

Mixed Reality & Microsoft HoloLens

Laboratorio di Realtà Virtuale

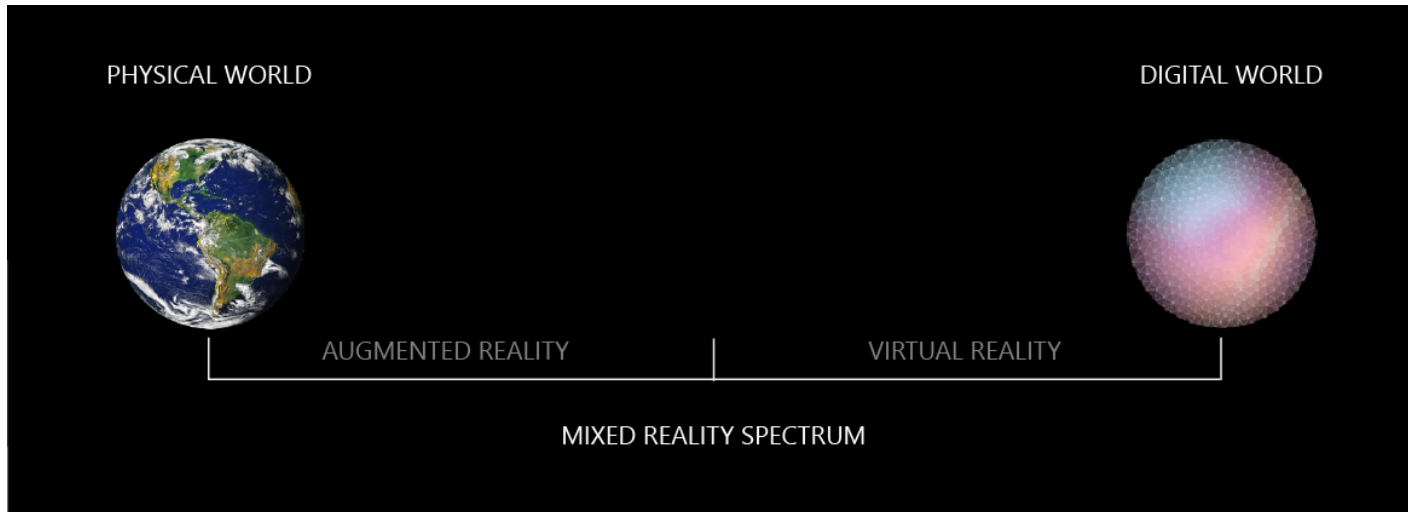


Manuel Pezzera – manuel.pezzera@unimi.it

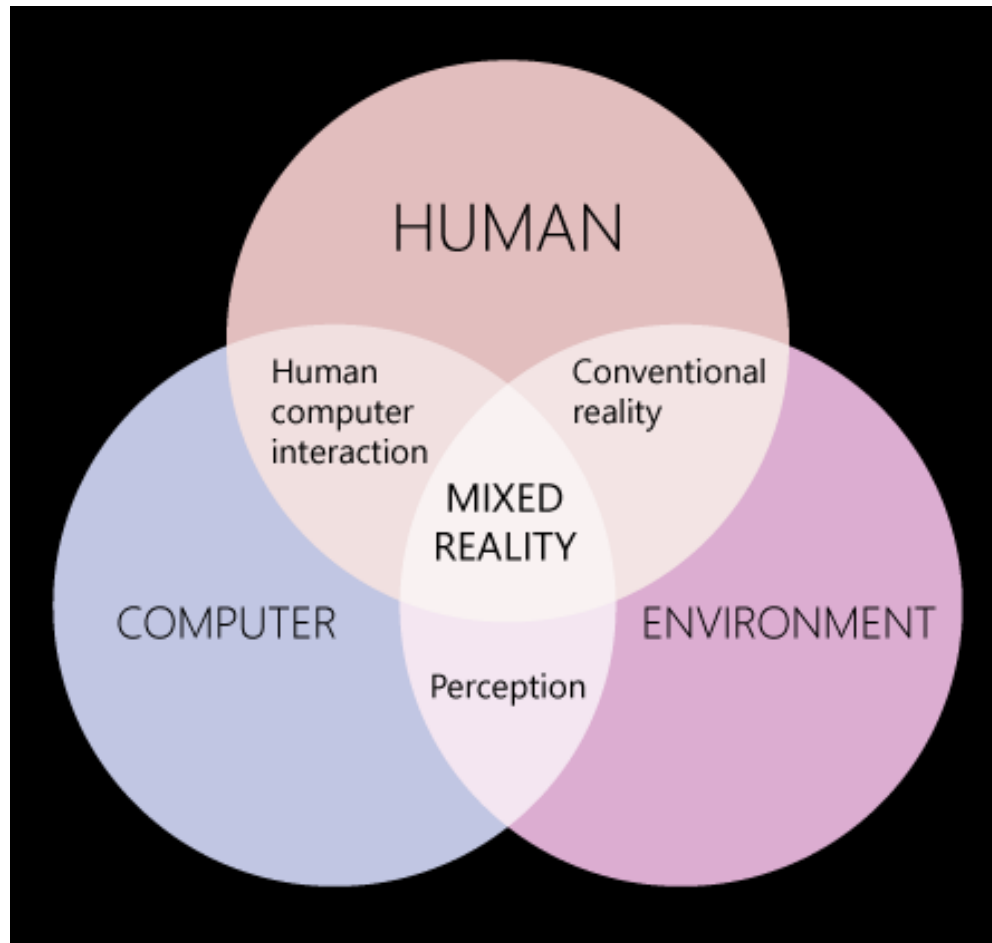


Mixed Reality

- *“Mixed reality (MR) is a spectrum of immersive experiences, connecting and blending physical and digital worlds together in augmented-reality and virtual-reality applications” [cit. Microsoft]*

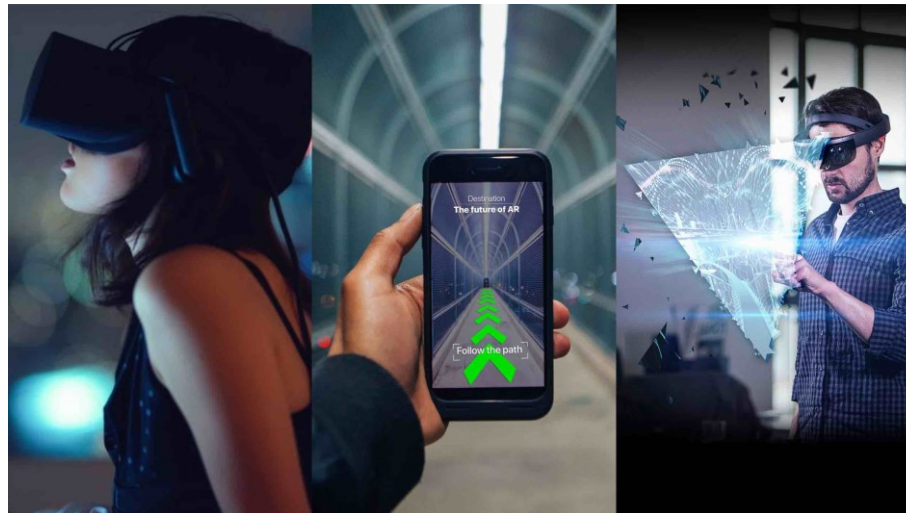


Mixed Reality



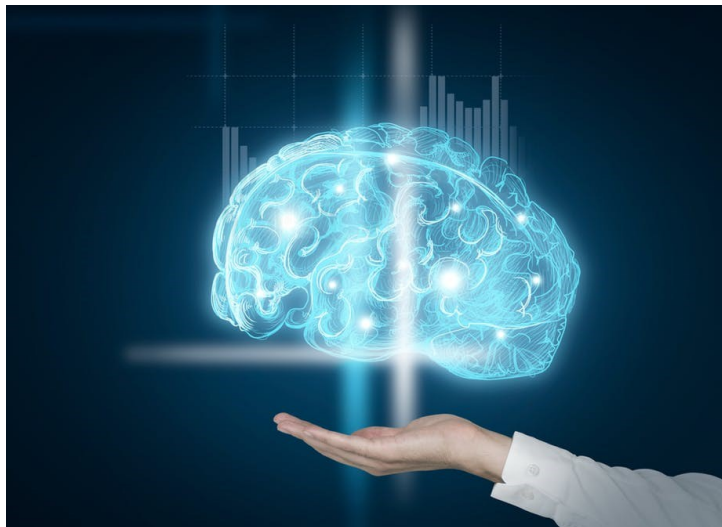
Augmented and Virtual Reality

- Virtual Reality: it tricks your eyes into thinking that you are somewhere else.
- Augmented reality: it is an extra dimension over VR since it can show the real world around wearer with graphic overlaid.
- AR can also totally immerse user just like VR by simply showing pixels everywhere with no transparency to the real world.
- In virtual world, users are best advised to stay seated to avoid collision with physical objects which they cannot see in the real world.



What is a hologram?

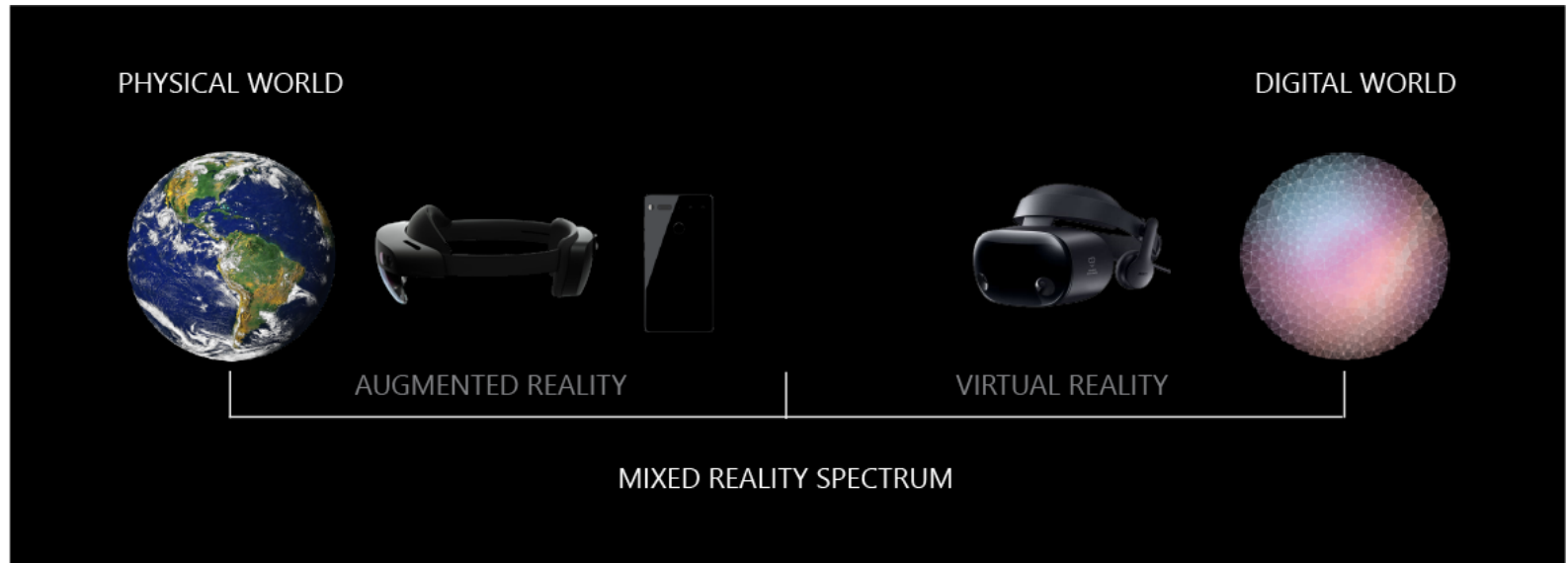
- A hologram is an object like any other object in the real world, with only one difference: instead of being made of physical matter, a hologram is made entirely of light.
- Holographic objects can be viewed from different angles and distances, just like physical objects. They can also interact with the environment (e.g., bouncing ball on a table)
- Hologram do not offer any physical resistance when touched or pushed because they don't have a mass.



Mixed reality vs Augmented Reality

- MR can be mentioned as an enhanced form of Augmented Reality (AR). In AR the visible natural environment is overlaid with a layer of digital content.
- But the case is slightly different with MR, as virtual objects are integrated into and responsive to the real world. For instance, a virtual ball under your desk would be blocked from a user's view unless he bends.
- MR takes the best qualities of AR and VR to create an immersive interface that overlays upon the user's reality. Rather than displaying simple images like AR, MR strives to put fully digital objects that are trackable and intractable in the user's environment.
- Users can view and manipulate things from different angles which is as complex as an anatomy model. This type of rendering demands more processing power than AR, which is one main reason MR applications and devices are still in the proof-of-concept phase and far from consumer availability.
- AR experiences aren't limited to visual additions to our world. You can create augmented experiences that are only audio additions to your physical world, or both audio and visual.

Mixed reality devices



What is Microsoft HoloLens?

- Microsoft HoloLens is the first holographic computer running Windows 10.
- It is completely unattached – no wires, phones, or connection to a PC needed.
- Microsoft HoloLens allows user to pin holograms in the physical environment and provides a new way to see his world.
- Microsoft HoloLens support augmented reality.



What is Microsoft HoloLens?



**New HoloLens
Demo is Crazy Cool**

<https://www.youtube.com/watch?v=29xnzxgCx6I>

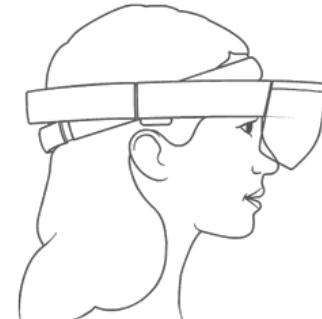
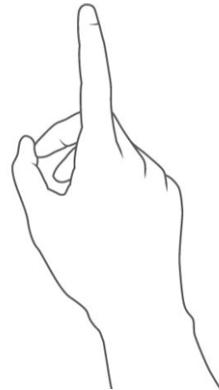
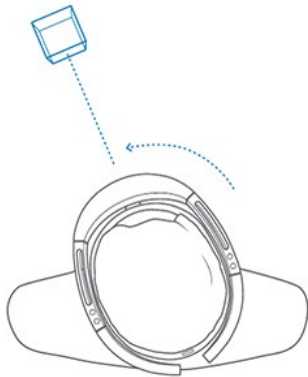
HoloLens

- Released in 2016
- Mixed reality head-mounted device from Microsoft.
- Self-contained and untethered.
- Sensors analyze real-world.
- Multi-Lens collaboration.
- Runs on Windows 10.



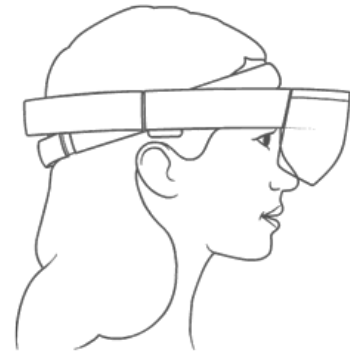
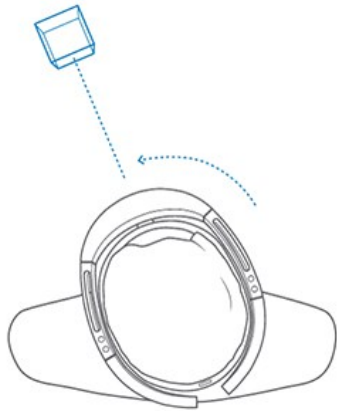
HoloLens 1

- Input: GGV paradigm
 - Gaze (look at)
 - Gesture (e.g., air tap)
 - Voice (speech)
- Output
 - Two see-through lenses
 - Spatial Sound

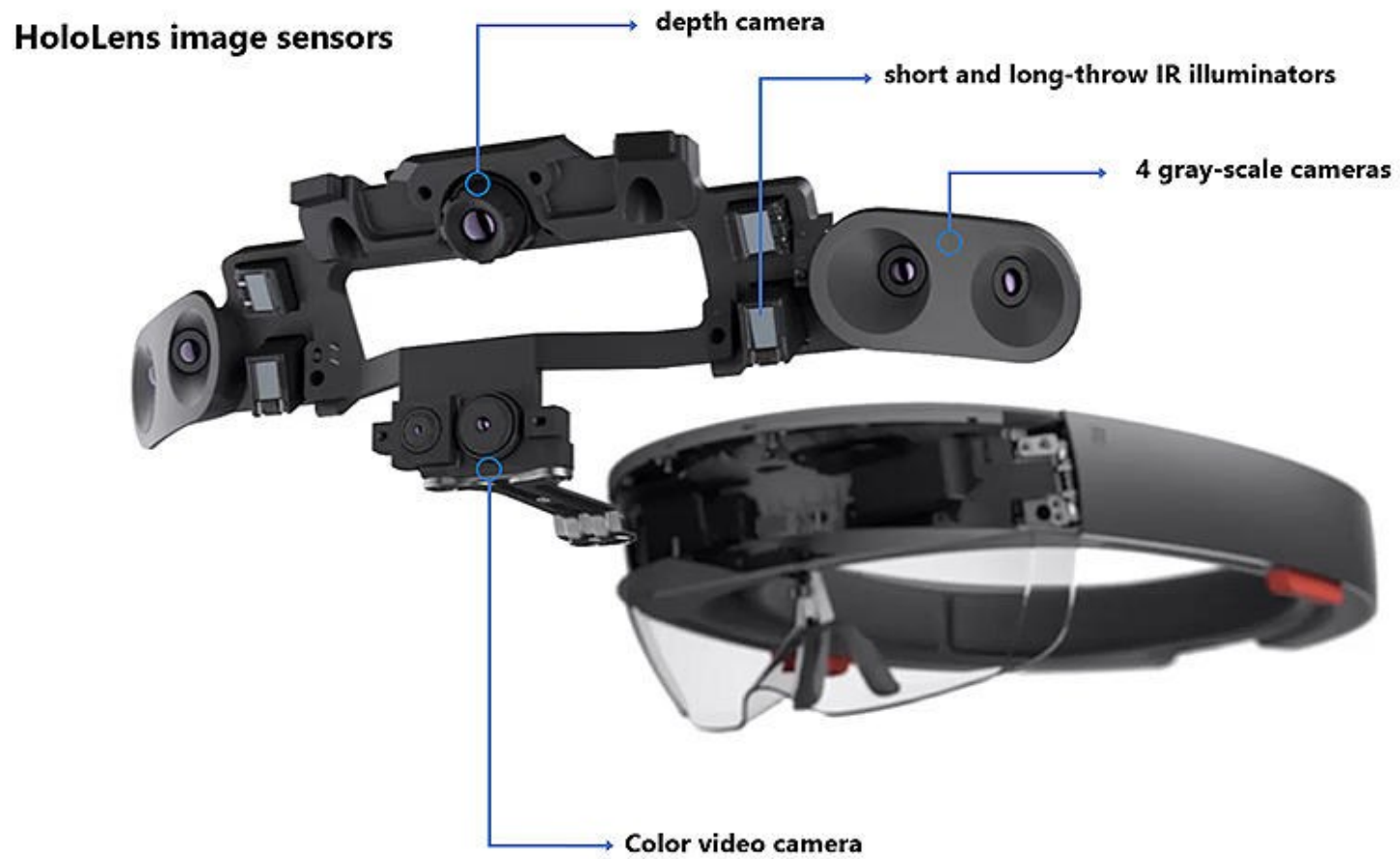


Developing for HoloLens

- Direct3D with C++ or C# via SharpDX.
- Unity
 - Highly efficient workflow
 - Cross-platform
 - Built-in VR support since v5.1



HoloLens components



Technical specification

- Tech specifications:
 - CPU: Intel Atom Cherry Trail 2.7 Ghz
 - 64 GB Flash
 - 2 GB Memory
 - 1 GB HPU: Holographic Processing Unit
 - See through high-definition holographic lenses
 - Infrared Optics
 - Battery (2-3 hours active, 2 weeks standby)

Technical specification

- Sensors:
 - Gyroscope
 - Magnetometer
 - Accelerometer
 - Ambient Light Detector
 - Microphone Array (4 mics)
 - Grayscale Cameras (4)
 - RGB cameras (1)
 - Depth sensor (1)

How does it work?

Sensors

- Sensors track where the wearer is looking and adjust the display.
- Motion sensor detect wearers movement.
- The sensor can also see wearers hands, the hands are an input system
 - User can interact with whatever he sees by just touching it.
- Wearer also give gesture as input thanks to the tracking of user movement.

Computer

- HoloLens is not just a visor connected to a computer; it is a computer on its own.
- It contains CPU, GPU, battery, and first of its kind HPU (holographic processing unit)

How does it work?

Camera

- The HoloLens depth camera has a field of vision that spans 120 by 120 degrees, so it can sense what your hands are doing even when they are nearly outstretched.

Vent

- The device is more powerful than a laptop (compared to a 2016 laptop) but won't overheat, warm air flows to the sides, where it vents up and out.

Buttons

- On the right side buttons allow user to adjust the volume and to control the contrast of the holograms.

Gestures: Air Tap

- HoloLens currently recognizes two core component gestures - **Air tap** and **Bloom**. These two core interactions are the lowest level of spatial input data that a developer can access. They form the foundation for a variety of possible user actions.



1. Finger in the ready position



2. Press finger down to tap or click

Air tap



Bloom

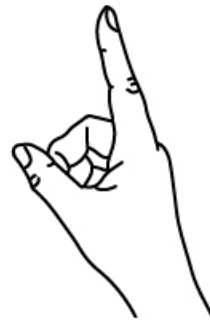
Gestures

Gaze-and-commit

- To take actions, hand gestures use head gaze as the targeting mechanism. The combination of **Gaze** and the **Air tap** gesture results in a **gaze-and-commit** interaction. An alternative to gaze-and-commit is **point-and-commit**, enabled by motion controllers. Apps that run on HoloLens only need to support gaze-and-commit since HoloLens does not support motion controllers. Apps that run on both HoloLens and immersive headsets should support both gaze-driven and pointing-driven interactions, to give users choice in what input device they use.

Gestures: Air Tap

- Air tap is a tapping gesture with the hand held upright, like a mouse click or select. This is used in most HoloLens experiences for the equivalent of a "click" on a UI element after targeting it with [Gaze](#). It is a universal action that you learn once and then apply across all your apps. Other ways to perform select are by pressing the single button on a [HoloLens Clicker](#) or by speaking the voice command "Select".



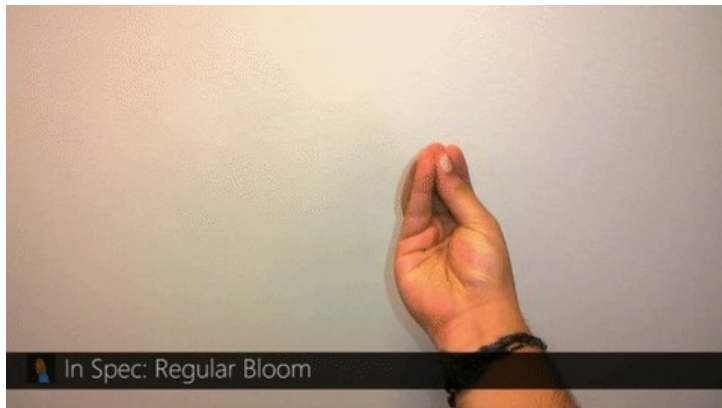
1. Finger in the ready position



2. Press finger down to tap or click

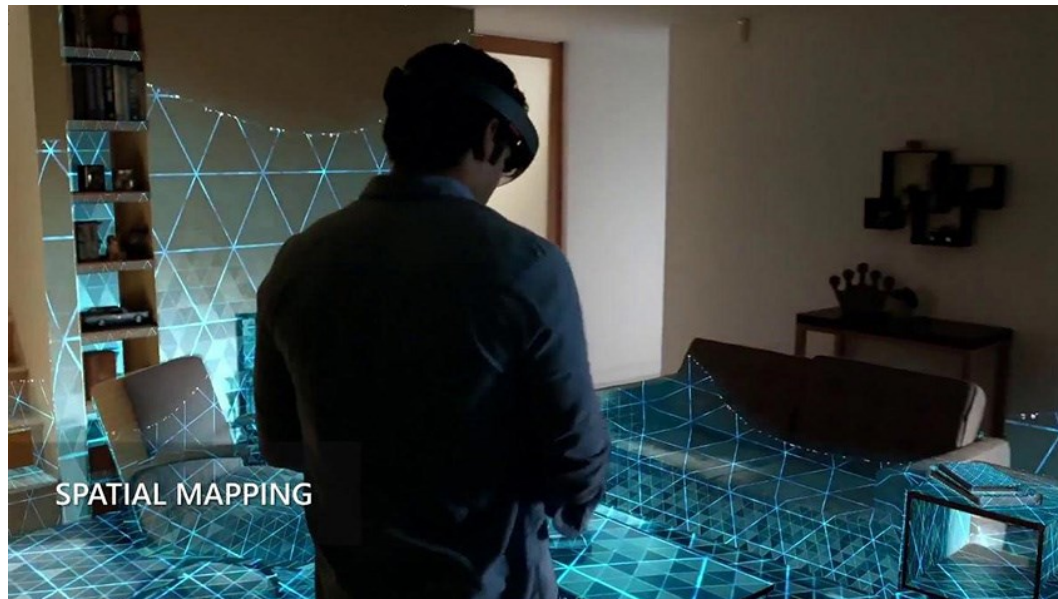
Gestures: Bloom

- Bloom is the "home" gesture and is reserved for that alone. It is a special system action that is used to go back to the Start Menu. It is equivalent to pressing the Windows key on a keyboard or the Xbox button on an Xbox controller. The user can use either hand.



Spatial perception: how does it work?

- Spatial perception and mapping
- Device scans your environment and builds a digital model in real time
 - It allows HoloLens to see different surfaces, walls and ceiling
 - Meshes & planes
- Possible to simulate a physical space
 - E.g., projecting a terrain over your floor



Spatial Mapping



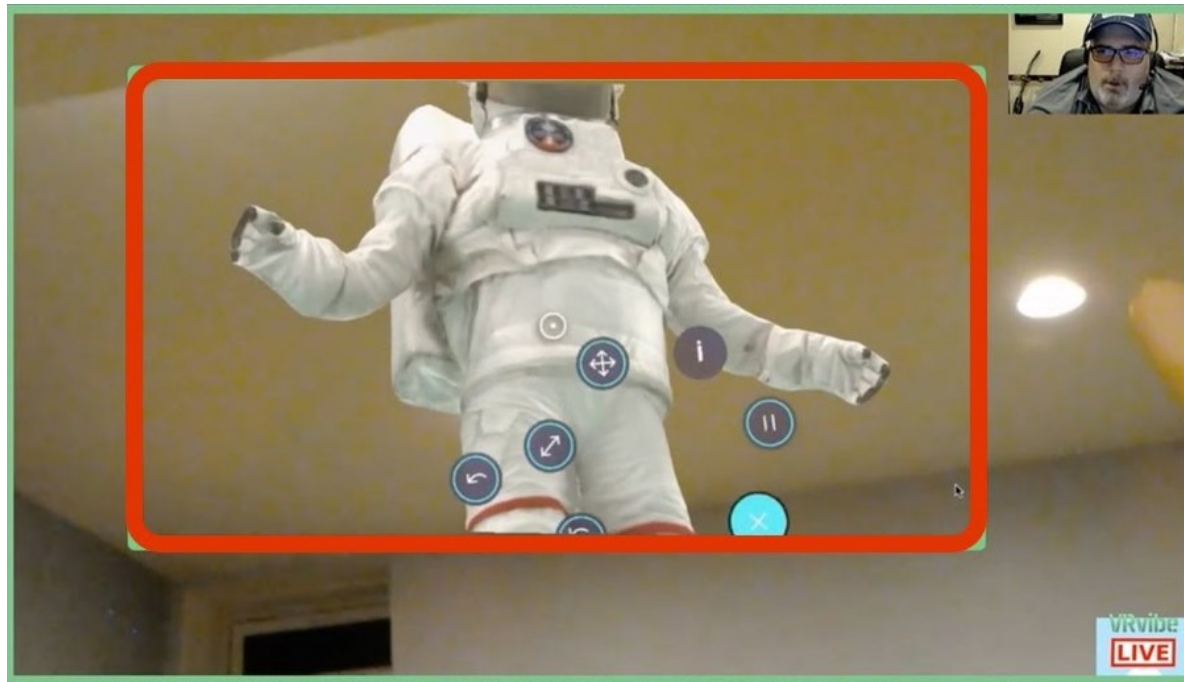
<https://www.youtube.com/watch?v=AFzR-r5Y2sk>

HoloLens: Advantages

- No wires.
- Transparent lens and advanced sensors allow you to see your world and move confidently in it.
- Lightweight and adjustable to fit any adult head size.
- Built-in spatial sound lets you hear holograms wherever they are in the room with pinpoint precision.

Problems

- However, Microsoft HoloLens is the best technology till yet, it has certain technical and non-technical problems.
- The biggest problem is the narrow field of view.



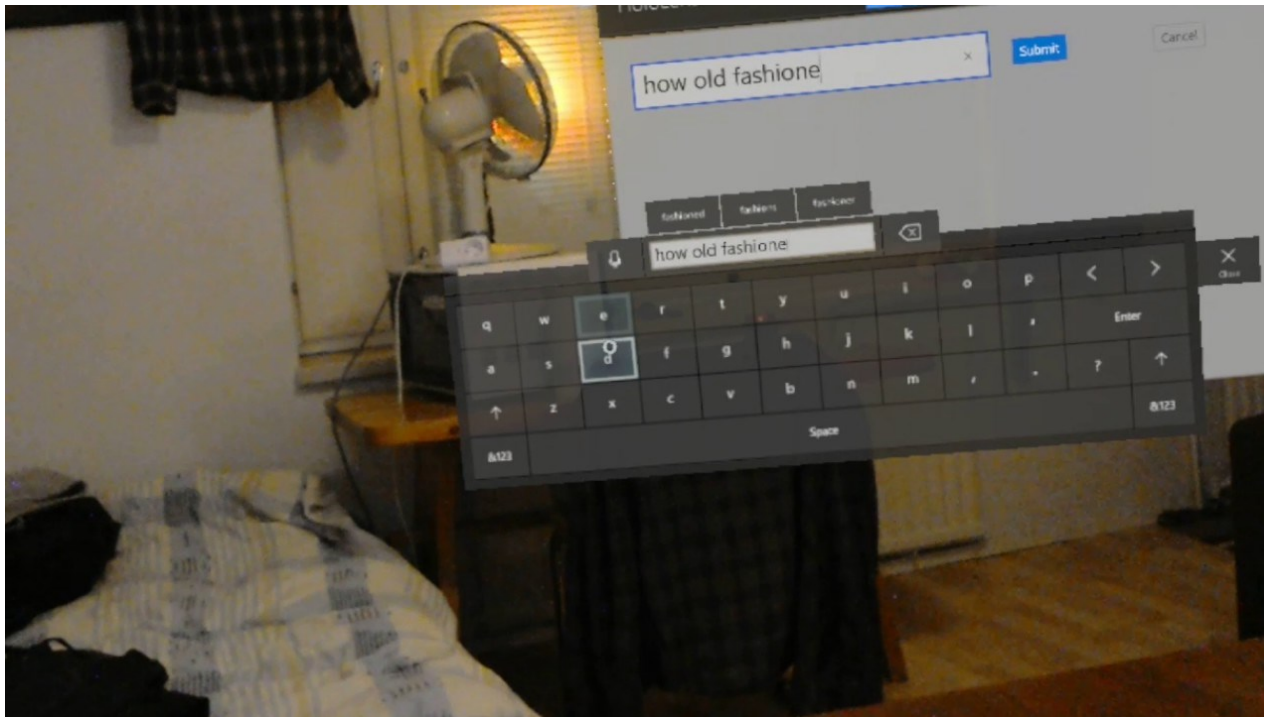
Problems



https://www.youtube.com/watch?v=RLneQuA3T50&ab_channel=VRvibe

Problems 2

- Another big problem is the GGV paradigm.
- Gaze and click is very slow and uncomfortable in the long time.



Problems 3

- Also, HoloLens itself is not quite comfortable.
- It is pretty heavy, and all the weight is in front.



HoloLens 2

- Presented on February 24, 2019.
- Only for enterprise use.
- Microsoft has improved on the previous technical, visual, and functional aspects of HoloLens 1 making it much more comfortable device for everyday usage.

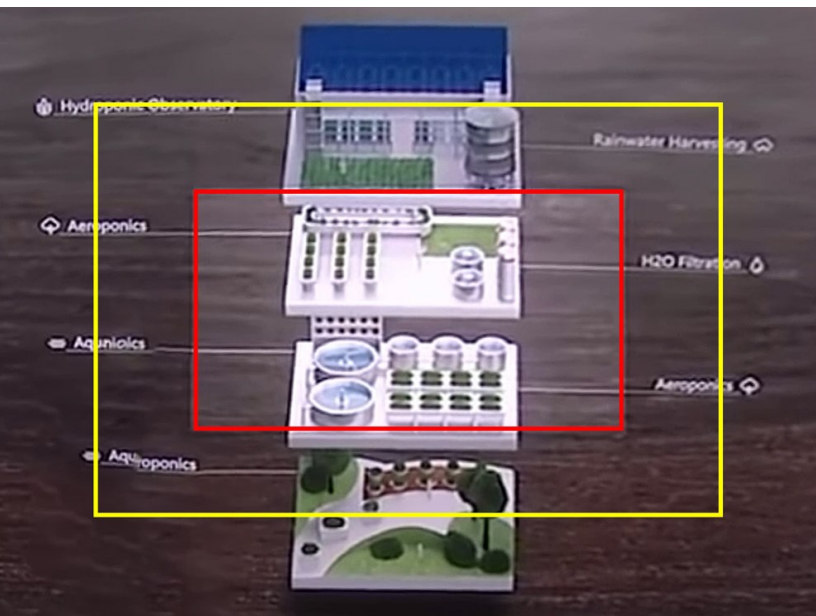


HoloLens 2

- Resolution – 2k 3:2 light engines
- Depth – 1-MP time-of-flight (ToF) depth sensor
- IMU – Accelerometer, Gyroscope, Magnetometer
- Camera – 8MP stills, 1080p30 video
- Microphone Array
- Speakers – Built-in, Spatial Audio
- **Hand Tracking** – Two-handed fully articulated model, direct manipulation
- **Eye Tracking** – Real-time tracking
- Voice – Command and control on-device, Natural Language with internet connectivity
- SoC: Qualcomm Snapdragon 850 Compute Platform
- HPU: 2nd Generation Custom-built Holographic Processing Unit
- USB: USB Type-C

HoloLens vs HoloLens 2: Field of View

- In the first version of HoloLens, you could see a really limited “square” where you had to try your best to fit the hologram into. Even the smallest head movement would make the hologram go out of sight making it extremely uncomfortable to work with the holograms.
- Now, the range of view nearly doubled making it much easier to perceive the virtual environment.



HoloLens vs HoloLens 2: Control and interaction

- The control system has changed significantly. In the first version of HoloLens, you were pretty much limited to a few gestures that you could use to control the virtual objects and interact with the computer-generated environment.
- HoloLens 2 went much further with the gesture manipulation allowing users to interact with the holograms in a natural way:
 - Resizing and scaling (using both hands as if unfolding a map)
 - Grabbing, dragging, holding (as if taking a cup of tea)



HoloLens vs HoloLens 2: Control and interaction

- Not only HoloLens is able to recognize your hands now, but it's also capable of tracking your eyes so that you can scroll social media feed just by looking at the bottom of the page.
- Also, a voice assistant was implemented that provides a wide range of commands, not the least of which is ordering your browser to follow you as you navigate within the virtual environment.



HoloLens vs HoloLens 2: Control and interaction



<https://www.youtube.com/watch?v=uIHPPtPBgHk>

HoloLens vs HoloLens 2: Mounting and Stabilization

- The new HoloLens device combines several key aspects that improve its wearing time and comfortability. Not only the weight of the device was reduced, but also the regulating mechanism was moved to the back of the head making the whole device well-balanced and convenient to use in terms of size and position adjustability.
- Also, the new version has a portable forehead gasket which makes HoloLens 2 more comfortable for everyday wearing. The gasket has a hygienic function too since it helps to hold the sweat and because the gasket is portable, it's convenient to clean it regularly.



Alternatives: Magic Leap One Creator Edition

- Available from 2018, only in some US cities.
- 2295\$
- Composed by:
 - Lightpack
 - Lightwear
 - Controller



Alternatives: Magic Leap One Creator Edition



- <https://www.youtube.com/watch?v=08ShnGBNo-c>

Magic Leap One Creator Edition

Lightpack

- CPU NVIDIA® Parker SOC; 2 Denver 2.0 64-bit cores + 4 ARM Cortex A57 64-bit cores (2 A57's and 1 Denver accessible to applications)
- GPU NVIDIA Pascal™, 256 CUDA cores; Graphic APIs: OpenGL 4.5, Vulkan, OpenGL ES 3.3+
- RAM 8 GB
- Storage Capacity 128 GB (actual available storage capacity 95GB)
- Power: Built-in rechargeable lithium-ion battery. Up to 3 hours continuous use. Battery life can vary based on use cases. Power level will be sustained when connected to an AC outlet. 45-watt USB-C Power Delivery (PD) charger
- Connectivity: Bluetooth 4.2, WiFi 802.11ac/b/g/n, USB-C

Lightwear

- Audio Input: Voice (speech to text) + real world audio (ambient)
- Audio Output: Onboard speakers and 3.5mm jack with audio spatialization processing

Magic Leap One Creator Edition



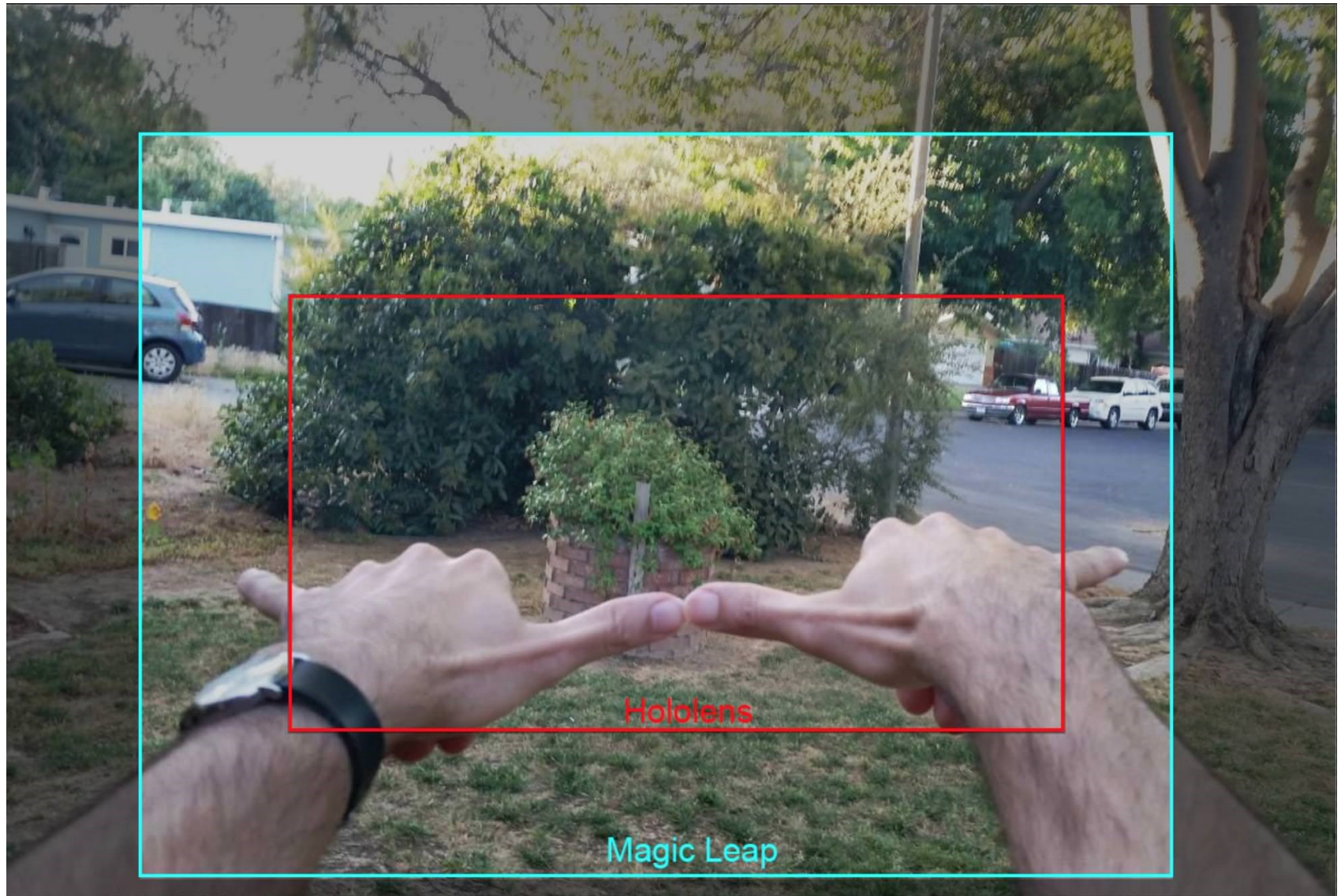
https://www.youtube.com/watch?v=GmdXJy_IdNw

Magic Leap One Creator Edition



<https://www.youtube.com/watch?v=D5Q7Edvn87g&feature=youtu.be>

Magic Leap One Creator Edition: Field of View



Unity & HoloLens

- Let's see how HoloLens in Unity works!



Mixed Reality Toolkit



<https://youtu.be/qfONIUCSWdg>

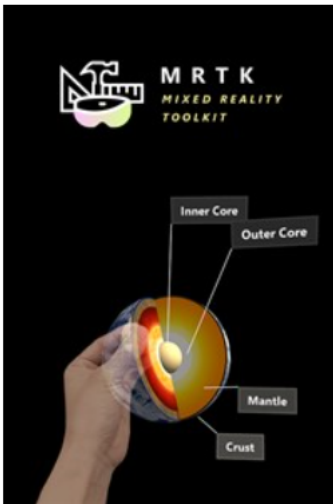
Mixed Reality Toolkit (MRTK)

MRTK-Unity is a Microsoft-driven project that provides a set of components and features, used to accelerate cross-platform MR app development in Unity. Here are some of its functions:

- Provides the cross-platform input system and building blocks for spatial interactions and UI.
- Enables rapid prototyping via in-editor simulation that allows you to see changes immediately.
- Operates as an extensible framework that provides developers the ability to swap out core components.
- Supports a wide range of platforms:
 - OpenXR (Hololens 2 + Windows Mixed Reality headsets)
 - Windows Mixed Reality (Hololens 1 & 2 + Windows Mixed Reality headsets)
 - Oculus Quest
 - OpenVR (Windows Mixed Reality headsets, HTC Vive, Oculus Rift)
 - Ultraleap Hand Tracking
 - iOS & Android

MRTK Examples Hub

In the very lucky case that you own HoloLens 1 or 2, you can download the MRTK Examples Hub, which provides a set of examples developed with the MRTK.



MRTK Examples Hub

Microsoft Design Labs • [Developer tools](#) > [Development kits](#)

[Share](#) [Wish list](#)

MRTK is an open-source project that provides building blocks for spatial interactions and UI in mixed reality. With the MRTK Examples Hub, you can explore MRTK's various types of example scenes. Visit <http://aka.ms/MRTK> for more information.



EVERYONE

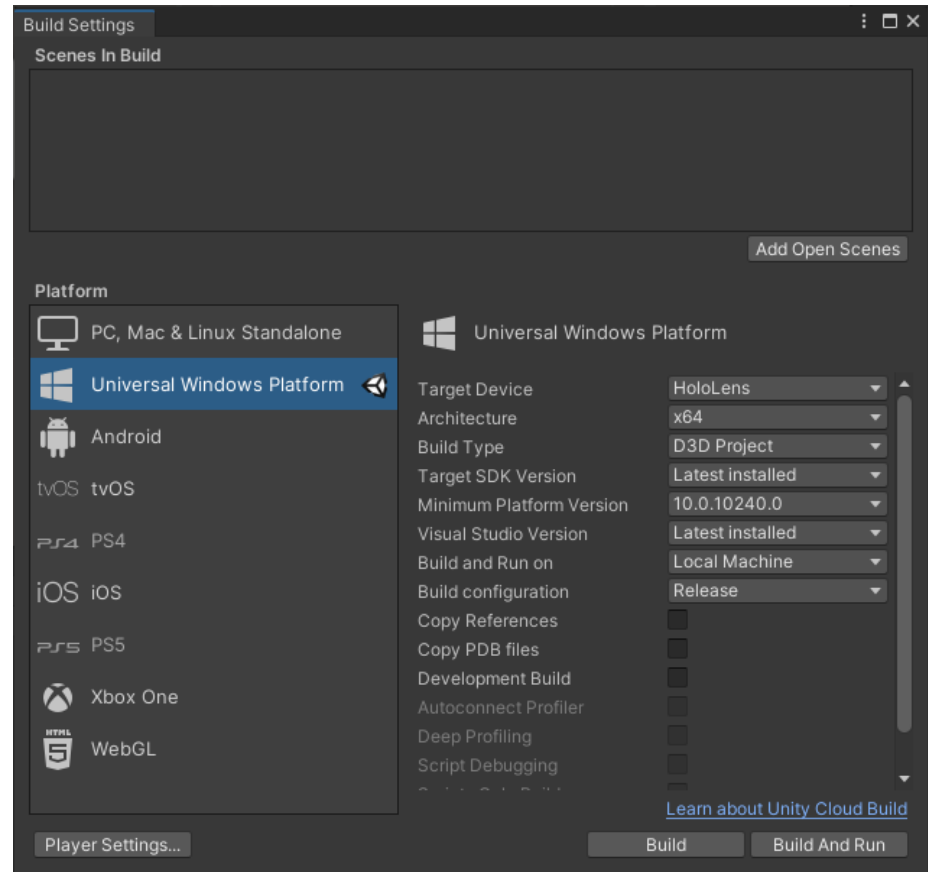
MRTK: Requirements

Requirements to use MRTK:

- Windows 10 (the most recent version as possible)
- Visual Studio 2019 (16.8 or higher)
 - Desktop development with C++
 - Universal Windows Platform (UWP) development
 - USB Device Connectivity (only if you develop with HoloLens)
- Windows 10 SDK (10.0.18362.0 or later)
- HoloLens 2 Emulator (Windows Holographic, version 21H1 May 2021 Update)
- Unity (LTS version recommended), Unreal or Native (OpenXR)
- For detailed information see:
 - <https://docs.microsoft.com/en-us/windows/mixed-reality/develop/install-the-tools?tabs=unity>

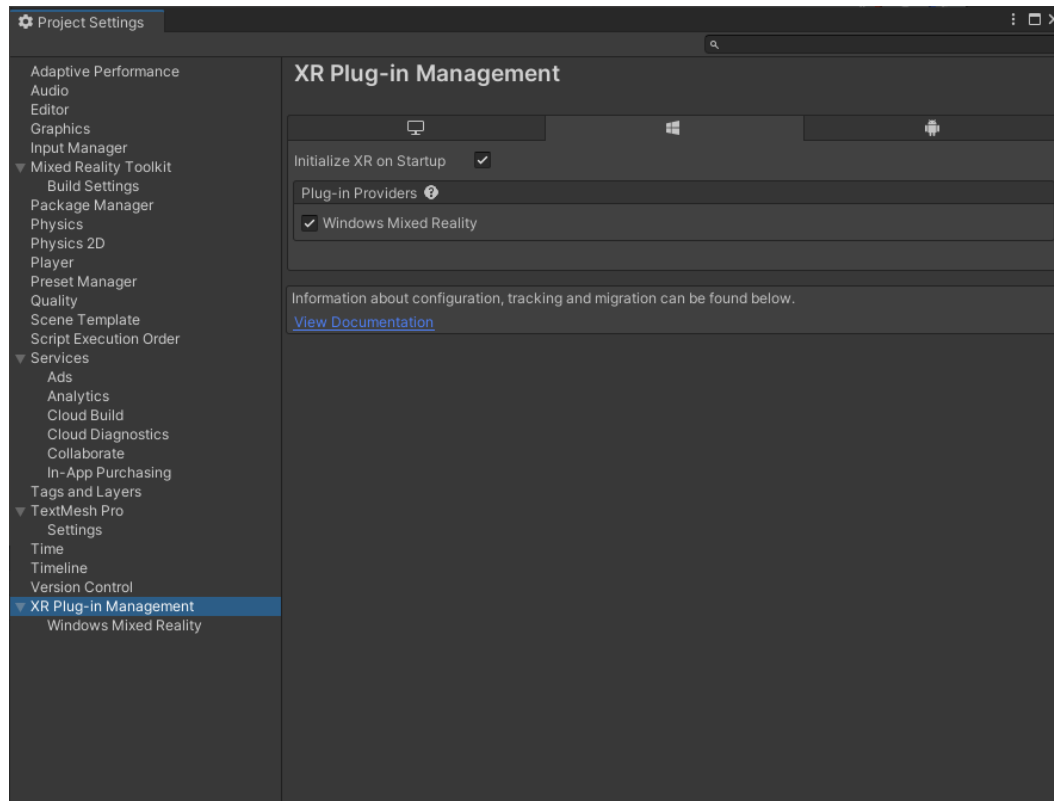
Introduction to MRKT

- Create an empty project with Unity (2019 or newer)
- Open Build Settings, and switch Target to **Universal Windows Platform**.
 - In order to do that, you must have installed the “Universal Windows Platform Build Support” module in your Unity version.
 - To do that, open the Unity Hub -> Install -> Select your unity version -> three-dot icon -> Add modules



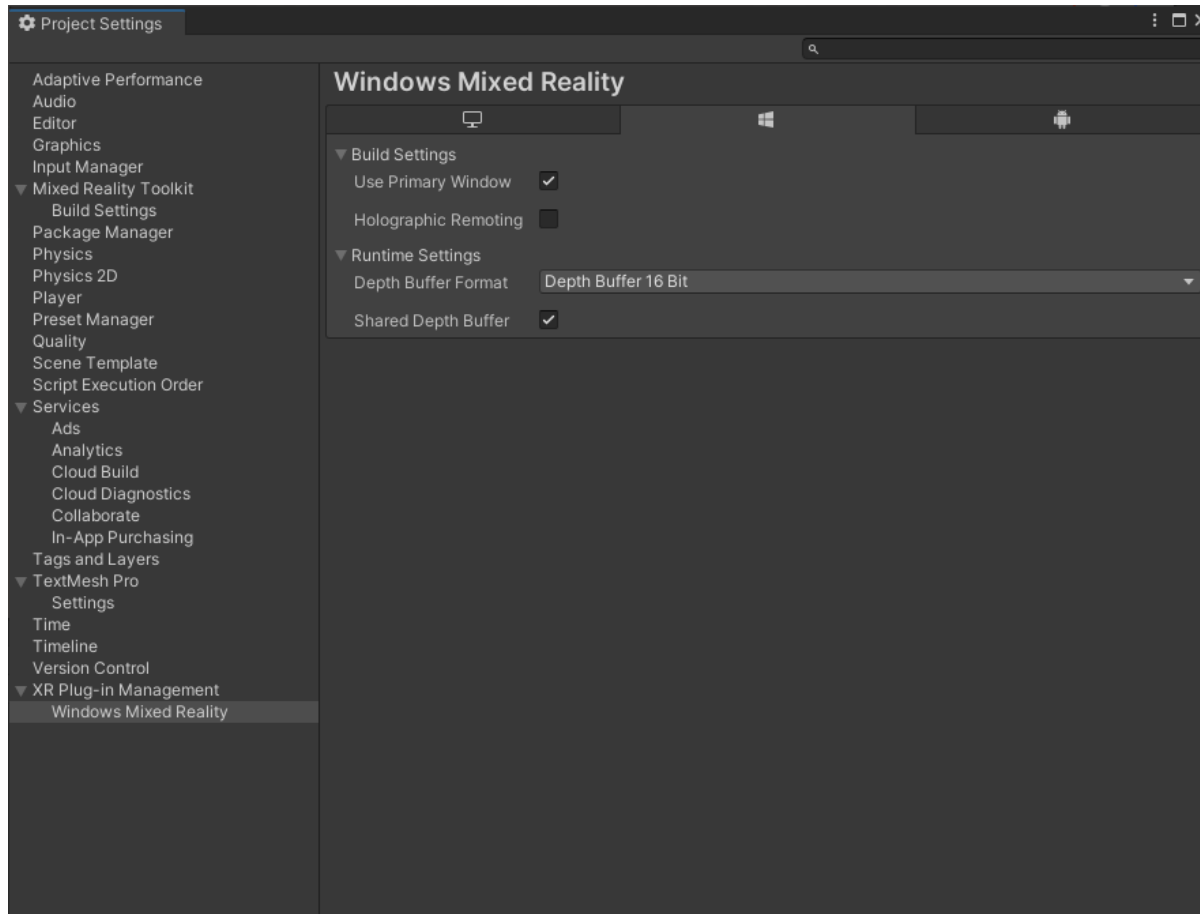
Introduction to MRKT

- Enable the Windows Mixed Reality plugin from Project Settings -> XR Plug-in Management
 - To do that, you must install the XR Plug-in, if you do not have it, you will see a button allowing you to install it.
- If you are using an older version of Unity, this could be different.



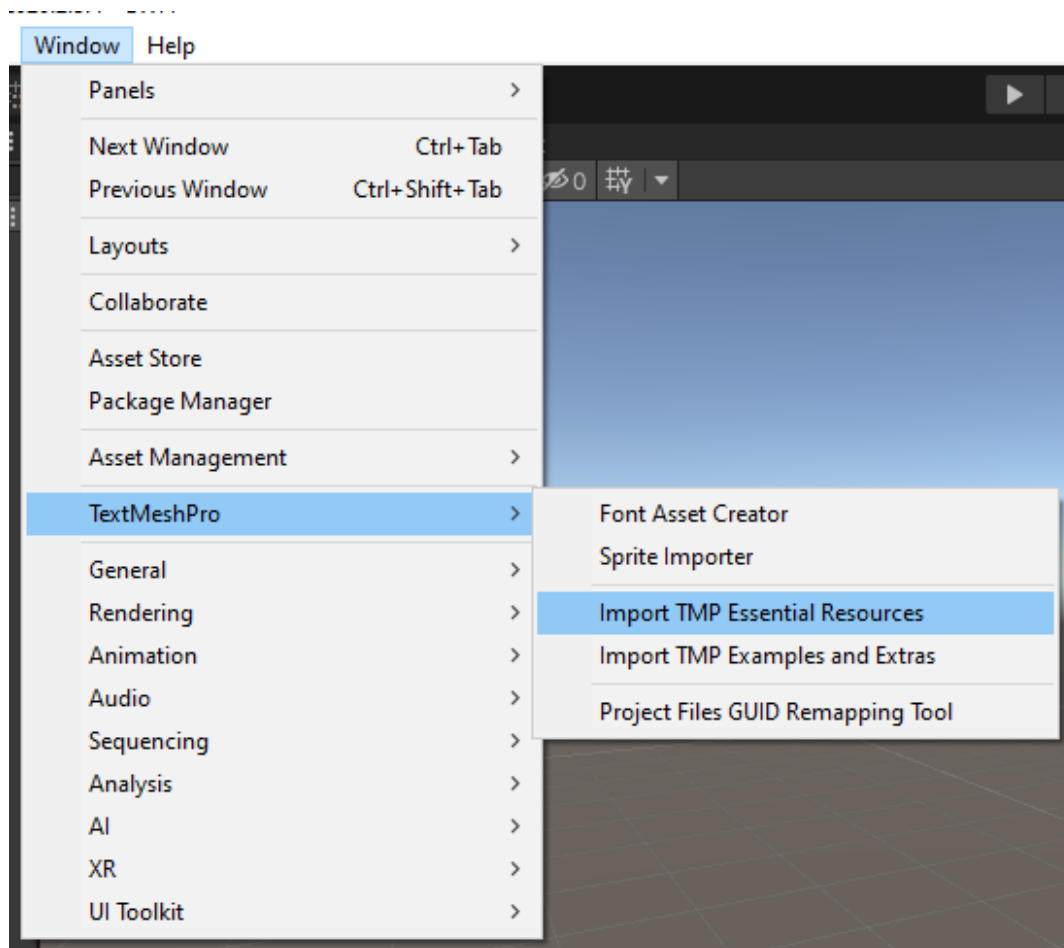
Introduction to MRKT

- To improve performance, ensure that “Depth Buffer 16 bit” is selected and Shared Depth Buffer is enabled.



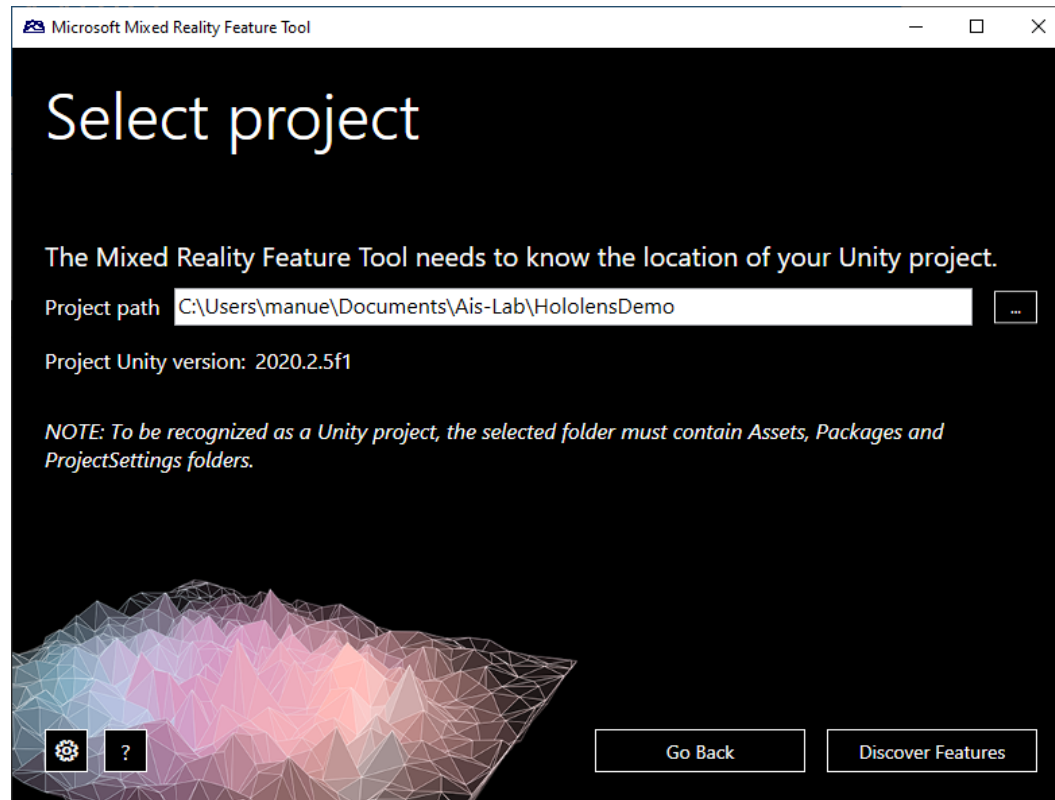
TextMeshPro

- TextMeshPro (TMP) is necessary to use MRTK.
- Import the TMP Essential Resources package from the Window menu (see image)



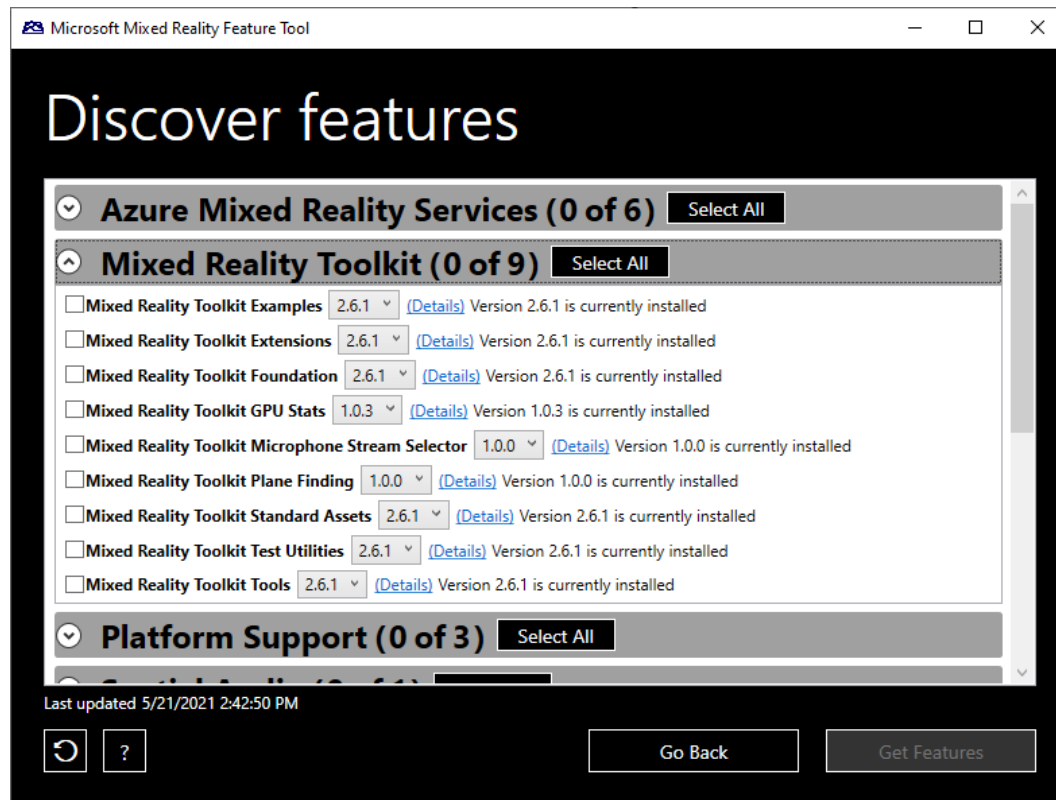
Introduction to MRKT

- Download the [Mixed Reality Feature Tool](#)
- It is a stand-alone tool which allows you to select a Unity Project and to add MRTK to it.
- Select the path of your Unity project and click on “Discover Features”



Introduction to MRKT

- From here you can install all the features you want.
- The only mandatory feature is Mixed Reality Toolkit -> Mixed Reality Toolkit Foundation.
- All the other tools offer additional functionalities



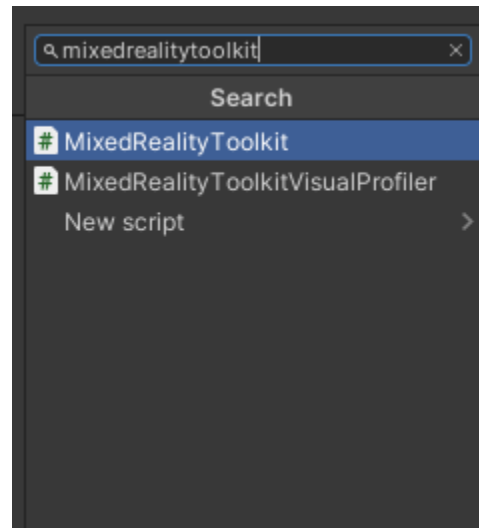
Introduction to MRKT

- Once you have selected the desired features, click on Get Features.
- It will take a while to download and extract all the packages.
- This will update the Unity project manifest.

```
com.unity.modules.video": "1.0.0",  
"com.unity.modules.vr": "1.0.0",  
"com.unity.modules.wind": "1.0.0",  
"com.unity.modules.xr": "1.0.0",  
"com.microsoft.mixedreality.toolkit.examples": "file:MixedReality/com.microsoft.mixedreality.toolkit.examples-2.6.1.tgz",  
"com.microsoft.mixedreality.toolkit.extensions": "file:MixedReality/com.microsoft.mixedreality.toolkit.extensions-2.6.1.tgz",  
"com.microsoft.mixedreality.toolkit.foundation": "file:MixedReality/com.microsoft.mixedreality.toolkit.foundation-2.6.1.tgz",  
"com.microsoft.mixedreality.toolkit.gpustats": "file:MixedReality/com.microsoft.mixedreality.toolkit.gpustats-1.0.3.tgz",  
"com.microsoft.mixedreality.toolkit.micstream": "file:MixedReality/com.microsoft.mixedreality.toolkit.micstream-1.0.0.tgz",  
"com.microsoft.mixedreality.toolkit.planefinding": "file:MixedReality/com.microsoft.mixedreality.toolkit.planefinding-1.0.0.tgz",  
"com.microsoft.mixedreality.toolkit.standardassets": "file:MixedReality/com.microsoft.mixedreality.toolkit.standardassets-2.6.1.tgz",  
"com.microsoft.mixedreality.toolkit.testutilities": "file:MixedReality/com.microsoft.mixedreality.toolkit.testutilities-2.6.1.tgz",  
"com.microsoft.mixedreality.toolkit.tools": "file:MixedReality/com.microsoft.mixedreality.toolkit.tools-2.6.1.tgz",  
"com.microsoft.mixedreality.sceneunderstanding": "file:MixedReality/com.microsoft.mixedreality.sceneunderstanding-0.6.0.tgz",  
"com.microsoft.spatialaudio.spatializer.unity": "file:MixedReality/com.microsoft.spatialaudio.spatializer.unity-1.0.196.tgz",  
"com.microsoft.mixedreality.input": "file:MixedReality/com.microsoft.mixedreality.input-0.9.2006.tgz"  
}  
}
```

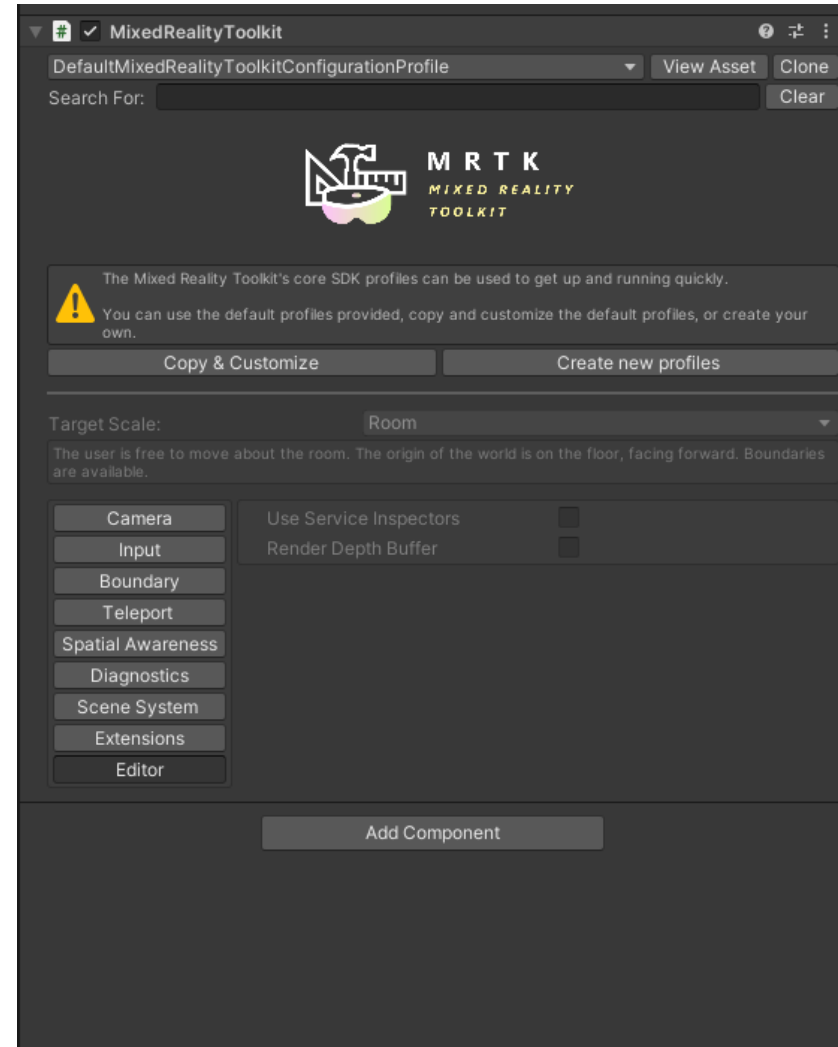
Introduction to MRKT

- Open an empty scene.
- Create a new empty GameObject and add to it the “MixedRealityToolkit” script.

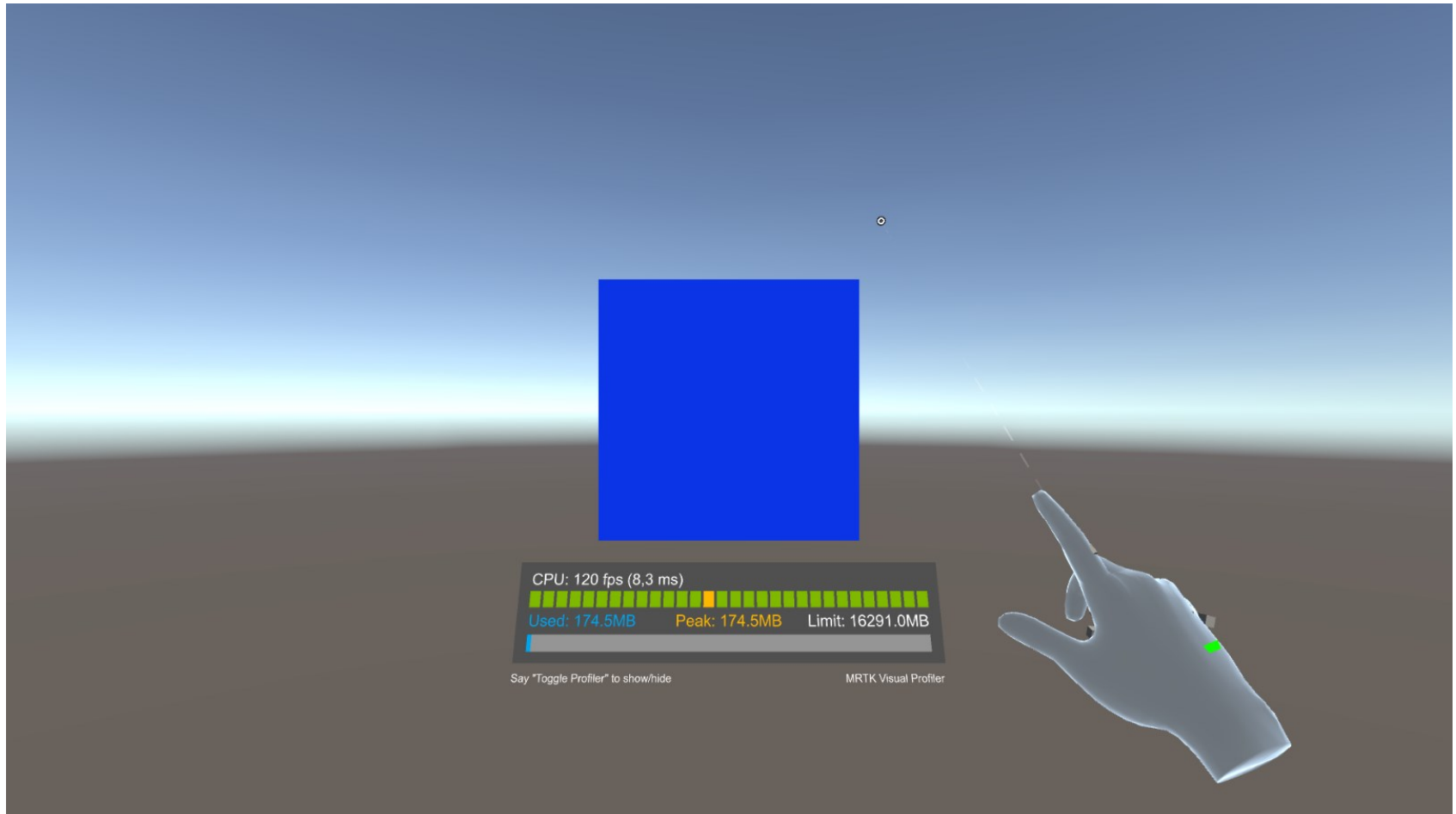


Introduction to MRKT

- This will create two new objects in the hierarchy: “MixedRealityToolkit” and “MixedRealityPlaySpace” (the latter will have the camera as child)
- Select MixedRealityToolkit.
 - On the right, select “DefaultMixedRealityToolkit ConfigurationProfile” (see image)
 - Now click on “Clone”, to clone the profile, in this way you will be able to create your own profile and change it as you want.



Introduction to MRKT



Click on the Play button. By pressing the spacebar, you will see the hand appears.

Unity: Input Simulation

The Unity in-editor input simulation allows you to test holographic object behavior when using hand or eye interaction.

How to move around in the scene:

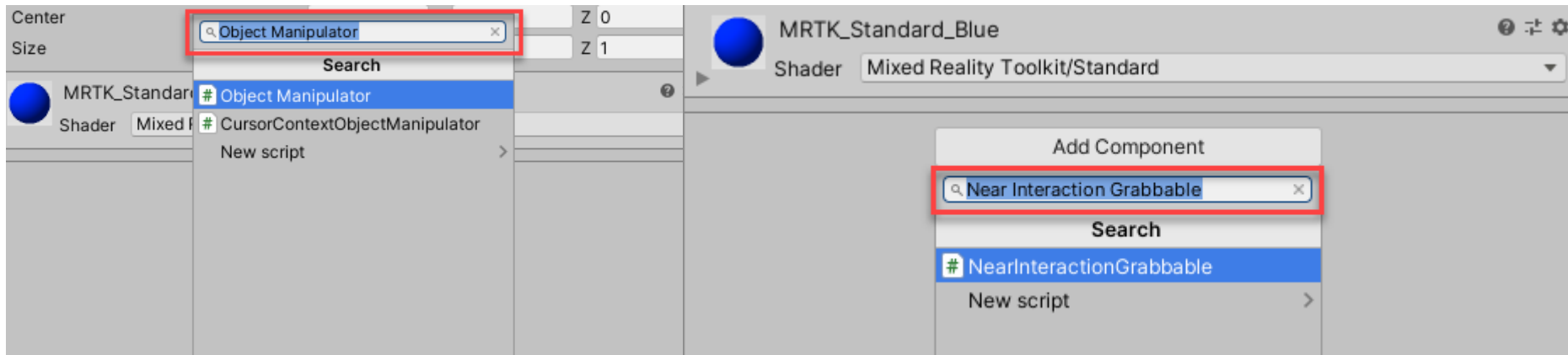
- 1) Use W/A/S/D keys to move the camera forward/left/back/right.
- 2) Use Q/E to move the camera vertically.
- 3) Press and hold the right mouse button to rotate the camera.

How to simulate hand input:

- 1) Press and hold the space bar to enable the right hand.
- 2) While holding the space bar, move your mouse to move the hand.
- 3) Use the mouse scroll wheel to adjust the depth of the hand.
- 4) Click the left mouse button to simulate pinch gesture.
- 5) Use T/Y keys to make the hand persistent in the view.
- 6) Press and hold the left shift key to enable the left hand.

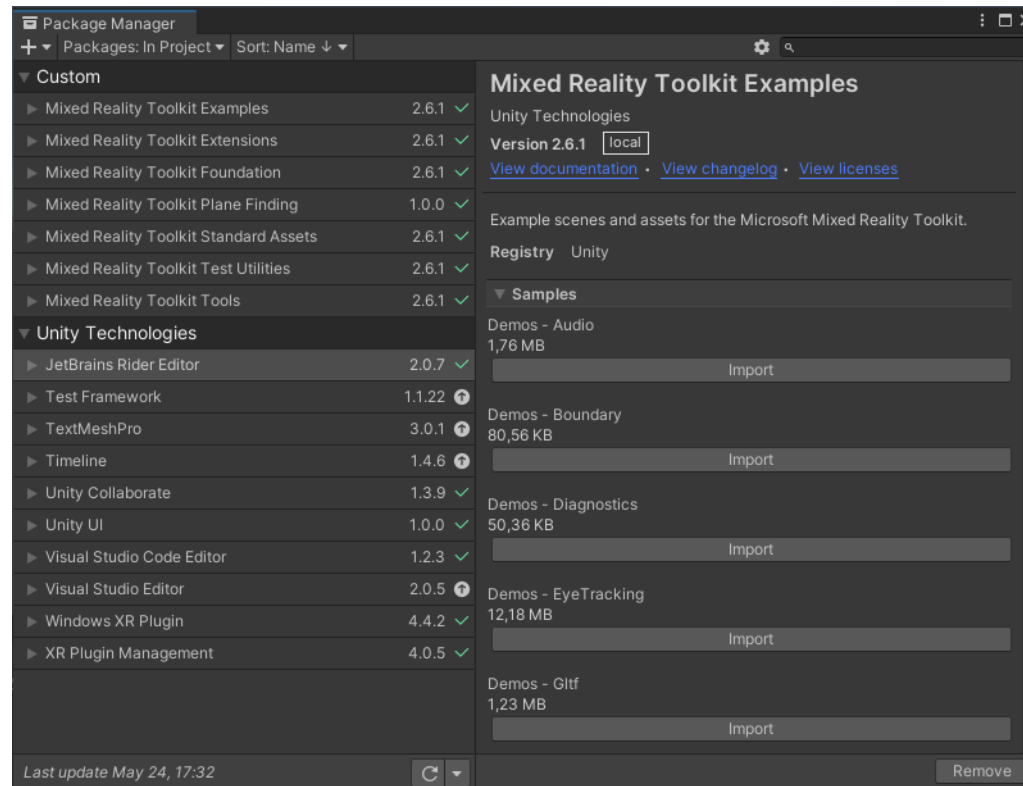
Unity: Input Simulation

- Add a Cube to the Scene
- Add the Object Manipulator script to it (see image, on the left)
- Add the NearInteractionGrabbable script to it (see image, on the right)
- You can now move/rotate/scale the cube with the simulated hands!
 - See [this](#), for all the details on how to use the virtual hands.



MRTK: Import examples

- From the MixedRealityFeatureTool, import the “Mixed Reality Toolkit examples” (from the Mixed Reality Toolkit tab)
- Open the Package Manager (Window / Package Manager)
- Select the Toolkit examples package and import some of the samples on the right side of the window (see image)



MRTK: Input System

- Original images available: <https://docs.microsoft.com/en-us/windows/mixed-reality/mrtoolkit/features/input/overview?view=mrtkunity-2021-01>

Interactable
Interactable is a UX component that listens to input events like focus enter/exit, input down/up and then updates its visual state in response to these events (Interactable.cs)

Cursor
An entity associated with a pointer that gives additional visual cues around pointer interaction.

Focus
A pointer's events will be directed to the object that is focused.

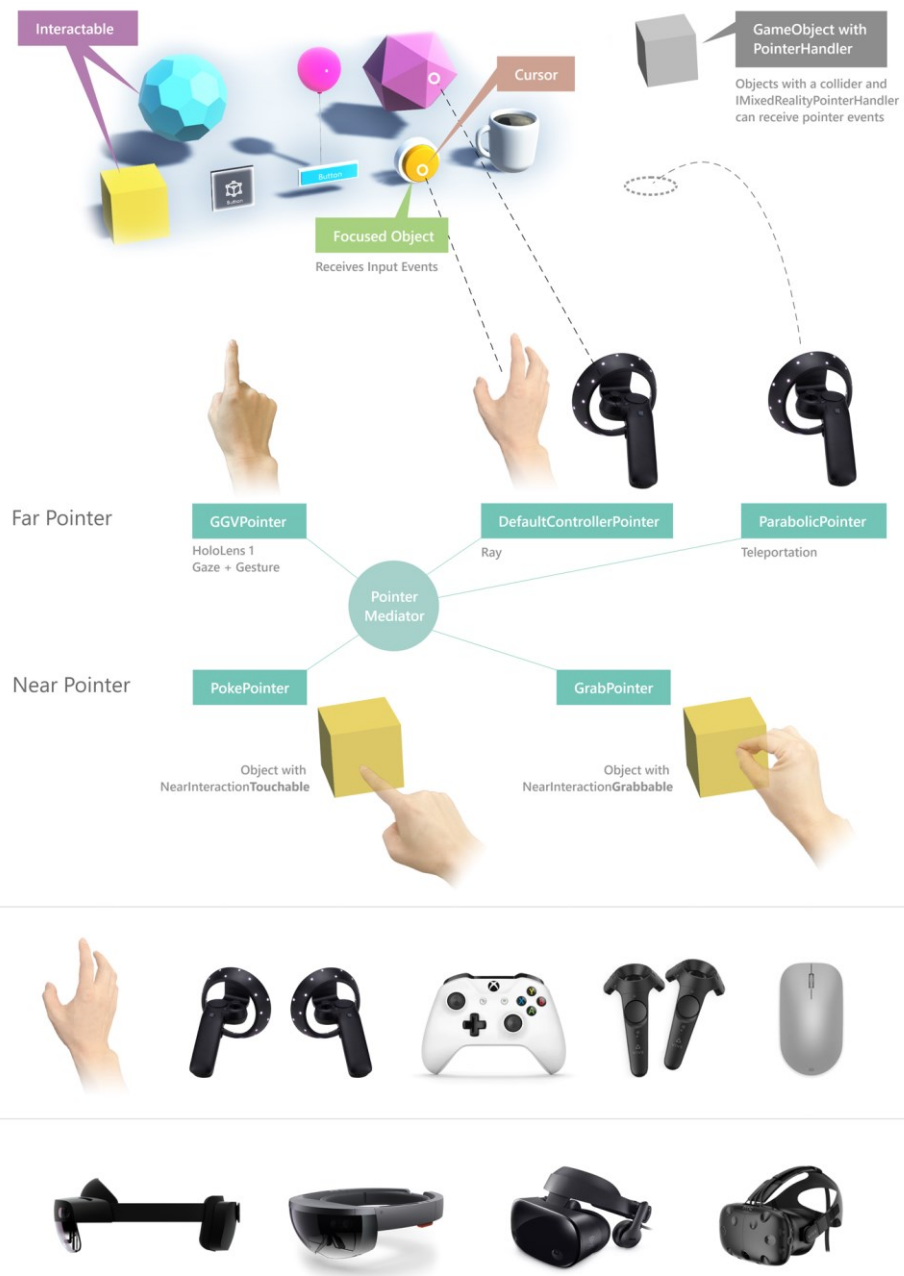
Pointer
Device Manager creates pointers in order to interact with other game objects in specific ways.

Since there could be multiple pointers, **Pointer Mediator** decides which pointers should be active based on each pointer's state. For example, the mediator disables all far interaction pointers on a given controller whenever a near pointer (PokePointer or GrabPointer) is active. You can create custom mediators that do interesting things such as turn off all poke pointers, or turn off everything except the GGVPointer.

Pointers have references to controllers. You can get the pointers for a controller through: `controller.InputSource.Pointers`

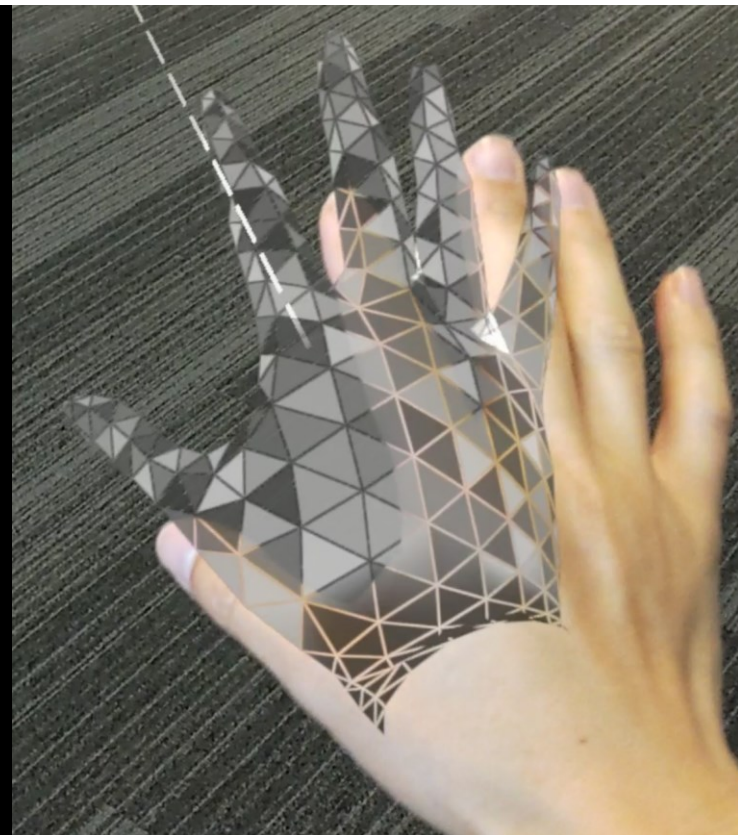
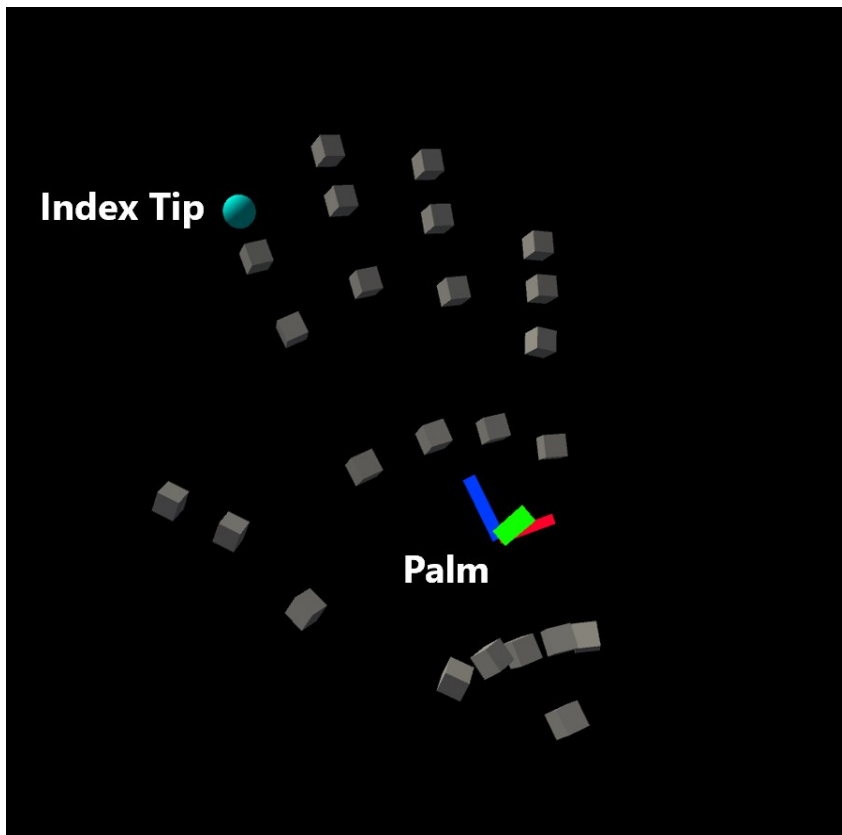
Controller
A representation of a controller. Controllers are spawned by device managers. For example, the WMR device manager will spawn a controller and manage its lifetime when it sees an articulated hand coming into existence.

Device Manager (Data Provider)
Also known as Input Data Providers, these entities are responsible for detecting, creating, and managing the lifetime of their controllers.



Hand Tracking (Hololens 2)

- Thanks to the RGB-D camera, Hololens 2 can track the user's hands.
- Hands visualization is configurable through the MRTK profile.
 - Use it in the editor, do not use in Player (if not really necessary).



Hand Tracking (Hololens 2)

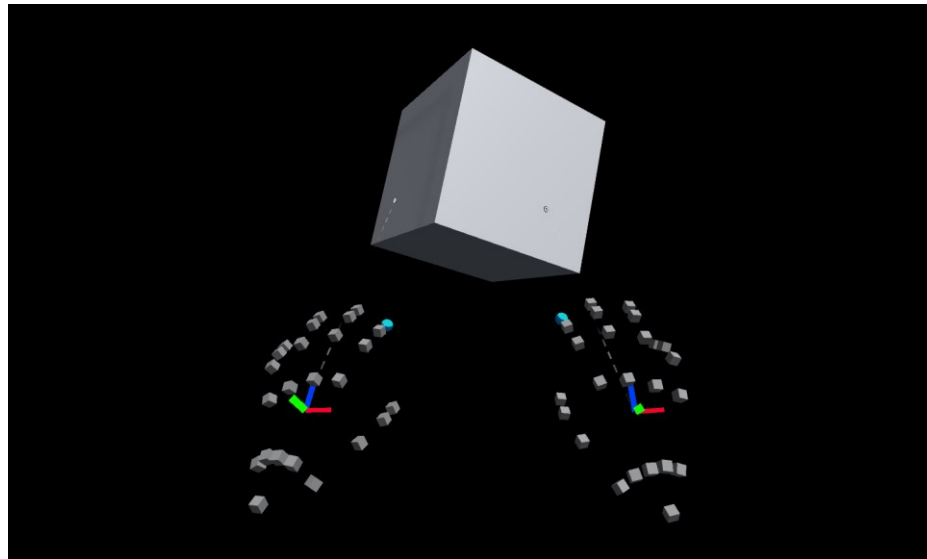
- It's possible to get the position and rotation of each joints (like Kinect APIs).

```
public void OnSourceDetected(SourceStateEventData eventData)
{
    var hand = eventData.Controller as IMixedRealityHand;
    if (hand != null)
    {
        if (hand.TryGetJoint(TrackedHandJoint.IndexTip, out MixedRealityPose
jointPose)
        {
            // ...
        }
    }
}
```

<https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/input/hand-tracking?view=mrtkunity-2021-01>

Hand Tracking (UltraLeap)

- MRTK also supports hand tracking using the Leap Motion device



<https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/supported-devices/leap-motion-mrtk?view=mrtkunity-2021-01>

Eye Tracking (Hololens 2)



<https://www.youtube.com/watch?v=etxh4mqQ9JY>

<https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/input/eye-tracking/eye-tracking-main?view=mrtkunity-2021-01>

Eye Tracking (Hololens 2)

- Get the point of a hologram that the user is looking at:

c#

Copy

```
// Show the object at the hit position of the user's eye gaze ray with the target.  
gameObject.transform.position = CoreServices.InputSystem.EyeGazeProvider.HitPosition;
```

- Showing a visual asset at a fixed distance from where the user is currently looking:

c#

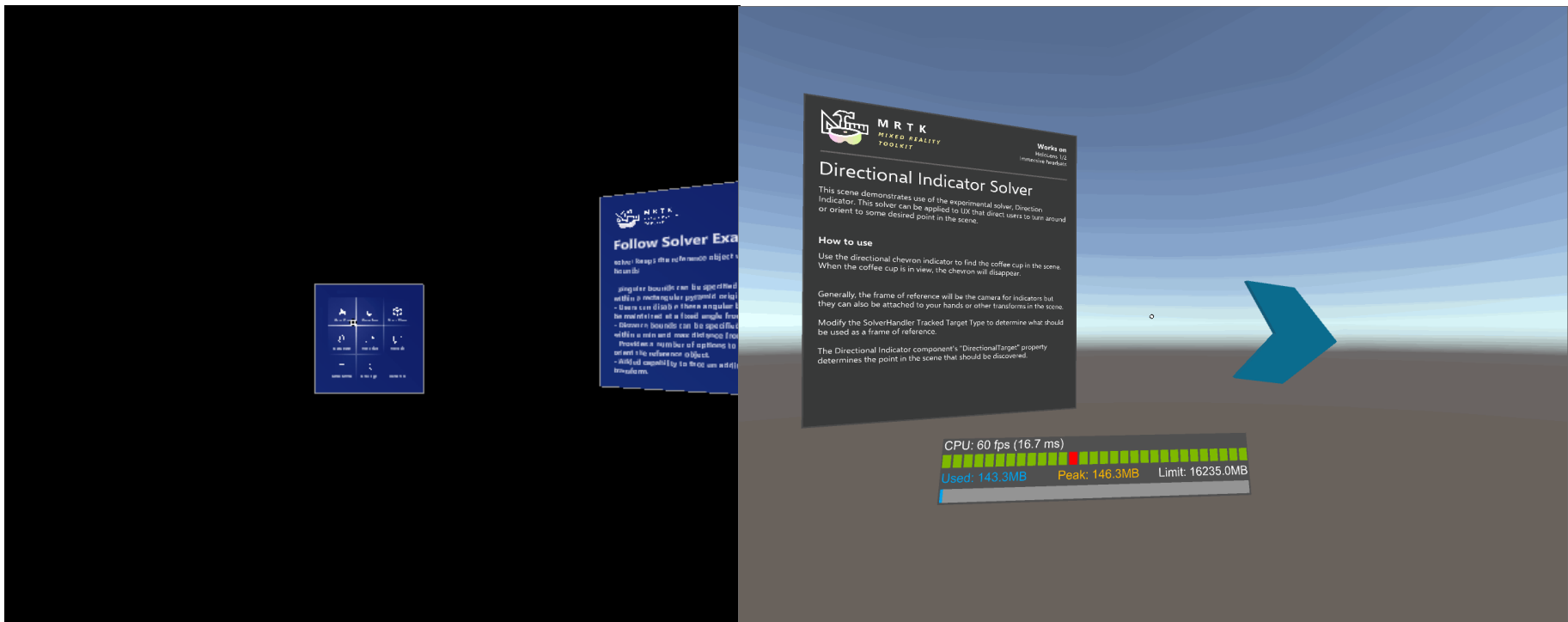
Copy

```
// If no target is hit, show the object at a default distance along the gaze ray.  
gameObject.transform.position =  
CoreServices.InputSystem.EyeGazeProvider.GazeOrigin +  
CoreServices.InputSystem.EyeGazeProvider.GazeDirection.normalized * defaultDistanceInMeters;
```

<https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/input/eye-tracking/eye-tracking-eye-gaze-provider?view=mrtkunity-2021-01>

Solver

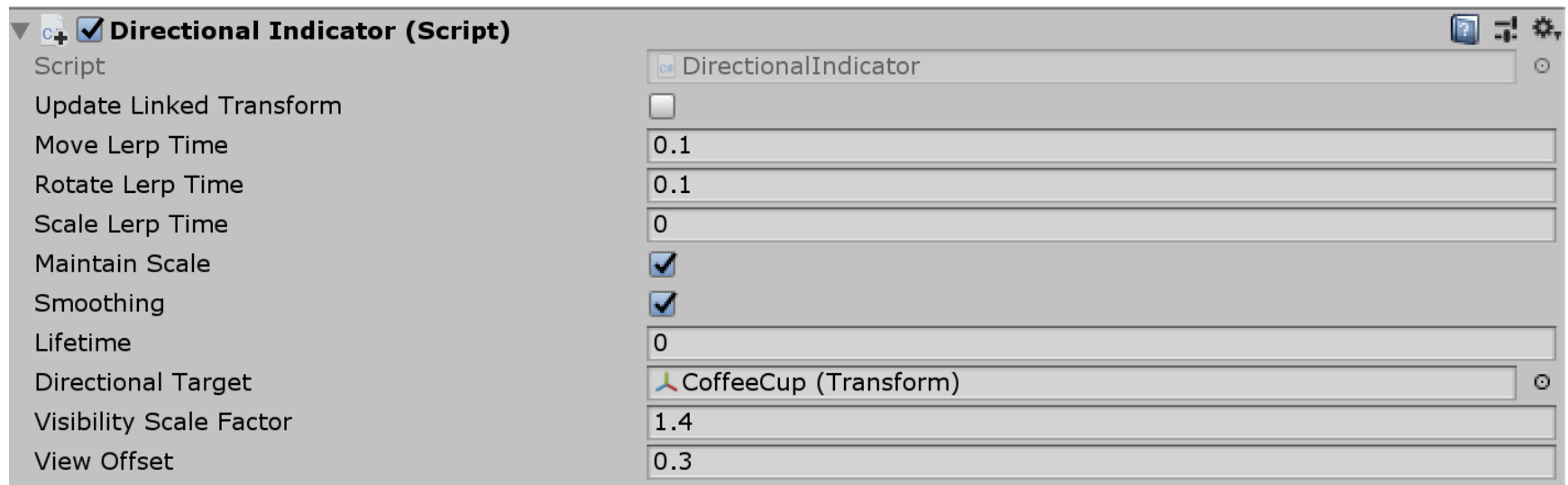
- Solvers are components that facilitate the means of calculating an object's position & orientation according to a predefined algorithm. An example may be placing an object on the surface the user's gaze raycast currently hits.
- Demo available in Unity



<https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/ux-building-blocks/solvers/solver?view=mrtkunity-2021-01>

Solver: Directional Indicator

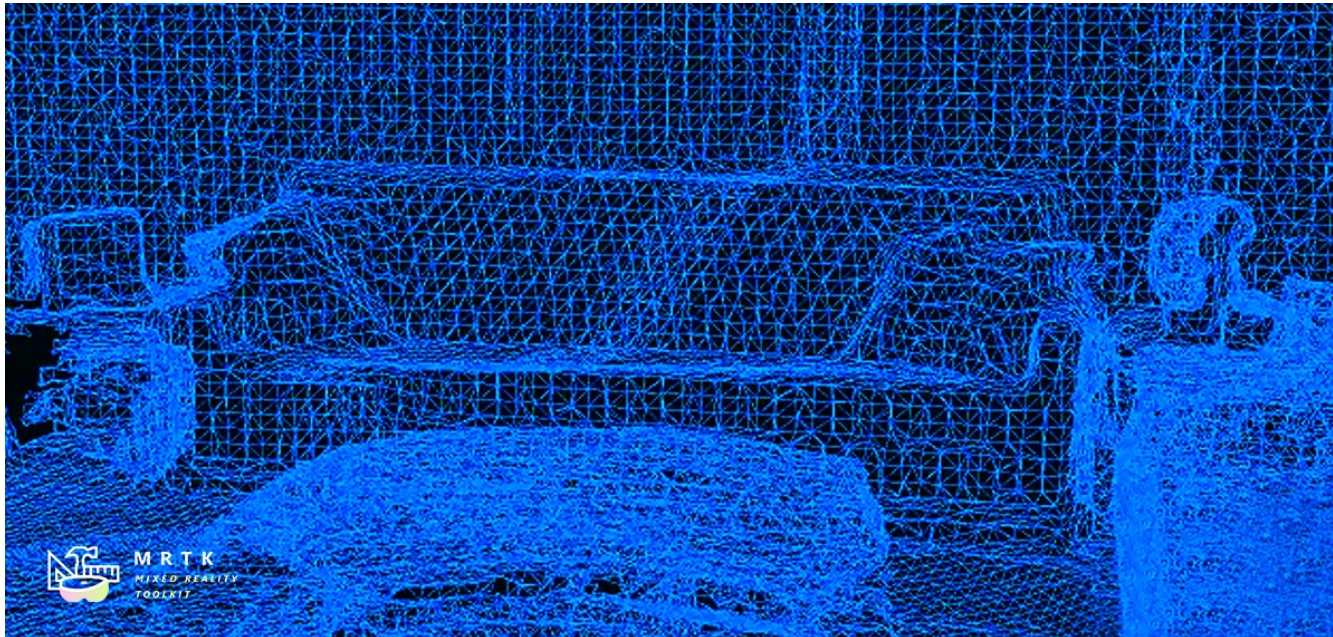
- MRTK also supports hand tracking using the Leap Motion device



<https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/ux-building-blocks/solvers/solver?view=mrtkunity-2021-01>

Spatial Awareness

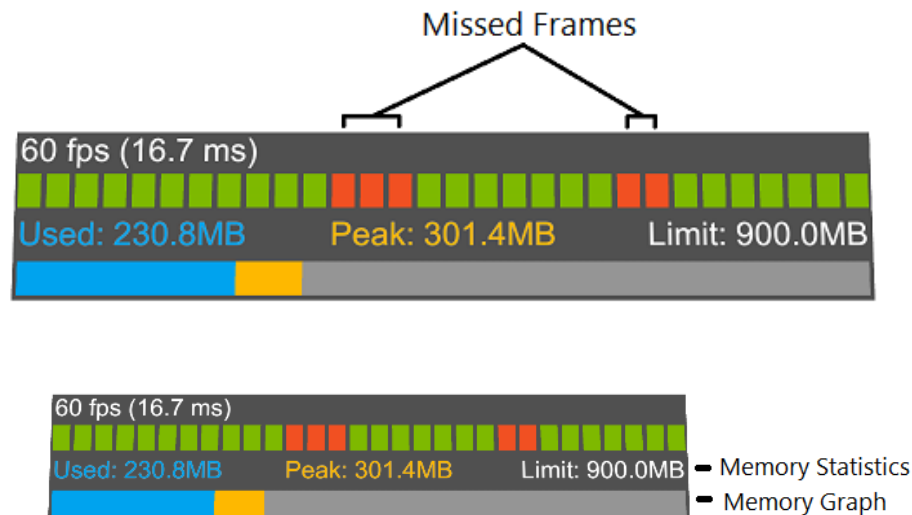
- The Spatial Awareness system provides real-world environmental awareness in mixed reality applications. When introduced on Microsoft HoloLens, Spatial Awareness provided a collection of meshes, representing the geometry of the environment, which allowed for compelling interactions between holograms and the real-world.



<https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/spatial-awareness/spatial-awareness-getting-started?view=mrtkunity-2021-01>

Diagnostic system

- The Mixed Reality Toolkit Diagnostic System provides diagnostic tools that run within the application to enable analysis of application issues.
- The first release of the Diagnostic System contains the Visual Profiler (see images) to allow for analyzing performance issues while using the application.



<https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/diagnostics/diagnostics-system-getting-started?view=mrtkunity-2021-01>

Speech

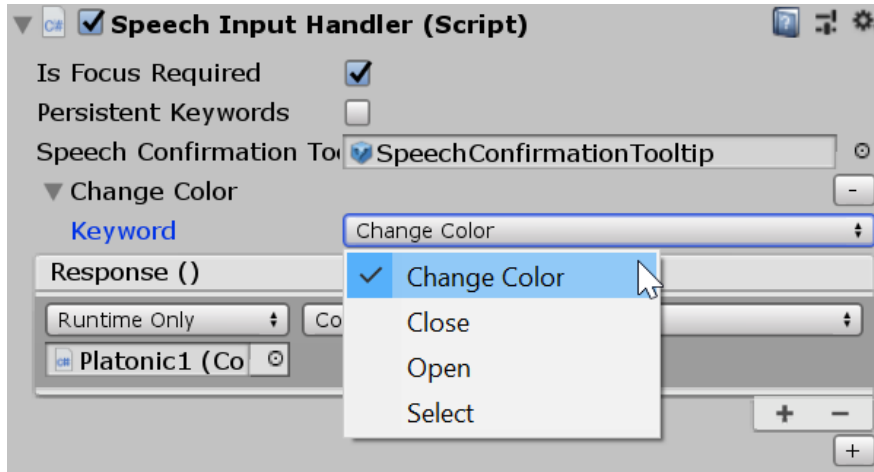
- MRKT allows to define a set of words/short sentences, that will trigger events. In this way, you can interact with your application using only the voice.



<https://docs.microsoft.com/en-us/windows/mixed-reality/mrkt-unity/features/input/speech?view=mrktunity-2021-01>

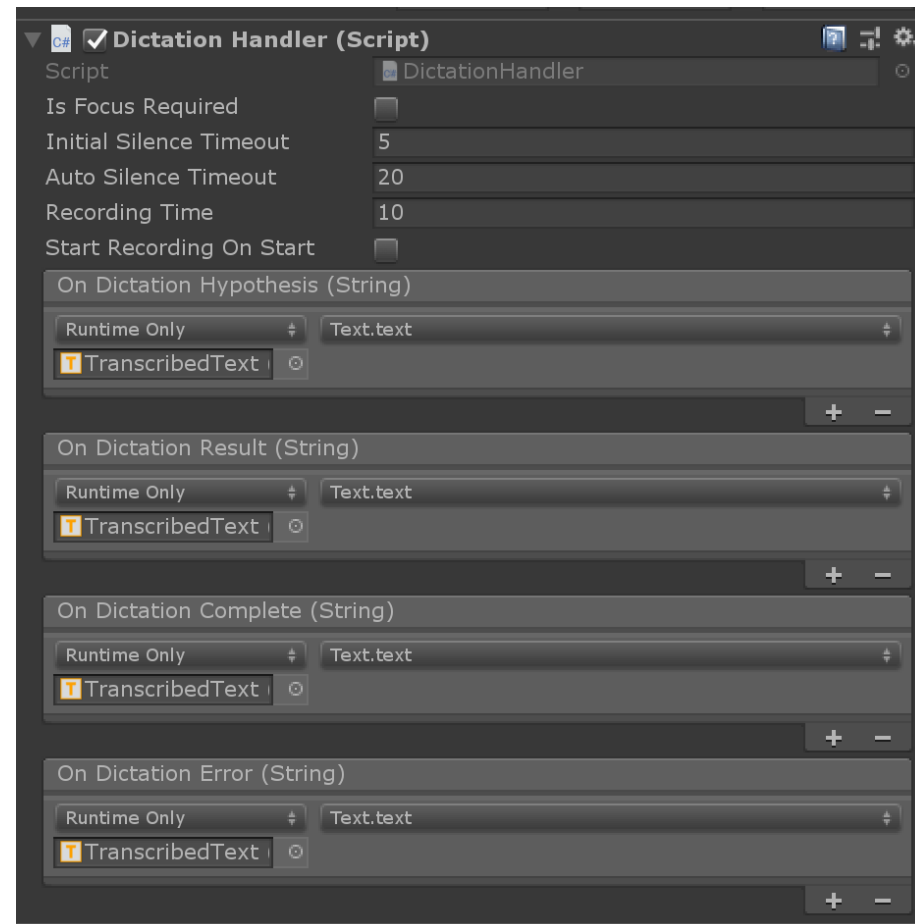
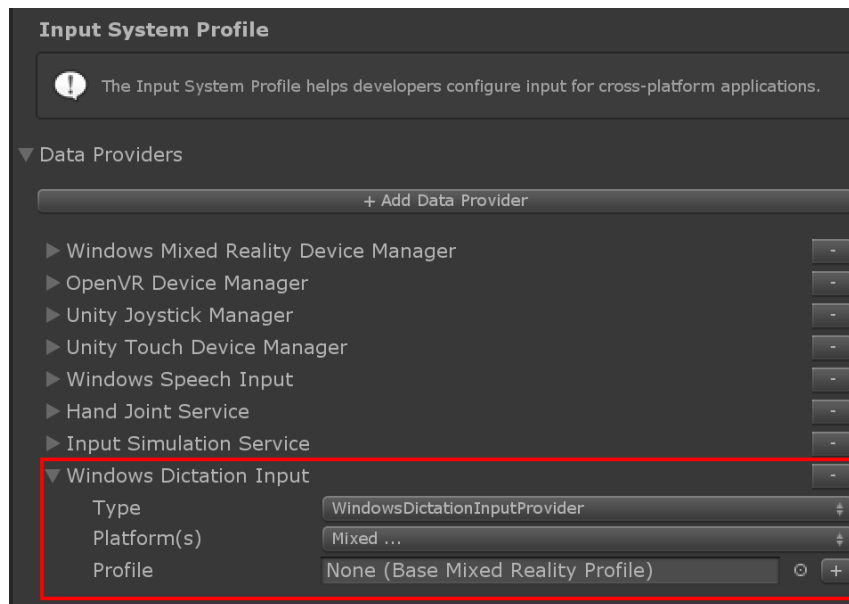
Speech

- In the profile settings/Input/Speech (right image), you can define a set of keyword (which can also be simulated using the keyboard). Then, you can associate to each keyword an even, using the SpeechInputHandler script.



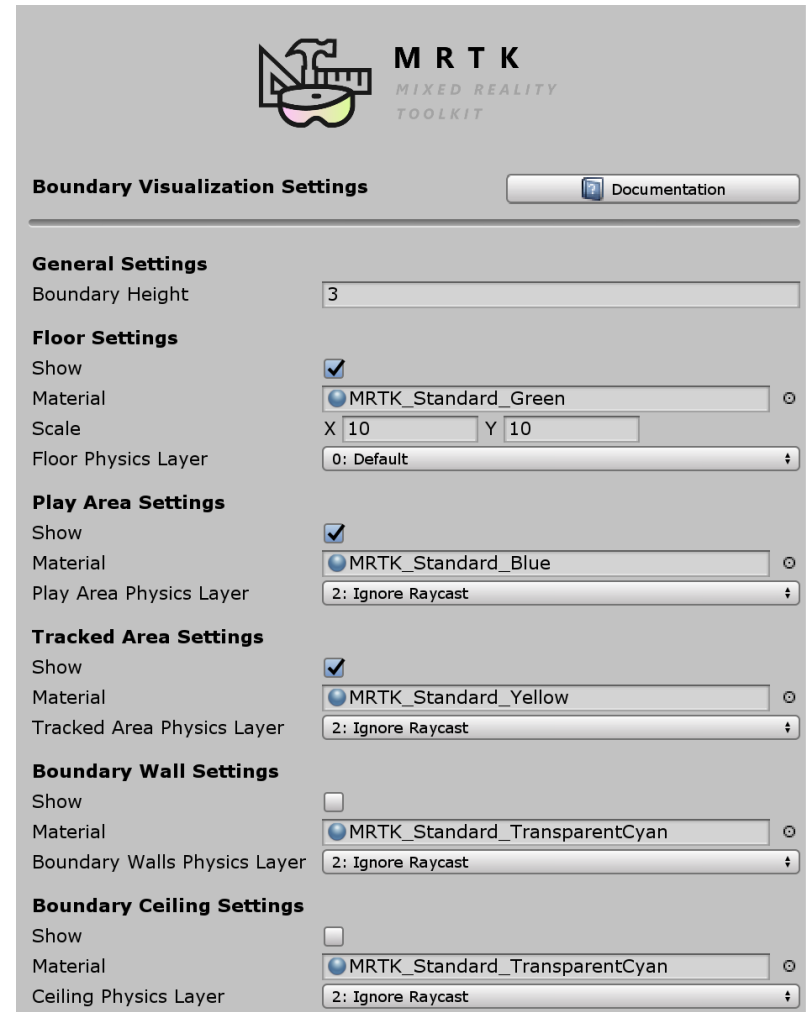
Dictation

- Using the Unity DictationRecognizer, it is also possible to dictate complex sentences.
- In order to do that, you need to add the WindowsDictationInputProvider as Data Provider, in the Input System Profile.
- The DictationHandler scripts, then, helps you to manage the dictation.



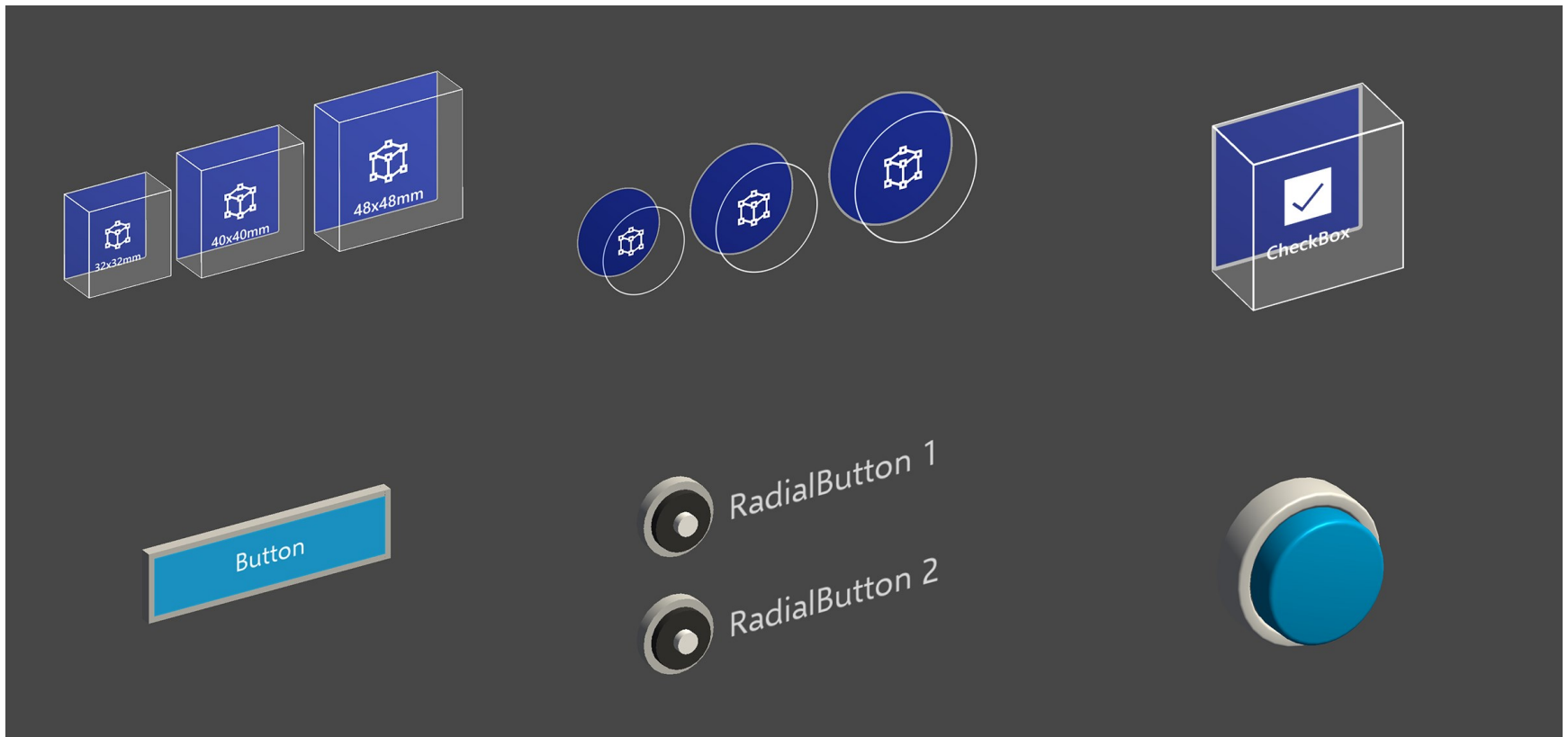
Boundary System

- The Boundary system provides support for visualizing VR boundary components in MR applications. Boundaries define the area in which users can safely move around while wearing a VR headset. Boundaries are an important component of a mixed reality experience to help users avoid unseen obstacles while wearing a VR headset.
- Many VR platforms provide an automatic display, for example a white outline superimposed on the virtual world as the user or their controller nears the boundary. The MRTK's Boundary System extends this feature to enable the display of an outline of the tracked area, a floor plane and other features that can be used to provide additional information to users.



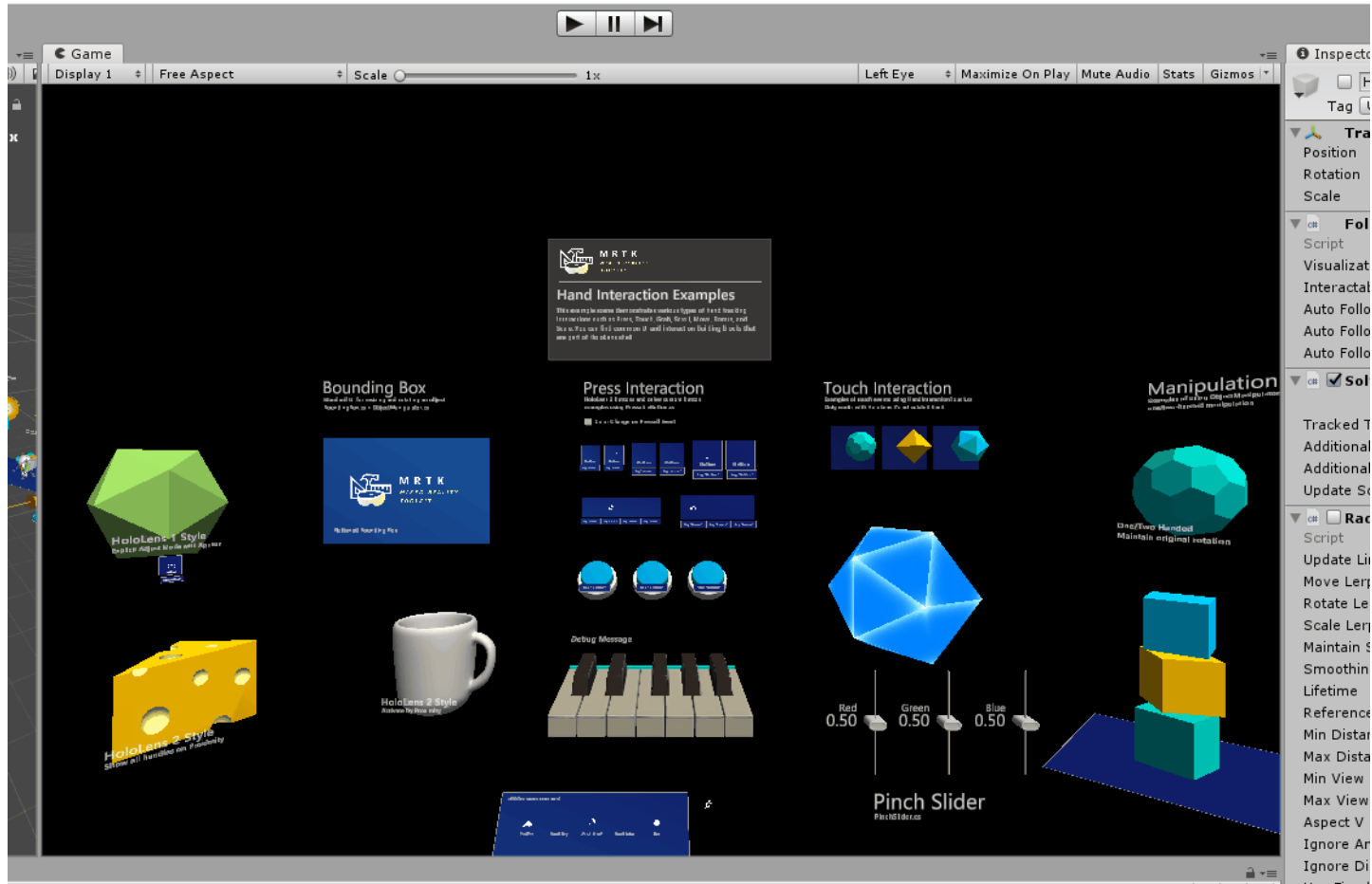
Boundary System

- MRKT is shipped with a set of different buttons (a few example in the image)



<https://docs.microsoft.com/en-us/windows/mixed-reality/mrkt-unity/features/ux-building-blocks/button?view=mrktunity-2021-01>

Demo scene



Open the HandInteractionExamples to try different types of interaction.

Documentation

Full documentation:

<https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/?view=mrtkunity-2021-01>