在把硬件、系统、软件、开发工具都准备好了之后，接下来就正式进入HoloLens的开发了。

虽然微软官方提供的Holograms 100教程（<https://developer.microsoft.com/en-us/windows/holographic/holograms_100> ）中详细说明了最基本的配置流程，不过既然是用Unity，那么第一时间想到的还是插件。很幸运的是微软官方提供了一个HoloToolKit-Unity插件，可以方便快捷的进行一些基本设置，而不需要用Holograms100教程里面那么繁琐的设置。

首先在浏览器中进入HoloToolkit-Unity的界面，<https://github.com/Microsoft/HoloToolkit-Unity>

可以看到在README.MD中对这款插件做了一个非常简单的介绍。一句话来说，HoloToolKit是一个插件集合，其中包含了各种脚本和组件，以方便开发者高效开发基于Windows Holographic（实际上就是HoloLens）的全息应用。

HoloToolKit中包含了以下的特性：

1. Input（输入）
2. Sharing（分享）
3. Spatial Mapping（环境空间映射）
4. Spatial Sound（空间环绕音效）
5. Utilities（工具）
6. Build（编译）

如果想了解每个特性的具体内容，可以详细参考下面的链接：

<https://github.com/Microsoft/HoloToolkit-Unity/blob/master/Assets/HoloToolkit/Input/README.md>

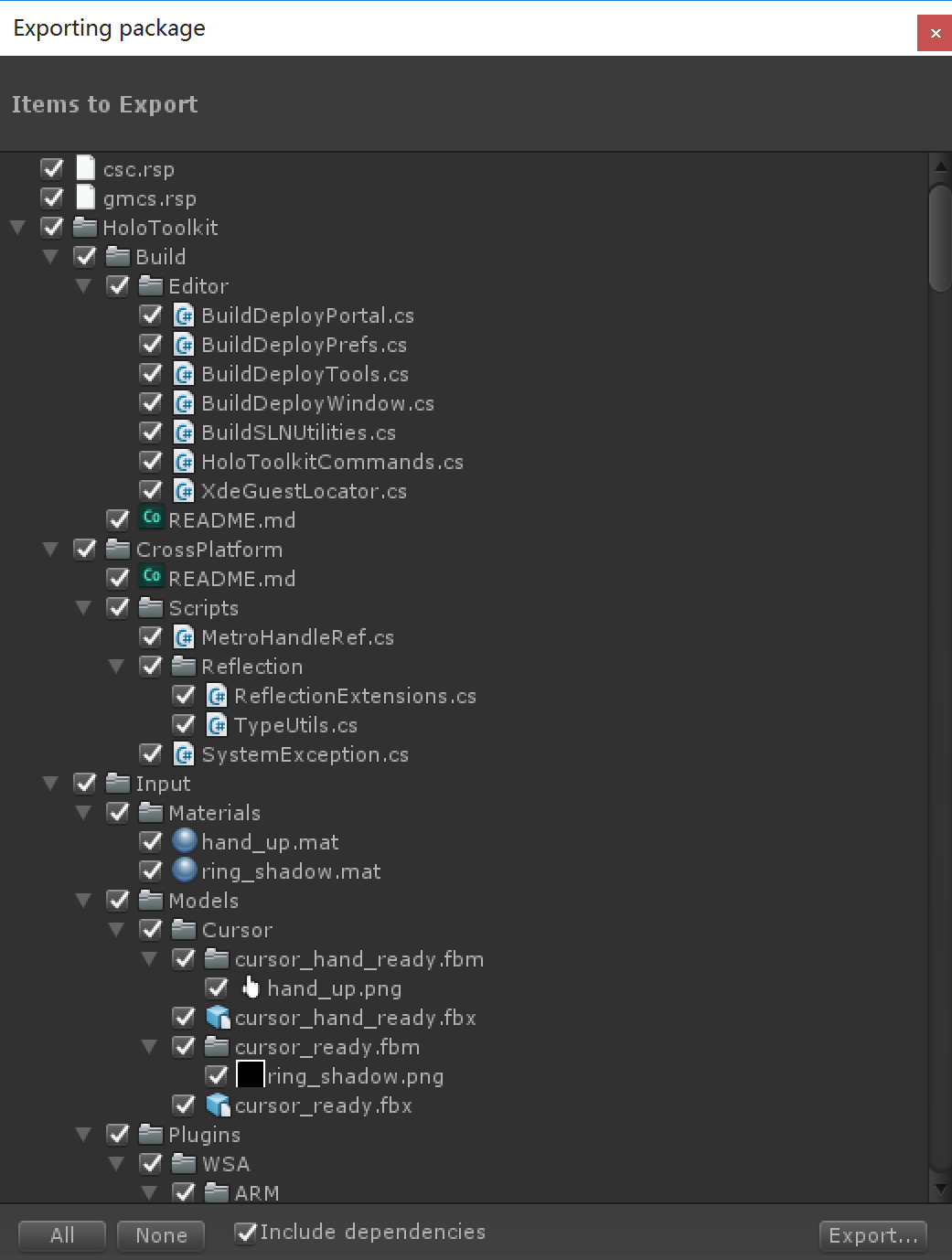
然后打开SourceTree这种基于Git的源代码管理工具软件，把相关的插件源代码Clone到本地的指定目录下。

有些童鞋可能习惯用Github官方的客户端，不过个人还是推荐使用SourceTree。

**准备好使用HoloToolKit-Unity的插件包**

在Unity中打开刚刚下载的文件夹，在Unity中选中整个Assets目录，然后导出package。需要特别注意的是，一定要选中Project面板中的Assets根目录。其中包含了重要的.rsp文件，比如csc,gmcs和smcs文件。

Assets -> Export Package…



把导出的文件命名为HololensToolKitUnity.unitypackage，当然后缀名是自动生成的。

在项目中使用HoloToolKit-Unity插件

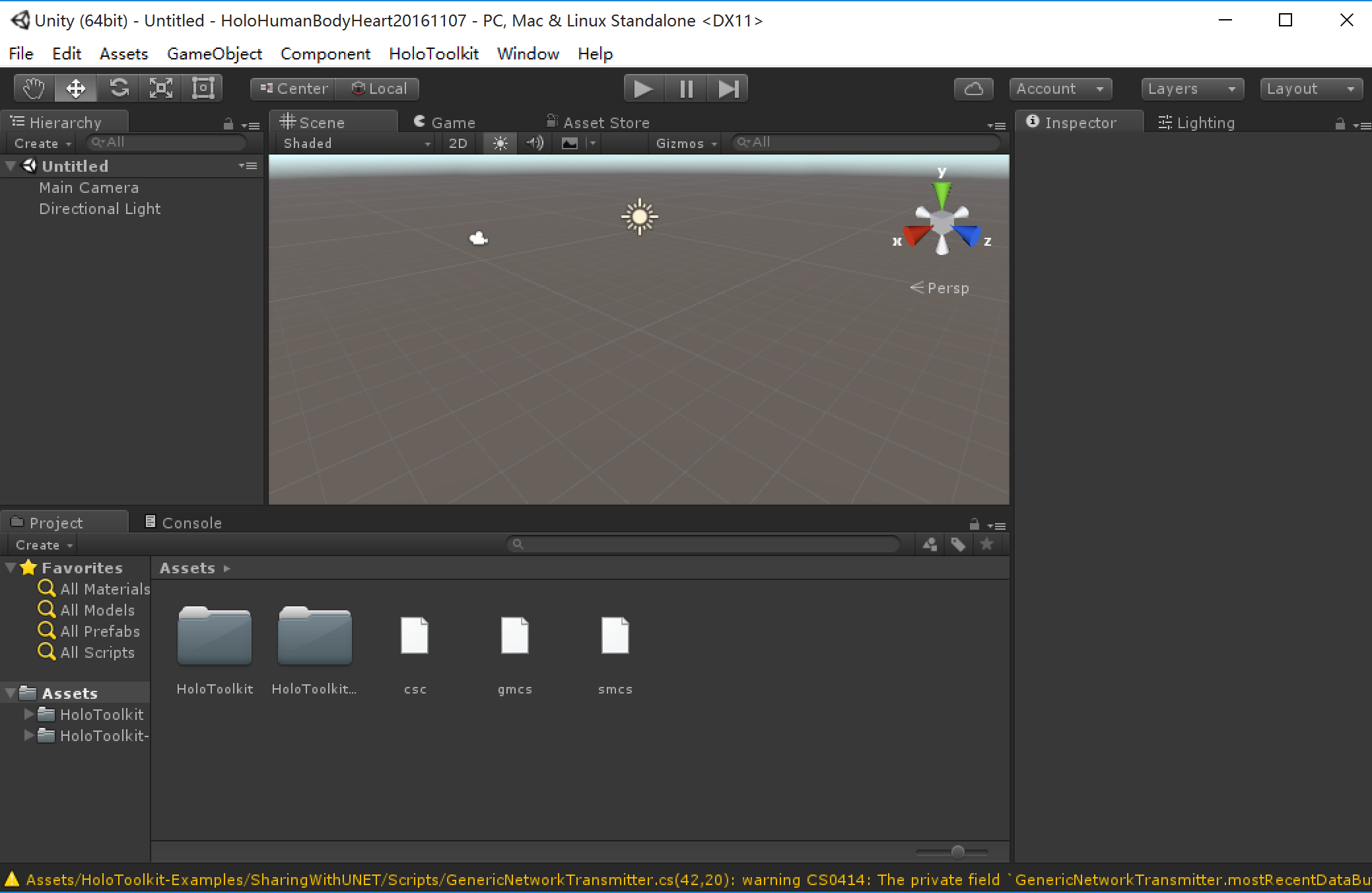
有了这个插件，接下来我们就可以轻松愉快的在Unity中开发全息应用了。

在Unity中创建一个新项目（也可以打开之前的项目），然后导入这个插件包。

Assets -> Import Package -> Custom Package…，选择刚刚导出的插件包。

需要注意的是：在导入的过程中，HoloToolKit Examples文件夹是否导入时可选的。只需要在Import Unity Package窗口取消选择相关的文件夹即可。

完成这步操作后，在Unity顶部的菜单栏上将会看到一个HoloToolKit菜单选项。



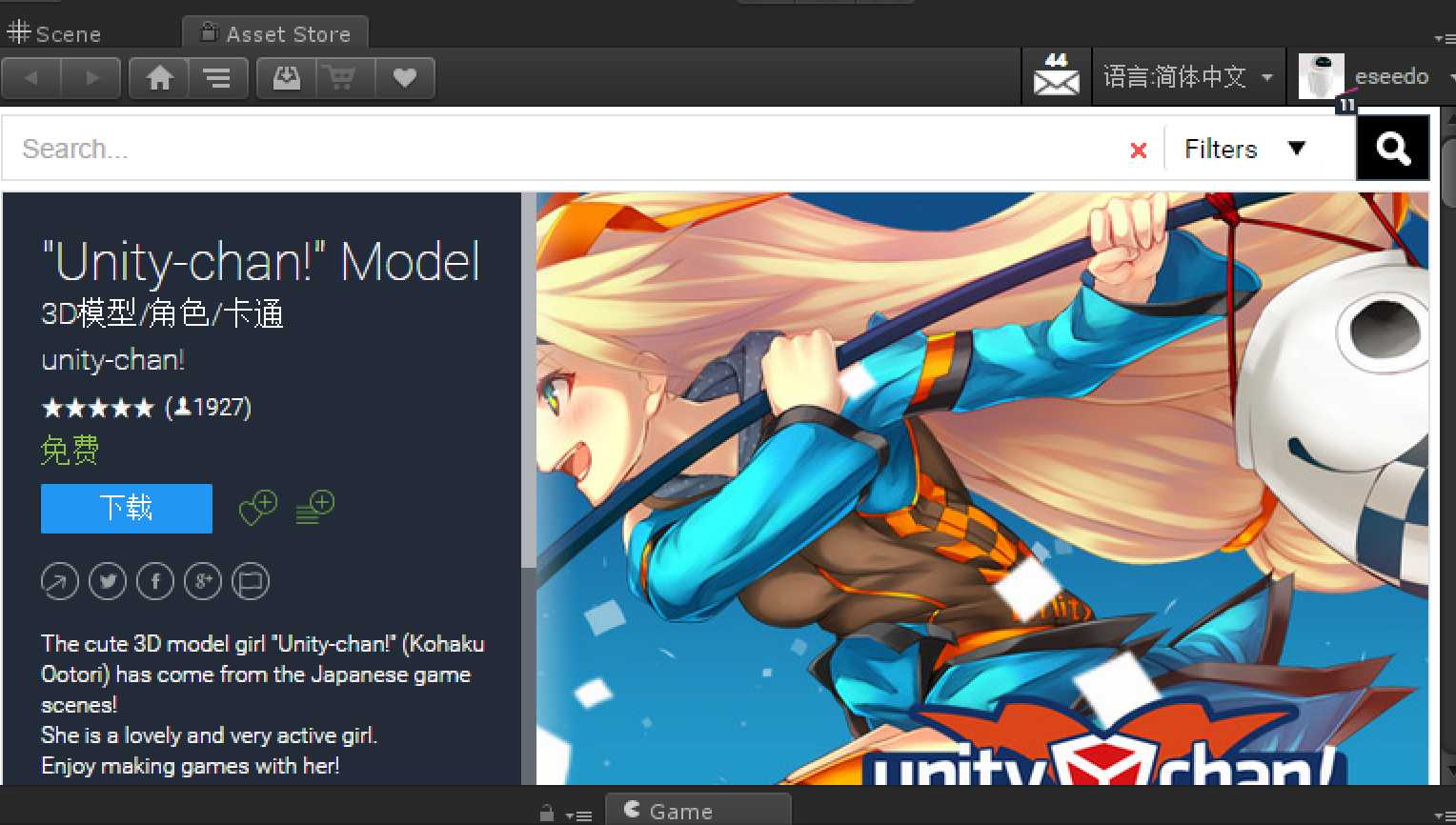
创建一个全息内容场景

首先保存当前的临时场景，命名为HoloTest。

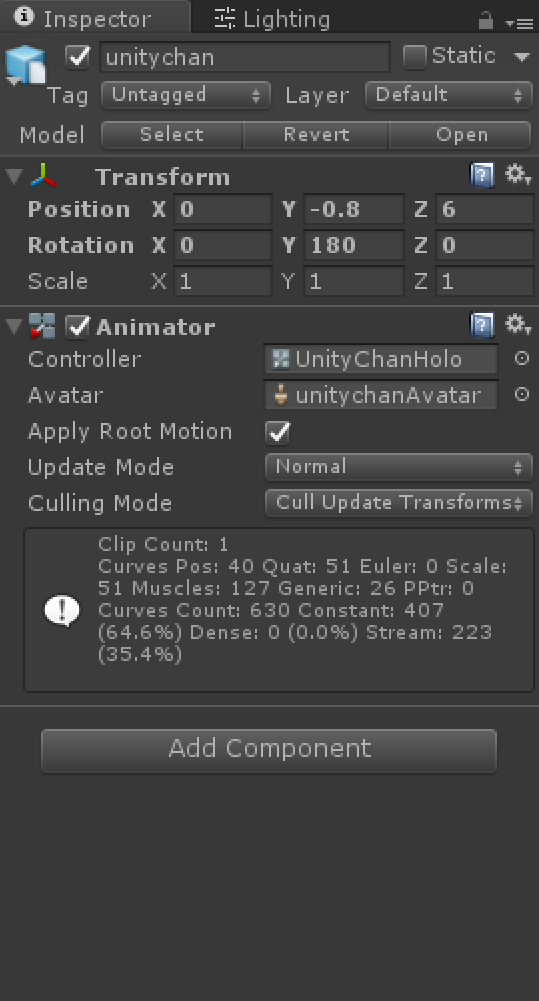
将场景中的默认Main Camera删除，然后将Main Camera.prefab拖到Hierarchy中。（在HoloToolkit/Utilities/Prefabs中）

接下来从HoloToolkit/Utilities/Scripts中找到ManualCameraControl.cs脚本，把它添加到Main Camera上。这样用户就可以在Unity player中手动控制摄像机了。

当然，现在场景是空的，为了展示的需要，我们添加一个模型进去。在菜单栏中选择Window-Asset Store，然后搜索Unity-chan,然后下载并导入这个非常酷炫而又免费的人物角色。

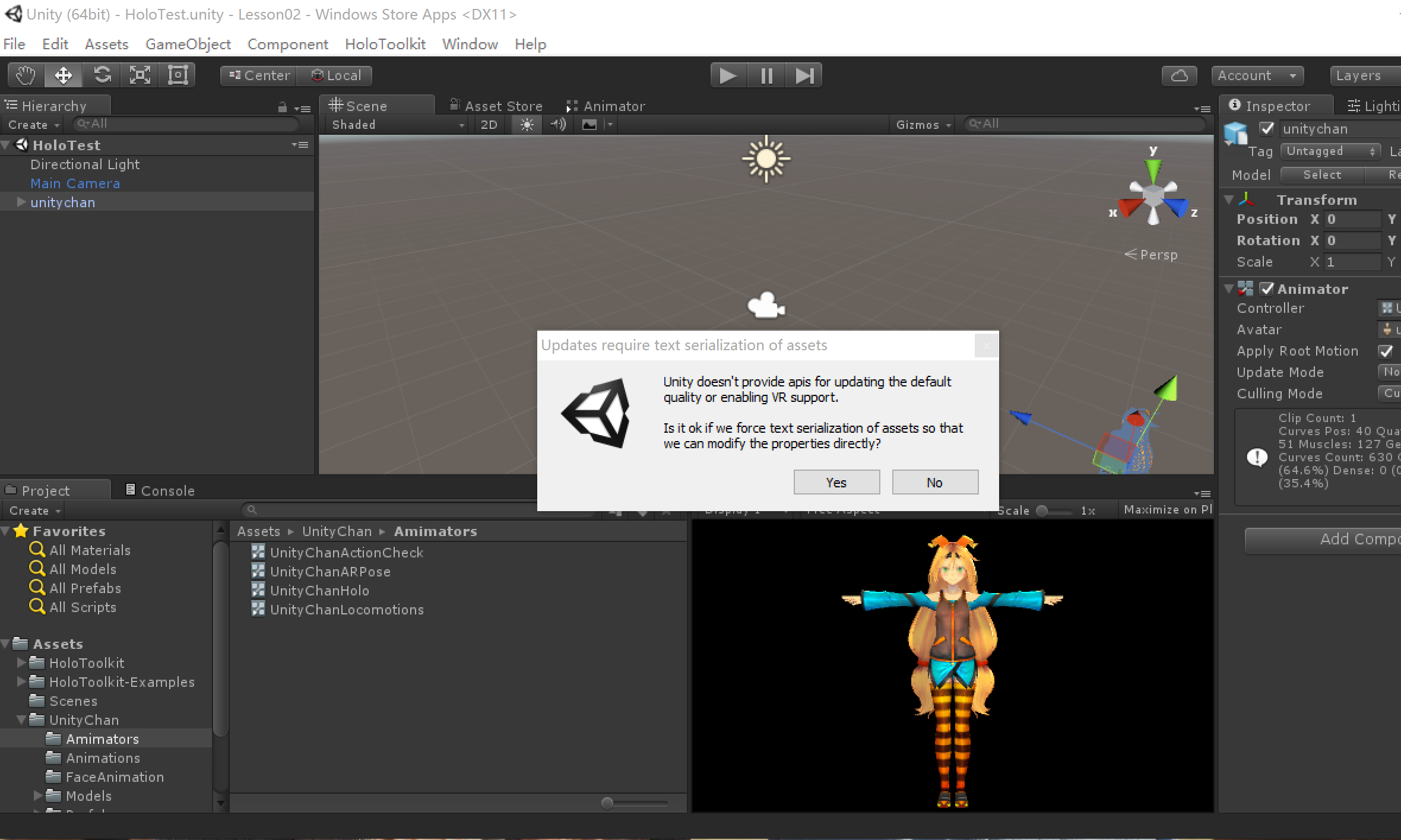


然后根据摄像机的显示，调整角色的Transform信息如图。

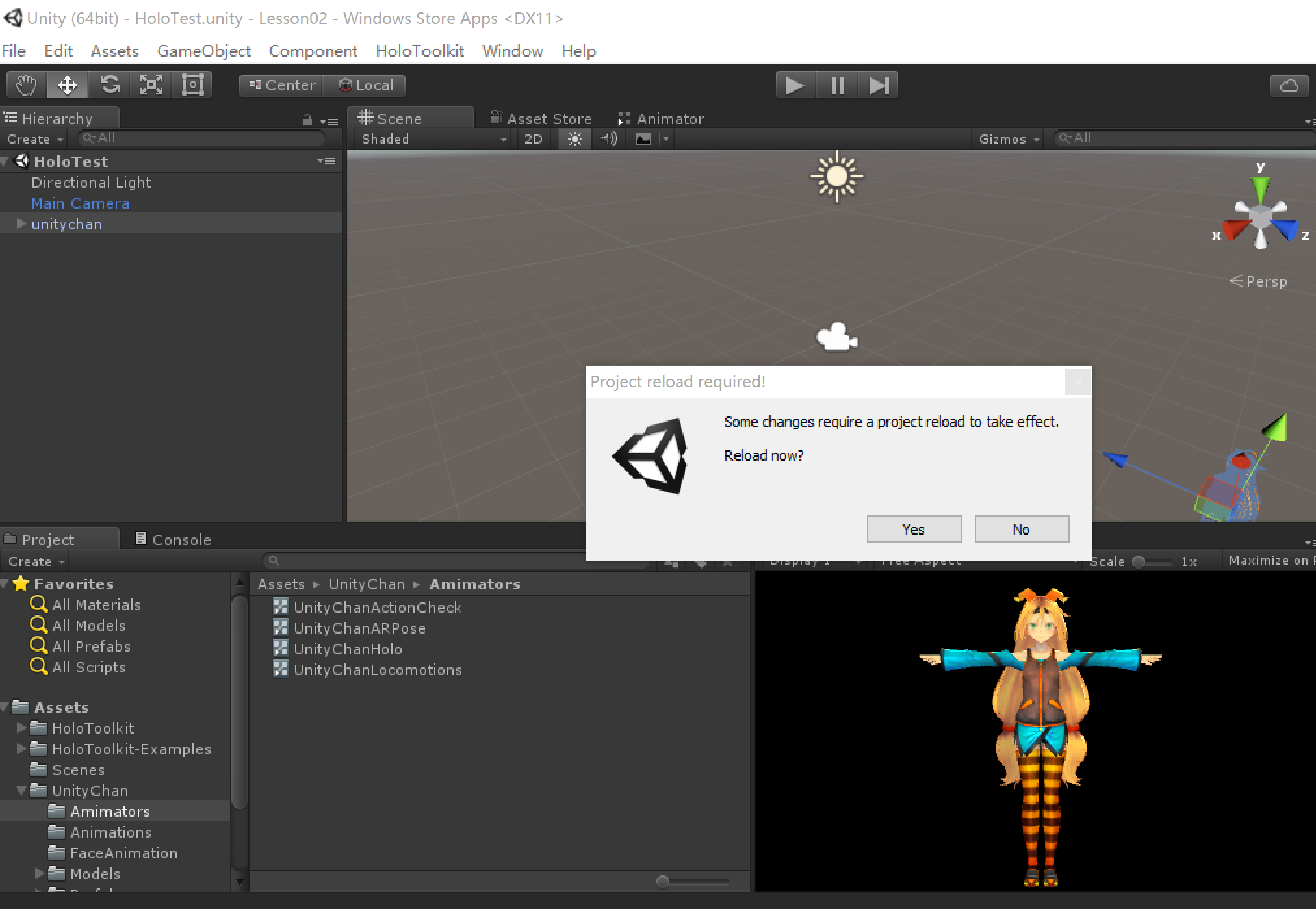


在Unity的菜单栏上选中HoloToolkit -> Configure -> Apply HoloLens Scene Settings，然后再选择HoloToolkit -> Configure -> Apply HoloLens Project Settings

此时可能会出现如下图的提示，选择Yes就好了。



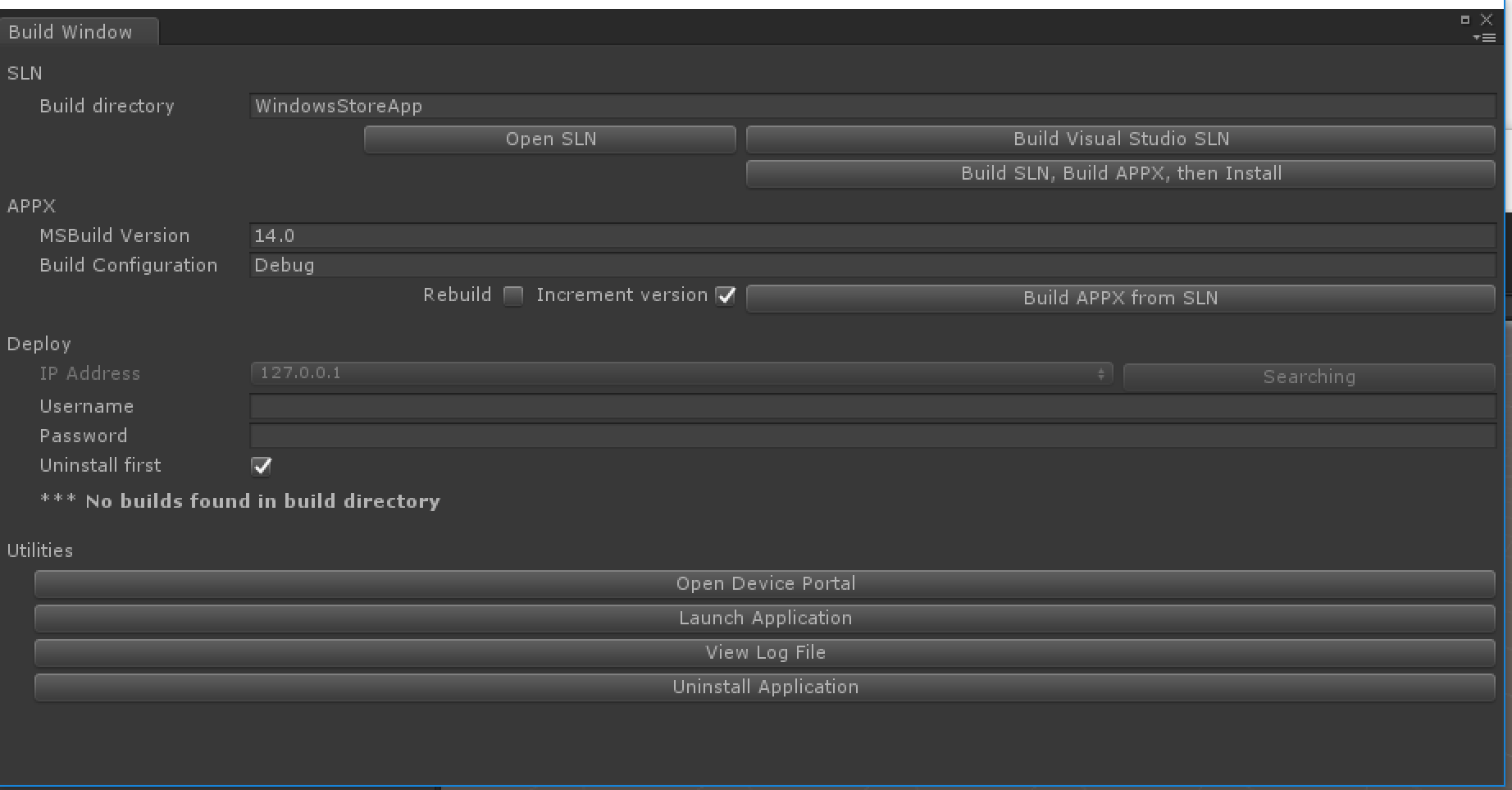
紧接着又会看到一个提示，依然是选择Yes。Unity会退出并重新打开项目。



编译项目

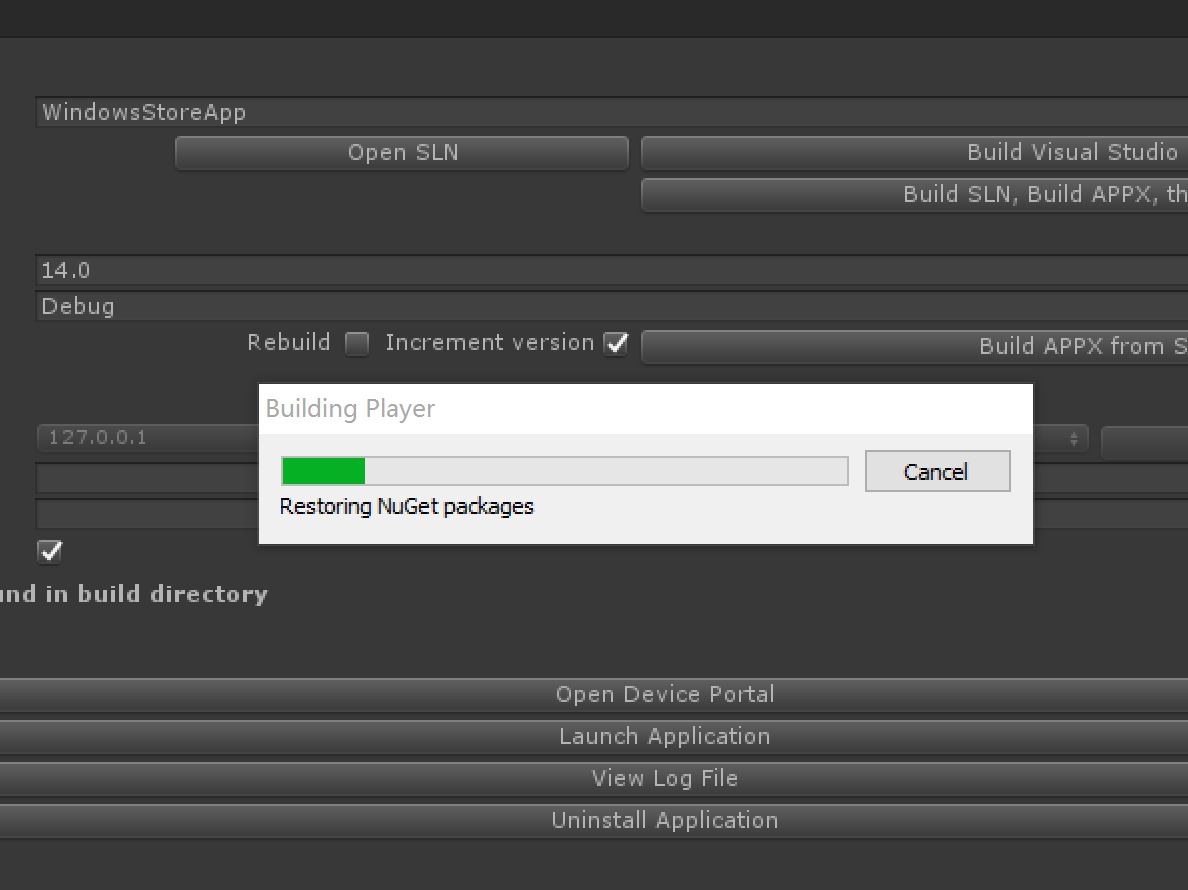
Unity重新打开项目后，在菜单栏上选择HoloToolkit -> Build Window

会看到以下提示：

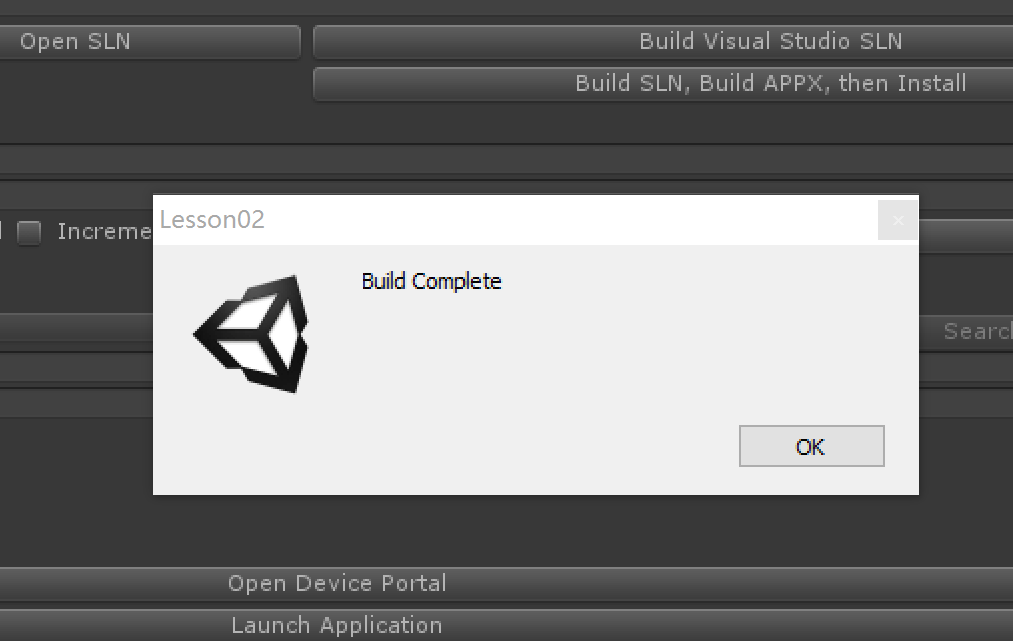


点击Build Visual Studio SLN.

然后就是漫长的等待，时间取决于你所用电脑的性能~



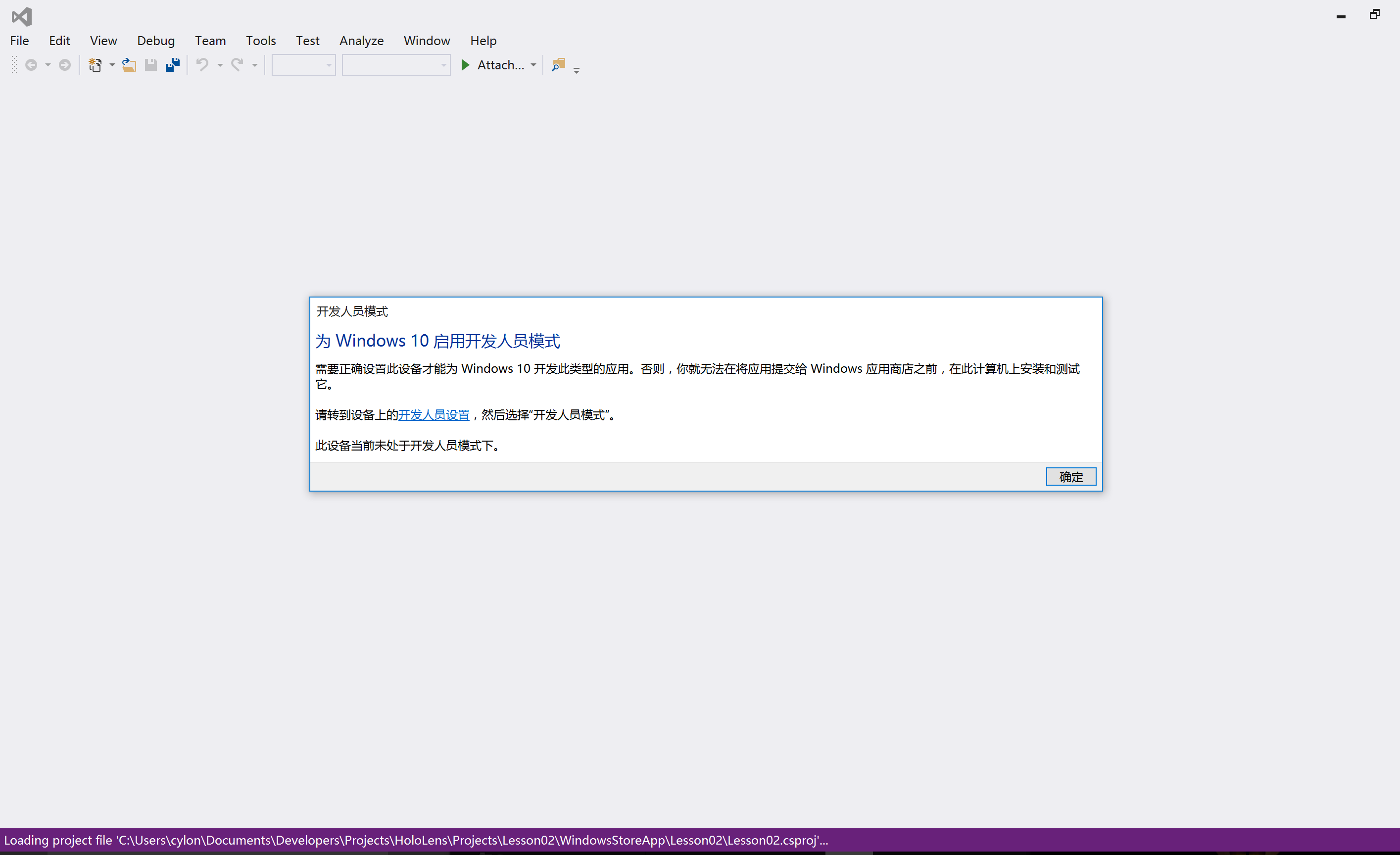
完成后会看到下面的对话框，



点OK确认。

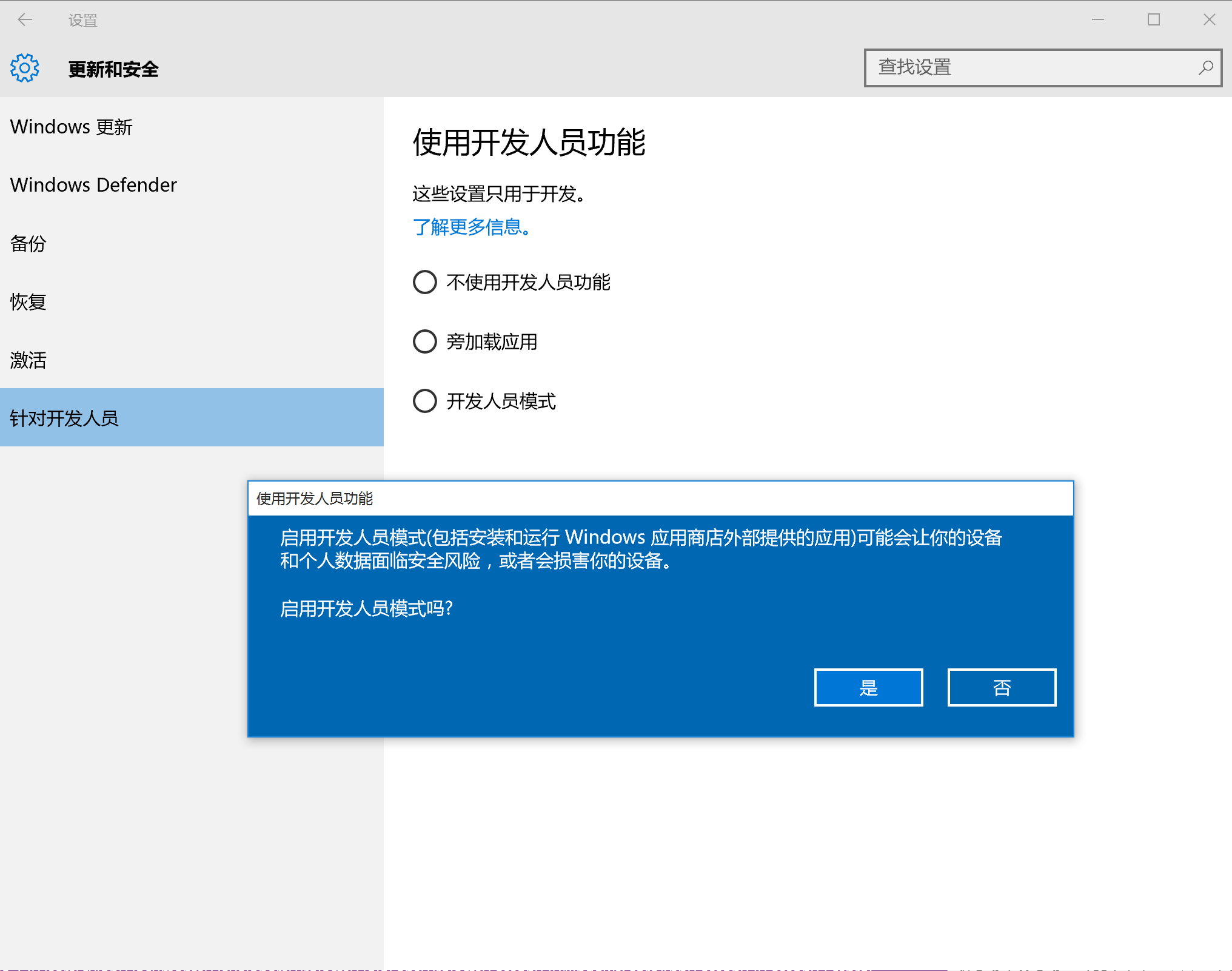
接下来点击Open SLN，会打开Visual Studio.

此时你可能会看到一个提示，让你启用开发人员模式。



这个没啥好说的，点击上面的“开发人员设置”链接，进入Win10的设置界面。

点击开发人员模式，会提示是否启用。



显然这是万年不变的微软啰嗦风格，就这点上跟Mac没法比。

没的说，选择“是”。然后关闭设置。

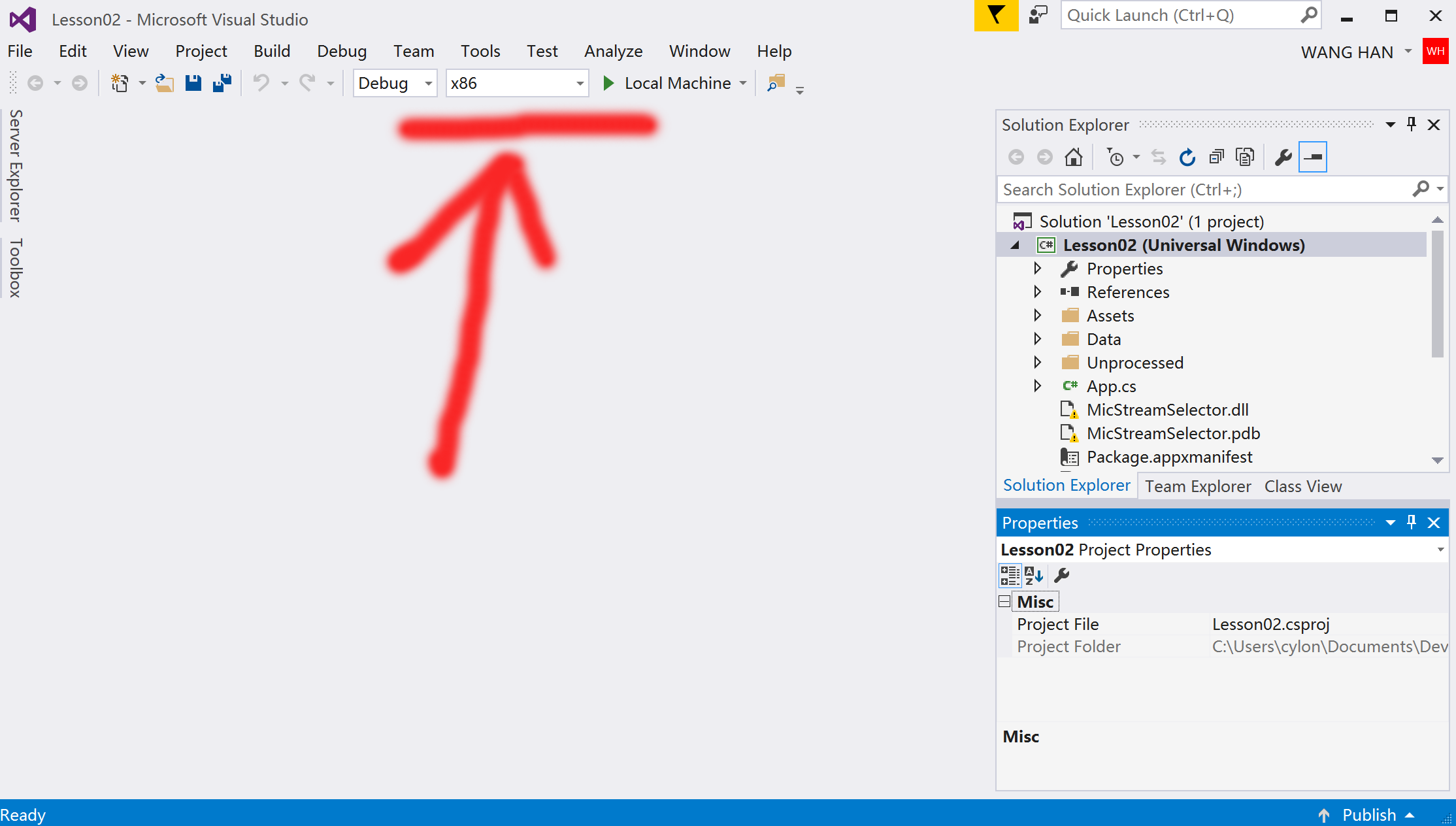
回到Visual Studio,点击确定。

VS会自动帮你把刚才的Unity全息项目载入。

**在HoloLens设备上部署全息应用**

总共分三步

Step 1 在VS的build设置中选中x86

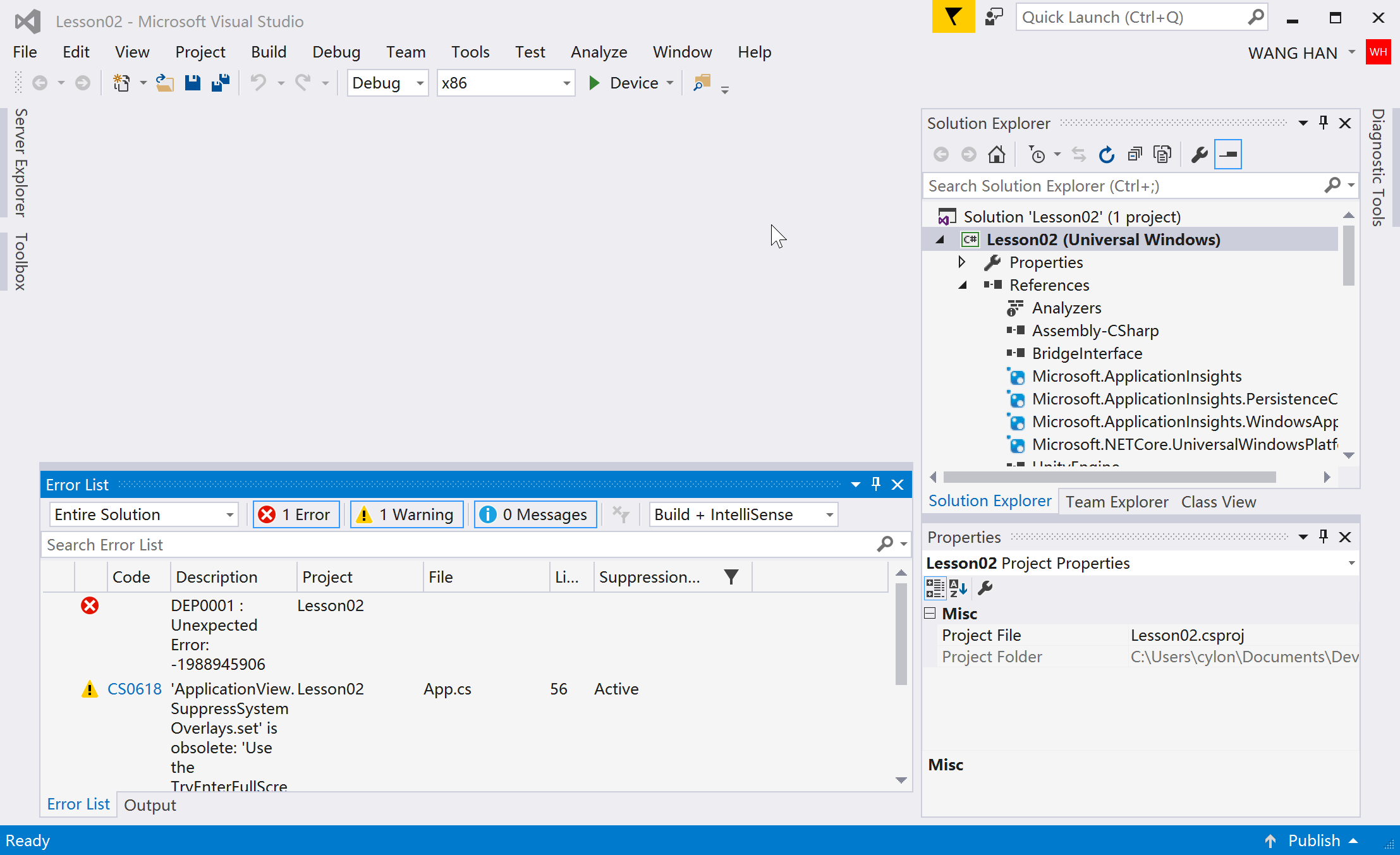


2.选择emulator或是所使用的设备

这里推荐使用HoloLens设备，对于VR/AR应用来说，用emulator真的是有点没道理。

当然，前提是你有一台HoloLens，然后把USB接口接到电脑上了，而且要开启HoloLens。

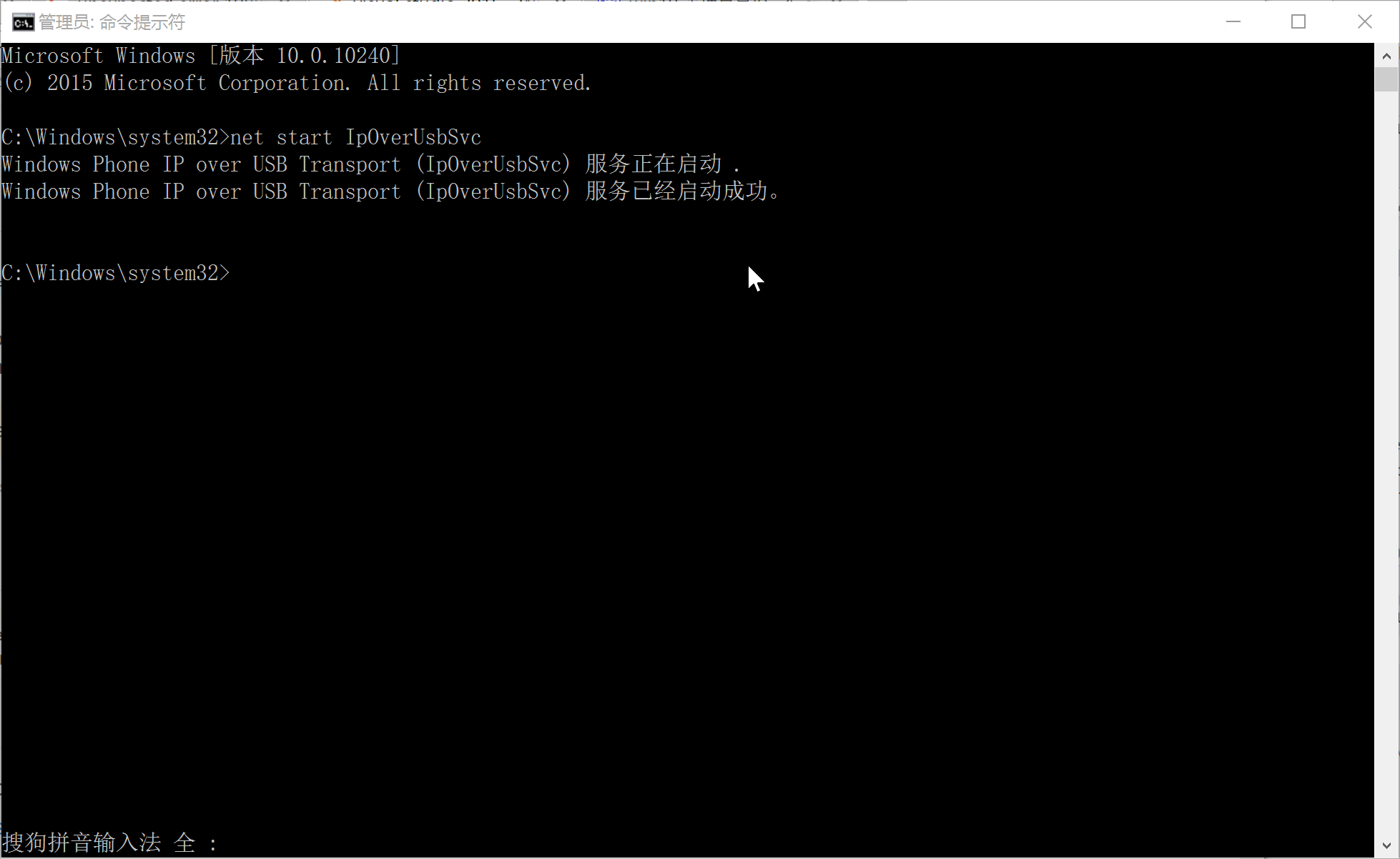
然而，这一步通常会碰到无数的坑，比如下面这一个



此时要以管理员身份打开cmd程序，具体方法是在搜索栏输入cmd，然后鼠标右键单击，选择以管理员身份运行。

然后输入以下命令：

net start IpOverUsbSvc



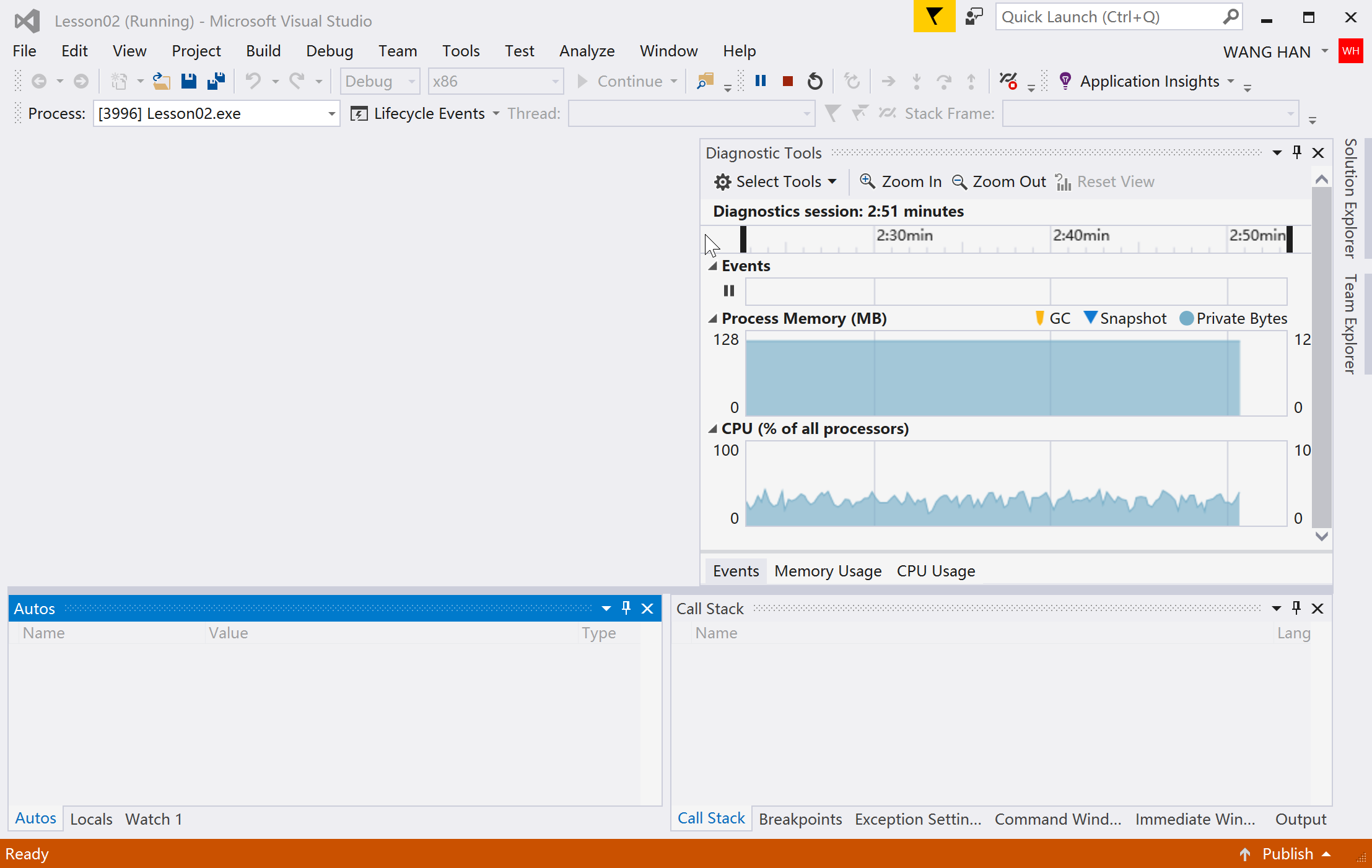
在Visual Studio中Clean刚才的项目，然后选择Build-> Deploy Lesson02

注意确保HoloLens连接到电脑上。

没有问题以后点击VS工具栏上的绿色运行按钮，等一会儿就可以在HoloLens上看到刚才的全息应用了。

3.在HoloLens上查看应用的运行状态

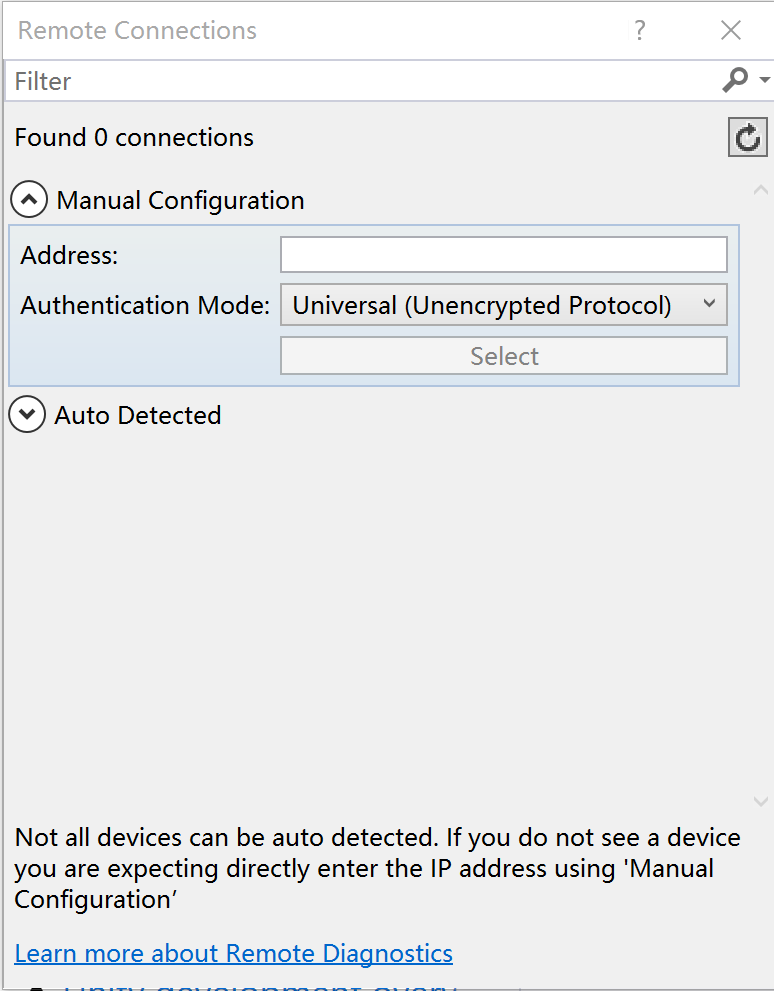
此时VS上会显示应用的相关状态。



OK,现在可以点击红色按钮Stop Debugging

当然，如果每次调试都要连接USB，也不是特别的方便，微软提供了另外一种调试方式，就是HoloLens over Wi-Fi。

在选择设备那里，在Local Machine的下面有一个Remote Machine。



在Address部分手动输入HoloLens的IP地址。

然后把Authentication Mode更改为Universal (Unencrypted Protocol)。

当然，这里就涉及到一个问题，如何识别HoloLens的IP地址。

在HoloLens中可以有三种方式获取IP。

第一种是进入设置应用，然后选择Network&Internet,然后选择Advanced Options，然后在IPv4 address旁边就会看到IP地址

微软推荐的第二种方式就是用语音说，”Hey Cortana, What’s my IP address?”，然后Cortana美女就会显示并读出IP地址。

然而我发现这样说Cortana只是帮忙打开了浏览器，然后搜索关键词”What’s my IP address”，略为坑爹。

个人尝试后发现得换一种说法，”Hey Cortana, Show me my IP address”，才会奏效。当然不知道是不是Cortana来大姨妈了。

微软推荐的第三种方式是在PC电脑上用web浏览器打开device portal，

具体方式是输入https:// <YOUR\_HOLOLENS\_IP\_ADDRESS>

我看到的感觉是不能忍，哥要是知道IP地址了，还需要到这儿来？

总之，在输入了IP和选择模式后，就可以确定了。

后面找时间再补充使用Emulator进行调试的教程。