

# unity3d 开发 VR 第二讲---开始开发 VR 程序

作者：VR 指南

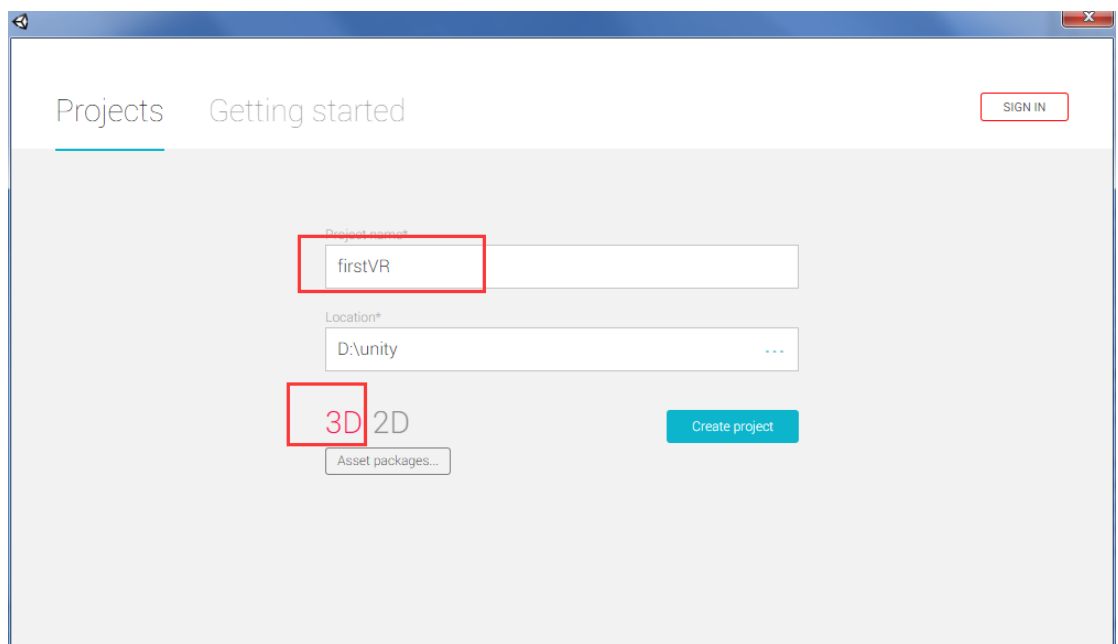
接着 unity3d 开发 VR 概述开始，我们今天实际编写一个小程序让大家对 unity3d 对于 VR 开发有个直观的认识。废话不多说，我们现在正如正题。

## 1、先决条件

首先 unity3d 的版本必须等于或者高于 5.3 版本，DK2 同电脑相连，并按照《概述》中要求的那样进行了配置。

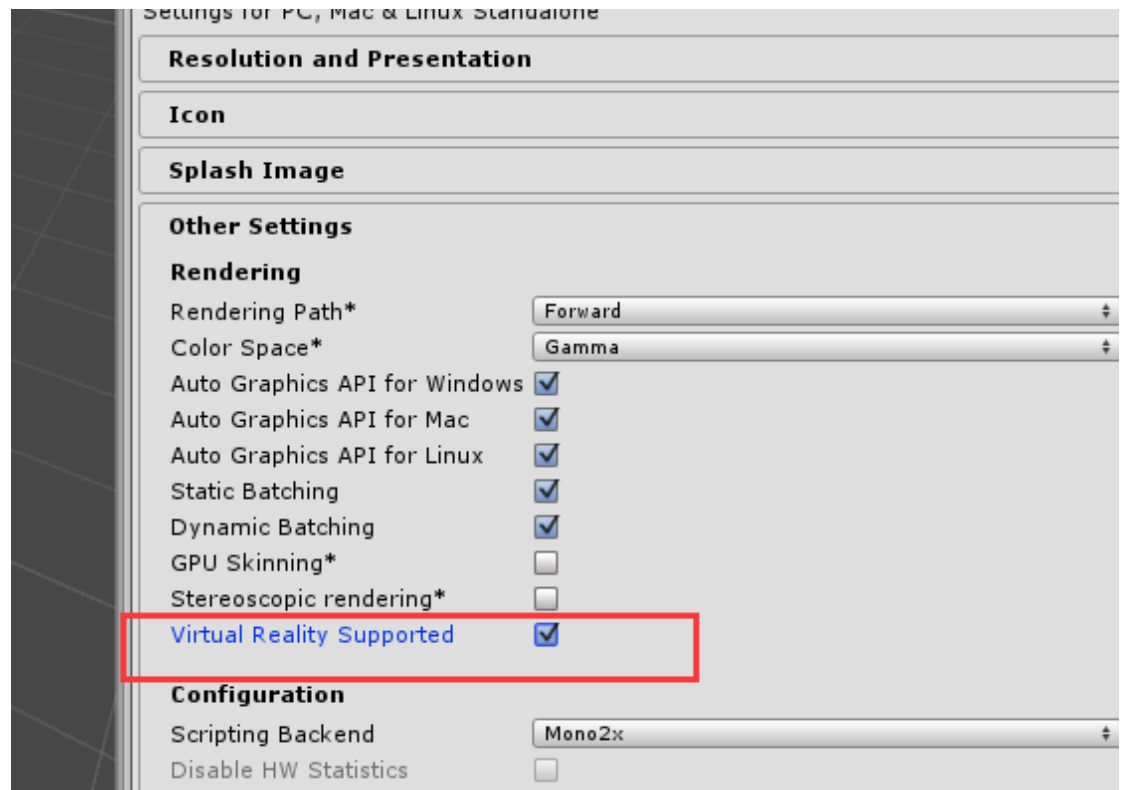
## 2、创建你的第一个 VR 例子

a) 第一步 创建一个空的项目 如下图，项目名称您随意啦

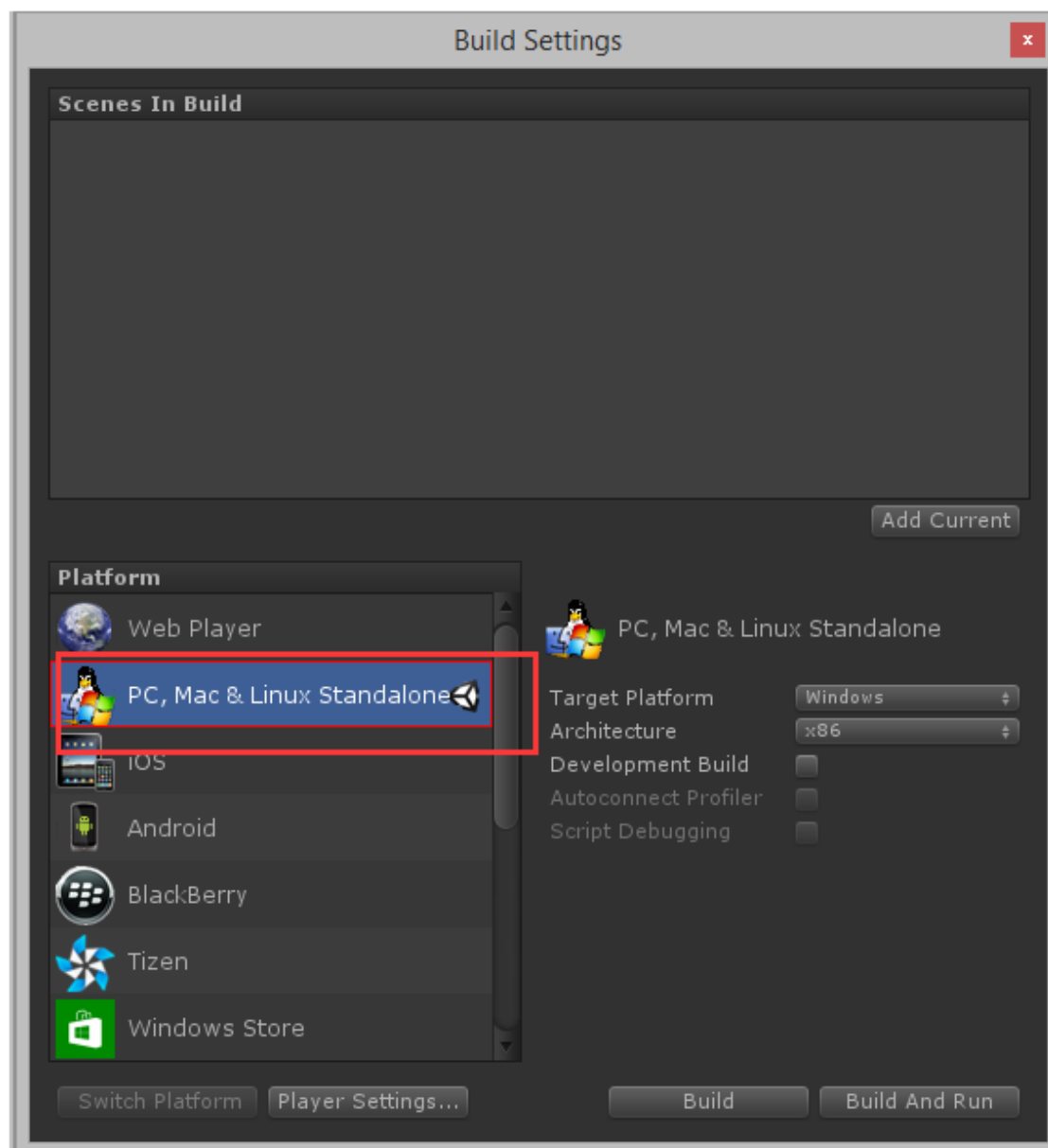


b) 配置环境。

i. 不要忘记勾选支持 vr 设备啊！《概述》里面都有进行了讲解。



- ii. 配置运行参数。需要选择 PC,Mac &linux Standalone



- iii. 创建一个 cube(Game Object >3D Object >Cube)，并把这个 cube 放置到摄像机的范围内进行显示。
  - iv. 保存你的场景，命名为 firstDemo(File>save Scene)
  - v. 点击运行进入运行模式。
  - vi. 这个时候你头戴的 DK2 里面就应该呈现出来场景了。
- 3、开发过程中的注意事项。

尽管上面的例子跟你在使用 unity 进行游戏开发的时候很相似，但是很是有的一些不一样的地方。

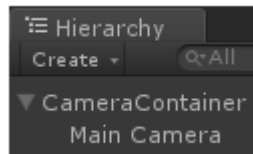
a) 编辑器中的帧率不同

当你在 editor 中查看你的项目的时候，由于需要绘制双倍的内容，所以会出现一定的抖动和延迟，所以你应该在一个功能开发完成后在进行试验。这样可以减少不必要的不适，比如眩晕。恶心等。

b) 摄像机移动不同

你不能直接的移动摄像机，如果你希望移动或者旋转摄像机，你需要确保这个摄像

机是挂载在一个 `GameObject` 下的子物体，通过改变父物体 `GameObject` 来改变 `Camera` 的位置和旋转。例如，`CameraContainer` 就是创建的一个空物体，把 `Main Camera` 当成他的子物体。



c) 相机节点不同

Unity 在开发 VR 程序的时候会自动的创建左眼摄像机和右眼摄像机，如果你想获得这两个节点。你必须使用 `InputTracking` 类。

如果你想获得人所在的位置，可以参考下面的这里脚本例子，并把它挂载到你的摄像机上。

```
using UnityEngine;

using UnityEngine.VR; //引入 VR 模块

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);

    }

}

}
```

d) 在 VR 中的图像效果不同

在你创建虚拟现实项目的时候很多的图像特效都很消耗内存，尤其是在你的 VR 设备上需要左右两边同时进行图像渲染的时候，这个开销就跟昂贵了，所以尽量在目前硬件条件下优化项目资源。

在一个虚拟空间中 VR 虚拟出来用户的眼睛，一些图像效果就变得没有意义了。比

如，景深，模糊，和镜头眩光一般没有任何意义，因为他们在现实生活中人眼根本感受不到这些变化。如果 VR HMD(Head Mounted Displays)最终支持眼球追踪，景深倒是可以被感应到。至于什么是景深（Depth of Field）参考 VR 指南官网提供的《VR 专属名词系列讲解》教程之 Depth of Field 景深解析。

<http://www.openmooc.net/forum.php?mod=viewthread&tid=41>

而对于一些效果可以使用：抗混叠(Anti-aliasing)特效可以使用(尤其对于分辨率较低的 HMD 设备),颜色分级可以使用，Bloom(中文翻译为开花，不知道专业术语叫什么，详情请看 VR 指南提供的《VR 专属名词讲解》书籍)可以使用。所以这里对开发者和场景设计师提出一点小小的建议，不断的尝试特效是不是在 VR 中体现是唯一的途径。不要偷懒!!!

Unity 也提供了一些图像特效你可以使用，在(Assets> import Package >effects)可以导入一些特效 在 Asset Store 中也包含有很多的图像特效，比如 Colorful, Chromatica, Amplify Color(这里我就不翻译了，大家可以复制这些名词在 assets store 中查找并使用。

#### e) 渲染级别不同

根据您的虚拟现实场景的复杂性和您的硬件配置，使用相应的渲染级别。这种控制像素：像素比之前的镜头校正，这意味着我们的贸易表现锐度。

这必须通过代码，并通过 <http://docs.unity3d.com/scriptreference/vr.vrsettings-renderscale.html>。这可以很容易地改变使用下面的脚本：

```
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;
        }
    }
}
```

当我们将渲染级别调整到 1.0 的时候看到的图像样子



如果我们增加到 1.5，你可以看到资源看起来更清晰：



最后，如果我们降低 render scale 0.5，你可以看到现在更多的像素：



这一讲你已经学会了如何创建一个项目，相机的使用，图像效果的不同，虽然很浅显，但是对于入门来说已经足够了，我们要循序渐进，踏实前行，这样才能掌握好 VR 开发，不要试图认为自己是个天才，一天都能学会这些 VR 开发的知识，要知道这是多少人通过多少年才研究出来的东西，你要是一天就学会那些牛人岂不是没有面子？？

好了，预告一下下一讲，下一讲我们主要讲解，虚拟现实的交互部分，这些也是 VR 中的重中之重。