

在设计VR应用的用户界面时要考虑很多因素，而这些因素对于传统应用或游戏的界面设计可能都不是事儿。下面我们会看看作为一个VR开发者在设计用户界面时可能会遇到的一些问题，以及跟硬件相关的一些东西。

UI分辨率和画质

目前DK2的分辨率是1920*1080(单目是960*1080)，而Gear VR是2560*1440（单目是1280*1440），因此对任何在宽度或高度上要占据几个像素的物体，都会出现比较明显的像素化现象。

其实特别要注意的是UI元素，记住它们在屏幕上要显示的大小。一个简单的原则是尽量使用大一点的粗体字，另外就是尽量不要使用在VR场景中容易变得像素化的细纹。

UI的类型

Non-diegetic（非剧情型UI）

在传统的非VR项目中，UI通常显示在界面的顶部，用来显示生命值、得分之类的信息，通常被称为HUD。这就是所谓的非剧情型UI-用户界面跟游戏世界没有什么关联度，但是对玩游戏的玩家有一定的作用。

在电影中也有所谓的“画外音”，比如电影或电视中的背景音乐。与此不同的是，剧情音通常跟故事紧密相连-比如角色之间的对话，或是实际发生的自然音效。

在Unity中，添加HUD样式的非剧情型UI相对简单，只需要在UI Canvas的Render Mode中选择使用Screen Space- Overlay或是Screen Space- Camera。

但这种UI界面对VR基本不适用，我们的眼镜无法聚焦在如此近的物体上，而Unity VR中根本就不支持Screen Space-Overlay。

Spatial UI（空间UI）

和前一种UI不同的是，我们需要将UI放置到环境之中，并在Canvas的Render Mode中选择使用World Space模式。通过这种方式，就可以让用户的眼镜聚焦到UI上。这就是所谓的Spatial UI（空间UI）。

将UI元素放在世界的哪个位置也需要认真考虑。太靠近用户会导致眼部疲劳，离得太远会感觉聚焦在地平线上-这种情况可能发生在室外环境，而不是在小屋子里。此外我们还需要对UI的比例进行适当调整，具体要根据产品的实际需求来定。

我们最好把UI放在一个舒适的可读距离，并进行相应的缩放。关于这一点可以参考Main Menu场景中的UI：它被放置在几米远的位置，因此图片和文字都比较大，看起来很舒服。

当我们把UI放在离用户有一定距离的时候，可能会发现UI重叠到其它物体上。关于这一点可以查看上一篇教程，看看如何修改shader让其绘制在其它对象上，或者直接使用VR Samples项目中的shader。这个shader也适用于文本。

很多开发者想要把UI关联到摄像机上，这样当玩家移动的时候，UI会保持在一个固定的位置。这样做对十字准星或其它小物体可能会比较合适，不过对于比较大的UI元素，就好比把一张报纸放到你的脸上，很容易让用户感到不舒服，甚至是眩晕。我们可以看一看Shooter 360(Target Arena) 场景中的UI，其中UI会在一个较短的延迟后移动到视野中，从而可以让用户四处查看进而熟悉身边的环境，而不是直接把UI固定到他们的视野中，导致视觉模糊。

虽然VR可以让用户探索360度沉浸式环境，不过有时候我们需要提醒用户关注某个特定的方向。在某些场景中，我们使用世界中的箭头来帮助引导用户的注意力。这些箭头会根据用户的朝向来淡入淡出。

这些功能可以在GUIArrows 预设中找到，而且很容易重用。其工作原理是比较用户头部和目标方向之间的角度。如果头部在预设角度之外（参考下面GUIArrows组件的Show Angle），箭头就会淡入。当用户向目标的方向注视时，箭头就会开始淡出。

Diegetic UI（剧情型UI）

Spatial UI的替代方案是让环境中的物体自行向用户展示信息。具体的形式可能是墙上的闹钟，电视、计算机屏幕、移动手机，或是未来枪械的全息展示。这就是所谓的剧情型UI。

下图展示了Flyer场景中的太空飞船，以及Shooter(Target) 场景中的枪支。

虽然这种交互不算严格意义上的剧情型UI，但是将用户界面跟游戏对象关联在一起，可以让我们大概了解Unity的剧情型UI是如何工作的。

关于UI的延伸阅读信息

关于UI类型的更详细分析（不涉及到VR），可以参考这篇文章。

UI交互

通过使用Interaction in VR一文中所提到的VREyeRaycaster, VRInput和VRInteractableItem，我们可以创建基本的UI交互元素，具体方法是创建一个订阅VRInteractableItem事件的类。

关于这一点的更多信息，可以参考Interaction in VR，特别是Maze场景中的switch。此外，在VRSamples中每个游戏的开始，我们都使用了UI交互指导用户。

关于如何在VR中使用Unity UI，可以参考Unity's UI System in VR这篇文章，其中还提供了示例代码。

VR Samples场景中的UI

接下来让我们看看以上所提到的概念在VR Samples场景中的具体实现：

菜单

Menu场景中的UI使用了定制的纹理来创建一个弯曲的密封环境。可以使用Interaction in VR一文中的相同方法跟这些纹理进行交互。

Flyer

游戏简介和游戏结束的UI使用静态的方式防止在世界空间中。

不过为了呈现跟游戏相关的信息，我们可以使用跟太空飞船关联的世界空间UI，也就是剧情型UI。

考虑到飞船必须时刻出现在用户的视野中，因此在飞船四周显示相关信息是靠谱的。

与此同时，UI元素会旋转以面向摄像机，这样就会避免倾斜，同时确保UI的显示对用户友好。

Maze

在Maze 场景中，我们使用Spatial UI来提供游戏引导和结束。

Spatial UI还可以用来提示用户进行交互：

Shooter 180(Target Gallery)

在游戏引导和结束部分，我们再次使用了静态的Spatial UI.

正如上面所提到的，我们在枪支上使用剧情型UI来显示倒计时和得分。

Shooter 360(Target Arena)

在这个场景中依然使用了Spatial UI,不过做了一些小的调整：当360度竞技场中发生了某个动作，比如当用户四处查看环境时，我们会让UI元素在经过一个短时间的延迟后移动到视野中间。通过这种方式可以让用户知道四处探查所处的环境。

在枪支上我们再次使用了剧情型UI。

VR中的文本抗锯齿

为了在VR世界中实现免费的文本抗锯齿效果，这里有一个小技巧，在Worldspace Canva中使用Canvas Scaler。在相关设置中，我们需要将”Reference Pixels Per Unit”设置为1，然后调整”Dynamic Pixels Per Unit”，直到文本的边缘柔化。在下图中，我们可以看到不同设置的实际显示效果。

看完本教程，我们对不同的用户界面应该有了基本的概念，也知道在VR中应该使用哪种UI。当然，我们还可以使用VREyeRaycaster,VRInput和VRInteractableItem来创建基本的UI交互。

关于VR UI的更多信息，可以查看官方博客的Unity’s UI System in VR。

在下一篇教程汇总，我们将了解Movement in VR的信息。

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