

## Htc Vive VR 手势识别插件教程

由 Htc Vive 开发者联盟萌主 CloudHu 特别翻译



群名称: Htc Vive开发者联盟  
群 号: 566189328



资源商店插件地址:<http://u3d.as/uF5> 或者直接搜索 VR Infinite Gesture



前言:该插件需要接近 90 美刀,折合人民币 601 元多,有点小贵(土豪可以无视),所以联盟的成员们决定一起合力购买,以节约大家的学习成本.关键是我们并不知道该插件的实际效果,是否好用,所以这算是一个供学习和测试的版本.

特此声明:该插件我们是用来学习的,根据国际软件学习惯例,我们可以免费学习任何软件,如果要用来商用,请购买正版.

联盟出钱的成员都是自愿的,所以并没有定价,可以贡献任意数值,在此特别感谢他们的信任和慷慨,我们才得以有机会学习和尝试这个特别的功能.

募集通道 [支付宝](https://www.alipay.com/):15386585129 [QQ](https://www.qq.com/):604746493 [微信](https://www.wechat.com/):cloud198923

优酷视频网址:[http://v.youku.com/v\\_show/id\\_XMTYyODQxMDQ2MA==.html](http://v.youku.com/v_show/id_XMTYyODQxMDQ2MA==.html)

联盟贡献者(随机顺序):凌东鑫 8 元 稳稳的幸福 15 小胖 20 武汉\_宇 10 ly10

凡维创客 10 元 若飞 13 小明 10 炜鹏 10 饮 15

超 5 志若 10 元 玥 5 晋 10 肖攀 30 mx10 kktc3322 10

Szywt666 10 元 大发 10 元 大翔子 10 王露 10 蓝冬瓜 10

春晓 8 元 无与伦比 10 元 笑笑 20 待续 10 前行者 10

梦龙 5 鬼鬼 10 Y10 元 骑士 20 Symae10 国人 10 zyfforce 10

守护 15 小子 10 木头 10 云图 20 longxiaolai 10 元 CloudHu31

liang110591 10 元 颖坤 10 利坤 20 晓峰 10 老王 10

yuwenhao0214 20 元 人来人往 10 冰澈 10 AcodeWang 10

共计 600 元整.

这一次大家能够集体响应联盟的号召实在是令人感动,我们只用了短短一上午就凑集了足够的资金购买这个昂贵的插件,大家的力量是巨大的,联盟万岁!

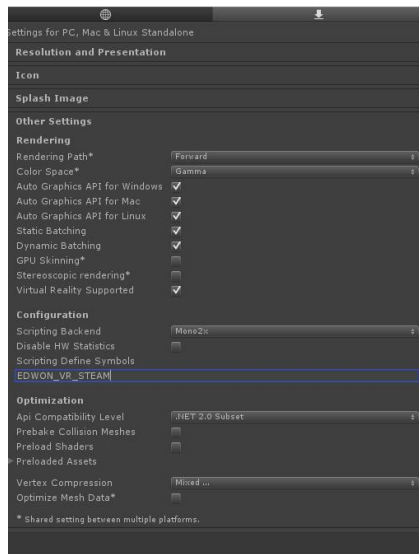
**那么接下来我来介绍下该插件的使用说明好了,特别说明该插件是支持 Oculus 的,这里只做**

**Htc 的部分,详情参考官网 <http://edwonstudio.com/vr-gesture> 教学一栏:**

## **一,新手教学**

**1.打开项目设置 Edit->project settings->Player.**

然后找到 *Scripting Define Symbols* 里面输入:EDWON\_VR\_STEAM.



然后导入 [SteamVR plugin](#).



启动 [Steam](#),并通过在 [Steam](#) 右上角的 [VR](#) 按钮启动 [Htc Vive](#),运行房间设置,确定头盔和手柄都一切正常.

2.然后导入 [VR Gesture Tracker](#), 打开 "[Getting Started](#)" 案例场景, 路径是:[Edwon/VR/Gesture/Examples/Getting Started](#).

3.把 [SteamVR plugin](#) 中的 "[[CameraRig](#)]" 预设拖拽到该场景当中,预设的路径为:[SteamVR/Prefabs/\[CameraRig\]](#).

4.添加 [VRGestureRig](#) 组件在 [camera rig](#) 上面,你可以在以下路径找到该脚本:

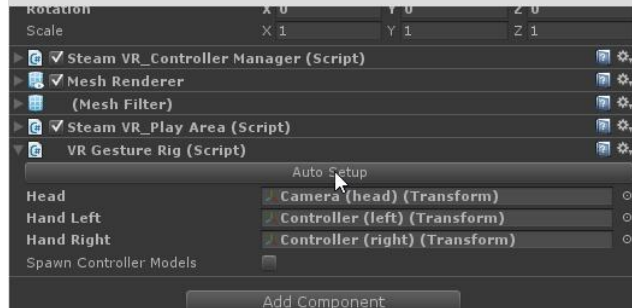
[Edwon/VR/Gesture/Scripts/VRGestureRig](#)

这个脚本就是 [vr gesture](#) 系统怎样知道你的头和手分别在哪里.

单击该组件上的 "Auto Setup"按钮.

这样这些变量会自动设置好自己了!

"Spawn Controller Models" 不要勾选!



5.创建一个新的神经网络,选择 VR Gesture Manager 跳到 inspector 细节窗口.

选择"Train"标签.

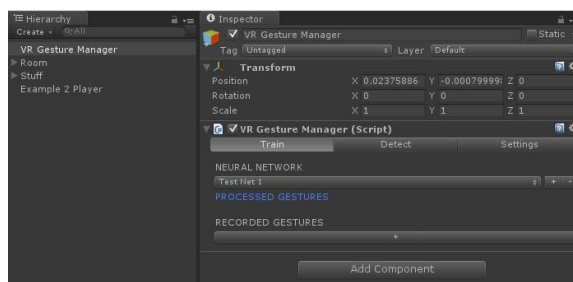
点击大大的“+” 按钮.

输入你的新网络的名称. 就叫它 “Test Net 1”好了!

点击创建 Create.

这会新建一个名为“Test Net 1”的文件夹在这个路径下:

Edwon/VR/Gesture/VRGestureData/



6.添加一些手势到网络里面.在 VR Gesture Manager 上面找到 NEURAL NETWORK,

确定“Test Net 1”被选择.

创建线条手势:

在 RECORDED GESTURES 下面按 “+” 按钮.

重命名 “Gesture 1”成“Line”.

## 创建彩虹手势:

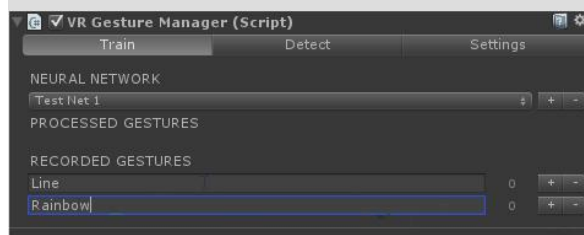
同样的方法再创建另外一个手势叫做“Rainbow”.

你可以通过在第一个手势旁边点击“+” 按钮来添加.

你应该有两个叫做“Line”和“Rainbow”文件夹在这个路径下:

`Edwon/VR/Gesture/VRGestureData/Test Net 1/`

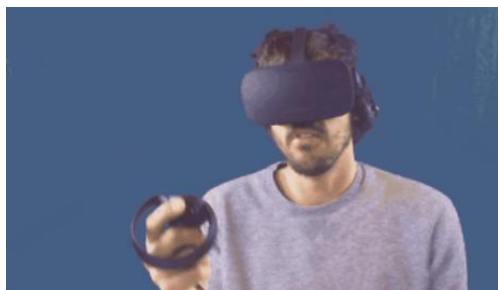
现在“Line”和“Rainbow”只是空文本(.txt)文件.



但是一旦我们开始记录手势,它将会存储所有的手势记录数据到这些文件中.

现在,我们到 VR 里面开始记录一些手势吧!

点击 *Play* 进入 VR 空间,戴上头盔,拿起手柄.



## 7.记录更多的“Line”手势案例

神经网络通过采集大量的样品数据来工作并“训练”神经网络知道怎样去识别,或者“分类”这些数据来增强准确性.你给神经网络越多的手势样本,它就能更好的识别手势.

## 记录 20 个“Line”手势样本:

来开始记录大约 20 多个线条的样本.

这些样本不一定要一模一样,实际上有一点变化是好的.

位置,朝向,和手势的方向,相对于你的头部视角和位置,是很重要的!

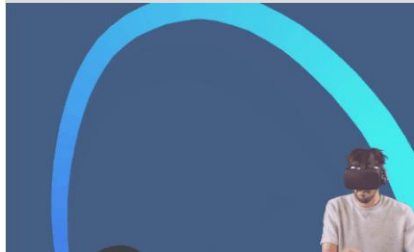


## 8.记录一些“Rainbow”手势样本

神经网络不能只探测一个手势,你需要至少两种类型的手势来探测.

**在 VR 里面, 从记录菜单中选择 "Rainbow"**

记录大约 20 多种这种彩虹手势.



## 9.编辑手势样本

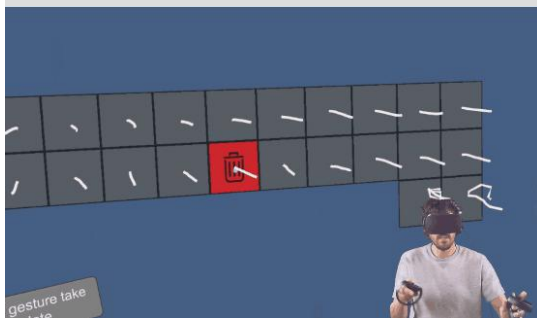
**在 VR 里面, 从主菜单中选择 "Edit" 然后选择 "Rainbow"**

所有记录好的“Rainbow”手势样本会在一个个格子中弹出来.

我们创建这个画廊,你才能知道你之前都记录了些什么.如果你有一段没有记录好,这会使神经网络困惑,所以我们可以编辑那些不好的记录.

### 试着删除一条记录

指向那条记录然后单击,你就把该记录删除了.



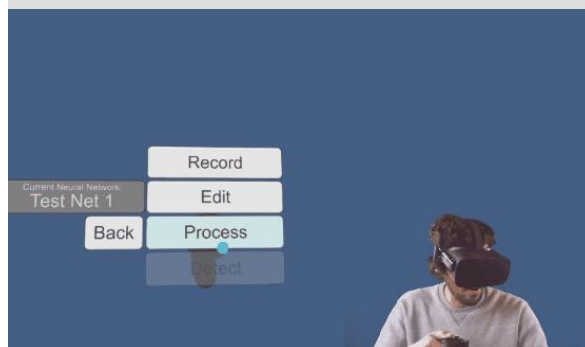
## 10.运行神经网络

现在我们将要“运行”叫做“*Test Net 1*”的神经网络了, 我们记录了一大堆动作样本.

“运行” 是我们实际上使用你刚刚记录的“*Line*”和“*Rainbow*”,并将其做成一个能够神奇地识别这些手势的小脑袋.

技术上说,这比那个更加地复杂,但是这超出这个教程的范围.

在 VR 里面, 从主菜单中选择 “**Process**” .



VR 会停顿一秒.不用担心,这只是神经网络最大化的使用电脑性能来创建一个新的生命形式.

只需要两种手势和一个带得动 VR 的主机(至少 i5CPU,最低 GTX970 显卡,最低 8G 内存,配置越高,效果越好),训练新的神经网络大脑只需要 1 秒.

更多的手势和更多的记录,这个过程会需要更多的时间.在我们项目中使用 6 或 7 种手势最多需要 10 秒.

记住,你只有在你想要添加新的手势或者手势记录到网络的时候需要再次运行,并且在大多数情况下,你不需要在玩游戏的时候训练网络大脑.

## 11.检查网络大脑

这个我们刚刚创建的“*Brain*”是一个叫做“*Test Net 1*”的文件夹, 你可以在“*Test Net 1*”文件夹下找到它.

你如果想看的话你可以看看文件里面的东西,但是请不要修改它.

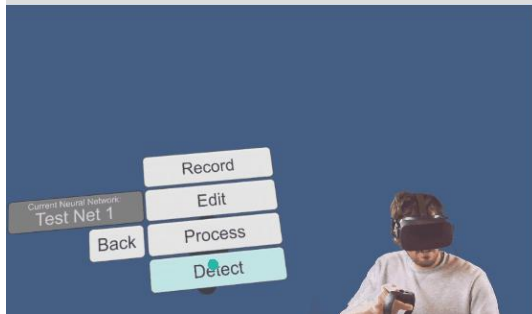
它只是一大堆代表着神经网络的疯狂数字并且总是被“*Process*”按钮自动生成.

## 12. 探测手势

好了,现在我们已经完成了“Test Net 1”的训练,一起来试试效果吧!

### 进入“Detect”模式:

在 VR 里面从主菜单中点击“Detect”.



### 随便尝试一些手势:

现在按住按钮试着做“Line”或“Rainbow”手势.

如果手势被正确探测到,它会在手上的 UI 中显示该手势的名称.

欧耶!

记住,网络会知道你是在画什么样的线条,无论是上下左右.

因此,你画的方向必须和你记录的一致.如果你想要探测来自不同方向的线条,你可以多记录一些手势样本并再次运行神经元网络.

## 13. 恭喜你!你刚刚已经创建了手势检测 AI 生命形式!

现在你已经创造了有意识的生命,你也许会想要改进它.

## 二. 使用案例

在这个教学中,我们将会设置我们的案例场景和案例数据,并且看它们是怎样工作的.

你也可以看我们的视频教程 [HERE](#)

别担心,视频教程和文字教程是一模一样的.

### 1. 导入案例包

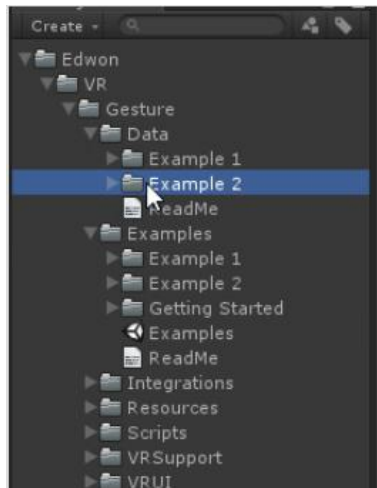


从下面的路径中导入"Examples"包:

*Edwon/VR/Gesture/Examples/Examples.unitypackage*

这会导入一些案例场景(也有一些美术资源和脚本)到 Examples 文件夹中.

它也会为每一个场景导入一些案例手势数据和预先运行过的神经网络.



这些案例中的神经网络我们已经预先运行过了,无需再记录任何手势来开始!

## 2.为 VR 设置"EXAMPLE 1"场景

只需跟着新手教学中的 1 到 4 设置即可.

## 3.在 VR 中体验"EXAMPLE 1"场景

点击 Play 进入 VR 空间.

选择"Example 1"并进入"Detect"模式

尝试一些手势!

在这个场景中有四种可以触发的手势.

每一种对应一个形状,如果你做对了手势会变成红色.

案例中的以下每一种手势和形状会高亮:

**三角形 圆形 推 拉**

## 4.看一下"Example 1 Player"脚本

在场景中选择"Example 1 Player".

打开叫做"Example 1 Player"的脚本看看.

这个脚本监听手势脚本.你可以从这里了解更多事件 [INTEGRATION](#)

在"Example 1 Player"脚本里面,当收到一个 `GestureDetectedEvent` 事件,`OnGestureDetected` 方法就会被调用.

在那个方法里面,我们根据"gestureName"变量来改变一个形状成红色.

## 5.为 VR 设置"EXAMPLE 2"场景

只需跟着新手教学中的 1 到 4 设置即可.

## 6.在 VR 里面试试"EXAMPLE 2"场景

这是我们在为这个插件制作的 [TRAILER](#) 中使用的同一个场景!

所有你在那个视频里面看到的手势你都可以制作.

现在轮到你使用虚拟的超能力了!

点击 Play 进入 VR 空间.

选择"Example 1"并进入"Detect"模式.

尝试一些手势!

在这个场景中有六种可触发的手势.

每一种都对应一种超能力!尽情去尝试他们吧!

六种不同的超能力分别是:

Air 空气炮 Earth 土遁 Fire 火遁 Gravity 地震 Ice 冰弹 Pull 吸星大法

## 7.看一下"Example 2 Player"脚本

在场景中选择"Example 2 Player".

打开叫做"Example 2 Player"的脚本看看.

这个脚本监听手势脚本.你可以从这里了解更多事件 [INTEGRATION](#)

在"Example 2 Player"脚本里面,当收到一个 *GestureDetectedEvent* 事件,*OnGestureDetected* 方法就会被调用.

在那个方法里面,我们根据"gestureName"变量来发射超能力.

### 关于这些超能力

实现这些超能力对于这个案例来说不是最重要的,但是它们基本上是通过在场景里生成粒子预设并且/或者通过对场景里面的物体对象应用一些物理力量来工作的.

其中一些预设 prefabs 也有它们自己的脚本(当它们和其他东西碰撞的时候会做一些特效).

### 三.如何构建

在这个教程中我们将使用 VR 手势追踪来构建你的游戏.

#### 1.从你的场景中删除 VR 手势 UI

假如你不想让玩家在游戏里面随机地弹出 VR 手势 UI 的话 ,确定这些 UI 在你的所有场景中删除了 .

#### 2.设置

确认你已经在 *Project Settings > Player* 中定义了:EDWON\_VR\_STEAM

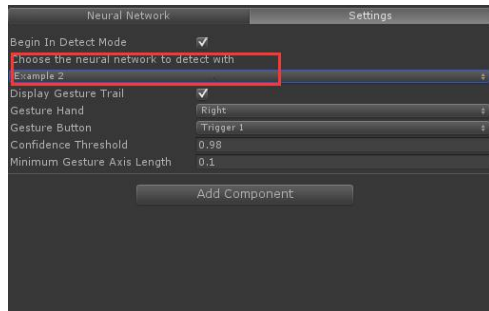
即新手教学里面的第一条设置.

#### 3.把 VR GESTURE MANAGER 设置成"DETECT"

在场景中选择"VR Gesture Manager".

因为它是单例,所以你的场景中应该只能找到一个.

确保它的第一栏设置成"*Detect*",并且你已经选择了你想要玩家使用的神经网络.



#### 4.复制神经网络数据到你构建好的工程中

复制那些你想要玩家使用的神经网络到你构建好的工程中.

复制这个文件夹和它所有的内容:*Edwon/VR/Gesture/Data/MyNeuralNet*

到你构建好的工程目录下:*MyGame/Assets/Edwon/VR/Gesture/Data/*

#### 5.试玩你的游戏

欧耶!你刚刚完成了一个拥有魔法手势的游戏构建!

推荐阅读:

- [HTC Vive 之 Unity3d 开发日记@0](#)
- [HTC Vive 之 Unity3d 开发日记@1](#)
- [HTC Vive 之 Unity3d 开发日记@2](#)
- [HTC Vive 之 Unity3d 开发日记@3](#)
- [HTC Vive 之 Unity3d 开发日记@4](#)
- [HTC Vive 之 Unity3d 开发日记@5](#)
- [HTC Vive 之 Unity3d 开发日记@6](#)
- [The Lab 渲染器\(CloudHu 译\)](#)