

## 如何用Unity和Cardboard做一款VR游戏

发布者: 小篱 | 发布时间: 2016-1-8 10:45 | 评论数: 0

### 游戏程序

平台类型:

程序设计:

编程语言:

引擎/SDK: Unity3D/2D



随着Oculus宣布1月6日开启预售，2016年很可能成为VR游戏元年，但很多的调研显示，手机设备才是市场增长的关键，SuperData发布的报告显示，2016年全球VR游戏市场规模预计在51亿美元左右，消费者设备安装量在3000万左右，凭借价格门槛的优势，以谷歌Cardboard和三星Gear VR为代表的移动VR设备很可能成为用户量最大的设备，可能占据2016年虚拟现实设备安装量71%的份额，达到2700万。

最近，一位从事VR游戏研发的开发者在自己的博客中展示了如何用Unity和Cardboard把一款3D游戏变成VR游戏的方法，他在文章最后还表示，你甚至可以用这种方法把游戏转化为AR游戏，或者从头开始做新的VR游戏：

很多的科幻电影都展示了用人类意志控制物体的能力，但到目前为止，这仍旧是一种幻想，目前还没有技术可以实现这一点。在这份新手教学博客中，你可以学到如何用Unity做一款兼但的虚拟现实游戏，配合谷歌Cardboard创造比较简单的VR体验，本文主要讲的是：

把Cardboard摄像头集成到你的游戏中；调整UI元素以适应VR模式；做VR模式下可以选择的按钮；在游戏运行的时候程序化地把游戏在常规和VR模式下进行切换。

### 什么是谷歌Cardboard？

理论上说，创造VR体验是非常直接的，你在屏幕上展示的并不是一个单独的图片，而是需要2个。因为它们来自位置相隔几英寸距离的两个摄像头，玩家们从左摄像头看到的是左边图像，右边看到的则是右边图像，这就创造了游戏景深。

关注我们：



QQ群



GameRes游资网 官方微信公众账号  
收听我们，更多精彩资讯每日推送

### 最新消息

五个原因告诉你 为什么水浒不是一  
决定是否独立发行前需要考虑的三大  
10岁做出两款游戏 看天才儿童是如

此外，再加上一些不错的动态传感器，你可以探测到用户面朝哪个方向。把这些和你所创造的3D世界结合起来的话，就可以获得一个沉浸感很强的游戏体验。在实际情况中，在一个高分辨率的屏幕上展示两个图像需要非常复杂的硬件，还需要可以追踪玩家的头部运动，把所有的功能集成到一个设备商，还要控制其重量，以免让用户戴起来太累。

然而，随着谷歌Cardboard的出现，每一台智能机都有可能与Cardboard协作，可以用你手机的屏幕与传感器，把它们变成VR设备的一种，而你所需要的，只是一些Cardboard和塑料镜片。

开始准备

需要注意的是，你可能大量用到Unity GUI，所以如果从来没有尝试过这个工具，那么最好是看看Unity GUI教程。想要用Google Cardboard做出自己的VR游戏，你需要以下工具：

Unity专业版，版本5以上；一部智能机，可以是iPhone 5代以后的iOS设备，也可以是4.3以上版本的Android设备。需要说的是，本文假设你是在用iPhone获得VR游戏体验。

如果你从来没有用过Unity而且也不熟悉界面，那最好是去读Unity教程介绍

可能会有人说，我如何获得一个Cardboard硬件呢？

当然，如果还没有Cardboard设备的话，最好是从供应商那里选购，价格加运费大约是20-30美元，如果你觉得自己非常善于DIY，也可以动手做自己的版本。在购买Cardboard的时候，要注意提到V2或者Cardboard 2.0的字样，因为它们适配大量的手机，包括大屏幕的iPhone 6+在内，它们还支持用户通过按钮进行输入操作。

没有Cardboard设备可以吗？

从某种程度上说，是可以的，你仍然可以在自己的设备上运行做出来的游戏，它看起来会是这样的（下图）：

在玩游戏的时候，如果你用恰当的方式盯着屏幕看，也可以找到VR的感觉。如果你四处移动

《上古卷轴》研发商将推出自己的游戏  
费用削减游戏事业表现强：世嘉飒美  
三七互娱联手芒果传媒成立视频孵化

招聘信息 免费发布 HOT  
[北京][上海][杭州][成都]  
[广州][深圳][厦门][更多]  
[轩意网络高薪诚招各类程序、策划、  
同门文化 聘 次世代美术](#)  
[极贝游戏新项目诚聘程序员|策  
擎天柱 诚聘系统、数值、文案策划、](#)  
[阿里巴巴集团&阿里游戏招聘](#)  
[厦门吉比特重大主推项目启动，各岗  
游艺春秋 猎投资团队&聘研发及运营](#)  
[厦门凡古网络招聘U3D程序、3D美](#)

手机，就可以进行控制。虽然你也可以玩这款游戏，还能够看到游戏的实际效果，但这种体验是比较有限的。

长话短说，如果你没有足够的耐心等待Cardboard到货，你仍然可以从本文中学到东西，只是，如果有合适的设备，你能够学的东西更多。

### **样本游戏：Ninja Attack is Back !**

投入一些时间尝试这个样本游戏，下载并解压这个Unity新手项目。接下来就是运行Unity，在欢迎界面，选择打开（Open）-找到StarterNinja文件夹，打开NinjaAttack项目。

在项目浏览器（Project Browser）中，在Assets里双击MainScene按钮，然后点击Play尝试一下这款游戏。在图片中，你是左边的忍者，随着怪兽在屏幕中出现，你可以在屏幕上点击任何地方释放一个忍者之星消灭怪兽，在击杀20个怪兽之后，你就赢了，但是，如果怪兽到达了左边的红色区域，你就输了。

这款游戏看起来很熟悉是吗？熟悉的读者可能会发现，它其实就是我之前介绍SpriteKit以及Cocos2D教程用到的游戏，不过与之不同的是，这里的游戏是用3D渲染的。当然，你并不一定能够真正地看到非常酷炫的3D效果，这款游戏使用的是上帝视角，所以对所有的polygon进行渲染总觉得浪费时间，所以，你现在可以发现为什么这款游戏非常适合做成VR了。

### **开始使用Cardboard**

你需要做的第一件事就是为Unity下载Cardboard SDK。接下来，把它导入到你的项目中。从Unity的主菜单，选择Assets\Import Package\Custom Package，然后选择你刚刚下载的CardboardSDKForUnity.unitypackage。确保所有东西都被选中，反选Legacy文件夹，然后点击Import按钮。

为了让你的游戏作为一个VR体验，你需要用到一些技巧，在Project Browser中的Cardboard\Prefabs文件夹里，把CardboardMain Prefab拖拽到你的场景中，在Inspector中，给予和你的忍者角色一样的位置（5.53,1.13,0.122），并且要进行90度Y Rotation。

你会发现的是，它比忍者的中心部位略高，这样可以展现出你在看他的眼睛。

接下来，选择主摄像头（Main Camera）并且在Inspector中反选，做raccoon忍者对象的时候也需要同样的操作。现在，把游戏在Unity编辑器中再次运行，你就会看到一些类似于3D场景的东西。如果你在移动鼠标的时候按住选择键，你的摄像头就会随着头部的运动进行旋转。

### 在iOS设备上运行你的游戏场景

在Unity编辑器中运行你的游戏是非常不错的，但最后一次检查的时候，用VR头盔配上电脑显示器看起来会让人很痛苦，所以我们需要适配到iPhone上。

选择File\Build Settings,iOS应该已经被选为你的默认平台，点击Click Player Settings然后转换到Inspector。

在Resolution and Presentation菜单下，把Default Orientation设置成Landscape Left。

在其他设置（Other Settings）中，把Bundle Identifier改成你的公司所需要的东西，比如com.(你的公司名).NinjaAttackVR。

把目标设备改为iPhone，把iPhone和电脑进行连接，选择Build和Run，然后给输出文件夹命名，这个名字你可以随意取。

这时候，Unity就会导出你的项目，然后它就会自动在Xcode中打开，如果没有打开，启动Xcode并且手动打开生成后的项目，运行然后在手机上尝试。首次运行游戏的时候，你需要一系列的设置过程，比如可以在Cardboard硬件上扫描二维码，这样Cardboard SDK就可以根据你的设备、距离等进行画质微调。

需要注意的是，如果设置过程中，你在扫描二维码的时候出现了 网址打开错误，那就必须调整Xcode项目中的info.plist，苹果开发者论坛中有说到这个问题，感兴趣的童鞋可以亲自查阅。

接着，把你的手机插入Cardboard当中，来回转头以调整摄像头视角，你就可以看到相对不错的3D画面了。

### **再一次把它做成游戏**

能够看到你的游戏世界是非常不错的，但初次之外，你还需要把玩法加入到游戏中，特别是你需要从忍者面朝的方向扔出忍者之星，这是你将要做的第一个玩法。

对于UI来说，Cardboard支持一个按钮，看起来可能是比较有限，但如果把它和头部的动态追踪相结合的话，它可以做出更为复杂的互动。在《Ninja Attack》中，你通过Cardboard.SDK.VRModeEnable资源侦测玩家是否开始了VR模式，检查按钮是否和Cardboard.SDK.Triggered资源同时按下去了，如果这些值都对，就可以在用户面朝的方向扔出忍者之星。

打开你的NinjaStarLauncher.cs脚本，你会发现它在Inspector里是和GameLogic GameObject连在一起的。

创建一个新的private变量：

```
private Vector3 _vrShooterOffset;
```

把它在Start() method中初始化：

```
_vrShooterOffset = new Vector3(0.0f, -0.4f, 1.0f);
```

用以下代码替换Update ()：

这就可以运行了，接下来我们看看Update()是做什么的：

你首先检查游戏是否在VR模式，用户是否按下了按钮检查Cardboard.SDK singleton object上的资源。

在此之后，你可以调用LaunchNinjaStarFrom()释放一个忍者之星，你需要用到两个parameter：

第一个是GameObject头文件，Cardboard库会为你调出来，所以它应该是已经指向了正确的位置；第二个是轻微偏移，这样你面前的忍者之星就会看起来更真实，否则的话你扔出去的忍者之星就像是在四只眼之间进行运动，虽然看起来很cool，但给人的感觉很奇怪。

由于你的Ninja Star GameObject已经被设计飞往特定方向，所以它会朝着正确的方向进攻。

再试一次，这时候，你可以转头扔向坏人，输赢logic仍旧起作用。

### 解决Game Over菜单

你可能已经注意到了，当游戏结束的时候，用之前的Game Over按钮，你的角色仍旧是在屏幕的左边。这款游戏用到了Display Canvas来展示Game Over界面，Unity最新的GUI教程中有讲到这一点，它总是出现在游戏窗口的顶部。这个标签适用于大多数的游戏GUI，因为它可以自动适应到你屏幕的上方，不论你的摄像头在做什么，而且它可以非常不错地适应不同屏幕尺寸。

但在这个案例中，你需要一个存在于游戏世界中的GUI canvas，有一部分原因是它可以在3D环境中更好渲染，但还因为不希望把玩家爱是叫锁定到摄像头上。你的玩家们需要可以上下自由地看，这样他们可以看到不同的UI元素，找到最活跃的然后点击按钮。



## 创建一个新的Canvas

在Hierarchy菜单下选择GameOverCanvas，右击并且选择复制，重命名为VRGameOverCanvas，这样可以使它与原来的进行区别开来，把GameOverTxt重命名为VRGameOvertxt。

在VRGameOverCanvas组件中，把渲染模式改为World Space。

在Rect Transform组件中，把位置改为（-2.24,1.1,0.07），然后进行90度Y Rotation

最后，把X和Y Scale改为0.009，当所有一切完成的时候，VRGameOverCanvas看起来因该是这样的：

你可以在Game View视角下看到两个canvas是大致重叠的（当游戏不运行的时候）：

这些值是从哪里来的呢？坦白地说，我其实是调整到自己通过Cardboard摄像头看起来比较不错为止。有时候，编程更多的是一门艺术而不是学科。

### 支持两个Canvas

接下来，你需要更改GameController.cs，这样它才会发现两个Canvas打开和GameLogic GameObject脚本，它也是和GameLogic GameObject联系在一起的。把以下的两个公共变量加入到你的class里：

```
01. public Canvas VRGameOverCanvas;
02. public Text VRGameOverTxt;
```

[复制代码](#)

在resetGame()的开始加入如下代码：

```
01. VRGameOverCanvas.enabled = false;
```

[复制代码](#)

用以下代码取代GameOver():

```
01. public void GameOver(bool didIWin) {
02.     isGameOver = true;
03.     _didIWin = didIWin;
04.     string finalTxt = (_didIWin) ? "You won!" : "Too bad";
05.     if (Cardboard.SDK.VRModeEnabled) {
06.         VRGameOverCanvas.enabled = true;
07.         VRGameOverTxt.text = finalTxt;
08.     } else {
09.         gameOverCanvas.enabled = true;
10.         gameOverTxt.text = finalTxt;
11.     }
12. }
```

[复制代码](#)

这个展示了正确的Canvas和Text对象，取决于你是否在VR模式（开启Cardboard.SDK.VRMode）。

在你保存了脚本之后，你需要把正确的对象分配到新的公用变量。在Inspector里找到GameController，点击每个新变量旁边的目标，然后选择VRGameOverCanvas对象作为你的VR游戏Over Over Canvas变量，把VRGameOverTxt对象选为你的VR Game Over Txt变量。

需要注意的是，可能你会奇怪，为什么是很麻烦地支持两个Canvas而不是至改变现有的一个呢？原因是，你既需要支持上帝视角，还需要支持VR模式，所以一定要进行优化。

如果你准备现在就运行游戏，就可以发现VR模式里的游戏结束界面展示的很自然。你可以上下看界面的不同部分，现在所有缺少的东西就是再来一次的按钮。

### 增加Gaze输入方式

幸运的是，Unity内置了‘在使用world-space GUI Canvas的时候，摄像头中心点可以作为鼠标使用’，但你需要提供额外的脚本才能使它在VR界面中进行使用。

首先，扩展Cardboard Main\Head，找到主摄像头并把它重命名为VR Main Camera。选中VRGameOverCanvas对象，你们应该可以看到一个事件摄像头，点击Hierarchy里的EventSystem项目，点击增加组件按钮，并增加GazeInput Module脚本。这个脚本可以确保Unity的GUI系统了解Cardboard摄像头的工作方式。

检查VR Mode Only，因为在VR模式下的时候，只需要这么运行就可以了。最后，点击你刚刚增加的Gaze Input Module Component，然后选择Move Up，重复一次，确保它可以出现在触屏输入和独立输入模式中，这可以确保Gaze Input Module在游戏进行的同时优先选择输入方式。当所有都做好的时候，它看起来应该是这样的：

现在，你就可以进行尝试了。这一次，当你把视角放到Play Again按钮的时候，它就会变绿，让你重新开始一场新的游戏。

### 玩法微调

或许你会发现这个版本的游戏在VR模式下玩起来有点难，这是因为你的视角是缩水的，所以在你看着错误的方向时，敌人很容易从你身边溜过去。而且，你无法迅速改变瞄准方向，你会因为脖子转速的限制而影响到游戏操作。你让玩家体验VR模式并不是为了惩罚他们，所以，你该如何调整呢？当然，可能会有人建议把敌人速度降低。

在Prefabs文件夹中选中EvilSlimeEnemy Prefab，然后打开EnemyMover.cs，把以下代码增加到Start()，随后设置速度：

这会让你的游戏在VR模式下变得更简单，所以玩家们不至于因为选择了VR模式就获得糟糕体验。

### 解决屏幕上的分数显示问题

你还需要解决的一个UI问题是屏幕上的分数，这个就需要不同的方式来处理了。虽然它仍然需要在VR模式中恰当的被显示出来，但你更希望它在你看任何方向的时候都固定在摄像头上。

选中Cardboard Main\Head，右击并选择UI\Canvas，重命名新的canvas为VRScoreCanvas，把渲染模式调整为World Space，为其赋予以下值：

位置(0, 1, 2.5)、宽度400、高度100、旋转(0, 0, 0)、Scale ( 0.0115, 0.0115, 1 )。当完成之后，游戏看起来该是这样的：

看起来可能你的文本很奇怪地放在了屏幕中央，但在VR模式里，你能够看到的世界比正常情况下是少很多的，所以你在游戏中看到分数的时候应该是在边缘部位的，你可以自由进行位置调整，以使其适应你的手机。

接下来，使用文本对象展示你的分数，这个过程和Game Over的做法类似。

打开GameController.cs并增加一个新的公用变量：

```
01. public Text VRScoreTxt;  
    复制代码
```

接下来，你每次更新scoreTxt的时候都需要更新VRScoreTxt，在ResetGame() method中，把以下代码加在每次更新后的scoreTxt之后：

```
01. VRScoreTxt.text = “-“;
```

[复制代码](#)

然后把这行代码增加到GotOne(), 也放在更新scoreTxt的后面:

```
01. VRScoreTxt.text = "" + _currScore;
```

[复制代码](#)

保存你的脚本, 回到Unity, 然后你会发现GameLogic当中的GameController Component如今可以输入VR Score Txt变量了, 点击临近的目标然后选择你的VRScoreTxt文本对象。

再次体验你的游戏, 现在, 你就可以看到分数出现在左上角了, 还可以允许你的头部进行运动。

### VR模式的切换

由于你的游戏同时支持上帝视角和VR模式, 你应该给用户自由切换的选择, UI做起来是很直观的, 你只需要在上帝视角模式中加入一个简单的按钮, 让玩家们来回切换模式即可。

首先, 你需要增加切换的代码, 选择Hierarchy里的GameLogic, 点击增加组件, 选择新脚本 ( New Script ) 然后把脚本命名为CardboardSwapper。

打开并用以下内容替换class代码:

这个class里最重要的method就是ActiveVRMode，它是用来激活Cardboard的VR模式的。其余的logic负责控制场景中的多个GameObject，取决于是否处于VR模式，你在上帝视角是看不到某些东西的。

你还可以发现的是，当你侦测后面按钮的时候调用了Switch(),这个功能非常适合测试。

你还需要为GameController脚本增加更多的logic，这样它才能在切换模式的时候展示或者隐藏一些东西，打开GameController.cs，把这个method加进去：

```
01.     public void RefreshGameOver() {  
02.  
03.         gameOverCanvas.enabled = false;  
04.  
05.         VRGameOverCanvas.enabled = false;  
06.  
07.         if (isGameOver) {  
08.  
09.             GameOver(_didIWin);  
10.  
11.         }  
12.  
13.     }
```

[复制代码](#)

保存一切然后重新回到Unity界面，选择GameLogic然后向下滚动到Cardboard Swapper组件，对于Cardboard Object数列，把它的大小赋值为1，然后放到场景中的CardboardMain GameObject中。这样不仅可以禁用你的Cardboard Head让你回到上帝视角摄像头模式，还可以禁用VRScoreCanvas。

对于Mono Object数列，把它的大小赋值为3，然后为你的场景选择Canvas、Main Camera和raccoon ninja，不要从Assets里选择。

最后你需要在上帝视角canvas为用户增加一个按钮，为了节约时间，我已经把它做好了，位于prefabs文件夹中。



从Assets\Prefabs中把CardboardButton推拽到Hierarchy，这样它就成为了你的Canvas对象的子对象，确保其位置设定是(-50,50,0):

在你的按钮对象底部，把它们连接起来，这样点击按钮的时候就可以调用CardboardSwapper.Switch() method，你可以从这个动画看看是如何做出来的：

再次尝试你的游戏，点击屏幕右下方的按钮切换至VR模式，然后单机Cardboard界面背后的按钮切换到上帝视角模式。做到这里，你的VR模式切换就已经完成了。

现在，你可以把Unity中的任何3D游戏做成VR游戏了，而且只需要一个Cardboard和一些塑料镜片，这是可以让所有人都能够体验的VR游戏。Android版本的做法和iOS大致相同，谷歌的

Unity Developer指南还提供了更多的技术信息。最后，你甚至可以为你的VR游戏增加AR功能。

最后，你可以尝试Unity里所有的3D游戏，看是否能够顺利地做成VR体验，或者，这个教程也可以让你做出全新的VR游戏。

相关阅读：[Unity5中优化VR 应用的12个技巧](#)

游戏大观编译



扫一扫获取行业消息



扫一扫获取最新产品信息

★ 收藏 1    ▲ 支持 0    ▼ 反对 0

最新评论