



TECH 1711 - Mixed Reality Studio

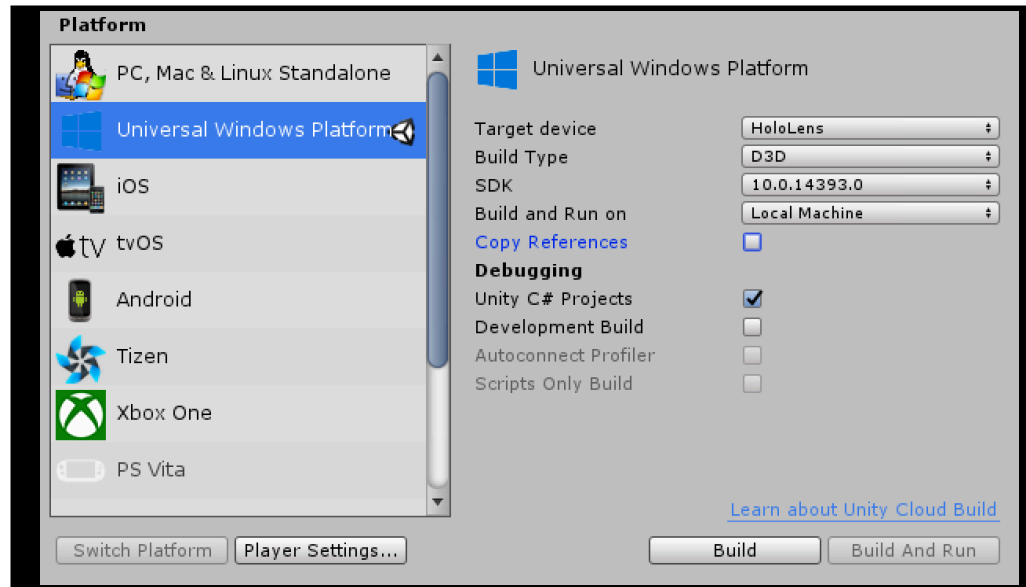


MixedReality Toolkit

(formerly HoloToolkit)

<https://github.com/Microsoft/MixedRealityToolkit-Unity>

<https://github.com/Microsoft/MixedRealityToolkit-Unity>

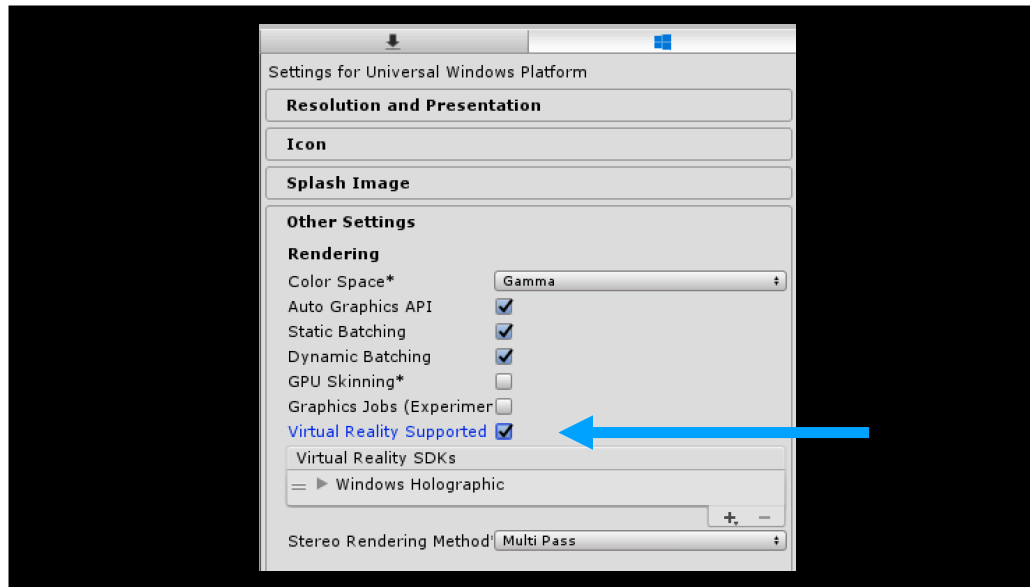


Unfortunately, we can only build to HoloLens on PC.

Go to Build Settings (File > Build Settings or Ctrl-Shift-B)

- Select Universal Windows Platform and click on “Switch Platform” if the Unity logo is not already next to it.
- set *Target Device* to *HoloLens*
- set *Build Type* to *D3D*
- *SDK* should be set to *Latest Installed* or match the version above
- *Build and Run on Local Machine*
- Make sure *Unity C# Projects* is checked

No need to hit “Build” yet. We’re just setting up our project format. Just close the window when you make these changes.



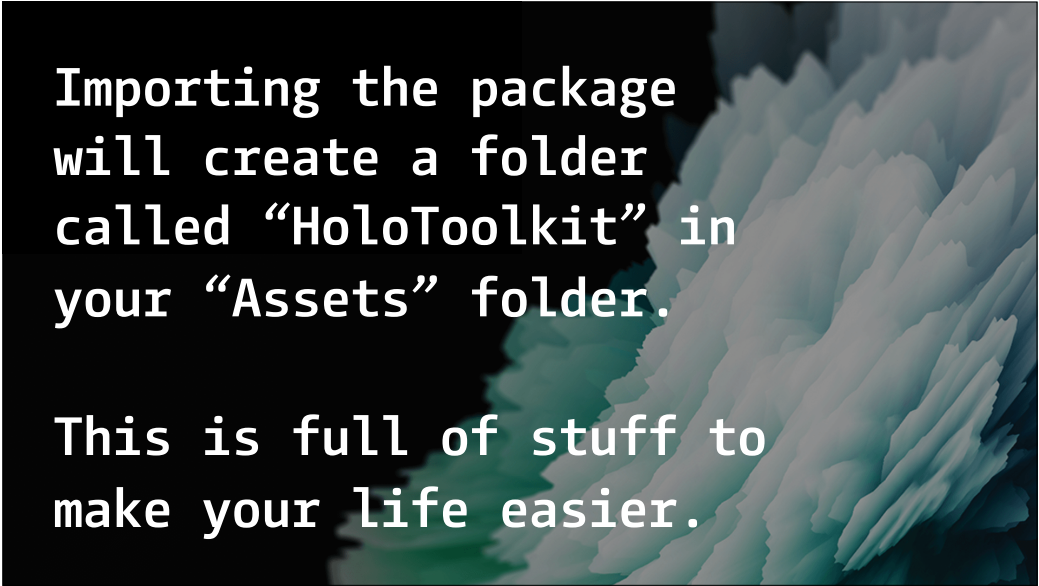
Click on “*Player Settings*”... or go to *Edit > Project Settings > Player*

(FYI - “Player” in this case doesn’t mean the user/player, but rather Unity itself as in “VCR Player”)

Make sure “Virtual Reality Supported” is checked and that “Windows Holographic” is the only option.

Useful components in HoloToolkit





Importing the package
will create a folder
called “HoloToolkit” in
your “Assets” folder.

This is full of stuff to
make your life easier.

Prefabs to include:

- HoloLensCamera
 - (Make sure to delete "MainCamera" from your scene)
- Input Manager
- InteractiveMeshCursor

Easy-to-use Interaction Scripts

`Assets/HoloToolkit/SpatialMapping/Scripts/TapToPlace`

Attach to a `GameObject` to allow you to pick it up and move it around with an air-tap. Hololens will try to drop the object at the point you're looking at.

`Assets/HoloToolkit/Input/Scripts/Utilities/Interactions/HandDraggable`

Attach to a `GameObject` to allow you to pick it up and hold it with an air-tap and move it through space as long as you hang on to it

More Easy-to-use Interaction Scripts!

Assets/HoloToolkit/Utilities/Scripts/**Billboard**

Attach to a GameObject to make it always rotate to face you.

Assets/HoloToolkit/Utilities/Scripts/**Tagalong**

Attach to a GameObject and it will follow your view around, ensuring that some part of it is always visible. Combine with **Billboard** for a great approach to menus and instructions.

Advanced Interaction Scripts: Events

What if you want to do trigger your own behaviors on taps and gazes?

HoloToolkit/Input/Scripts/InputHandlers/

These are scripts that define events you can listen for. You can't add these to objects directly...

Advanced Interaction Scripts: Events

`HoloToolkit/Input/Scripts/InputHandlers/`

These scripts define the methods you have to have in your script to listen for these events.

Examples on the github (with comments)

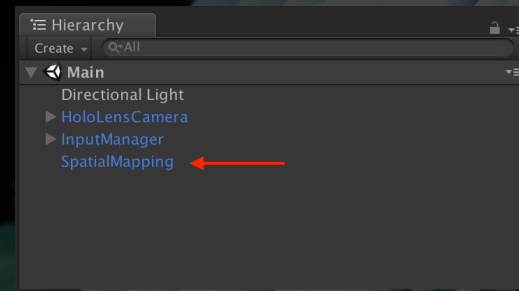
Spatial Mapping

As soon as you start a program on Hololens, it begins to scan its surroundings and make this data available to you through the **Spatial Mapping** API.

Spatial Mapping

HoloToolkit makes it super easy to use this info in our programs!

Drag the **SpatialMapping** prefab into the root of your scene

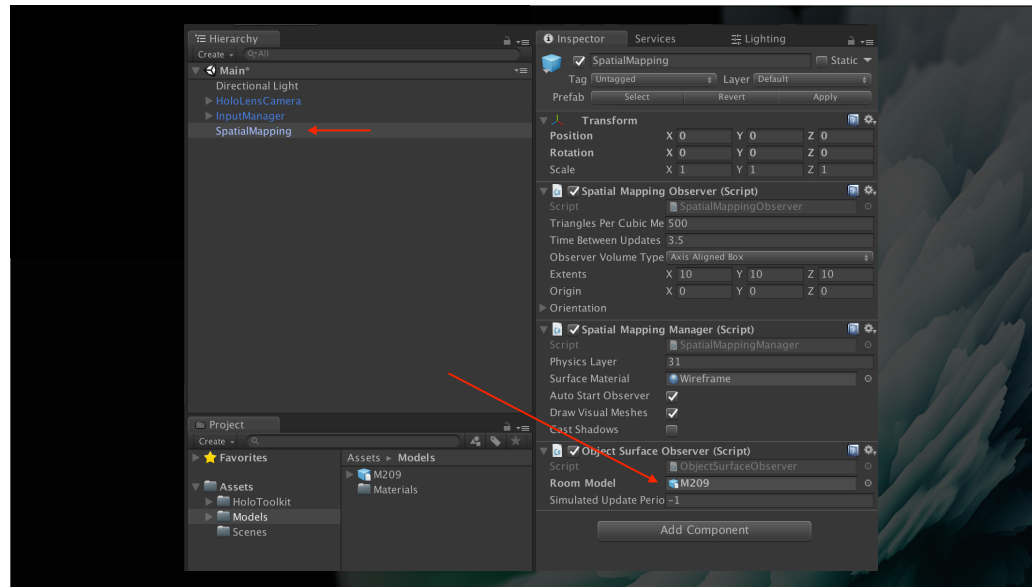


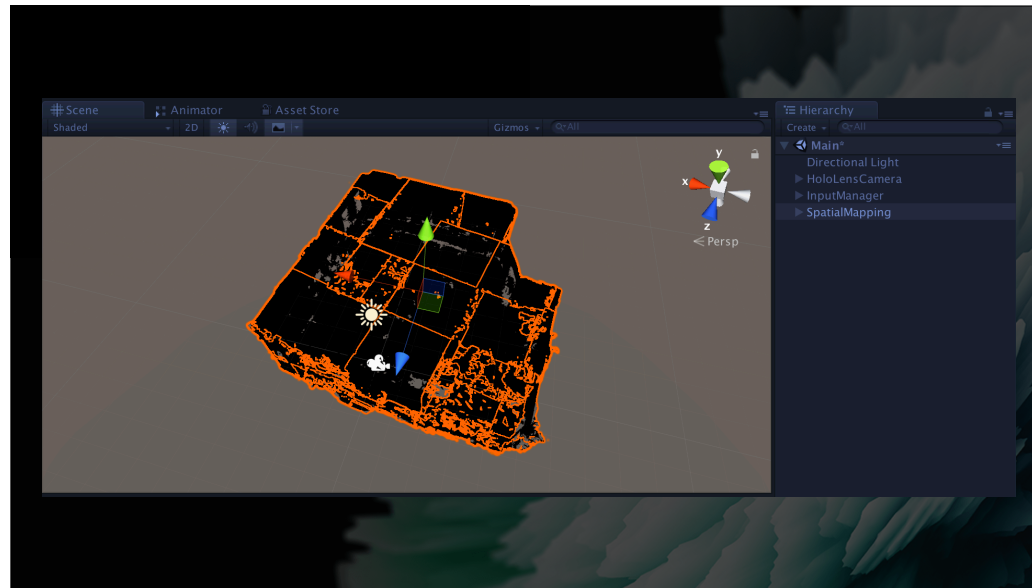
Spatial Mapping

On an actual device, you'd be good to go. Unity will start seeing the spatial information as it comes in.

To use this in our *simulator* though, we have to provide a room model. Any OBJ model will work.

An OBJ of M209 is on the github, pulled directly from one of our Hololenses (so it is *exactly* how it would see our room)



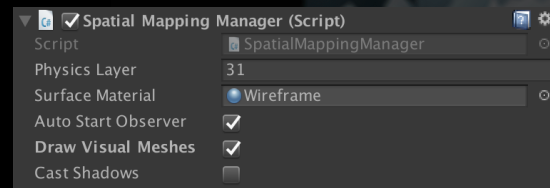


Spatial Mapping

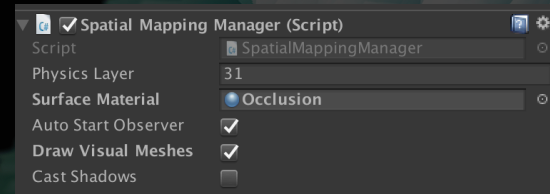
By Default, **SpatialMapping** uses a black-and-white wireframe material, but we usually don't want to see this.

HoloToolkit comes with a material called **Occlusion** which is *invisible* but still obscures things behind it. Most of the time you want to tell **Spatial Mapping** to use this material.

Default:



Occlusion:



Global Event Listeners

Earlier we talked about how HoloLens uses **events** to “listen” for events instead of “asking” 60x per second if an interaction has happened.

We went over how to add create your own air-tap “listener” and detect if the user tapped while they were looking at something.

Global Event Listeners

But what if we want to “listen” for an interaction *globally*?

That is, we want to react to a tap or a hold when we are *not* looking at a specific object.

Global Event Listeners

The script is the exact same as the normal event listener with one extra line:

```
void Start() {  
    // This line registers this script as the "fallback" event-  
    // handler for events of this type not bound to another object  
    InputManager.Instance.PushFallbackInputHandler(gameObject);  
    // Your other Start stuff goes here...  
}
```

You can find the whole script on the [github page](#)



TECH 1711 - Mixed Reality Studio