Hololens 2 Unity tutorial (Windows)

Setup and first build

In this section, I'll be teaching you how I got my first project up and running in Hololens 2, from an application built in Unity.

First step - installing packages and applications

1. Let's start by installing Visual Studio Community 2022, in case you don't have it yet. Here you can find the microsoft page to install it. Whether you downloaded it just now or already had VS on your machine, be sure to add the following workloads on your installation (You can have access to that if you go to Settings -> System -> Storage -> Apps & Features, then search for Microsoft Visual Studio Installer, click Modify and then Modify again):
   - .NET desktop development (or download manually here).
   - Desktop development with C++.
   - Universal Windows Platform (UWP) development.
   - Game development with Unity.
2. Now, if you don't have the Windows 10 SDK, you can find the installer here.
3. Download the Emulator here (optional, just in case you don't have the Hololens in hand).
4. Open Unity Hub and install the Universal Windows Platform Build Support module for the unity version that you want to use (unity 2019.3 not recommended due to compile errors in ARM build architecture). You can do that by opening unity hub -> Installs -> clicking the gear button for the version you want -> Add Modules -> Universal Windows Platform Build Support.
5. Last but not least, install the Mixed Reality Feature Tool from Microsoft here.

We're all set, now it's time to build our first application!

Second step - creating first unity project

1. Open Unity Hub.
2. In the Projects tab, click New Project.
3. In the dropdown underneath the New project text, select the unity version that you have installed Universal Windows Platform Build Support.
4. The template must be 3D Core.
5. Click Create Project.

Third step - configuring build for Hololens 2

1. Go to File -> Build Settings
2. Select Universal Windows Platform.
3. Click Switch Platform if needed.
4. Now make sure to set the following settings: (if any error occurs, check if all the installations are properly working)
   - Target device: HoloLens

- Architecture: ARM64
- Build Type: D3D Project
- Target SDK Version: Latest Installed
- Minimum Platform Version: 10.0.10240.0
- Visual Studio Version: Latest installed
- Build and Run on: Local Machine
- Build configuration: Release (there are known performance issues with Debug)

Forth step - importing and configuring MRTK in Unity

1. Open the previously installed Mixed Reality Feature Tool, and run MixedRealityFeatureTool.exe (unzip the folder if necessary).
2. In the Mixed Reality Feature Tool, select Start.
3. Select the project folder that we've created by clicking the three dots in Project Path.
4. Click Discover Features.
5. On the Discover Features page click the "+" button to the left of Mixed Reality Toolkit (0 of 10) and select the latest version of Mixed Reality Toolkit Foundation.
6. On the Discover Features page Click the "+" button to the left of Platform Support (0 of 5) and select the latest version of Mixed Reality OpenXR Plugin.
7. Click Get Features -> Import -> Approve -> Exit.
8. Go back to Unity.
9. Restart Unity if asked to do so.
10. The MRTK Project Configurator should appear. If it doesn't, on the menu bar select Mixed Reality -> Toolkit -> Utilities -> Configure Project for MRTK.
11. Click Unity OpenXR Plugin (recommended).
12. On the Welcome to MRTK! screen, click Show XR Plug-In Management Settings.
13. On the Project Settings window now open be sure that you're on the XR Plug-in Management page with the Universal Windows Platform settings (Windows logo tab) displayed.
14. Select Initialize XR on Startup if not selected, and then, under Plugin Providers, click Open XR.
15. Two items will appear underneath OpenXR. Select Microsoft HoloLens feature group.
16. Now a yellow warning triangle next to OpenXR. Hover your cursor over the triangle, and then select it.
17. Click Fix All.
18. One issue still remains "At least one interaction profile must be added ..."
19. Click Edit in that warning. You're now in the settings for the OpenXR plugin in the Project Settings window.
20. First click the Depth Submission Mode drop down and then select Depth 16 Bit.
21. Underneath Interaction Profiles, there is a plus sign (+) button.
22. Click the button three times, with the following profiles each time:
    - Eye Gaze Interaction Profile
    - Microsoft Hand Interaction Profile
    - Microsoft Motion Controller Profile
23. If any profile added appear with a yellow triangle next to it, fix the warnings by clicking the triangle and then Fix All button.
24. In the Project Settings window under OpenXR Feature Groups, make sure that the following are selected:
    - Microsoft HoloLens
    - Hand Tracking

   - Motion Controller Model
25. Close the Project Settings window.
26. Back at the MRTK Project Configurator window, select Apply Settings.
27. Click Next, then Apply.

Now you're ready to start working with unity and Hololens. The next step is not necessary, so you can jump to sixth step if you want to.

Fifth step - example scene with many interactions implemented (optional)

1. Go to the official MRTK github repository here
2. Go to tags next to branches and select the latest release. For me it's Microsoft Mixed Reality Toolkit v2.7.3.
3. Scroll down to Assets and download the package with examples in the version you chose, in my case, "Microsoft.MixedReality.Toolkit.Unity.Examples.2.7.3.unitypackage".
4. Open unity and on the menu bar, select Assets -> Import package -> Custom package, and click on the example package that you've just downloaded from github.
5. Just click Import when prompted with the package contents.
6. If it worked, there should be a folder named "Examples" with the path Assets/MRTK/Examples.
7. Now open the example scene in Assets/MRTK/Examples/Demos/HandTracking/Scenes/HandInteractionExamples.unity.
8. If asked to import Text Mesh Pro (TMP), click Import TMP Essentials. Close the TMP window.
9. Last but not least, go to File -> Build Settings on the menu bar, and click Add Open Scenes.
10. Click Build and choose a folder to save it.
11. Wait until it's finished.

Sixth step - enable developer mode in Hololens and setting Visual Studio to build remotely thought wifi or cable

1. Turn on your Hololens 2.
2. Enter the menu and Select Settings -> Update and Security -> For developers and enable developer mode.
3. Now on your pc, go to the folder that you have built the unity app and click on the file with the sln extension, in my case, "HoloTutorial.sln".
4. That should open Visual Studio. If not, open it manually by selecting in the menu bar File -> Open -> Project/Solution and choose the file with sln extension.
5. Now on the menu bar, go to Project -> Properties.
6. Configure Visual Studio for HoloLens by selecting the Master or Release configuration and the ARM64
7. Under Configuration Properties click General and ensure that you have the correct installed Visual Studio in the Platform Toolset box, in my case, Visual Studio 2022, then click Apply and close this window.architecture. In my case, release.
8. Click the deployment target drop-down and then do one of the following:
   - If you're building and deploying via USB, select Device
   - If you're building and deploying via Wi-Fi connect to the same network as the Hololens, select Remote Machine and do the following:
     - On the menu bar, go to Project -> Properties, and then under Configuration Properties click Debugging.

- You must enter the machine name of the Hololens which is the local IP address of the device (to find the IP of the hololens, open the Settings again on the device, and go to Network & Internet -> Wi-Fi, scroll all the way down and click Hardware properties. It should be displayed as IPv4 address)
- Click Ok and close the window

9. Now build and run the application by clicking the green play button.
10. If Visual Studio asks for a PIN, open the menu in Hololens 2 and Select Settings -> Update and Security -> For developers, then click Pair. There should appear a PIN.

If all worked, the application should appear at any seconds on your device as soon as it finish building! That's all!