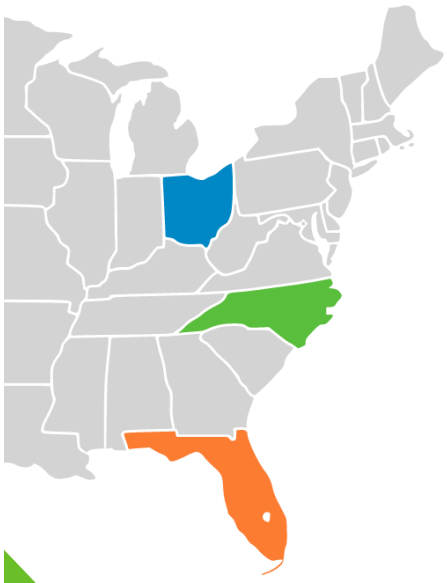
Good morning everyone –
Thanks for being here to take a look
At different digital realities,
The HoloLens, and Unity.
My name is Jeff McKenzie, and before we get started
I'd like to tell you a little bit about myself –
I like puppies, long walks on the beach…
And I'm also a Principal Consultant
At Cardinal Solutions – a little bit about Cardinal...



OHIO
Cincinnati
Columbus

NORTH CAROLINA
Charlotte
Raleigh

FLORIDA
Tampa

WHO WE ARE

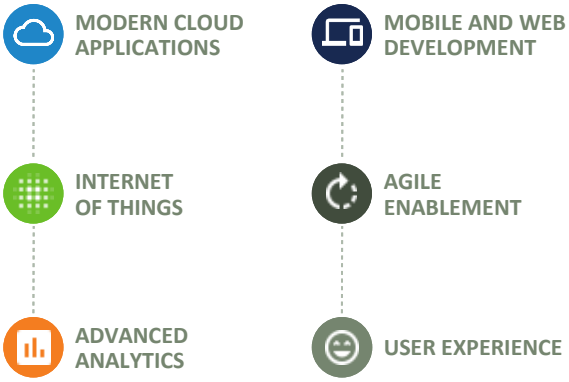
Cardinal has nearly 500 full-time consultants and has been in business for 20 years with over $60 million in revenue.

A little bit about cardinal – we are an IT services firm with
About 500 full-time consultants,
across the Midwest and east coast
We've got an office here in Columbus,
downtown in the arena district,
With about 100 people, with a mix of
Inhouse project work and
On-site client work.

CORE CAPABILITIES

INNOVATIVE TECHNOLOGY SOLUTIONS

At Cardinal, we leverage our extensive experience to imagine, develop, manage and deliver technology solutions that enable your business.



As far as what Cardinal does, we pretty much cover all areas
Of fundamental and cutting edge technology.
We've had significant IoT projects,
Including several AR and HoloLens apps,
And We do quite a bit of cloud work,

@jeffreymckenzie

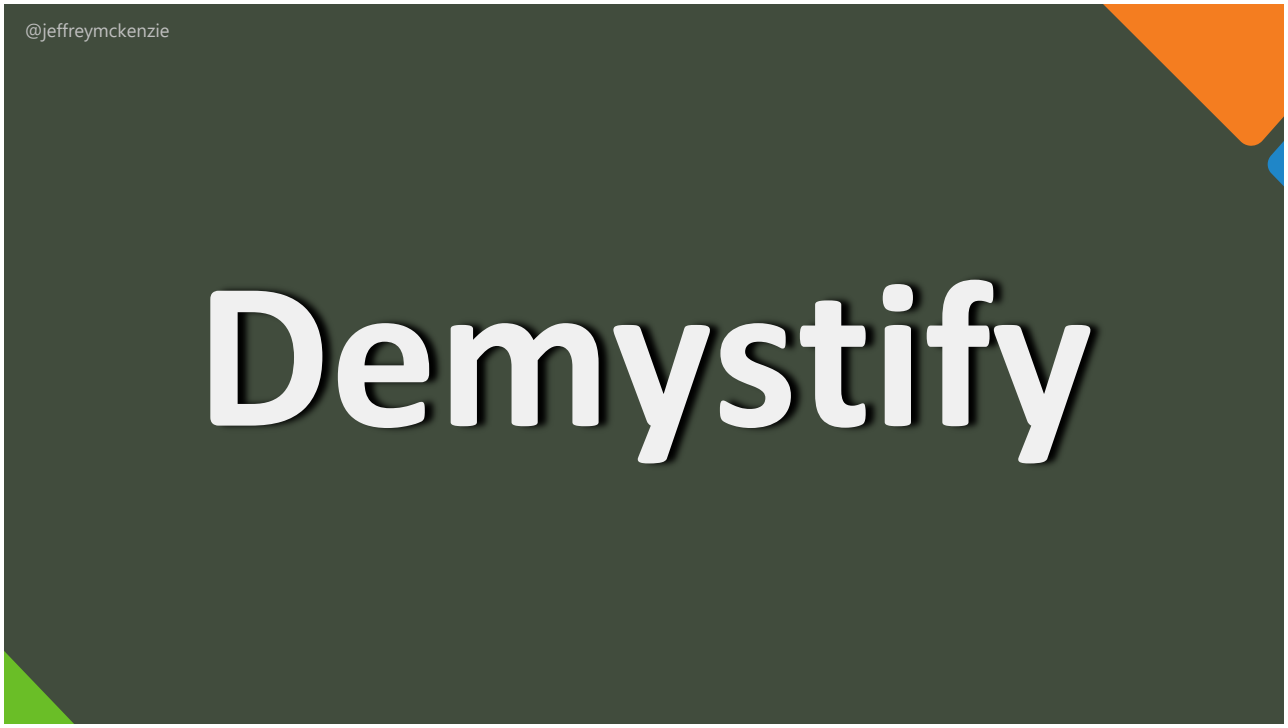
So actually my experience with VR and AR this started this summer, pretty soon after I joined Cardinal.
I was on bench waiting for my first assignment.
And my practice manager comes in and asks me,
Hey do you know anything about HoloLens?
I'm like, well I've heard about it…
And he says, we may have to give a demo of it
in a couple weeks, and I've got this app I need loaded onto it.
On one hand, I'm like, wow that's pretty cool
And on the other hand I'm like TWO WEEKS, you gotta be kidding me.
=========
https://wallpaperscraft.com/download/the_office_tv_series_steve_carell_boss_96802/3840x2160
https://wallpaperscraft.com/image/the_office_tv_series_steve_carell_boss_96802_3840x2160.jpg

@jeffreymckenzie

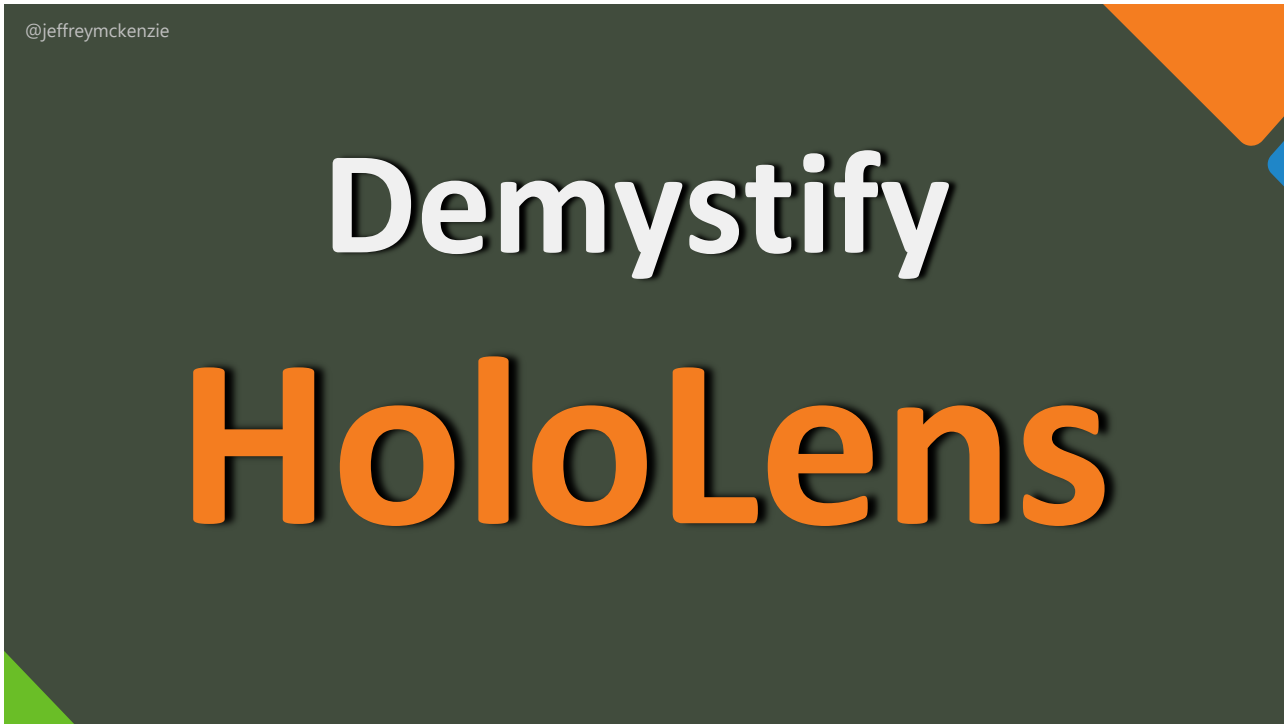
At that point in time, everything regarding
Virtual Reality, Augmented Reality, and Mixed Reality --
Not to mention the Hololens –
Was just one big mystery.
It's cool that these devices can totally
Change your experience, but what do these different
Devices do, and how do they work?

=========

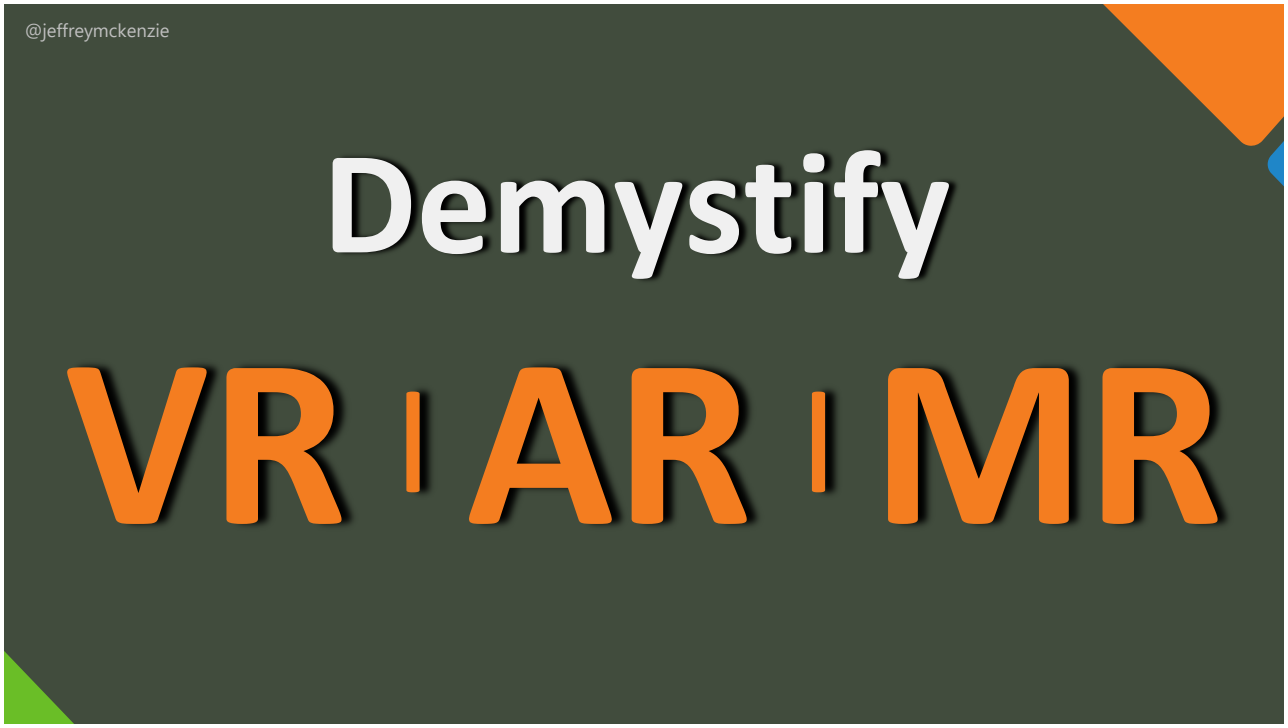
By Tshrinivasan (Own work) [CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons https://commons.wikimedia.org/wiki/File%3AEntrance_of_mystery_spot%2CUSA.jpg



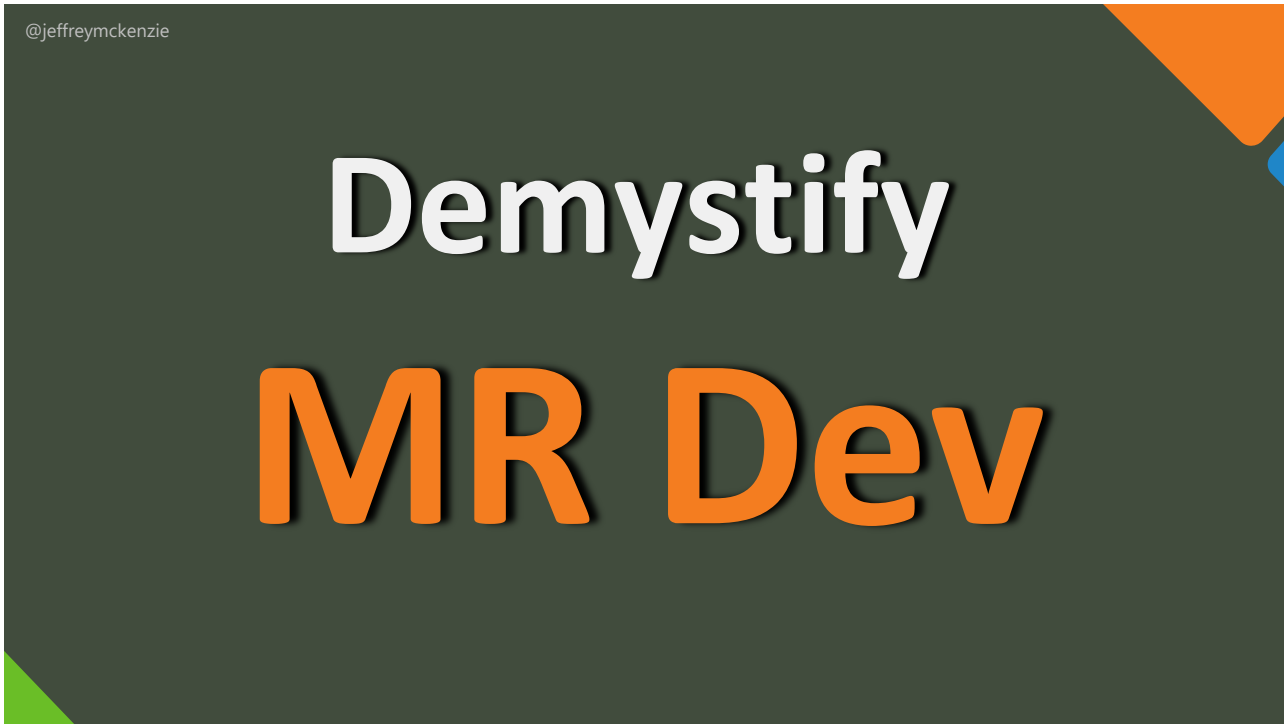
So my goal for today is to help try, or start to Demystify all of that, by taking a closer look First, at the HoloLens itself…



Second, unraveling the different kinds
Of computer-generated realities…



From Virtual Reality,
Augmented Reality,
And Mixed Reality..
And Third….



The process of creating your own
Mixed reality application
For the HoloLens.
So let’s get started with the first --

The Microsoft HoloLens, which I have right here.
[How many of you have seen this or are familiar with what it does? Used it?]
And what this device does is allow the wearer to view
3D projections,
Within the context of the world around them.
I'll show you that in a minute here, but
Here's an example of what that kind of looks like….

---

https://www.pcworld.com/article/2917379/my-hololens-acid-trip-was-fascinating-but-left-a-funny-taste-in-my-mouth.html
https://images.techhive.com/images/article/2015/05/hololen_hardware-100582804-orig.jpg

@jeffreymckenzie

Who doesn't love Pokemon, right.
So here we have a 3d object sitting on a real-world table.

The term Microsoft uses to describe this experience
Is Mixed Reality, because there's a mixture
Of 3D objects with our surrounding environment, or reality.

So let's take a look.

---

https://www.engadget.com/2016/07/15/pokemon-go-gets-in-your-face-with-an-unofficial-hololens-demo/
https://o.aolcdn.com/images/dims?quality=100&image_uri=http%3A%2F%2Fo.aolcdn.com%2Fhss%2Fstorage%2Fmidas%2F93edcc5da453de1364c53bc86139c354%2F204087798%2Fpokemon_hololens-ed.jpg&client=cbc79c14efcebee57402&signature=802a82585468a2481eaff2f6c5871b259819773d

DEMO

@jeffreymckenzie

Unraveling Realities

Now that we've seen the HoloLens itself,
I'd like for us to get a picture of
Where it stands among different kinds -- or brands -- of realities,
And show some examples to hopefully understand them better.

First up in our line of realities is VR, or virtual reality.

In my own experience, and when I've talked to other people,
Generally people are familiar with VR, both as a technology and a term
–
It's when the other types of computer-generated-realities come into play
That things start to get confusing.

I got my first chance to try out a VR headset this summer.

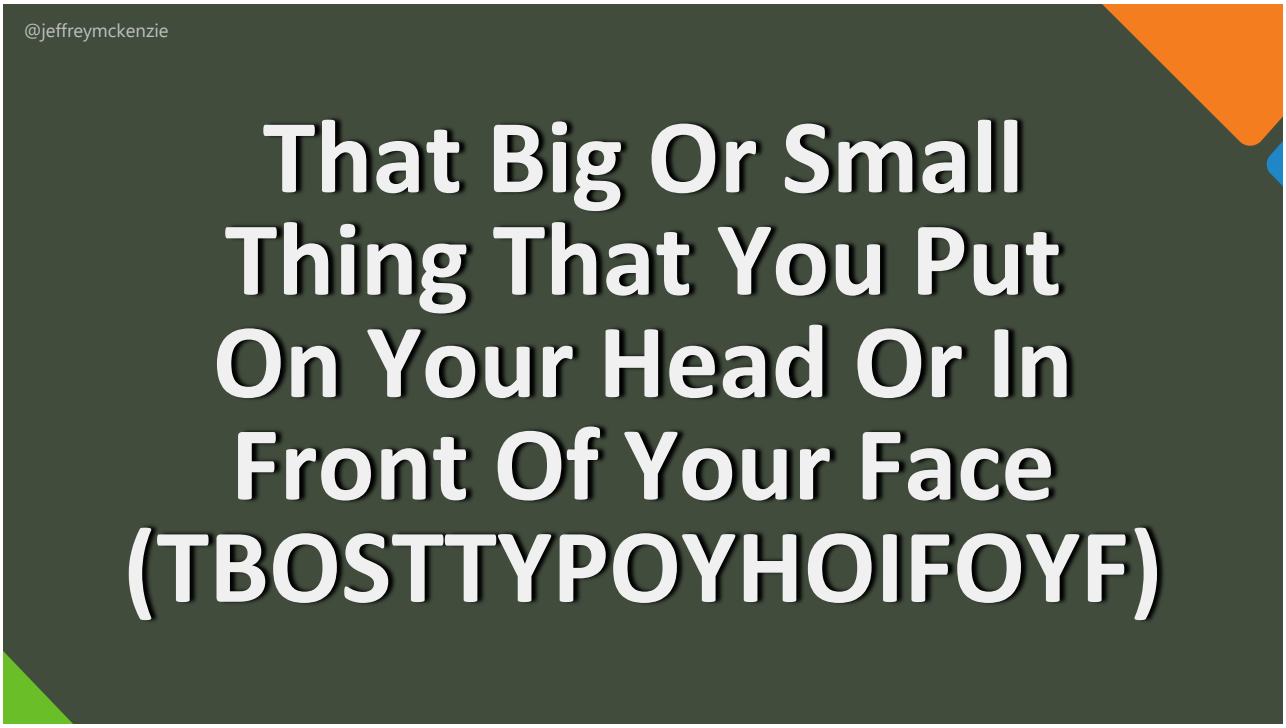
And when I say headset, perhaps I should use the more precise term…

@jeffreymckenzie

Head-Mounted Display (HMD)

Head-mounted display..

Which is the technical term for…



Something that you put on your head,

Whether that something is…

@jeffreymckenzie

An Oculus Rift….

By Sergey Galyonkin from Kyiv, Ukraine (Orlovsky and Oculus Rift Uploaded by Yakiv Gluck) [CC BY-SA 2.0 (http://creativecommons.org/licenses/by-sa/2.0)], via Wikimedia Commons https://commons.wikimedia.org/wiki/File%3AOrlovsky_and_Oculus_Rift.jpg

Or an HTC Vive….

That’s attractive, isn’t it? They look so much better in the marketing pictures.

This is the one I used this summer….

---

https://commons.wikimedia.org/wiki/File%3AHTC_Vive_(19).jpg
By Maurizio Pesce [CC BY 2.0 (http://creativecommons.org/licenses/by/2.0)], via Wikimedia Commons

@jeffreymckenzie

Or Google Cardboard…

---

https://www.wareable.com/vr/wareable-why-google-cardboard-not-oculus-rift-will-drive-the-future-of-vr-976
https://www.wareable.com/media/images/2015/10/google-cardboard-virtual-014-1445269232-CcMt-full-width-inline.jpg

@jeffreymckenzie

So, like I said, this is the device I used this summer.

I’m part of the Application Development group at Cardinal Solutions,
And for one of our quarterly practice events,
We went to the Idea Foundry,
Near west side

---

https://commons.wikimedia.org/wiki/File%3AHTC_Vive_(19).jpg
By Maurizio Pesce [CC BY 2.0 (http://creativecommons.org/licenses/by/2.0)], via Wikimedia Commons

They have co-working spaces, art exhibits,
Drone racing, and VR is one of the activities they do.

[Anyone here tried out VR?]
[what VR apps have you tried?]
So one of the apps we got to try
Is called….

---

http://schaefer-inc.com/project/columbus-idea-foundry-columbus-ohio/
http://schaefer-inc.com/wp-content/uploads/2017/06/Columbus-Idea-Foundry-1.jpg

Richie’s Plank Experience --
there are several VR apps with a similar concept,
But I’m pretty sure this is the one we tried.

So you have the headset on and the person helping
Turns you around so you are facing an elevator
That’s in your view.

You walk forward, physically, into the elevator…

---

https://toast.gg/wp-content/uploads/2016/07/Richies-Plank-200.jpg
https://toast.gg/product/steam-key-for-richies-plank-experience/

Then you have to turn around and press a button,
Which takes you to the top floor.

The elevator doors open…

---

https://i0.wp.com/scontent.oculuscdn.com/t64.5771-25/q92/s1440x1440/12680265_1622885584706631_1328453282_n.jpg?ssl=1
https://skidrowgamesreloaded.com/the-elevator-ritual-vr/

@jeffreymckenzie

And this is what you see –

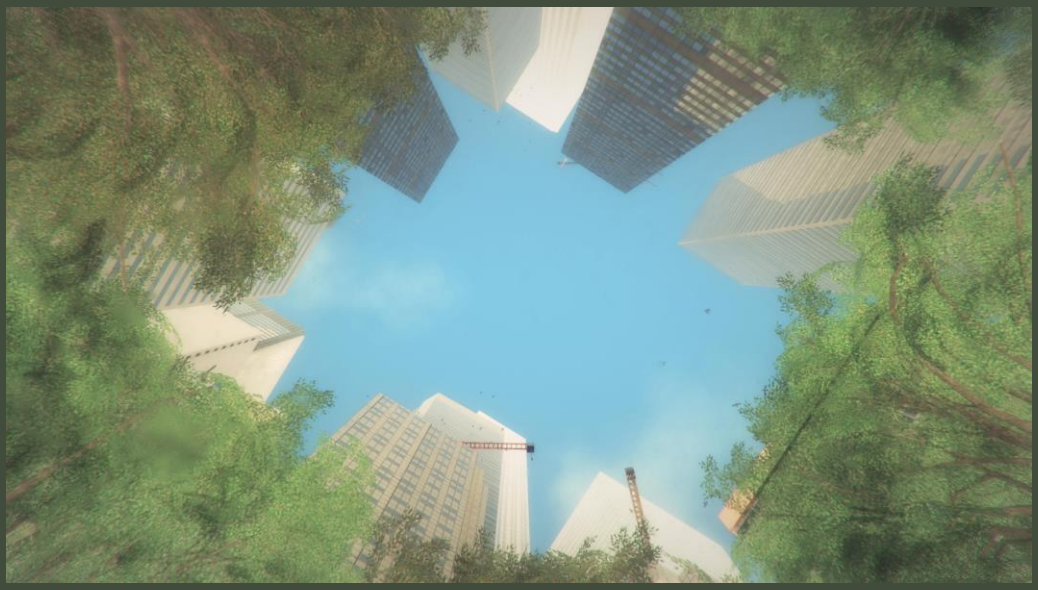
You have to walk all the way out
To the end of the plank and back.

And if you fall, the game simulates
The feeling of falling through the air,
Until, well, you “hit” the ground and it’s over.

---

http://steamcommunity.com/sharedfiles/filedetails/?id=845010789
https://steamuserimages-a.akamaihd.net/ugc/10398138998207371 8/121A54F71CC6AA74DA6D6C337A6FB3CEF81C9AEC/

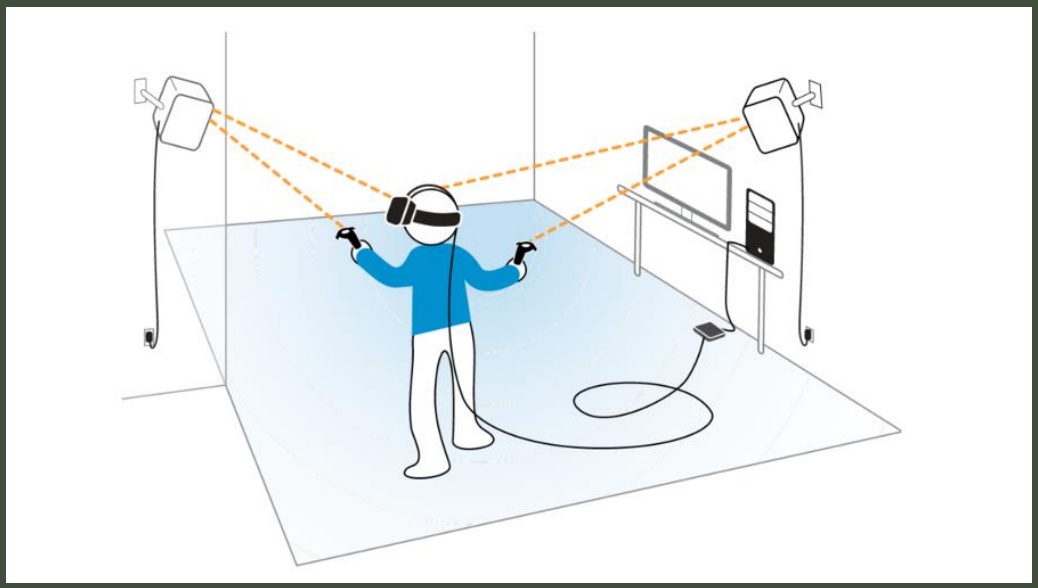
@jeffreymckenzie



Until, well, you “hit” the ground and it’s over.

---

https://steamcdn-a.akamaihd.net/steam/apps/517160/ss_1e86943cdf3357e46962545d7009dab115d0786b.1920x1080.jpg?t=1472578524



So what was fascinating is that we were in a setup
Kind of like this, you know, just walking around in a room,
And people were TERRIFIED of getting on that plank.
Now full disclosure, I didn't even try this, so
I'm a big coward and everyone who tried it is braver than me.
They have a monitor that shows what they are seeing,
And you could see people looking around,
Trying to put one foot in front of the other…

---

http://www.techradar.com/news/wearables/thinking-of-getting-a-htc-vive-check-that-your-room-is-this-size-first-1314450
http://cdn.mos.cms.futurecdn.net/a5b850e121a5aee3c4a6bbc624fbda1c.jpg

Some people just gave up, others made it a few steps out
And turned back.
One guy was just determined to get to the end,
He just kind of threw himself out there,
Almost hyperventilating, and made it all the way back.
And he looked just exhausted.
So that's the kind of power that virtual reality
Can have on our experiences.

---

http://steamcommunity.com/sharedfiles/filedetails/?id=845010789
https://steamuserimages-a.akamaihd.net/ugc/10398138998207371 8/121A54F71CC6AA74DA6D6C337A6FB3CEF81C9AEC/

So AR is up next.

AR stands for Augmented Reality.
Where Virtual Reality replaces your entire view
With a computer generated experience,

Augmented Reality is more of a light touch –
It takes your real environment and layers
Computer-generated objects over it.

@jeffreymckenzie

As in this example, where graphics are overlaid
On something like a smartphone or heads-up display.

So I'd like to explore by talking a little bit about
And one of the biggest successes
In the AR format.

Can anyone guess what that might be?

---

https://commons.wikimedia.org/wiki/File%3AMediatedReality_on_iPhone2009_07_13_21_33_39.jpg
By Glogger (Own work) [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0) or GFDL (http://www.gnu.org/copyleft/fdl.html)], via Wikimedia Commons

That’s right, the craze of summer 2016, Pokemon GO.

[How many here have at least downloaded this
Onto your phone and tried it out?]

Yeah, me too.

Actually I think I have 150 apps on my phone,
And 120 of those my daughter installed,
So OF COURSE I have Pokemon GO on my phone.

---
https://upload.wikimedia.org/wikipedia/commons/thumb/2/23/Pok%C3%A9mon_GO_logo.svg/1000px-Pok%C3%A9mon_GO_logo.svg.png

@jeffreymckenzie

Did it seem particularly realistic?

Why or why not?
[…..]
maybe the images weren't realistic,
But the fact that they were "appearing" in actual places
I think gave them a sense of realism]
There were actual physical places where you know
You would find Pokemon.

---
https://www.gamecrate.com/sites/default/files/field/image/pokemon%20go%20header%201.jpg
https://www.gamecrate.com/20-tips-and-secrets-pok%C3%A9mon-go-wont-tell-you/14078

From what we've seen so far with the HoloLens,
Mixed Reality kind of combines VR and AR,
With more realistic projections that appear
To be part of our actual reality.
VR gives you an intense experience,
But it's very hard to recreate the physical world.
AR gives us the physical reality, but again,
What it augments is often not quite as engaging as VR.

Allright, so the time has come to take a closer look
At actually building a Mixed Reality application for HoloLens.
To get a look at the full list of what you need,
You can visit this link…

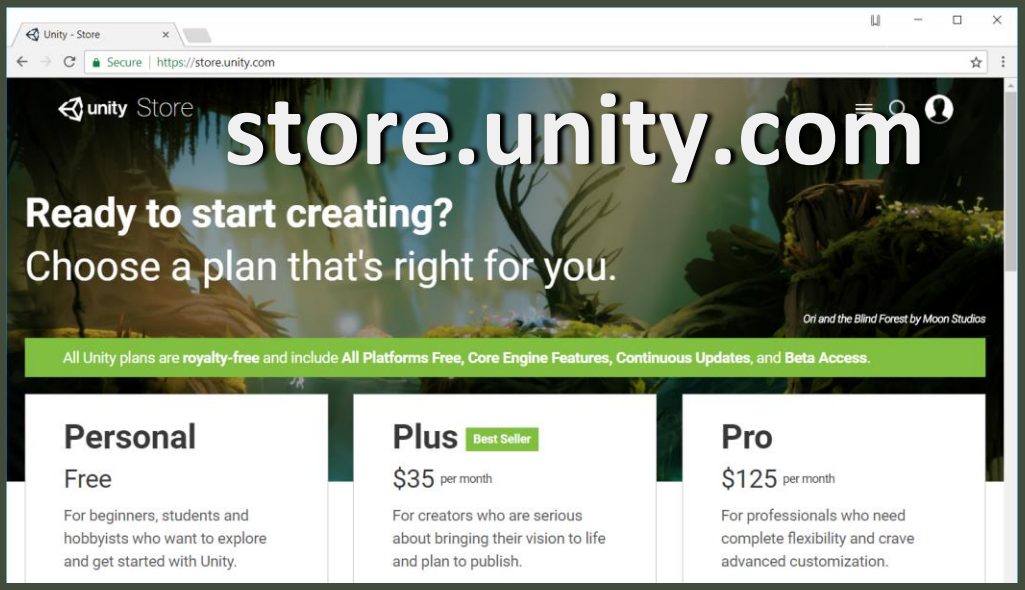
developer.microsoft.com/en-us/windows/mixed-reality/install_the_tools

Or you can just google hololens tools,
Which will get you to this link.
There are a few prerequisites –
The first being the operating system…

You will need the Windows 10 SDK,
Which is supported on multiple operating systems,
However, not all tools for HoloLens development work
On all operating systems
Windows 10 is recommended for the best experience.
Next you will need Unity --

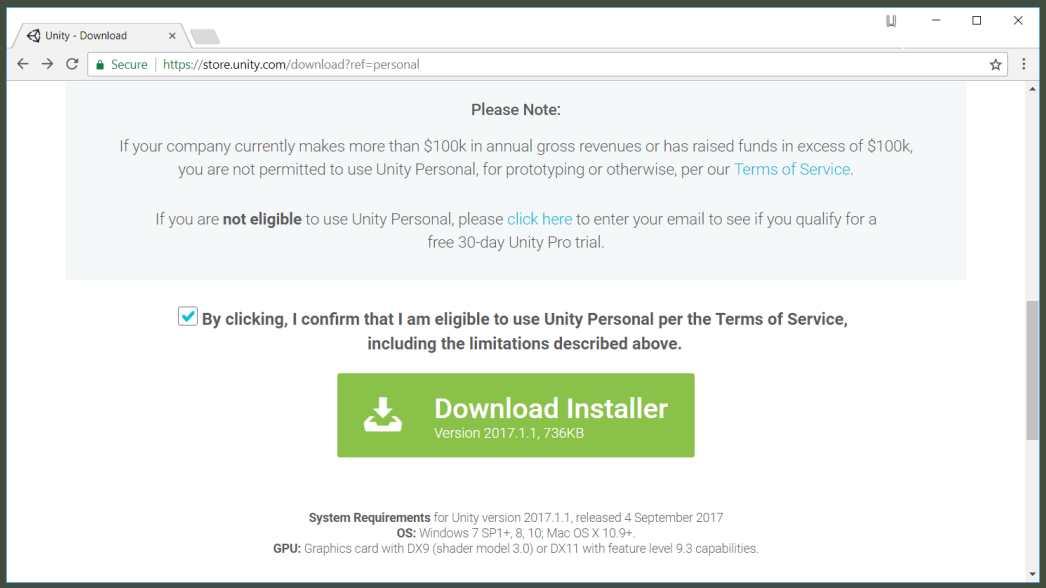
Which is a 2D and 3D game engine created by Unity technologies.
[Anyone here used Unity before?]
[What have you done with it?]
So it’s called a game engine, but really
It can be used to create many different kinds
Of graphic design and animations.
you can get it at….

@jeffreymckenzie

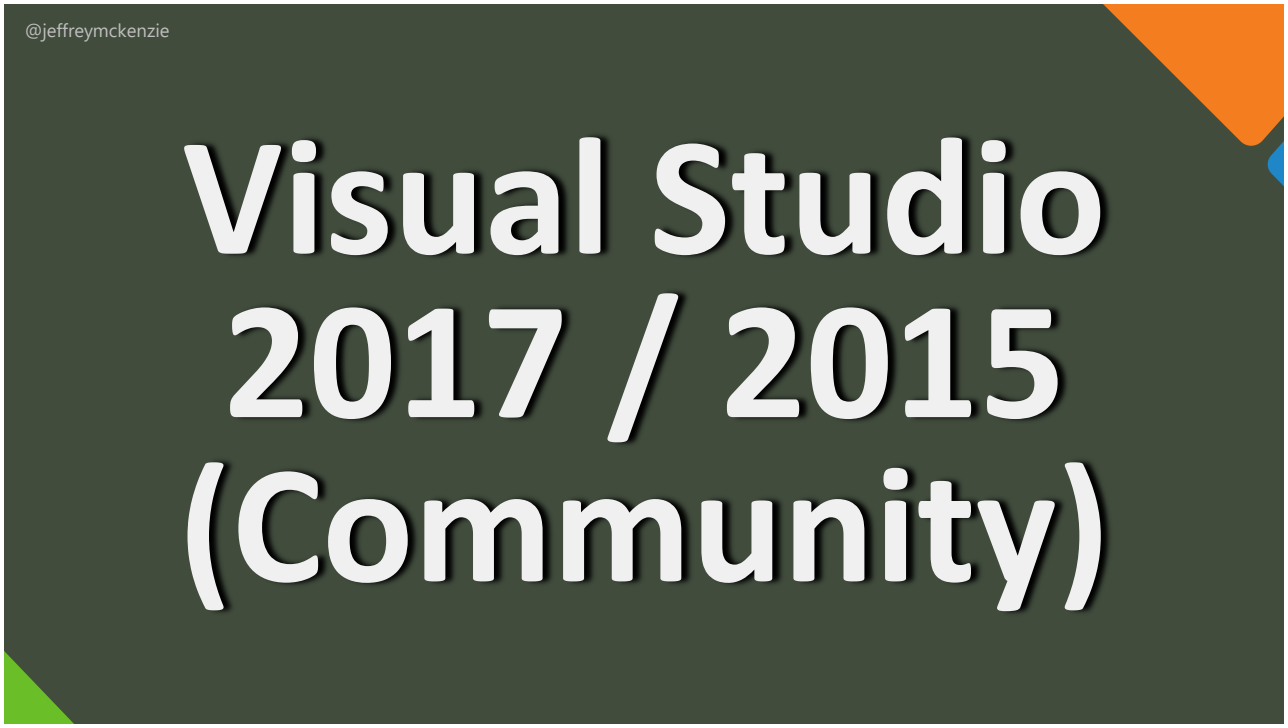


Store.unity.com.
If you click on Personal
(on the lower left)..
That takes you to the download page.

@jeffreymckenzie

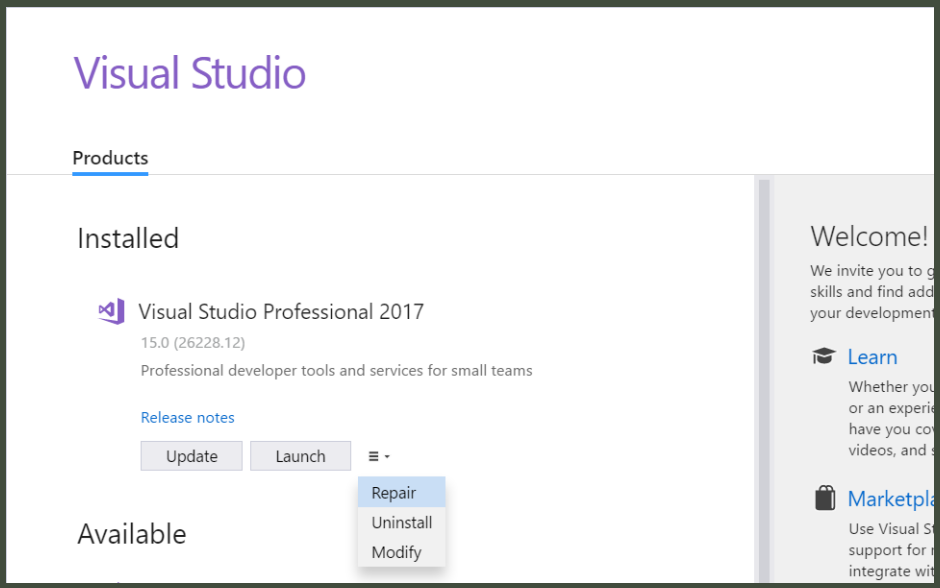


One note for the personal edition
is that you can't use it for company work,
if the company is larger than 100K.
You also can't use it if you personally
Have made 100K from using Unity during the last year.
The latest version is 2017.1.1.
Next you will need an integrated development environment,
Or IDE…

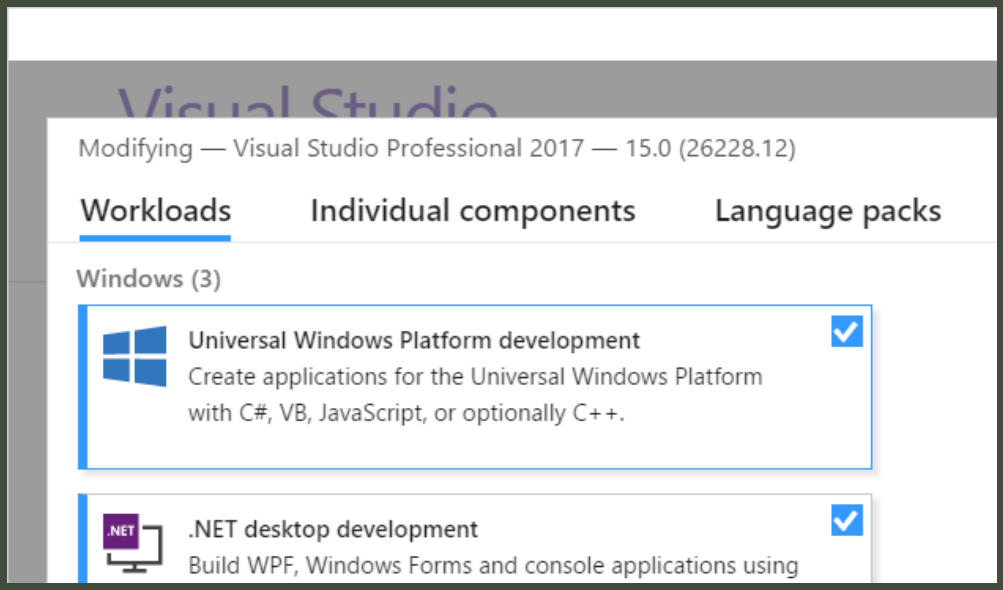


Microsoft recommends Visual Studio 2017,
And that's what you should probably use
If you already have it.
Right now I am using 2015 update 3
and that works fine as well.
The community edition is free and
Provides full support.

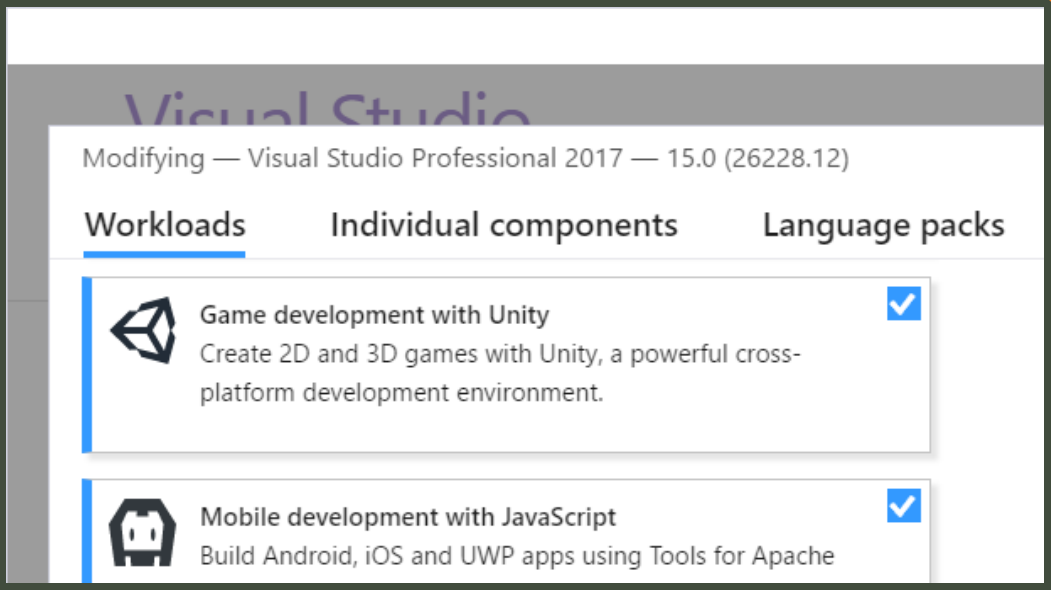
@jeffreymckenzie



To make sure you have the correct features installed
For 2017, you can go into the Visual Studio installer
And select the Modify option.

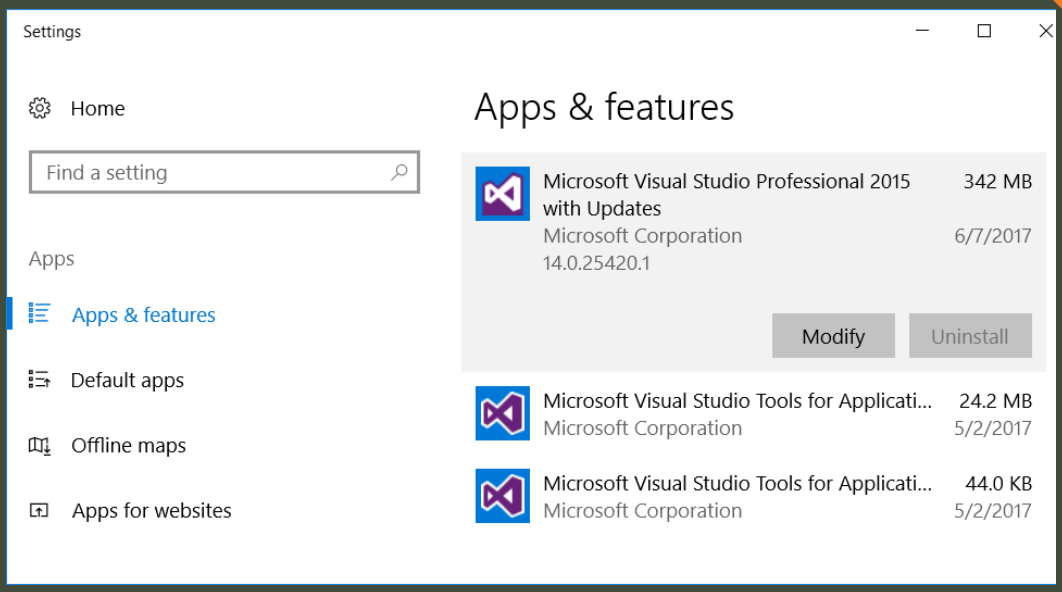


Select Universal Windows Platform Development…
This includes the Windows 10 SDK I mentioned earlier.



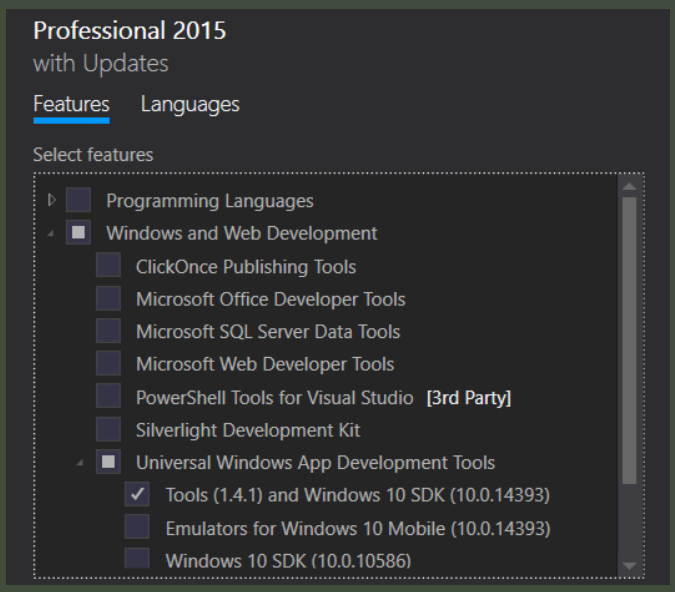
And also Game Development with Unity

@jeffreymckenzie

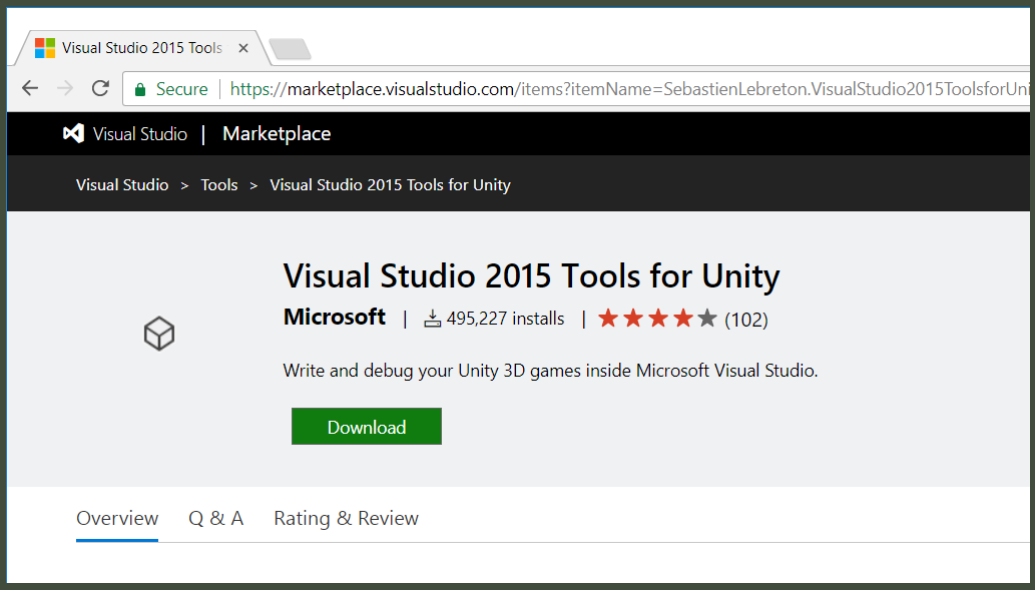


If you're working with Visual Studio 2015,
You can go into Apps & Features in settings,
And select the Modify option for your edition.

@jeffreymckenzie



Under 2015 Features you'll want to select
Tools and Windows 10 SDK
As part of Universal Windows App Development Tools,
Under Windows and Web Development



Another thing you'll need for VS 2015 is the
Visual Studio Tools for Unity –
This allows you to do things like run an application in Unity
And set breakpoints within Visual Studio.
Visual Studio 2017 already comes with Unity Integration.
And then, in order to see what your completed app
Looks like, you can buy a $3000 hololens,

@jeffreymckenzie

HoloLens Emulator

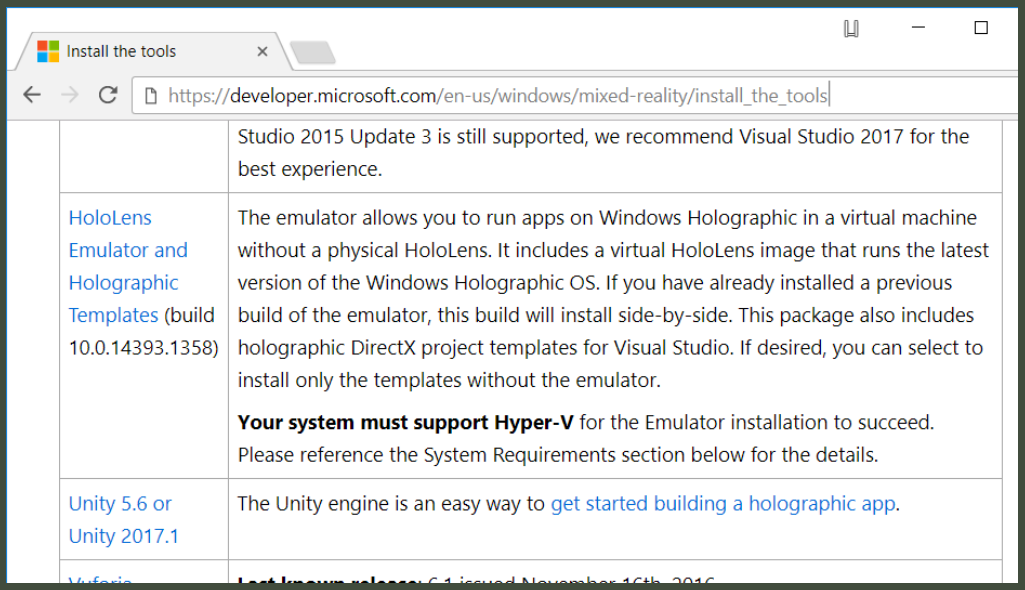
Or you can download and install
the HoloLens Emulator

@jeffreymckenzie

This is what it looks like,
And we will get a better look here in a few minutes.
You don't get to see to see what the app looks like
In actual physical surroundings,
But it's a pretty good indicator
Of how the app is going to behave.

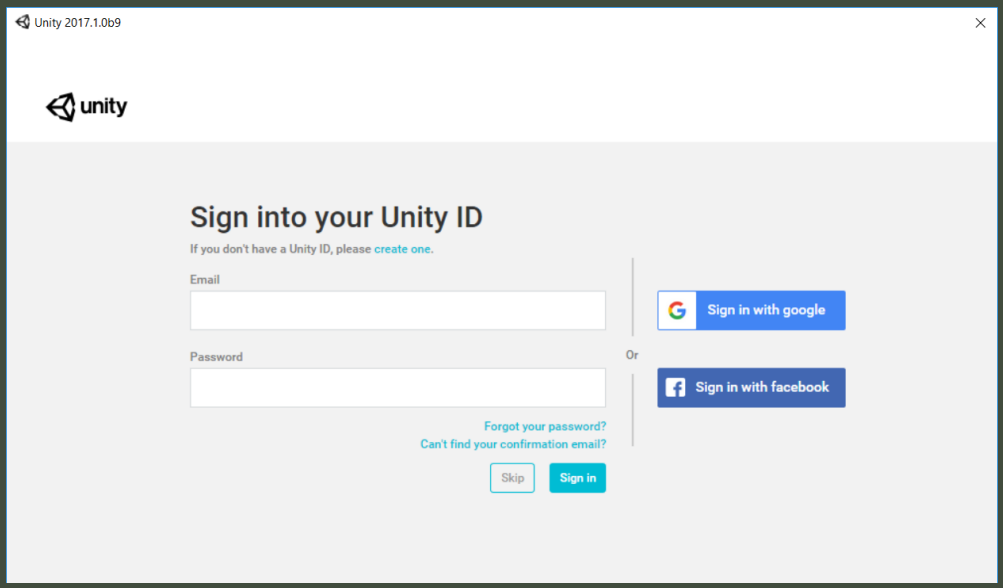
---

https://developer.microsoft.com/en-us/windows/mixed-reality/using_the_hololens_emulator
https://az835927.vo.msecnd.net/sites/mixed-reality/Resources/images/Emulator.PNG


Install the tools
https://developer.microsoft.com/en-us/windows/mixed-reality/install_the_tools
Studio 2015 Update 3 is still supported, we recommend Visual Studio 2017 for the best experience.
HoloLens Emulator and Holographic Templates (build 10.0.14393.1358)
The emulator allows you to run apps on Windows Holographic in a virtual machine without a physical HoloLens. It includes a virtual HoloLens image that runs the latest version of the Windows Holographic OS. If you have already installed a previous build of the emulator, this build will install side-by-side. This package also includes holographic DirectX project templates for Visual Studio. If desired, you can select to install only the templates without the emulator.
Your system must support Hyper-V for the Emulator installation to succeed. Please reference the System Requirements section below for the details.
Unity 5.6 or Unity 2017.1
The Unity engine is an easy way to get started building a holographic app.

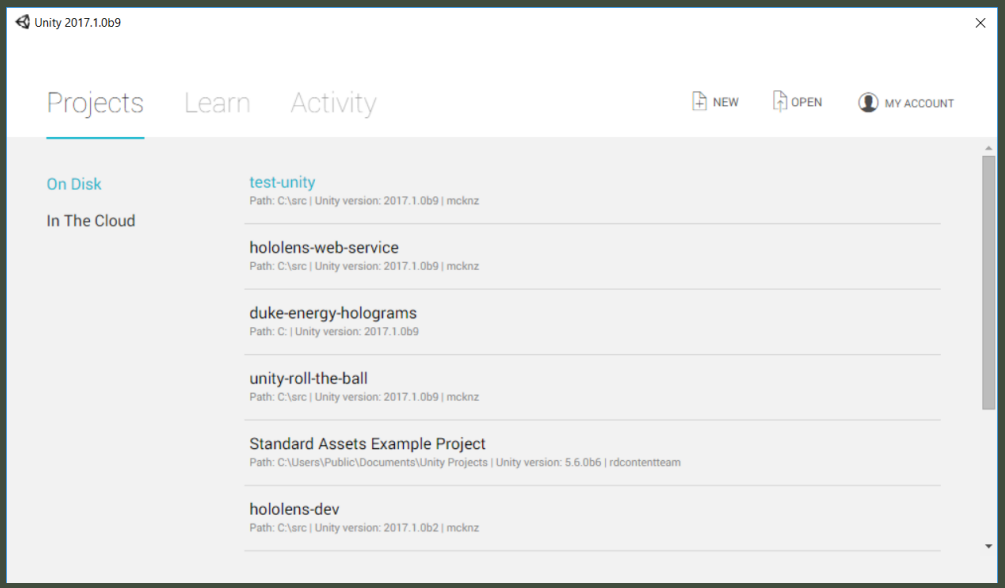
OK lets take a look at Unity
To get started with our project.
Once you've downloaded and installed
Unity for the first time…

@jeffreymckenzie

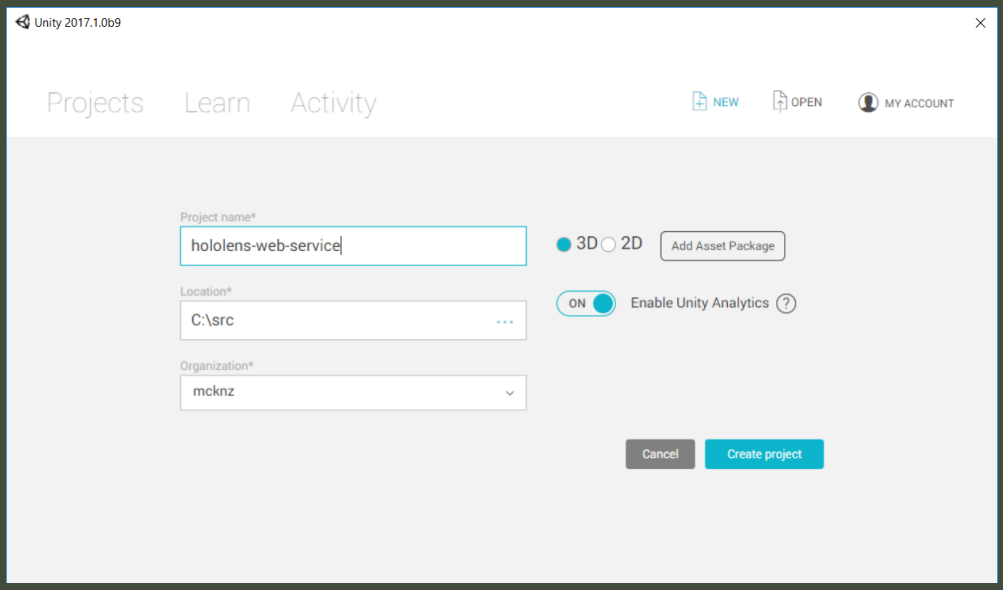


You'll need to sign in.
You can either create an account
With Unity.com, or use your Google or Facebook account.
Once you sign in….

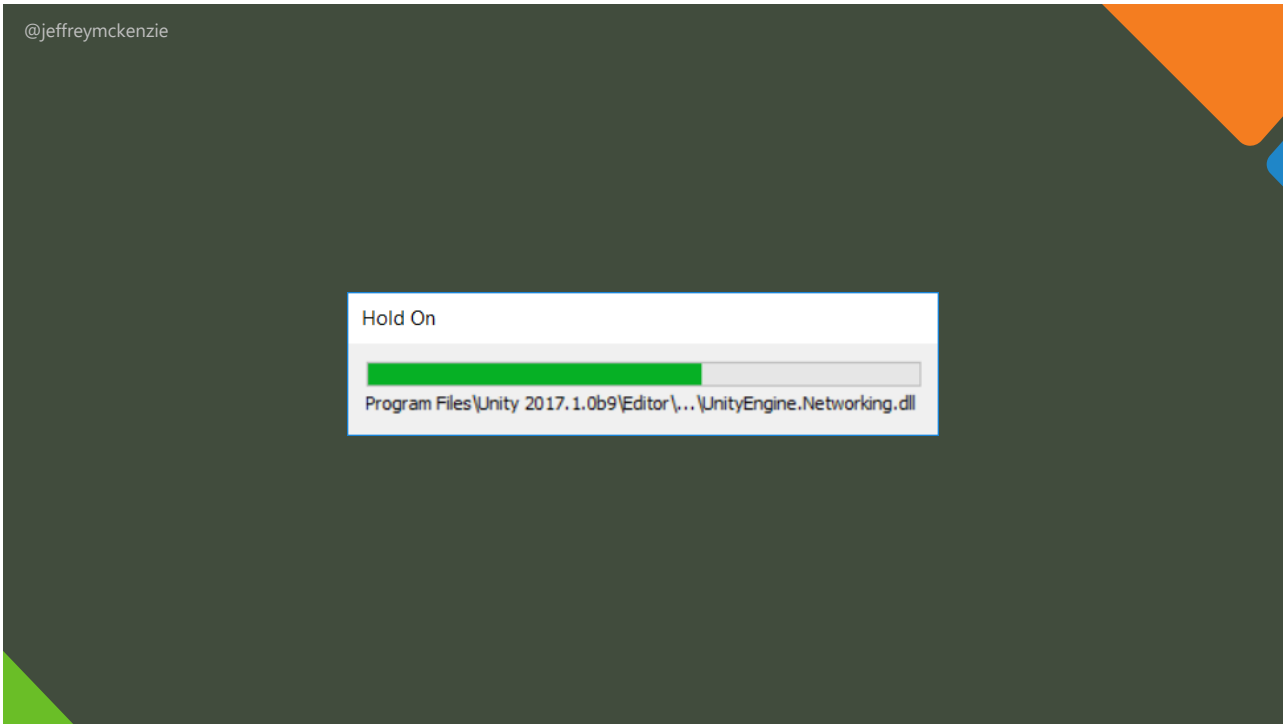
@jeffreymckenzie



You're shown a list of any projects
You might already have –
If this is the first time you've used Unity
You won't have any projects here.
To start a new project, we can
Click the New icon on the top right side.

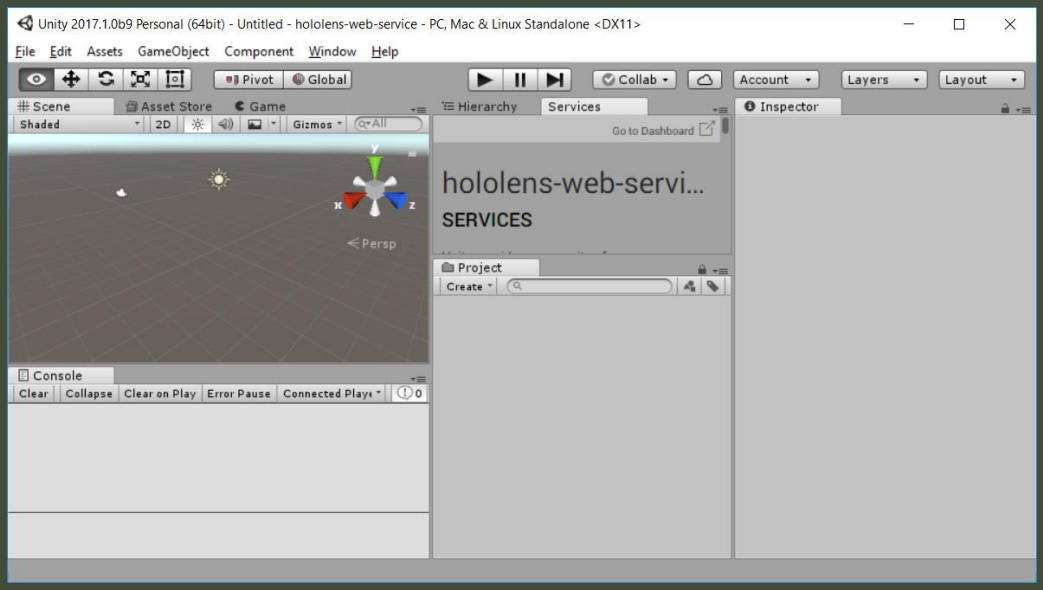


Here we can give a project name --
We'll call it hololens-web-service --
We can give A location on the local drive
where we want the project to live.
Here I'm setting a location just under my C drive.
On the right, we will want to select a 3D project.
Then we click create project….



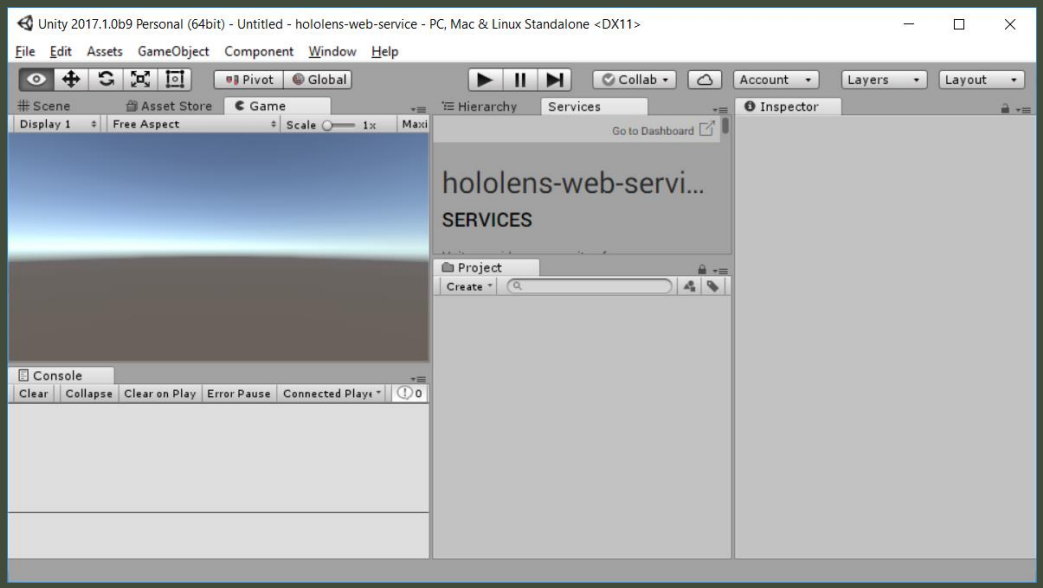
the new project gets created…

@jeffreymckenzie



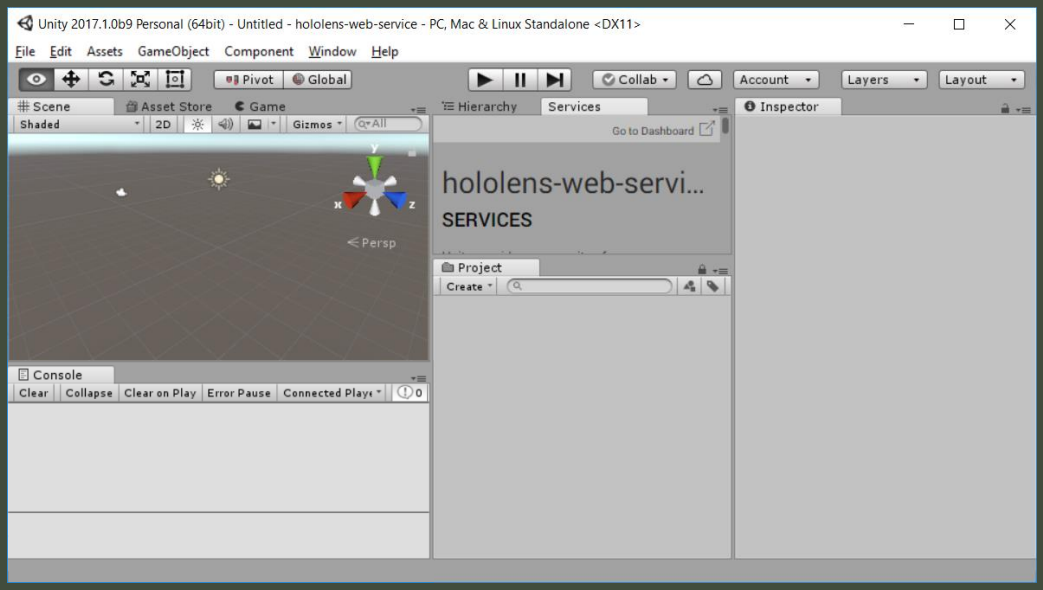
Once it's done loading we can see the full Unity interface.
There are 4 main areas here we'll be paying attention to.
First is the scene panel, on the top left.
This is where you place and manipulate 3d objects.

@jeffreymckenzie



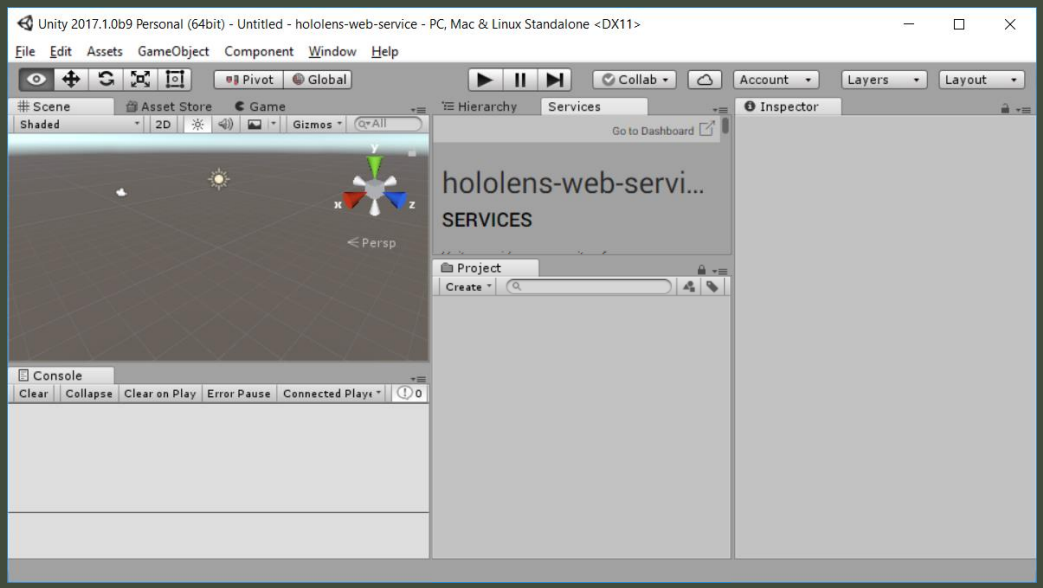
In this same area is also the game tab,
Which shows us a preview of the scene
When it's run.
So for example, I can create a 3D object

@jeffreymckenzie



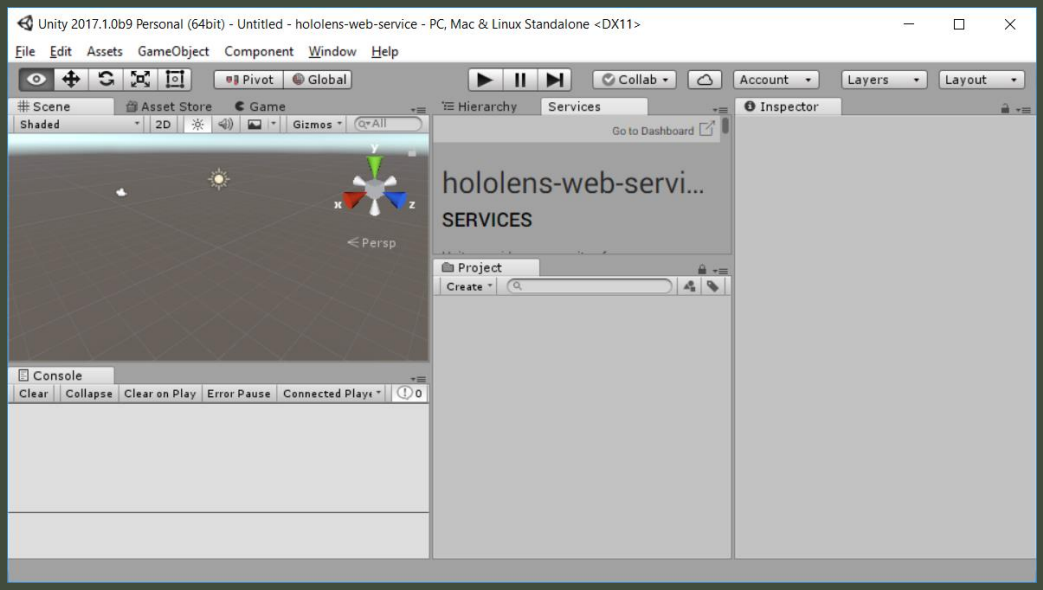
On the bottom left is the console,
Which displays output,
Such as logging and errors.

@jeffreymckenzie



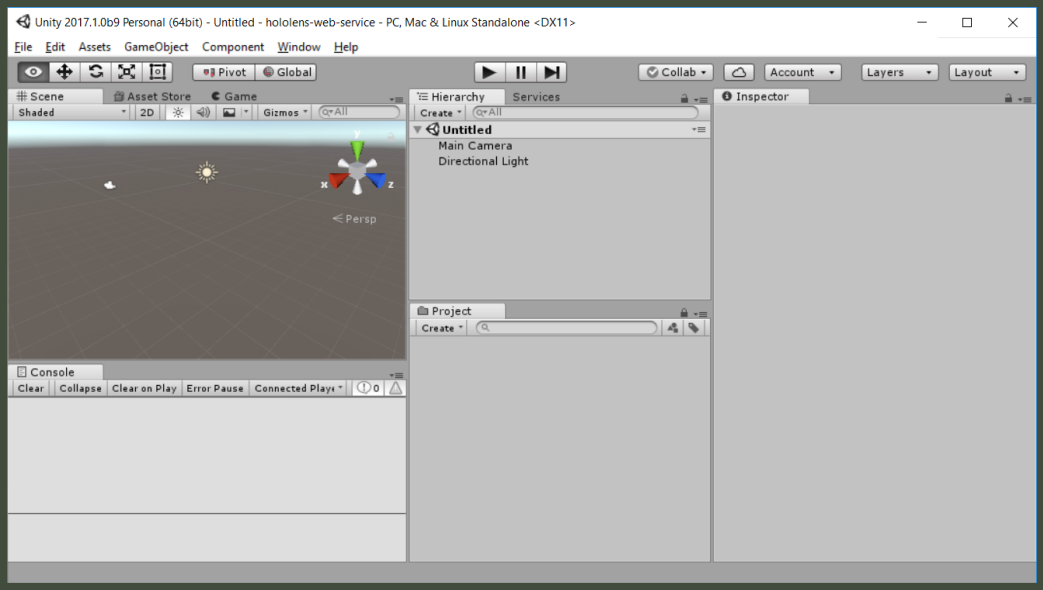
In the center we have a top and bottom panel as well.
On the bottom is the project tab,
Which will show everything we add to the project

@jeffreymckenzie



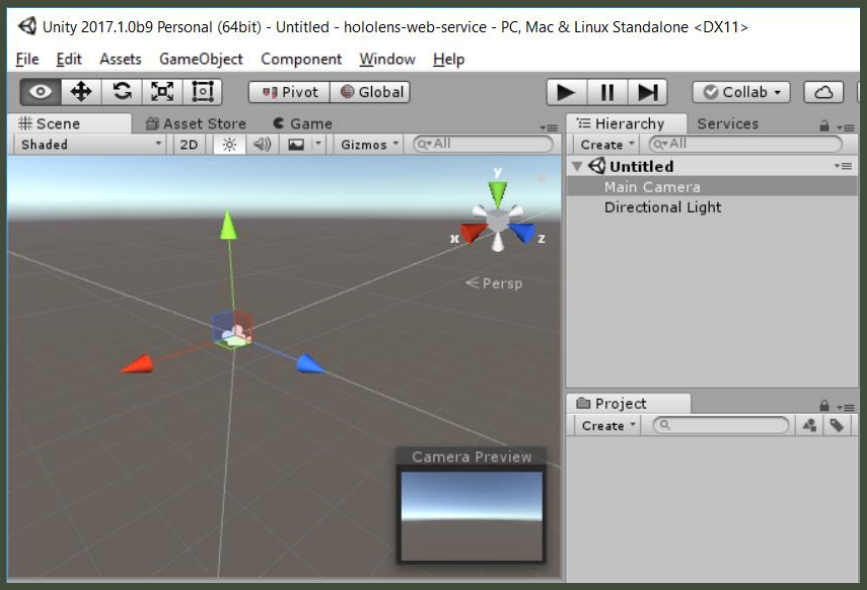
The top middle tab shows Services by default,
Which provides links out to different Unity resources –
But we really are interested in the Hierarchy tab,
Which shows us all the contents of the scene.

@jeffreymckenzie

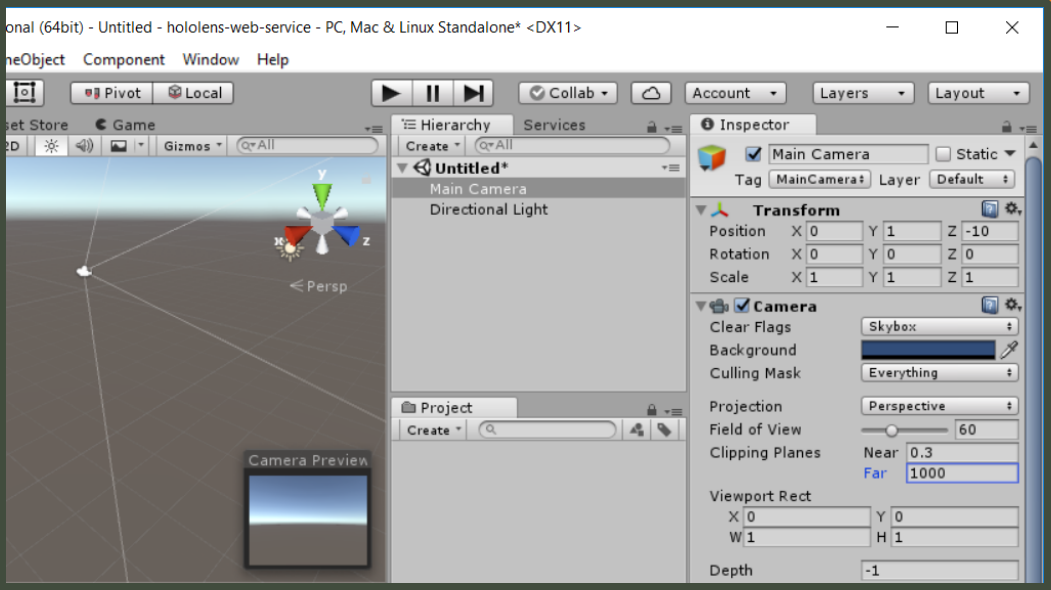


Here at the top we have the name of the scene,
Which is “Untitled” by default.
We also have a camera, called Main Camera
And a light source, called Directional Light.

@jeffreymckenzie

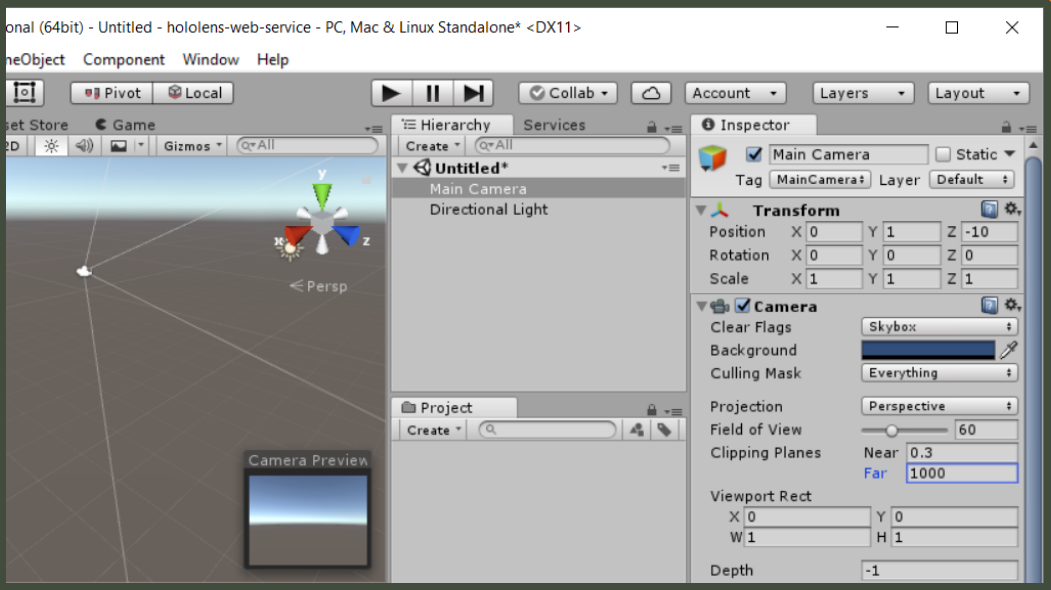


Let's take a closer look at the camera –
When you select it, you see the geometrical shape
That represents what we are looking at.

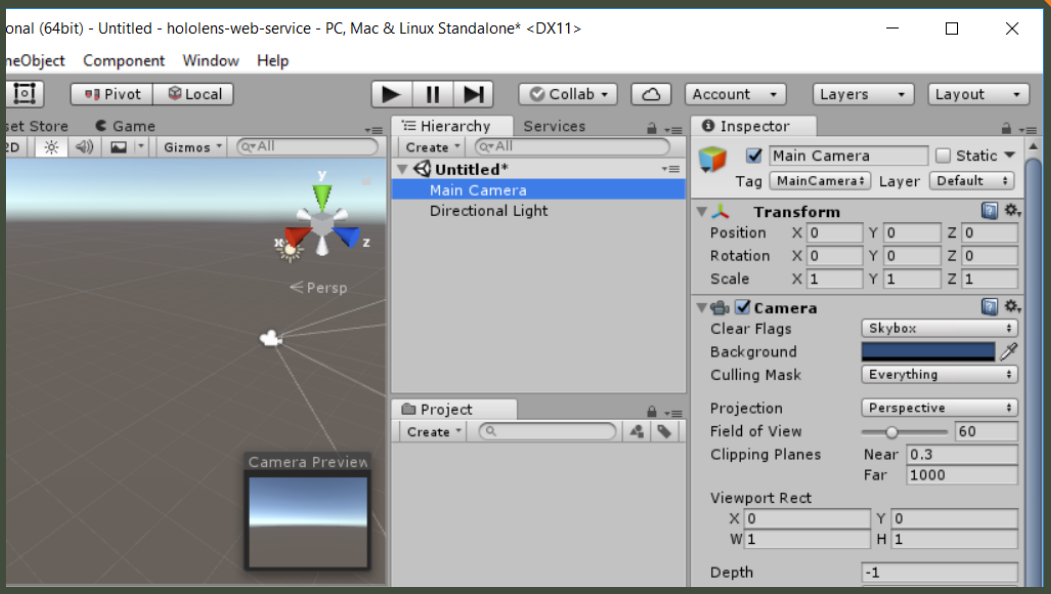


The fourth area we want to look at
Is the Inspector tab on the far right.
This area gives you more information
And settings about the selected item,
In this case, the camera.

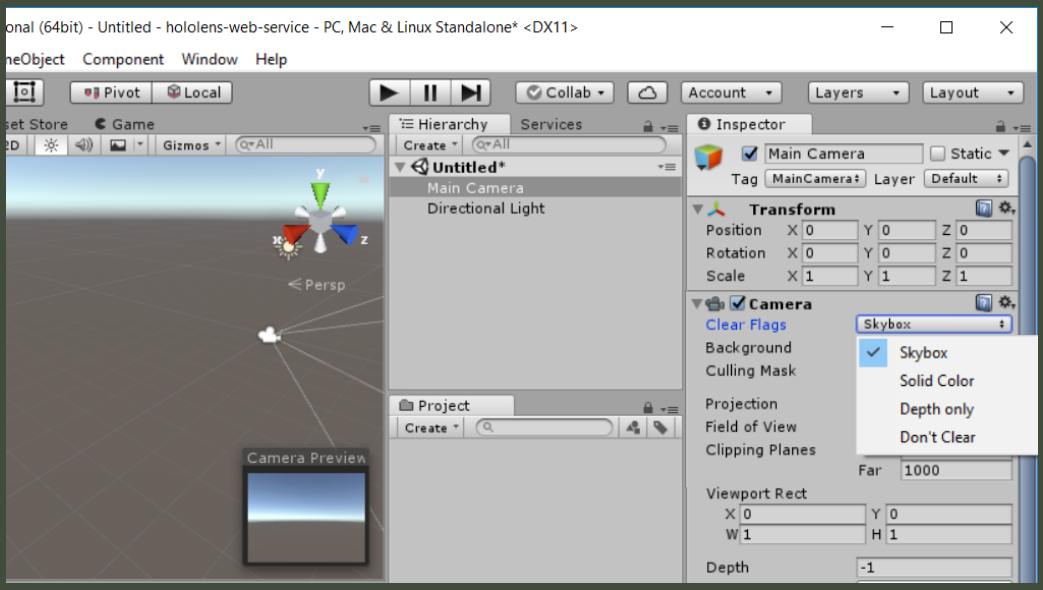
@jeffreymckenzie



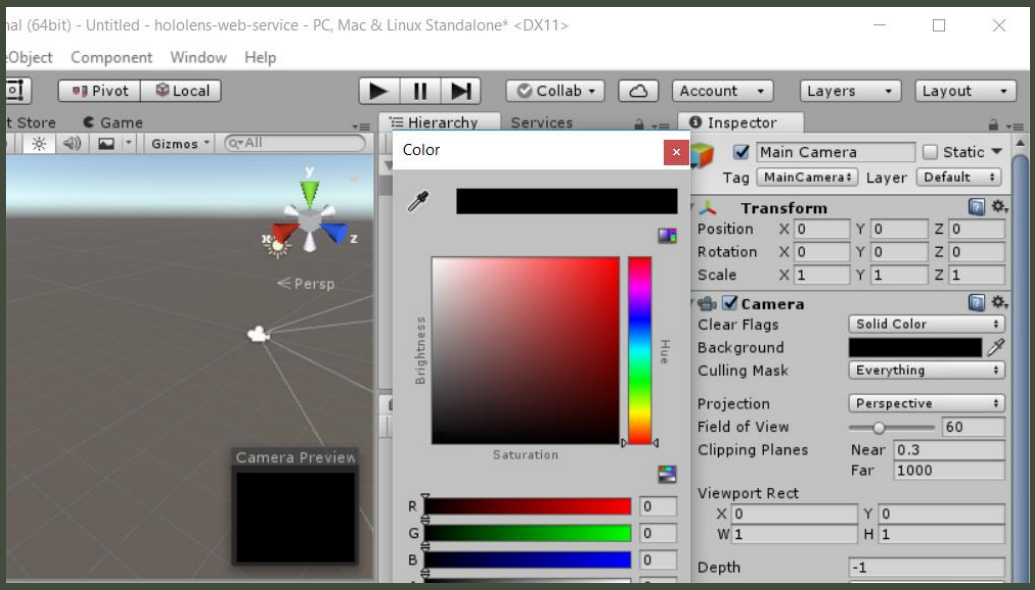
The first setting on the camera
That we want to look at is transform,
Specifically the Position section.
X is left to right, Y is up and down,
And Z is forward and backward.
These numbers correspond to meters
In the rendered world.
So by default we have X 0, Y 1, and Z: -10.
This is a good position for a traditional game –
Slightly up and back.
But for HoloLens we want that to be right
In front of the user, so we set it to
All zeros.



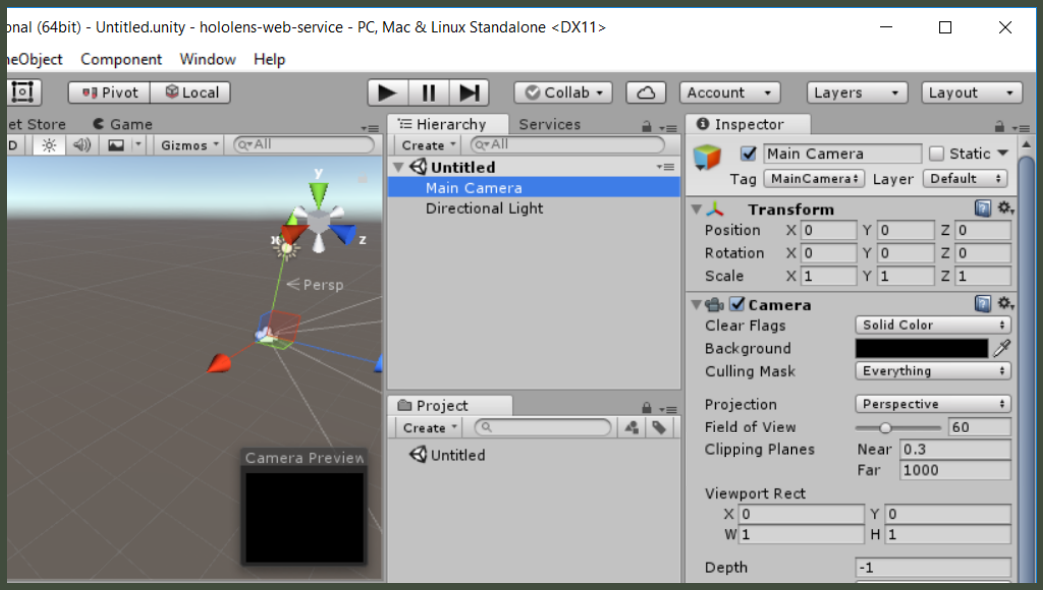
So that moves it forward and down a bit.
Next we need to change a couple of things
In the Camera area: Clear Flags and Background



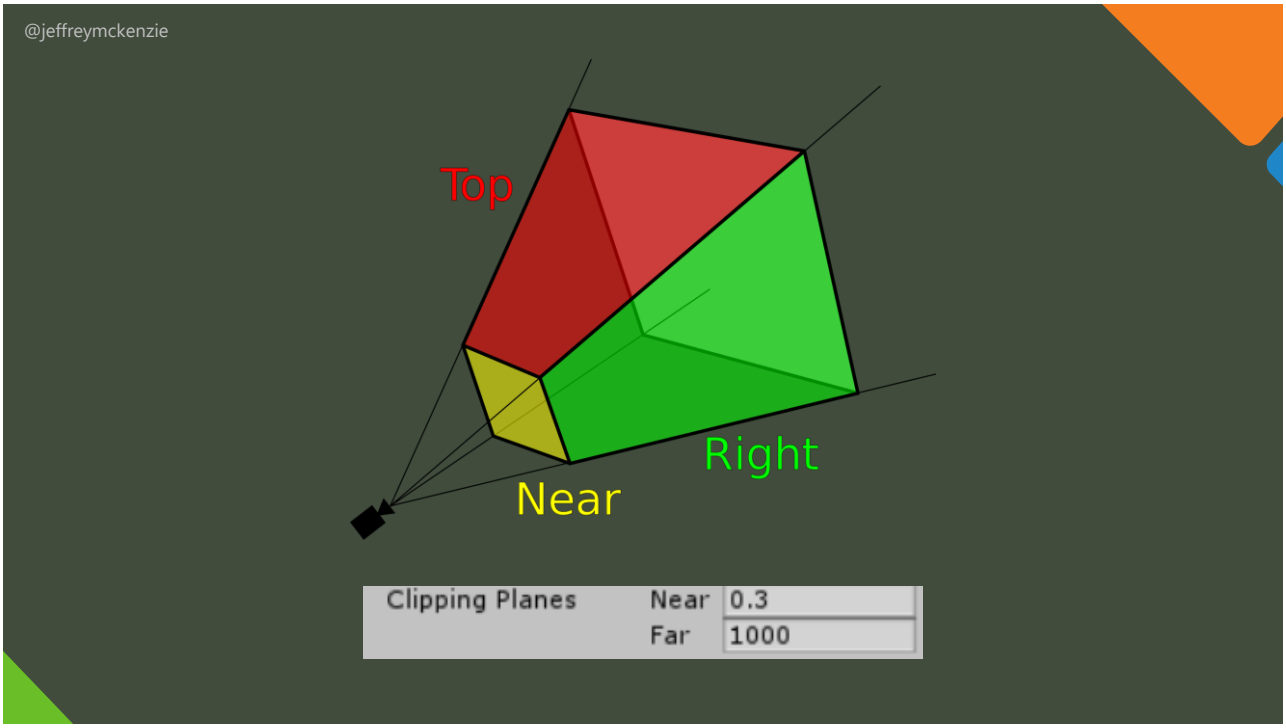
The default for clear Flags is skybox –
[CLICK]
This is something you can apply a material to
To give the effect of something on the horizon.
We don't want that in HoloLens,
Because we want everything to be transparent.



So we select solid color,
and change the background to black –
This ensures that whatever our scene is,
It will render as transparent in the HoloLens.
When light is applied to the scene,
Essentially whatever is black will disappear.
And you can see our Camera preview there
Has turned black as well.



Last item we need to adjust on the camera
is the clipping panes, down on the lower right [CLICK].
Clipping Panes determines the points where
Objects are rendered, from the nearest point
To the farthest places.
To help you visualize a little bit….



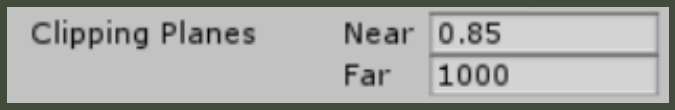
Here's a better view of what the camera sees –
You have the near clipping point in yellow
And the far clipping point at the back of the shape.
So the default near clipping pane for unity
Is point-three meters, which is less than a foot.
Looking at objects so close up can be uncomfortable,

==========

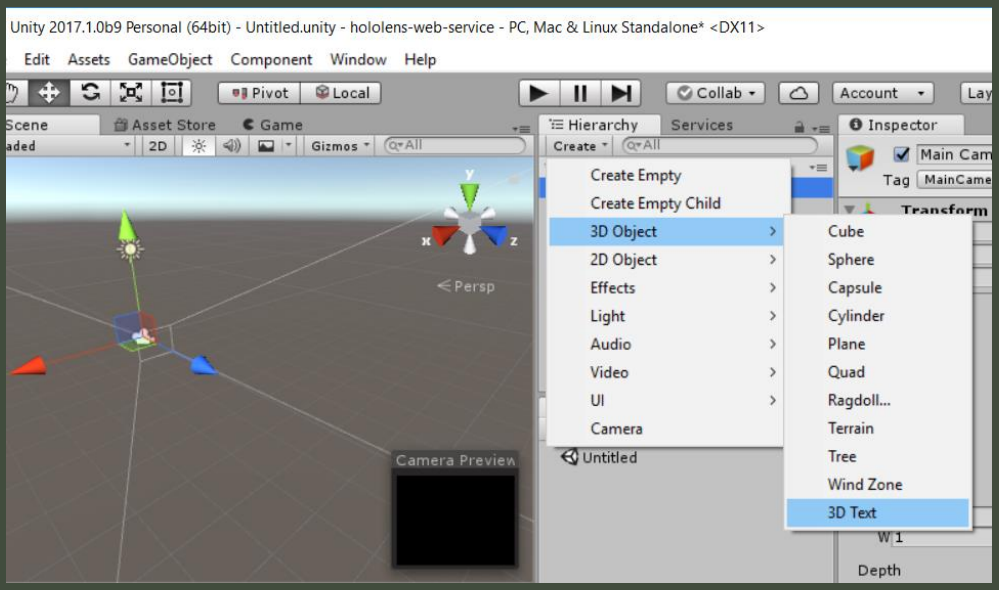
https://upload.wikimedia.org/wikipedia/commons/0/02/ViewFrustum.svg

By MithrandirMage [CC BY-SA 3.0

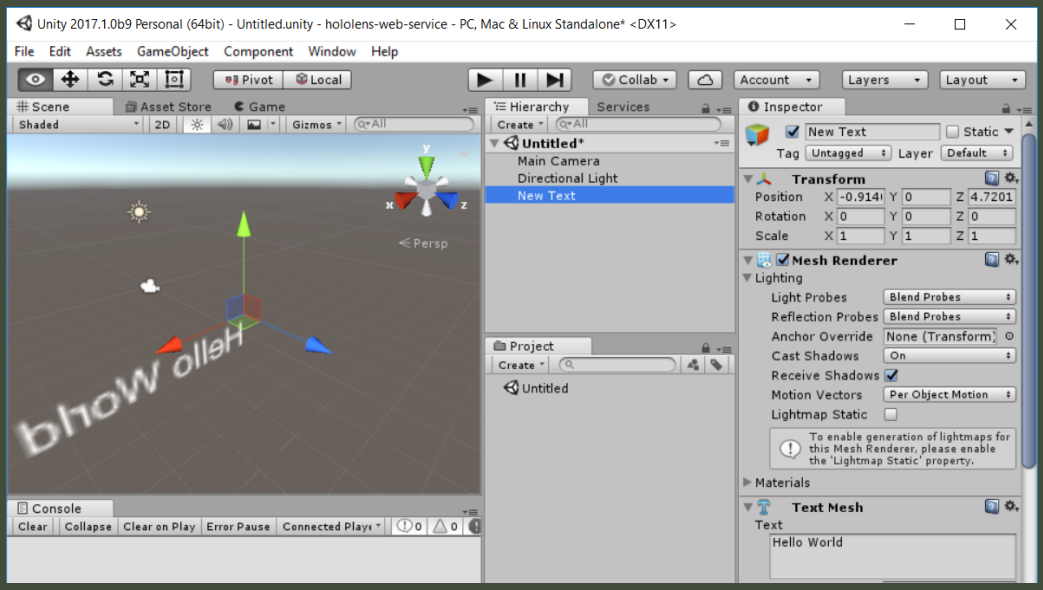
(https://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons



So Microsoft recommends a setting of point-85,
Or about 2 and a half feet away
We can keep the far setting the same,
Because we don't want to restrict
how far back objects appear.
Ok, so that's it for the camera –
As far as the lighting, the defaults are fine,
So we don't need to make any changes there.

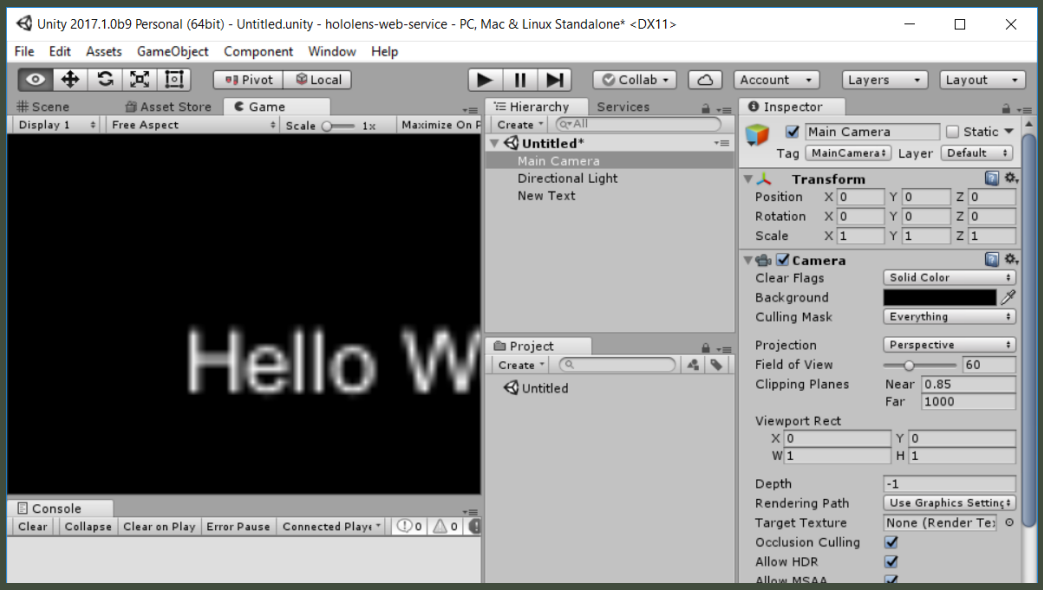


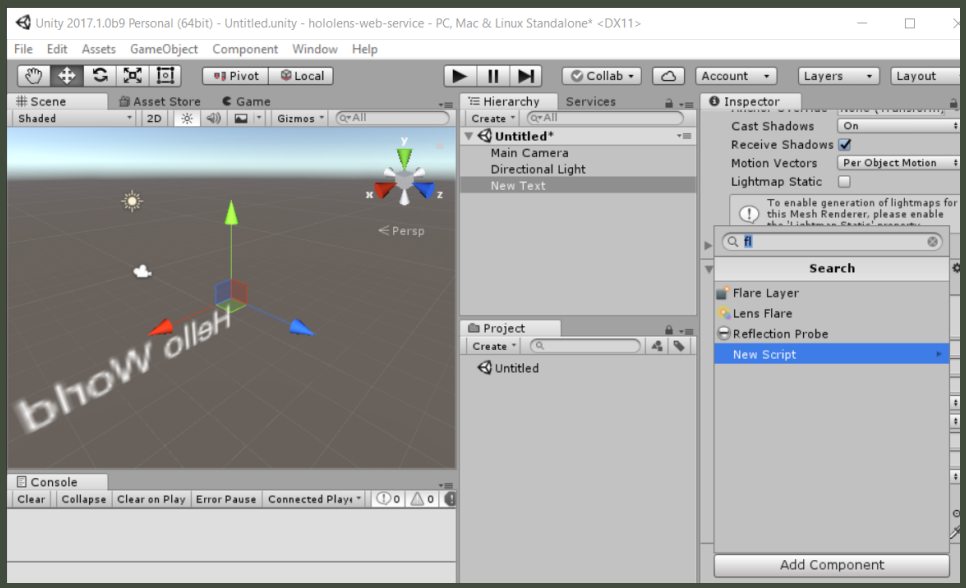
Let's go ahead and add our 3D text.
I've moved the camera over to the left a bit
So we can see the working area better.
Then from the Hierarchy we click the create menu –
Move to 3D object, then to 3D text.



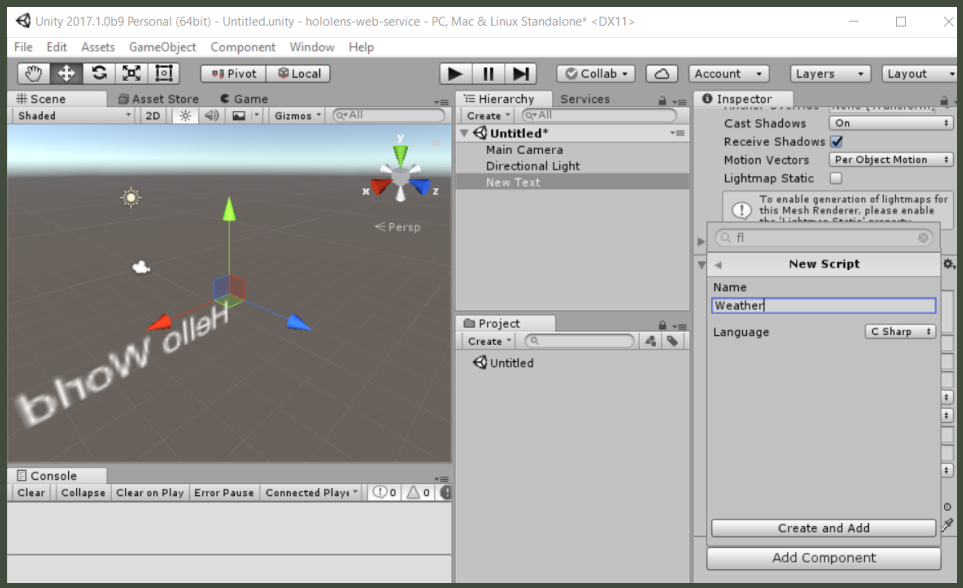
Unity drops that into the scene –
You can see it looks backward
because it's in front of the camera.
When we click on the Game tab,
We can see how it's actually
Going to look.

@jeffreymckenzie

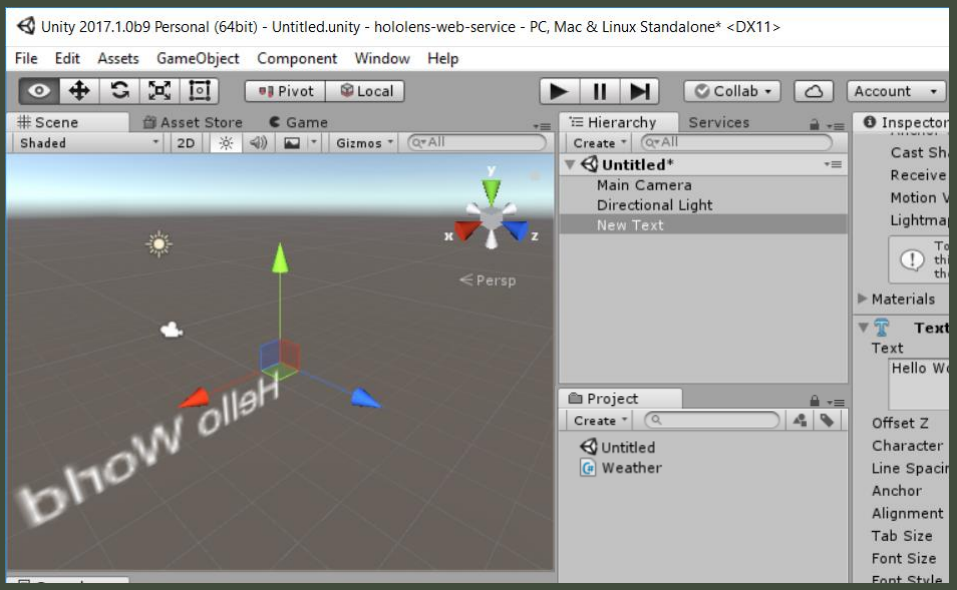




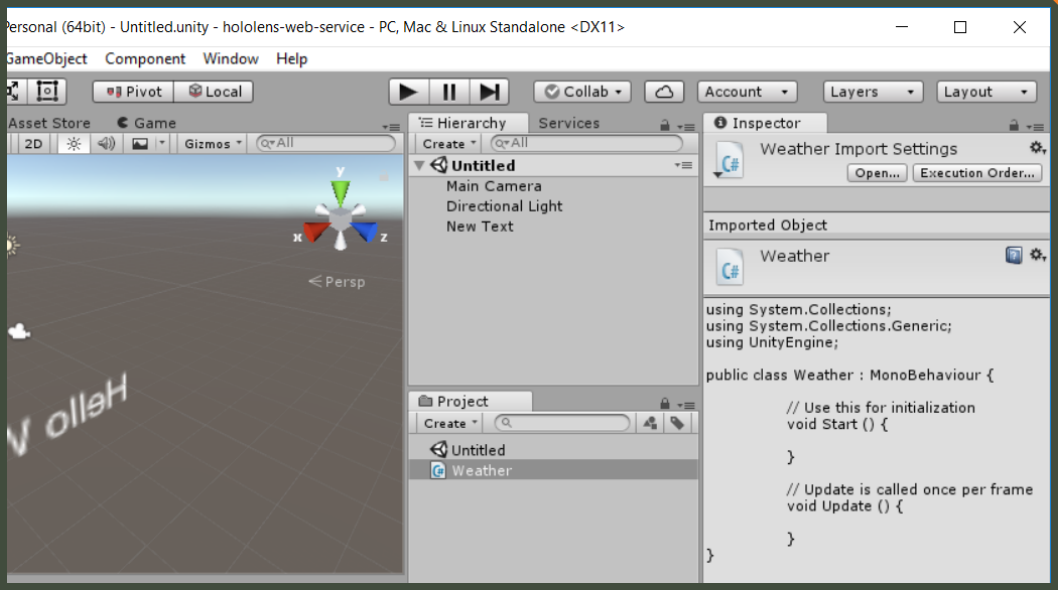
One way to attach a script to an object
Is to go to the bottom of the inspector tab
And click on Add Component.
A component in Unity is a module
That affects the appearance or behavior
Of a given object.
If we click on New Script here,



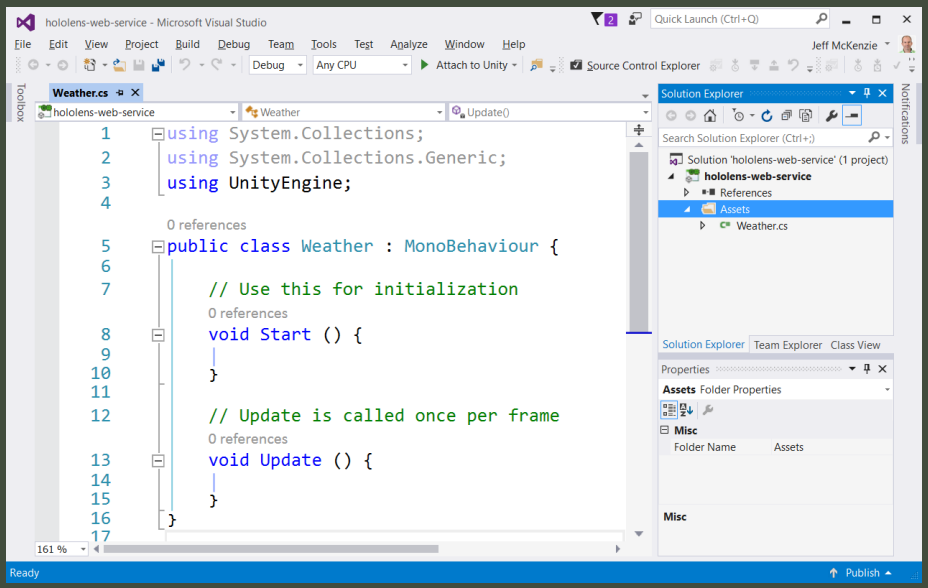
We can specify the name of the script,
Which here we'll name Weather, since our text
Will be displaying the current weather.
You can also set the language, C#.



Once we do that, you can see that the script
is added to our Project tab on the lower right.
When I click on the script, I get a view
Of the code in the Inspector tab….

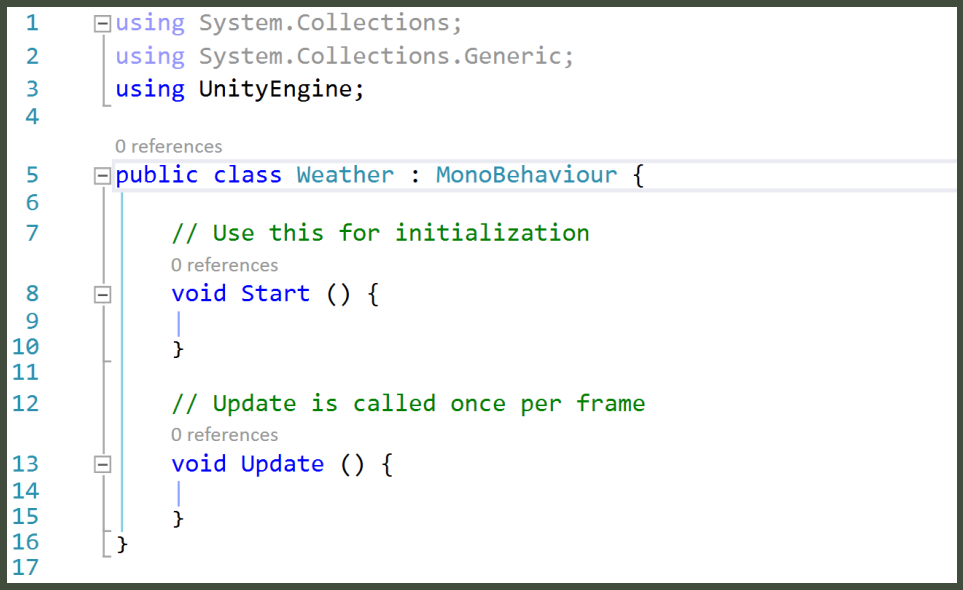


Now this is not a very convenient way
To edit text and code, but if we double-click
On the script file in the project tab,
This opens up in Visual Studio,



And we can work on it
just like any other piece of code.
Let's take a closer look at the
Generated C# class here.

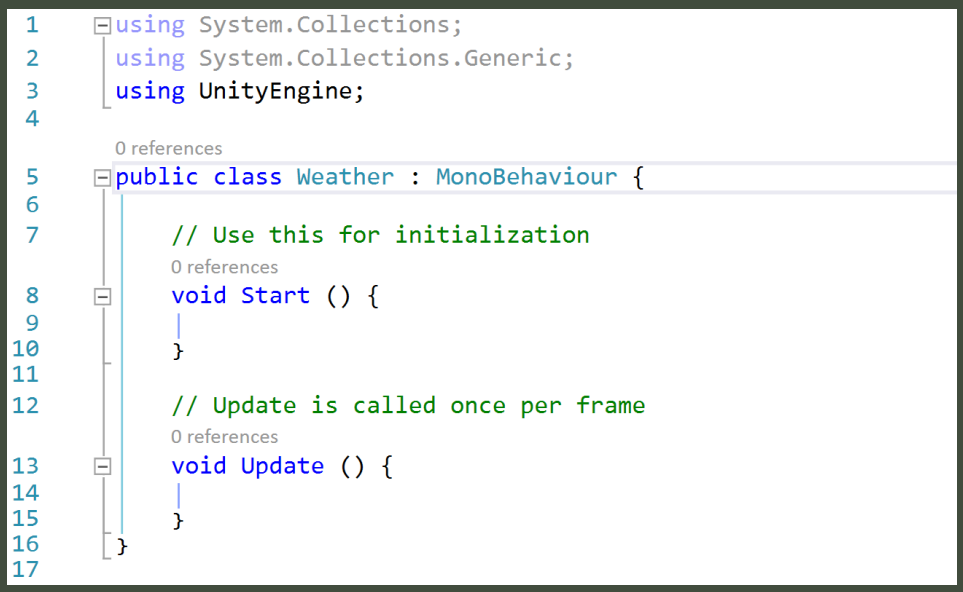
@jeffreymckenzie



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

0 references
public class Weather : MonoBehaviour {

    // Use this for initialization
    0 references
    void Start () {

    }

    // Update is called once per frame
    0 references
    void Update () {

    }
}
```

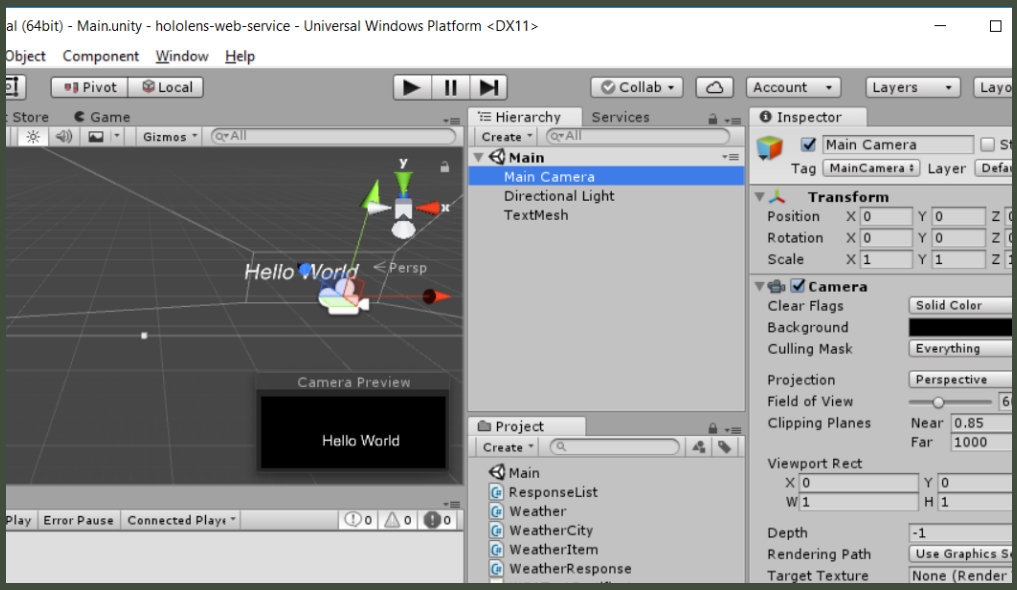
First thing you'll notice at the top
Is a reference to the UnityEngine library –
This is the way to programmatically
Manipulate just about everything in Unity.
Second thing you'll notice is the base class,
MonoBehavior. It's called Mono because
Unity uses the Mono open source .NET framework
To support scripting, and Behavior because
It provides access to the unity object
To alter the way it acts.

@jeffreymckenzie



```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
        0 references
5   public class Weather : MonoBehaviour {
6
7       // Use this for initialization
            0 references
8       void Start () {
9
10      }
11
12      // Update is called once per frame
            0 references
13      void Update () {
14
15      }
16  }
17
```

Every script attached to an object
needs to inherit from MonoBehavior.
Then we have a couple of methods to start from.
Once is called start, which is called
When the object is instantiated, so here's
Where you would set any default values.
Then we have Update, which the comment says
Is set once per frame, so anything that needs to change
Periodically, such as movement or position,
Can be set here.



OK, we're going to skip ahead a bit
And take a look at the code
That runs this app.
If you look down at the project tab
You can see we have 5 different scripts now.
Let's revisit the Weather Script.

@jeffreymckenzie

```
25      void Start()
26      {
27          _textMesh = GetComponent<TextMesh>();
28
29          string url = "http://api.openweathermap.org/data/2.5/forecast?id=5177396&APPID=
30          WWW www = new WWW(url);
31          StartCoroutine(WaitForRequest(www));
32      }
        1 reference
33      private IEnumerator WaitForRequest(WWW www)
34      {
35          yield return www;
36
37          if (String.IsNullOrEmpty(www.error))
38          {
39              WeatherResponse response = JsonUtility.FromJson<WeatherResponse>(www.text);
40              _textMesh.text = response.list[0].weather[0].description;
41          }
42          else
43          {
44              Debug.Log("WWW Error: " + www.error);
45          }
46
47      }
```

In the start method, first I'm getting the instance
Of the 3D text and assigning it to my class member variable.
To do that we use the GetComponent method.
Then we set a local variable called URL
To the REST request.
Next line creates a WWW object –
This provides simple access to a web response.
So we pass the URL as an argument.

@jeffreymckenzie

```
25      void Start()
26      {
27          _textMesh = GetComponent<TextMesh>();
28
29          string url = "http://api.openweathermap.org/data/2.5/forecast?id=5177396&APPID=
30          WWW www = new WWW(url);
31          StartCoroutine(WaitForRequest(www));
32      }
        1 reference
33      private IEnumerator WaitForRequest(WWW www)
34      {
35          yield return www;
36
37          if (String.IsNullOrEmpty(www.error))
38          {
39              WeatherResponse response = JsonUtility.FromJson<WeatherResponse>(www.text);
40              _textMesh.text = response.list[0].weather[0].description;
41          }
42          else
43          {
44              Debug.Log("WWW Error: " + www.error);
45          }
46
47      }
```

Once we have the WWW object,
We can pass it to what's called a CoRoutine,
Which is a way in Unity
To process something out of band, without
Freezing the main UI.
If we didn't use a CoRoutine here,
The Hololens would essentially freeze
Until we got a response back from the web.
So WaitForRequest is our CoRoutine.

@jeffreymckenzie

```
25      void Start()
26      {
27          _textMesh = GetComponent<TextMesh>();
28
29          string url = "http://api.openweathermap.org/data/2.5/forecast?id=5177396&APPID=
30          WWW www = new WWW(url);
31          StartCoroutine(WaitForRequest(www));
32      }
        1 reference
33      private IEnumerator WaitForRequest(WWW www)
34      {
35          yield return www;
36
37          if (String.IsNullOrEmpty(www.error))
38          {
39              WeatherResponse response = JsonUtility.FromJson<WeatherResponse>(www.text);
40              _textMesh.text = response.list[0].weather[0].description;
41          }
42          else
43          {
44              Debug.Log("WWW Error: " + www.error);
45          }
46
47      }
```

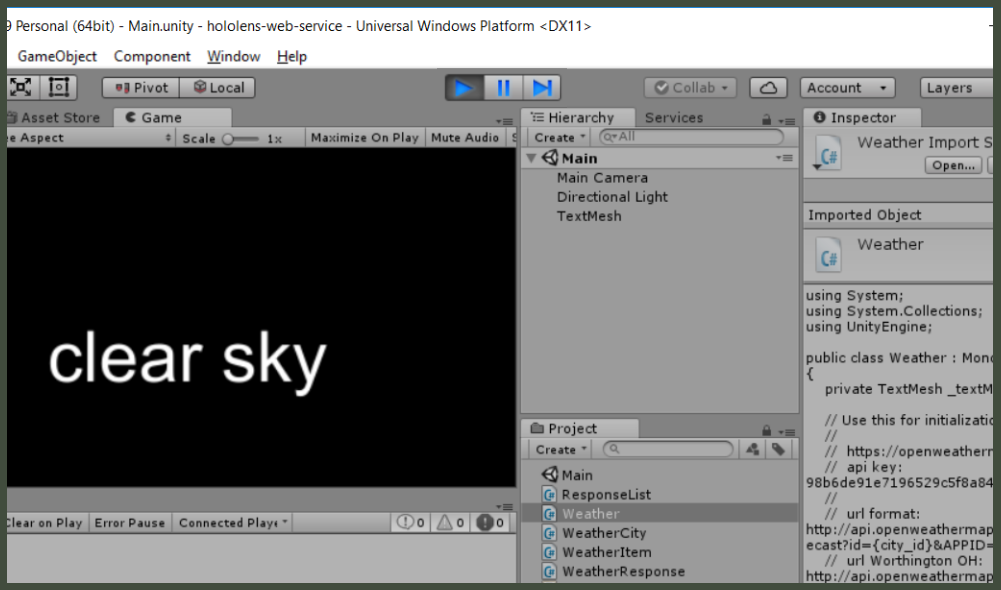
So in the WaitForRequest method,
By using the yield keyword on the WWW object,
We signal that we don't want to return
to the main method until the response
has been downloaded.
When we get a response back,
We can capture that in the Weather Response class.

@jeffreymckenzie

```
25      void Start()
26      {
27          _textMesh = GetComponent<TextMesh>();
28
29          string url = "http://api.openweathermap.org/data/2.5/forecast?id=5177396&APPID=
30          WWW www = new WWW(url);
31          StartCoroutine(WaitForRequest(www));
32      }
        1 reference
33      private IEnumerator WaitForRequest(WWW www)
34      {
35          yield return www;
36
37          if (String.IsNullOrEmpty(www.error))
38          {
39              WeatherResponse response = JsonUtility.FromJson<WeatherResponse>(www.text);
40              _textMesh.text = response.list[0].weather[0].description;
41          }
42          else
43          {
44              Debug.Log("WWW Error: " + www.error);
45          }
46
47      }
```

We can do that through the Unity method JsonUtility,
Which transforms the text from the web response
Into an object.
Once we have object, we can drill down into
The specific value that we want,
In this case the weather description.
Then we set the text property of the 3D text
To the weather description.

@jeffreymckenzie



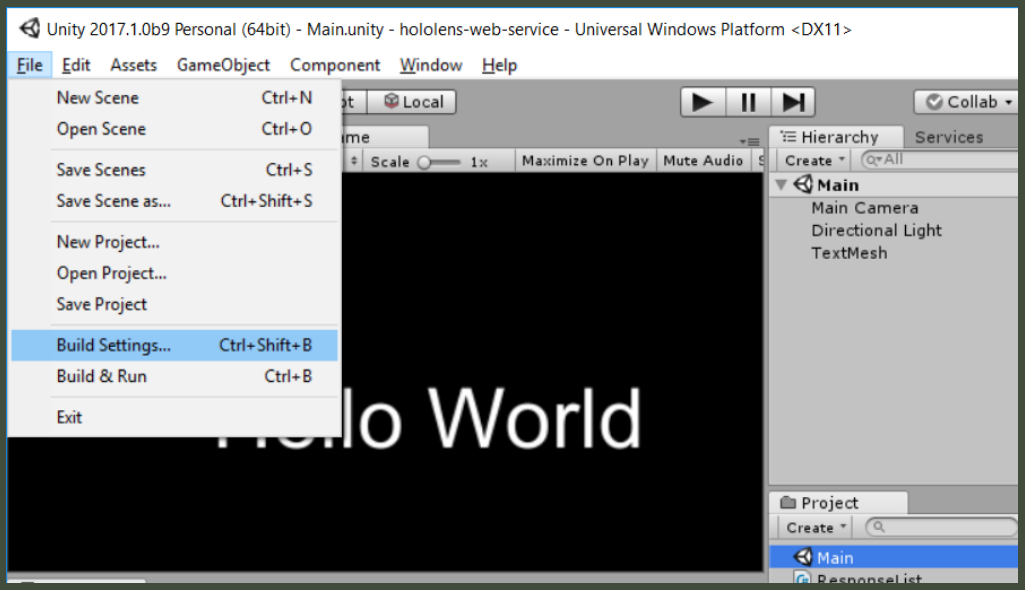
We can test this out in Unity
By clicking the Play button at the top –

This runs the project and displays the output in the game tab.

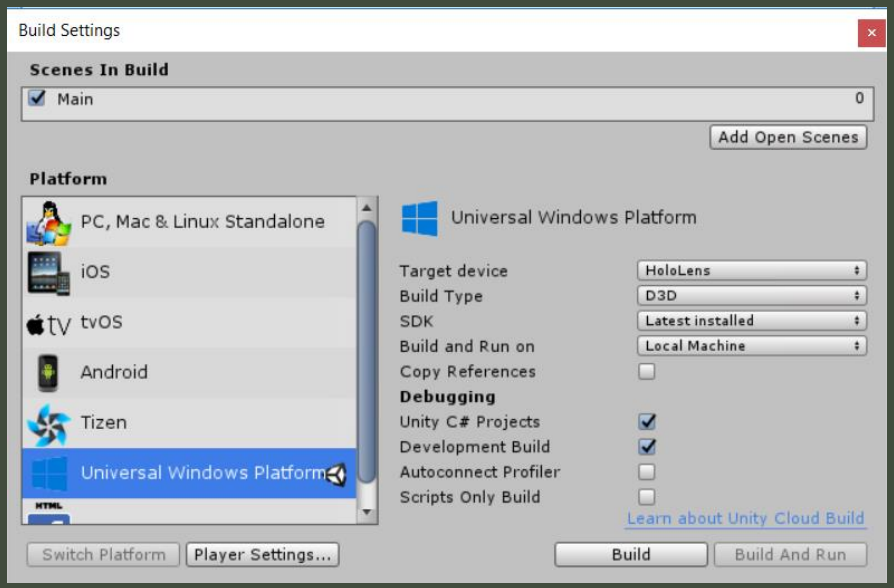
And here we have the weather returned from the service.

Now that we have the project working,
We can build it to run in HoloLens.

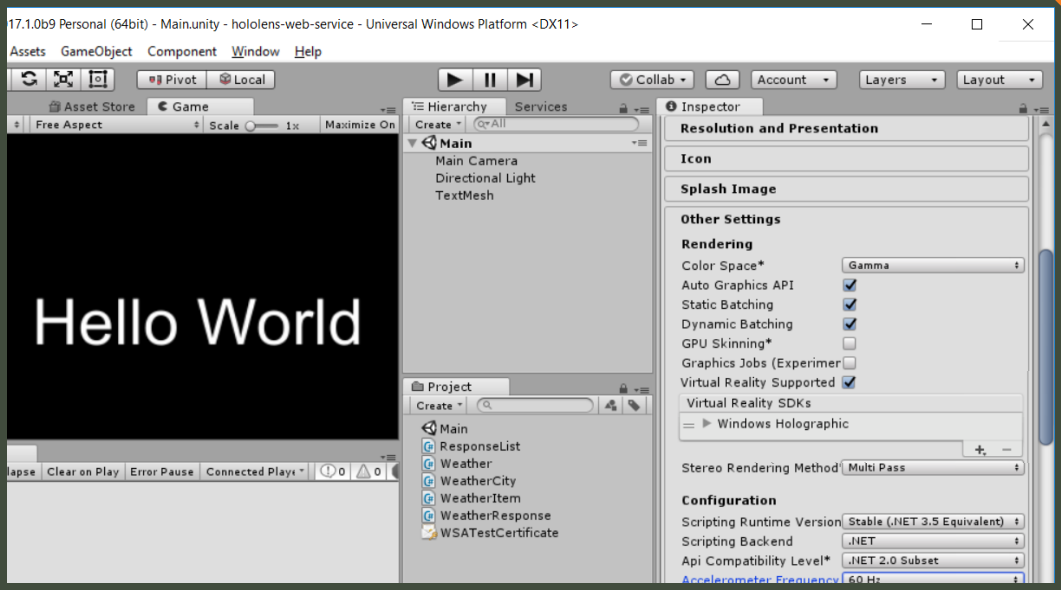
@jeffreymckenzie



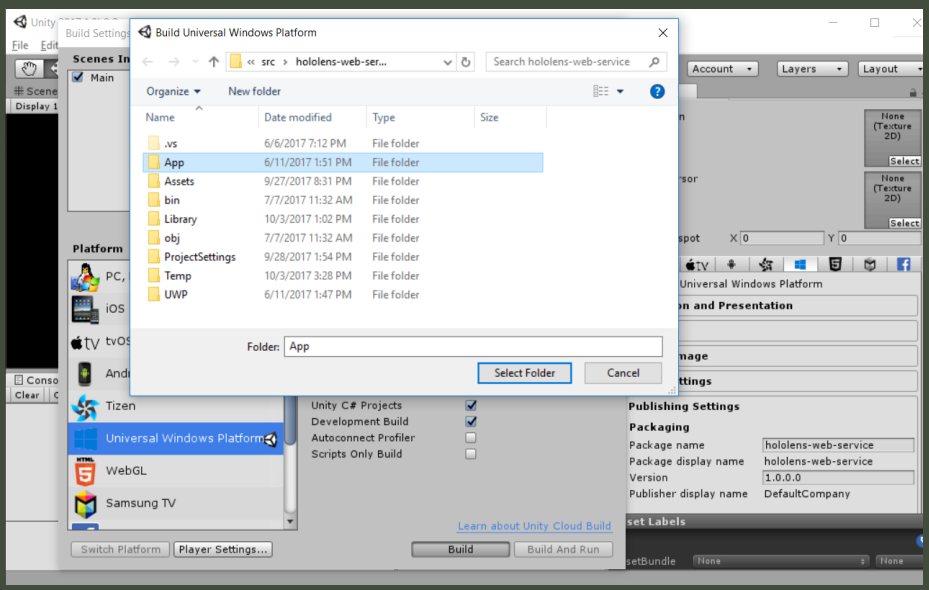
To do that, we go up to File, and Build Settings…



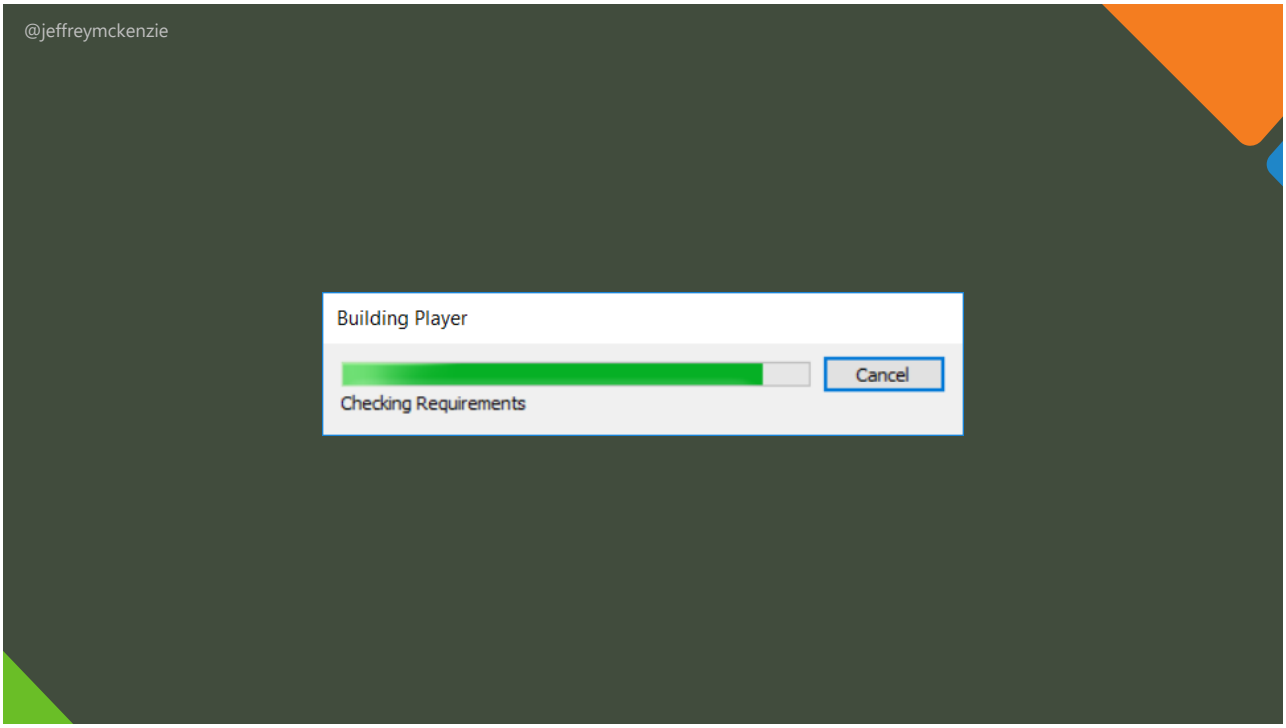
That brings up the Build Settings dialog box.
Here we want to select Universal Windows Platform
And specify some of the settings on the right hand side –
Our target device is Hololens,
Build type is direct 3D,
We can also specify the SDK version we want to target.
The debugging settings allow you
To work with Visual Studio when running
And debugging the application.
We also need to change some player settings,
Using the button at the bottom left.



This brings up the player settings
In the inspector tab.
Here you can set player settings for
A particular platform – here
It's Universal Windows Platform
On the bottom right you'll want to enable
Virtual Reality Support,
And select the Windows Holographic SDK.
This allows the Hololens to account
For head and spatial tracking in the application.

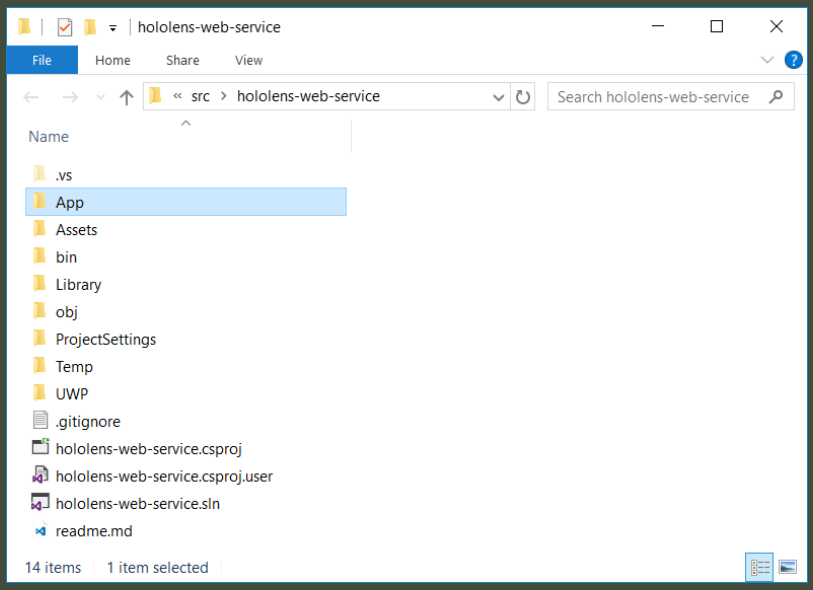


Once we click on Build,
Unity brings up the folder view for the project,
And prompts you for a folder to build into.
The App folder is selected by default,
So we will go ahead and build into that.



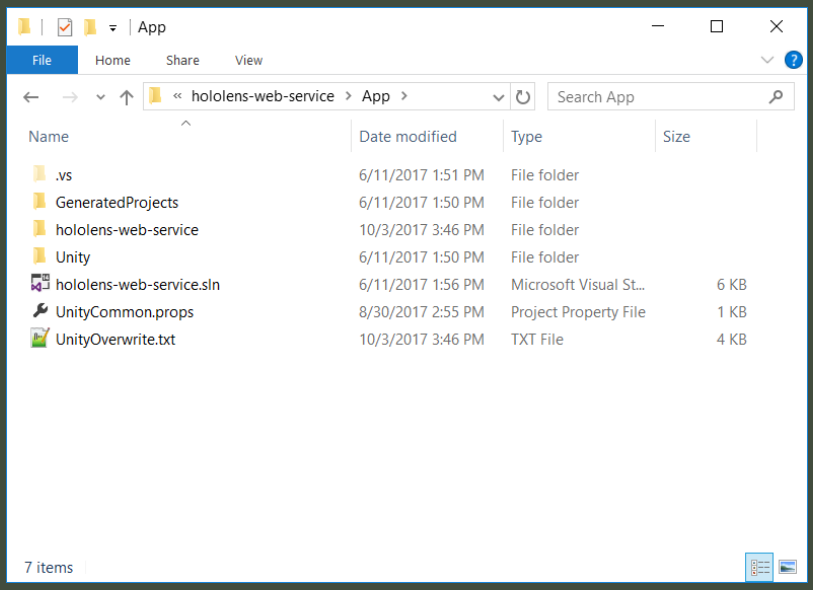
So Unity goes through the build process….

@jeffreymckenzie

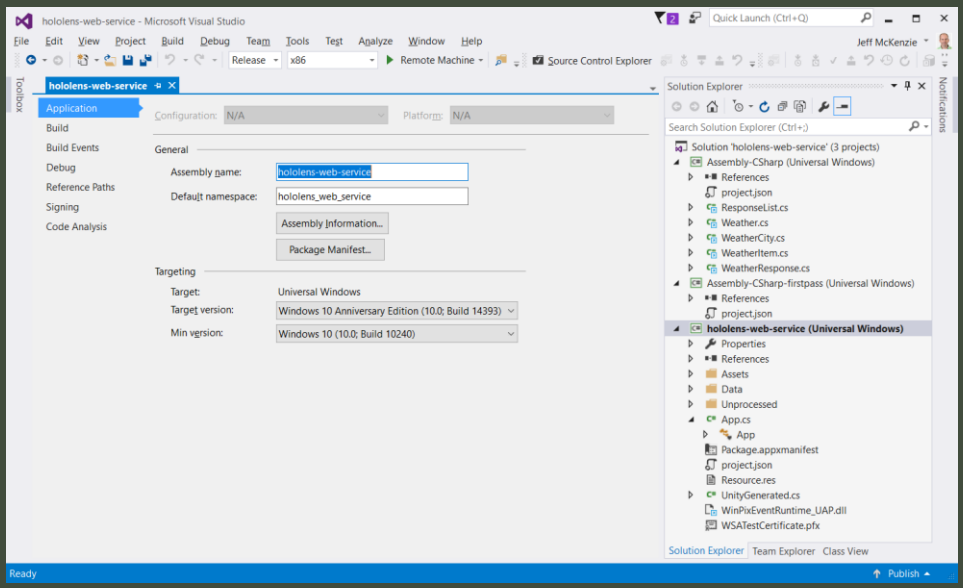


Then it displays the project in Windows explorer.
So we can drill down into that folder…

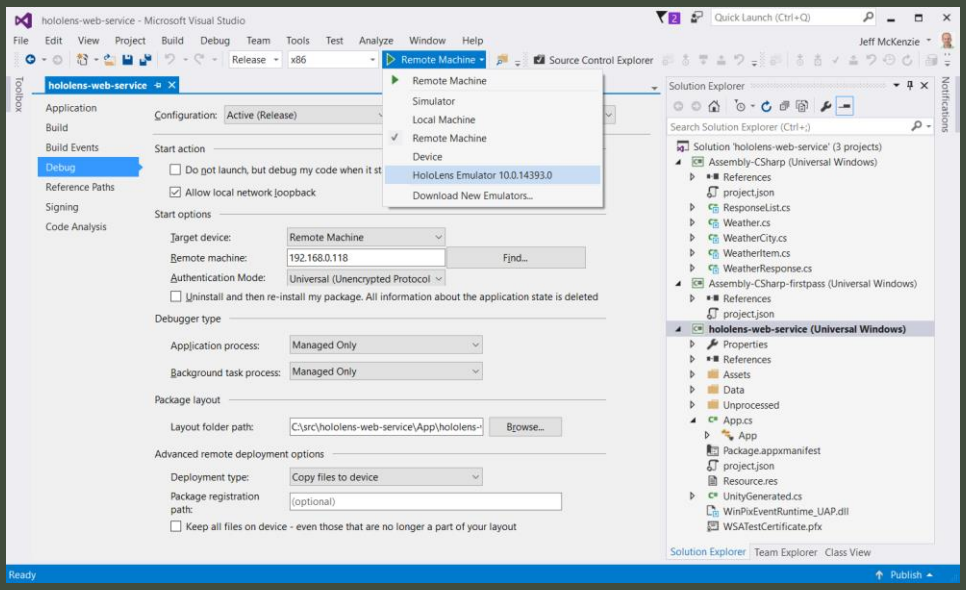
@jeffreymckenzie



And we get a view of the Visual Studio solution
Generated by Unity.
Lets open that solution…

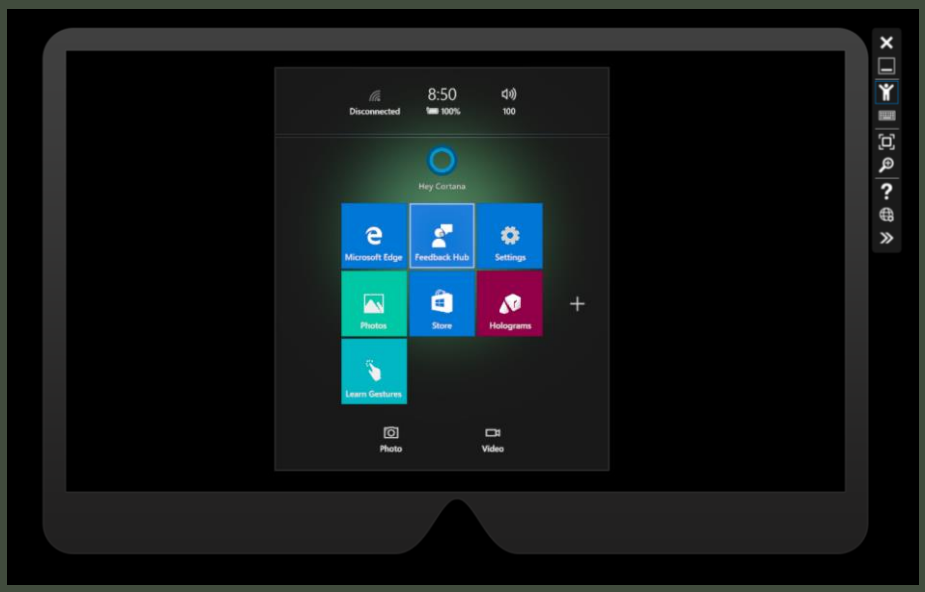


In the solution explorer view, you can get an idea
Of what the built app looks like –
There's one project with our source code,
Another project with code references,
And the startup project which is what actually
Gets run on the device.



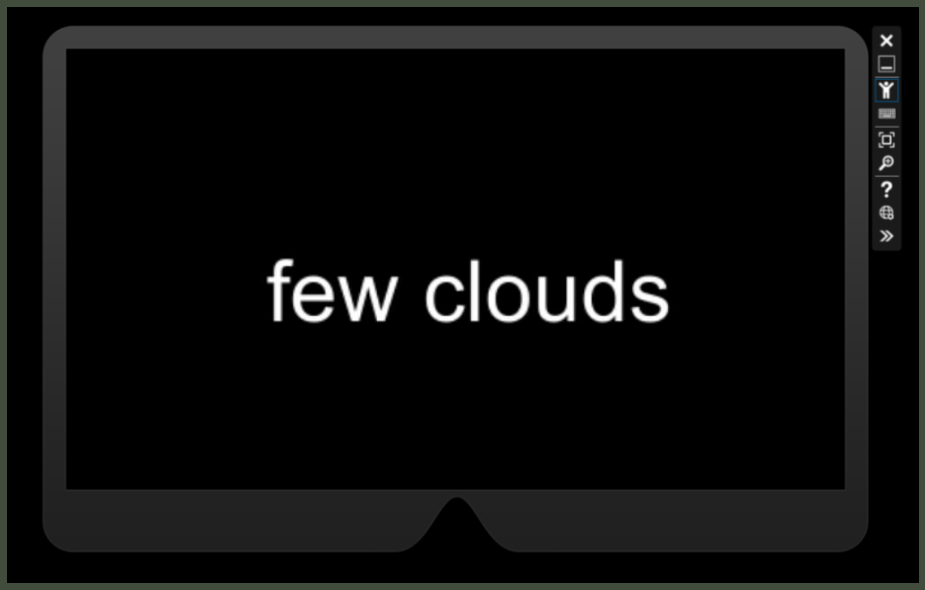
Now to run this on the emulator,
We need to make sure that it's set to
Release configuration for x86,
and change Our target
to the HoloLens Emulator,
Which is in the drop down list there at the top.

@jeffreymckenzie



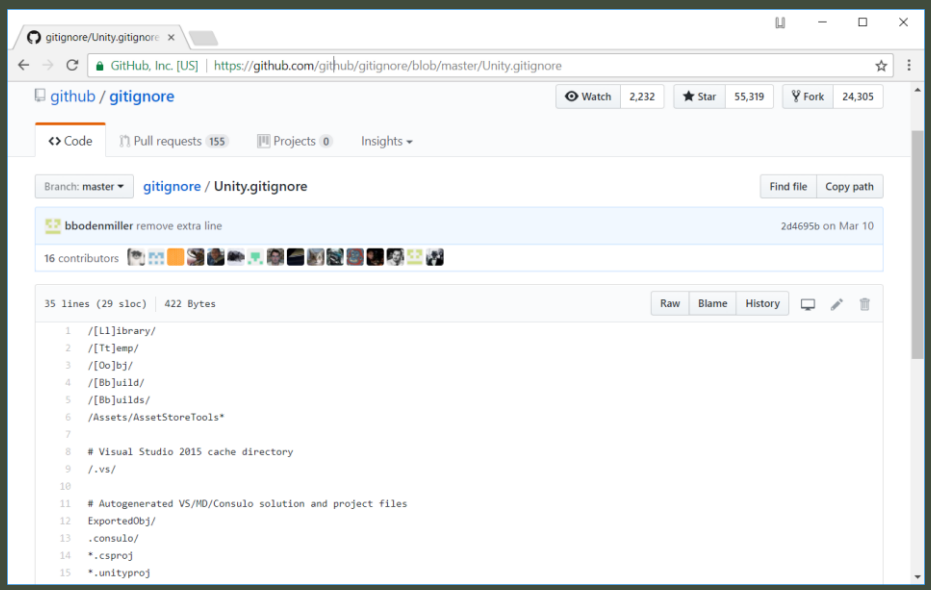
Then while the app is deploying to the emulator,
The main menu appears –
This is how you interact with the hololens interface,
Using gestures or voice commands
To select applications.

Then we have the splash screen of the app –
The free, personal edition of Unity adds this splash screen
To every app that’s built.



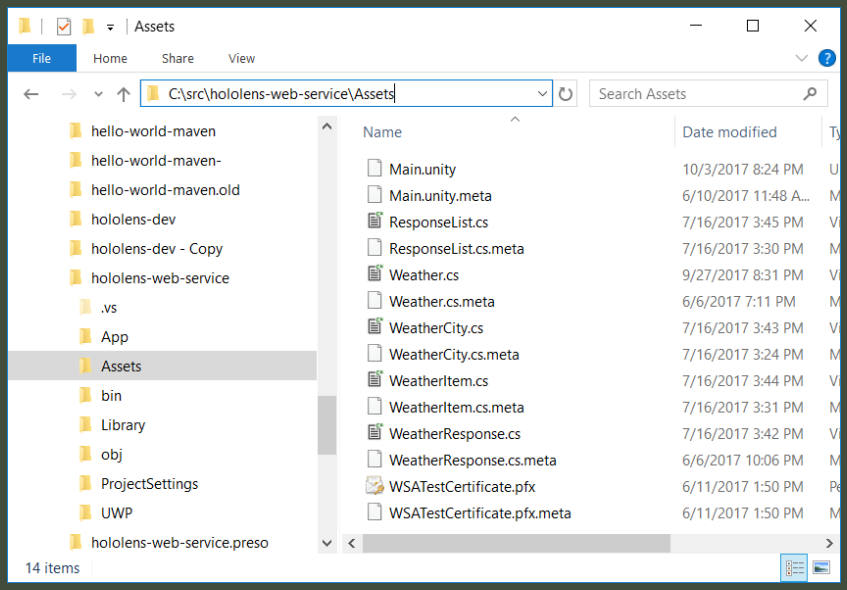
And finally our app launches,
Goes out to openweather.org,
Grabs the current weather, and displays it
In the hololens emulator,
Just like it does in the hololens itself.

@jeffreymckenzie



Ok, so one more thing I'd like to discuss,
And that's version control.
One nice thing about a Unity project
Is you don't have to store a lot
Of files in version control.
First thing to get is an ignore list --
Here's an example of a Unity ignore file
From GitHub.
This contains all the files, folders, and patterns
You don't need to keep in version control.
You can just google Unity Ignore and that will bring up
A lot of other examples.

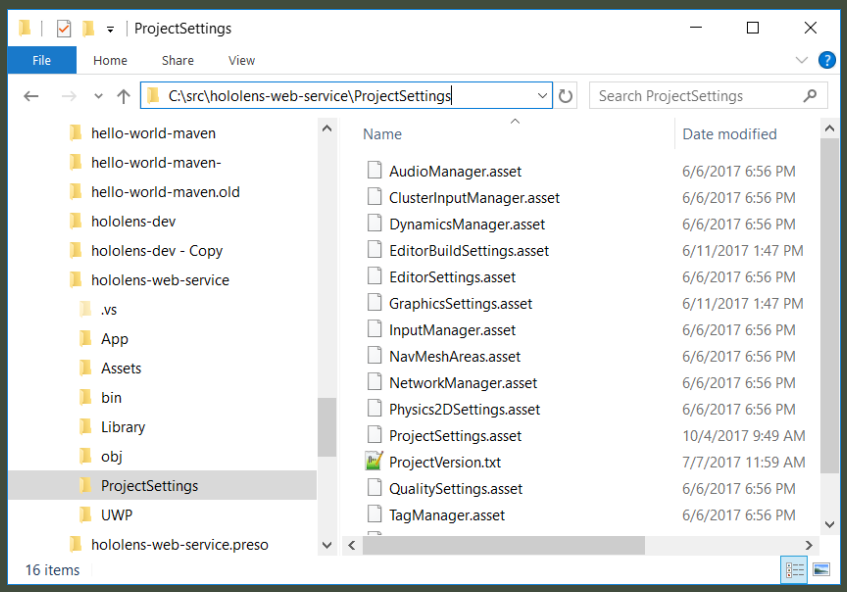
@jeffreymckenzie



If we take a look at the directory structure,
There are very few folders we need to preserve.

One is the Assets folder – as you can see on the right,
This contains our scene and the scripts we used.

@jeffreymckenzie



Next is the project settings folder –
This contains our project version file,
As well as the asset files we use in the project,
Which store project-specific settings.
And that's it.

Everything else gets regenerated by Unity.

So thank you for listening & for being here.

So does anybody have any questions about --

The HoloLens

Or Unity?