

欢迎继续我们的学习。

北京时间2016年1月7日凌晨，Oculus正式开放对首款消费版Rift产品CV1的预订。599刀的价格让很多VR粉丝心头一紧，不过想想第一代iPhone的价格也是这样，只能期待后续的版本降价了。

## 筑基

为了准备好学习使用Unity开发VR应用，我们首先要检查下自己的电脑硬件和软件配置是否满足要求。

简单来说，显卡要NVIDIA GTX790或AMD290以上，CPU要Intel i5-459以上，内存8GB以上，要有两个USB3.0接口，一个USB2.0接口，一个HDMI 1.3接口。

操作系统比较坑爹，不支持Mac，也不支持Linux。支持Win7,Win8,Win10。

当然，还要把自己的显卡驱动升级到最新版本。

Oculus官方的检测地址：

[http://oculus.us5.list-manage.com/track/click?  
u=88dbd06829e35d5cbf84bbc2e&id=b436d0da47&e=86f0296884](http://oculus.us5.list-manage.com/track/click?u=88dbd06829e35d5cbf84bbc2e&id=b436d0da47&e=86f0296884)

关于电脑配置的相关信息，可以参考：

<http://www.cylonspace.com/2016/01/06/oculus-rift%E6%98%8E%E5%A4%A9%E5%BC%80%E5%90%AF%E9%A2%84%E8%AE%A2%EF%BC%8C%E4%BD%A0%E7%9A%84%E7%94%B5%E8%84%91%E9%85%8D%E7%BD%AE%E5%A4%9F%E4%B8%8D%EF%BC%9F/>

一旦完成了这种烧钱的工作，就该安装Unity了。注意在打开Unity前要把DK2连接好并开启。在继续之前，打开Oculus Configuration Utility应用，并检查Demo Scene可以正常运行。注意在运行Demo Scene之前看可能需要在Oculus Configuration Utility中设置一个新用户。

## 创建第一个VR项目

接下来我们将使用Unity创建一个简单的VR项目demo，其效果是在VR头盔中观察一个立方体。如果你想研究更多VR示例，不妨下载我们在上一篇教程中提到的VR示例项目（<https://www.assetstore.unity3d.com/#/content/51519>）。

### Step 1.

打开Unity，创建一个新的空项目。

说明一下，我当前用的Unity版本是5.3.1f1，可能你看到这篇教程的时候又已经升级换代了。

### Step 2.

在Unity的菜单中选择File- Build Settings，并选中PC,Mac& Linux Standalone

### Step 3.

在场景中创建一个新的立方体，从菜单中选择Game Object - 3D Object -Cube，通过Translate工具把立方体放到默认的Main Camera 前面，类似下面。

Step 4.保存界面（File- Save Scene，或是使用快捷键）。

Step 5.在菜单中选择Edit- Project Settings - Player，在”Other Settings“部分勾选”Virtual Reality Supported”

Step 6.点击Unity界面上的Play按钮进入Play模式。

如果之前的设置没有问题，现在你应该可以通过DK2看到这个场景，四处看看，Unity中的摄像机将自动根据DK2的位置和旋转变化的反应。

### 出错了怎么办？

如果你没有在DK2中看到期望的场景，那么检查以下的事项：

- 1.确保你在打开Unity项目前就已经把DK2接好并且开启了。
- 2.打开Oculus自带的Oculus Configuration Utility，看看Demo Scene是不是可以正常运作
- 3.更新你的显卡驱动到最新版本
- 4.确保你在电脑上装了最新的Oculus Runtime 0.8，或更高版本。

当然，如果还是有问题，可以到论坛里面参与讨论。<http://forum.unity3d.com/forums/virtual-reality.80/>

关于VR开发的一些有用信息：

虽然VR应用开发和标准Unity应用开发很相似，不过还是有些区别需要注意的。

#### 1.编辑器中显示的帧速(Frame rate)

当你通过编辑器查看项目时，注意体验上可能有些延迟，因为电脑需要将同样的内容渲染两次。所以在实际测试项目的时候，最好创建一个可执行版本，在测试设备上实际体验。

#### 2.摄像机的移动

注意，我们不能在Unity中直接移动VR摄像机。如果你希望调整摄像机的位置和旋转，需要确保将它设置为其它游戏物体GameObject的子物体，然后通过所依附的物体进行移动。

关于这一点，可以查看VR Samples项目中的Flyer和Maze场景。

### 3.相机节点

左眼和右眼的摄像机并不是由Unity创建的。如果你在开发中需要获取这些节点的位置，则必须使用InputTracking类。

如果你想要获取场景中眼睛的不同位置（比如测试时），请使用下面的示例脚本，并将其attach到摄像机上。

#### C#脚本

```
using UnityEngine;
using UnityEngine.VR;

public class UpdateEyeAnchors : MonoBehaviour
{
    GameObject[] eyes = new GameObject[2];
    string[] eyeAnchorNames = { "LeftEyeAnchor", "RightEyeAnchor" };
    void Update()
    {
        for (int i = 0; i < 2; ++i)
        {
            // If the eye anchor is no longer a child of us, don't use it
            if (eyes[i] != null && eyes[i].transform.parent != transform)
            {
                eyes[i] = null;
            }

            // If we don't have an eye anchor, try to find one or create one
            if (eyes[i] == null)
            {
                Transform t = transform.Find(eyeAnchorNames[i]);
                if (t)
                {
                    eyes[i] = t.gameObject;
                }

                if (eyes[i] == null)
                {
                    eyes[i] = new GameObject(eyeAnchorNames[i]);
                    eyes[i].transform.parent = gameObject.transform;
                }
            }

            // Update the eye transform
            eyes[i].transform.localPosition = InputTracking.GetLocalPosition((VRNode)i);
            eyes[i].transform.localRotation = InputTracking.GetLocalRotation((VRNode)i);
        }
    }
}
```

```
}
```

#### 4.VR中的图像效果(Image Effect)

在VR项目中使用众多图像效果是很奢侈的事情。考虑到你需要两次渲染同一个场景（每只眼睛一次），因此很多当前经常使用的图像效果对VR应用来说会很浪费，会严重影响游戏的运行帧速。

因为VR将用户的眼睛置入一个虚拟的空间，因此部分图像效果对VR来说没有任何意义。举例来说，深度视角、模糊效应和镜头光晕效果对VR来说没有任何意义，因为在现实世界中我们是看不到这些效果的。不过如果以后VR头戴设备可以支持眼动跟踪，那么深度视角可能会有意义。

不过有些效果还是可以考虑使用的：比如抗锯齿是有用的（特别是考虑到某些头戴设备的低分辨率），色彩分级也很有用（关于这一点可以查看这个链接：<http://blogs.unity3d.com/2015/05/12/color-grading-with-unity-and-the-asset-store/>），对有些游戏来说Bloom会有用。不过在使用任何效果之前，最好先在游戏中实际测试下有没有用。

Unity自身提供了很多图像效果（Assets-Import Package-Effects），此外Asset Store里面也提供了很多效果，比如Colorful,Chromatica,Amplify Color，还有更多。

#### 5.Render Scale

根据VR互动应用场景的复杂度和所运行的硬件环境，你可能需要更改render scale设置。通过此项设置可以调整镜头校正前texel:pixel的比率，这样可以牺牲游戏性能来换取画面的清晰度。

这项设置应该通过代码进行，可以参考这里：  
<http://unity3d.com/cn/VRSettings.renderScale>

通过使用以下代码可以改变render scale的设置：

```
using UnityEngine;
using System.Collections;
using UnityEngine.VR;

namespace VRStandardAssets.Examples
{
    public class ExampleRenderScale : MonoBehaviour
    {
        [SerializeField] private float m_RenderScale = 1f;

        //The render scale. Higher numbers = better quality, but trades performance

        void Start ()
        {
            VRSettings.renderScale = m_RenderScale;
        }
    }
}
```

```
}
```

关于这项设置，可以参考我们的VR Samples，具体示例是Scenes/Examples/RenderScale场景。此外在MainMenu场景中也有该项设置的应用。

更改render scale的效果示例如下：

Unity的默认render scale是1.0，效果如下：

如果将render scale设置成1.5，可以看到显示效果更锐利：

接下来把renderscale设置成0.5，可以看到像素化很严重：

根据游戏场景的不同，可以考虑降低render scale来提升游戏运行性能，或者通过增加render scale的数值让画面效果更锐利，但会以牺牲游戏运行性能作为代价。

好了，看到这里，你应该知道如何在Unity项目中整合VR，如何设置游戏中摄像机的移动，以及和非VR游戏相比应该如何使用图像效果。

在下一篇教程中，我们将学习如何跟场景中的物体进行交互。

对VR开发感兴趣的朋友可以通过邮件([eseedo@gmail.com](mailto:eseedo@gmail.com))或微信(iseedo)联系我，希望跟大家一起学习。另外在我的网站(<http://www.cylonspace.com>)和博客(<http://blog.sina.com.cn/eseedo>)上也会放VR/AR开发的相关内容。