使用 Unity 开发一个简单的 HoloLens 应用

文/张昌伟

HoloLens 概述

在经历数个月的期待与等待后,终于拿到了预订的 HoloLens 开发者版本套件。作为市面上第一款发售的 AR/MR 设备,HoloLens 开发者版本具有很多独特的黑科技。今天,我们就来了解下 HoloLens 的开发特性。

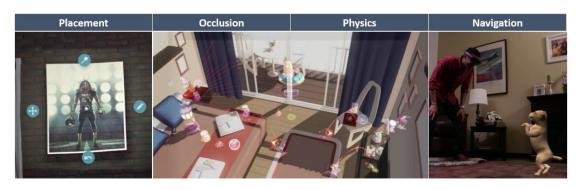


图 1 Hololens 应用场景

1. 空间映射

借助微软特殊定制的全息处理单元 (HPU), HoloLens 实现了对周边环境的快速扫描和空间 匹配。这保证了 HoloLens 能够准确地在真实世界表面放置或展现全息图形内容,确保了核 心的 AR 体验。

2. 场景匹配

HoloLens 设备能存储并识别环境信息,恢复和保持不同场景中的全息图像对象。当你离开当前房间再回来时,会发现原有放置的全息图像均会在正确的位置出现。

3. **自然交互**

HoloLens 主要交互方式为凝视 (Gaze) 语音 (Voice Command) 和手势 (Gesture), 这构成了 HoloLens 的基本输入要素。同时传统的键盘鼠标等设备也被支持,自然的交互方式更

贴近人类习惯,提高了交互效率。

4. 通用应用

HoloLens 平台的操作系统为 Windows Holograpic,同样基于 Windows 10 定制。所以 Windows 10 UWP 通用应用程序可以顺利地在 HoloLens 上运行。这不仅降低了研发和迁移 成本,也让开发效率能够大幅提升。

当然,说了很多 HoloLens 独特的特性和优点后,HoloLens 开发者版本也存在一些亟待解决的问题,包括视野较窄、凝视体验不佳、抗光线干扰弱和重量续航等问题。但瑕不掩瑜,HoloLens 带来了真正的混合现实体验,拥有着强烈的冲击感,未来将大有作为。

开发一个 HoloLens 应用

在了解 HoloLens 设备后,我们来试着开发一个简单的 HoloLens 应用。当然你可以开发一个传统的 UWP 应用,但是我们这里则采用 Unity 引擎来构建 HoloLens 应用,使用 Unity 开发是官方推荐的做法。

开始之前

在开始之前,我们要确保正确配置了开发环境,需安装以下工具和SDK:

- 1. Visual Studio 2015 Update 1 及以上版本;
- 2. Windows 10 SDK 10586 及以上版本;
- 3. HoloLens 模拟器,如图 2;
- 4. Unity HoloLens 技术预览版。

以上工具和 SDK 均可在微软官方网址获取,详细教程可以访问: https://developer.microsoft.com/en-us/windows/holographic/install_the_tools。



图 2 HoloLens 模拟器

集成 HoloToolkit-Unity 项目

在创建了标准 Unity 项目之后,我们需要集成微软官方提供的 HoloToolkit-Unity 项目。
HoloToolkit-Unity 项目是微软官方的开源项目,用于帮助开发者快速开发 HoloLens 应用,
能够快速为项目集成基本输入、空间映射和场景匹配等特性。以下是此项目的结构和内容分析,如图 3:

- Input 目录
 - ➤ GazeManager.cs 用于快速集成凝视射线特性;
 - ➤ GestureManager.cs 用于快速集成手势识别特性;
 - > KeywordManager.cs 用于快速集成语音命令特性;
 - > CursorManager.cs 用于快速集成可视化凝视组件。
- Sharing 目录
 - > Sharing Prefab 组件用于快速集成场景共享特性。

- SpatialMapping 目录
 - > SurfacePlane Prefab 组件用于描述和渲染真实世界表面;
 - > SpatialMapping Prefab 组件用于快速集成空间映射特性;
 - ▶ RemoteMapping Prefab 组件用于快速集成远程空间映射信息导入特性;
- SpatialSound 目录
 - ▶ UAudioManager.cs 用于快速集成空间声音特性。
- Utilities 目录
 - ▶ Billboard.cs 用于实现跟随用户视线特性;
 - > Tagalong.cs 用于实现跟随用户移动特性;
 - Main Camera Prefab 组件用于快速集成 HoloLens 标准主摄像机。

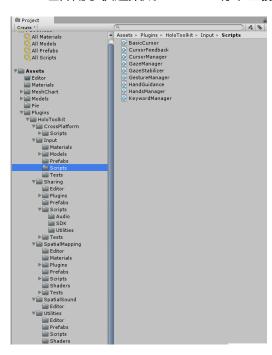


图 3 HoloToolkit-Unity 项目结构

构建场景

新建空白场景后,我们需要删除原有的 Main Camera 对象,同时从 HoloToolkit 目录中拖拽

一个 Main Camera Prefab 组件到场景中,如图 4,这样就集成了满足 HoloLens 需求的基本主摄像机。对于 HoloLens,将主摄像机渲染背景设为纯色,颜色设为 RGBA(0,0,0,0)。因为任何纯黑的颜色将会被 HoloLens 渲染为透明,以达到不遮挡现实世界的目的。此外,HoloLens 建议摄像机视角近距离为 0.85,这个距离最符合真实人眼的体验。同时主摄像机位置必须重置为世界零点,即 xyz(0,0,0,0),任何全息图像将会以此为原点在周边世界中绘制出来。

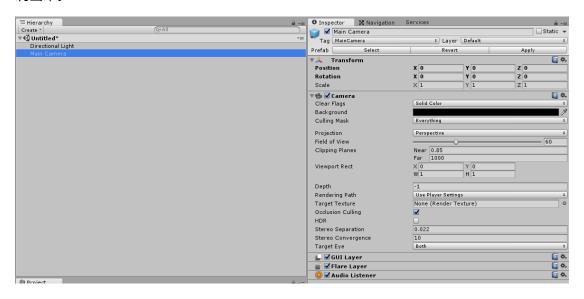


图 4 设置主摄像头

然后点击 "Create Empty" 创建一个空游戏对象,并将其命名为 Input ,如图 5。为 Input 对象添加核心脚本组件,分别为 GazeManager.cs、GestureManager.cs、HandsManager.cs 和 KeywordManager.cs。这样就集成了凝视、手势和语音命令三大核心特性。对于凝视射线、手势识别和语音命令功能,均建议使用单例来进行管理,这样可以避免功能混乱,方便管理。同时为凝视设置可视化的指针,可以提高用户的交互体验和效率。

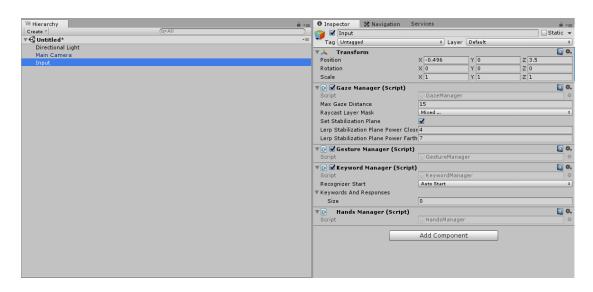


图 5 集成输入组件

接下来集成可视化凝视组件,从 HoloToolkit 目录下拖拽 CursorWithFeedback Prefab 组件 到场景中,如图 6。这样当凝视在全息对象时,其表面会出现可视化凝视组件。当用户手 势被识别到时,会出现一个蓝色的手掌图像,能够贴心的告诉用户可以操作了。

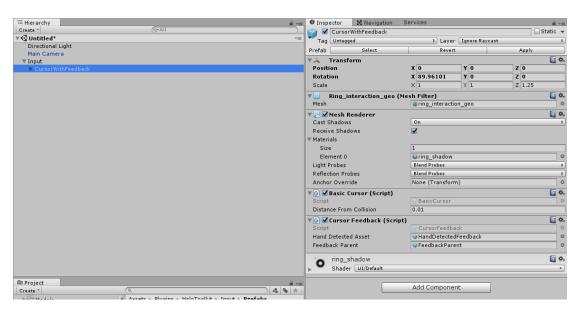


图 6 集成凝视组件

创建一个 Cube 对象并创建一个新的 C#脚本,命名为 HoloTest.cs。 Cube 作为我们的全息 图像主体,将它的 Transform 参数设为如图 7 所示。这样 Cube 的位置方便我们近距离观 察其实际变化情况,你也可以根据自己偏好来放置它。

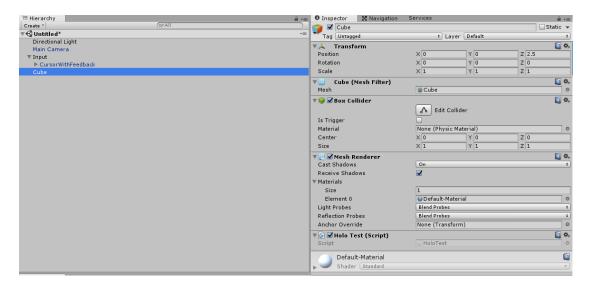


图 7 设置 Cube 的 Transform 参数

HoloTest.cs 脚本的功能为随机更换对象的材质颜色,遵循 GestureManager.cs 中预设的

OnSelect 消息名称, HoloTest.cs 脚本中将会在 OnSelect 方法中实现此功能代码如下:

```
public void OnSelect()
{
    //随机变换物体颜色
    gameObject.GetComponent<MeshRenderer>().material.color = new
Color(Random.Range(0, 255) / 255f, Random.Range(0, 255) / 255f, Random.Range(0, 255) / 255f);
}
```

进入 Input 组件检视选项卡,为 KeywordManager.cs 组件配置语音命令。图 8 语音命令触发时将会执行相应的组件行为。本例中,当我说出"test"时,机会触发 Cube 的 OnS elect 方法,来随机改变 Cube 颜色。

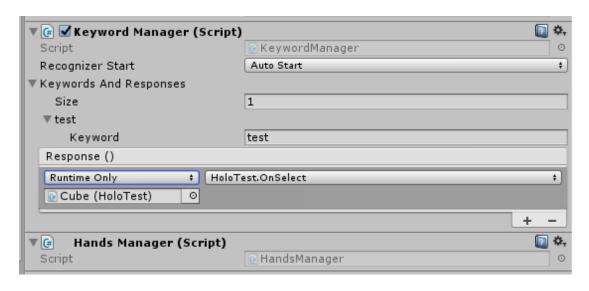


图 8 设定语音关键词行为

编译项目

为了满足 HoloLens 的需求,我们需要在 Player Settings 里面开启 Virtual Reality Support,并在下拉列表中选中 Windows Holographic,如图 9。只有这样 HoloLens 才会将此应用渲染为 3D 应用,这一点十分关键。

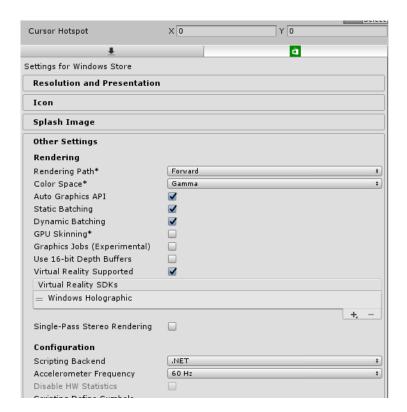


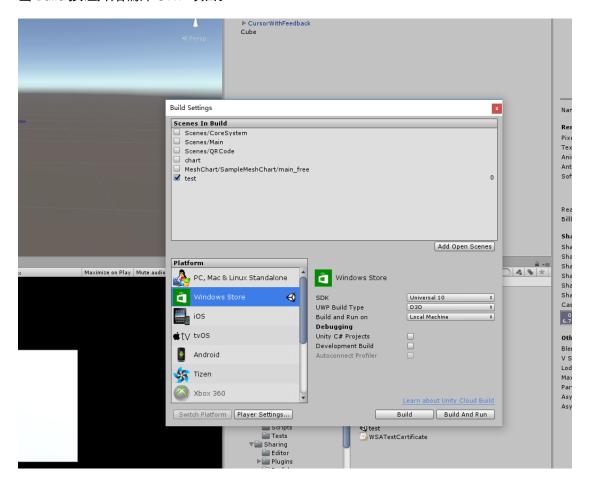
图 9 添加 HoloLens 支持

同时从工具栏 Edit->Project Settings->Quality 选项卡中,将 UWP 平台默认画质设为 Fastest,如图 10。这是为了降低性能开销,官方推荐帧率为 60fps。



图 10 设定默认画质

如图 11, Build Settings 视图中选择目标平台为 Windows Store, SDK 为 Universal 10,点击 Build 按钮开始编译 UWP 项目。



部署调试应用

使用 Visual Studio 打开编译后的 UWP 项目,在 Debug 选项上设置如图 12 所示。

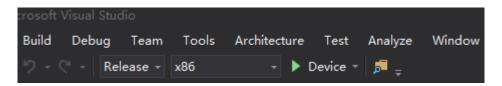


图 12 设置 Debug 选项

连接 HoloLens 到 PC,完成 Build 和 Deploy 后,我们在 HoloLens 中打开此应用。实际效果如图 13 所示。当我使用手势点击 Cube 时,它会随机变化颜色;而当我说出语音命令"test"时,Cube 仍会正常的变换颜色,这完全符合我们的预期。

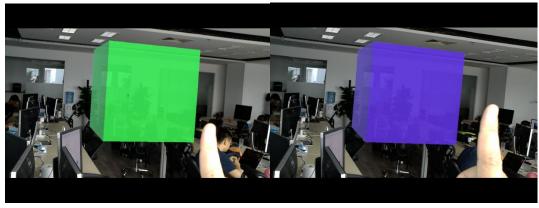


图 13 实际效果图

HoloLens 开发总结

使用 Unity 引擎开发 HoloLens 应用是非常容易的事情,大部分流程与开发 UWP 项目并无不同。但仍有不少需要注意的雷区和特殊要求,以下就是部分要注意的部分:

- 1. Main Camera 一定要按照官方要求配置,背景纯色且 RGBA 值为 (0,0,0,0), 这样才能避免遮挡现实内容;
- 2. Gaze 凝视特性需要我们使用 Raycast 来实现,注意处理射线未命中目标情形,默认凝

视最远距离为15米,若是未击中物体,使用时可能会出现空引用异常;

- 3. 手势识别、拍照和语音命令等均需使用 Windows 特有 API, 空间映射和场景匹配需要使用 HoloLens 特有 API;
- 4. 其他很多细节上的体验,例如可视化凝视组件、目标区域可视化指引组件,使用他们来给用户提示,可以帮助用户理解应用操作方法,提高使用体验。

最后,AR/MR 技术独特的交互体验与开发特性,代表了未来自然交互的发展方向,相较于目前成熟的 VR 技术,它们具有更光明的发展前景和更广阔的用途。无论是微软还是 Magic Leap,它们无疑会是未来市场的引领者,而目前也是我们学习的黄金阶段,能够迎头赶上这波浪潮,对于相关从业者具有重要的意义。