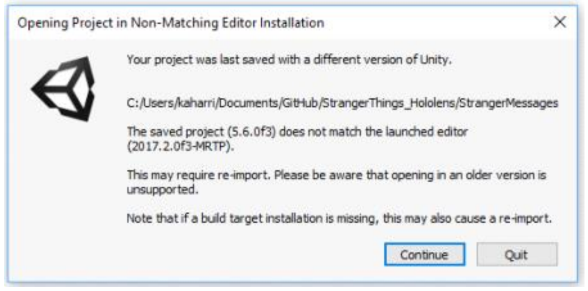
Step by Step

This process will be different for every project due to every project using the HoloToolkit scripts in their own way and developer architecture being different. However, here are some steps and helpful hints that will walk you through what you need to do

Update to newest MRTK

Opening project in Unity 2017.2.0f3 MRTP Version

This is a special version of Unity that runs Mixed Reality that will eventually be merged with the main branch of Unity.



Opening Project in Non-Matching Editor Installation

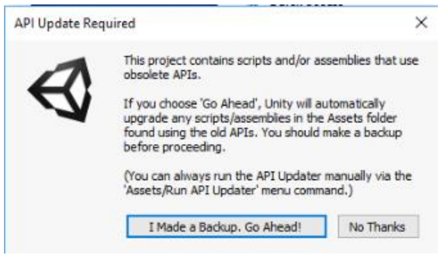
Your project was last saved with a different version of Unity.

C:/Users/kaharri/Documents/GitHub/StrangerThings_Hololens/StrangerMessages

The saved project (5.6.0f3) does not match the launched editor (2017.2.0f3-MRTP).

This may require re-import. Please be aware that opening in an older version is unsupported.

Note that if a build target installation is missing, this may also cause a re-import.



API Update Required

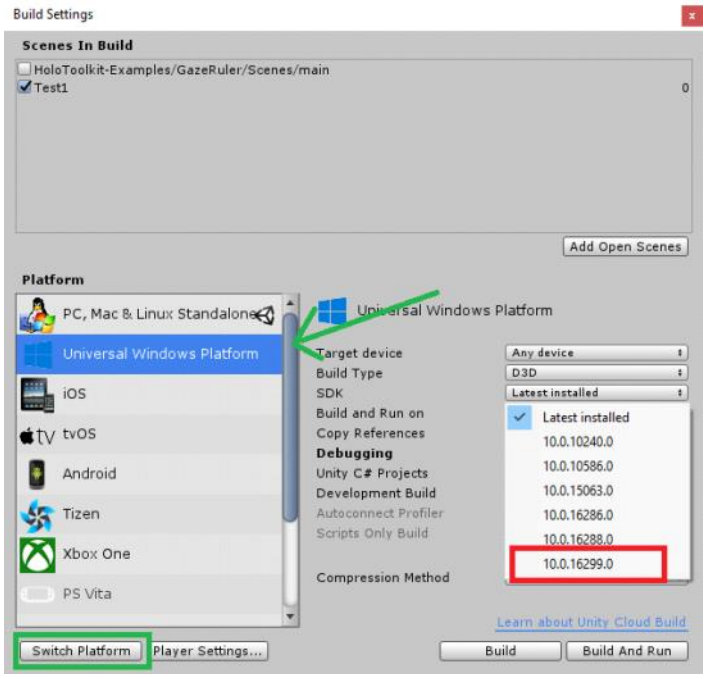
This project contains scripts and/or assemblies that use obsolete APIs.

If you choose 'Go Ahead', Unity will automatically upgrade any scripts/assemblies in the Assets folder found using the old APIs. You should make a backup before proceeding.

(You can always run the API Updater manually via the 'Assets/Run API Updater' menu command.)

Change Build Settings to UWP - Make sure latest SDK Package is installed

Initially opening the project with a new version of Unity will change your build settings resulting in errors. To fix most of them you'll need to switch to UWP setting.



Build Settings

Scenes In Build

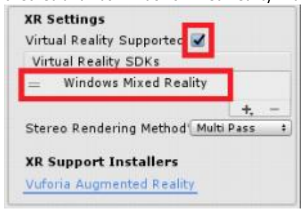
- [ ] HoloToolkit-Examples/GazeRuler/Scenes/main
- [x] Test1 0

Add Open Scenes
Platform
PC, Mac & Linux Standalone
Universal Windows Platform
iOS
tvOS
Android
Tizen
Xbox One
PS Vita
Universal Windows Platform

Target device: Any device
Build Type: D3D
SDK: Latest installed
Build and Run on
Copy References
Debugging
Unity C# Projects
Development Build
Autoconnect Profiler
Scripts Only Build
Compression Method

10.0.16299 is the latest available build of our SDK. Update Visual Studio if you do not have this SDK install.
https://developer.microsoft.com/en-us/windows/mixed-reality/install_the_tools

When you hit player settings make sure the new XR Tab in unity has the Virtual Reality Supported box checked and has Windows Mixed Reality in the Virtual Reality SDKs section.



XR Settings

Virtual Reality Supported ✓

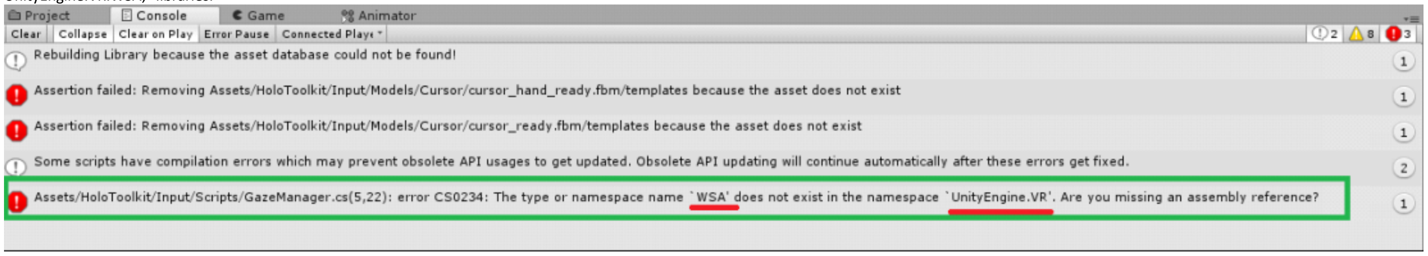
Virtual Reality SDKs

Windows Mixed Reality

Stereo Rendering Method: Multi Pass

XR Support Installers

WSA ERRORS
These errors are from Unity changing their naming to the XR and the old toolkit using the old "using UnityEngine.VR.WSA;" libraries.



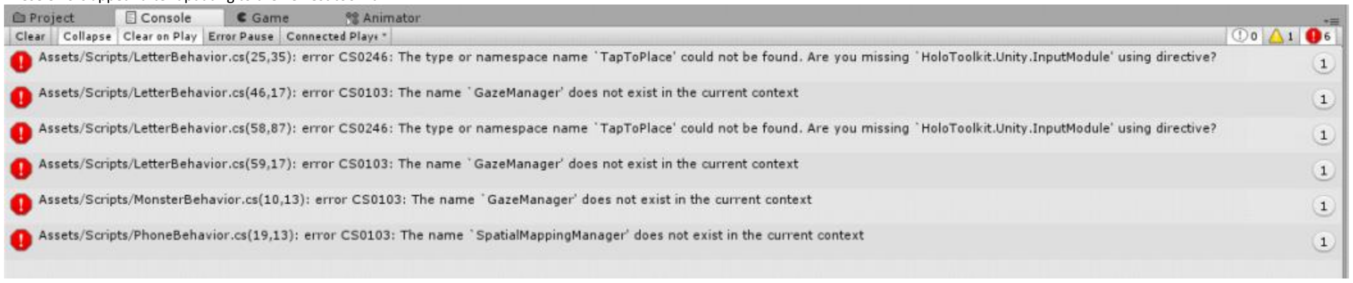
```
Rebuilding Library because the asset database could not be found!                                                1
Assertion failed: Removing Assets/HoloToolkit/Input/Models/Cursor/cursor_hand_ready.fbm/templates because the asset does not exist    1
Assertion failed: Removing Assets/HoloToolkit/Input/Models/Cursor/cursor_ready.fbm/templates because the asset does not exist    1
Some scripts have compilation errors which may prevent obsolete API usages to get updated. Obsolete API updating will continue automatically after these errors get fixed.    2
Assets/HoloToolkit/Input/Scripts/GazeManager.cs(5,22): error CS0234: The type or namespace name `WSA' does not exist in the namespace `UnityEngine.VR'. Are you missing an assembly reference?    1
```

To fix Add the new Mixed Reality Toolkit from GitHub, instead of

Adding new MRTK
Delete the old HoloToolkit and Add in the new HoloToolkit from github:
https://github.com/Microsoft/MixedRealityToolkit-Unity/tree/master/Assets/HoloToolkit
No more WSA errors but new errors will appear because of library issues.

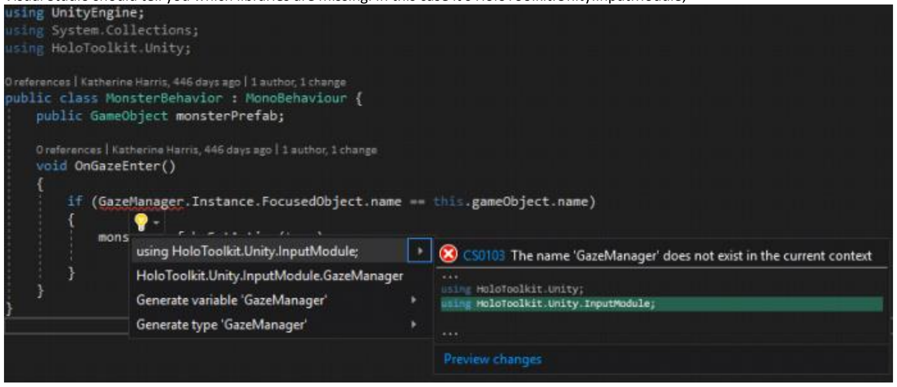
Replace the new Mixed Reality Camera for HoloLensCamera

Updating Libraries and Code
These errors appear after updating to the newest toolkit:



```
Assets/Scripts/LetterBehavior.cs(25,35): error CS0246: The type or namespace name `TapToPlace' could not be found. Are you missing `HoloToolkit.Unity.InputModule' using directive?    1
Assets/Scripts/LetterBehavior.cs(46,17): error CS0103: The name `GazeManager' does not exist in the current context    1
Assets/Scripts/LetterBehavior.cs(58,87): error CS0246: The type or namespace name `TapToPlace' could not be found. Are you missing `HoloToolkit.Unity.InputModule' using directive?    1
Assets/Scripts/LetterBehavior.cs(59,17): error CS0103: The name `GazeManager' does not exist in the current context    1
Assets/Scripts/MonsterBehavior.cs(10,13): error CS0103: The name `GazeManager' does not exist in the current context    1
Assets/Scripts/PhoneBehavior.cs(19,13): error CS0103: The name `SpatialMappingManager' does not exist in the current context    1
```

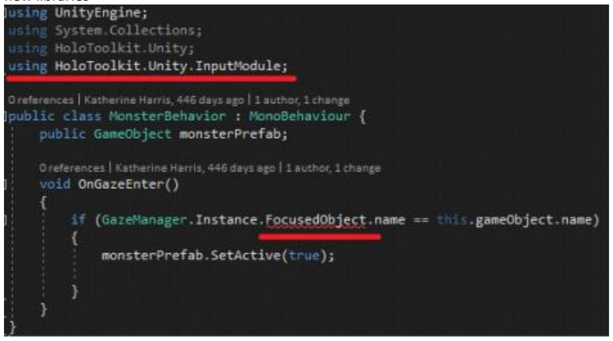
Add the new libraries
Visual Studio should tell you which libraries are missing. In this case it's HoloToolkit.Unity.InputModule;


using HoloToolkit.Unity.InputModule;
HoloToolkit.Unity.InputModule.GazeManager
Generate variable 'GazeManager'
Generate type 'GazeManager'
Preview changes

```
using UnityEngine;
using System.Collections;
using HoloToolkit.Unity;

0 references | Katherine Harris, 446 days ago | 1 author, 1 change
public class MonsterBehavior : MonoBehaviour {
    public GameObject monsterPrefab;

    0 references | Katherine Harris, 446 days ago | 1 author, 1 change
    void OnGazeEnter()
    {
        if (GazeManager.Instance.FocusedObject.name == this.gameObject.name)
        {
            mons
        }
    }
}
```

```
CS0103 The name 'GazeManager' does not exist in the current context
...
using HoloToolkit.Unity;
using HoloToolkit.Unity.InputModule;
...
```

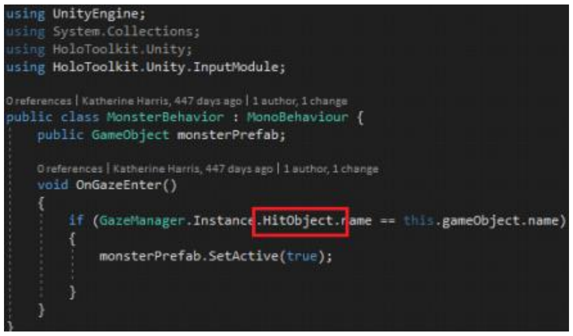
Fix the code to work with the new libraries
After you add the new libraries some of the functions will not be correct since they no longer exist in the new libraries



```
using UnityEngine;
using System.Collections;
using HoloToolkit.Unity;
using HoloToolkit.Unity.InputModule;

0 references | Katherine Harris, 446 days ago | 1 author, 1 change
public class MonsterBehavior : MonoBehaviour {
    public GameObject monsterPrefab;

    0 references | Katherine Harris, 446 days ago | 1 author, 1 change
    void OnGazeEnter()
    {
        if (GazeManager.Instance.FocusedObject.name == this.gameObject.name)
        {
            monsterPrefab.SetActive(true);

        }
    }
}
```

You have to look through the new libraries to find the correct replacement function.

In this case we are replacing FocusedObject with HitObject



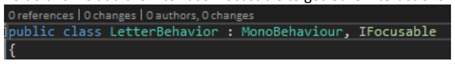
```
using UnityEngine;
using System.Collections;
using HoloToolkit.Unity;
using HoloToolkit.Unity.InputModule;

0 references | Katherine Harris, 447 days ago | 1 author, 1 change
public class MonsterBehavior : MonoBehaviour {
    public GameObject monsterPrefab;

    0 references | Katherine Harris, 447 days ago | 1 author, 1 change
    void OnGazeEnter()
    {
        if (GazeManager.Instance.HitObject.name == this.gameObject.name)
        {
            monsterPrefab.SetActive(true);

        }
    }
}
```

Do this throughout your code, until there are no more library errors.

Replacing Old listener functions with new ones

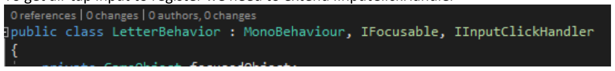
Since the focus architecture has changed the listeners like "OnGazeEnter" have changed. To use them now we need to extend the class' functionality.

To do this we add the interface IFocusable to get Gaze interactions working



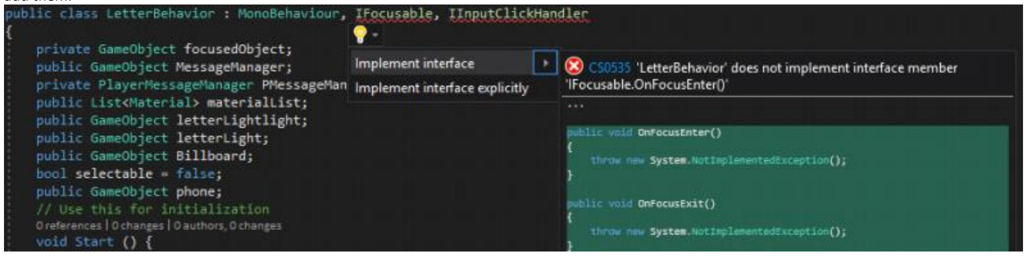
```
0 references | 0 changes | 0 authors, 0 changes
public class LetterBehavior : MonoBehaviour, IFocusable
{
```

To get air tap input to register we need to extend IInputClickHandler



```
0 references | 0 changes | 0 authors, 0 changes
public class LetterBehavior : MonoBehaviour, IFocusable, IInputClickHandler
{
```

When you add these interfaces, they expect certain methods to be added. Visual studio will warn you to add them.

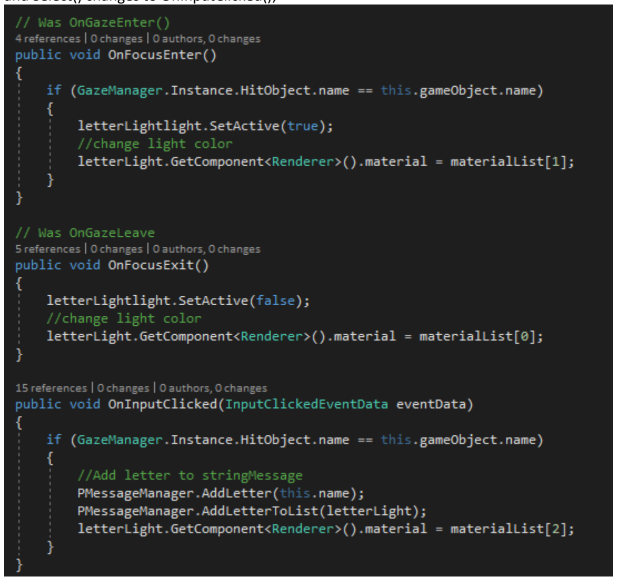


```
public class LetterBehavior : MonoBehaviour, IFocusable, IInputClickHandler
{
    private GameObject focusedObject;
    public GameObject MessageManager;
    private PlayerMessageManager PMessageMan
    public List<Material> materialList;
    public GameObject letterLightlight;
    public GameObject letterLight;
    public GameObject Billboard;
    bool selectable = false;
    public GameObject phone;
    // Use this for initialization
    0 references | 0 changes | 0 authors, 0 changes
    void Start () {
```

CS0535 'LetterBehavior' does not implement interface member 'IFocusable.OnFocusEnter()'

```
...

public void OnFocusEnter()
{
    throw new System.NotImplementedException();
}

public void OnFocusExit()
{
    throw new System.NotImplementedException();
}
```

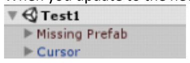
If you were using OnGazeEnter() it changes to OnFocusEnter(). OnGazeLeave() changes to OnFocusExit() and Select() changes to OnInputClicked();



```
// Was OnGazeEnter()
4 references | 0 changes | 0 authors, 0 changes
public void OnFocusEnter()
{
    if (GazeManager.Instance.HitObject.name == this.gameObject.name)
    {
        letterLightlight.SetActive(true);
        //change light color
        letterLight.GetComponent<Renderer>().material = materialList[1];
    }
}

// Was OnGazeLeave
5 references | 0 changes | 0 authors, 0 changes
public void OnFocusExit()
{
    letterLightlight.SetActive(false);
    //change light color
    letterLight.GetComponent<Renderer>().material = materialList[0];
}

15 references | 0 changes | 0 authors, 0 changes
public void OnInputClicked(InputClickedEventData eventData)
{
    if (GazeManager.Instance.HitObject.name == this.gameObject.name)
    {
        //Add letter to stringMessage
        PMessageManager.AddLetter(this.name);
        PMessageManager.AddLetterToList(letterLight);
        letterLight.GetComponent<Renderer>().material = materialList[2];
    }
}
```

Delete old broken prefabs add new Prefabs

When you update to the new toolkit your prefabs may be broken or missing.



To fix the missing prefabs click the prefab and the scripts and names of the prefab should appear. Replace the Missing Prefabs with the appropriate Prefabs.

Update to Immersive

With all the updates complete, the application should still work in the HoloLens device. And we can begin updating to immersive.

1. Create an Environment
2. Change Locomotion
3. Change Input

Gaze input is HoloLens's main way of interacting with objects in Mixed Reality. However, with immersive headsets gaze is not the most common interaction model used; motion controllers are. With Roomscale experiences users are used to having handheld devices to help the interact and navigate their virtual world, not with the position of their head.

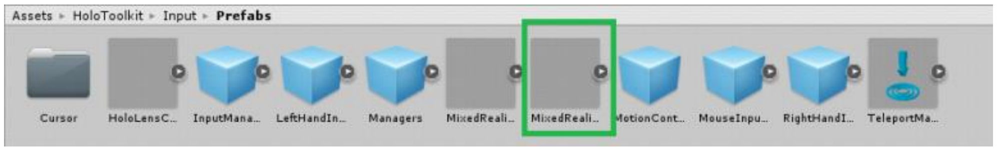
Adding Environments

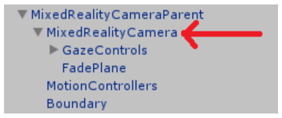
Immersive MR experiences are different than HoloLensMR experiences in that HoloLens doesn't need a virtual environment, the physical world itself is your physical environment. Unity has many free asset packages from their Asset Store that developers can use to create their world.

This part will take time. You will need to find textures and add them to materials, get models and add them to the scene.

Change Camera Prefab

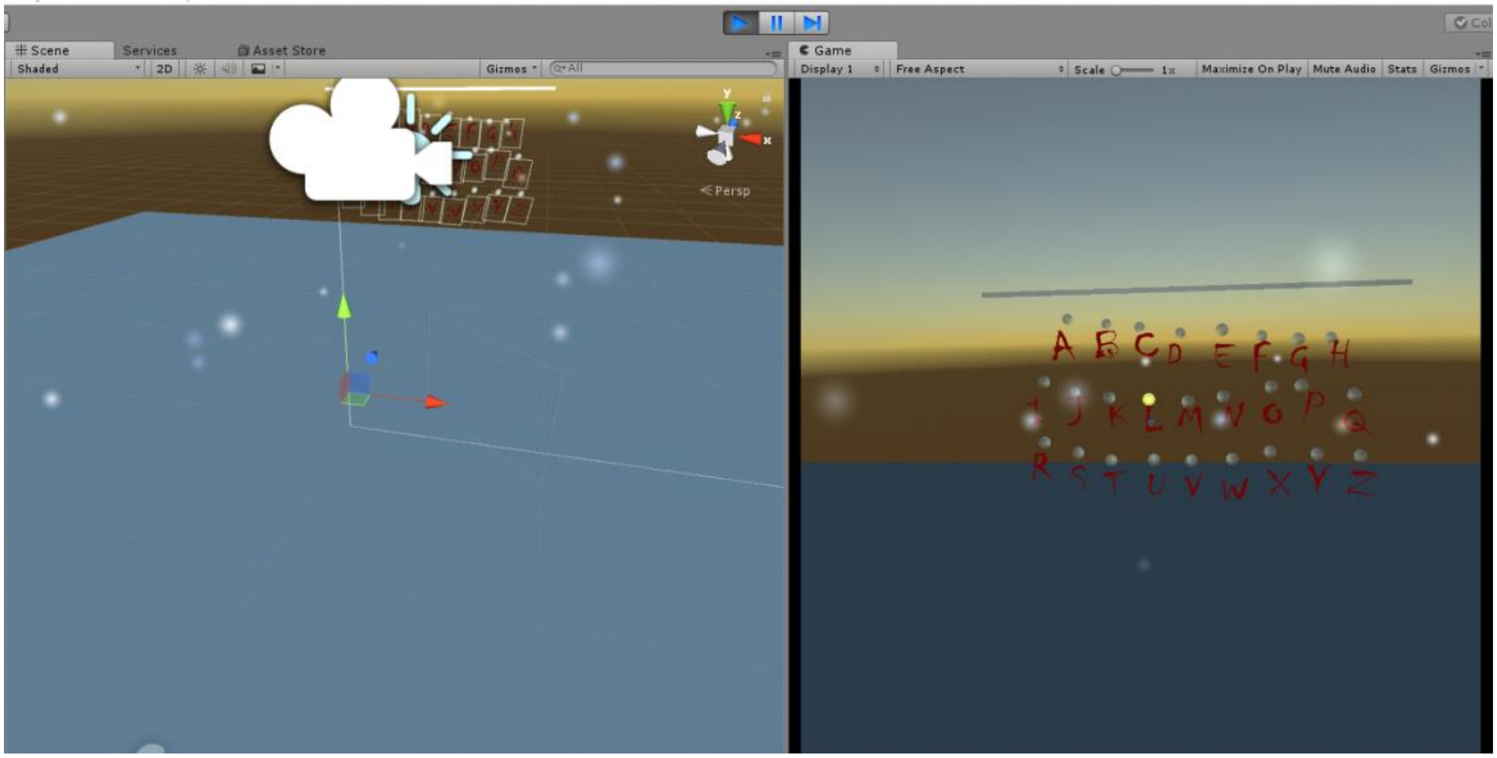
Mixed Reality Camera prefab works for Mixed Reality Immersive and HoloLens headsets. To use Room Scale delete the Mixed Reality Camera prefab and replace it with the Mixed Reality Camera Parent prefab.





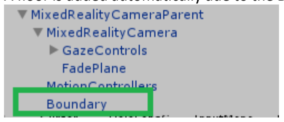
The Mixed Reality Camera Parent has the Mixed Reality Camera inside it as well as other Prefabs that enable teleportation and other input controls.

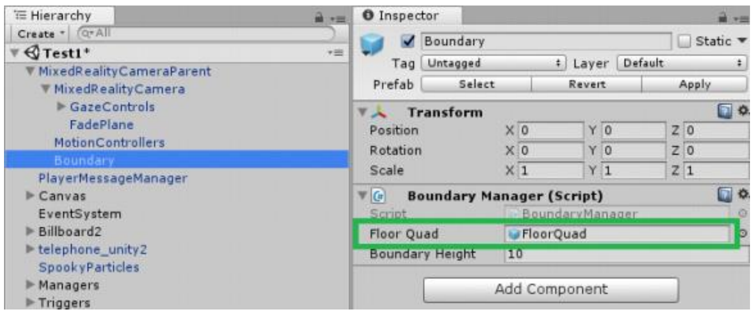
Play in Editor



Even without changing the environment the application should run within the editor - and the camera should be manipulated by the headset.

A floor is added automatically due to the Boundary component of the Prefab.





Test1*
MixedRealityCameraParent
MixedRealityCamera
GazeControls
FadePlane
MotionControllers
Boundary
PlayerMessageManager
Canvas
EventSystem
Billboard2
telephone_unity2
SpookyParticles
Managers
Triggers

Boundary — Static
Tag: Untagged — Layer: Default
Prefab: Select, Revert, Apply

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

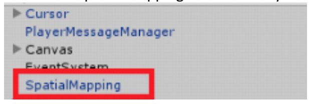
Boundary Manager (Script)
Script: BoundaryManager
Floor Quad: FloorQuad
Boundary Height: 10

Add Component

You can change whether or not the floor appears by changing the mesh renderer or the FloorQuad prefab, or by substituting your on Quad prefab.

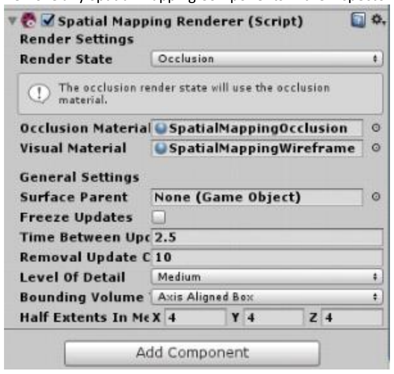
Remove Spatial Mapping Prefab and References

Delete the Spatial Mapping Prefab as they are no longer needed with fully immersive experiences.



Cursor
PlayerMessageManager
Canvas
EventSystem
SpatialMapping

Remove any Spatial Mapping Components in the inspector and Game Objects as well.



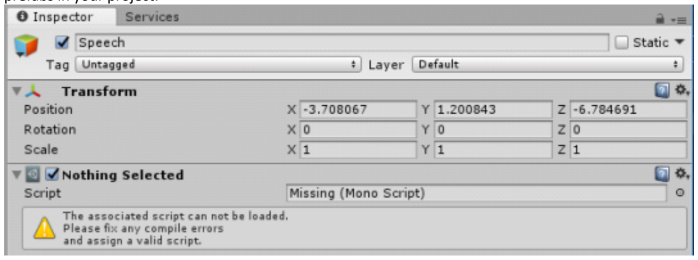
Spatial Mapping Renderer (Script)
Render Settings
Render State: Occlusion
The occlusion render state will use the occlusion material.
Occlusion Material: SpatialMappingOcclusion
Visual Material: SpatialMappingWireframe

General Settings
Surface Parent: None (Game Object)
Freeze Updates: ☐
Time Between Upc: 2.5
Removal Update C: 10
Level Of Detail: Medium
Bounding Volume: Axis Aligned Box
Half Extents In Me: X 4 Y 4 Z 4

Add Component

Voice Input

Voice Input uses a different input scripts now. Keyword Recognizer has been converted to Speech Handler. In your project the Keyword Recognizer and speech scripts may be no longer linked to their prefabs in your project.



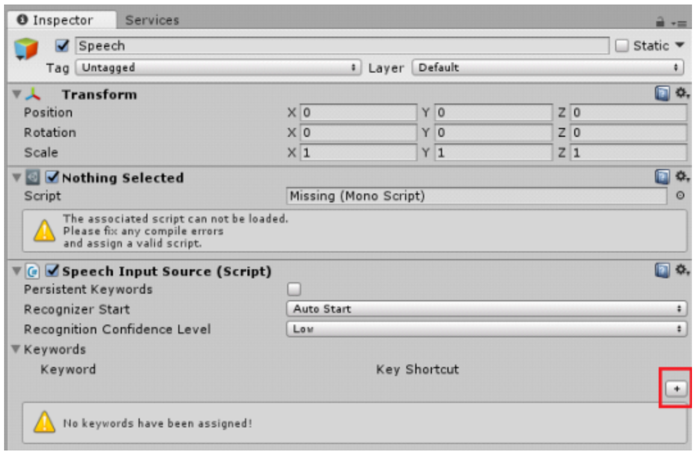
Speech — Static
Tag: Untagged — Layer: Default

Transform
Position X -3.708067 Y 1.200843 Z -6.784691
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Nothing Selected
Script: Missing (Mono Script)
The associated script can not be loaded. Please fix any compile errors and assign a valid script.

This is okay since the script will still be in your Project.

To use speech recognition you now need a Speech Input Source and a Speech Input Handler component added to your Speech Manager Gameobject. The Speech Input Source component determines which keywords will be detected.



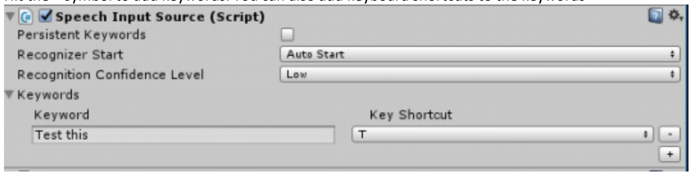
Speech — Static
Tag: Untagged — Layer: Default

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Nothing Selected
Script: Missing (Mono Script)
The associated script can not be loaded. Please fix any compile errors and assign a valid script.

Speech Input Source (Script)
Persistent Keywords: ☐
Recognizer Start: Auto Start
Recognition Confidence Level: Low
Keywords
Keyword — Key Shortcut
+
No keywords have been assigned!

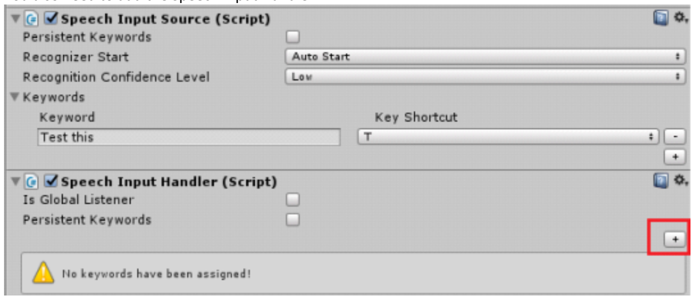
Hit the + symbol to add keywords. You can also add keyboard shortcuts to the keywords



Speech Input Source (Script)
Persistent Keywords
Recognizer Start: Auto Start
Recognition Confidence Level: Low
Keywords

Keyword	Key Shortcut
Test this	T

You also need to add the Speech Input Handler

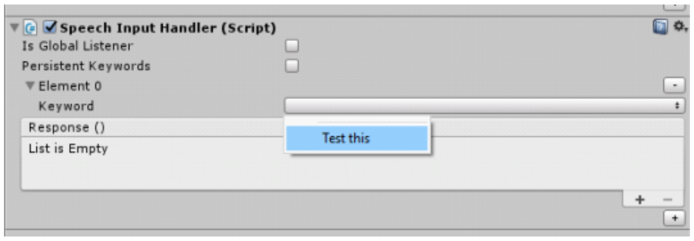


Speech Input Source (Script)
Persistent Keywords
Recognizer Start: Auto Start
Recognition Confidence Level: Low
Keywords

Keyword	Key Shortcut
Test this	T

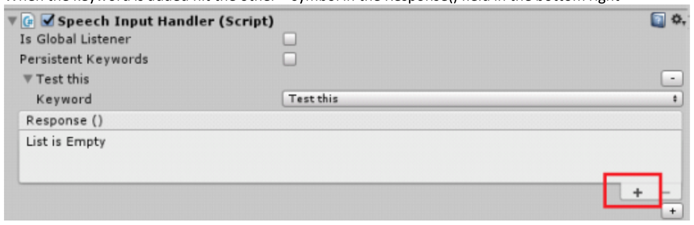
Speech Input Handler (Script)
Is Global Listener
Persistent Keywords
No keywords have been assigned!

Hit the + symbol to add how the keywords will be handled



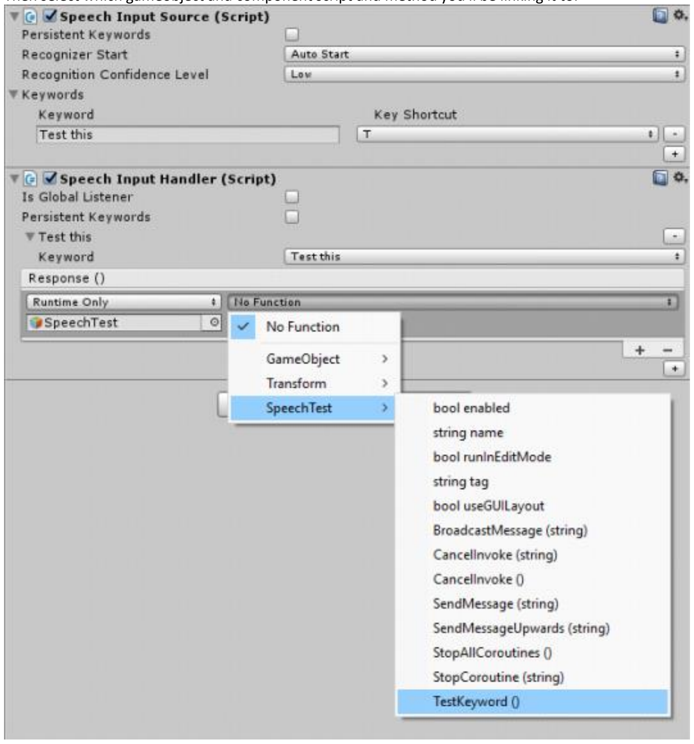
Speech Input Handler (Script)
Is Global Listener
Persistent Keywords
Element 0
Keyword
Response ()
List is Empty

When the keyword is added hit the other + symbol in the Response() field in the bottom right



Speech Input Handler (Script)
Is Global Listener
Persistent Keywords
Test this
Keyword: Test this
Response ()
List is Empty

Then select which gameobject and component script and method you'll be linking it to.



Speech Input Source (Script)
Persistent Keywords
Recognizer Start: Auto Start
Recognition Confidence Level: Low
Keywords

Keyword	Key Shortcut
Test this	T

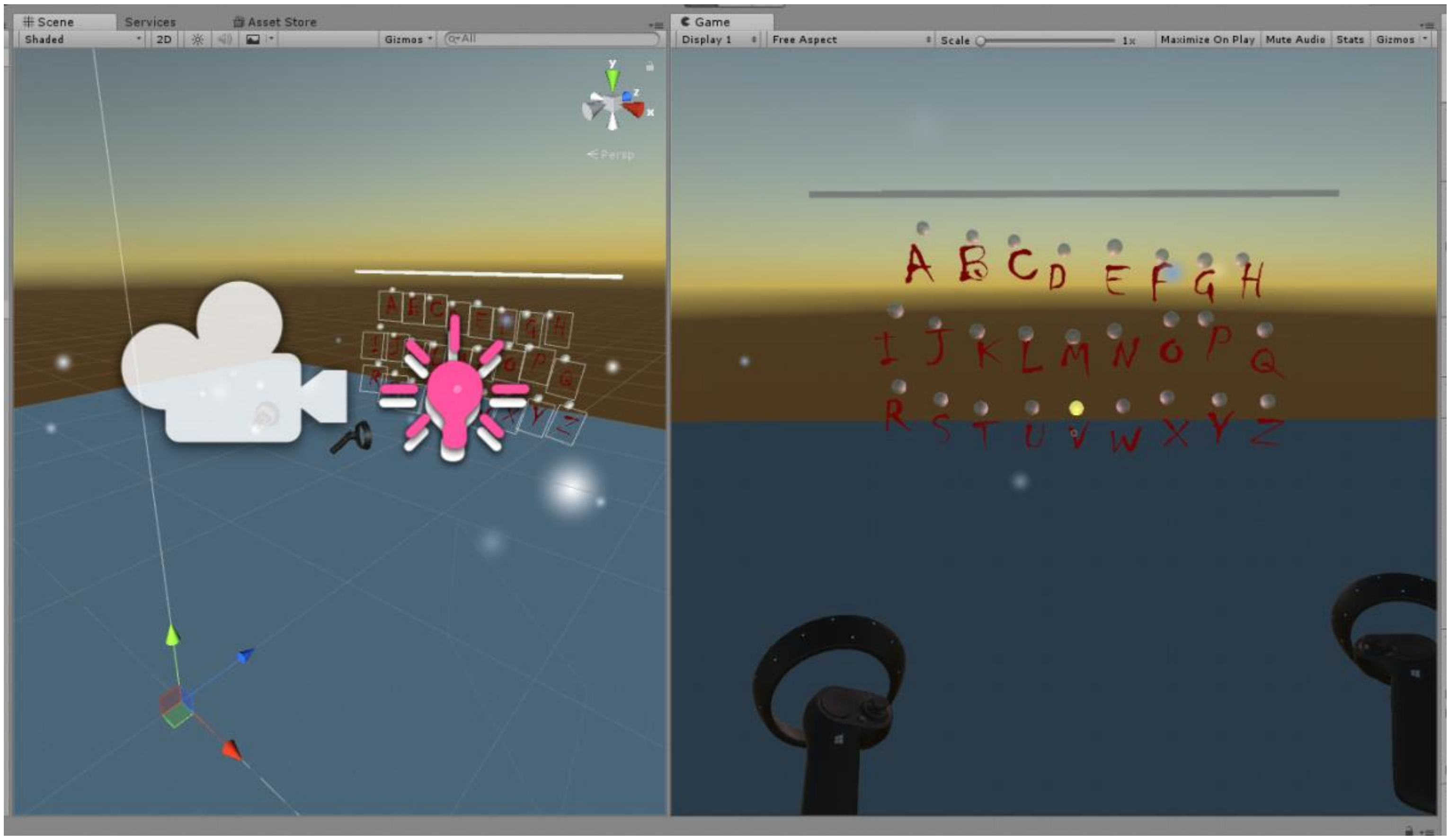
Speech Input Handler (Script)
Is Global Listener
Persistent Keywords
Test this
Keyword: Test this
Response ()
Runtime Only | No Function
SpeechTest

Linking the functionality like this should work the same way as keyword recognizer making the code run in the same manner.

Adding Controllers

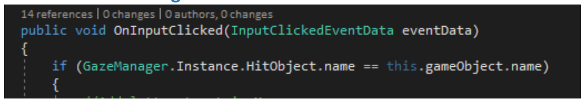
Pair controllers with your PC.
The pairing button is on the back under the battery cover towards the bottom of the controller.

When you hit the play button controllers should appear.



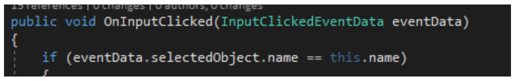
I you want to use a different model for the controllers, like your own model hands, tools, or items. You can insert your own prefabs into the Motion Controller Visualizer component under the MotionControllers prefabs.

Remove GazeManager Code



```
14 references | 0 changes | 0 authors, 0 changes
public void OnInputClicked(InputClickedEventData eventData)
{
    if (GazeManager.Instance.HitObject.name == this.gameObject.name)
    {
```

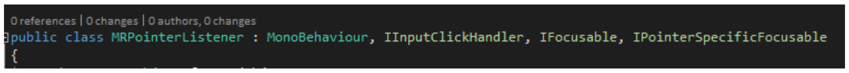
If you want focus specific information you now need to use the eventData parameter.



```
public void OnInputClicked(InputClickedEventData eventData)
{
    if (eventData.selectedObject.name == this.name)
```

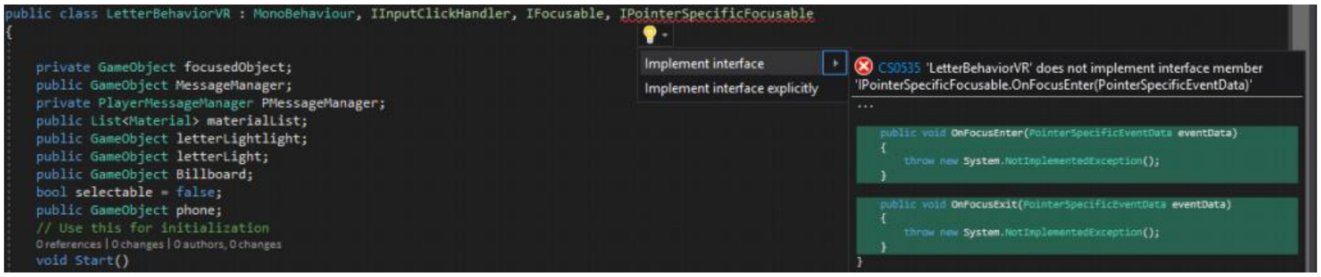
Trigger input

Because the MRTK works for both HoloLens and the immersive headsets there shouldn't be much to change for interactions if they were modularized in the code architecture. The main change will be to add IPointerSpecificFocusable so that the controllers can use focusable as well.



```
0 references | 0 changes | 0 authors, 0 changes
public class MRPointerListener : MonoBehaviour, IInputClickHandler, IFocusable, IPointerSpecificFocusable
{
```

Visual Studio will warn you to add the appropriate methods to the class now that it is extending IPointerSpecificFocusable



```
public class LetterBehaviorVR : MonoBehaviour, IInputClickHandler, IFocusable, IPointerSpecificFocusable
{

    private GameObject focusedObject;
    public GameObject MessageManager;
    private PlayerMessageManager PMessageManager;
    public List<Material> materialList;
    public GameObject letterLightlight;
    public GameObject letterLight;
    public GameObject Billboard;
    bool selectable = false;
    public GameObject phone;
    // Use this for initialization
    0 references | 0 changes | 0 authors, 0 changes
    void Start()
```

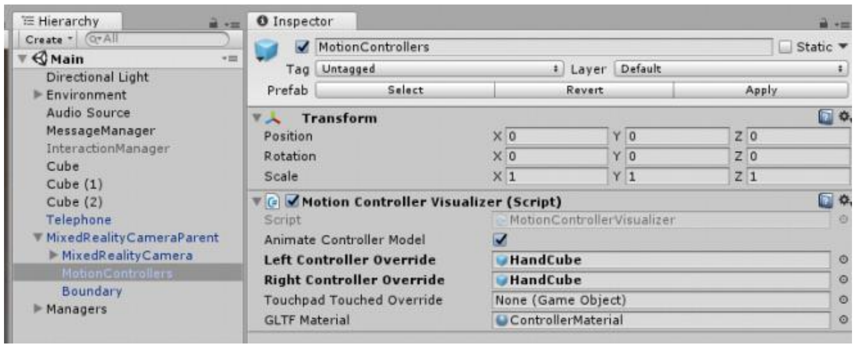
CS0535 'LetterBehaviorVR' does not implement interface member 'IPointerSpecificFocusable.OnFocusEnter(PointerSpecificEventData)'

```
...

    public void OnFocusEnter(PointerSpecificEventData eventData)
    {
        throw new System.NotImplementedException();
    }

    public void OnFocusExit(PointerSpecificEventData eventData)
    {
        throw new System.NotImplementedException();
    }
}
```

You can copy the same code snippet into these stubs from the OnFocusEnter() from Tap if you want the trigger and tap to have the same functionality.

Adding your own controller models

To add your own Controller models Select MotionControllers under the MixedRealityCameraParent Add in your own prefab to the Left Controller Override and Right Controller Override.



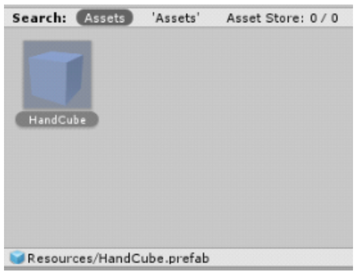
MotionControllers
Tag Untagged Layer Default
Prefab

Transform

Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Motion Controller Visualizer (Script)

Script	MotionControllerVisualizer
Animate Controller Model	✓
Left Controller Override	HandCube
Right Controller Override	HandCube
Touchpad Touched Override	None (Game Object)
GLTF Material	ControllerMaterial



Resources/HandCube.prefab