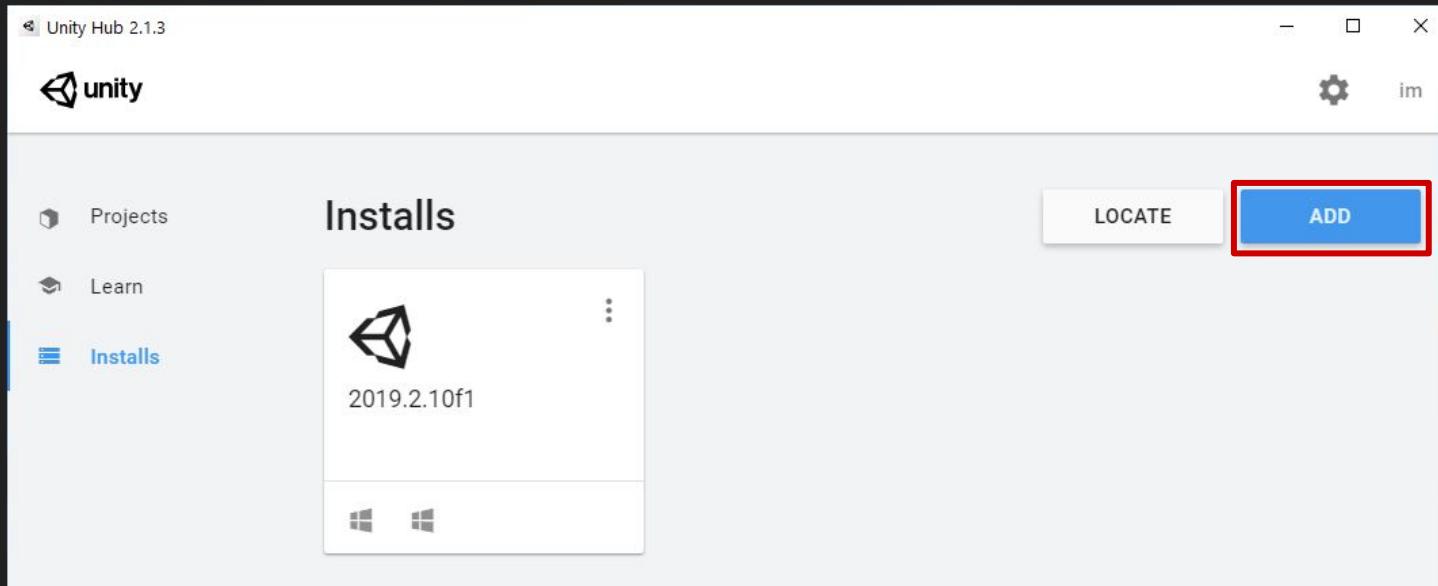


Setting up the development environment

1. Download & install Unity Hub

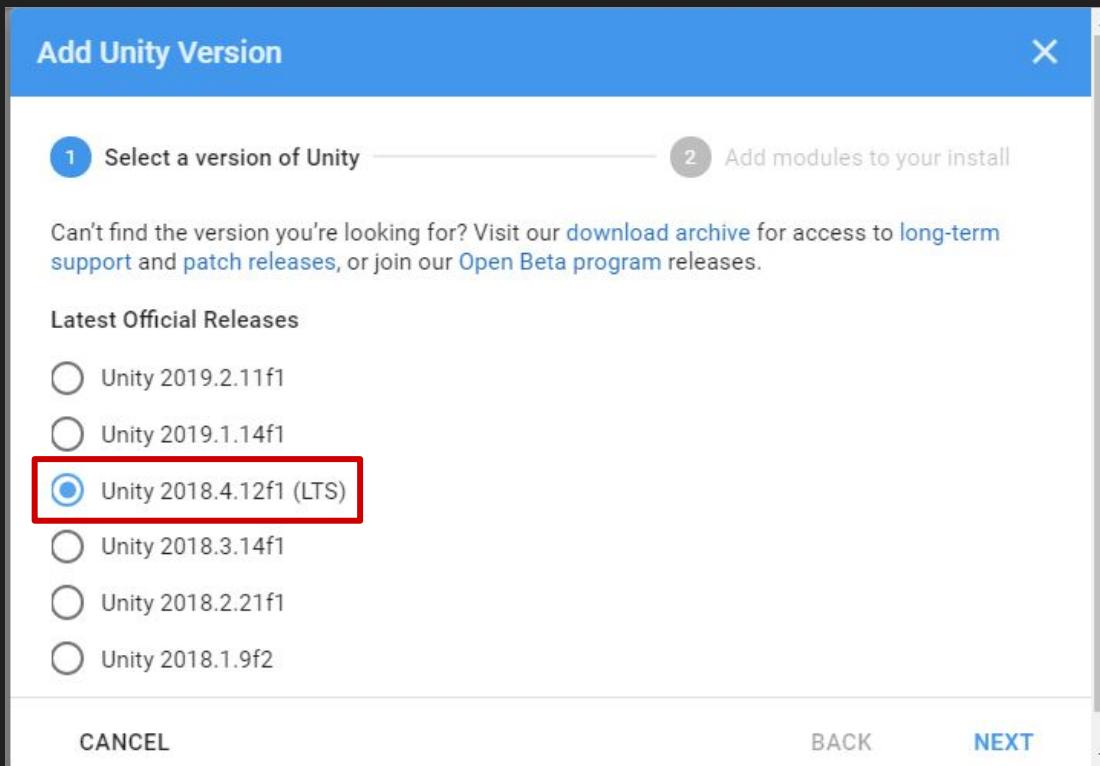
<https://unity3d.com/get-unity/download>

2. Open Unity Hub to install a specific version Unity in PC



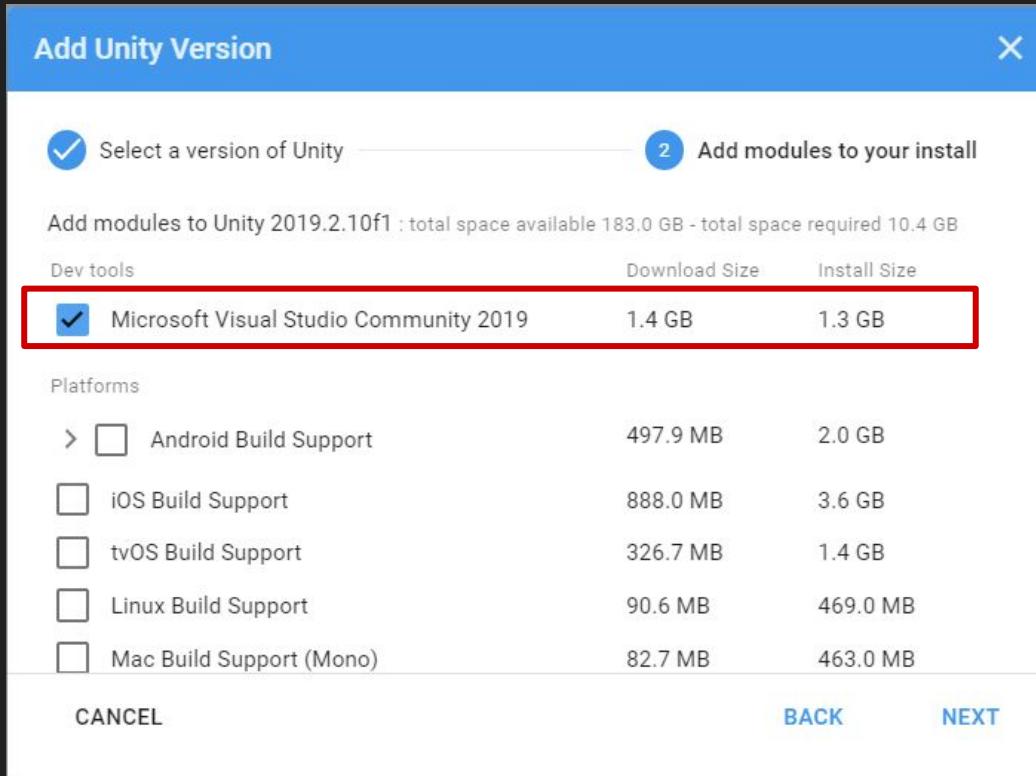
Setting up the development environment

3. Select Unity 2018.4.x (LTS) or 2019.2.x (Latest)



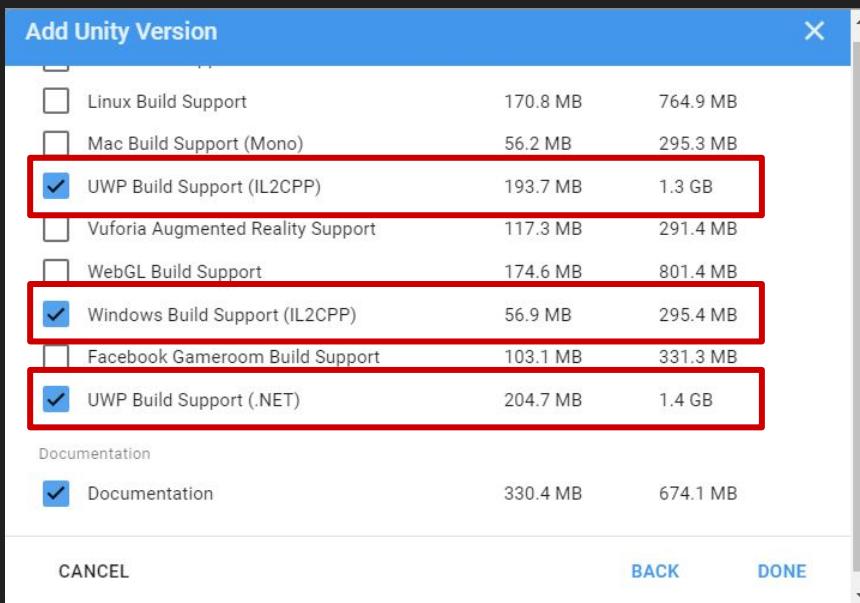
Setting up the development environment

4. Check Microsoft Visual Studio Community 2019

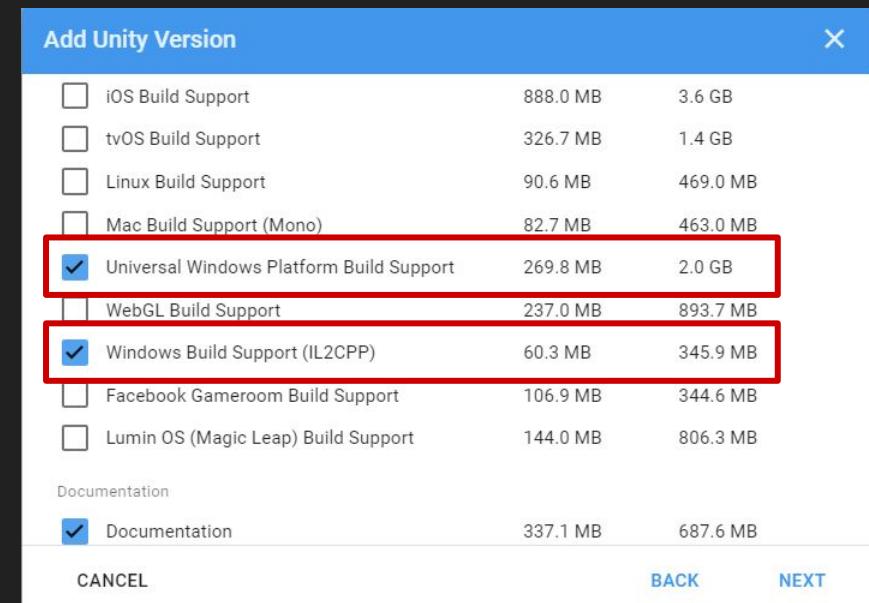


Setting up the development environment

5. Check UWP Build Support (IL2CPP), Windows Build Support (IL2CPP), and UWP Build Support (.NET) (**Optional**)



Unity 2018.4.x (LTS)



2019.2.x (Latest)

Supplement: IL2CPP vs .NET Scripting

.NET
Scripting



IL: Intermediate Language

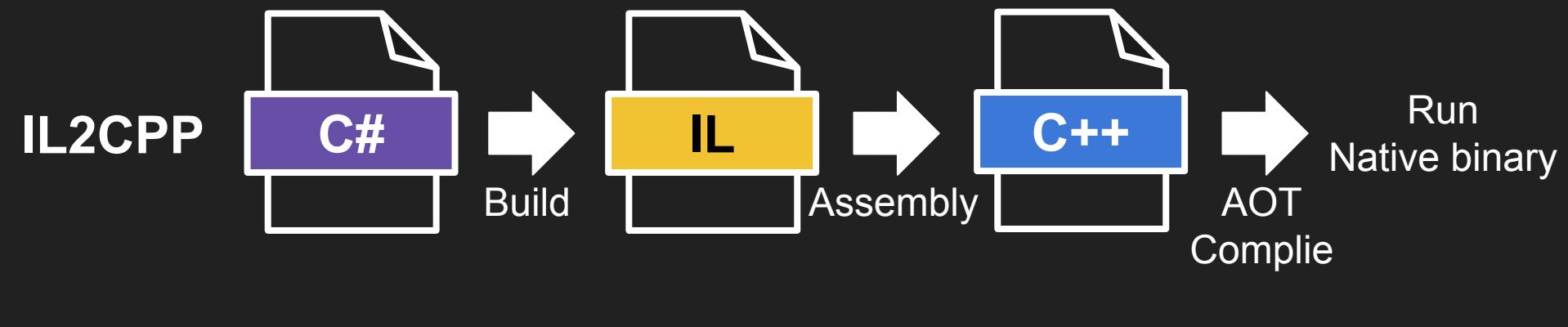
ILR: Intermediate Language Runtime

JIT: Just In Time

Converting projects from mixed mode to pure intermediate language

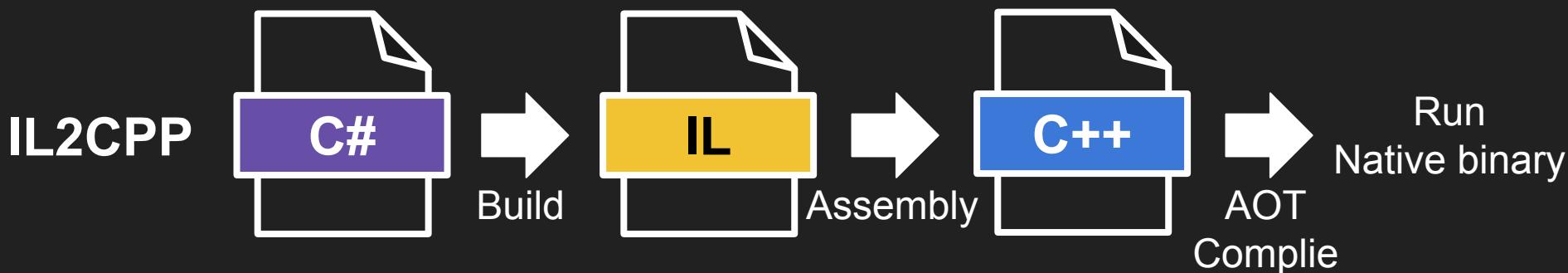
<https://docs.microsoft.com/en-us/cpp/dotnet/converting-projects-from-mixed-mode-to-pure-intermediate-language>

Supplement: IL2CPP vs .NET Scripting



AOT: Ahead of Time

Supplement: IL2CPP vs .NET Scripting



AOT: Ahead of Time

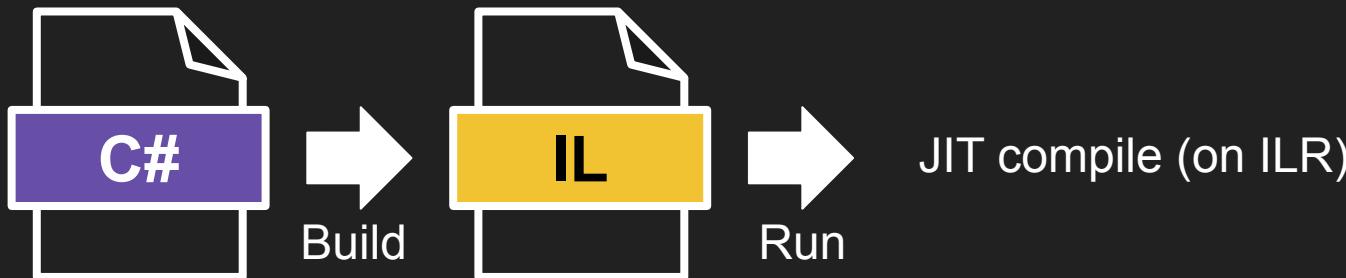
IL2CPP **increases** the **performance** of your HoloLens app
but the **build time** is **longer** compared with .NET Scripting

Unity Documentation: IL2CPP

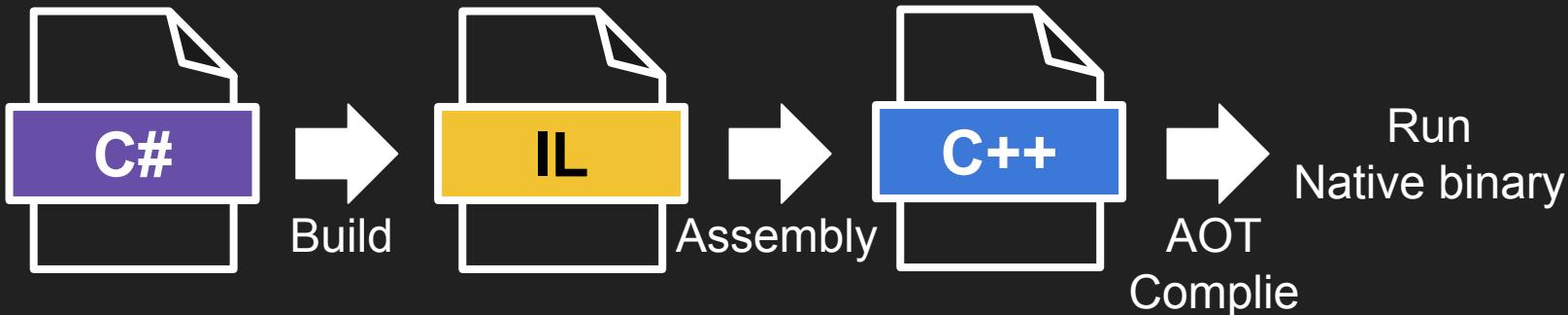
<https://docs.unity3d.com/2019.2/Documentation/Manual/IL2CPP.html>

Supplement: IL2CPP vs .NET Scripting

.NET
Scripting



IL2CPP



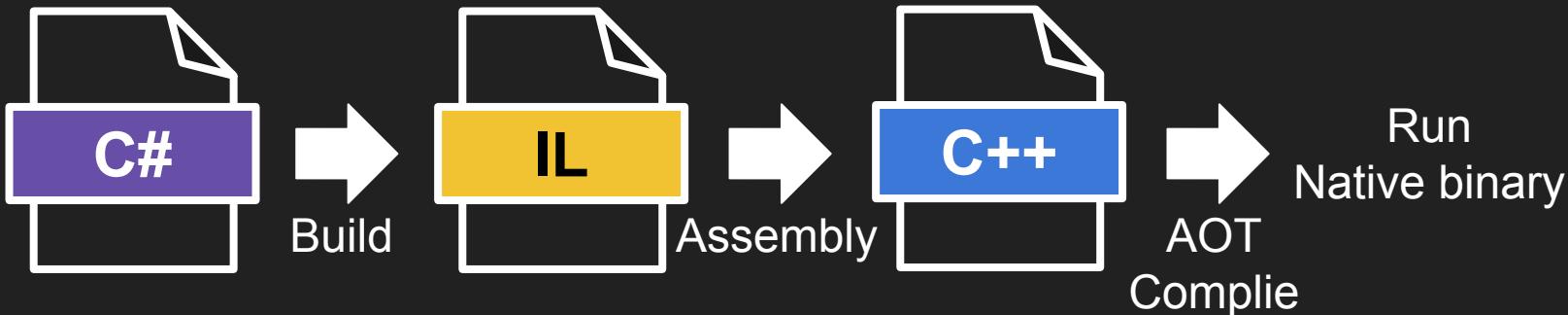
The .NET scripting backend is deprecated and will be removed in a future release of Unity.
<https://docs.unity3d.com/2018.4/Documentation/Manual/windowsstore-dotnet.html>

Supplement: IL2CPP vs .NET Scripting

.NET
Scripting



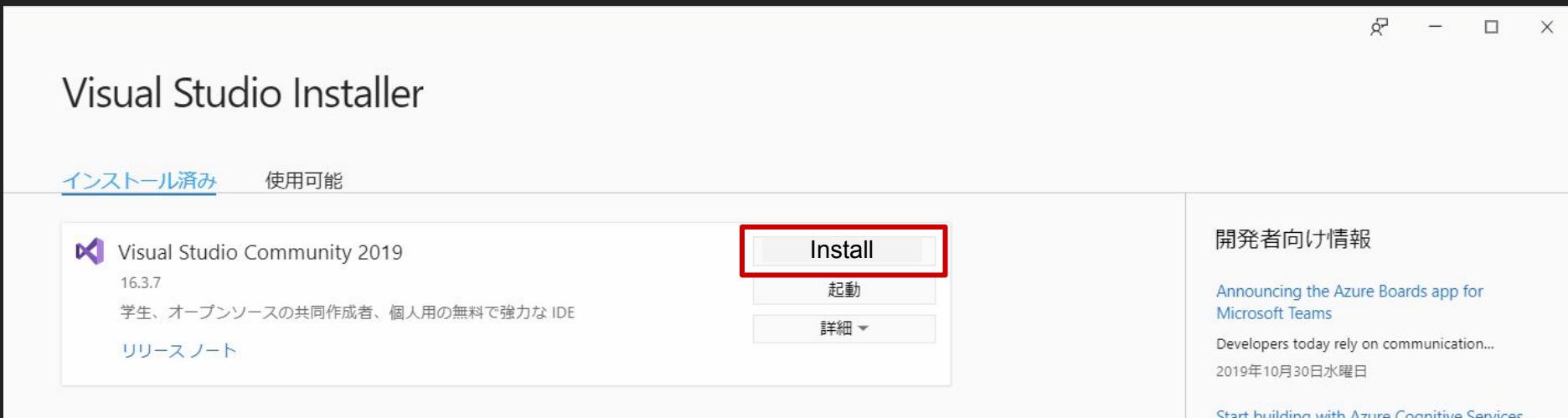
IL2CPP



The .NET scripting backend is deprecated and will be removed in a future release of Unity.
<https://docs.unity3d.com/2018.4/Documentation/Manual/windowsstore-dotnet.html>

Setting up the development environment

6. Install Visual Studio Community 2019.



Setting up the development environment

7. Check Game development with Unity and **checkout Unity 64-bit Editor**

Installing — Visual Studio Community 2017 — 15.7.4

Workloads Individual components Language packs Installation locations

Office/SharePoint development

Create Office and SharePoint add-ins, SharePoint solutions, and VSTO add-ins using C#, VB, and JavaScript.

Mobile & Gaming (5)

Mobile development with .NET

Create cross-platform applications for iOS, Android or Windows using Xamarin.

Game development with Unity

Create 2D and 3D games with Unity, a powerful cross-platform development environment.

Optional Unity 2017.2 64-bit Editor

Summary

- > Visual Studio core editor
- ✓ Game development with Unity Included
 - ✓ Visual Studio Tools for Unity
 - ✓ C# and Visual Basic

Setting up the developmental environment

8. Check Universal Windows Platform Development

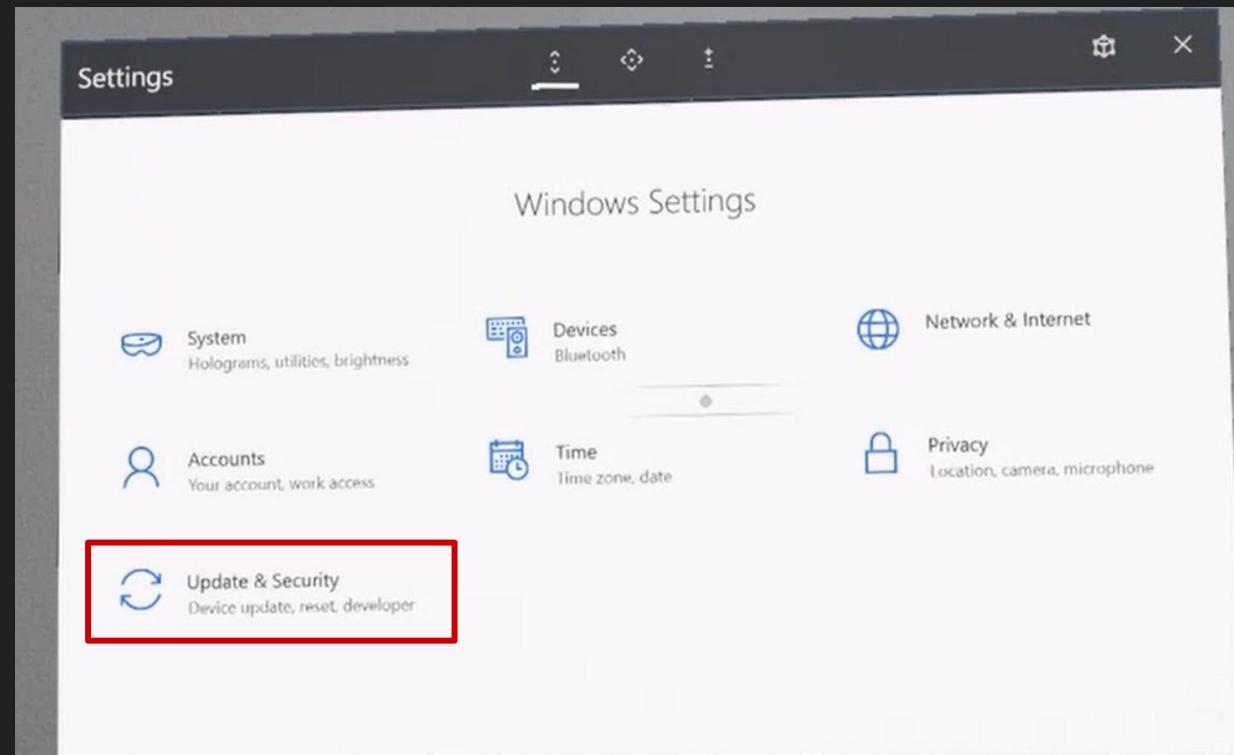
- USB device connection
- C++ (v142) Universal Windows Platform Development
- Windows 10 SDK



Windows Device Portal

Setting up HoloLens

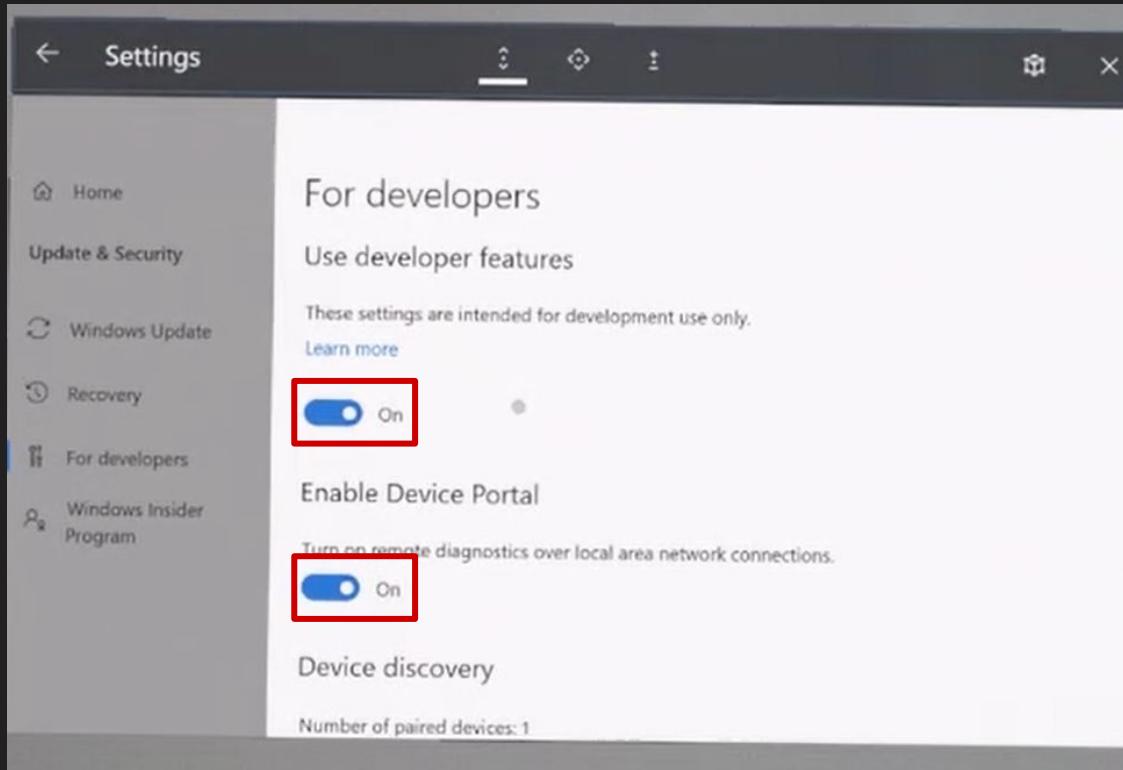
- Open [Settings] -> [Update] -> [For developers] on HoloLens



Windows Device Portal

Setting up HoloLens

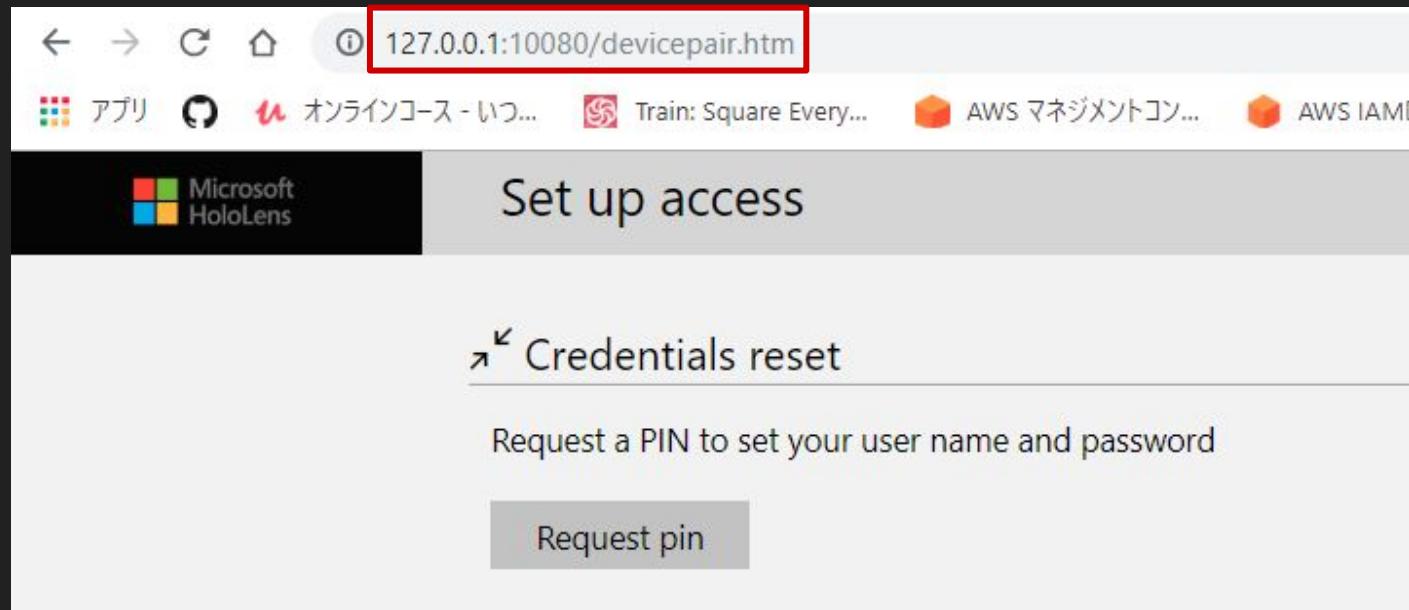
- Open [Settings] -> [Update] -> [For developers] on HoloLens
- Enable [Use developer features] & [Enable Device Portal]



Windows Device Portal

Setting up PC

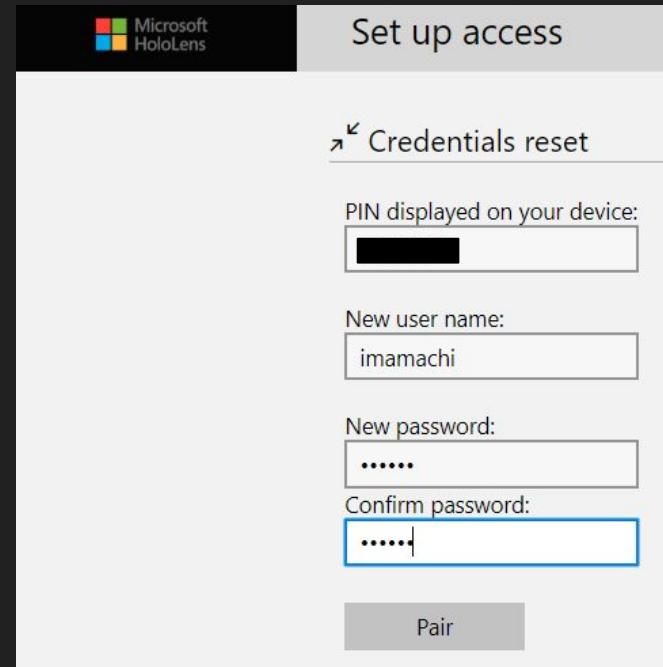
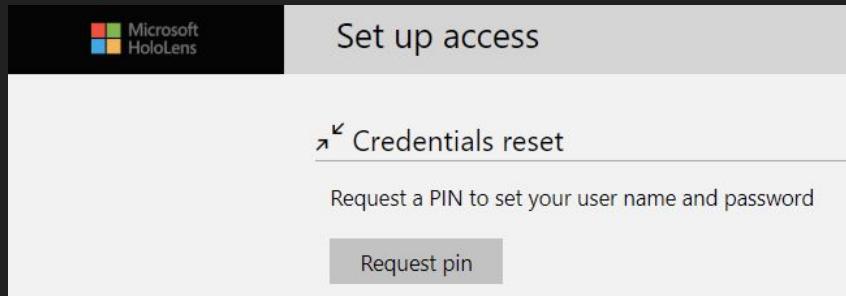
- In a web browser on your PC, enter the IP address of the HoloLens.
 1. <https://192.168.xx.xx/devicepair.htm> (Local Network)
 2. <https://127.0.0.1:10080/devicepair.htm> (USB Connection with HoloLens)



Windows Device Portal

Setting up PC

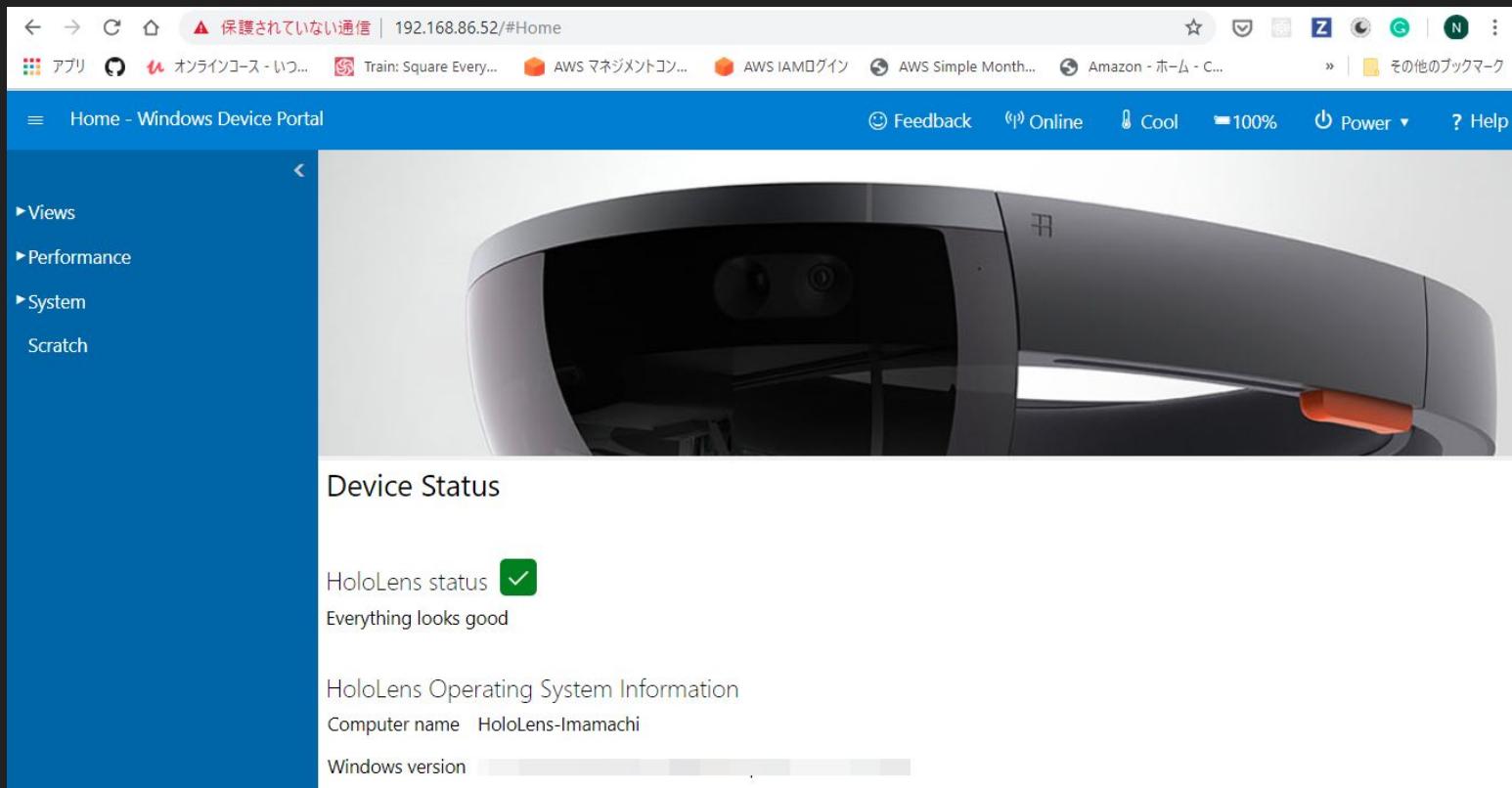
- Click Request pin
- Enter [PIN displayed on your HoloLens], [New username], [New password]



Windows Device Portal

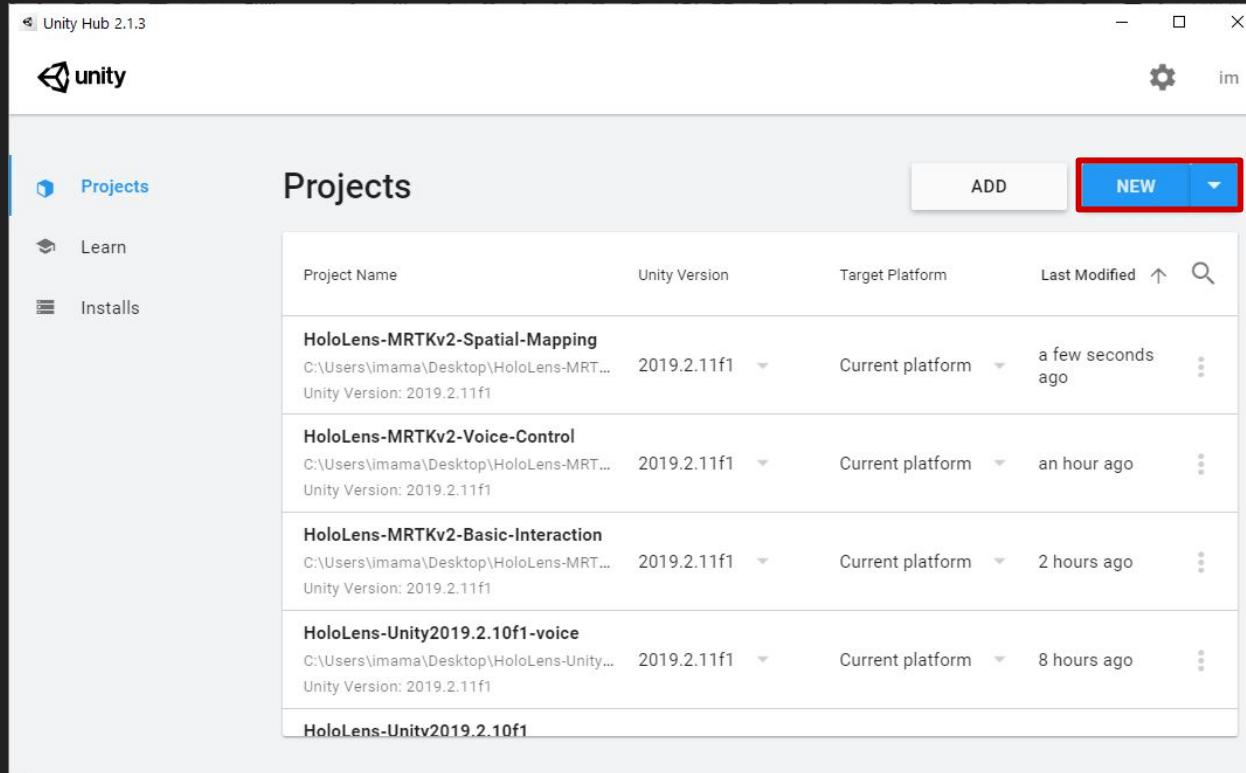
Setting up PC

- Go to 192.168.xx.xx or 127.0.0.1:10080 to open Device Portal for HoloLens



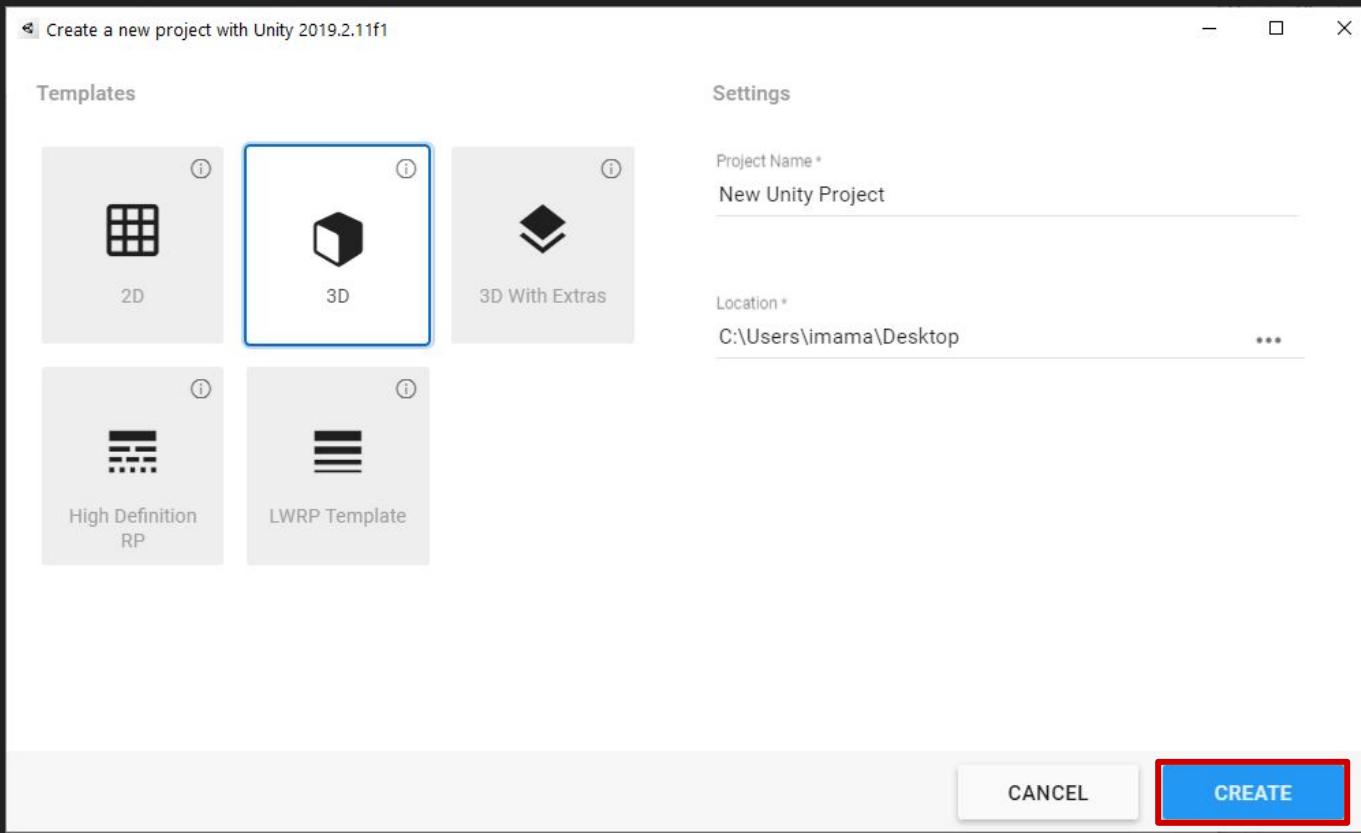
Create your first HoloLens app

- Open Unity Hub
- Click “NEW button” in project page.



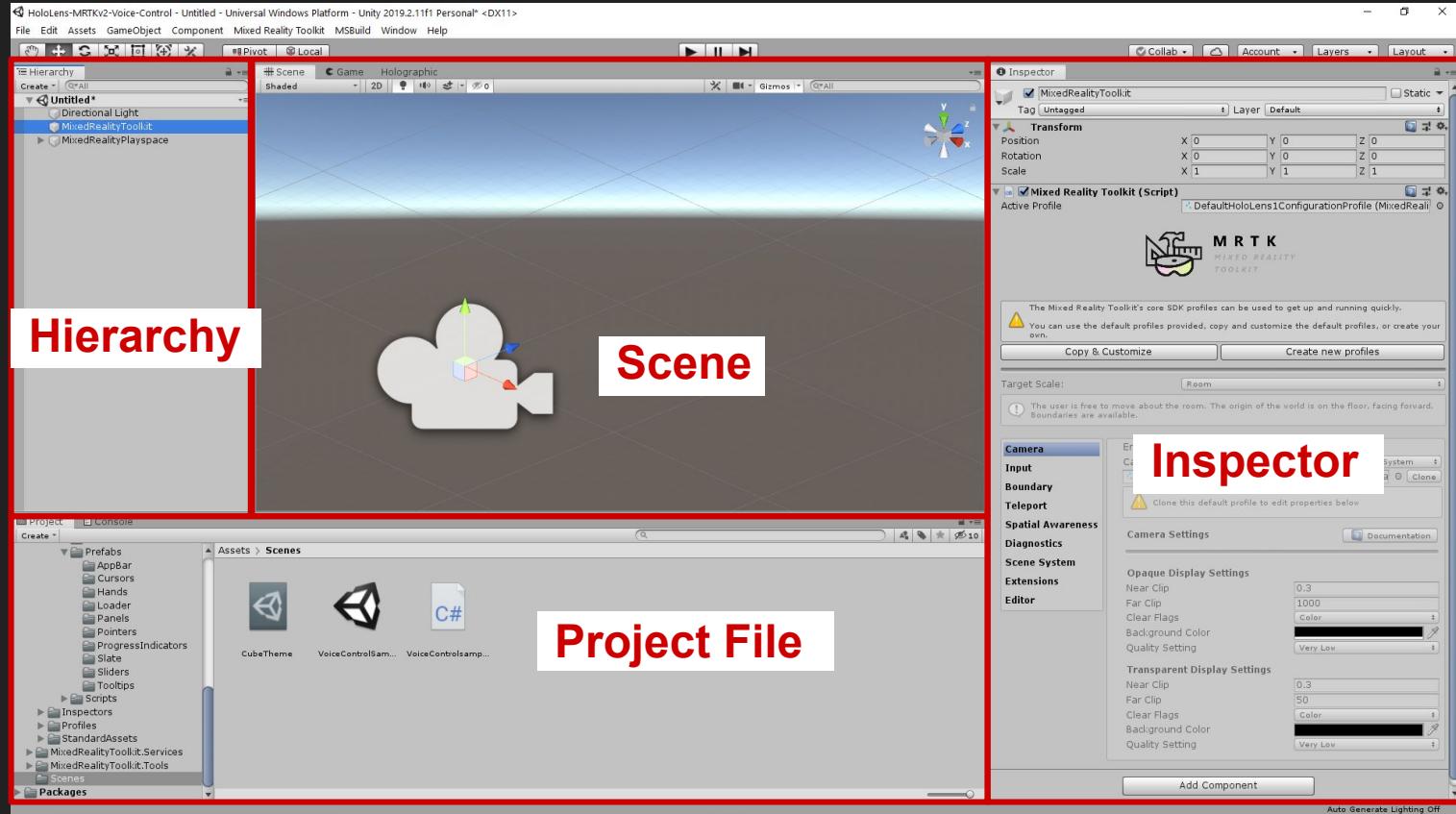
Create your first HoloLens app

- Select 3D, input Project name & Location, and click “CREATE button”



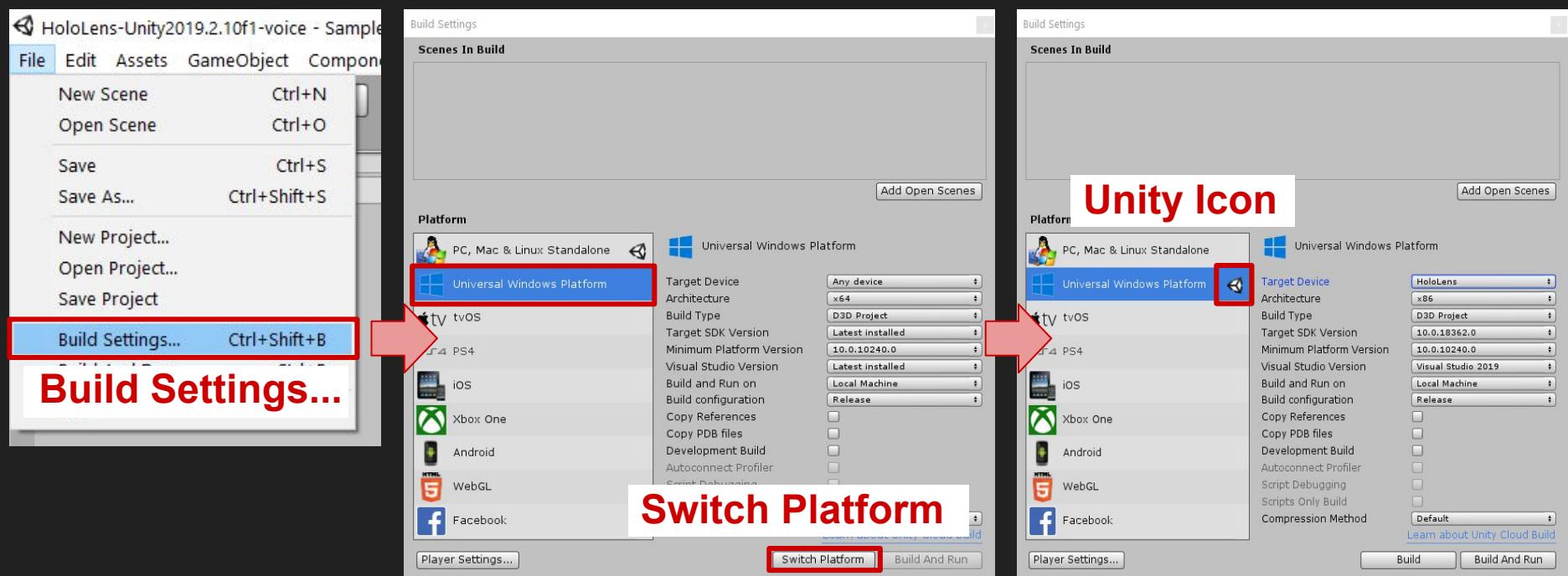
Create your first HoloLens app

- Open Unity Editor



Create your first HoloLens app

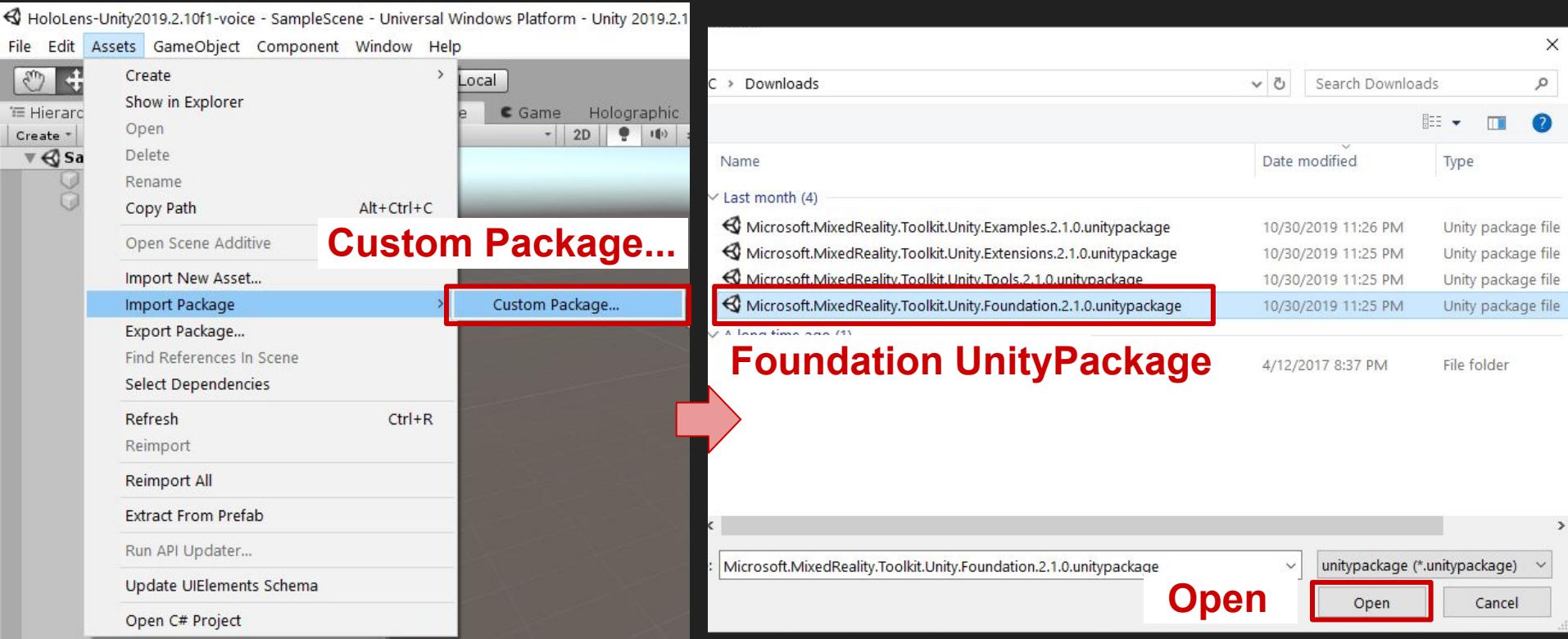
- Change a build platform to **Universal Windows Platform**



IMPORTANT!! If you do not switch to UWP before importing MRTKv2, your projects will be sometimes broken or Unity settings for HoloLens will be broken in my experience...

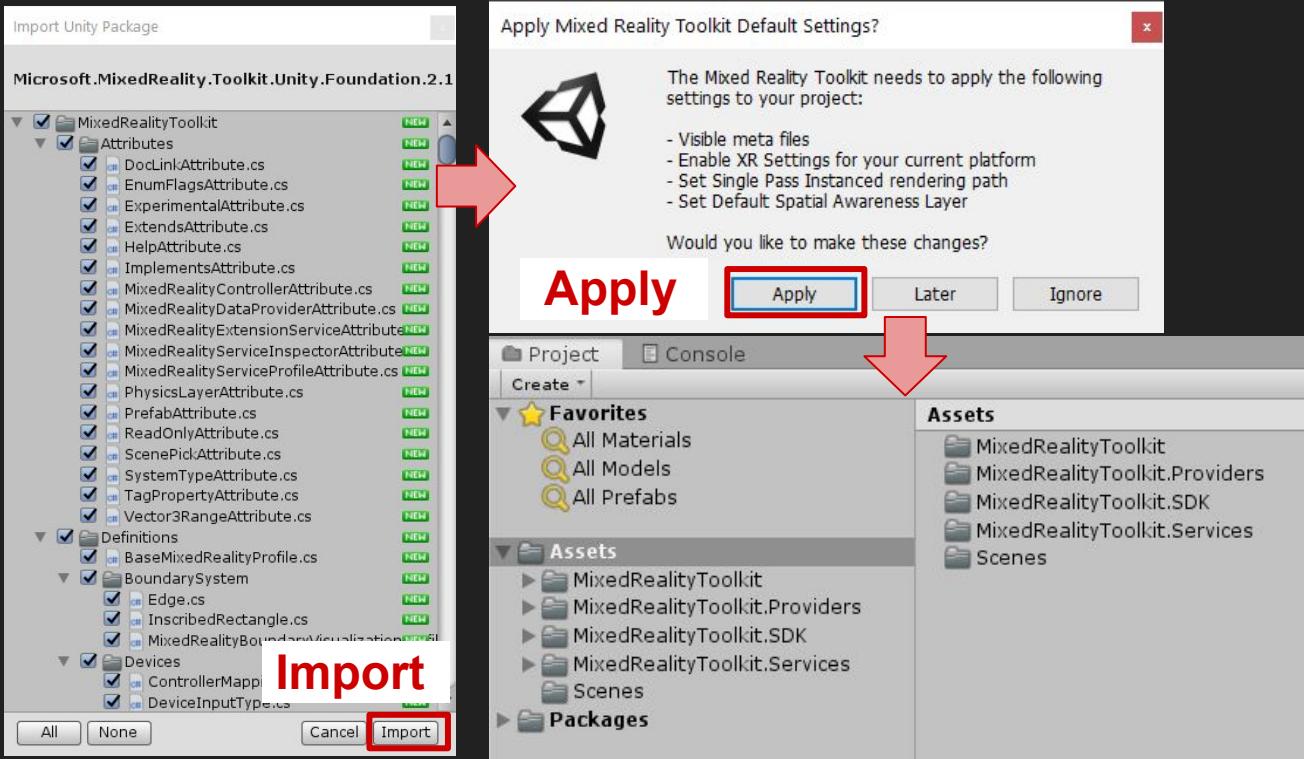
Create your first HoloLens app

- Select [Assets] -> [Import Package] -> [Custom Package]
- Select Microsoft.MixedReality.Toolkit.Unity.**Foundation**.2.1.0.unitypackage



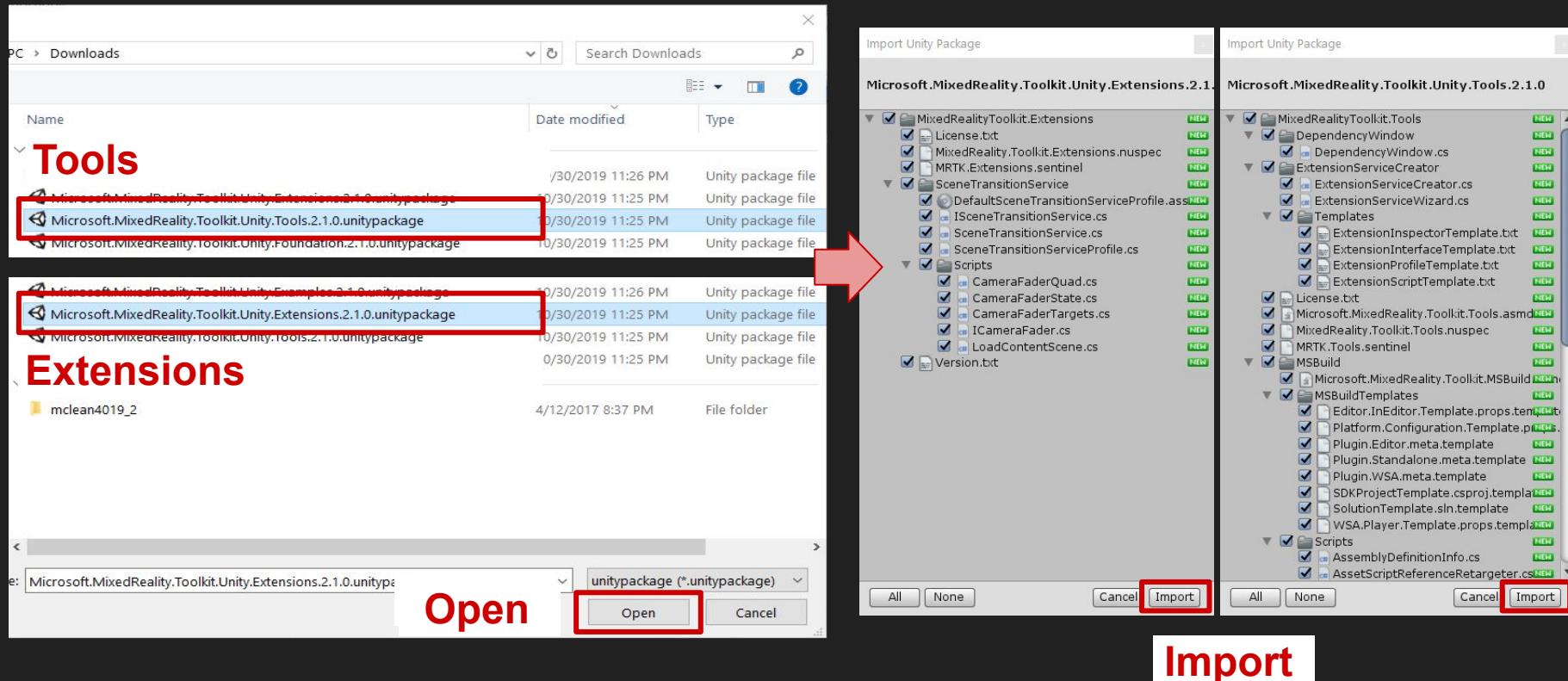
Create your first HoloLens app

- Import Unity Package into your project
- Apply Mixed Reality Toolkit Default Settings



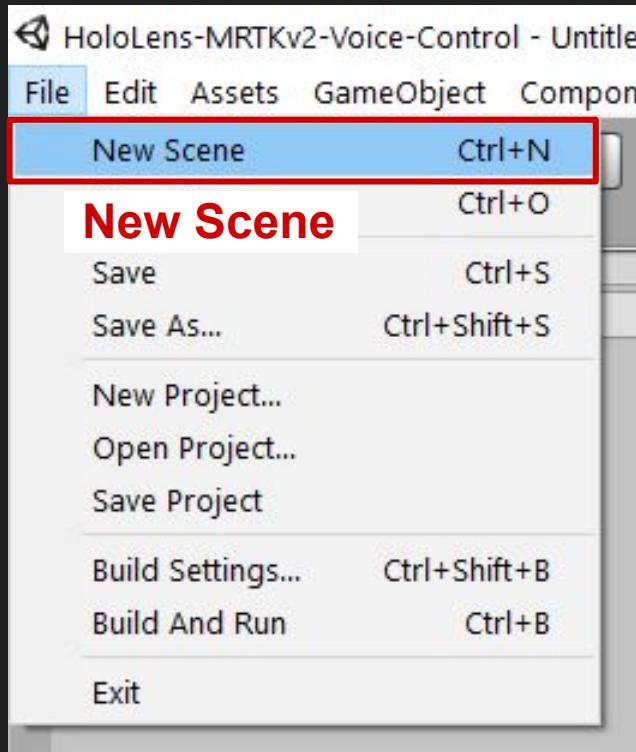
Create your first HoloLens app

- Also import Extensions & Tools UnityPackage as the same



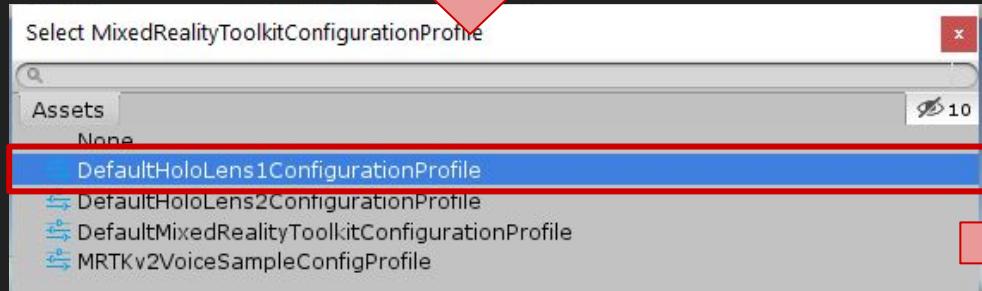
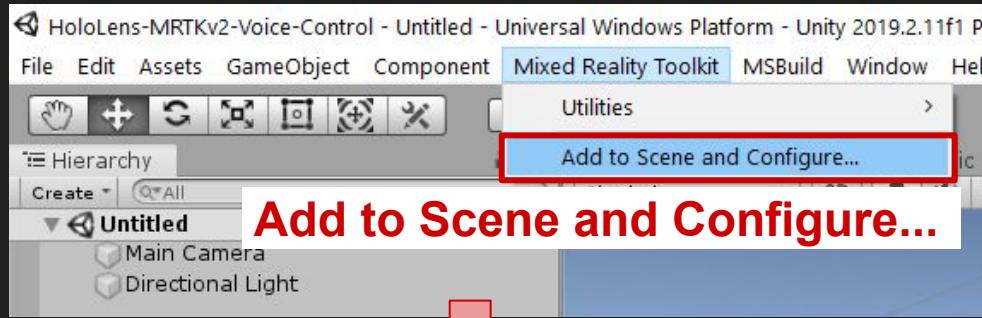
Create your first HoloLens app

- Select [File] -> [New Scene]
- Create a new Scene

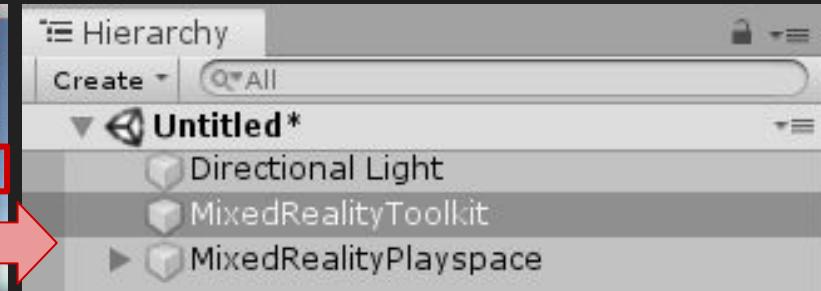


Create your first HoloLens app

- Select [Mixed Reality Toolkit] -> [Add to Scene and Configure...]
- Select **DefaultHoloLens1ConfigurationProfile**



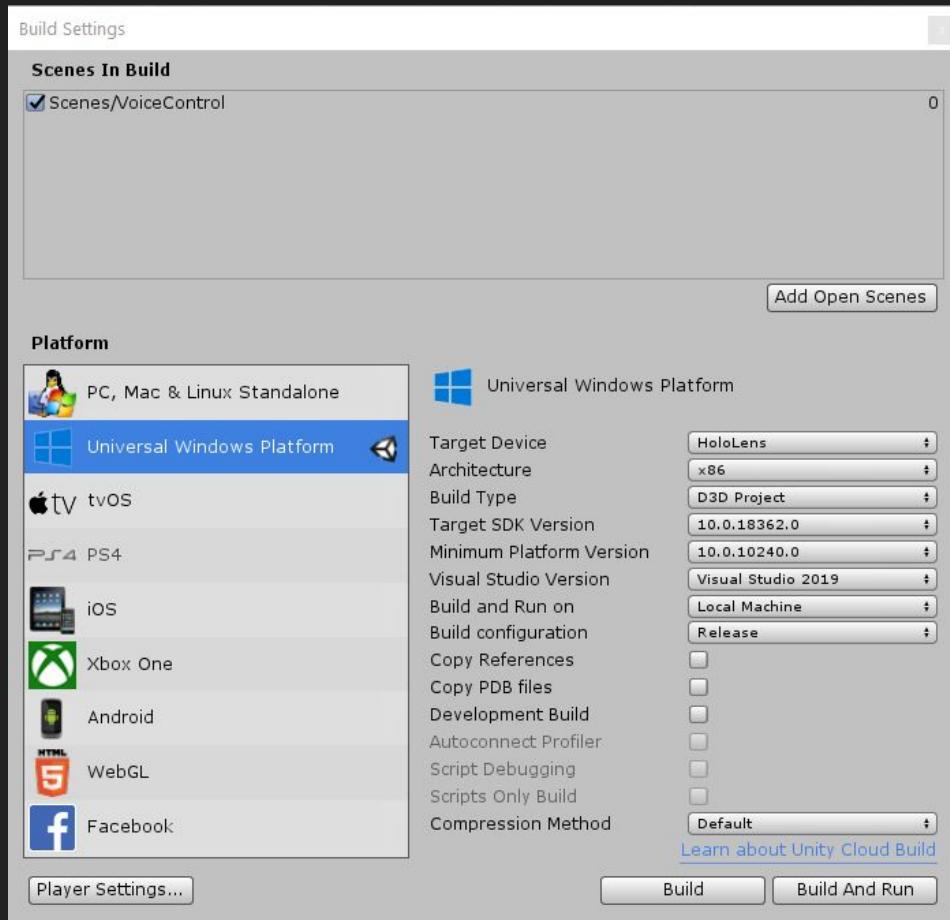
- **MixedRealityToolkit**
Providing the central configuration entry point.
- **MixedRealityPlayspace**
The parent object for the headset



DefaultHoloLens1ConfigurationProfile

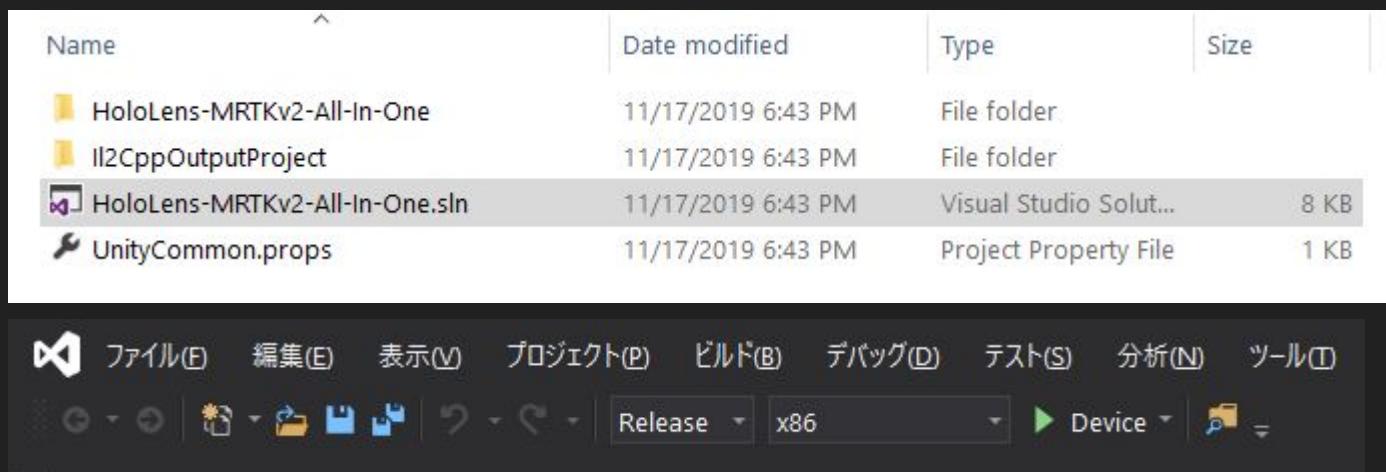
Create your first HoloLens app

- Select [File] -> [Build Settings]
- Add Open Scenes.
- Target Device: HoloLens
- Architecture: x86
- Click Build

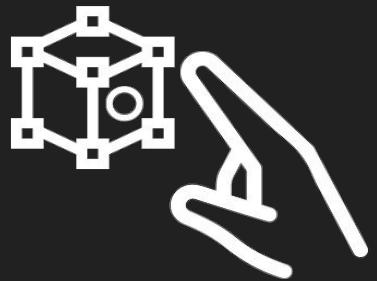


Create your first HoloLens app

- Open solution file in App folder.
- Select Release, x86, Local Computer.
- Connection HoloLens with your PC via USB.
- Build & Deploy to HoloLens



Demo

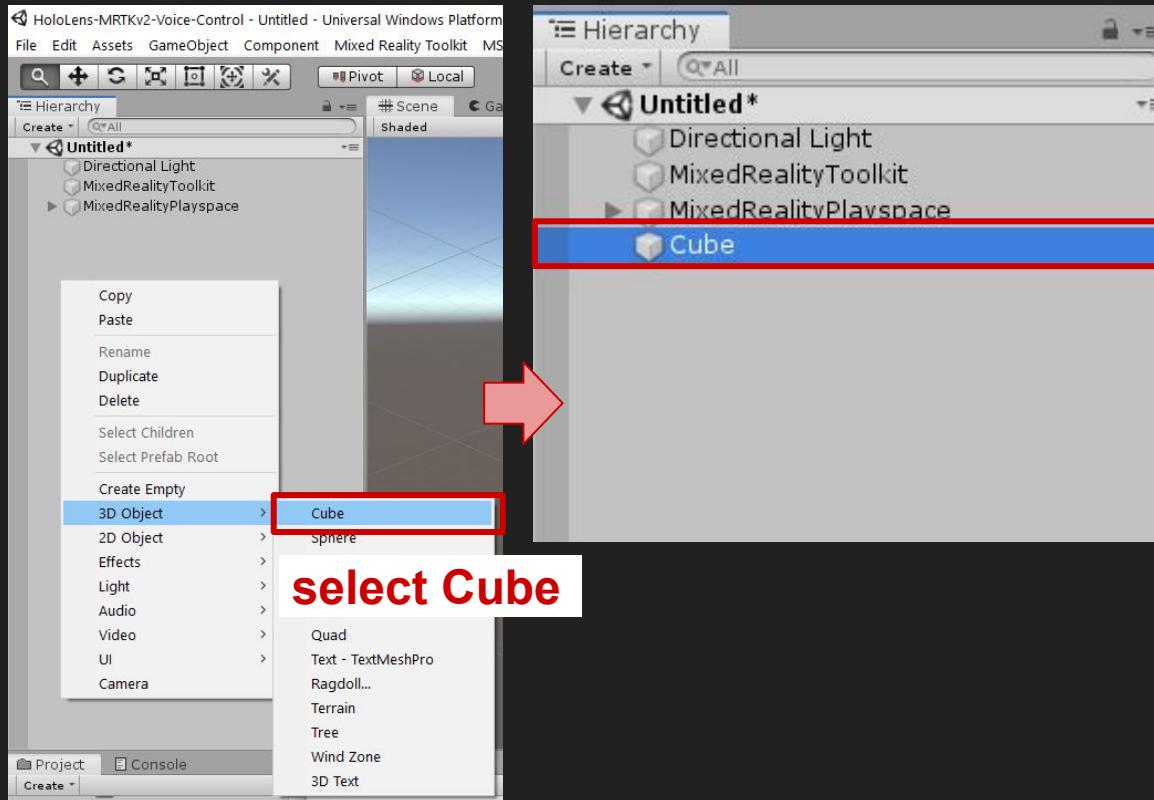


Interaction



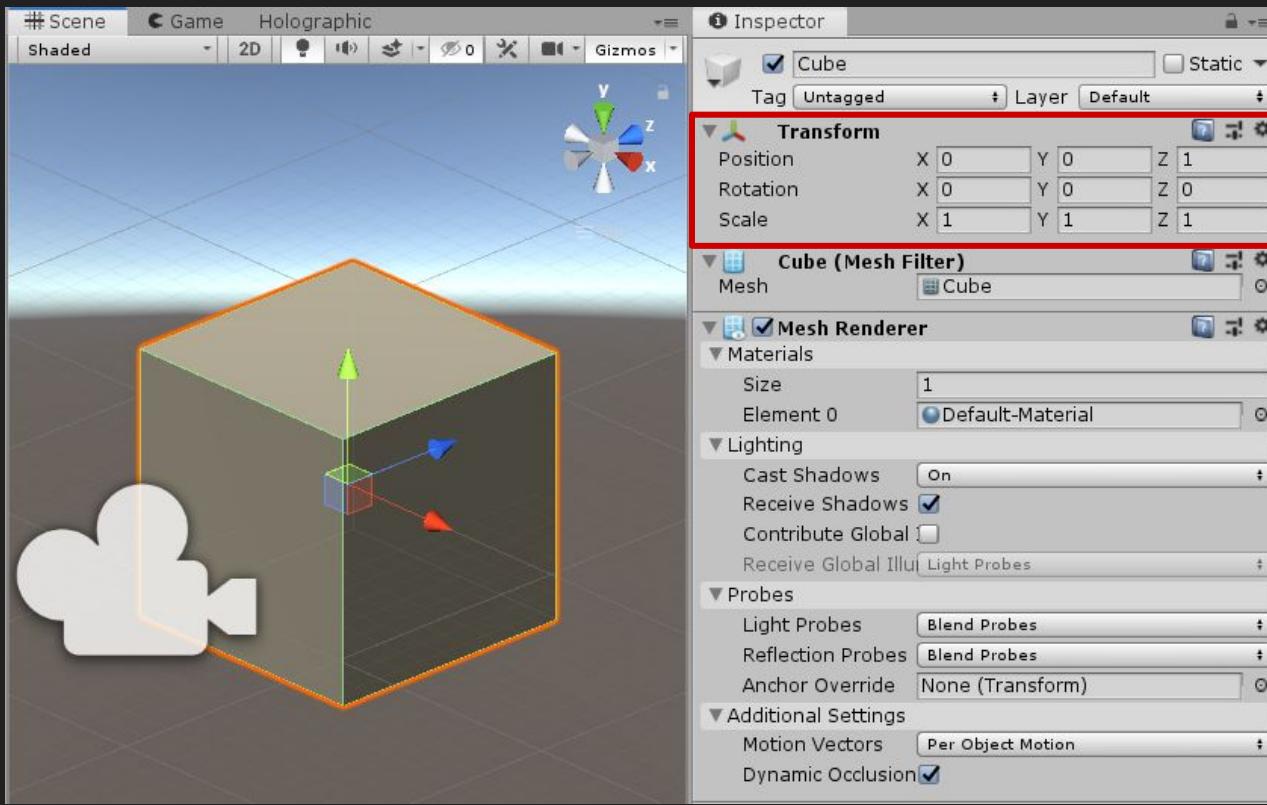
Interaction

- Right click on Hierarchy and create a new cube in your scene



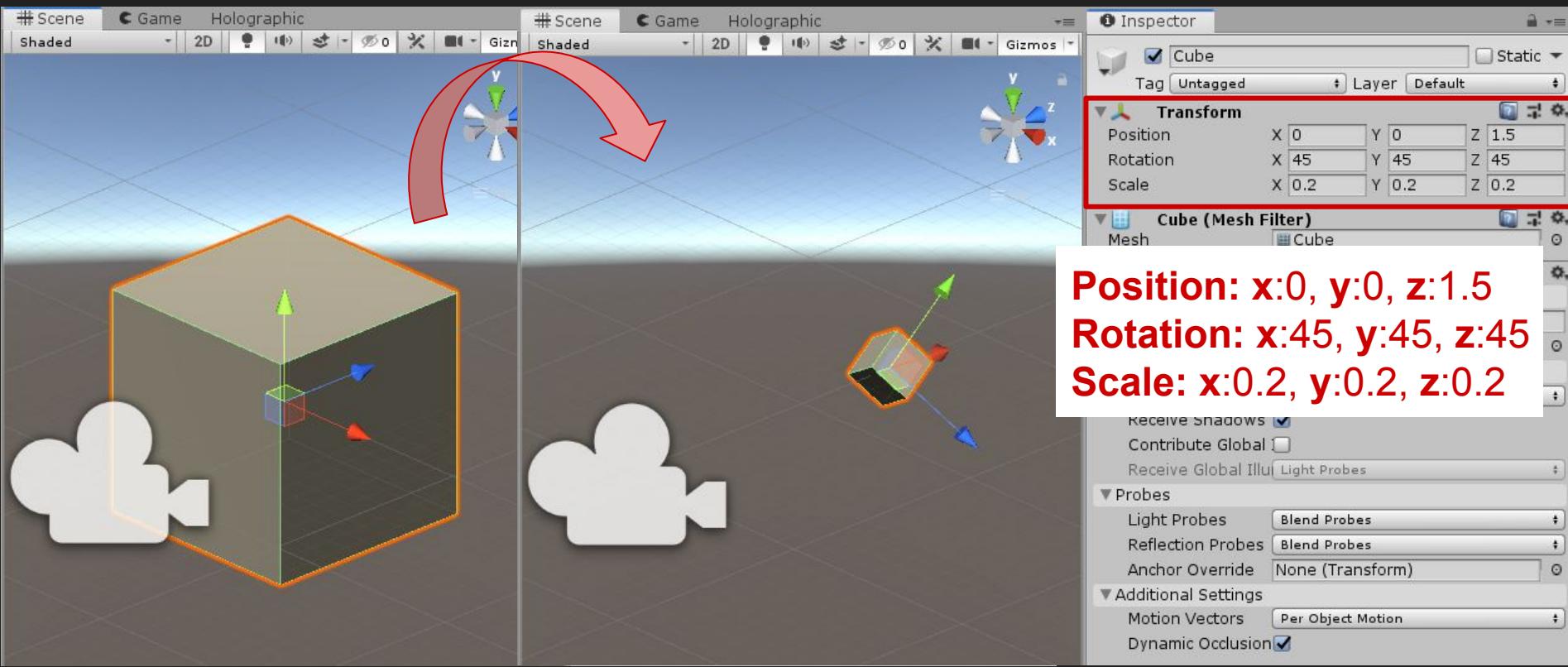
Interaction

- Select Cube in hierarchy.



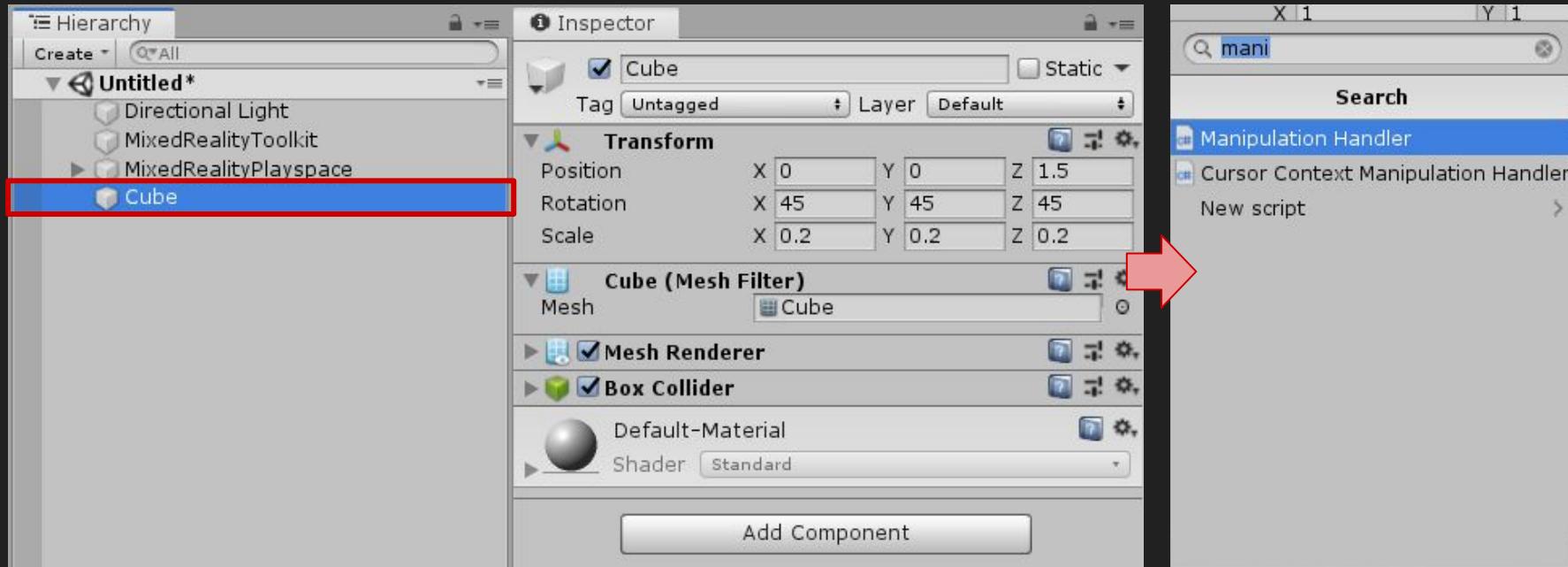
Interaction

- Select Cube in hierarchy and change Transform as the bottom.



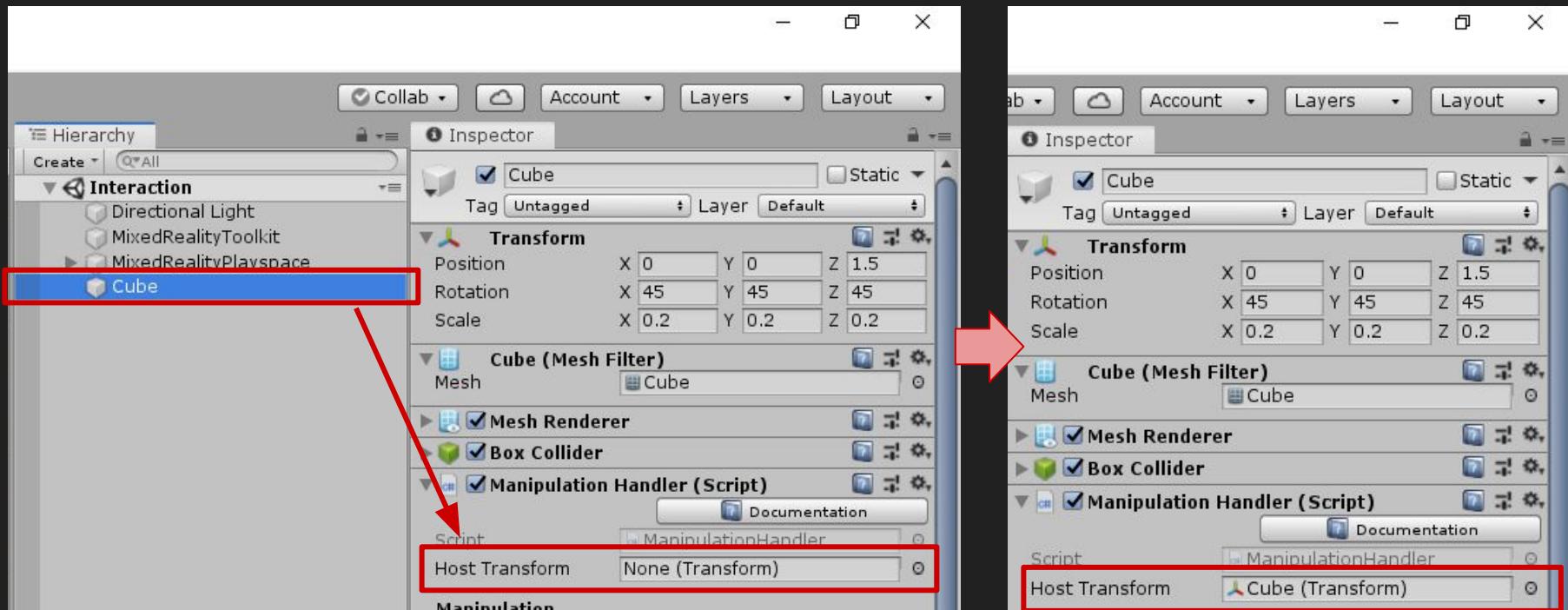
Interaction

- Select a Cube object
- click “Add Component”, and add **Manipulation Handler** in Inspector.



Interaction

- Drag & Drop Cube object into Host Transform in Manipulation Handler.

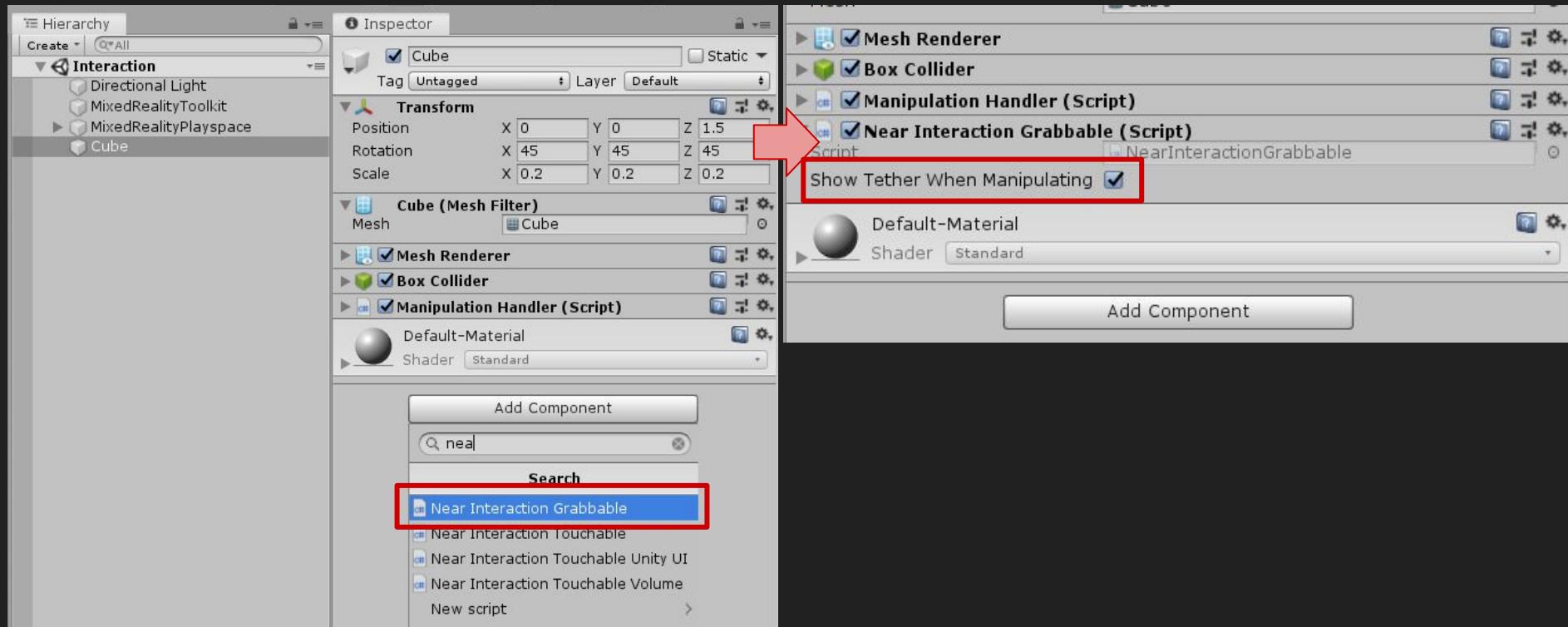


Before

After

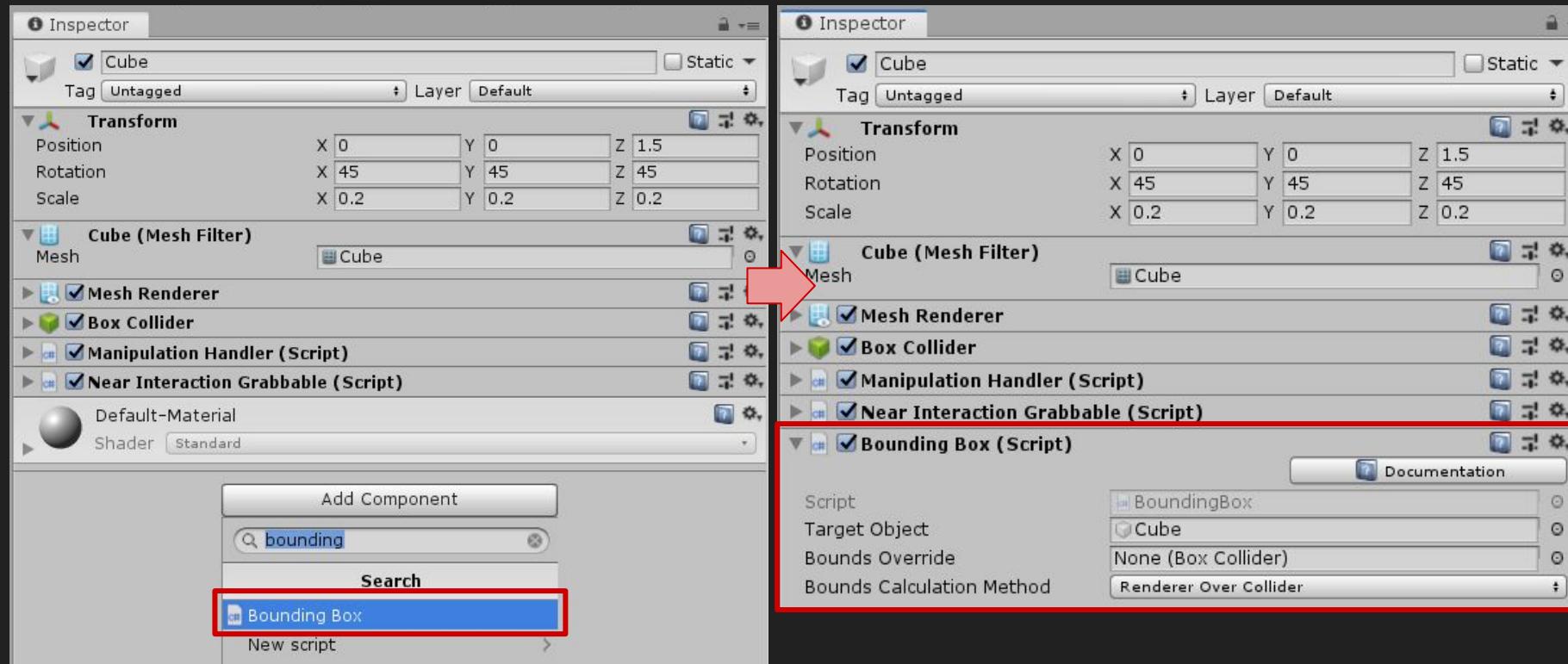
Interaction

- Click “Add Component”, and add **Near Interaction Grabbable** in Inspector.
- Check “Show Tether When Manipulating”.



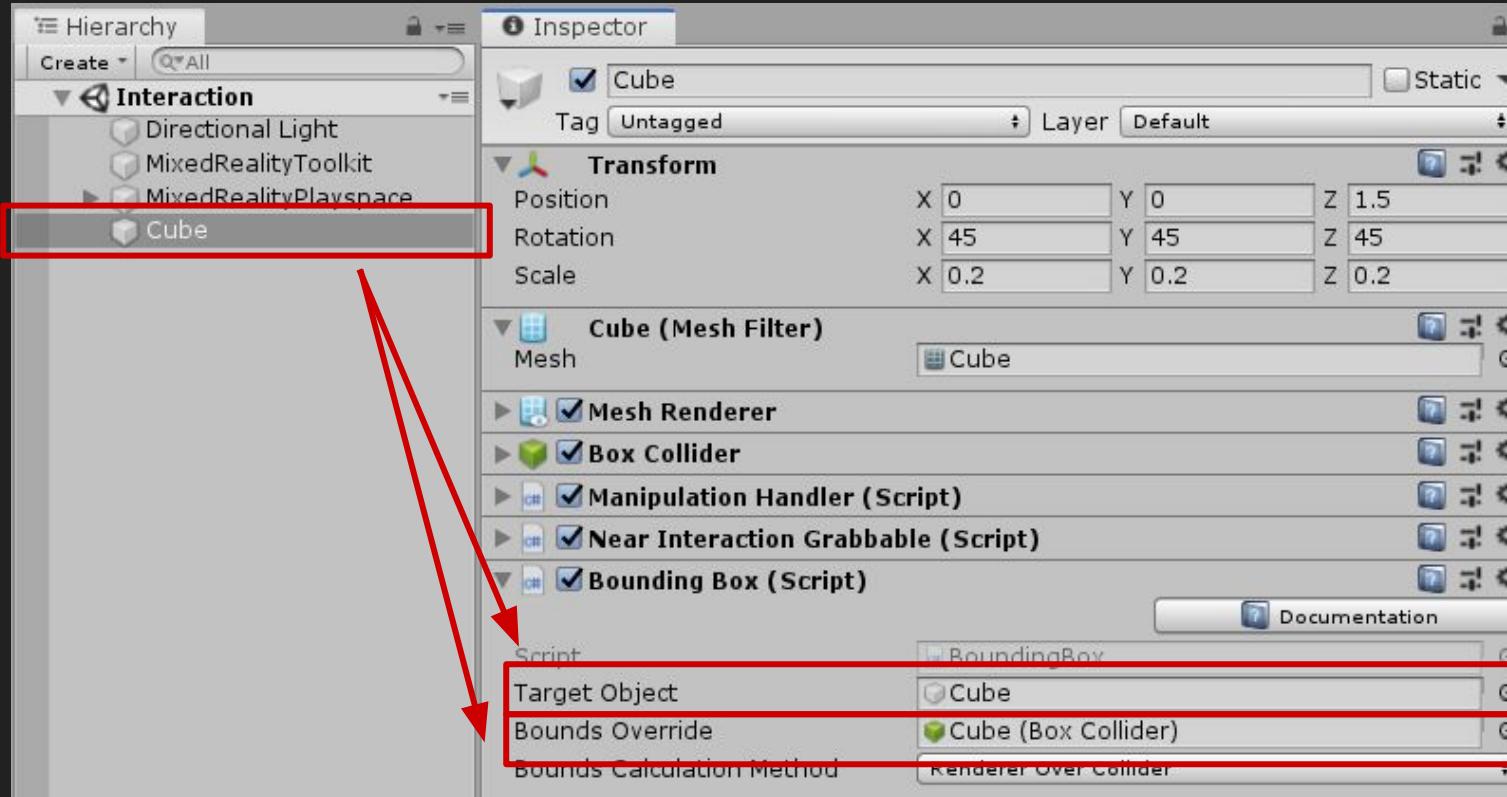
Interaction

- Click “Add Component”, and add **Bounding Box** in Inspector.



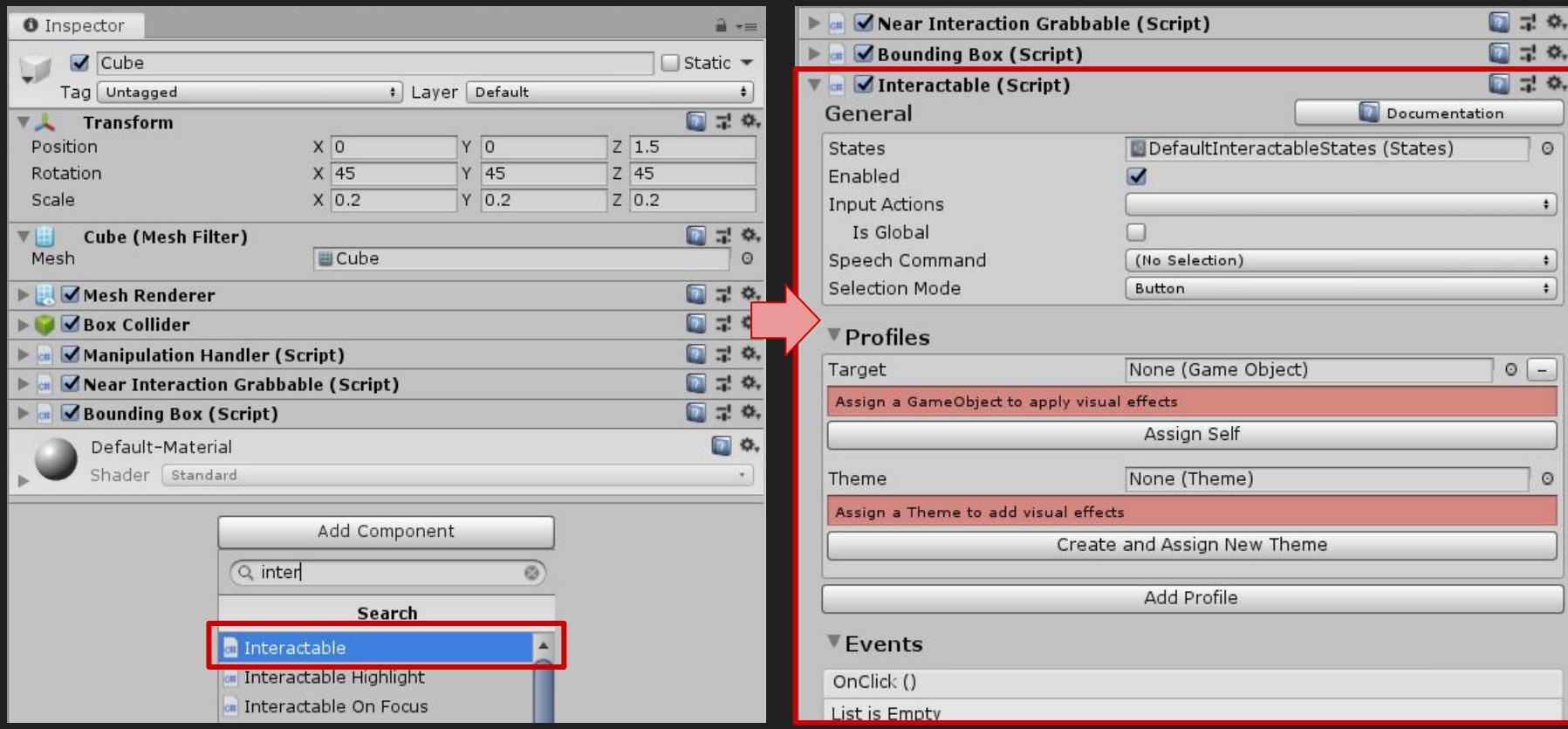
Interaction

- Drag & Drop Cube object into **Target Object & Bounds Override** in Bounding Box.



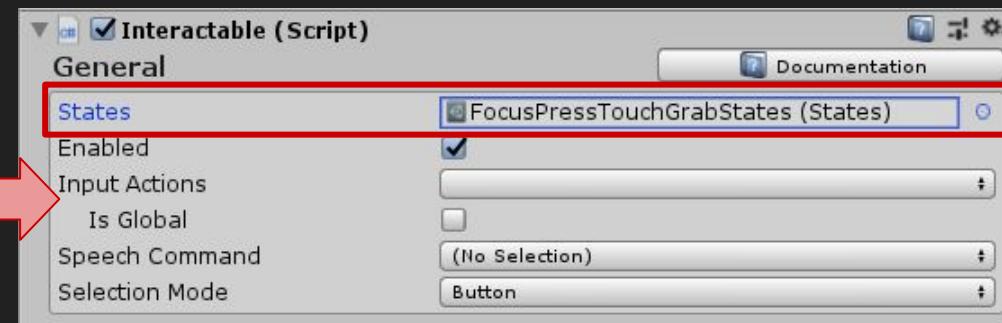
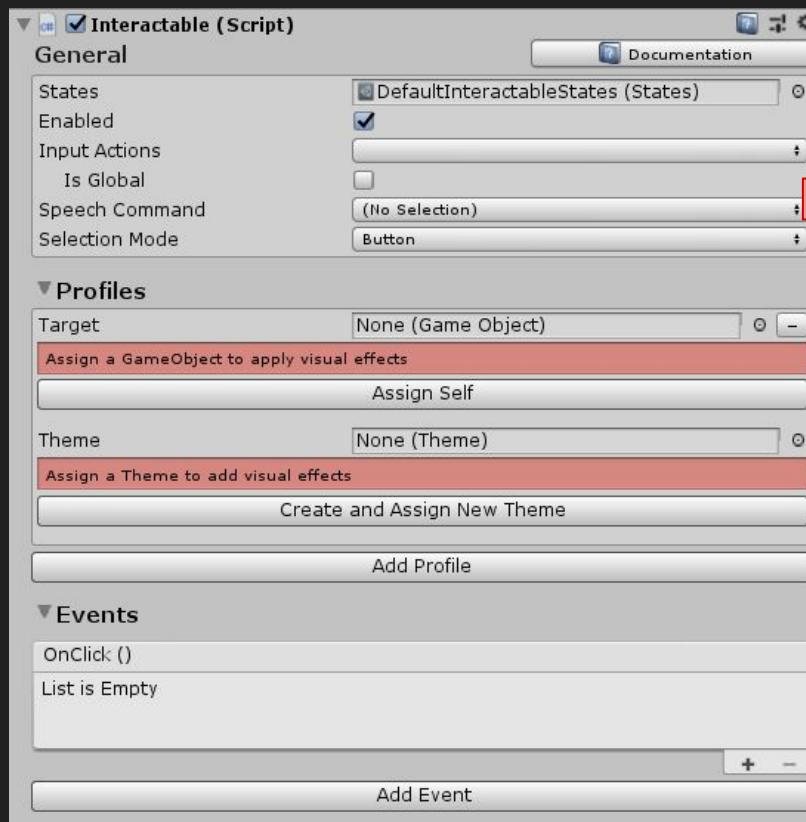
Interaction

- Click “Add Component”, and add **Interactable** in Inspector.



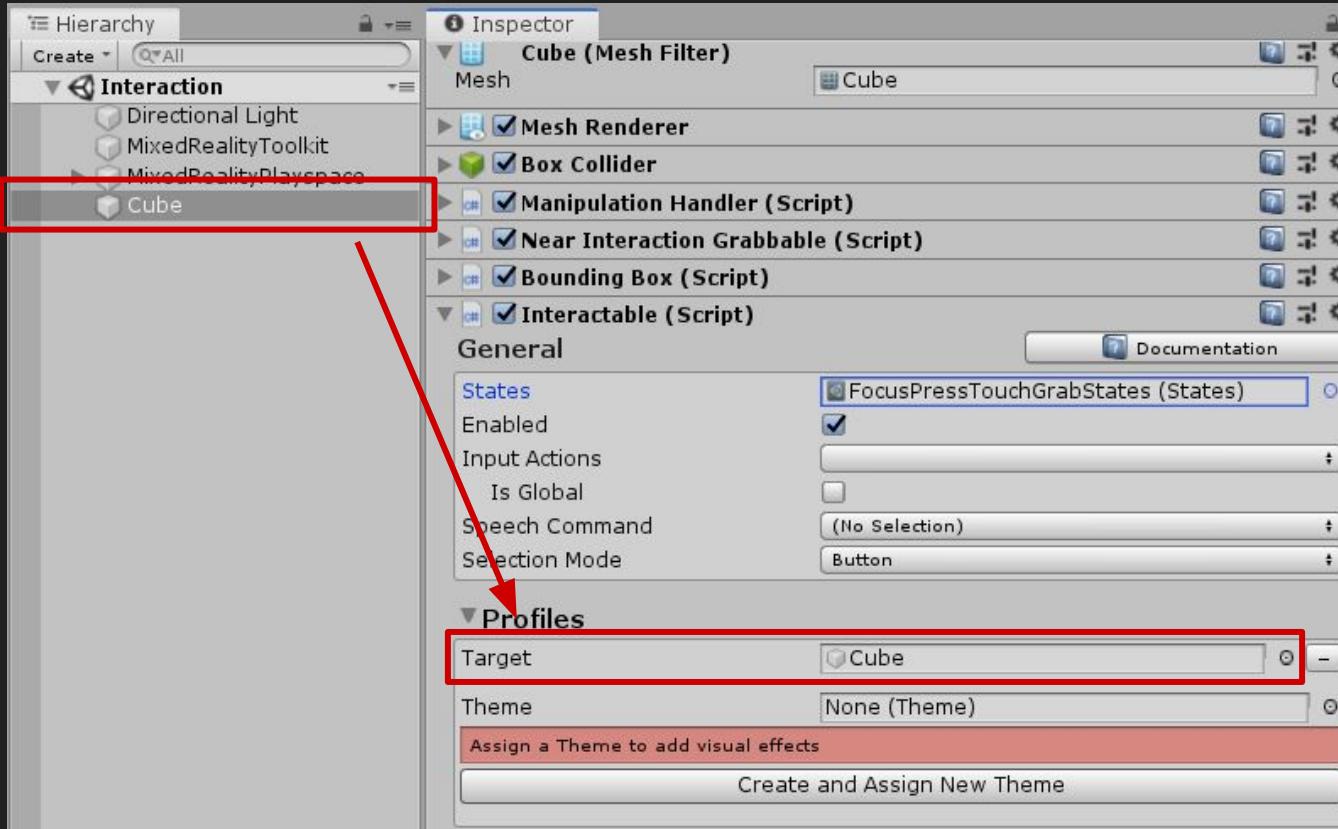
Interaction

- Set FocusPressTouchGrabStates (States) in Interactable.



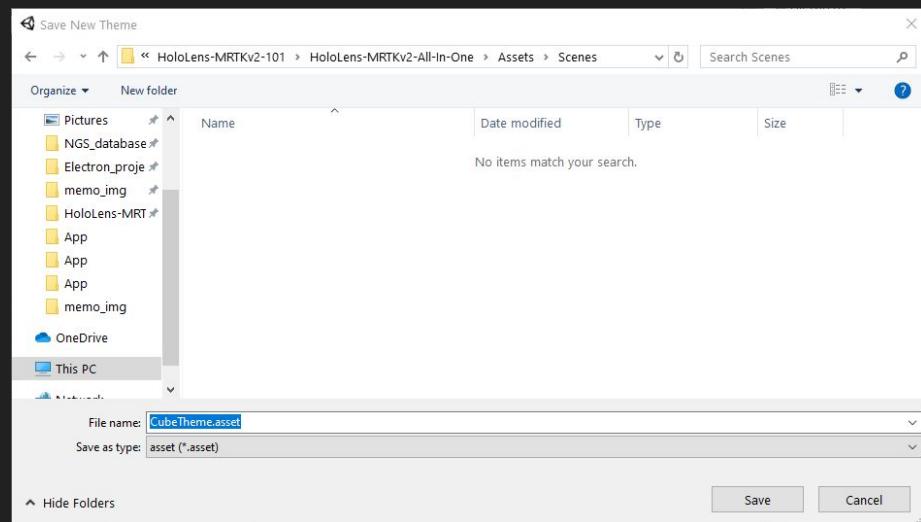
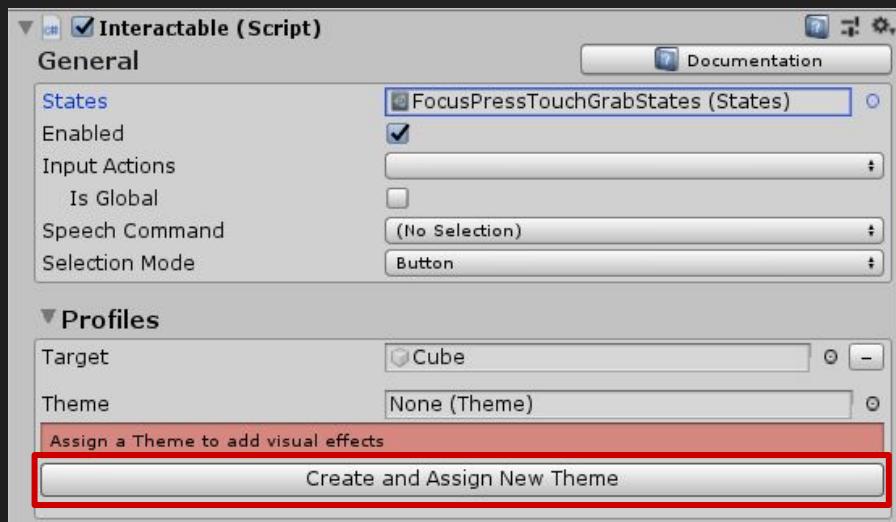
Interaction

- Drag & Drop Cube object into **Target** in Profiles in Interactable.



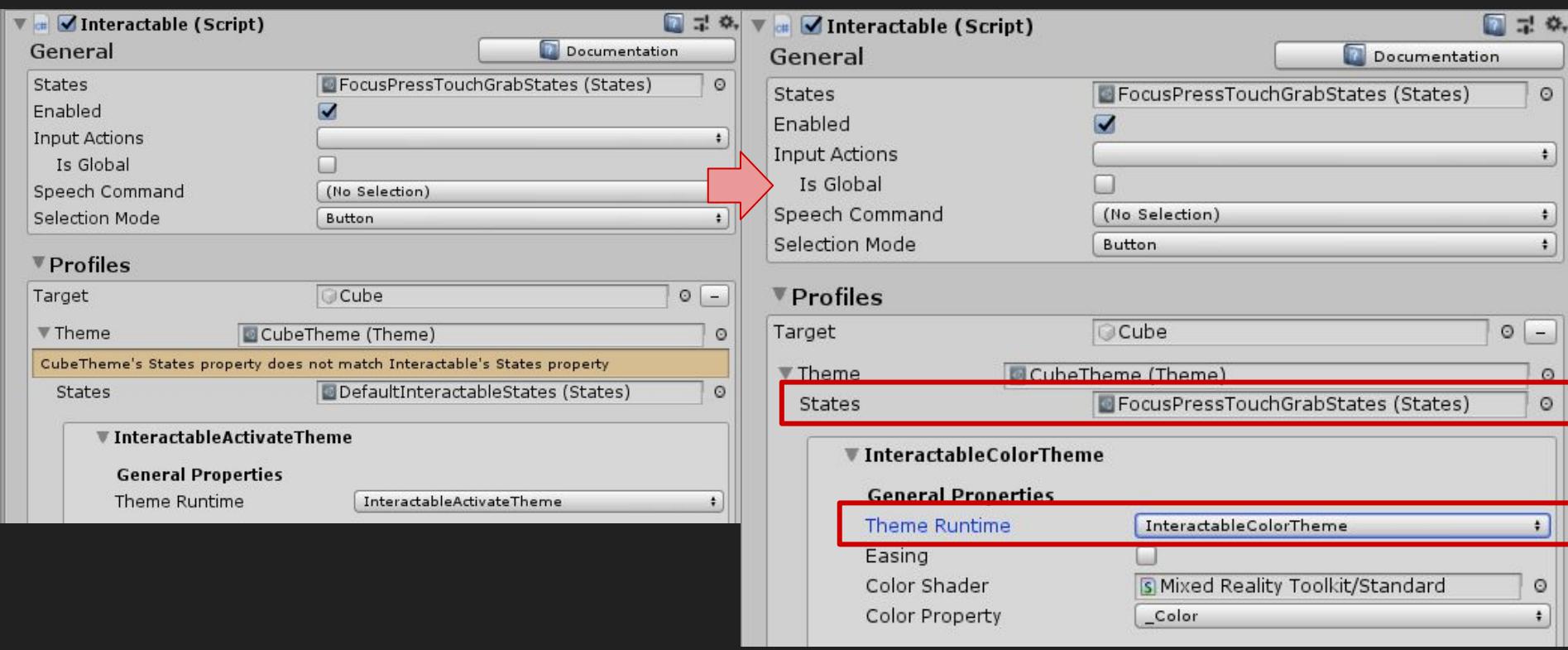
Interaction

- Create and Assign New Theme as named “CubeTheme.asset”



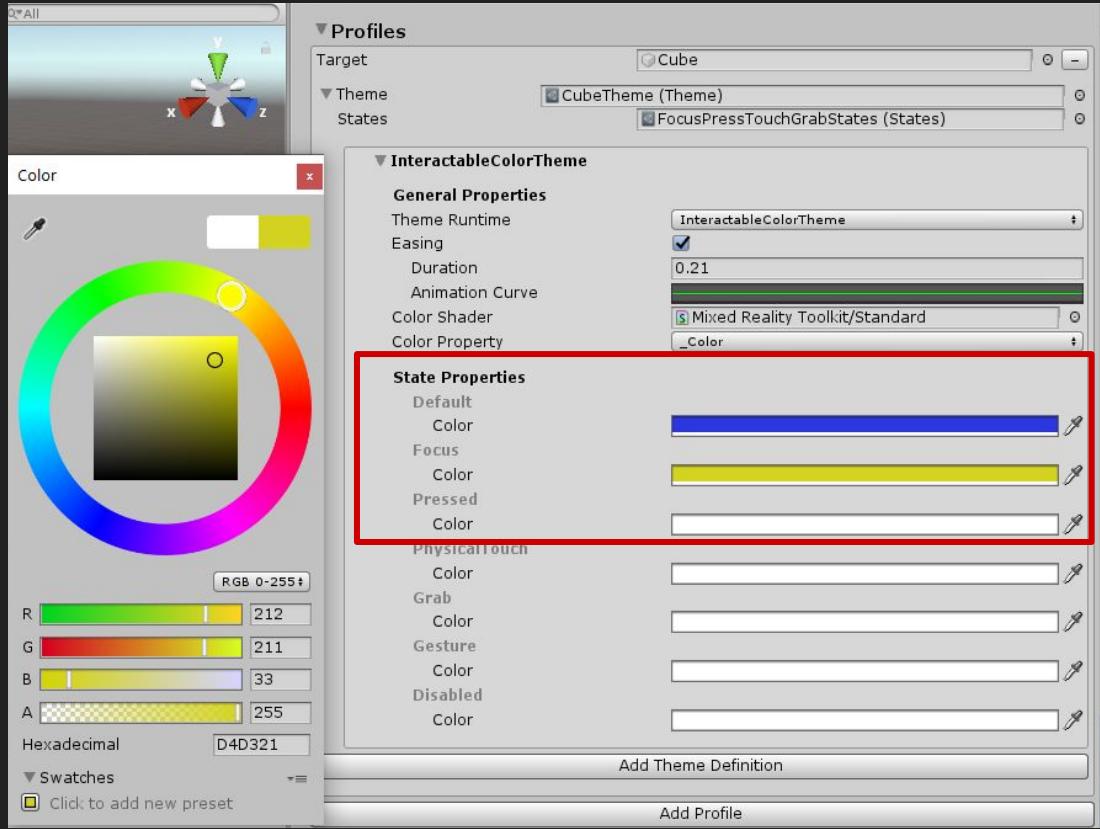
Interaction

- Set **FocusPressTouchGrabStates** in States & **InteractableColoeTheme** in Theme Runtime.



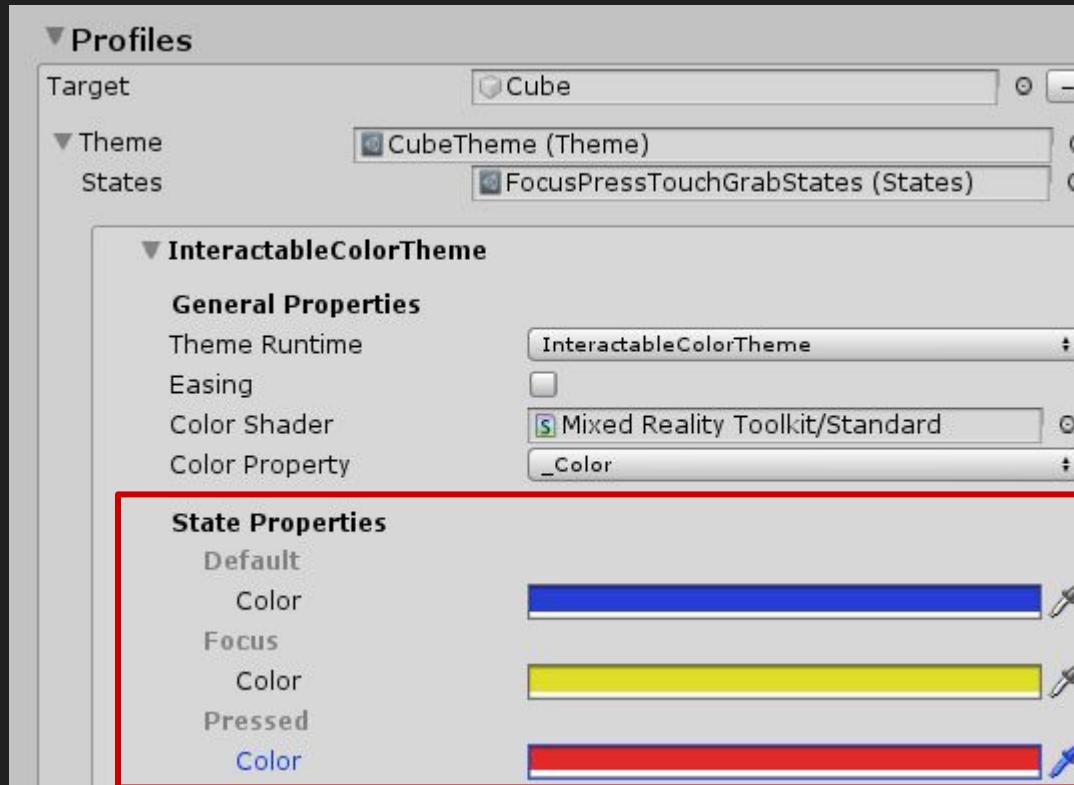
Interaction

- Change the color in State Properties using Color palette.



Interaction

- Change the color in State Properties using Color palette.

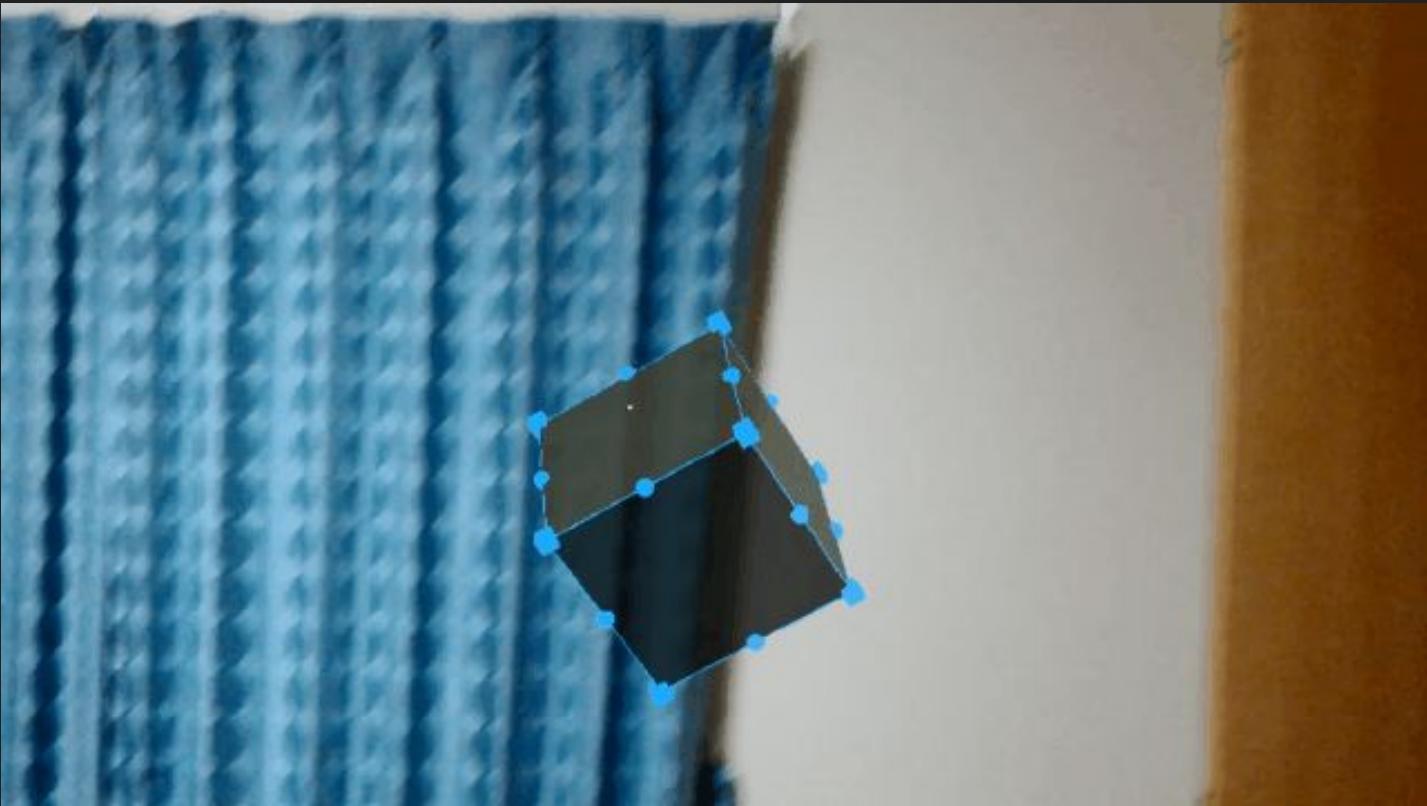


- **Default:** Blue
- **Focus:** Yellow
- **Pressed:** Red

Demo

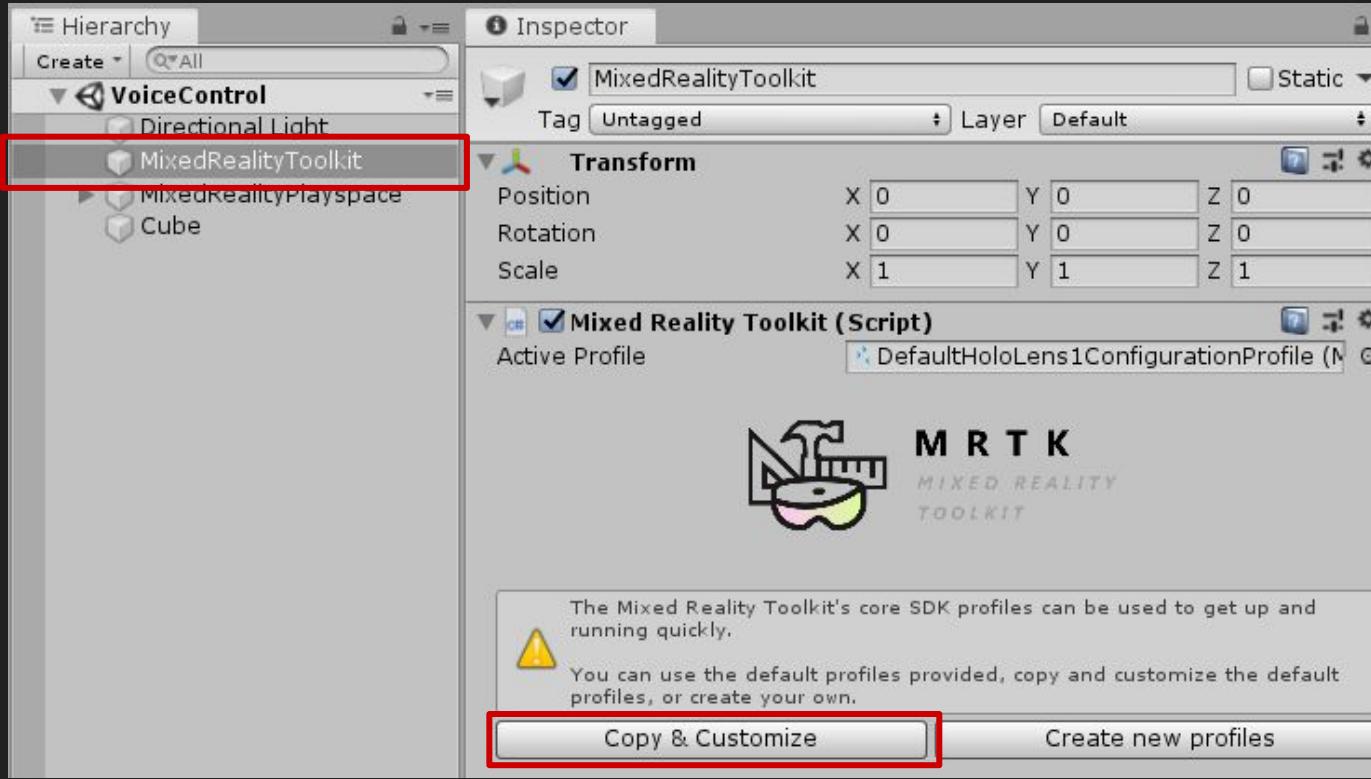


Voice
Command



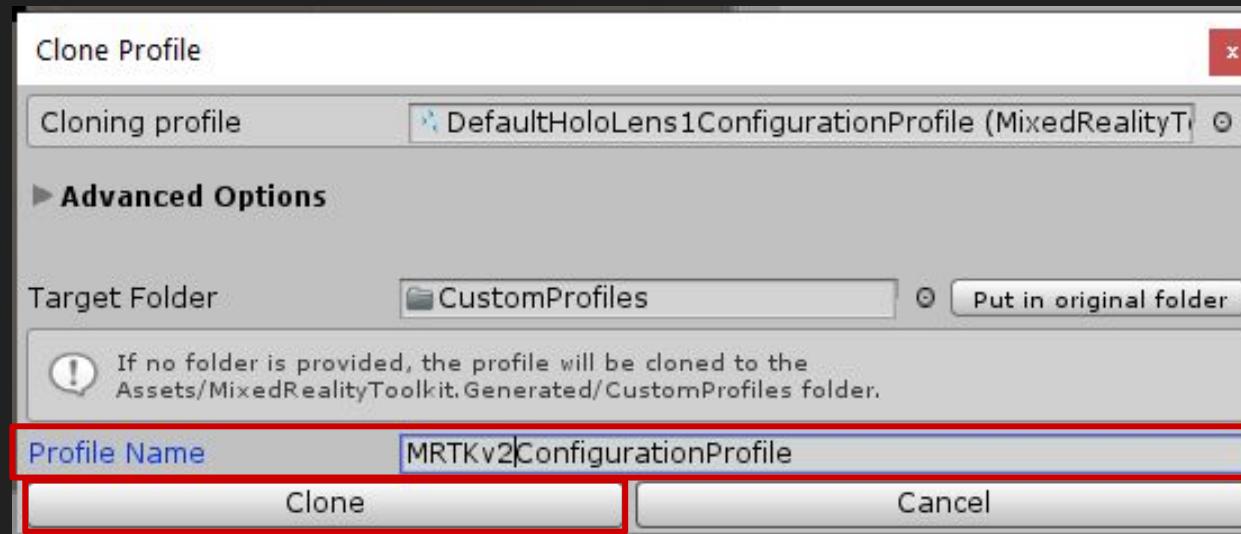
Voice Control

- Select MixedRealityToolkit in hierarchy
- Click “Copy & Customize” in Mixed Reality Toolkit.



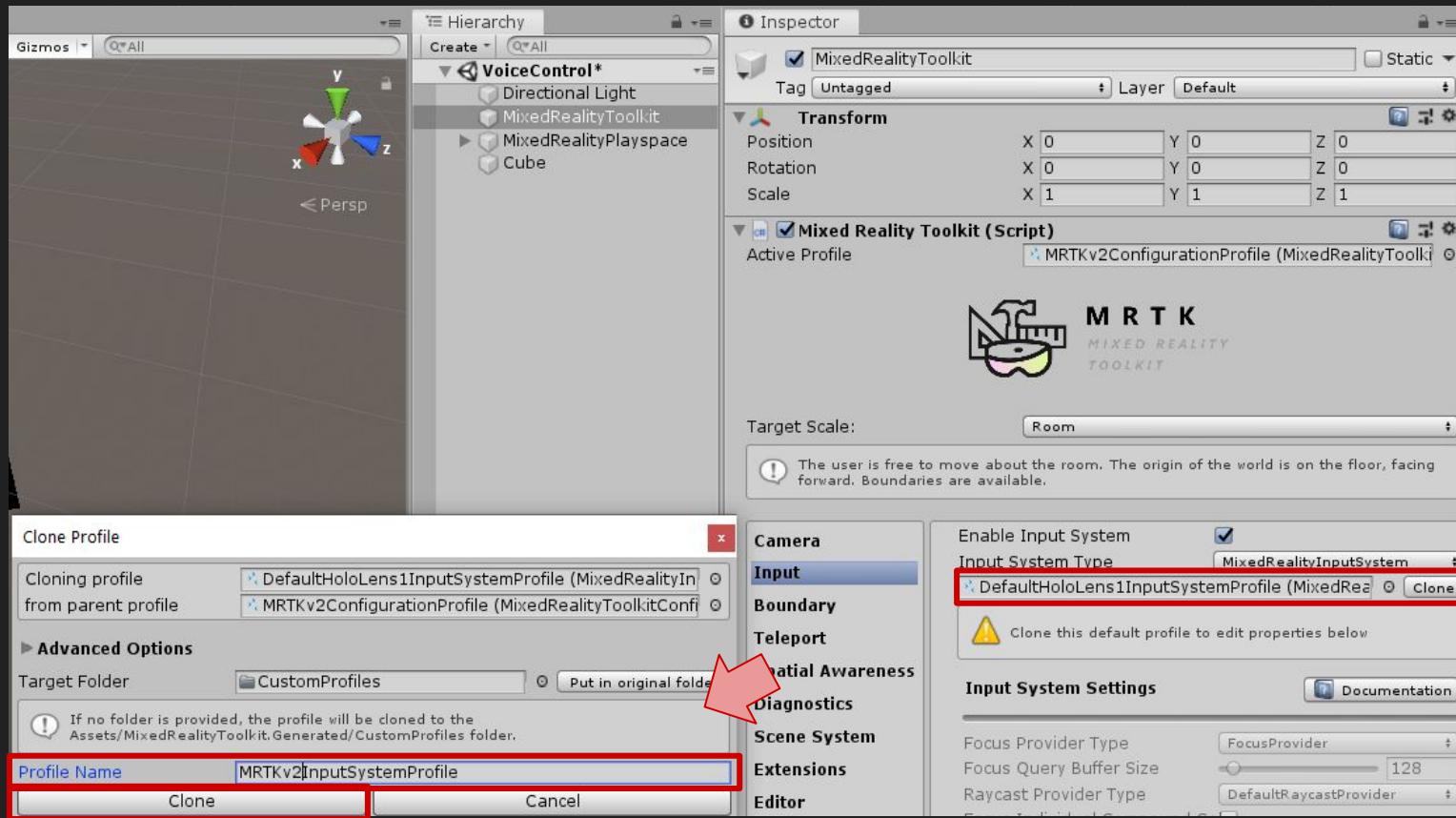
Voice Control

- Clone a Profile as named “MRTKv2ConfigurationProfile”.



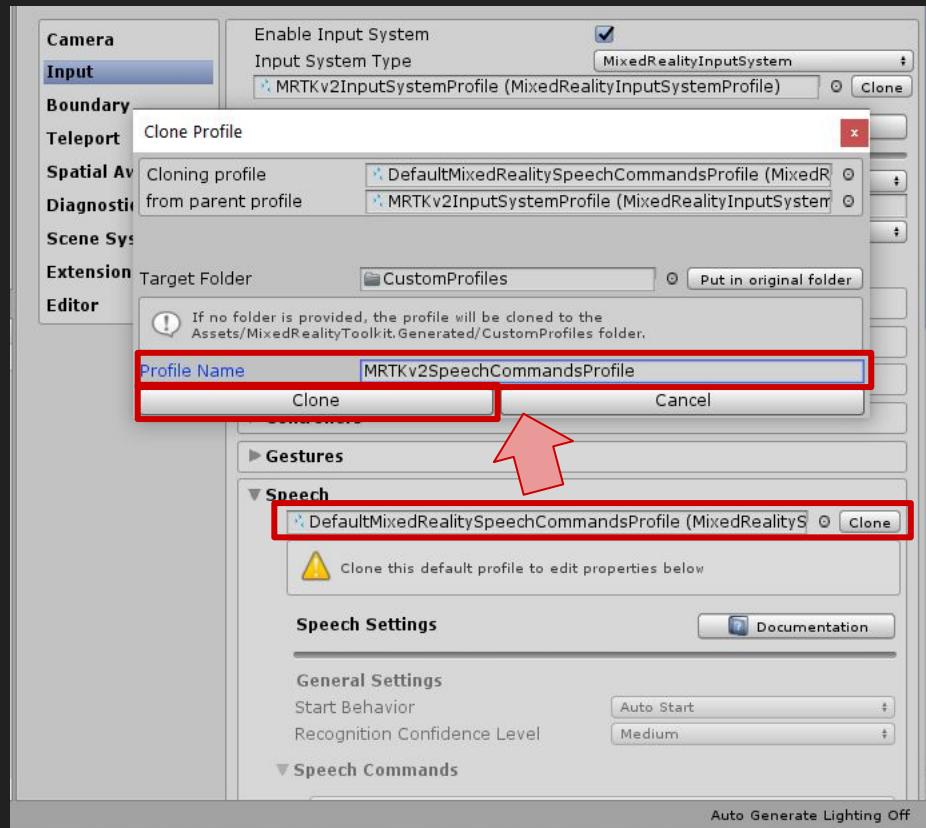
Voice Control

- Clone a Input profile as named “MRTKv2InputSystemProfile”.



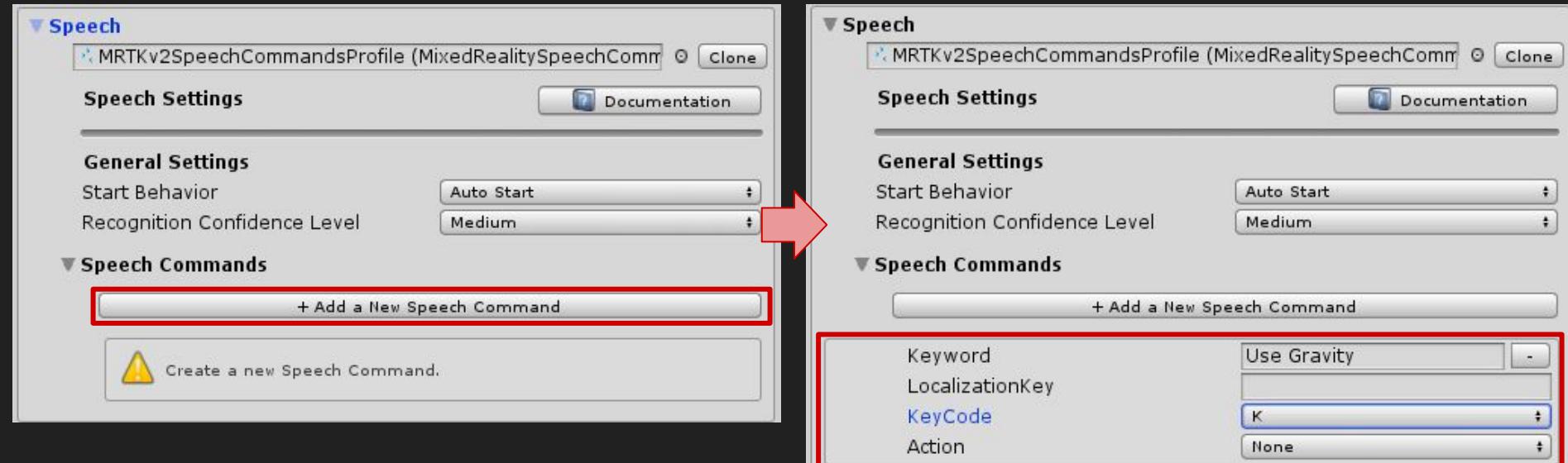
Voice Control

- Clone a Speech command profile as named “MRTKv2SpeechCommandsProfile”.



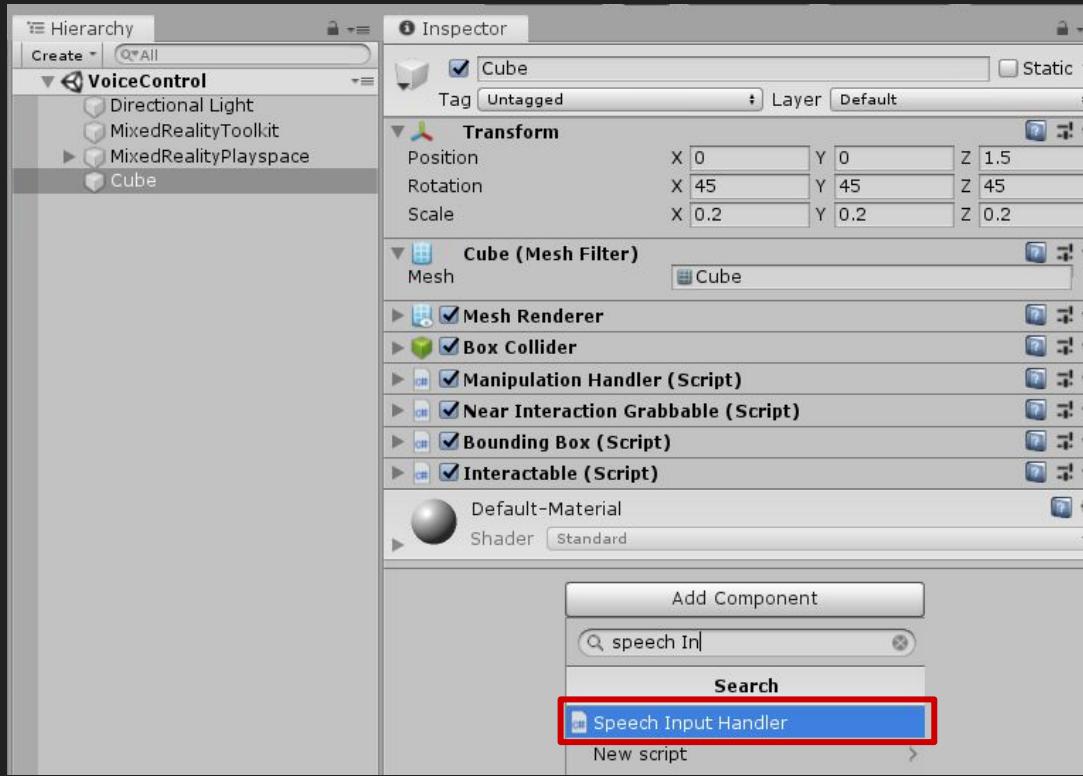
Voice Control

- Add a new Speech Command as the following.



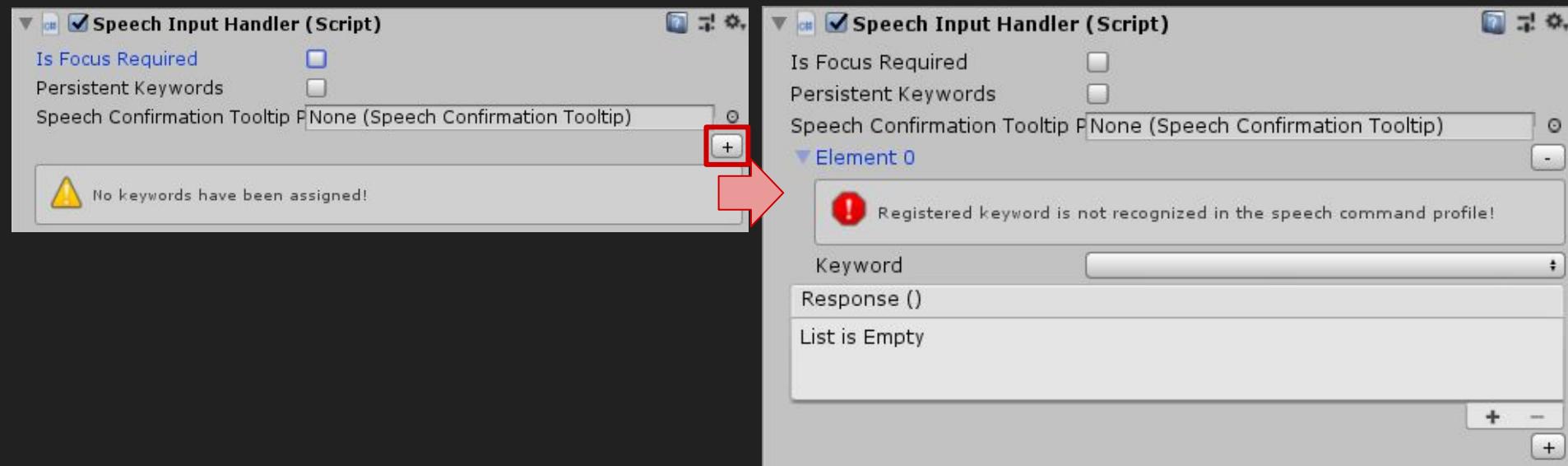
Voice Control

- Select Cube object in Hierarchy.
- Click “Add Component”, and add **Speech Input Handler** in Inspector.



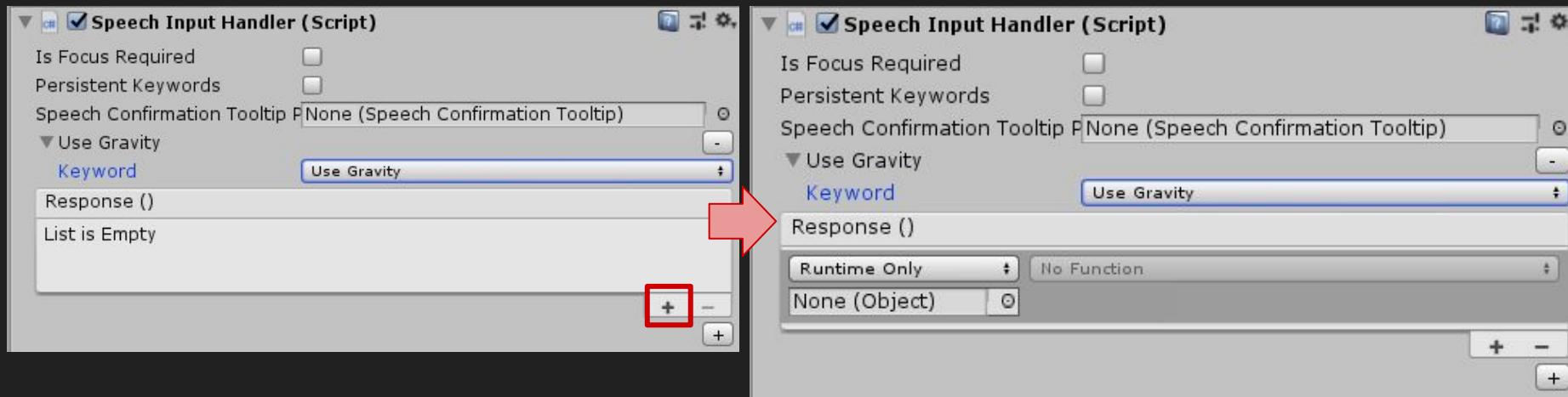
Voice Control

- Checkout “Is Focus Required” in Speech Input Handler
- Click “+”.



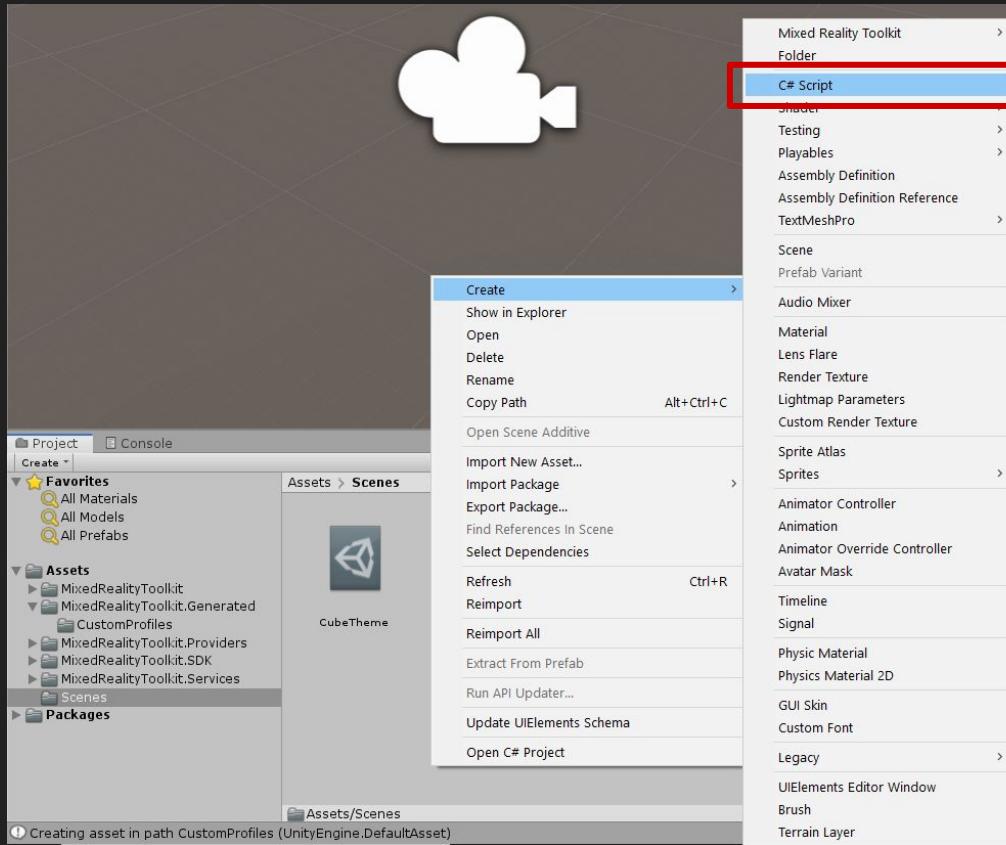
Voice Control

- Select “Use Gravity” in keyword.
- Add a new Response action.



Voice Control

- Right Click on Project & select [Create] -> [C# script].



Voice Control

- Open the C# script and edit as the following.

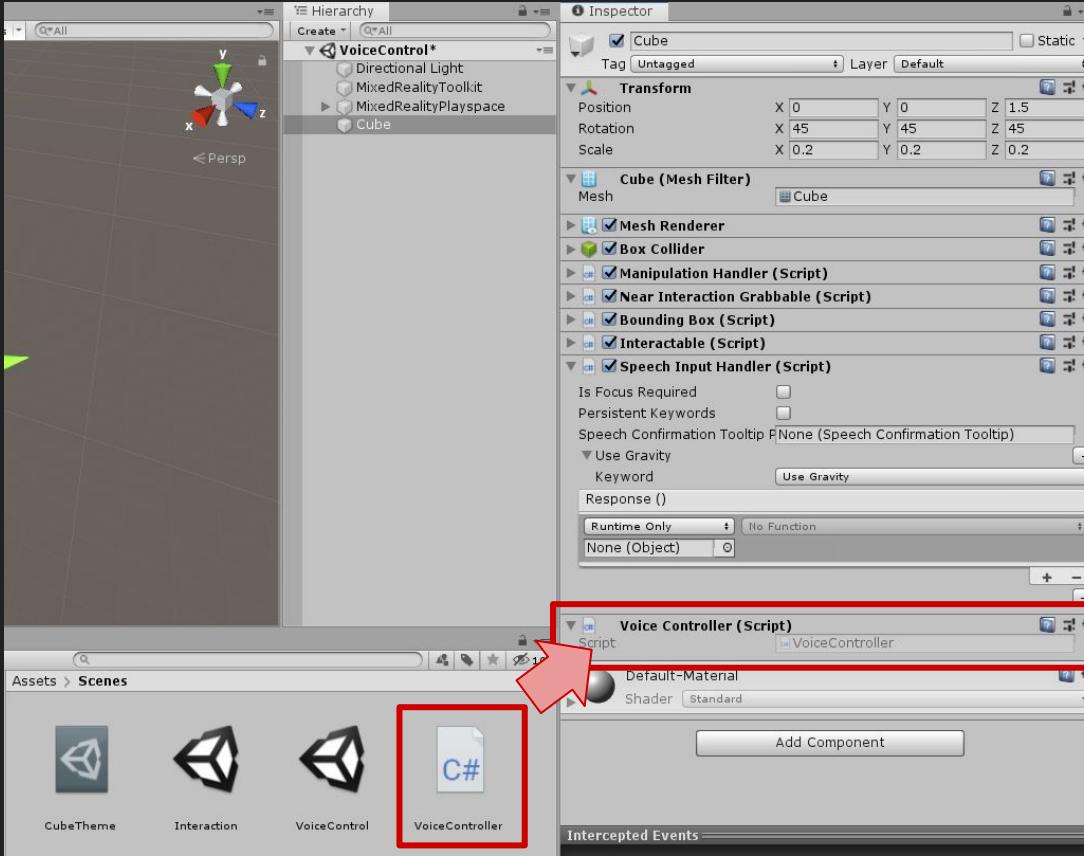
The screenshot shows the Unity Editor's code editor window with the file "VoiceController.cs" open. The window title bar says "VoiceController.cs" and "Assembly-CSharp". The code editor displays the following C# script:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class VoiceController : MonoBehaviour
6  {
7      public void triggerGravityOn()
8      {
9          Debug.Log("TriggerOn.");
10         GetComponent<Rigidbody>().useGravity = true;
11         GetComponent<Renderer>().material.color = Color.red;
12     }
13 }
14
```

The line "Debug.Log("TriggerOn.");" is highlighted with a light gray background, indicating it is selected or being edited.

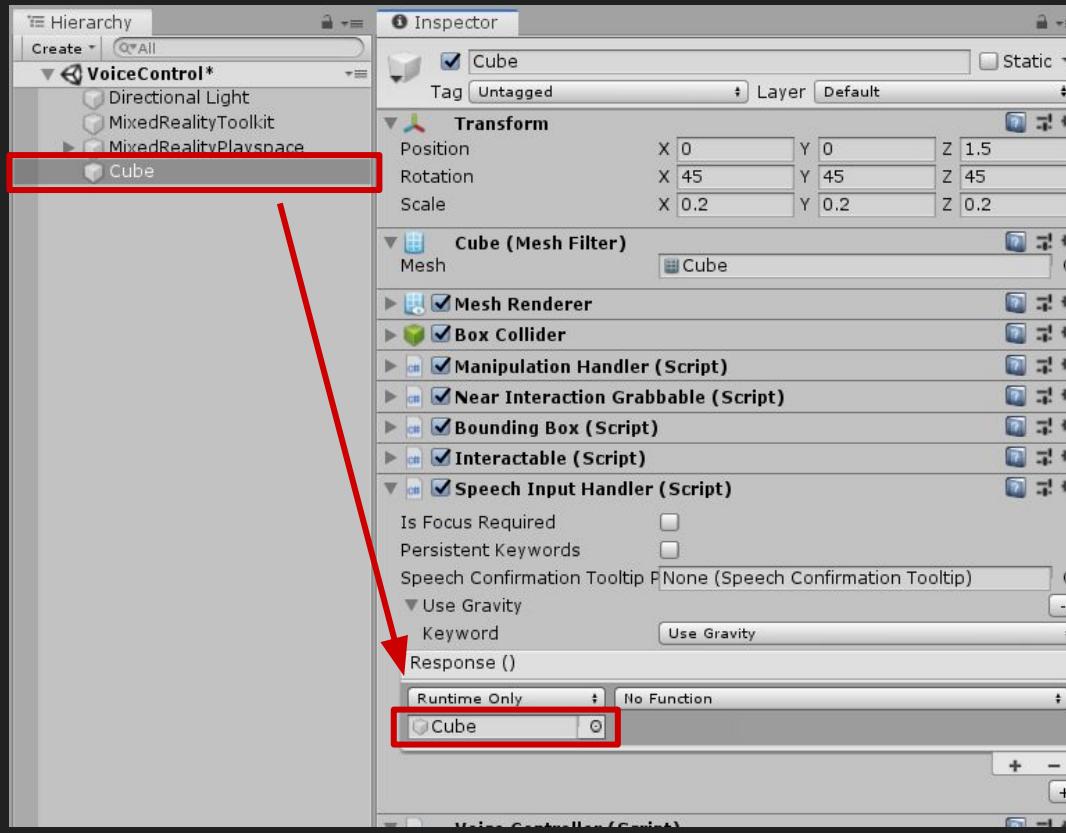
Voice Control

- Attach the C# script to Cube object.



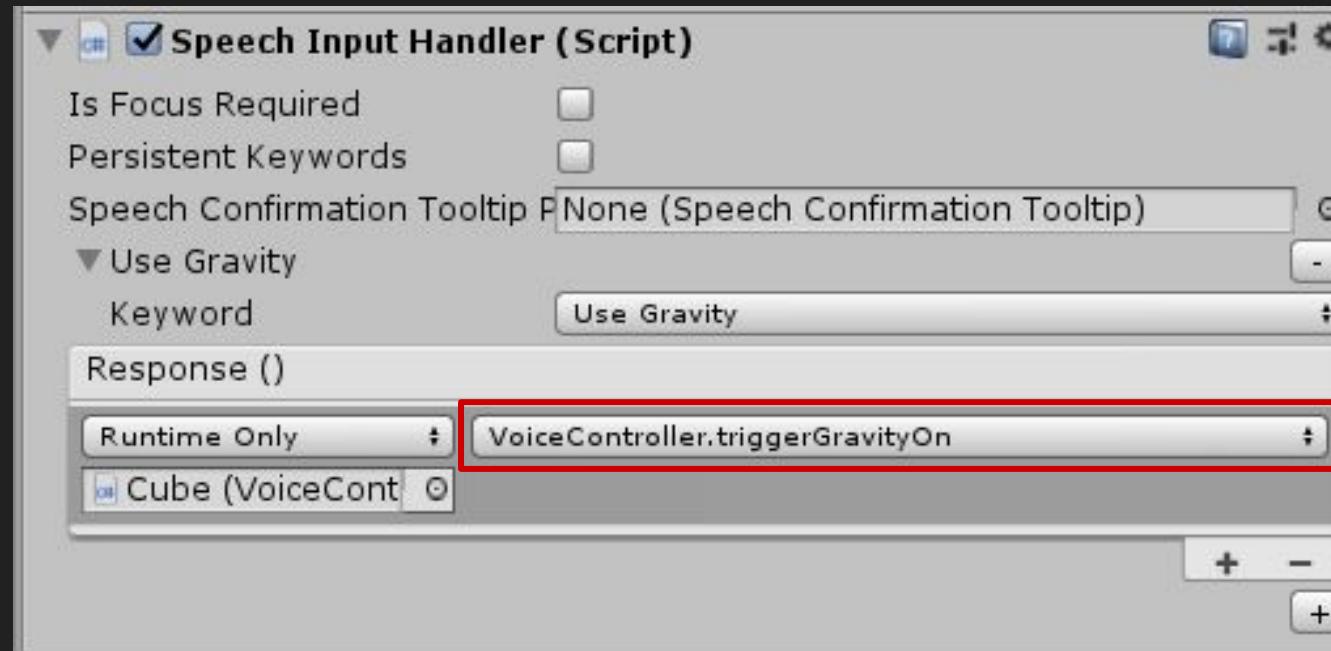
Voice Control

- Attach Cube object to target object in Speech Input Handler.



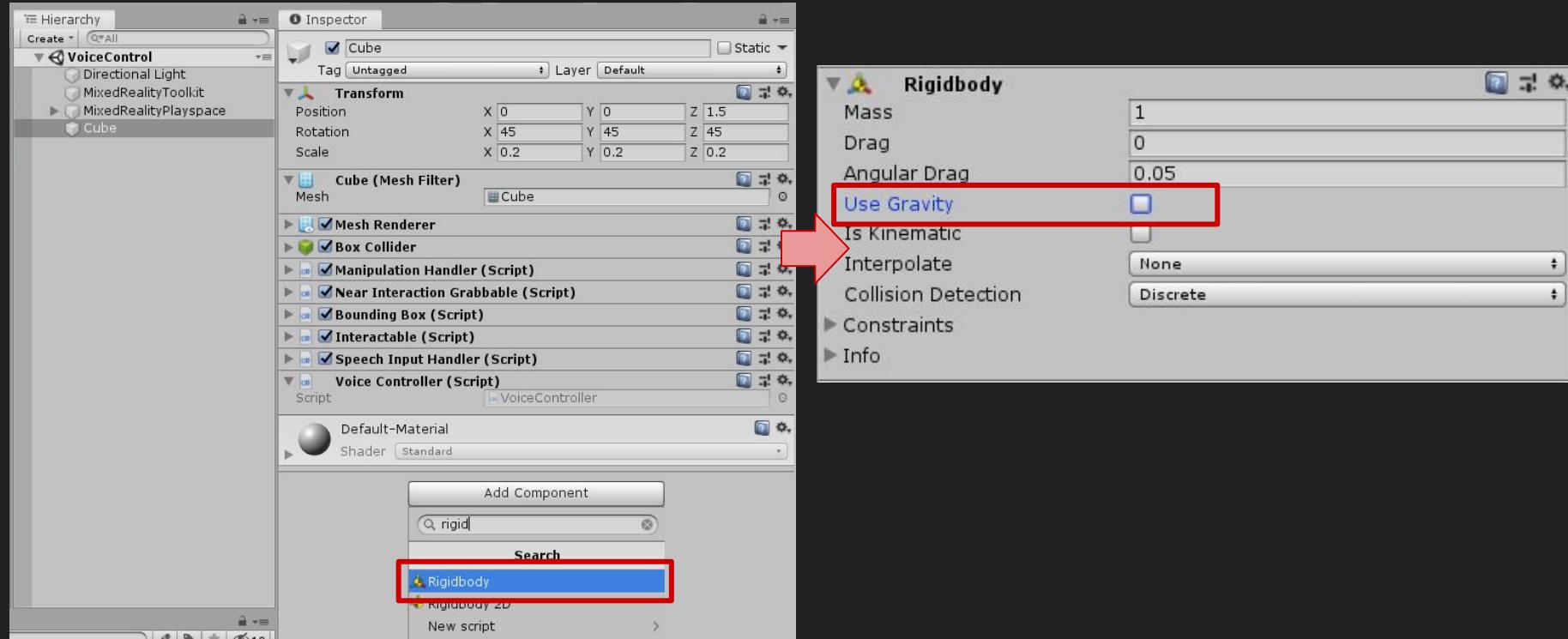
Voice Control

- Set triggerGravityOn() function as a voice response.



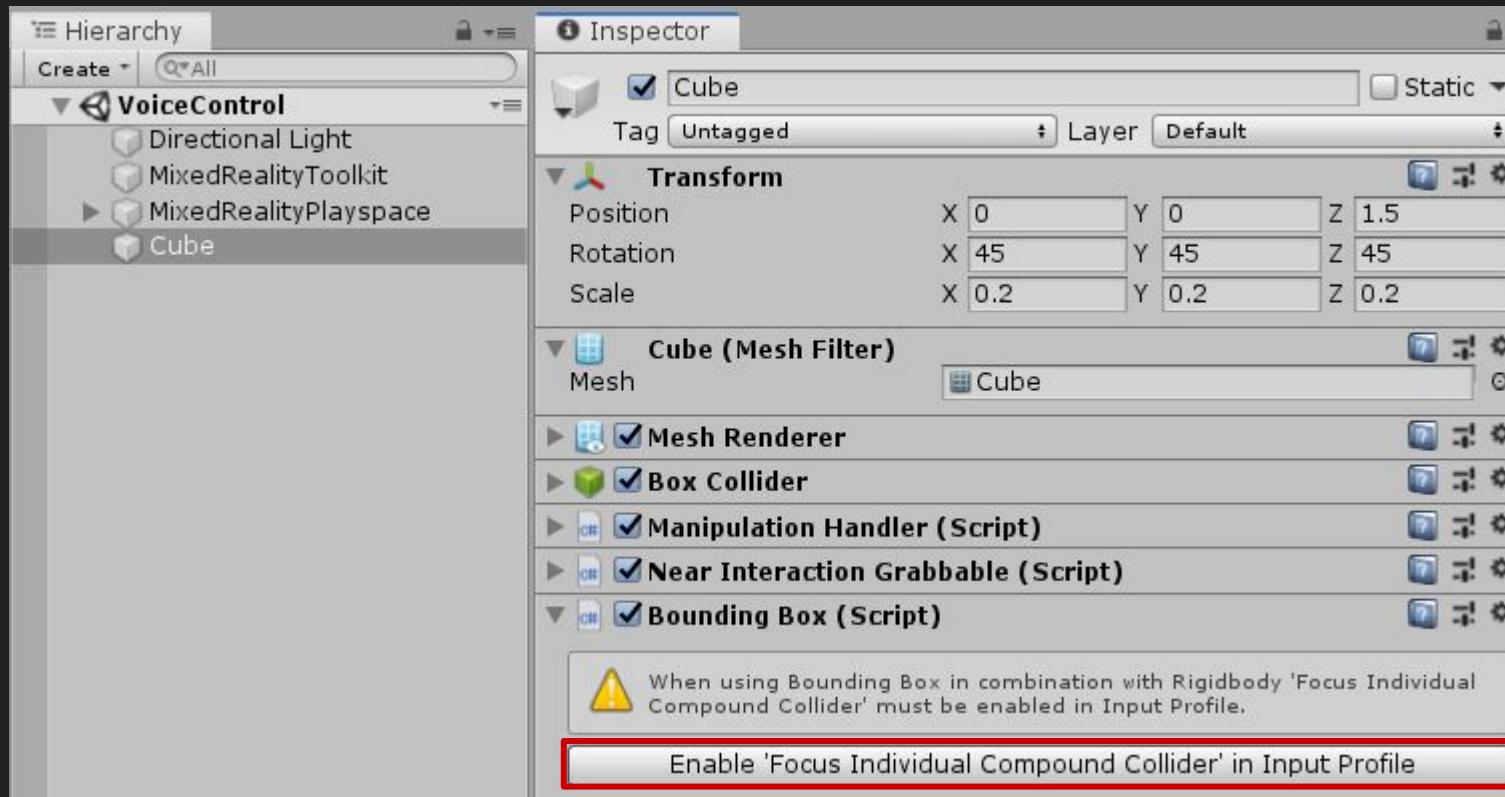
Voice Control

- Click “Add Component”, and add **Rigid Body** in Inspector.
- Checkout **Use Gravity** in Rigidbody.



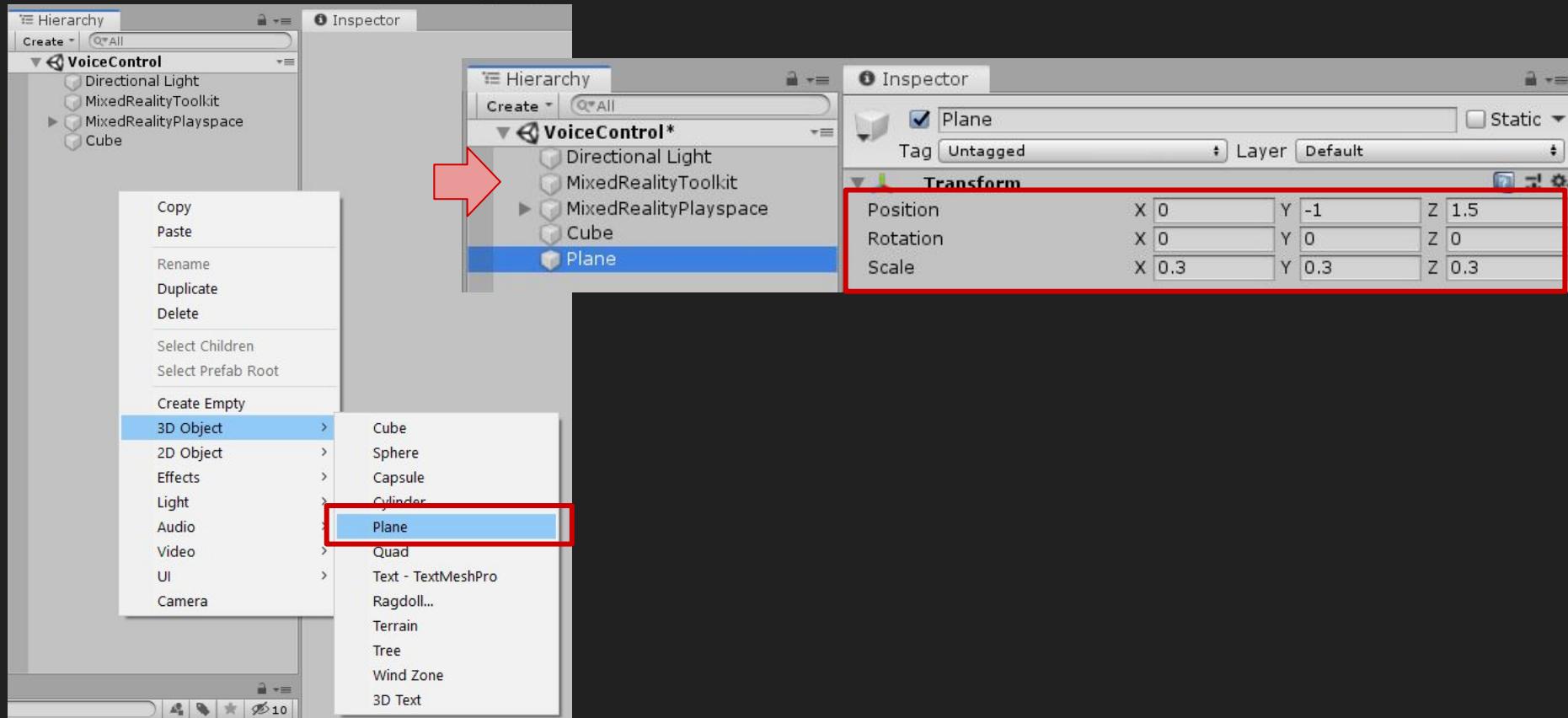
Voice Control

- Enable “Focus Individual Collider in input Profile”.



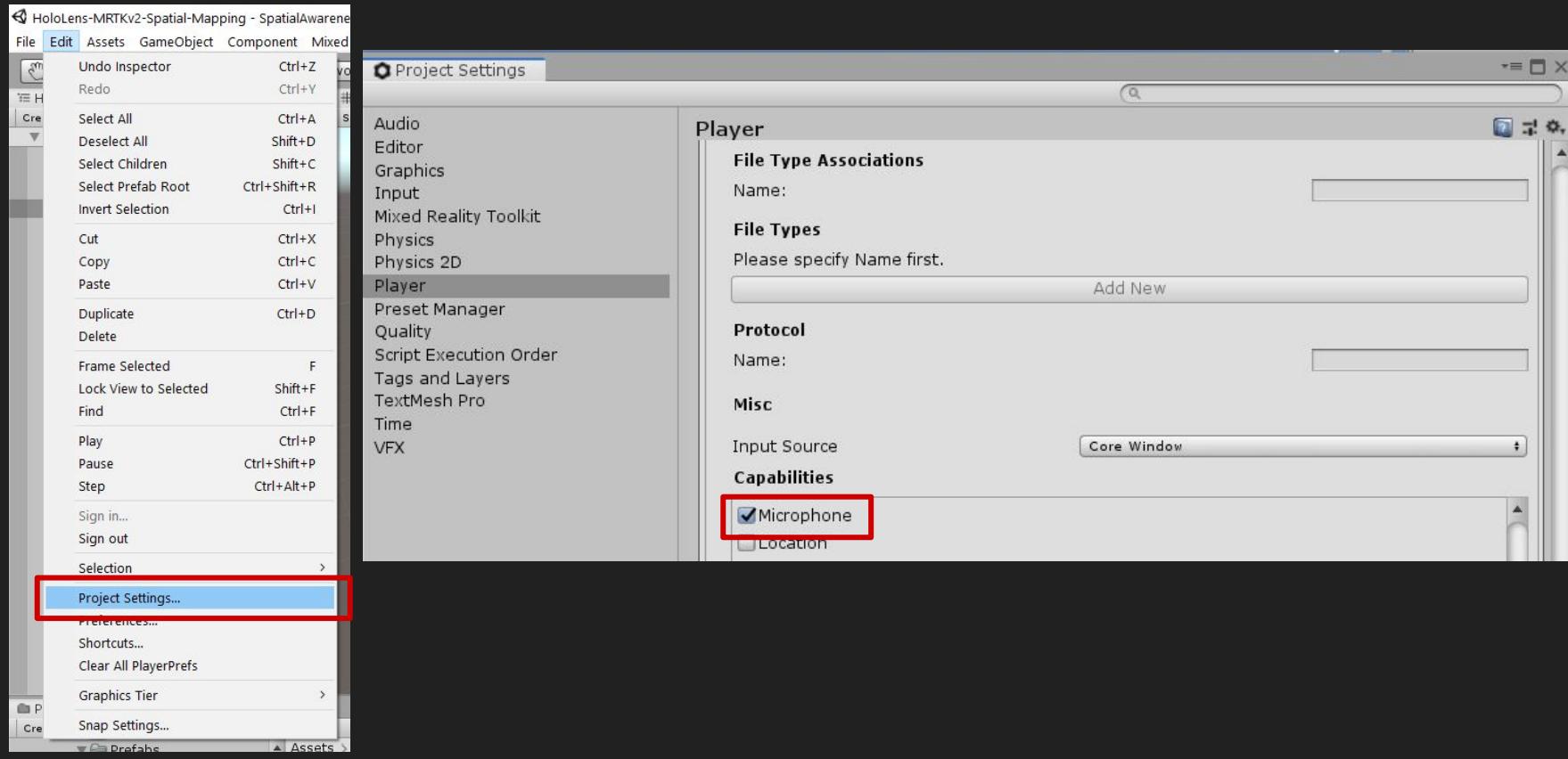
Voice Control

- Create a new Plane object.

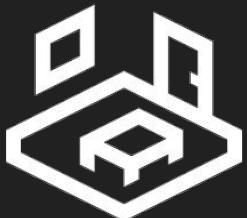


Voice Control

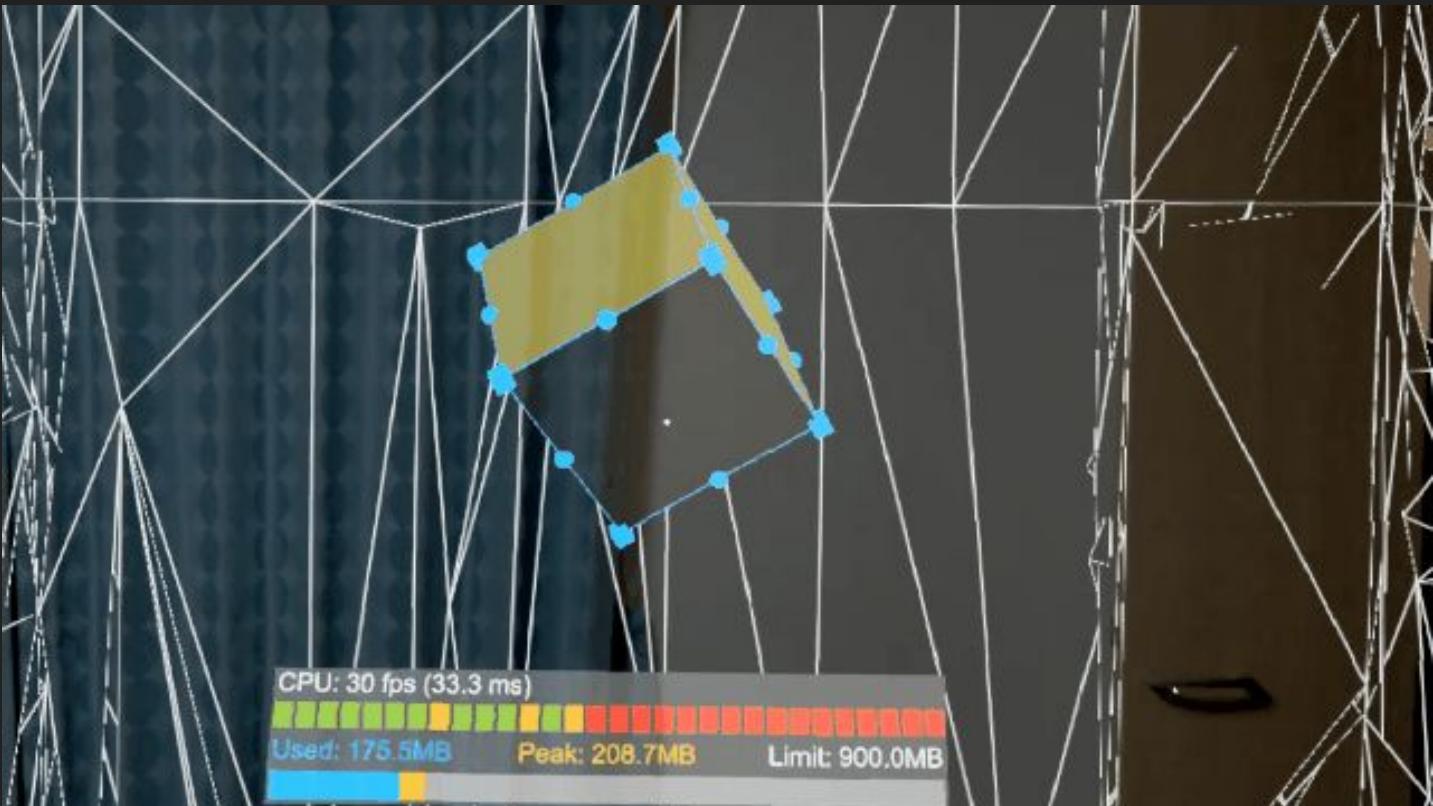
- Enable “Microphone” in Player setting.



Demo

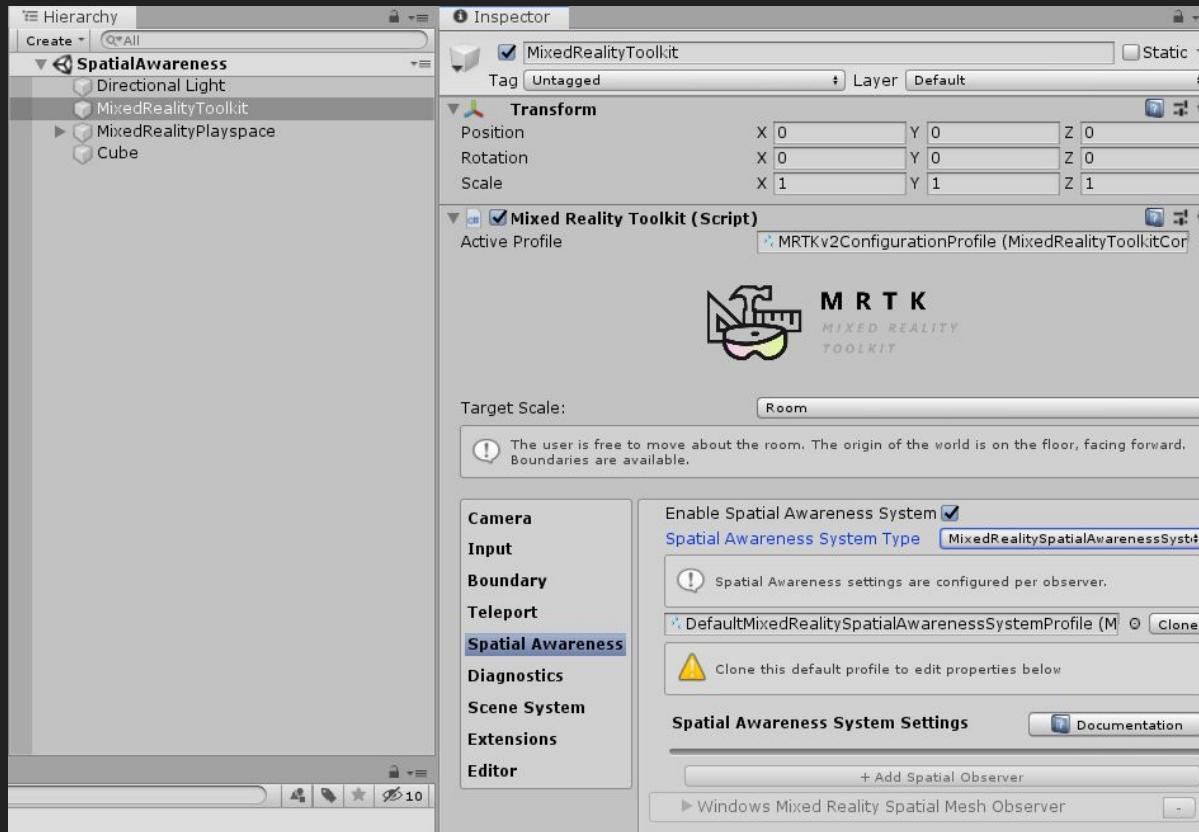


Spatial Awareness



Spatial Awareness

- Check “Enable Spatial Awareness System”



Spatial Awareness

- Clone a Profile as named “MRTKv2SpatialAwarenessProfile”.

