Gaze Data Capture

A short summary of capturing gaze data on 3D objects in Unity, exporting the data, and processing it into voxels and point cloud that overlaps with the 3D model.

Video demo: https://drive.google.com/file/d/1Edbv7P_Gu9Oe8DqdgrTcawvuWrME-s18/view?usp=sharing

Speed up video of HoloLens 2 Emulator: https://drive.google.com/file/d/1NJ4swOU7Pcr9e_bjFpZia7RtBmgX4Wwy/view?usp=sharing

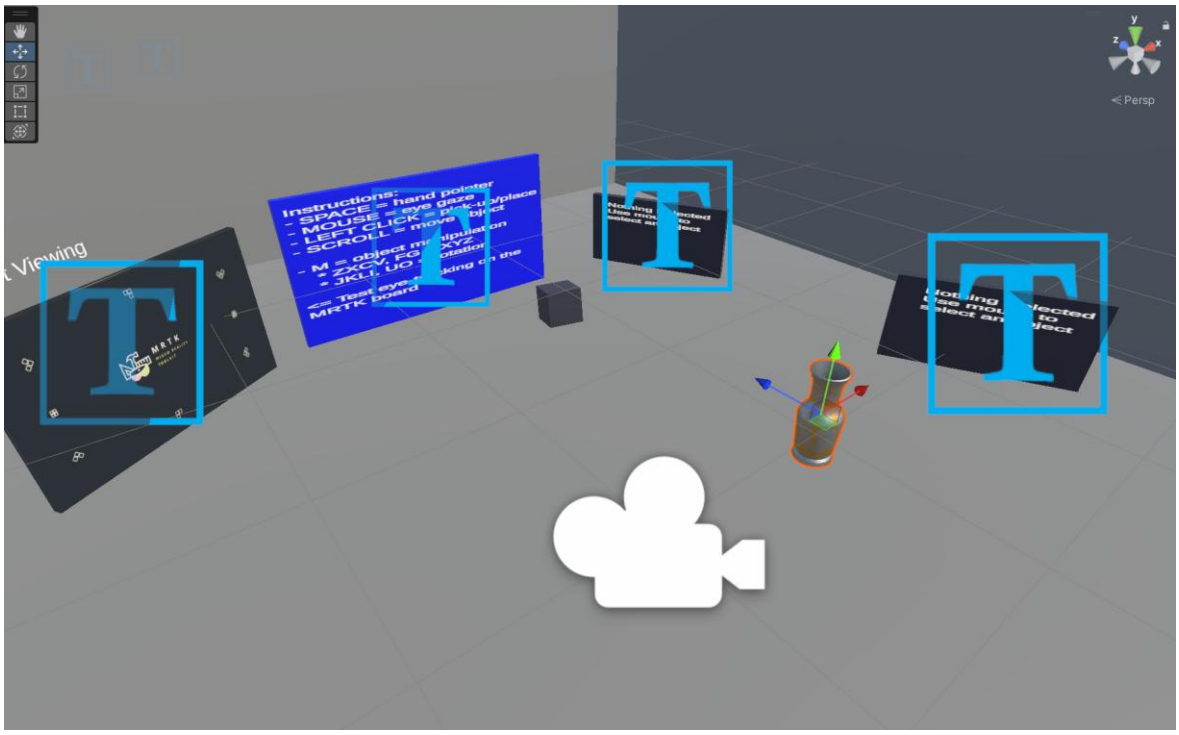
Contents

1 Unity Setup 1

2 Processing 3

3 HoloLens 2 Emulator 4

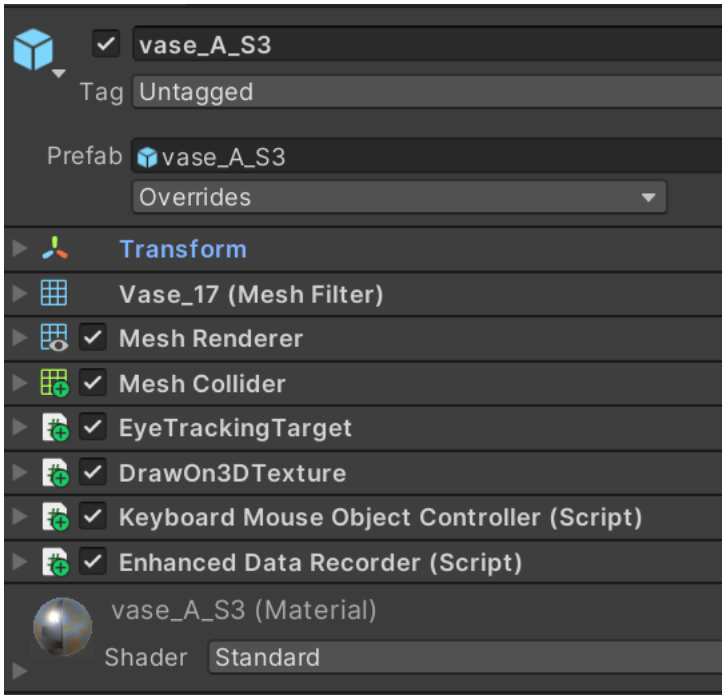
1 Unity Setup



Scene setup can be found here: https://github.com/luhouyang/SampleDazeData.git

Code written for HoloLens 2 Emulator is directly compatible with HoloLens2 without additional modifications. The gaze data collection process for the current demo, HoloLens 2 Emulator, and a proposed control method on HoloLens 2 are as follows:

Function	Current Demo	HoloLens 2 Emulator	HoloLens 2
Select Object	**LEFT MOUSE CLICK**	**Mouse Cursor**	**Gaze [DOCS]**
Start Capture	**M Key**	**RIGHT MOUSE CLICK**	**Air tap gesture [DOCS]**
Rotate Object around XYZ-axis	**JKLI, UO Keys**	**Arrow Keys + Q/E**	**Hand Orientation [DOCS]**
Gaze	**Mouse Cursor**	**Mouse Cursor**	**Eye Gaze [DOCS]**
Save Data	**Backtick (`) Key**	**RIGHT MOUSE CLICK**	**Air tap gesture [DOCS]**



The picture shows the basic scripts for manipulating objects and recording gaze data. The details about each script are as follows:

Script	Functionality	Compatibility
EyeTrackingTarget	- MRTK pre-defined script for tracking eye gaze input. - Can be used to trigger actions.	Can be used in HoloLens 2 Emulator and HoloLens 2 without changes.
DrawOn3DTexture	- User-defined script. - Uses the **GazeProvider** to access information such as **GazeDirection, GazeOrigin, GazeTarget, HitInfo, HitNormal, HitPosition, etc. [MORE]**.	Can be used in HoloLens 2 Emulator and HoloLens 2 without changes.

	- It has a draw function that overlays the object mesh with the heatmap.	
Keyboard Mouse Object Controller	- User-defined script. - Captures keystrokes as command. - On **M** key, starts the recording. - Handles user input for rotating and moving object.	All interactions and input have to be remapped to **Gaze**, **HandTracking**, **Gestures**
Enhanced Data Recorder	- User-defined script. Uses the **GazeProvider**. - Listen to the 'isRecording' state from 'Keyboard Mouse Object Controller' - Records all relevant data that might be needed. - **Gaze related**: Timestamp, HeadX, HeadY, HeadZ, HeadFwdX, HeadFwdY, HeadFwdZ, EyeOriginX, EyeOriginY, EyeOriginZ, EyeDirX, EyeDirY, EyeDirZ, HitX, HitY, HitZ, TargetName - **Transform related**: TransformTimestamp, PosX, PosY, PosZ, RotX, RotY, RotZ, RotW, ScaleX, ScaleY, ScaleZ, VelX, VelY, VelZ, AngVelX, AngVelY, AngVelZ - Provides functions to **save** collected data to csv files, filter and process data into point cloud (with intensity).	All functions can be used in HoloLens 2 Emulator and HoloLens 2 without changes. Only the key binding on backtick (`) for save function has to be remapped to 'Right Mouse Click' or 'Air Tap Gesture'

2 Processing

GitHub: https://github.com/luhouyang/3d-heatmap-generation/tree/main/experiment/unity

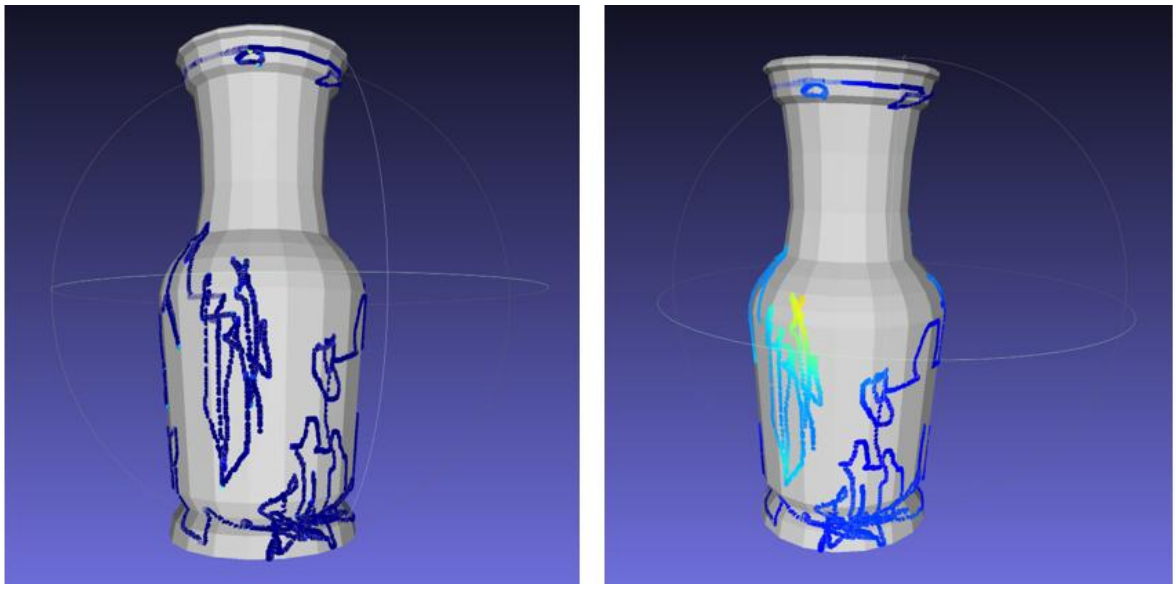


Figure 1: LEFT (Point cloud without gaussian smoothing) RIGHT (Point cloud with gaussian smoothing sigma=0.02)

Using point cloud data exported from Unity in csv format, a Python script is used to convert it to ply format and visualize it. The point cloud on the left shows the original data points (~7000) without gaussian smoothing, as the point density is high and intensity (gaze duration) of close points are not aggregated together almost all points have similar colour. The point cloud on the right obtained after applying gaussian smoothing has a more informative visualization. Further reading will be done on aggregating and processing saccades and fixation to produce realistic heatmaps.

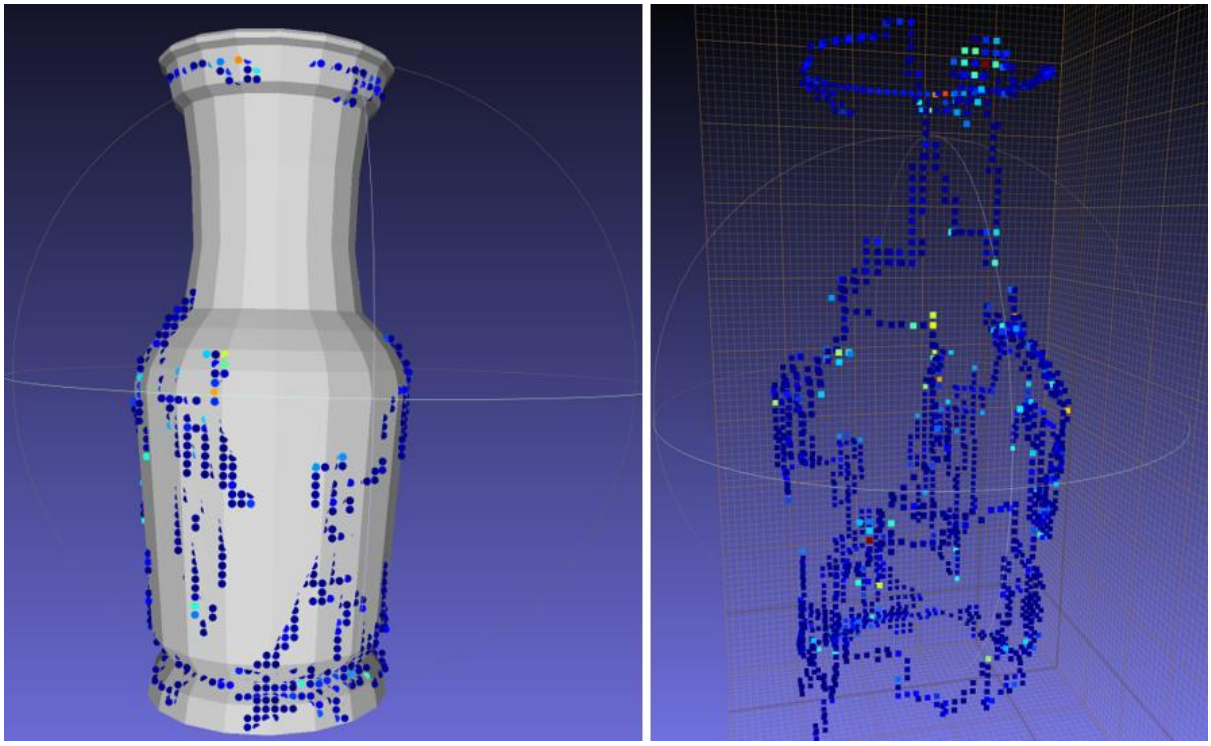


Figure 2: Voxel grid from point cloud csv without gaussian smoothing.

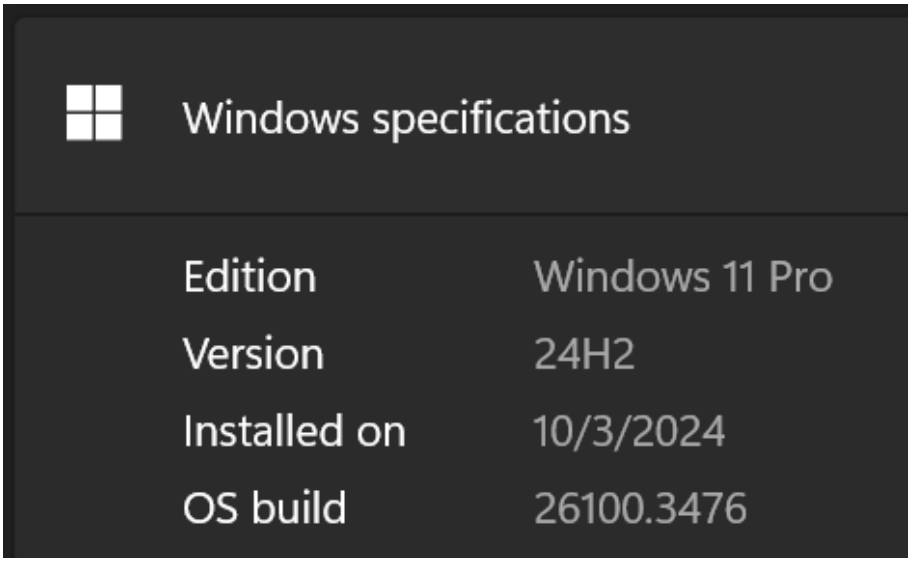
Using the same point cloud data from Unity, a Python script is used to generate a voxel grid without gaussian smoothing. This method shows a realistic heatmap just by aggregating the points and their intensities into the bounding voxels.

3 HoloLens 2 Emulator

https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/using-the-hololens-emulator

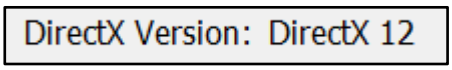
SETUP

1. Check device compatibility
    a. 64-bit Windows 11 Pro, Enterprise, or Education

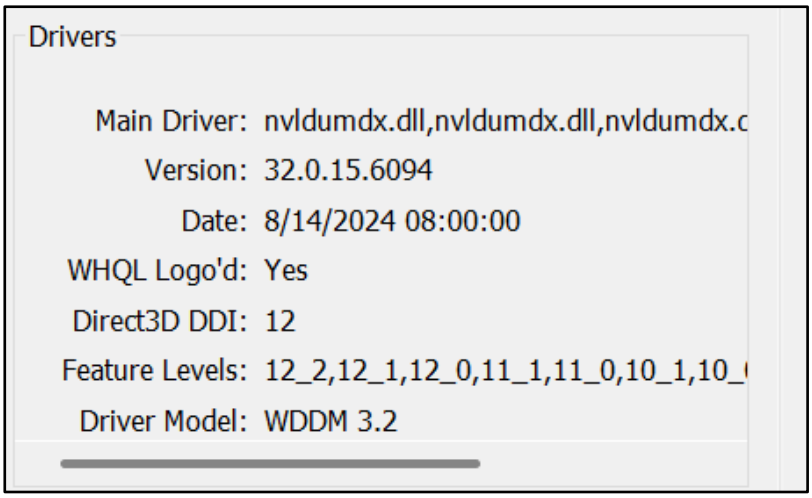


b. CPU with four cores (or multiple CPUs with a total of four cores)
c. 8 GB of RAM or more
d. Other checks

2. Check GPU requirements
   a. DirectX 11.0 or later. Press **WINDOWS_KEY + r**, type 'dxdiag' into the search and press **ENTER**.

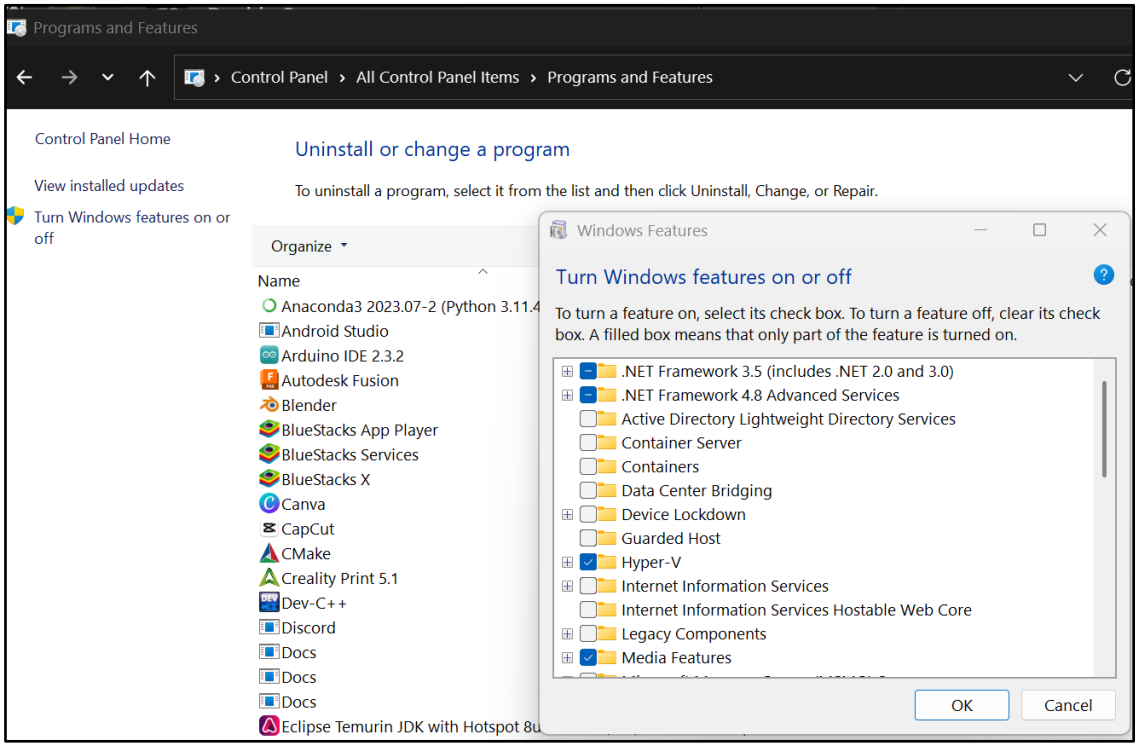
   

   b. WDDM 2.5 graphics driver (HoloLens 2 Emulator). On the same DirectX panel, click on **Next Page** and under **Drivers** check your **WDDM** version.



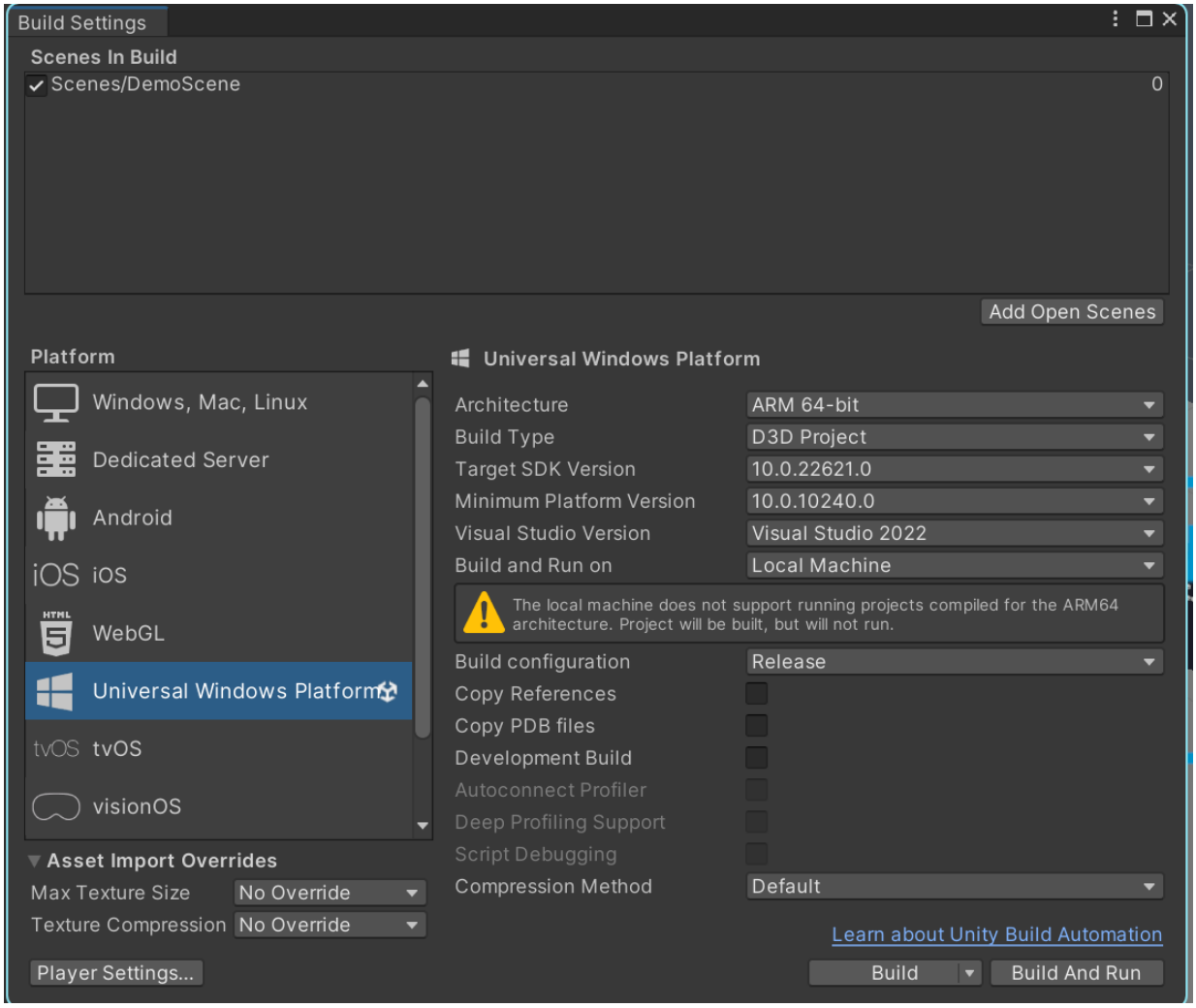
3. Check BIOS settings (requirements)
    a. Hardware-assisted virtualization
    b. Second Level Address Translation (SLAT)
    c. Hardware-based Data Execution Prevention (DEP)

4. Enable Hyper-V

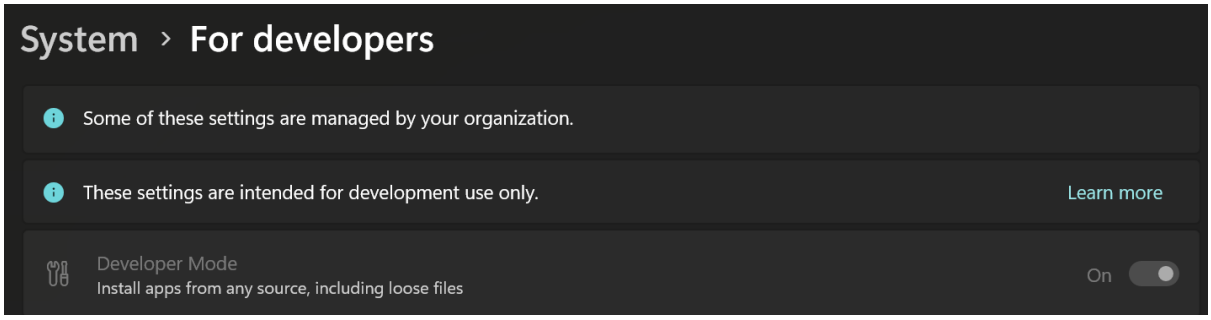


5. Download and install HoloLens2 Emulator 10.0.22621.1402 (here).

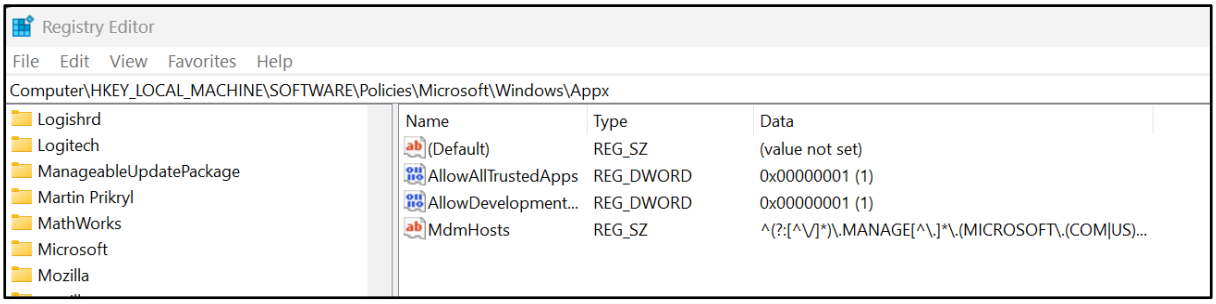
6. Build the Unity project into ‘build’ folder (create manually). Double check the settings. Navigate to **File** > **Build Settings**.



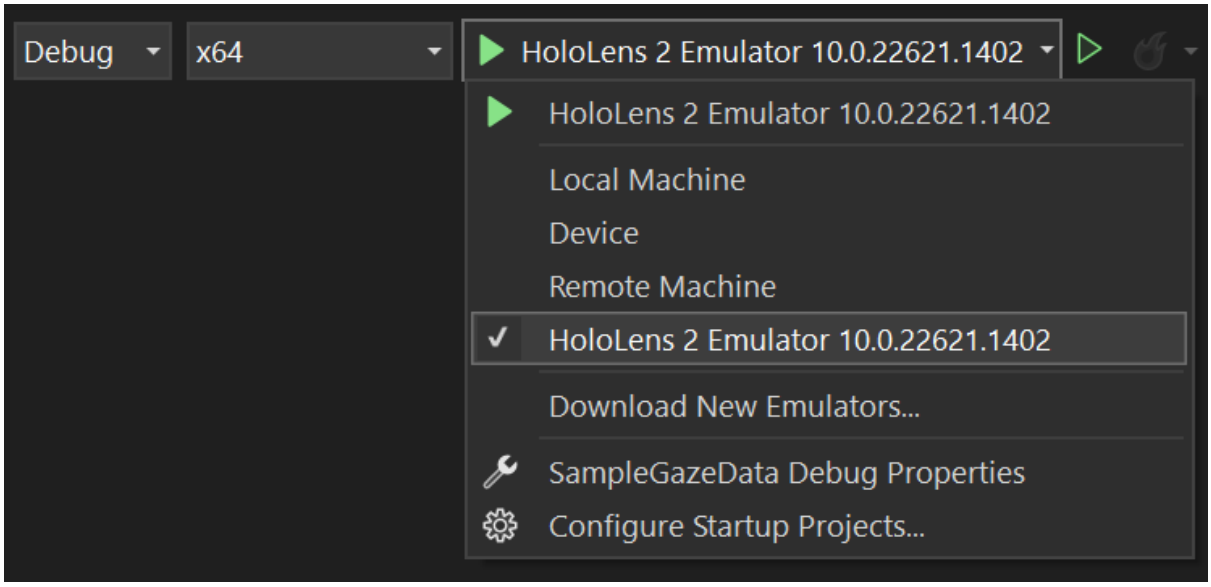
7. Enable **developer mode**.
    a. From settings.



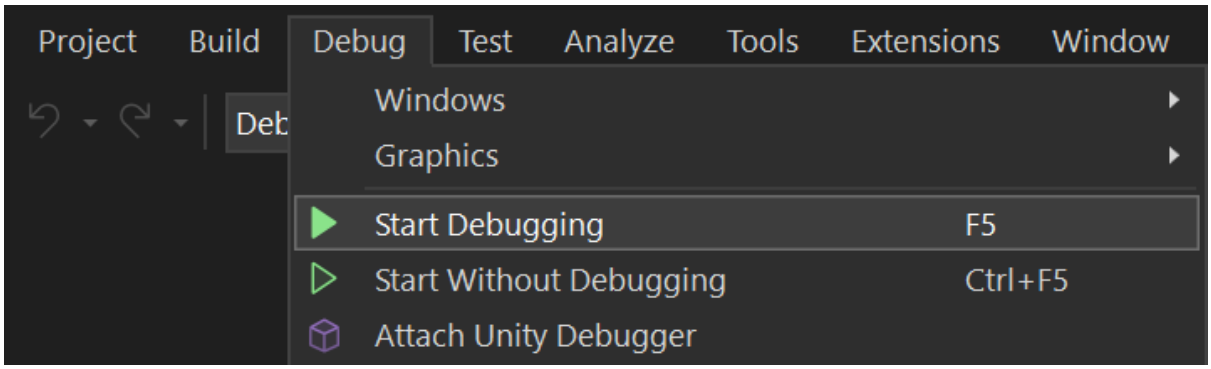
b. IF NOT ABLE TO ENABLE. From the search bar open ‘Registry Editor’, **Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Appx** then edit **AllowAllTrustedApps = 1**, **AllowDevelopmentWithoutDevLicense = 1**



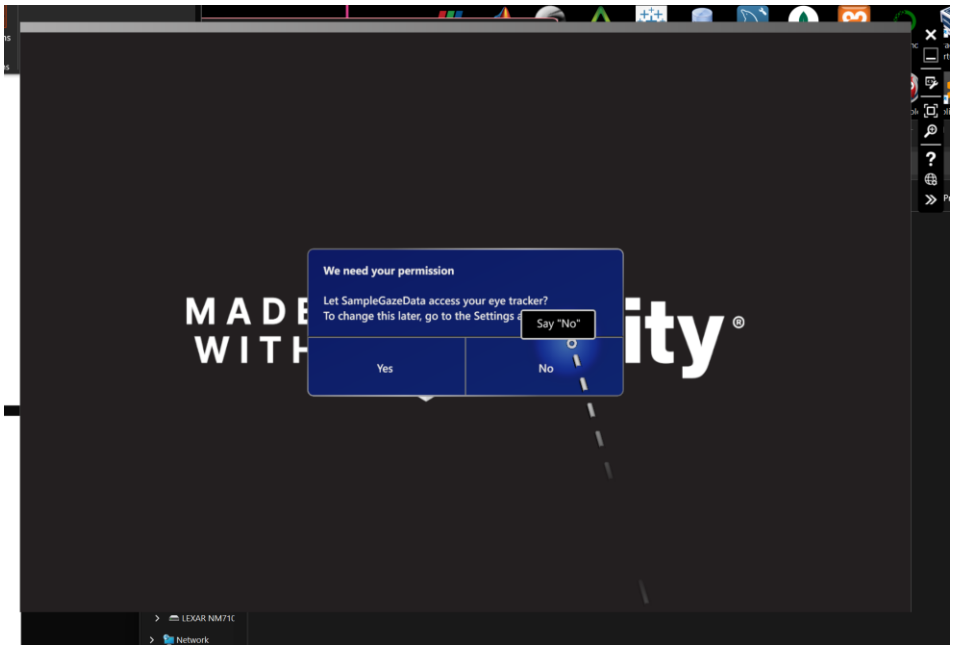
8. Open the solution (.sln) in Visual Studio 2022. Configure to run on HoloLens 2 Emulator [documentation]



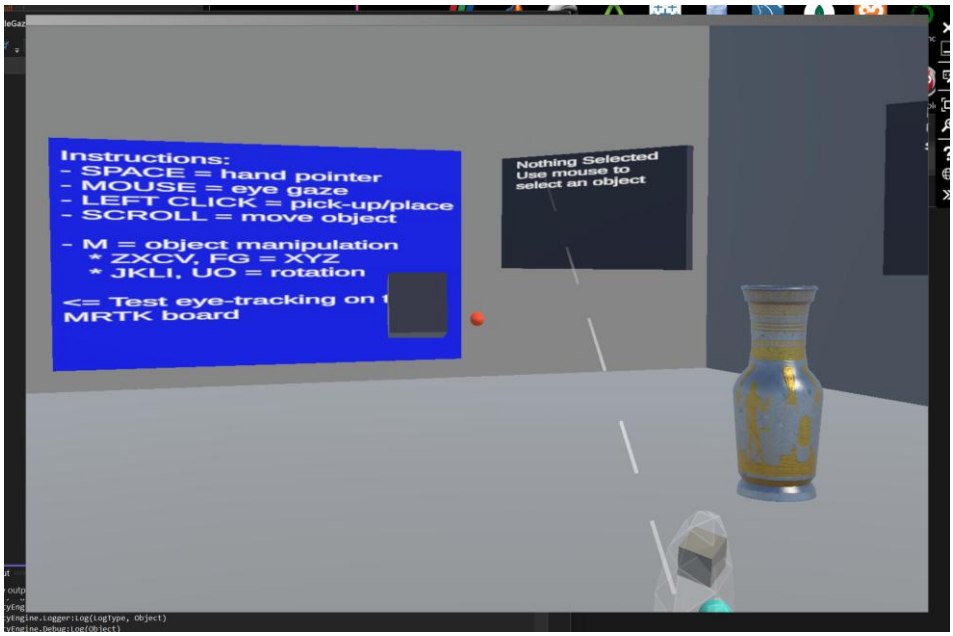
Start Debugging



Permission



Scene



Upcoming

Migrate the user controls from 'Keyboard Mouse Object Controller' C# script to the equivalent user input scripts provided by MRTK, Unity for HoloLens 2 support. Record data using HoloLens 2 Emulator (equivalent to real physical device according to Microsoft documentation).