


点击 Play 逐一查看



运行过程可动态设置查看不同的手、不同的手势

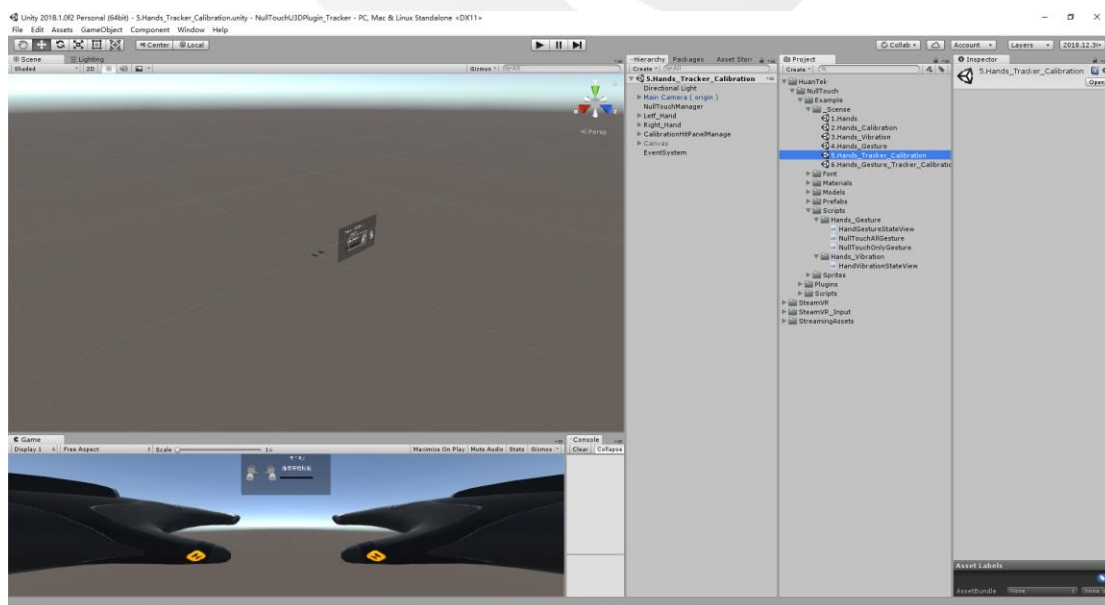




## 2.5. 有双手并且可以在虚拟现实环境中校准手套功能的示范场景

导入 NullTouch Unity 插件后，打开 HuanTek>Examples>\_Scenes

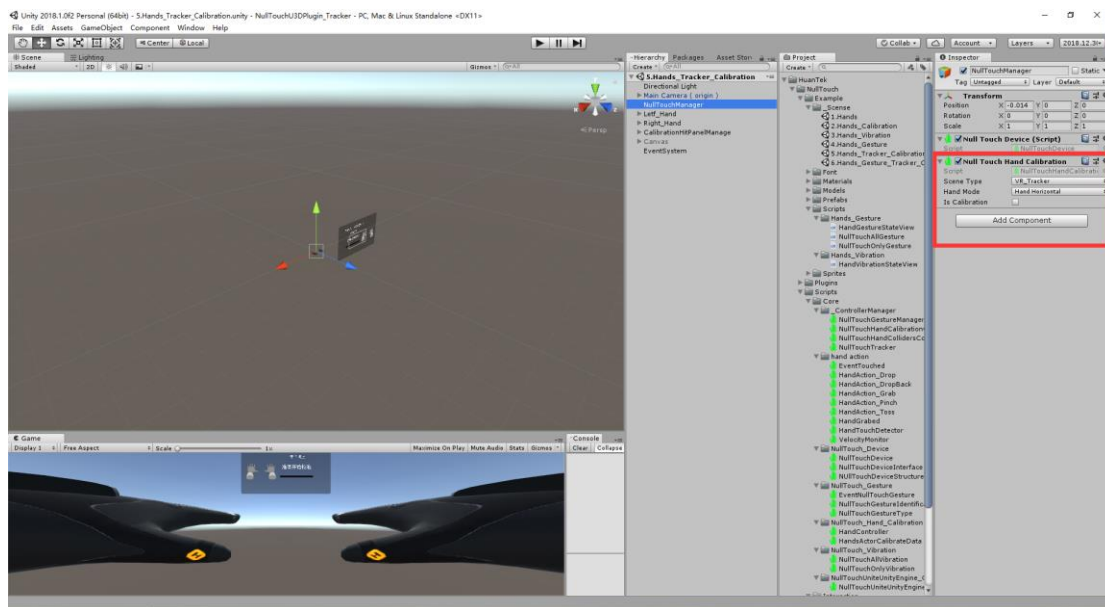
> 5.Hands\_Tracker\_Calibration





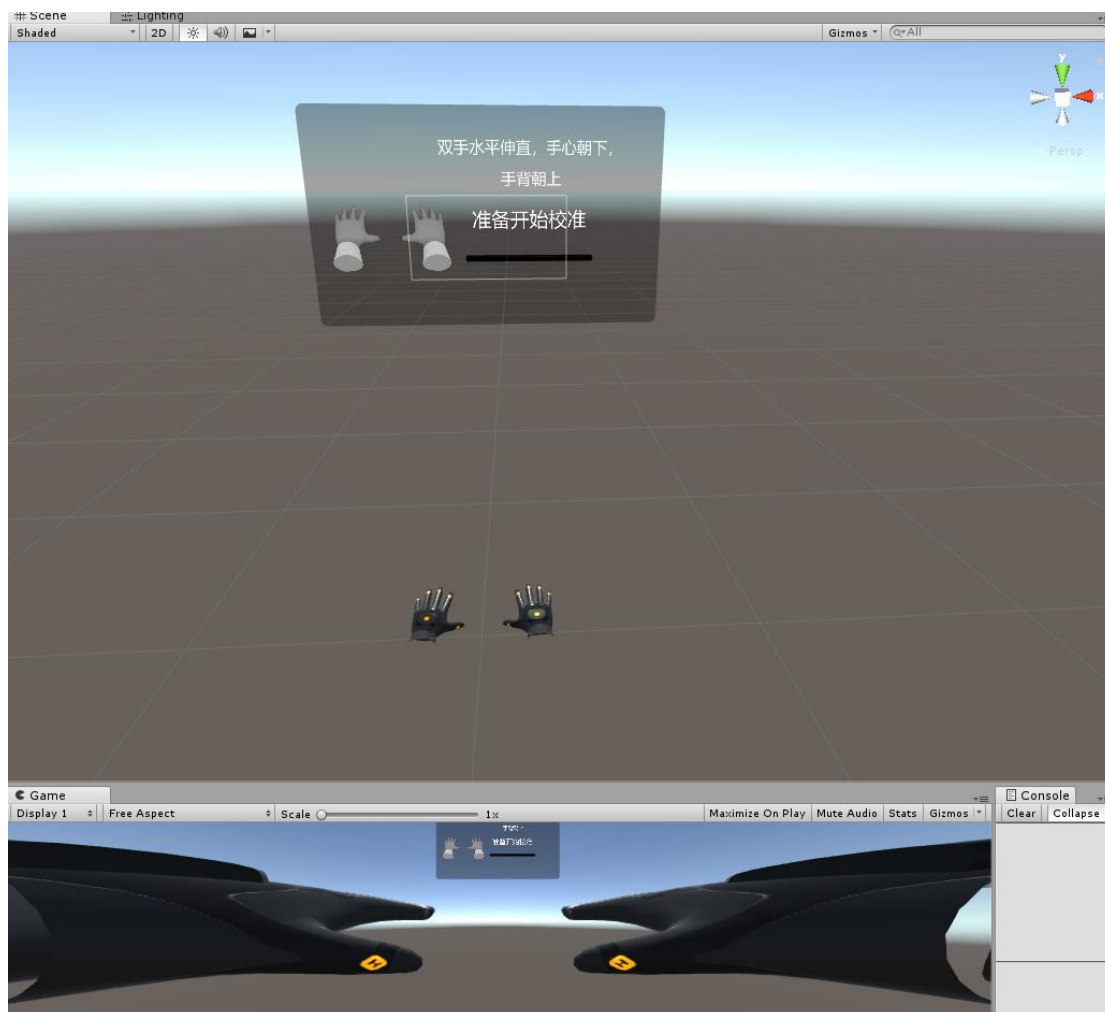
场景中包含核心的几个脚本容器对象分别是

1). NullTouchHandCalibrationController.cs (针对不同的应用场景, 采用不同的双手校准方式)



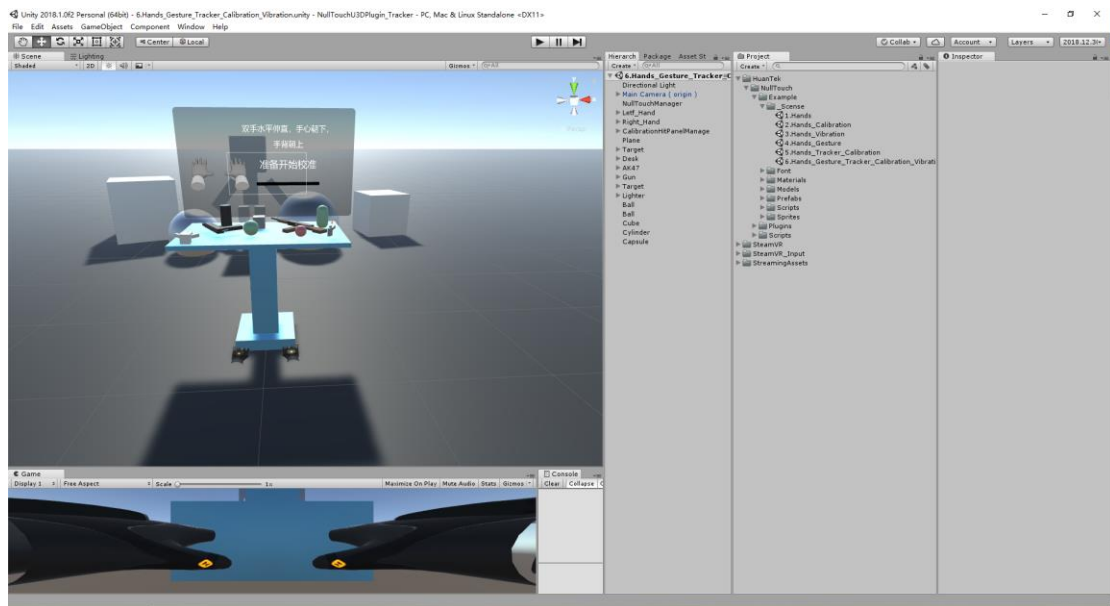
操作过程:

运行前准备双手水平伸直, 手心朝下, 手背朝上>点击 Play 运行



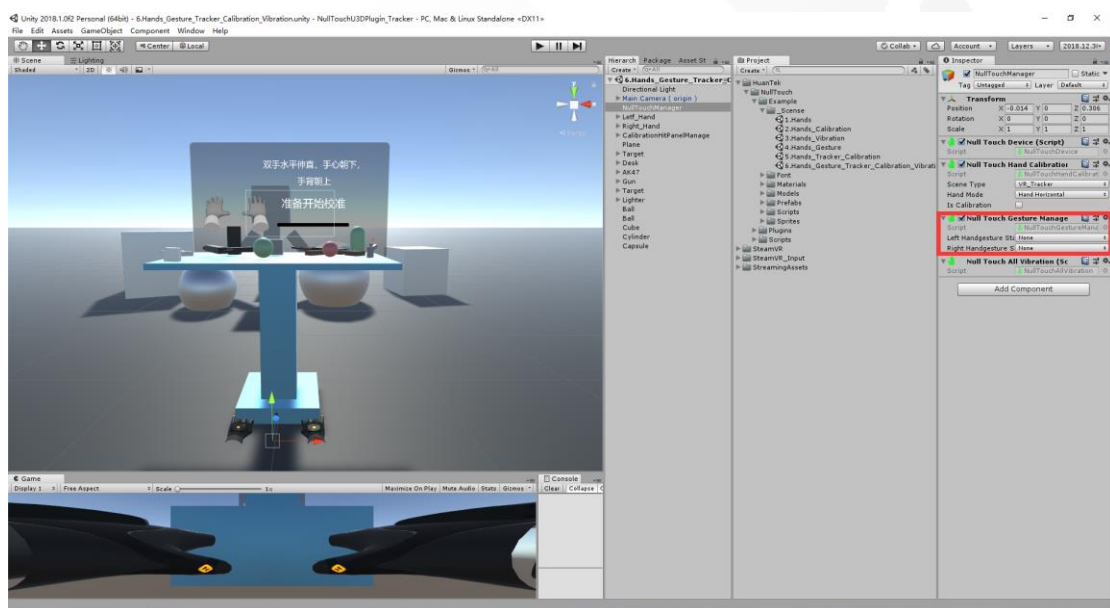
## 2.6. 有双手并且可以在虚拟现实环境中校准手套、抓取、震动功能的示范场景

导入 NullTouch Unity 插件后，打开 HuanTek>Examples>\_Scenes  
> 6.Hands\_Gesture\_Tracker\_Calibration\_Vibration



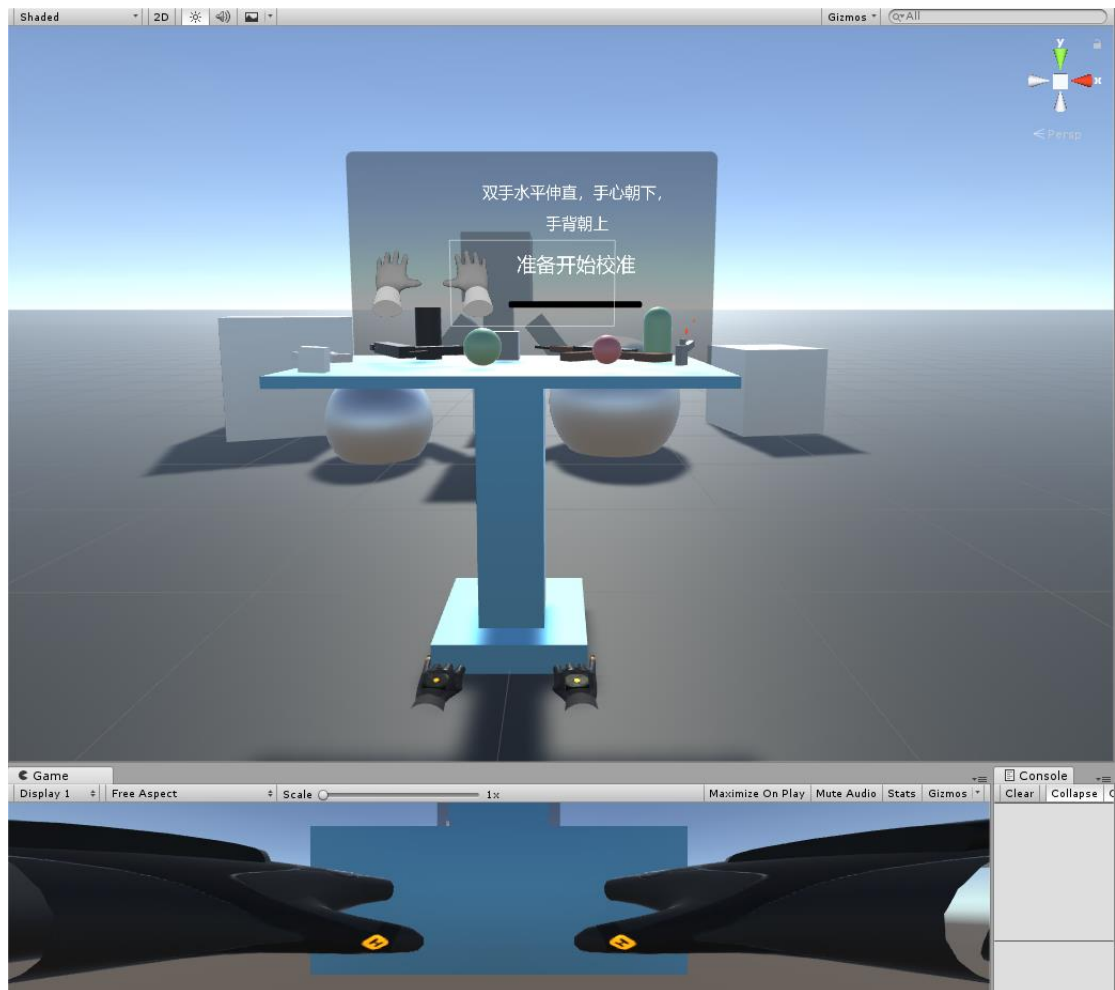
场景中包含核心的几个脚本容器对象分别是

## 1). NullTouchGestureManager.cs(手势管理类，合适复杂功能项目)



操作过程：

运行前准备双手水平伸直，手心朝下，手背朝上>点击 Play 运行>校准完成后双手都可任意抓取桌面上的物体

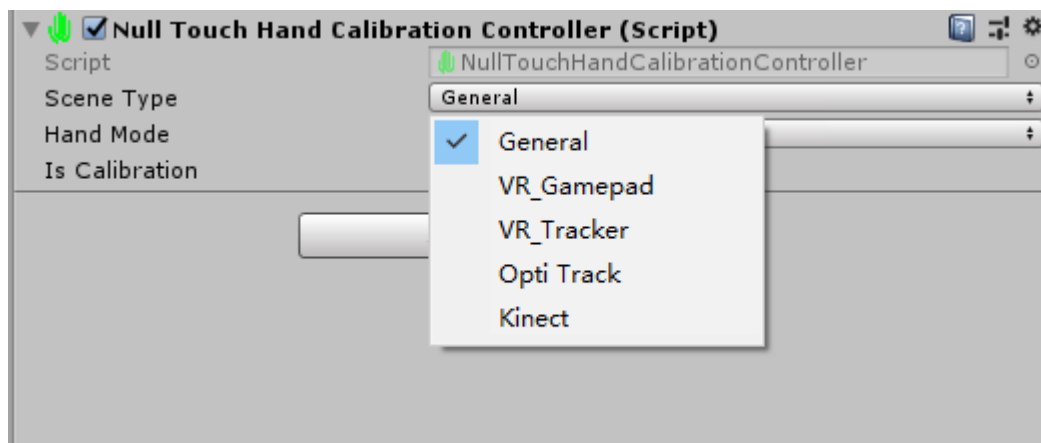


### 3. 开发说明

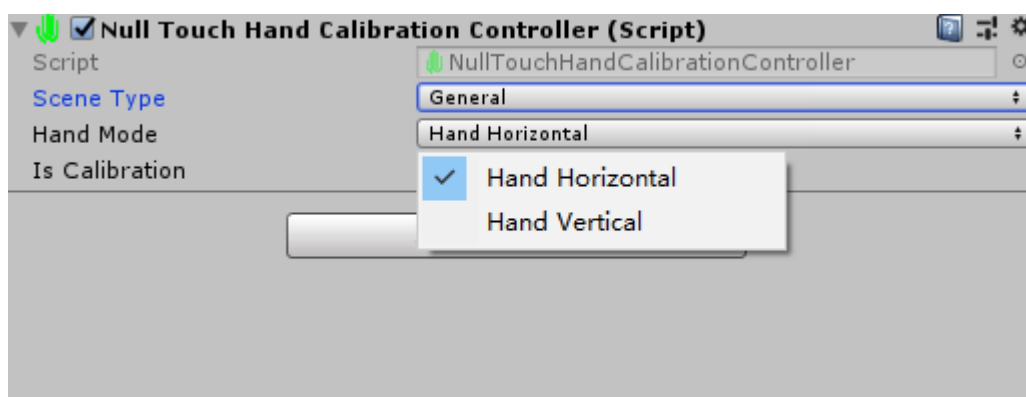
#### 3.1.1. 校准

核心脚本 NullTouchHandCalibrationController.cs, 其中包含可多种不同操作方式的校准方案，在使用的过程中只需要手动配置即可

##### 1).选择定位的方式

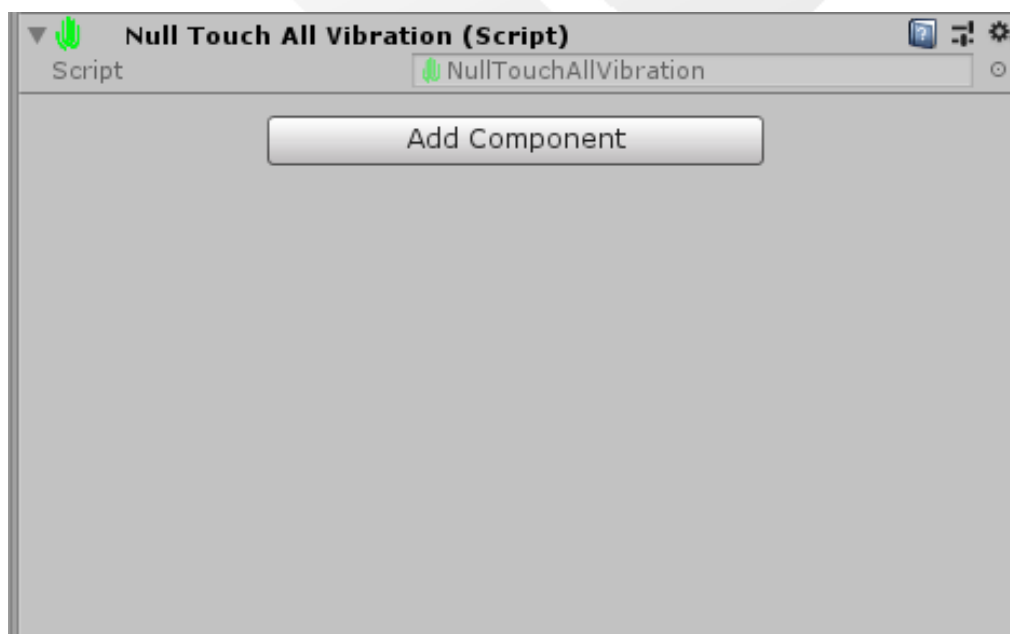


## 2).选择校准时手的朝向



## 3.1.2. 震动

1). 核心脚本 NullTouchAllVibration.cs, 其中提供了手套关节的可震动调用接口。





## 左手

```
/// <summary>
/// 左手 拇指 震动效果
/// </summary>
public void SetLeftThumbVibrstion(uint strength, uint time)
{
    SetVibrstions(NT_HAND.NT_LEFT, NT_MOTORS.NTM_THUMB, strength, time);
}

/// <summary>
/// 左手 食指 震动效果
/// </summary>
public void SetLeftIndexVibrstion(uint strength, uint time)
{
    SetVibrstions(NT_HAND.NT_LEFT, NT_MOTORS.NTM_INDEX, strength, time);
}

/// <summary>
/// 左手 手掌 震动效果
/// </summary>
public void SetLeftPalmVibrstion(uint strength, uint time)
{
    SetVibrstions(NT_HAND.NT_LEFT, NT_MOTORS.NTM_PALM, strength, time);
}

/// <summary>
/// 左手 食指+手掌 震动效果
/// </summary>
public void SetLeftIndexPalmVibrstion(uint strength, uint time)
{
    SetVibrstions(NT_HAND.NT_LEFT, NT_MOTORS.NTM_INDEX_PALM, strength, time);
}

/// <summary>
/// 左手 拇指+食指 震动效果
/// </summary>
public void SetLeftThumbIndexVibrstion(uint strength, uint time)
{
    SetVibrstions(NT_HAND.NT_LEFT, NT_MOTORS.NTM_THUMB_INDEX, strength, time);
}

/// <summary>
/// 左手 拇指+手掌 震动效果
/// </summary>
public void SetLeftThumbPalmVibrstion(uint strength, uint time)
{
    SetVibrstions(NT_HAND.NT_LEFT, NT_MOTORS.NTM_THUMB_PALM, strength, time);
}

/// <summary>
/// 左手 所有部位 震动效果
/// </summary>
public void SetLeftAllVibrstion(uint strength, uint time)
{
    SetVibrstions(NT_HAND.NT_LEFT, NT_MOTORS.NTM_ALL, strength, time);
}
```

## 右手



```
/// <summary>
/// 右手 拇指 震动效果
/// </summary>
public void SetRightThumbVibrstion(uint strength, uint time)
{
    SetVibrstions(NT_HAND.NT_RIGHT, NT_MOTORS.NTM_THUMB, strength, time);
}

/// <summary>
/// 右手 食指 震动效果
/// </summary>
public void SetRightIndexVibrstion(uint strength, uint time)
{
    SetVibrstions(NT_HAND.NT_RIGHT, NT_MOTORS.NTM_INDEX, strength, time);
}

/// <summary>
/// 右手 手掌 震动效果
/// </summary>
public void SetRightPalmVibrstion(uint strength, uint time)
{
    SetVibrstions(NT_HAND.NT_RIGHT, NT_MOTORS.NTM_PALM, strength, time);
}

/// <summary>
/// 右手 食指+手掌 震动效果
/// </summary>
public void SetRightIndexPalmVibrstion(uint strength, uint time)
{
    SetVibrstions(NT_HAND.NT_RIGHT, NT_MOTORS.NTM_INDEX_PALM, strength, time);
}

/// <summary>
/// 右手 拇指+食指 震动效果
/// </summary>
public void SetRightThumbIndexVibrstion(uint strength, uint time)
{
    SetVibrstions(NT_HAND.NT_RIGHT, NT_MOTORS.NTM_THUMB_INDEX, strength, time);
}

/// <summary>
/// 右手 拇指+手掌 震动效果
/// </summary>
public void SetRightThumbPalmVibrstion(uint strength, uint time)
{
    SetVibrstions(NT_HAND.NT_RIGHT, NT_MOTORS.NTM_THUMB_PALM, strength, time);
}

/// <summary>
/// 右手 所有部位 震动效果
/// </summary>
public void SetRightAllVibrstion(uint strength, uint time)
{
    SetVibrstions(NT_HAND.NT_RIGHT, NT_MOTORS.NTM_ALL, strength, time);
}
```

### 3.1.3.手势



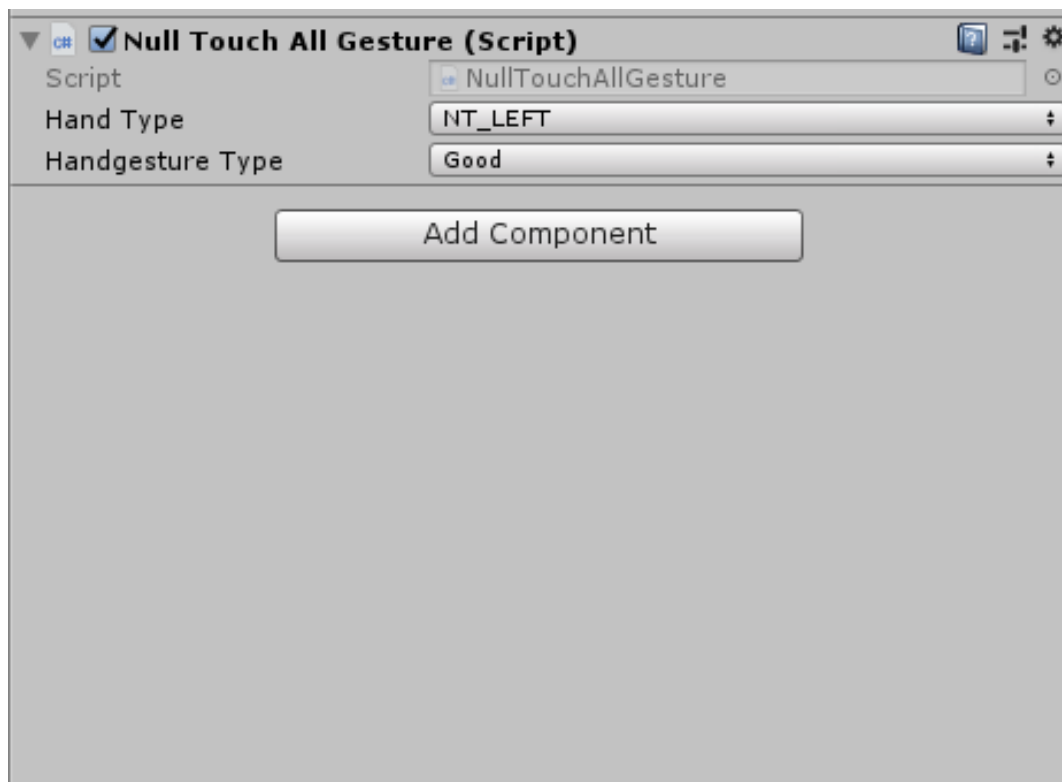


Name	Label	手势用途	描述	手势照片
Good	1	呼出菜单 UI	拇指伸直，其余手指弯曲	
Fist	2	关闭菜单	手紧握拳头	
OK	3	确认	拇指食指闭合成圆，其余手指伸直，手指分开。	
Nico	4	呼出手势提示	中指无名指并拢弯曲，其余手指伸直	
Paper	5	释放物体，可以做碰撞处理	五指伸直，手指可并拢或稍微分开	
Point	6	选择物体	食指伸直，其余手指弯曲	
Grab	7	把物体吸引到手心	手指中度弯曲	
Scale	8	放大或缩小物体	拇指食指伸直，其余并拢弯曲	
Gun	9	发射子弹	拇指食指中指伸直，中指食指并拢，其余并拢弯曲	
Rotate	10	旋转物体	拇指小指伸直，其余并拢弯曲	

注：上图为 NullTouch 中可提供的所有手势



1). 核心脚本 NullTouchAllGesture.cs, 提供识别当前指定的手势的类型



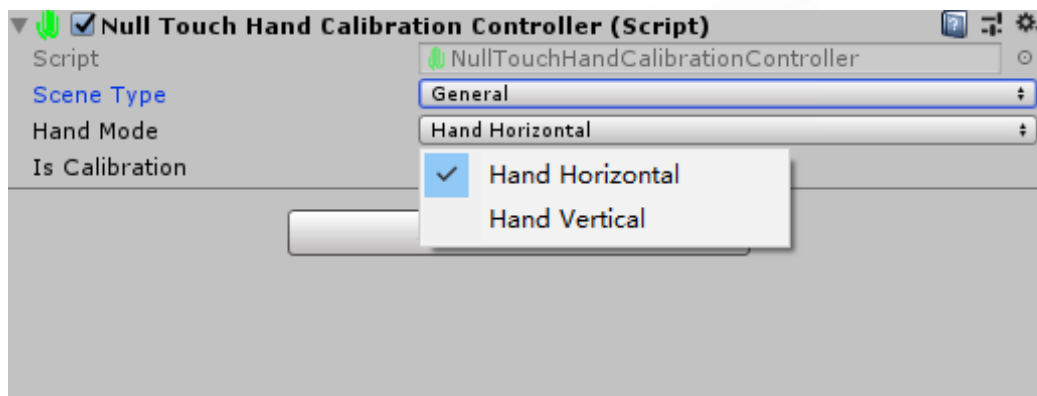
### 3.1.4 虚拟现实（主要说明在此类场景中 NullTouch 如何配合使用 Tracker 功能）

核心脚本 NullTouchHandCalibrationController.cs, 其中包含可多种不同操作方式的校准方案, 在使用的过程中只需要手动配置即可

1).选择 VR\_Tracker 的定位方式



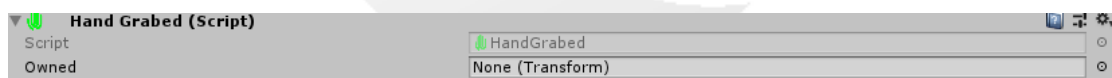
## 2).选择校准时手的朝向



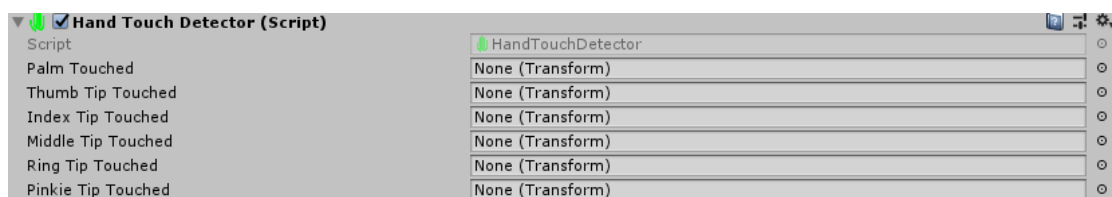
## 3.1.5. 抓取

场景中包含核心的几个脚本容器对象分别是

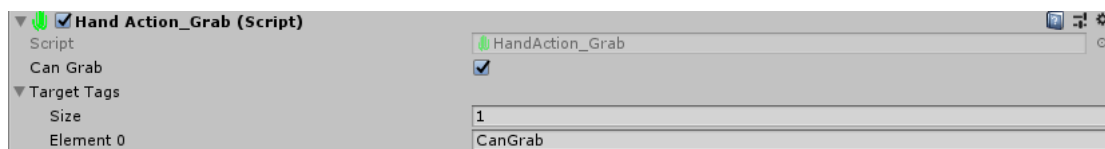
### 1). HandGrabed.cs, 判断手中持有的物品



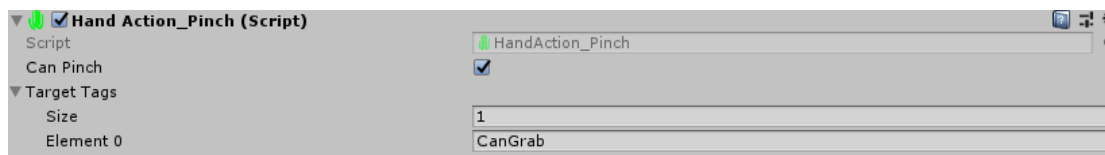
### 2). HandTouchDetector.cs, 手的接触检测器, 检测手掌与手指都碰到了什么物体



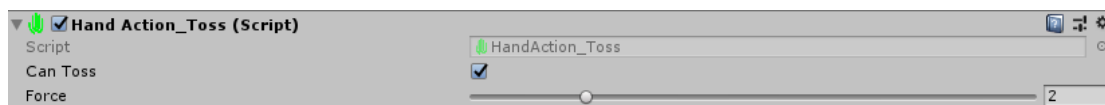
### 3). HandAction\_Grab.cs, 指定为手做出 Grab 的手势时的抓取功能



4). HandAction\_Pinch.cs, 指定为手做出 Pinch 的手势时的抓取功能

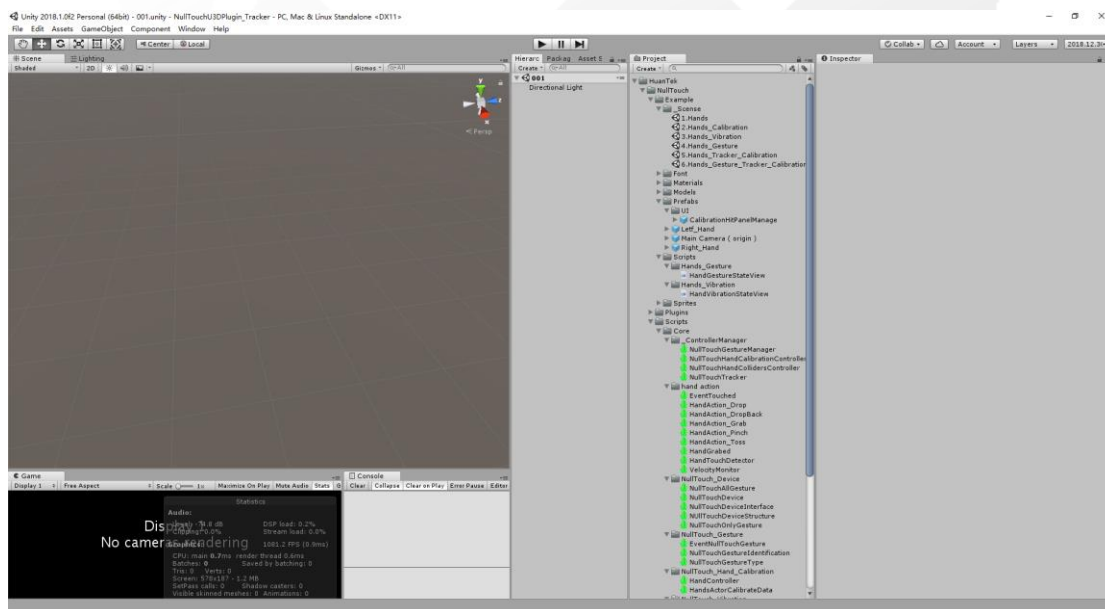


5). HandAction\_Toss.cs, 指定为手做出 Toss 的手势时的抛投功能

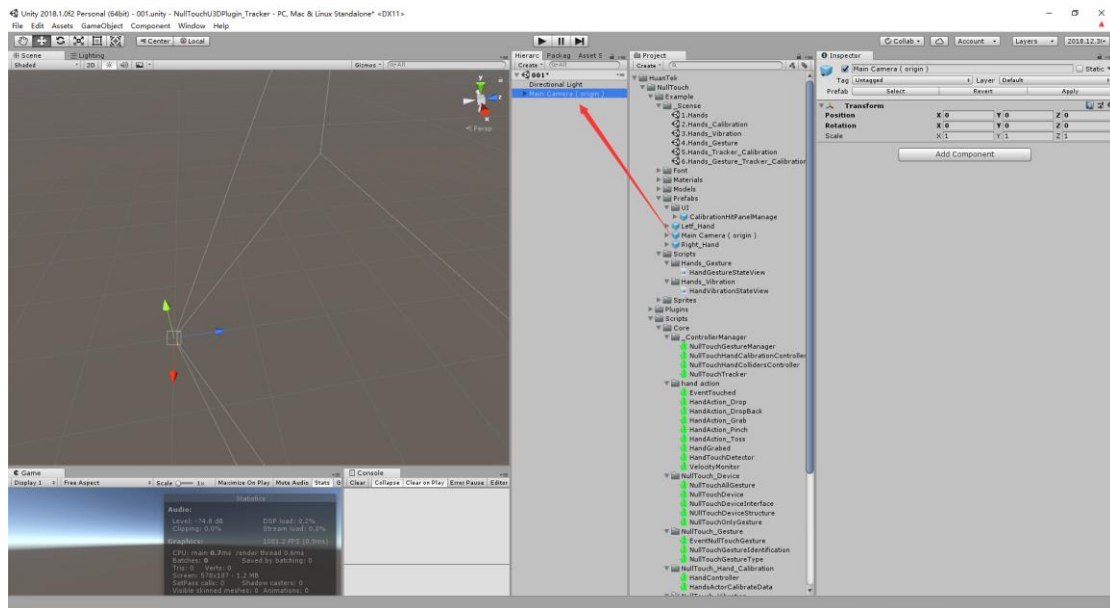


## 4.自定义场景(虚拟现实环境, 校准, 抓取, 震动, 手势, 更换手模型)

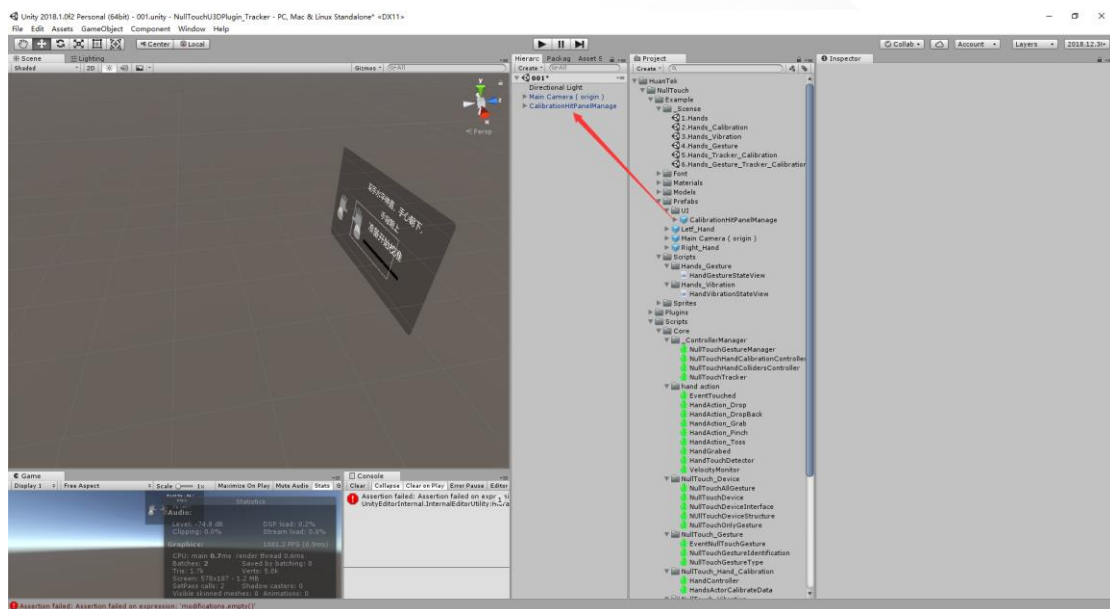
1).新建空白场景, 删除默认摄像机



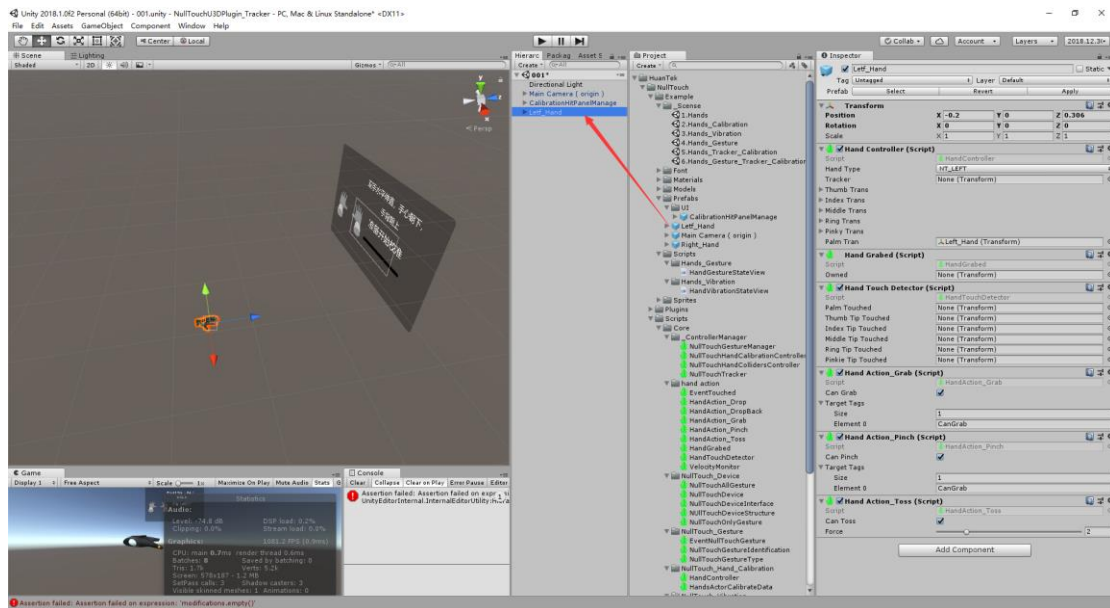
2).添加 Main Camera ( origin )预制体



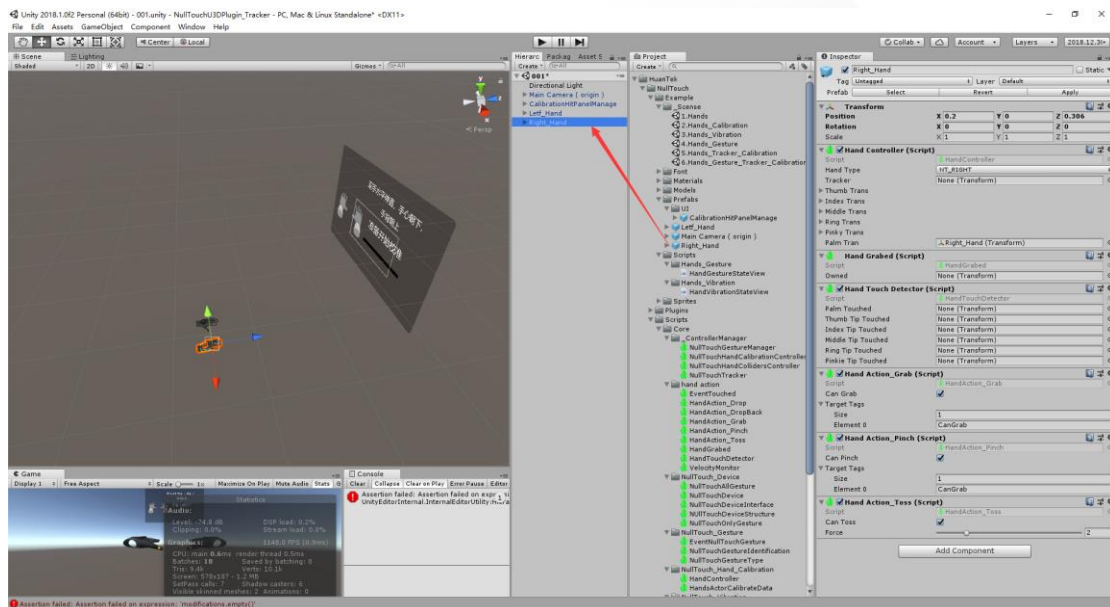
### 3).添加 CalibrationHitPanelManage 预制体



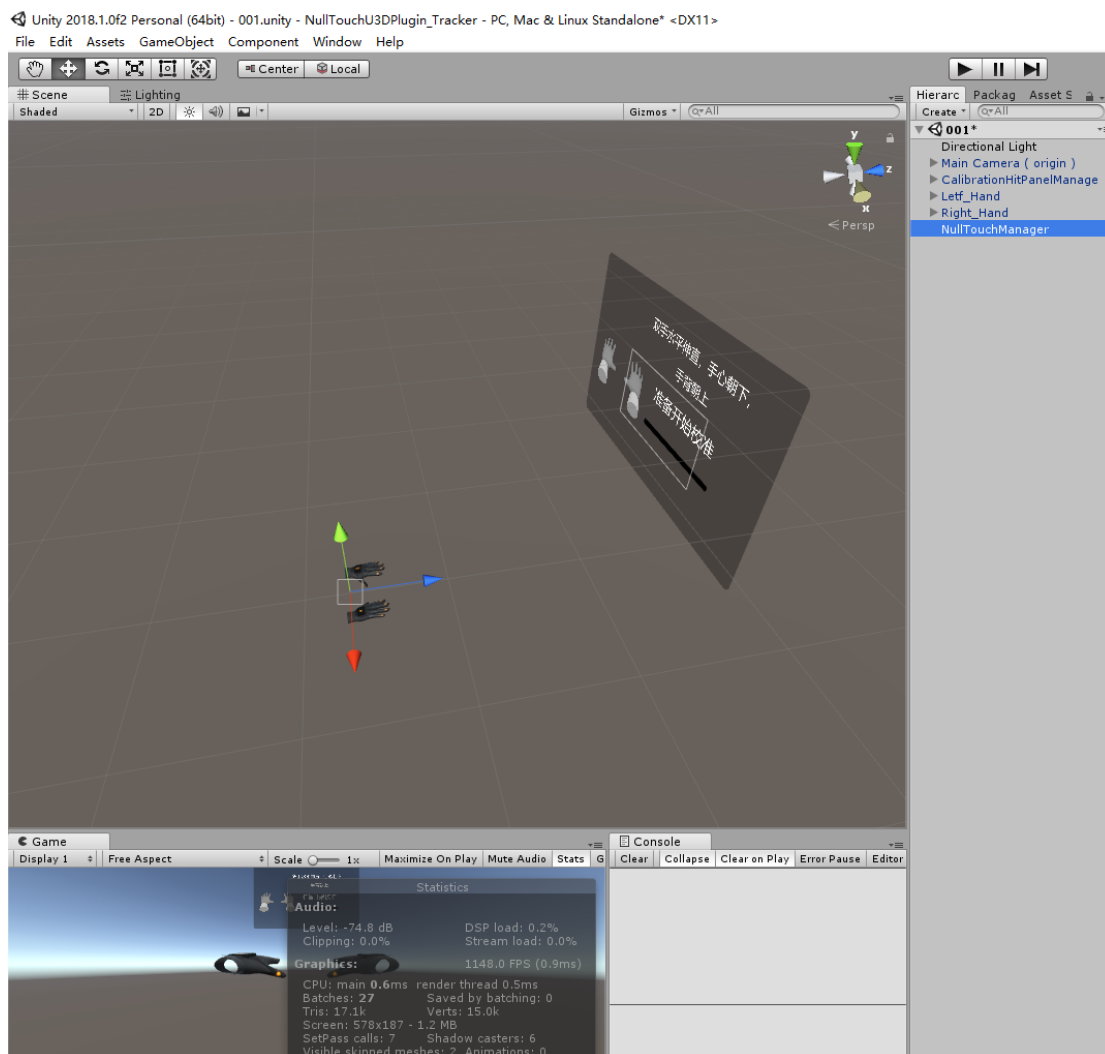
### 4).添加 Left\_Hand 预制体



## 5).添加 Right\_Hand 预制体

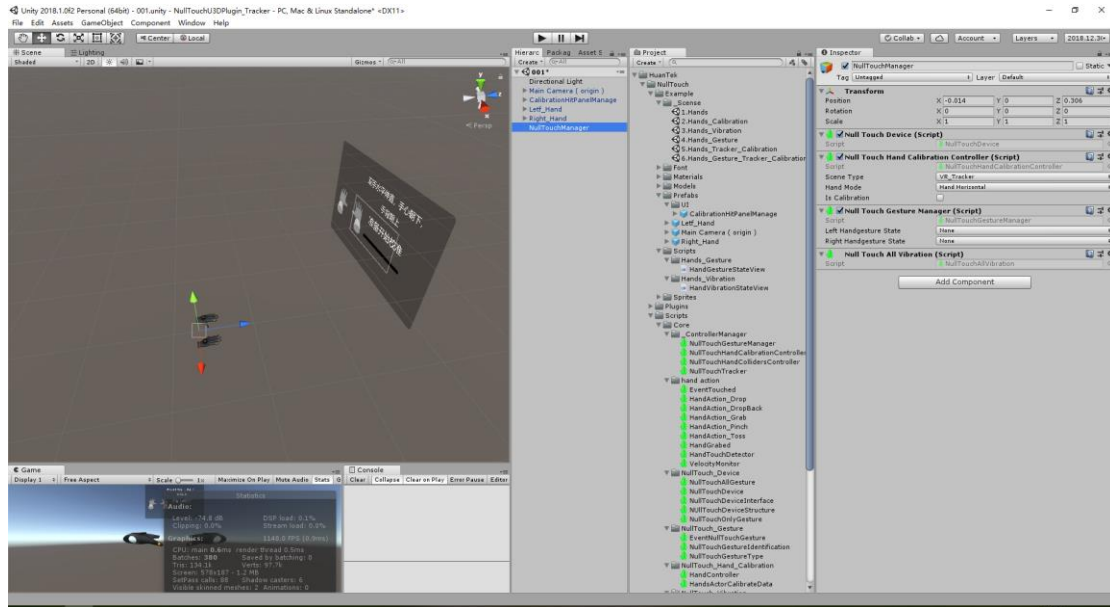


## 6).创建一个 NullTouchManager 的游戏对象

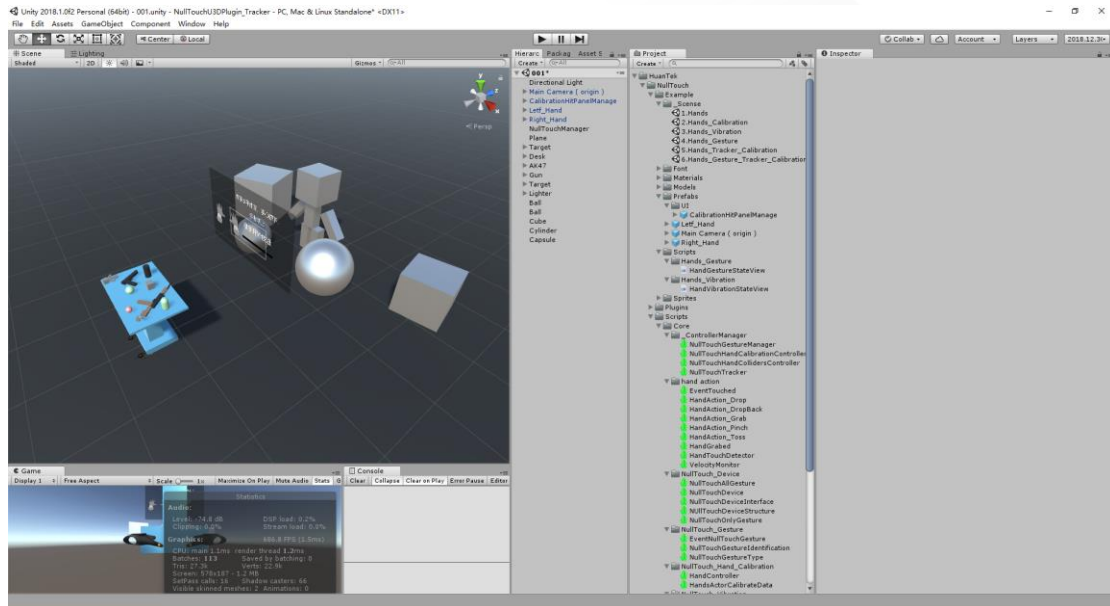


7).给 NullTouchManager 的游戏对象挂载，  
NullTouchDevice、  
NullTouchHandCalibrationController（记得设置指定的定位方式以及手的摆放方式）、  
NullTouchGestureManager、  
NullTouchAllVibration 组件



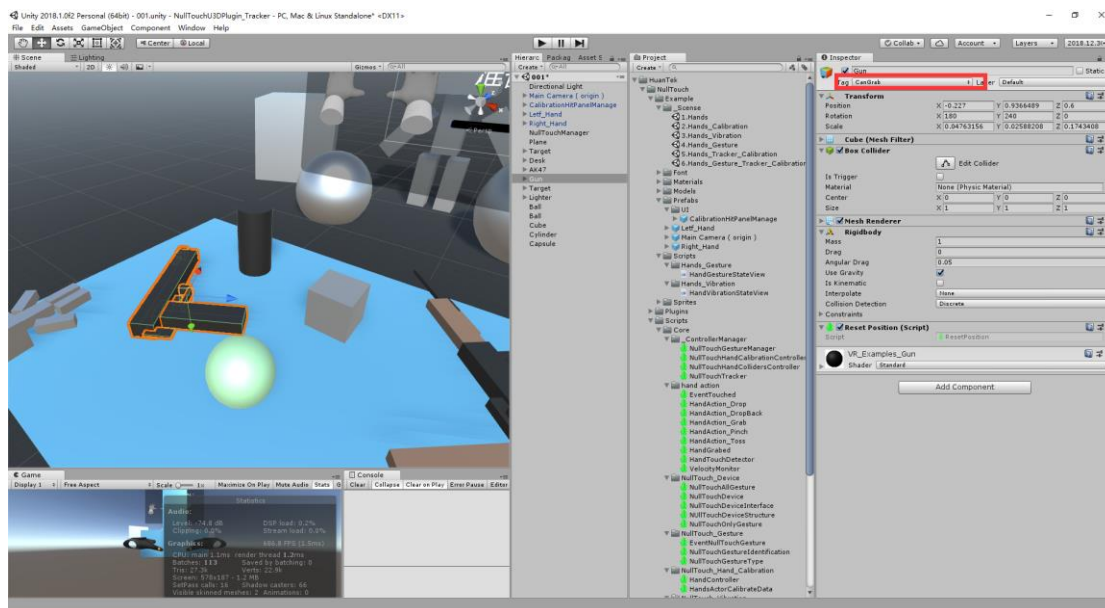


## 8).在场景中添加可抓取的游戏对象

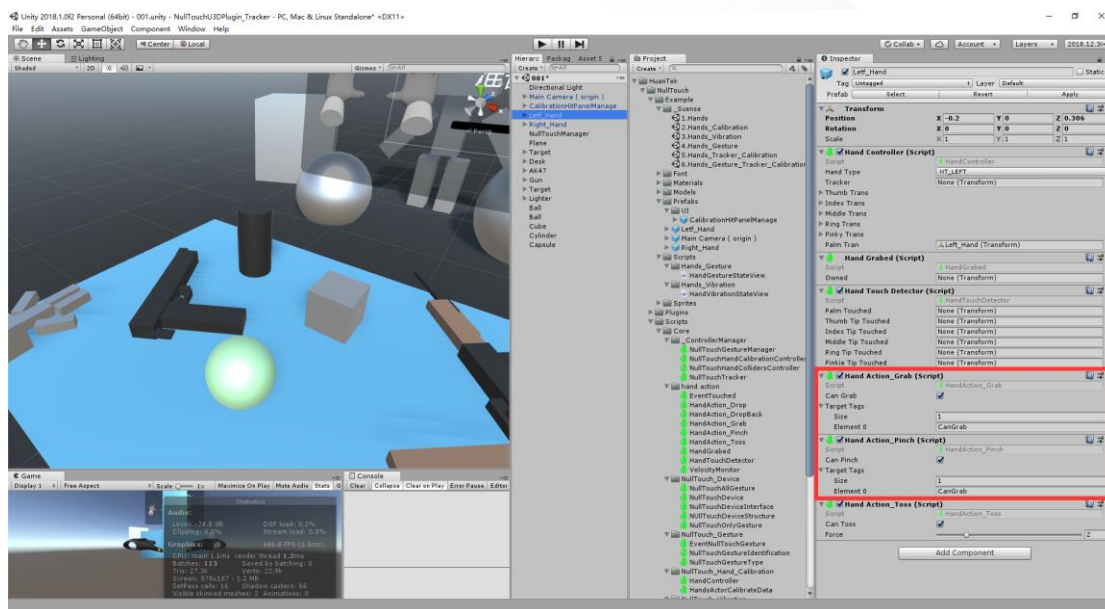


并设置其指定的标签（标签为手势类型）





手势类型参考此组件设定的目标标签



## 5.报错说明

收集问题中...