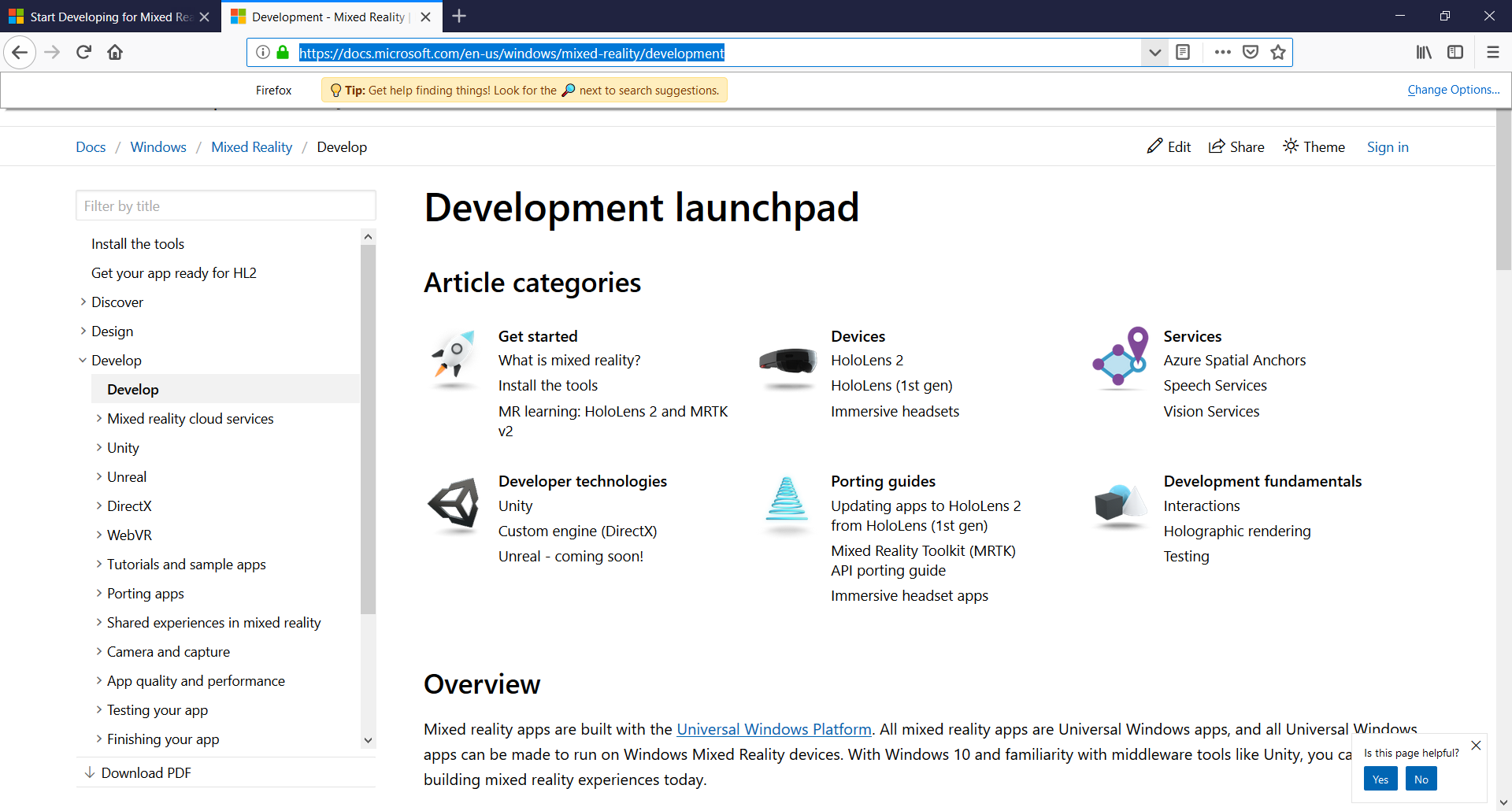
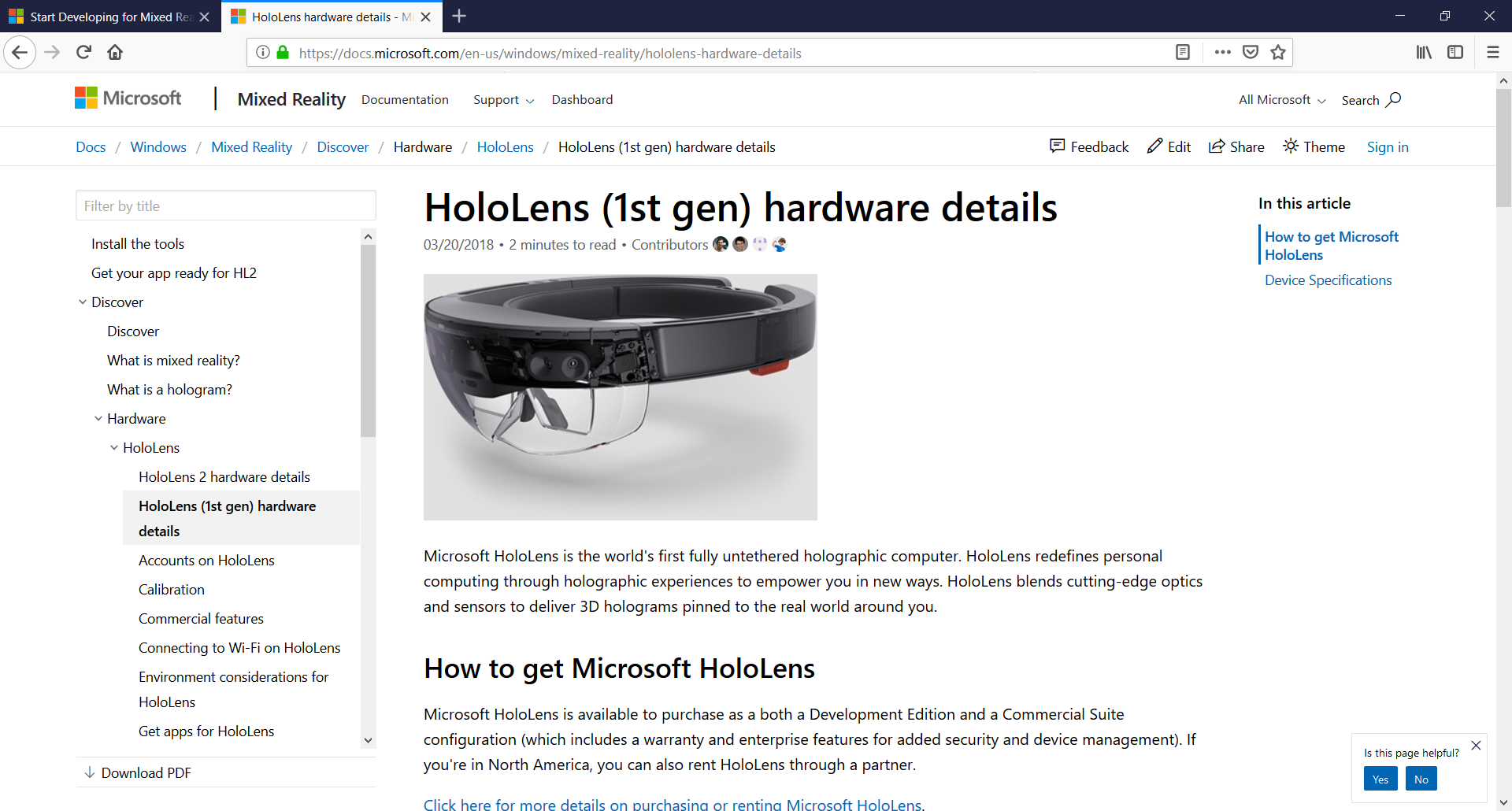
Getting Started with HoloLens

# HoloLens Website Photos





<https://docs.microsoft.com/en-us/windows/mixed-reality/hololens-hardware-details>

Where do we need to calibrate this to? What sort of operating systems work with HoloLens?

We will calibrate in the calibration system built into HoloLens device. The operating systems for the HoloLens depends on who is using the device, but Unity is usable mainly on Windows and Andoid.

No documentation on HoloLens 2 so good thing we have the first one.

## Device Specifications:

* See-through holographic lenses (waveguides)
* 2 HD 16:9 light engines
* Automatic pupillary distance calibration
* Holographic Resolution: 2.3M total light points
* Holographic Density: >2.5k radiants (light points per radian)

## Human Understanding

* Spatial sound
* Gaze tracking
* Gesture input
* Voice support

## Input / Output / Connectivity

* Built-in speakers
* Audio 3.5mm jack
* Volume up/down
* Brightness up/down
* Power button
* Battery status LEDs
* Wi-Fi 802.11ac
* Micro USB 2.0
* Bluetooth 4.1 LE

## Power

* Battery Life
* 2-3 hours of active use
* Up to 2 weeks of standby time
* Fully functional when charging
* Passively cooled (no fans)
* Intel **32 bit** architecture with TPM 2.0 support

**Safety Eyewear**

HoloLens has been tested and found to conform to the basic impact protection requirements of ANSI Z87.1, CSA Z94.3 and EN 166

## Missing components from current box (5/13/19):

Overhead strap

Charging cable

## Basic things to remember for environment:

If things look shaky, redo calibration in the Calibration app.

## TL;DR

This stuff is basically magic. Don’t fret if you can’t fully understand. Images appear in front of you and in your reality. That’s all there is to it.

# What is a hologram?

<https://docs.microsoft.com/en-us/windows/mixed-reality/hologram>

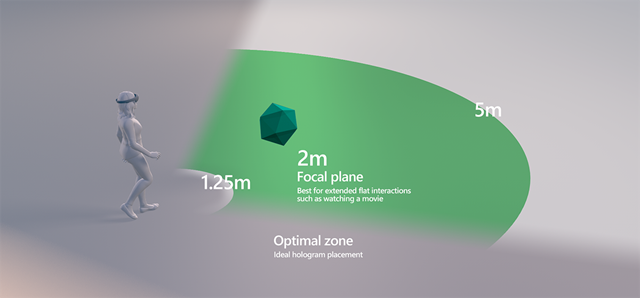
Defined as objects made of light and sound that appear in the virtual world around you.

**“Display locked” vs. “Body-locked”**

* Many developers stay away from display locked as it feels like it is out of the corner of your eye at all times
* Body-locked is more forgivable, and is tethered to a “gaze vector” (see pg. \_\_)

**Good practices:**

* Place hologram at a comfortable distance about an arms-length (3-5 feet) away, and within a 16 ft radius
  + So that the user doesn’t stick their hand through the object and so that it isn’t hard to see
* Provide an amount of drift for elements or consider animating content to one side of the user’s viewing area



Mixed Reality Toolkit Documentation and Learning Capabilities

<https://microsoft.github.io/MixedRealityToolkit-Unity/README.html>

API Computer Visual Data Imaging with Artificial Intelligence

<https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/tutorials/csharptutorial>

<https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/HTKToMRTKPortingGuide.html>

# Using RFID in Mixed Reality (MR)

Mixed and augmented reality and radio frequency identification.

Could use the masterblock to implement into the HoloLens what devices are connected. Then, take the data from the masterblock’s devices and show that to the user. Look At VuMark instead, this could directly link to the NodeRed ui URL

# Implementing MRTK Into Unity

1. https://github.com/Microsoft/MixedRealityToolkit-Unity/releasesDownload the Unity Hub (found on the Microsoft Website
2. Install Unity 2017.3.14f (full version)
   1. This is the latest version that still uses MRTK and doesn’t single pass and take 2 hours to compile the shaders
3. Open Unity with the latest installer
4. Download MixedRealityToolKit.Foundation.Download off of the github website. This will be the installer that is necessary for you to use all of the resources in the toolkit.
5. Move the toolkit to the ~/projectFolder/Assets/
6. Install the TextMesh package by selecting Window>TextMeshPro>Import TMP Essential Resources
7. Click GameObject>UI>TextMeshPro – Text. This will open up a prompt that allows you to open up another prompt. This will allow you to put text in front of a scene for logo purposes

This is how you create your own MRTK that can be more user-friendly

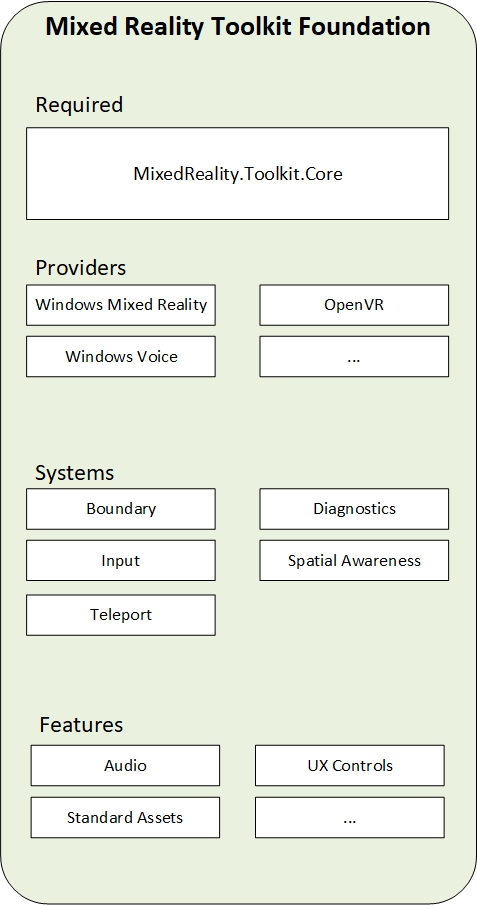
<https://docs.microsoft.com/en-us/windows/mixed-reality/mrlearning-base-ch2>

<https://www.youtube.com/watch?v=cBAOALaHys4>

## Near Interaction Touchable

<https://docs.microsoft.com/en-us/windows/mixed-reality/mrlearning-base-ch4>

This will give you instructions on how to create a 3D Object and interact with it. You will use this for every single object you Create that you want to be interactable.



## Types of Mixed Reality UX Objects:

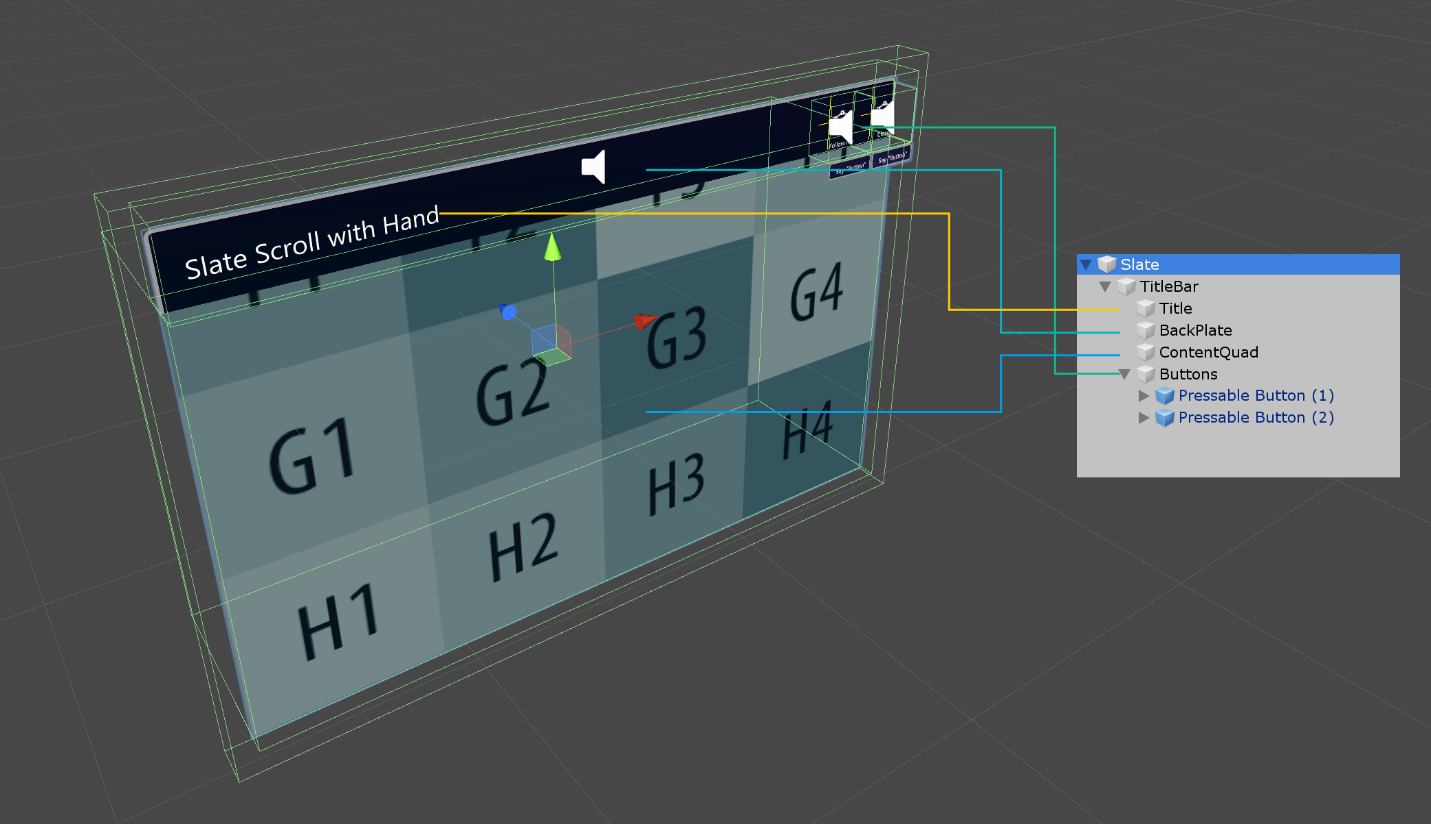
### Bounding Box

### Slate

<https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_Slate.html>

A slate control is composed of the following elements:.

* TitleBar: The entire title bar on top of the slate.
* Title: The title area on the left side of the title bar.
* Buttons: The button area on the right side of the title bar.
* BackPlate: The back side of the slate.
* ContentQuad: Content is assigned as material. The example uses a sample material 'PanContent'.

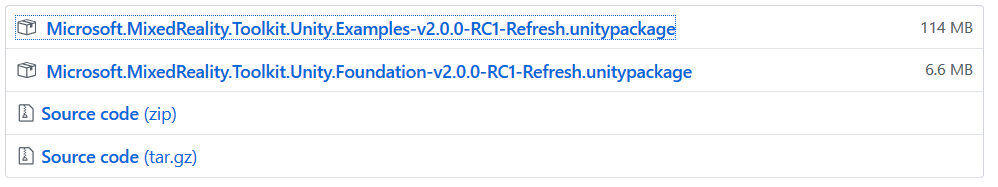


## TL;DR

You can place objects in your Unity Project and interact with them regardless of where they come from. Add components like NearInteractionTouchable for a more user experience

## How To Set Up MRTK on Unity

1. Download the MixedRealityToolKit (MRTK) from the Microsoft website

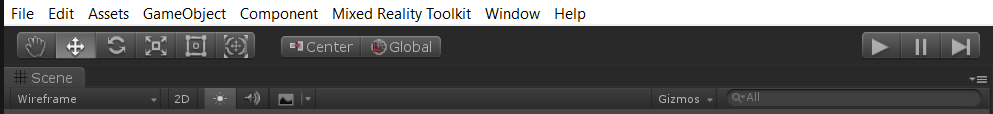


2. Import the MRTK .unitypackages into Unity by right-click>Import Package>Custom Package. The prompt will come up, select your .unitypackage and open it.

If you want, after downloading the .unitypackage, immediately open it through your designated browser (tested through Firefox).

2a. If you get an “Opening file failed … TempFile… etc.” error, restart your computer and reinstall the .unitypackage

3. Once installed you should get another tab added to your File … Edit … Assets… etc. like this:



This will add MRTK to your arsenal and you can start adding cool new things like spatial awareness that allows your HoloLens to become usable through Unity.

4. Add and Configure the scene with Mixed Reality Toolkit > Add to Scene and Configure

5. Import the TMP Package (TextMeshPro)

6. Create a cube with right click in Scene > Create > 3D Object > Cube

7. To add a World Anchor, simply pick your designated position, then use Component > AR > World Anchor

8. Replace your camera with MRTK add to scene and configure

9. Add the Vuforia Behavior to the camera you have created

Set up into HoloLens with MixedReality Camera in the area as well. Having them separate could help it show up better.

## TL;DR

Install all modules, drag and drop, and setup HoloLens correctly (look for PerceptionRemotingPlugin.dll error)

# Creating the Environment

## Buttons

I have just created a new environment for Buttons that allows me to open a new bounding box with Unity and then turn off the scaling so that the abilities will improve.

How did I do this?

First I added a new button by copy/pasting the “cube” button from the BTL Button, and changing the prefab to a more suitable and visually appealing button. Then I added these components.

Somehow able to move the button to a new area, but the button does not persist in that area.

Look at YouTube video for some more information

Create a better version of the grabbing mechanism and maybe allow for a way for Moving the Items in the world Space

Added a Close All Mechanism

<https://tommytdotnet.wordpress.com/2019/04/14/mrtk-v2-rc1-tutorial-02-create-an-interactive-cube-with-highlighting/>

Drag and Drop handler?

Vuforia Handler instead

IUnityRenderPipeline not Recognized?:

Change the if statements to 2018\_4 instead of 2018\_3, this will fix the errors in RuntimeOpenSourcePipeline

# Using Vuforia

## Vuforia Import

Vuforia creates an industrial and CAD environment that lets you project your predetermined images onto the environment around you. In my opinion it is the most important part of the implementation of Mixed and Augmented reality.

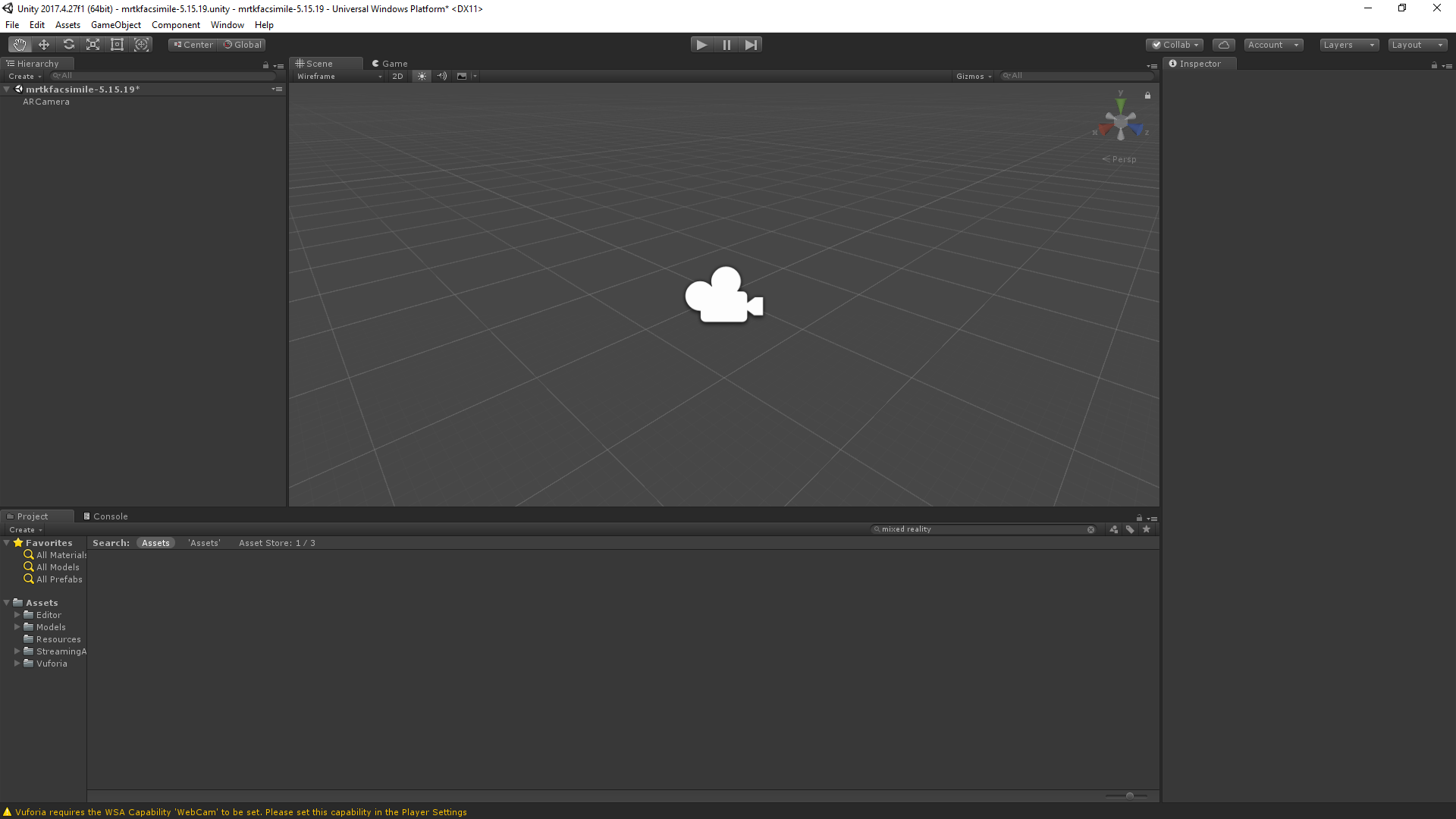
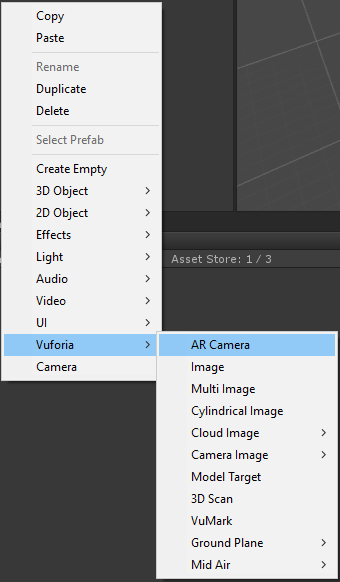
Important Performance Techniques:

<https://docs.microsoft.com/en-us/windows/mixed-reality/performance-recommendations-for-unity>

If you get a *Compiling Shaders* loading screen with more than 30 files (got up to 6500 for me once), this is a bug and an error and should be cancelled immediately. This process will result in a lot of lost time during simple testing, and cancelling will not affect project.

Vuforia 8.1.1 will be your basis for augmented reality. To get started: begin with opening your Unity 2018.4.xxfx project window. This will be your canvas for the objects you create in your program that will project onto the environment around you.

## Starting Vuforia



1. Start with right clicking in the hierarchy area and selecting the AR camera. This will create what the environment will look at.

2. Create a VuMark object and place it at XYZ (0,0,0)

3. Go to your Target Manager on the developer webpage at <https://developer.vuforia.com/targetmanager/project/targets?projectId=3f8d685518504d49bb98646ab5cd449b&av=false>

4. Turn on the HoloLens and log into it using the IP given by the HL. Type this in as https://<HL-IP>/devicesecurity.htm and continue without recognition or security.

5. Take a photo using the HoloLens and add that via the HTML image file into Vuforia. This will take the image and scan it as a VuMark image and allow the image to show the unit scale

6. Create the image in Unity and use it as a target image file and that will allow you to use that to find the product IDs

**A few things to watch out for**

* Even though Visual Studio comes with its own C# compiler, and you can use it to check if you have errors in your c# **scripts**  
  , Unity still uses its own C# compiler to compile your **scripts**. Using the Visual Studio compiler is still quite useful, because it means you don’t have to switch to Unity all the time to check if you have any errors or not.
* Visual Studio’s C# compiler has some more features than Unity’s C# compiler currently supports. This means that some code (especially newer c# features) will not throw an error in Visual Studio but will in Unity.
* Unity automatically creates and maintains a Visual Studio .sln and .csproj file. Whenever somebody adds/renames/moves/deletes a file from within Unity, Unity regenerates the .sln and .csproj files. You can add files to your solution from Visual Studio as well. Unity will then import those new files, and the next time Unity creates the project files again, it will create them with this new file included.

<https://library.vuforia.com/content/vuforia-library/en/articles/Training/Developing-Vuforia-Apps-for-HoloLens.html>

**Here is what occurs...**

1. Vuforia Engine’s target Tracker recognizes the target
2. Target tracking is then initialized
3. The position and rotation of the target are analyzed to provide a robust pose estimate for HoloLens to use
4. Vuforia Engine transforms the target's pose into the HoloLens spatial mapping coordinate space
5. HoloLens takes over tracking if the target is no longer in view.

<https://library.vuforia.com/content/vuforia-library/en/articles/Solution/Working-with-the-HoloLens-sample-in-Unity.html>

I installed Unity 2018.4 because that is the last one compatible with Vuforia 8.1.1. Always look at which Vuforia pairs with which Unity because it will affect this.

If you get a **“namespace name YADAYADA could not be found/does not exist**, then you need to check your version of Vuforia, because Unity changes the names in their packages and Vuforia can’t keep up with it. That’s the cost of being a third-party engine installer I guess.

Using Image Targets in Vuforia

<https://www.youtube.com/watch?v=Wqbg9mB84V4>

1. Create a new Scene in your Project

2. Add an AR Camera under the Assets>Vuforia>AR Camera

3. Create an Image Target with Assets>Vuforia>Image Target

4. Create the image and change the ‘Predefined’ Type to ‘User Defined’

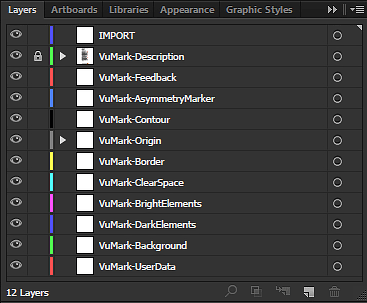
5. Select your database of images that you will have for your project (mine will start with the Masterblock). This will be the Canvas (that’s why it’s in the Hierarchy) that you will display other images on.

6. We need to create objects in this environment and allow our box to be detected, so we will add the DefaultTrackable script to the object’s components to implement it into a trackable interface.

7. VuMark is your very own RFID for your HoloLens. It allows you to scan the object you’re using and it will identify what that object is

## What does a VuMark Consist of?

|  |  |
| --- | --- |
| **IMPORT** | It is not required that you use this layer, but we have found it helpful to have a work space where you can paste your VuMark design. From this layer, you can begin separating the various parts of your VuMark design by moving them to the corresponding layers. |
| **VuMark-Description** | INFORMATIONAL, LOCKED DO NOT ATTEMPT TO MODIFY THIS LAYER This layer contains the details of your VuMark such as the Name, Type, Length and number of required Elements. The Exporter script uses the information contained in this layer and that is why it is locked by default to prevent accidental deletion. |
| **VuMark-Feedback** | This layer is empty at first, but after you run the Verify script, it will be populated with feedback presented on top of your VuMark design to help you verify that it meets the design requirements. |
| **VuMark-AsymmetryMarker** | CAN BE EMPTY When creating a rotational symmetric Contour, this layer can be used to add Asymetry Markers that prevent from accidental wrong ID readout. See section TODO on more details. |
| **VuMark-Contour** | This layer shall only contain one closed path with 4-20 straight edges. Once you have placed your Border and Clear Space parts into their corresponding layers, you must use the Pen tool in Illustrator to draw a closed path along the invisible line that is formed where these two parts touch. This path must be placed in the VuMark-Contour layer. |
| **VuMark-Origin** | This layer contains a circle shaped object that represents a point that will be considered the origin on your VuMark design. Augmentations that will appear registered to the VuMark relative to this point. Move them in your design to match required origin. Circle can be scaled. |
| **VuMark-Border** | This layer should only contain the object representing your VuMark s Border. |
| **VuMark-ClearSpace** | This layer should only contain the object representing your VuMark s Clear Space. |
| **VuMark-BrightElements** | This layer will contain the Elements in their Bright state. The number of objects placed in this layer must match the number of required Elements exactly. Make sure Elements are actually brighter than Dark state. |
| **VuMark-DarkElements** | This layer will contain the Elements in their Dark state. The number of objects placed in this layer must match the number of required Elements exactly. Make sure Elements are actually darker than Bright state. |
| **VuMark-Background** | CAN BE EMPTY Contains background design. Use rich background if you intend to track VuMark in Vuforia Engine. |
| **VuMark-UserData** | CAN BE EMPTY This layer can contain graphical elements for various purposes, e.g. UI overlays, printing marks, metadata, etc. The graphic placed in this layer will be exported as part of the VuMark template SVG file and provided with VuMarkTemplate::getUserData() method inside the SDK when the VuMark is loaded. |



<https://library.vuforia.com/content/vuforia-library/en/articles/Solution/Designing-a-VuMark-in-Adobe-Illustrator.html>

<https://library.stage.vuforia.com/articles/Solution/Working-with-VuMarks.html>

# Spatial Anchors

Best way: group cluster anchors at the center of the counter so that the anchors are all representative of the MasterBlock (being the countertop’s sensors are all stationary objects therefore you can display the data in the same positions relative to the MasterBlock).

Can cluster by making the Objects a “child” of the nodes

<https://docs.microsoft.com/en-us/windows/mixed-reality/spatial-anchors>

# Canvas

## Creating Canvas

Using a canvas, you can overlay any of your UI into Unity.

To Do This:

1. Game Object > UI > Canvas

2. Inspector > Render Mode ^ World Space

3. Scale it using this formula: i.e. (real world scale) / (unity scale) 0.52 meters / 800 pixels

4. Click the Canvas Utility Button (it will prompt you)

5. Create a 3D Object and drag it into the Hierarchy as a “child” object of the Canvas.

Looks like this:



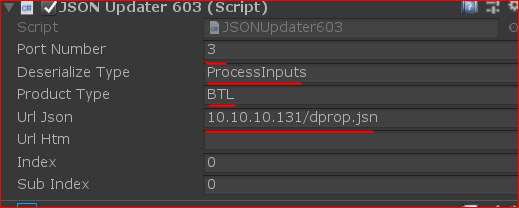
Now anything under the canvas will have less decimal points and calculations!

Note:

You can do division and multiplication by just dividing your values with a “(slash)/[insert divisor]”

Here’s how you use JSON Updater

i.e.



Switch to 2018.4. Otherwise, your inputs in the HoloLens will no longer work.

Orientation for HoloLens:

# NodeRED With HoloLens

Could use HoloJS to develop then easier to implement into NodeRED (nope)

Using a WebSocket (Check development)

Need to know what HTTP is before using something like WebSocket:

Here are some websites that you can use:

<https://developer.mozilla.org/en-US/docs/Glossary/HTTP>

TCP/IP Ports

<http://www.steves-internet-guide.com/tcpip-ports-sockets/>

Use Exec feature with arguments and outputs in order to access the Balluff IO Link node

Started learning the abilities of animation in order to get pop ups to work organically

Use the IP from MasterBlock and access the network with everything except the last 123.456.789.xxx, then type in the MasterBlock’s IP into the search engine and access the MasterBlock’s homepage.

We can now take the information from the html under the [IP]/dprop.htm and use a Windows Forms Application to access the sensor information needed.

