Adaption Tips:

This file is intended to provide information and ideas that may be helpful in porting the projects to HoloLens 2. Please note that at the time of writing we do not have HoloLens 2, so we this is off the information and documentation Microsoft has released.

When porting the app to HoloLens 2, you should make a backup copy of the FinalMarkingPrototype. This can serve two purposes: Saving the original version in case something goes wrong, and to give a frame of reference when making changes.

Based on the Microsoft docs, there is quite a bit of work that will need to be done to port the FinalMarkingPrototype to HoloLens 2, due to the backend changes. To help show some specifics, here are the current specs compared to the recommended ones for HoloLens 2:

Current Specs (HoloLens 1):

Unity version: Unity 2018.3.14f1

Toolkit: HoloToolkit 2017.4.2.0-Refresh

Scripting Backend: .NET

Build Compilation: x86

Reccomended Specs (HoloLens 2)

Unity version: Unity 2018.3.x or 2019.1+

Toolkit: Mixed Reality Toolkit 2

Scripting Backend: IL2CPP

Build Compilation: ARM32 or ARM64

The changes likely to cause the biggest challenges are upgrading the Toolkit and the scripting backend.

Toolkit:

We initially started with the MRTK 2.0, but rolled back to the HoloToolkit due to difficulties making the MRTK work with HoloLens 1. We have noticed that each has scripts and prefabs that the other does not, and Microsoft notes that the MRTK 2.0 has a quite different structure. Make sure to save any of the HoloToolkit assets that are being used before porting, with the most important script to pay attention to when converting is the WorldAnchorManager. As described in the README, we modified the WorldAnchorManager to have the functions needed for this project. When switching Toolkits, make sure you either replace the MRTK 2.0 WorldAnchorManager with the modified version or modify the MRTK 2.0 version to make DeleteAnchor public.

Microsoft also advises to expect compilation errors after updating the Toolkit due to name changes, so make sure to fix these as appropriate. Some examples of what to expect are the interface names, such as IFocusable to IMixedRealityFocusedHandler.

Scripting Backend:

This is the change that will cause the most trouble. While we are not certain of all the changes that will be necessary, brief experimentation has confirmed that the Newtonsoft Json Serialization library utilized by the Save/Load, BoundingBoxTags, SerializableVector3, and SerializableTag is not compatible with IL2CPP. This means a new JSON serialization library will be necessary. There likely will be other libraries that break that are not known at this time. If strange errors or seemingly random null pointer exceptions occur in any libraries or scripts, check to see if they are compatible with IL2CPP as that may be the cause.

Interaction Model:

The HoloLens 2 has more interactions avaliable to use, including custom gestures and eye tracking. While we will leave the specific implementations for how to update the prototype's interaction model, we do have a few ideas and suggestions. But first, to help assist in updating the current interaction model here is a breakdown of it:

Gaze + Tap is the primary mode of interaction for the application. The target is determined based on where the Gaze cursor is. To move the Gaze cursor, the user must move their head. When the cursor overlaps with a collision box, that object is considered "In Focus." When porting to the HoloLens 2, it would be a good idea to base the Gaze cursor on Eye Tracking instead of Head Tracking. Here are the functions that use Gaze+Tap:

\*Opening tags: When the Gaze cursor lingers on a tag for 1 second, the tag will expand to reveal its text. This is an interaction that is recommended to keep, in order to prevent cluttering the viewing area with a ton of open tags.

\*Spawning Tags: On the menu, tags can be spawned by selecting the corresponding button with Gaze + Tap. This could be replaced with just a tap, or with dedicated gestures per tag in HoloLens 2.

\*Stopping Dictation: Due to dictation pausing Speech Input, to stop tag dictation the user must focus on the tag currently accepting dictation and tap on it. A dedicated dictation gesture would be an ideal replacement.

\*Turning Help Pages: To turn pages on the Help menu, tap on the menu. A dedicated page turn gesture could be used here on HoloLens 2, which can be shared with page turning in the tags.

\*Moving tags: Moving tags is done by focusing on them and then doing the tap and hold gesture. Tags move based on the movement of the tapping and holding hand. In HoloLens 2, this could theoreitcaly be done with grabbing the tag and grabbing it in a similar manner.

\*Resizing Bounding Boxes: This is done by tapping and holding with both hands. Bring both hands close together to shrink the BB or pull apart to increase the size.

The secondary mode of interaction is Speech input, used for functions that are more complex. These could all theoretically be replaced with gestures or buttons for the user to press. The list of functions is as follows:

\*Opening the menu: Done by saying "Menu", "Open Menu", or "Show Menu".

\*Closing the menu: Done by saying "Close Menu" or "Hide Menu".

\*Closing the help menu: Done by saying "Close Help" or "Hide Help".

\*Opening help: Done by saying "Help", "Open Help", or "Show Help"

\*Deleting Every Tag: Meant only for debugging and emergencies, permanently deletes every single tag. In order to prevent accidental use, the code phrase is "Raspberry Tickle Bears". Don't give users an easier way to do this, probably best to remove from final version.

Some interactions use a combination of Gaze + Speech. These are:

\*Deleting tags: Individual tags can be deleted by focusing on them with the Gaze cursor and saying "Delete".

\*Dictation: When a tag is in focus with Gaze, by saying "Dictate" the user can start dictation mode to save text on the tag. A possible replacement/addition can be a dictation gesture, which can also be utilized to stop dictation.

\*Turning pages: When gazing on tags, if there is enough text for multiple pages you can turn the page forwards or backwards respectively with "Next" and "Back". A turn page gesture would be good for these.