HoloLens 2.0 Virtual Human Tutorial

Adarsh Viju Kumar -- **a.vijukumar@aston.ac.uk**

Contents

Outline

In this tutorial we will first go through familiarization of HoloLens 2.0 and then moves to creating a virtual human in Unity and configuring Unity app for visualizing the virtual human through HoloLens 2.0.

HoloLens 2.0

Microsoft HoloLens 2.0 is a Microsoft-developed and produced augmented reality (AR) headset. It is the successor to Microsoft's first HoloLens. The HoloLens 2.0 are stereoscopic and full-color mixed reality smartglasses developed and produced collaboratively by Microsoft and MicroVision, Inc. It's a self-contained device with an on-board computer and built-in Wi-Fi that functions as a client for data stored on the device, the Azure cloud, or the internet in general. It's an almost entirely different device in comparison to its predecessor, with higher computer power, better sensors, longer battery life, and a few other features worth mentioning. It has a larger FOV (field of view), allowing the user to see more information and making the device more appealing to use with applications that display a lot of information.

There are whole new, more natural hand movements that allow for simpler content management. Any command might be a mix of the user's hand movement and vocal control. The gadget is lighter and smaller than the previous generation, which is the most noticeable improvement over the previous model. This makes it considerably more pleasant to wear and work with. We may expect that this will allow for extended regular usage times.
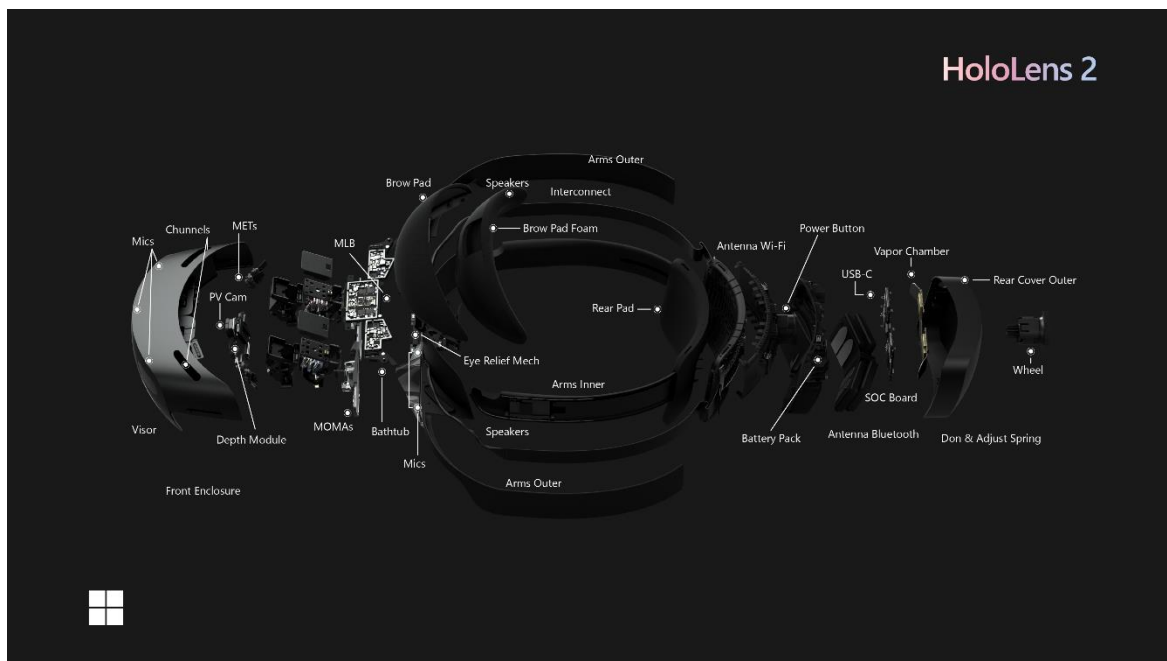
Figure 1: Using Hololens 2.0 (Source: https://www.be-terna.com)

Figure 2: Sensor aggregation and System Integration (Source: https://learn.microsoft.com/)

Virtual Humans

Virtual humans (or digital humans) are computer simulations of human beings. Their depiction, mobility, and behaviour are all investigated in this arena. They have several applications in simulation, gaming, film industry, training, and a variety of other sectors. Recent advances in artificial intelligence (AI) and mixed reality (MR) hardware have resulted in an exponential increase in the production of high-fidelity virtual human avatars with whom humans can interact with. In this tutorial, we will use Unity software to develop humanoid avatars, provide basic movement animations, and learn how we can visualise them using the Hololens 2.0.

Figure 3: A graphical demonstration of virtual humans in mixed reality environment. (Source: https://www.virtualrealityexps.com/)

Creating Virtual Humans (VH) in Unity

Unity is a cross-platform game engine developed by Unity Technologies. The engine has been continuously expanded to accommodate a wide range of PC, mobile, console, and virtual/augmented reality applications. It is mainly used to develop 3D and 2D games, as well as interactive simulations and other extended reality experiences.

Let's go through how to design and animate a virtual person in Unity software and then visualize it in a mixed reality environment using HoloLens 2.0 step by step.

If you are new to unity, please see https://unity.com/download to download and install unity. Also, you can find the official documentation on Unity at https://docs.unity3d.com/Manual/index.html

First, create a new unity project by opening the Unity Hub and clicking the *New Project* button as shown in figure 4. After that, in the next window, select *3D* and give a name to your project. Here is 'Virtual_Human'. Then press *Create Project* button. See Figure 4.
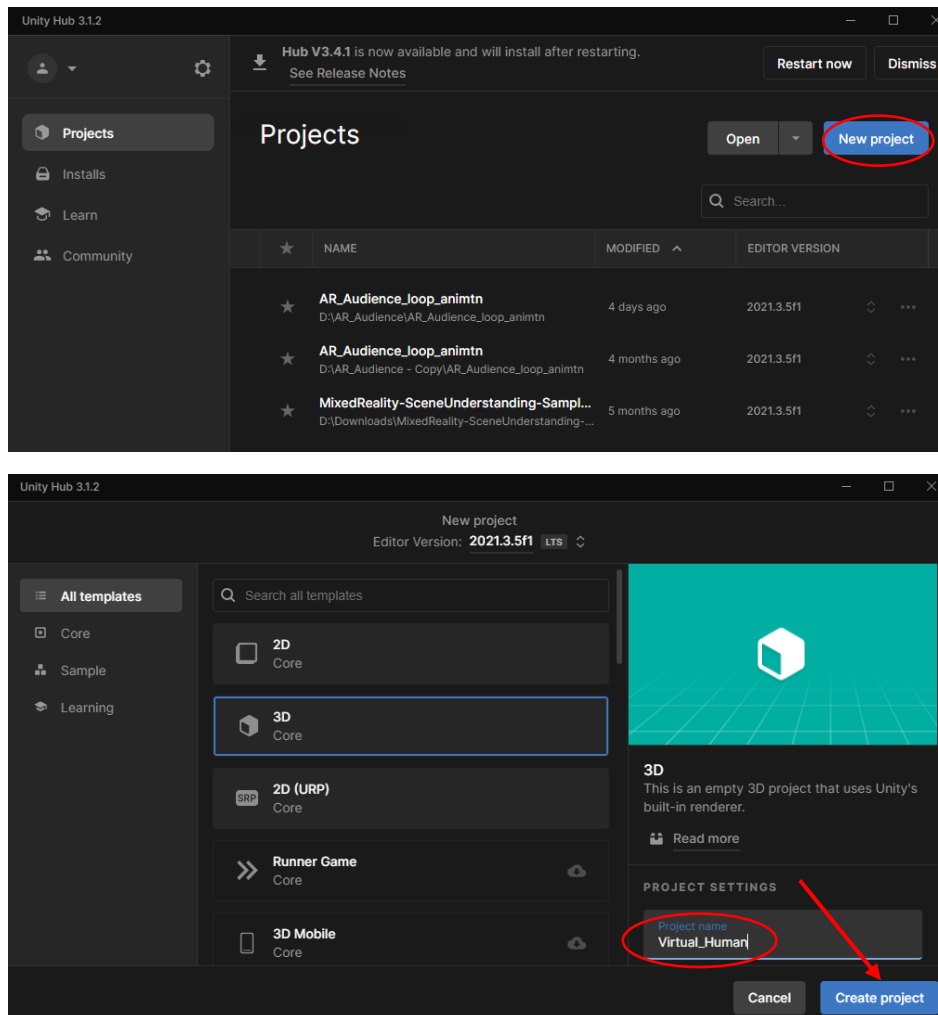

Figure 4: Creating a new project

Once you have done this, the unity development window opens for you. There will be a scene with two objects, Main Camera and Directional Light, in it.
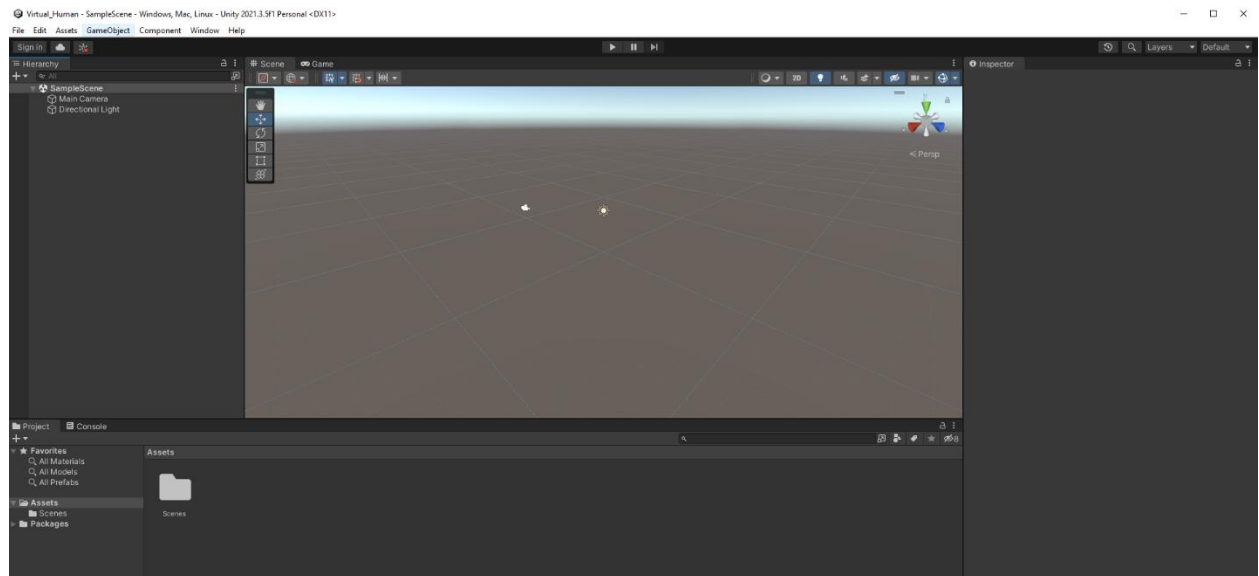

Figure 5: Unity development window

Now we have to download the virtual human character and some animation effects. We can get virtual characters for unity from various online sources. Here we will be downloading the characters from https://www.mixamo.com/#/ . First create an account in Mixamo and select a character from their characters list.
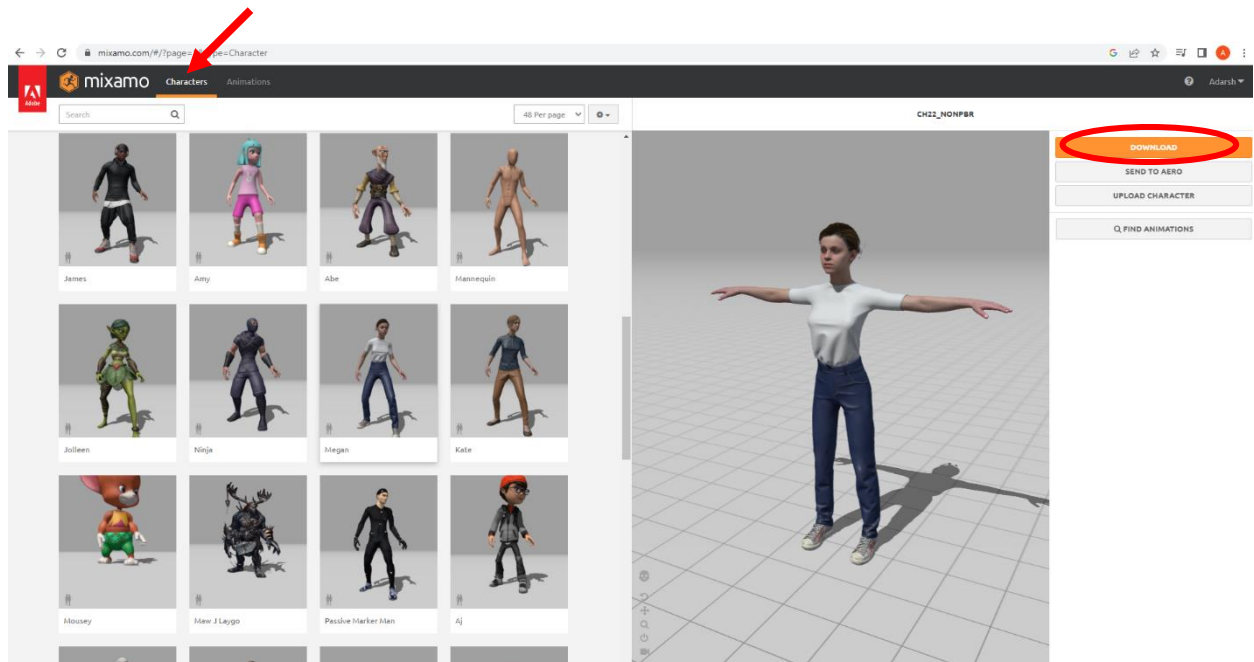
Figure 6: Downloading virtual human character from Mixamo

Once you click download, the following dialog box will be shown. In the format option, select FBX for Unity(.fbx) and click download.
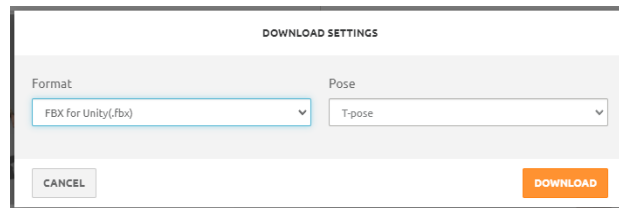

Figure 7: Choosing format for downloading virtual character

Now we have to download an animation effect to the character. Click on the animations tab on top of the Mixamo web page. Then choose an appropriate animation effect. Here we will be using a talking animation. We can change some of the animation parameters here itself. Finally, click on the download button.
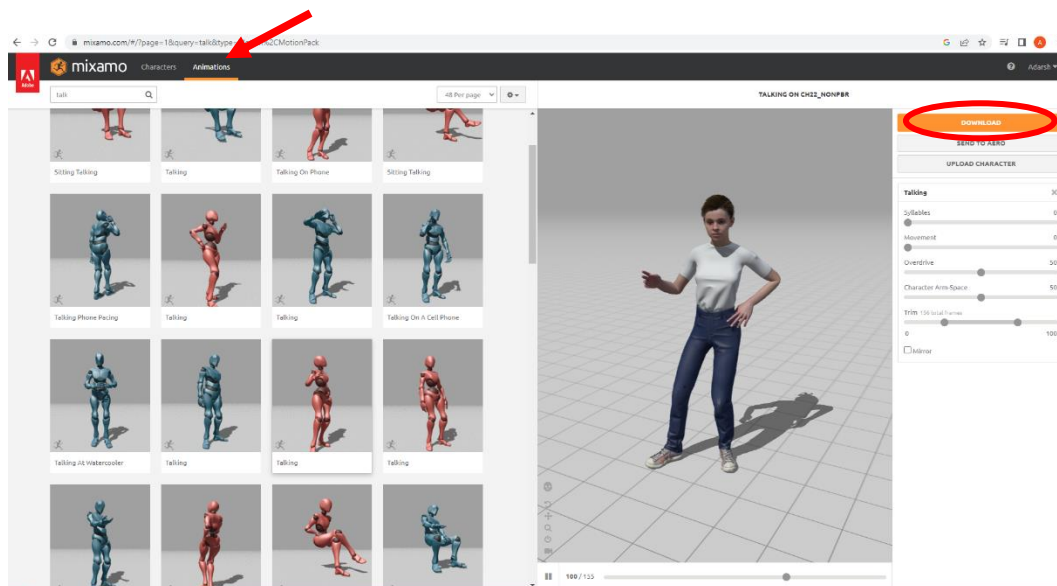

Figure 8: Downloading animation effect from Mixamo

Now a dialog box will appear. In the format option choose FBX for Unity(.fbx) and select 'without skin' for the skin option. Select the default options for others.
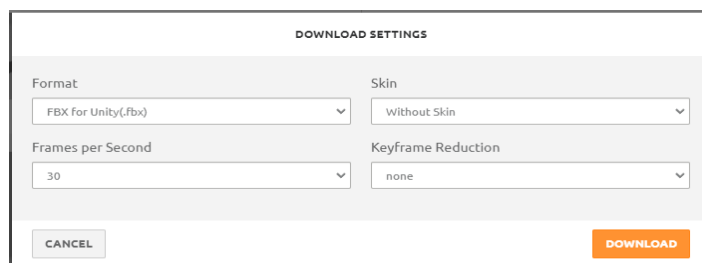

Figure 9: Selecting format and skin for animation effect

Now coming back to unity, in the project hierarchy at the bottom, create two new folders by right clicking the assets option and selecting *create>>folder.* Name one folder as character and the other as animation.
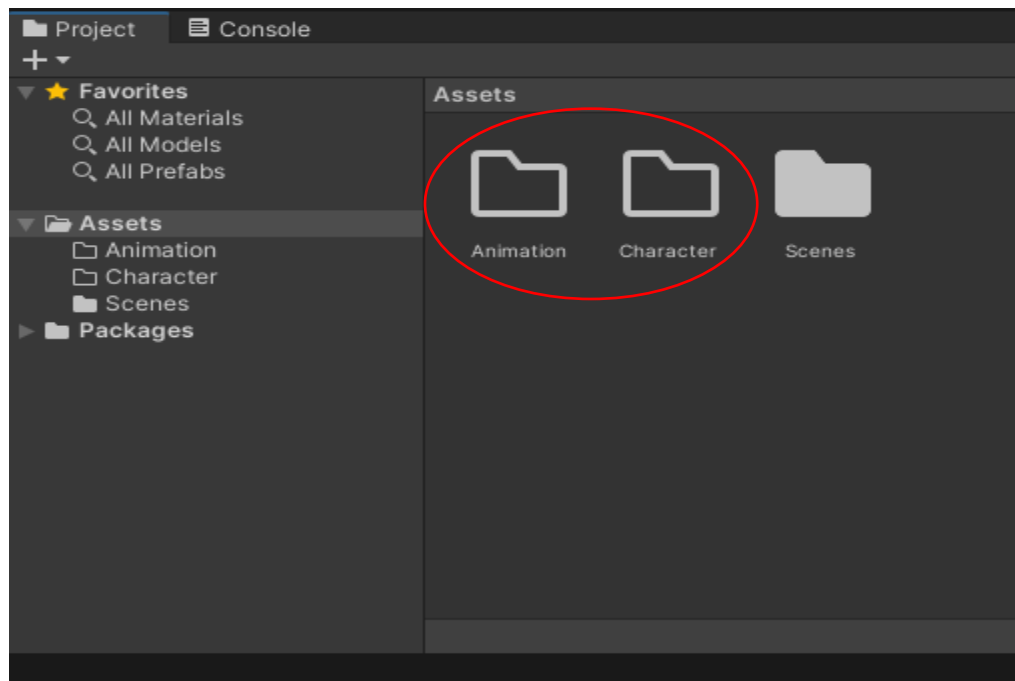

Figure 10: Creating folders for character and animation

Now, move the downloaded character file to the Character folder and the animation file to the Animation folder. You can simply drag and drop the files to these folders.

Next, we must bring our character to the scene. For this, simply drag and drop the file from the character folder to the scene and position the character in front of the main camera. We can adjust the position and orientation using the editing tools on top of the scene hierarchy (pointed by red arrow). In some versions, these tools will be shown there in the scene itself.
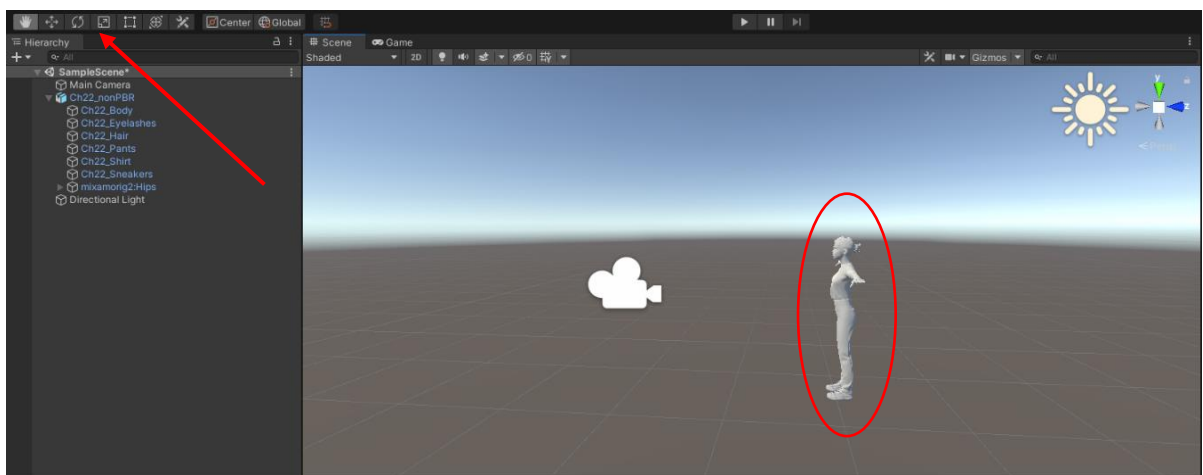

Figure 11: Inserting and adjusting orientation of character in the scene

At this point, the character will be just shown as a white body without any textures or colours. For getting the textures and other missing properties we have to extract the materials and textures. Inorder to do this, click on the character file, then an inspector window will be open on the right side. Select the materials option and click the Extract textures and Extract materials buttons. Select the folder for saving the texture and material files. You can either create separate folders for this or it will be saved in the same location of the character file.
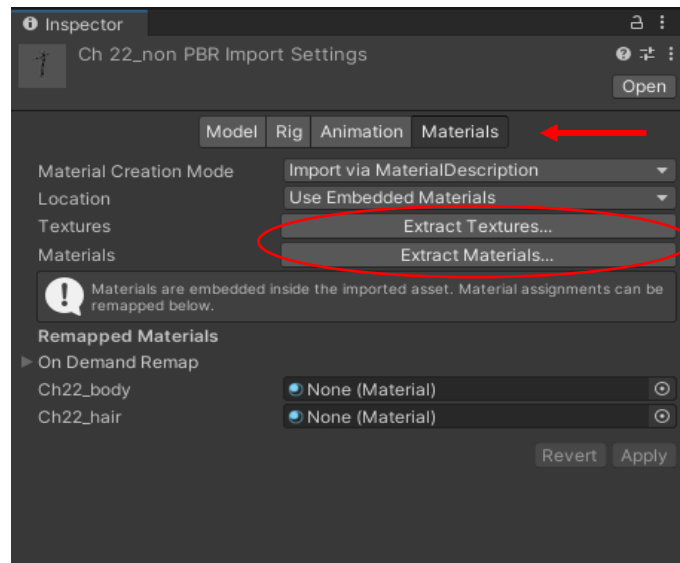
Figure 12: Extracting materials and textures of the character

Now, if we look into the scene, we can see that the character appears to be normal. And, we can see the extracted properties in the character folder. We can change the parameters of these properties by selecting it.
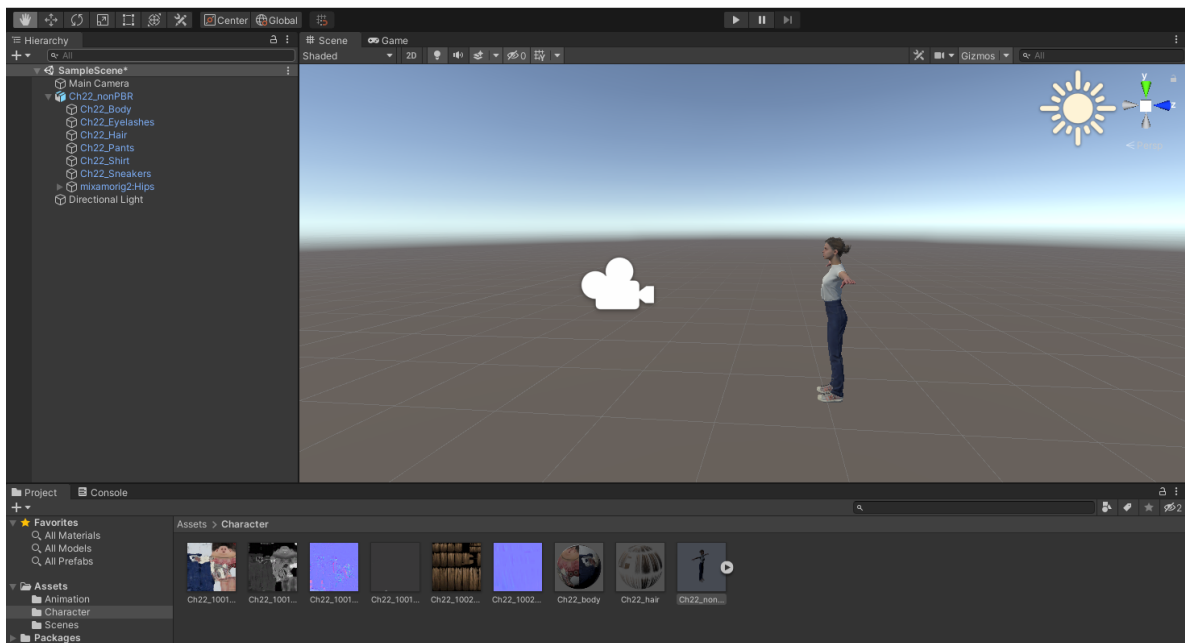

Figure 13: Character after extracting texture and materials

Then we have to create an avatar definition for the character. For this, again click on the character file icon. In the inspector menu, click on *'Rig'* option. Then for the Animation Type, choose *'Humanoid'* and for the avatar definition, select '*Create From This Model'* and click *Apply.*
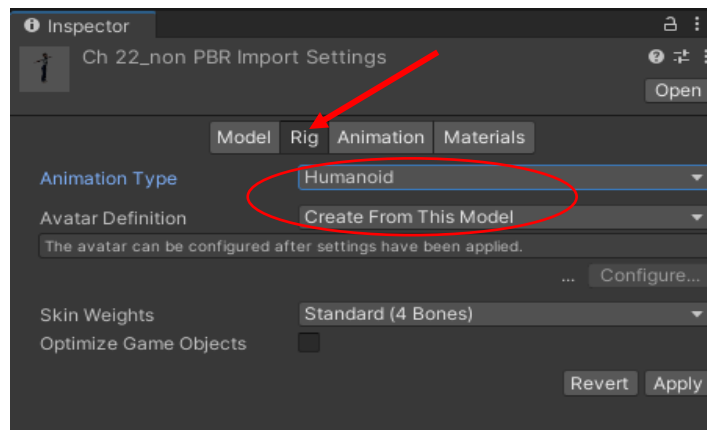

Figure 14: Setting the avatar definition for character

After this, we have to add this avatar definition to the animation. Click on the animation file icon, go to the inspector menu, select *'Rig'* option. Under this, select the Animation Type as '*Humanoid*' and for the Avatar Definition, choose '*Copy from other avatar*'. Now we must choose the source. Click on the dot on right, then it will show the available avatars. Select the avatar of our character.
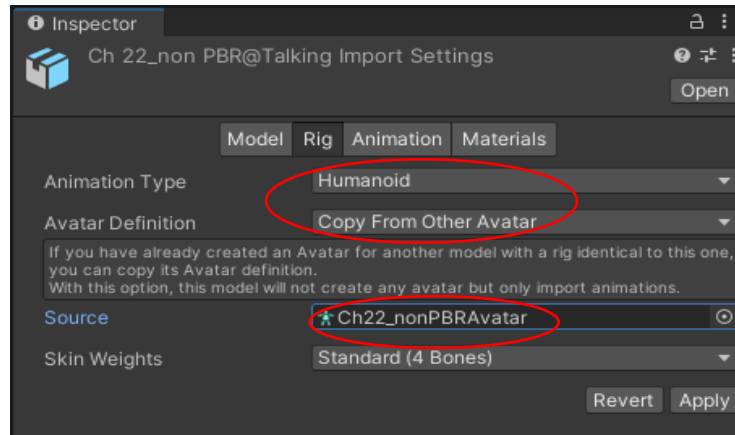

Figure 15: Setting the avatar definition for animation

As we are using only a single animation effect, in order for getting a continuity in the animation, we can loop it. For this, again go to the inspector menu of the animation file. Select Animation option and check the Loop Time box.
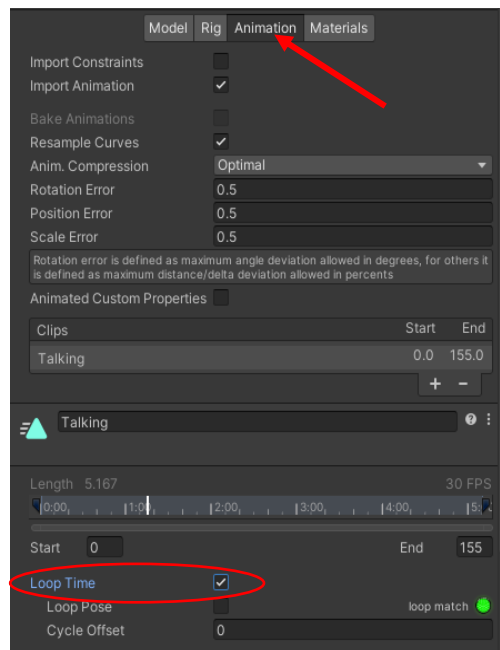

Figure 16: Giving a loop effect to the animation

Now, we need to create an animator controller to give this animation effect to our virtual human. For that, go to the animation folder, or we can create a new folder, then right click and select *create>>Animator Controller,* and name it as *'Animator'.*
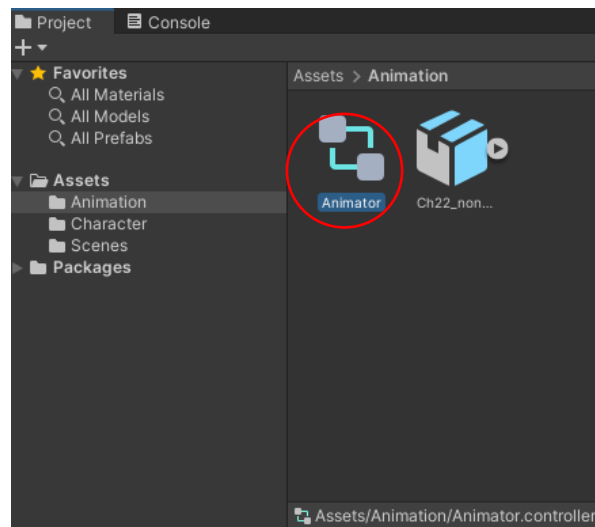

Figure 17: Animator Controller

Double click on the animator icon and a controller editor window will open for you. Drag and drop our animation file to this area and we can see an arrow pointing from the *'Start'* option to the animation. We can add more animation effects like this and connect all to make a loop so that the character goes through different animations and makes it more realistic.
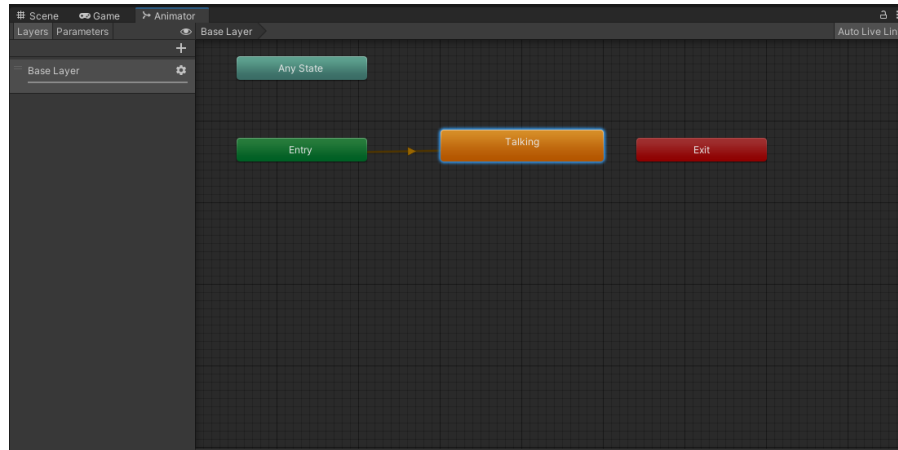

Figure 18: Animator controller window

Now click on the character icon in the scene hierarchy on the left side of the scene, and go to its inspector window. There we can see a field called '*Controller'*. In that field, add our '*Animator*' controller either by dragging and dropping the icon or selecting it from a list by clicking the dot on the right side of the field.
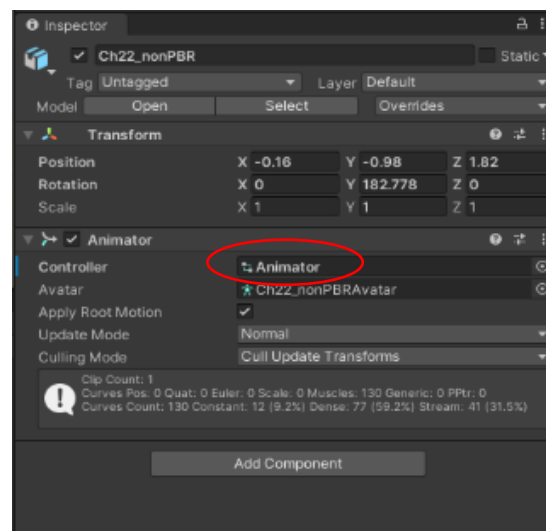

Figure 19: Adding controller to the character.

Now we can check whether our VH character is showing the animation effects. For this, click on the play button on the top of the scene window. Now we can see that the character is exhibiting some motion.
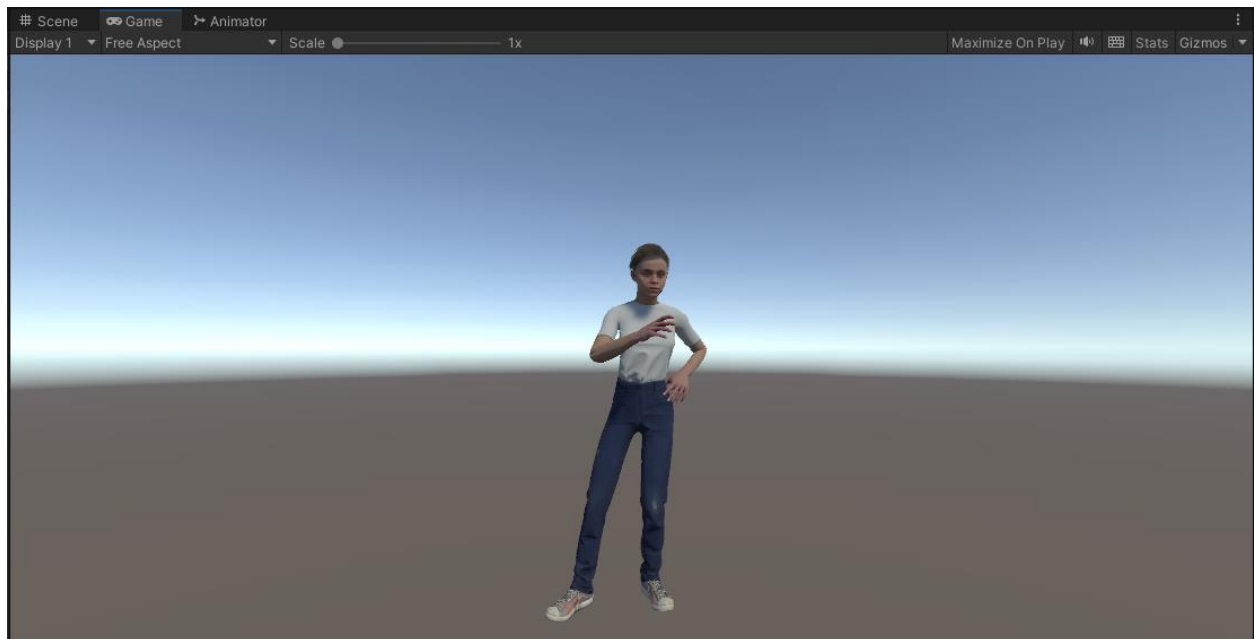

Figure 20: Animated character in the scene

So, we are done with creating a virtual Human in unity. Now we have to set up the configurations for visualizing this VH in front of us, in a mixed reality environment, with the help of HoloLens 2.0.

Configuring Unity for HoloLens 2.0.

To integrate the unity scene in HoloLens, first we have to make some changes in the build settings of the project. Go to File menu and choose build settings. Change the platform to *'Universal Windows platform'* and click on *'Install with Unity Hub'*.
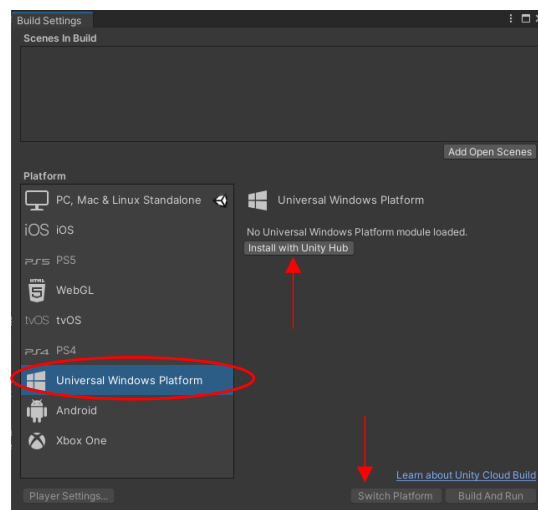

Figure 21: Installing Universal Windows Platform

Once it got installed, click on '*Switch Platform'* button. The Unity will get restarted. Again go to the build settings, select the '*Player Settings'* at the bottom of the window. Then a new window will be opened. From the list on the left panel, choose '*XR Plugin Management'* and click on '*Install XR Plugin Management*'
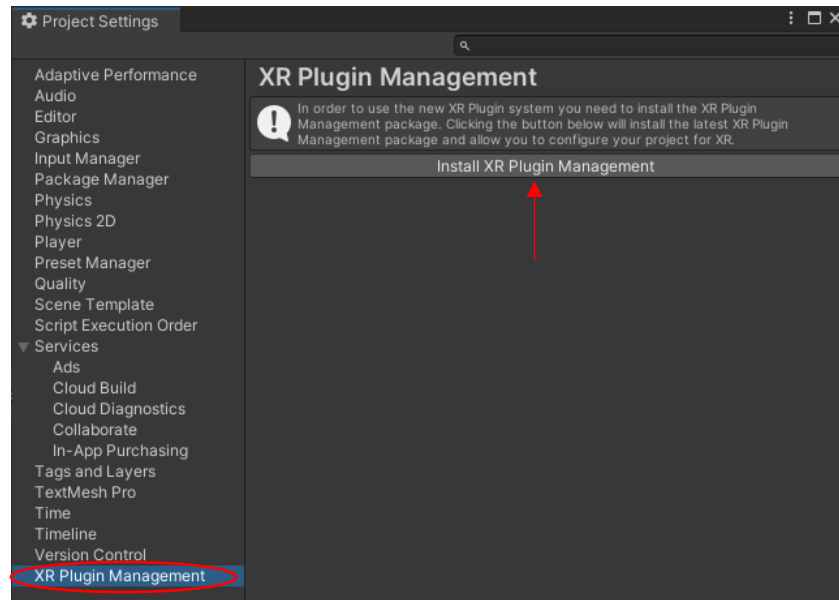

Figure 22: Installing XR Plugin Management

After the installation, the window looks like the following. Click on the '*Open XR*' option. . A dialog box will appear asking for restarting the editor, Click on the '*Yes'* button.
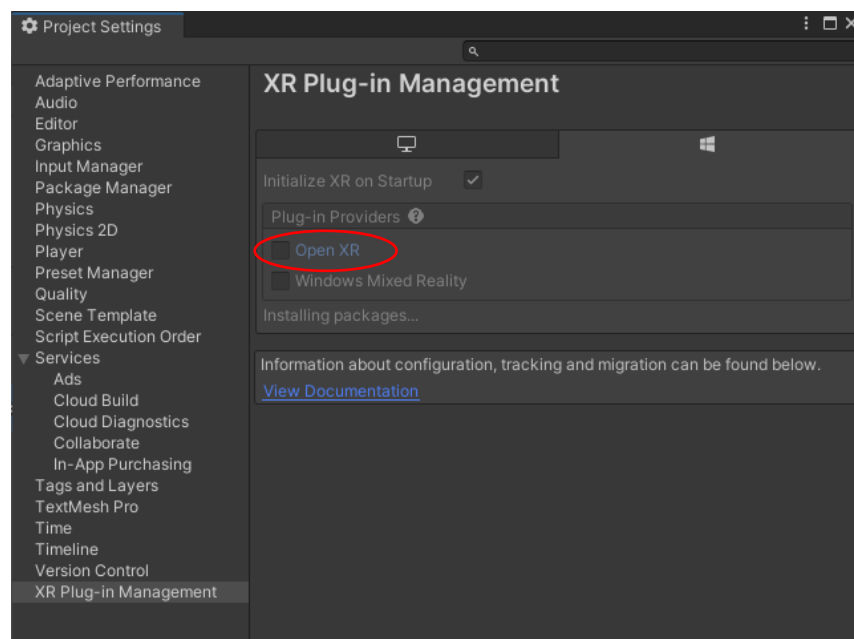

Figure 23: Adding OpenXR plugin

After installing this, Unity may show some warnings and issues. You can click on the '*fix All'*.
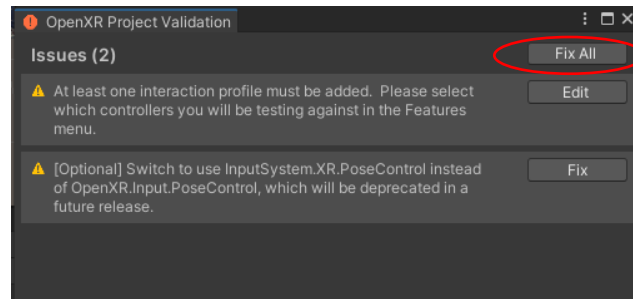

Figure 24: Fixing issues

We are now done with the project settings. Now coming back to the scene, we have to change the main camera to XR Rig. This will convert the main camera to align with the motion and orientation of HoloLens. For this, right click on the '*Main Camera'* option in the screen hierarchy, click *XR >> Convert Main Camera to XR Rig.* Now if we check the scene hierarchy, we can see that the main camera is now a sub component under the XRRig.
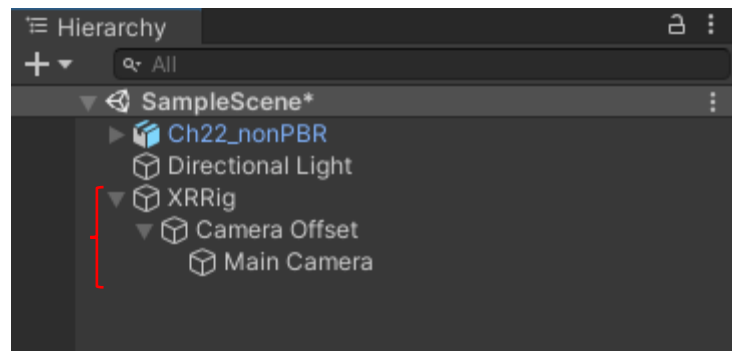

Figure 25: Main camera after converted to XR Rig

Next thing we have to do is to install the 'Mixed Reality Feature Tool for Unity'(MRFT). It can be downloaded from https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unity/welcome-to-mr-feature-tool. You can also go through its documentation as well, in this website.
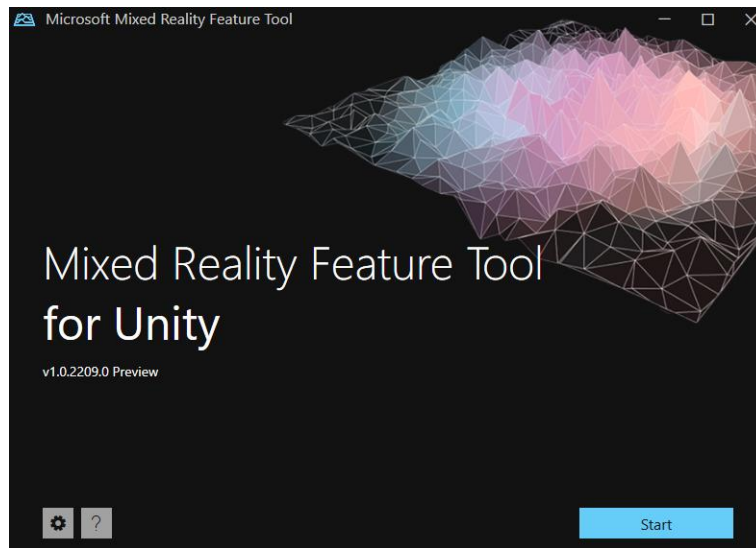
After downloading, open it and click on start.

Figure 26: Mixed reality feature tool for unity

In the next window, select the folder path, then click on the settings icon and check the box '*Overwrite existing package files*'. Now click on discover features.
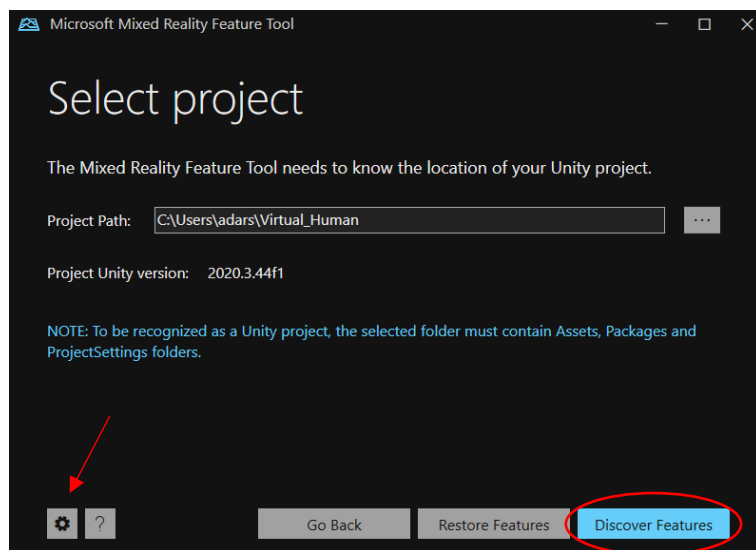

Figure 27: Adding project path in MRFT

The next window will display a list of features. From that select the *'Platform Support'* option and check the *'Mixed Reality OpenXR Plugin' box.* Now click on *'Get Features'*.
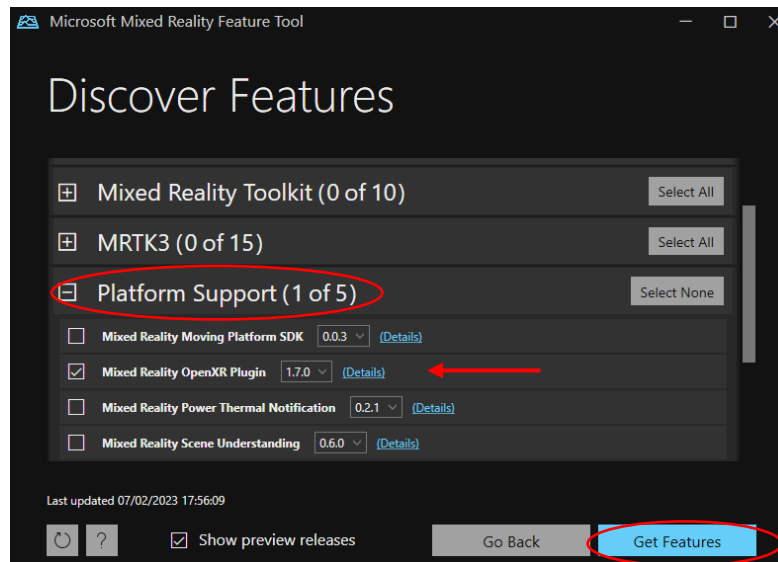

Figure 28: Selecting platform support plugins

Then click on the import button. You can also click the validate button before importing, so make sure there are no validation errors.
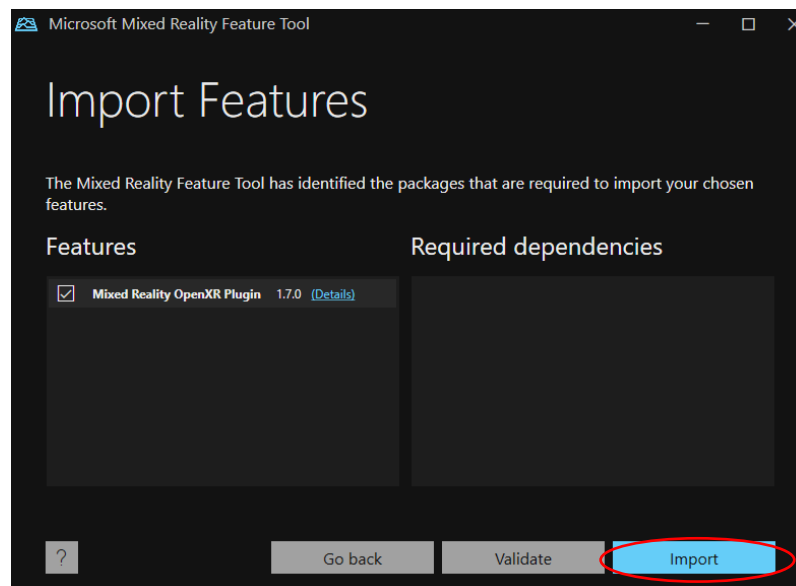

Figure 29: Importing Mixed Reality OpenXR Plugin

Go back to Unity and now if we check the menu bar, we can see a new option called '*Mixed Reality'.*
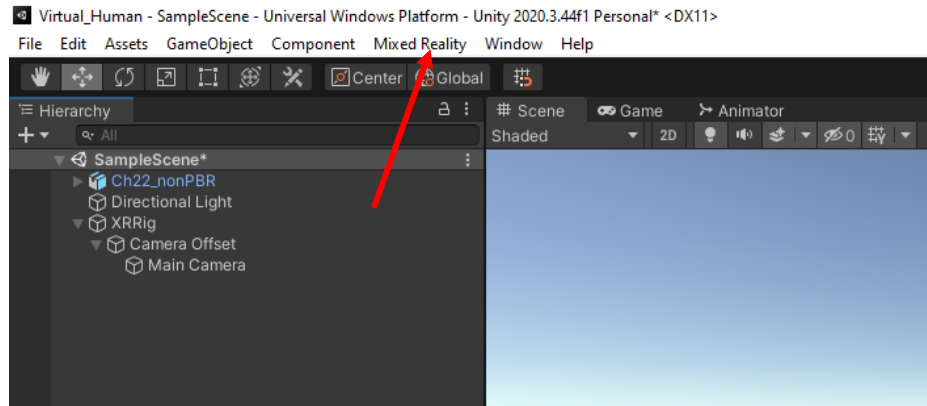

Figure 30: Mixed Reality option in the menu bar

Now we have to apply the project settings for HoloLens. From the menu bar select *Mixed reality >> Project>> Apply recommended project settings for HoloLens 2.* Then open the build settings and go to player settings. From the left panel, select '*XR Plug-in Management*'. On the right side window, check the boxes as shown in figure below (Figure 31).
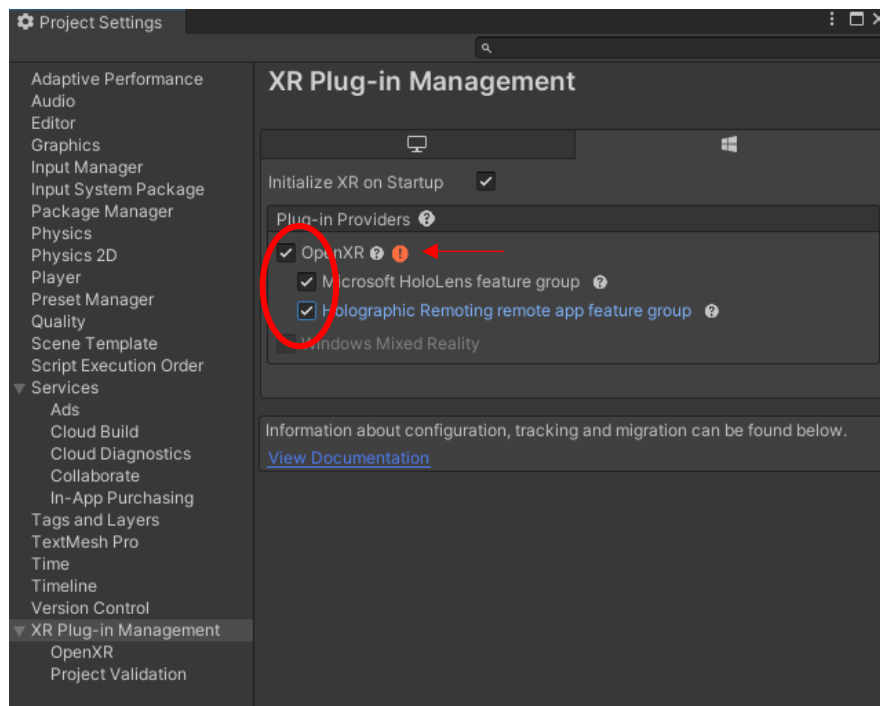

Figure 31: Choosing OpenXR plugin features in project settings

If Unity shows some warnings or issue symbols, simply click on it and click the '*Fix*' button.

Now again go to the *'Mixed Reality'* option in the menu bar and select *Remoting >> Holographic Remoting for Play Mode.* A small window will be opened. There we have to give the remote host details of the HoloLens. We will get the host name from the holographic remoting app in HoloLens 2.0.
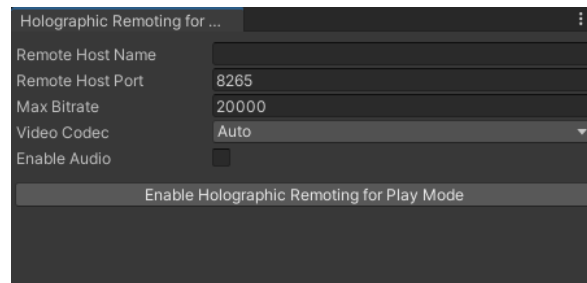

Figure 32: Details of holographic remoting

Before playing, make sure the position and orientation of the camera and character are adjusted properly for getting a good view.
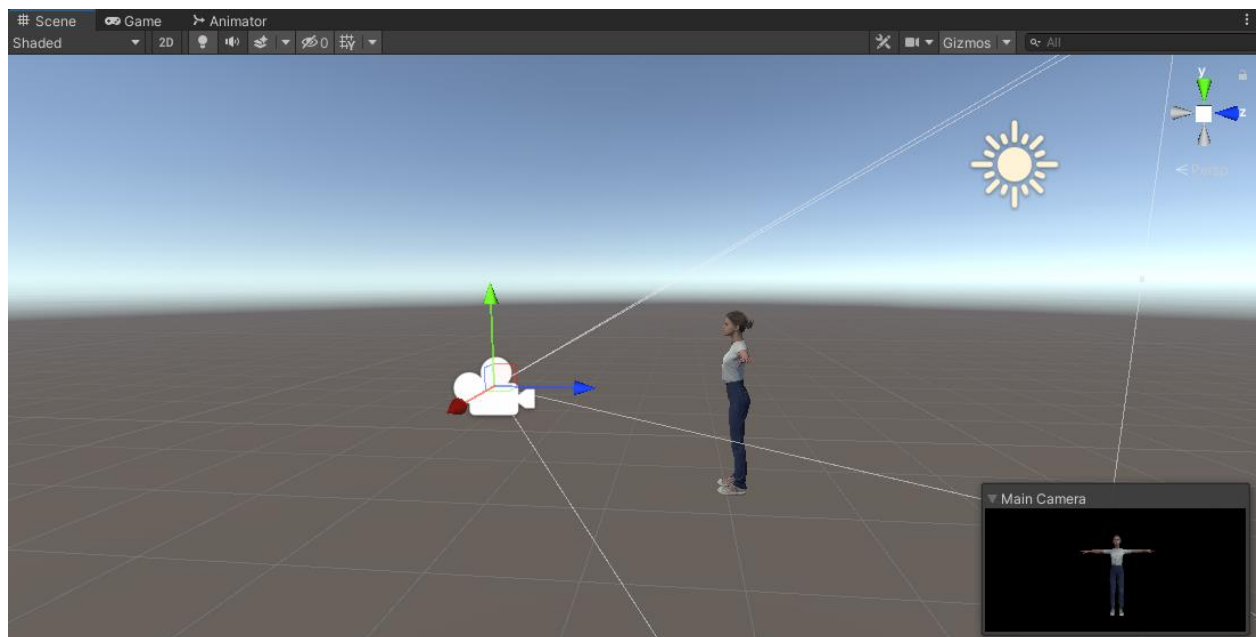

Figure 33: Adjusting orientation and position of camera and character

Visualizing with Hololens 2.0.

Wear the HoloLens. Switch it on and sign in into it. Go to All Apps and select the *Holographic Remoting Player* App. Also make sure that the computer and HoloLens are connected to the same network.
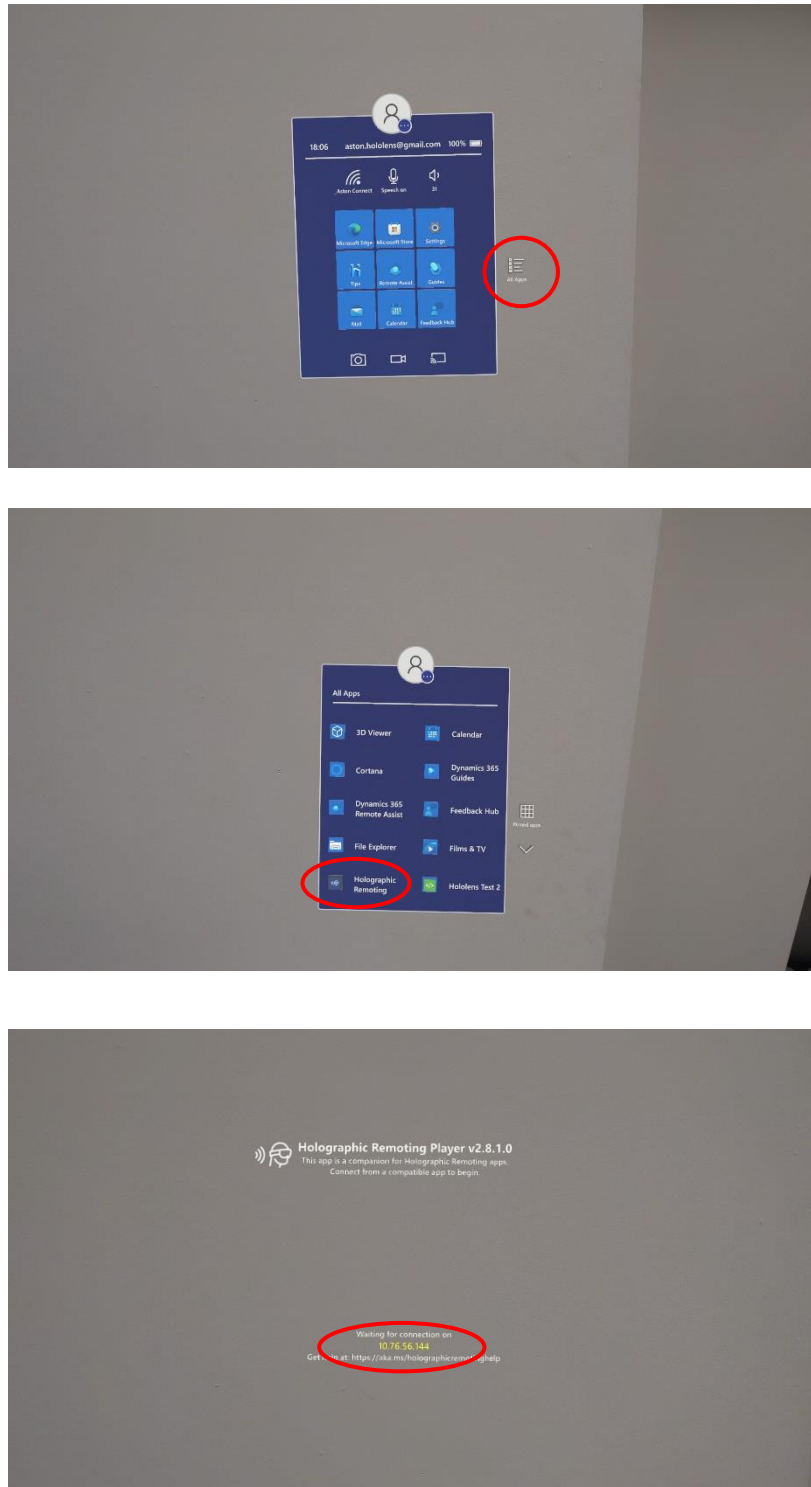

Figure 34: Opening Holographic Remoting Player in HoloLens 2.0

Once the Holographic Remoting Player is opened, it shows the host name at the bottom. Use this host name at the 'Remote Host Name' field in the holographic remoting details window, discussed earlier above. Now we are all set to visualize our Virtual Human in front of our eyes. Click on the Play button on the top of Unity scene window and now we can see the Virtual Human standing next to us.

Figure 35: Virtual Human visualized through HoloLens 2.0