Unity Development Workshop Preparation

ECE 209AS Spring 22

There are so many outdated resources online, and there are a lot of things to configure. But everything is worth it when we see our app in VR! It is the first time for 209 to introduce VR headsets to the classroom. No prior documents, no prior experience. It can be difficult but you can do it! This document will walk you through it step by step. Be ready to spend hours on this. Please post on slack and @Siyou should you have any trouble :)

GET STARTED

Set Up Development Environment

We should have **an oculus quest headset (the cable included), a smartphone, and a 64-bit computer** ready.

Minimum System Requirements for the computer:

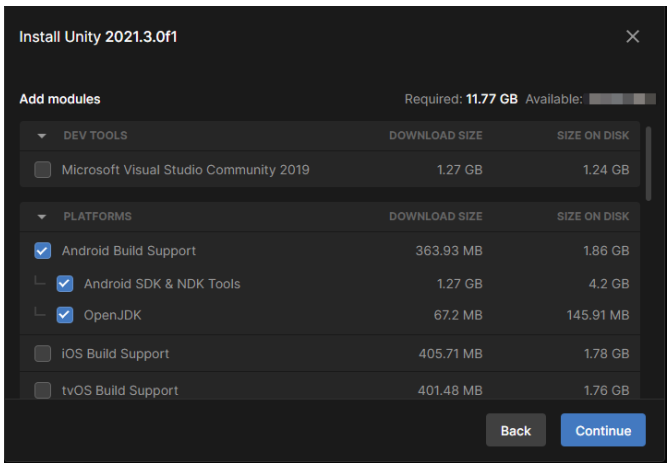
- 2.0+ GHz processor
- 2 GB system RAM

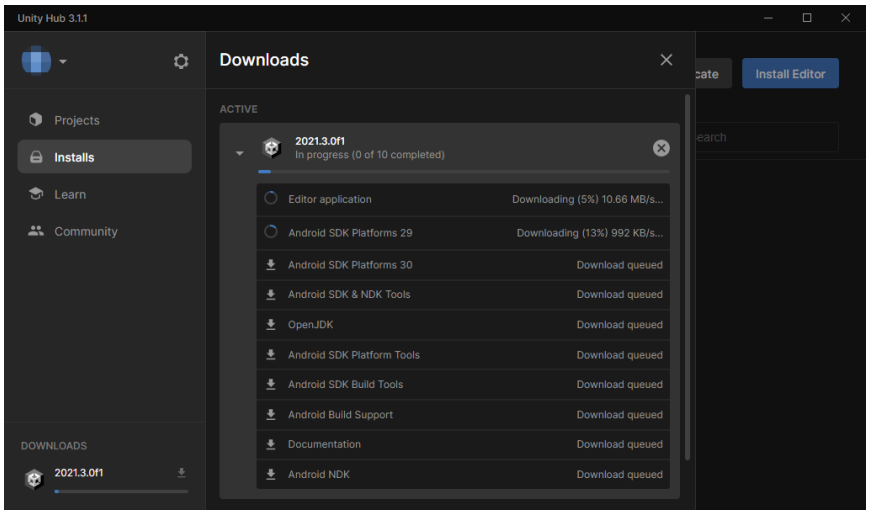
Headset and phone:

1. Sign up to be a developer https://developer.oculus.com/sign-up/ (either option is okay)
2. Download the Oculus app on phone and sign in with the account
3. Pair the headset and controllers
4. Wear the headset to finish the setup tutorial

The computer:

1. Install Unity https://store.unity.com/#plans-individual (The version I installed is 2021.3.0f1 64bit for Win 10. If you want another version, I recommend one later than 2020.3). Make sure we select packages “Android Build Support”, “Android SDK & NDK Tools” and “OpenJDK”. I already have VS 2019 on my machine, so I cleared it. If you don’t have it, please leave it selected. If you don’t want to use VS 2019 but want to use just VS code or other IDE, feel free to clear it. Other packages are optional. Click continue then install.





Ref link: https://developer.oculus.com/documentation/unity/book-unity-gsg/#install-unity-editor

https://docs.unity3d.com/Manual/android-sdksetup.html

Enable Device for Development and Testing

We should connect headset over a USB-C cable (USB3 is preferred) to the computer before we do the following:

Headset:

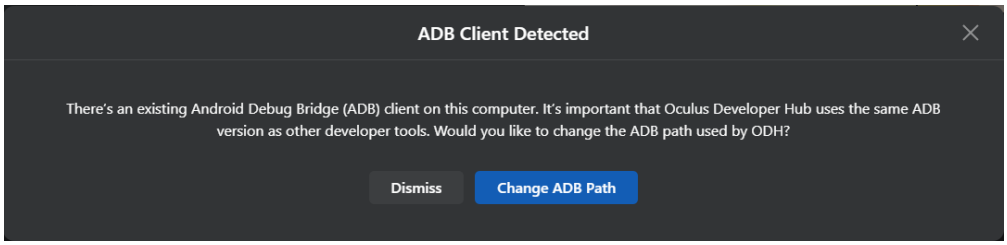
1. go to **Settings** > **System** > **Developer**, and then turn on the USB Connection Dialog option. (Alternatively, you can open the Oculus mobile app, go to **Menu** > **Devices** > **Developer Mode** > turn it on.)

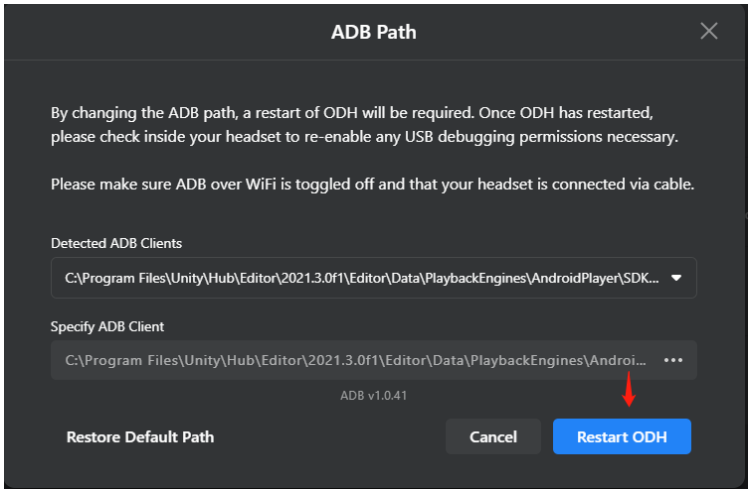
2. go to **Settings > Device > Hands & Controllers**, and turn on Hand Tracking. Leave the **Auto Switch Between Hands And Controllers** selected to let you use hands when you put controllers down.

3. We will see a pop-out bubble in the headset, Click Allow for “allow access to data”/”USB debugging”/”Always allow from this computer”.

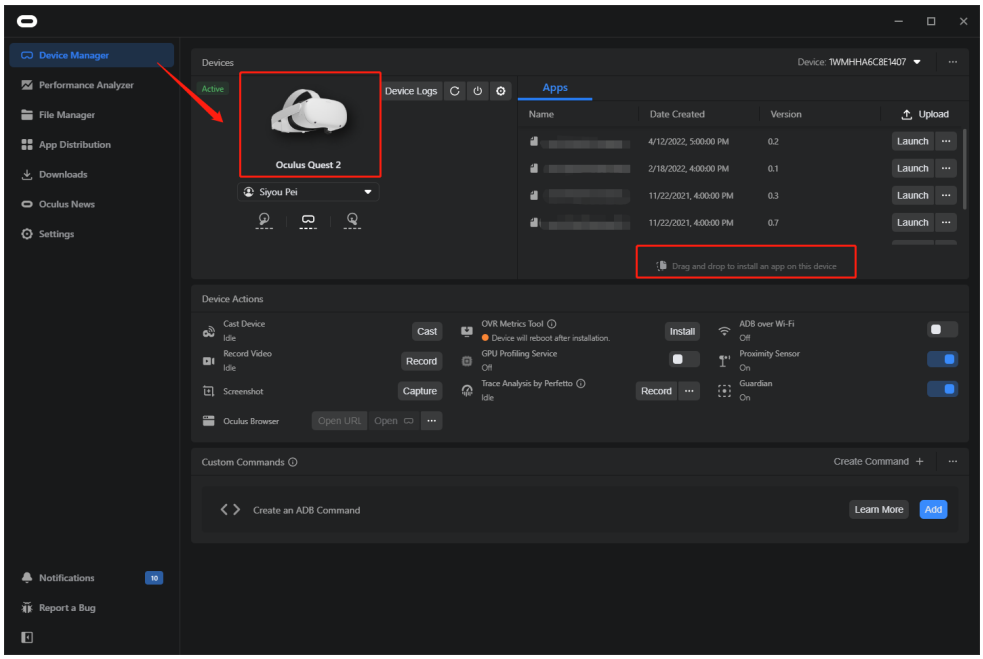
Computer - ODH:

1. Download and configure oculus developer hub following this tutorial https://developer.oculus.com/documentation/unity/ts-odh/#set-up (“Set Up Headset from ODH” sessions and after are optional, dependent on your need)
2. Make sure you change the ADB path to Unity’s ADB path



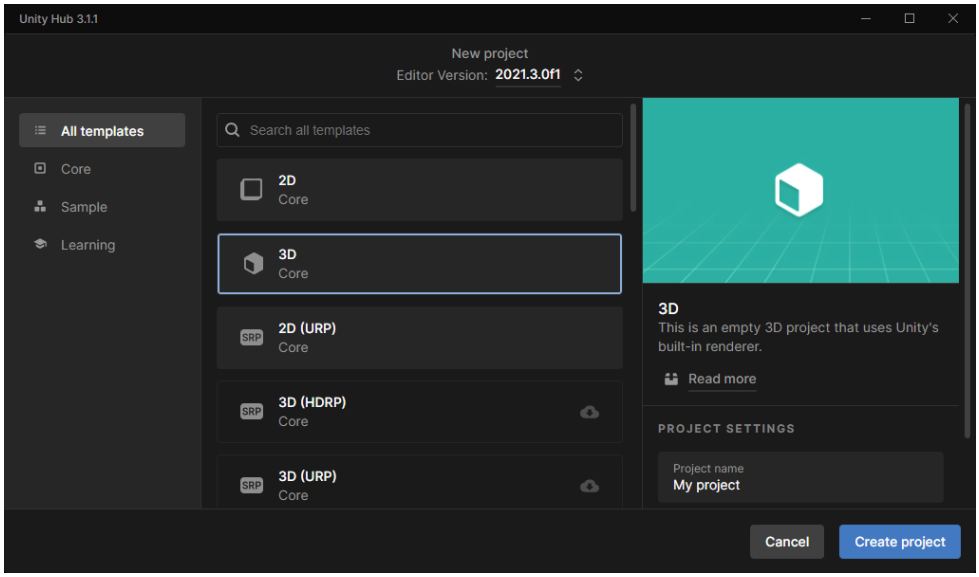


3. You will see your device connected in the developer hub app.

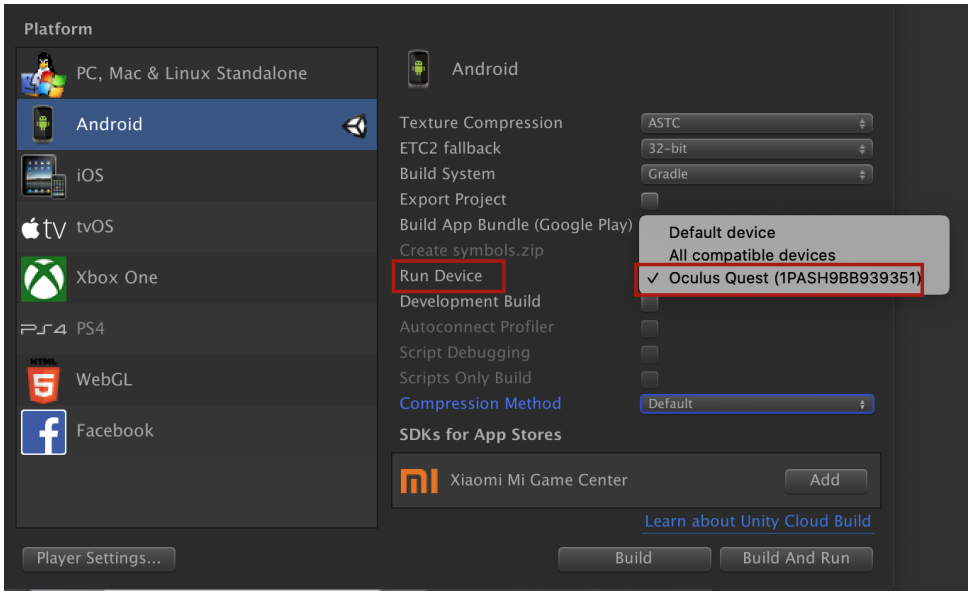


Computer - Unity:

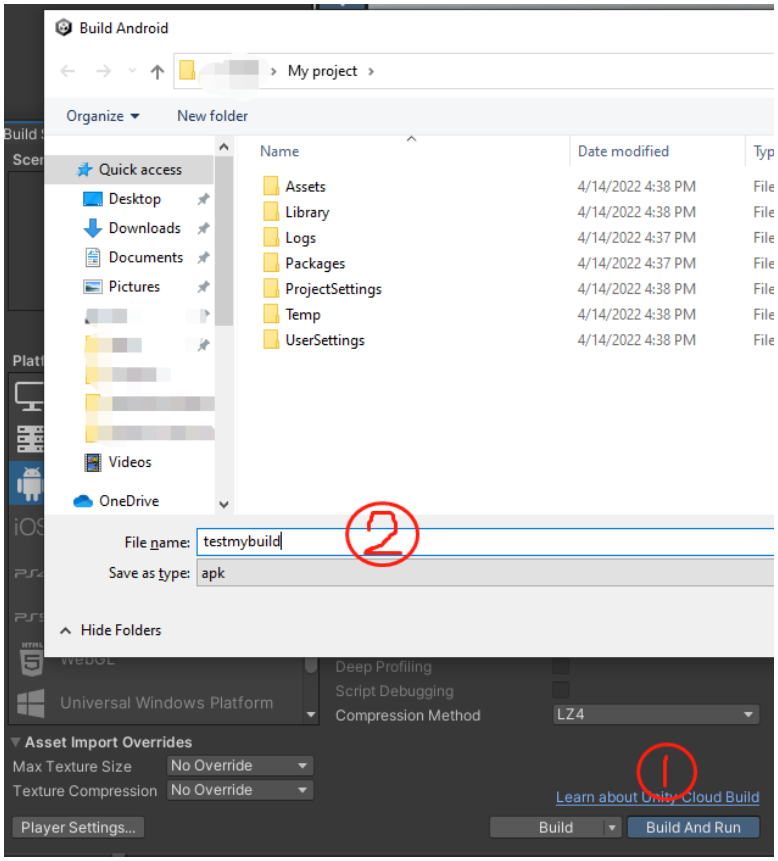
1. In Unity, Create a unity project (Select 3D). Grab a cup of coffee. The first time will take some time.



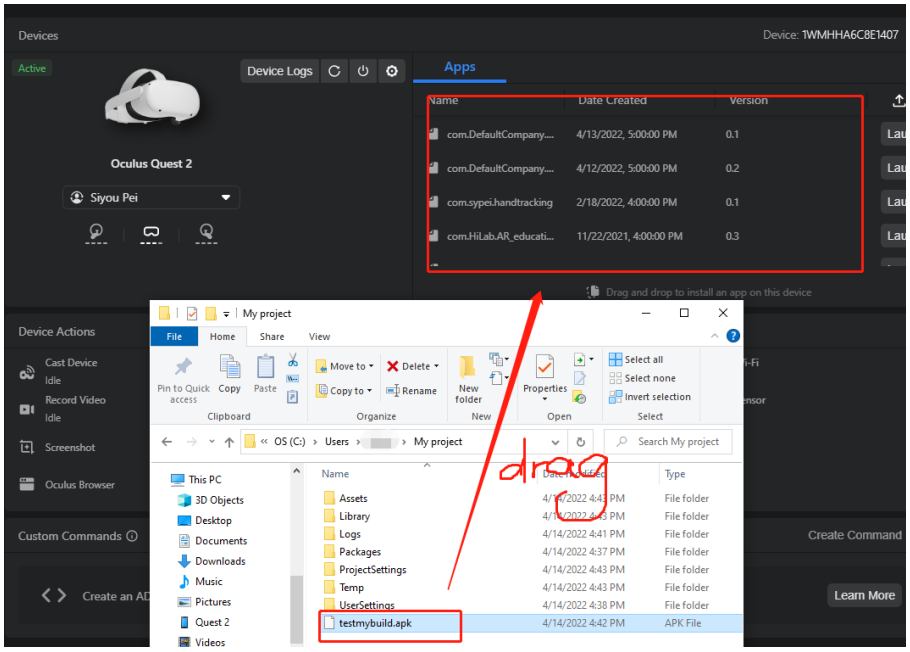
2. Go to **File > Build Settings**. In the Platform list, switch to platform Android. In the Run Device list, select the Oculus headset.



4. Click build, save the apk with a name (take some time to finish). Please remember where you saved it!

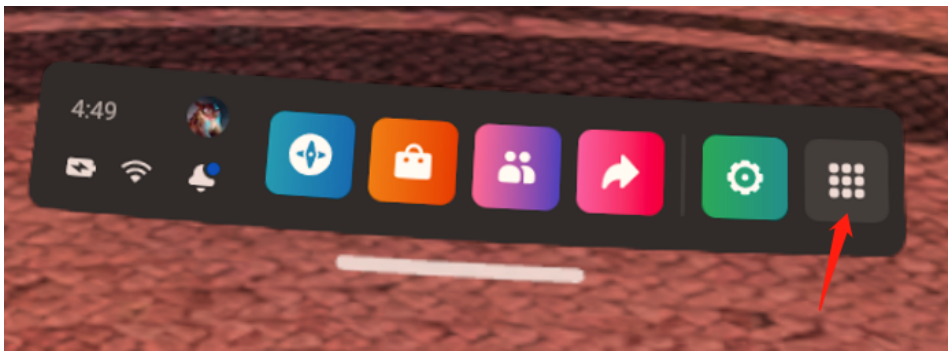


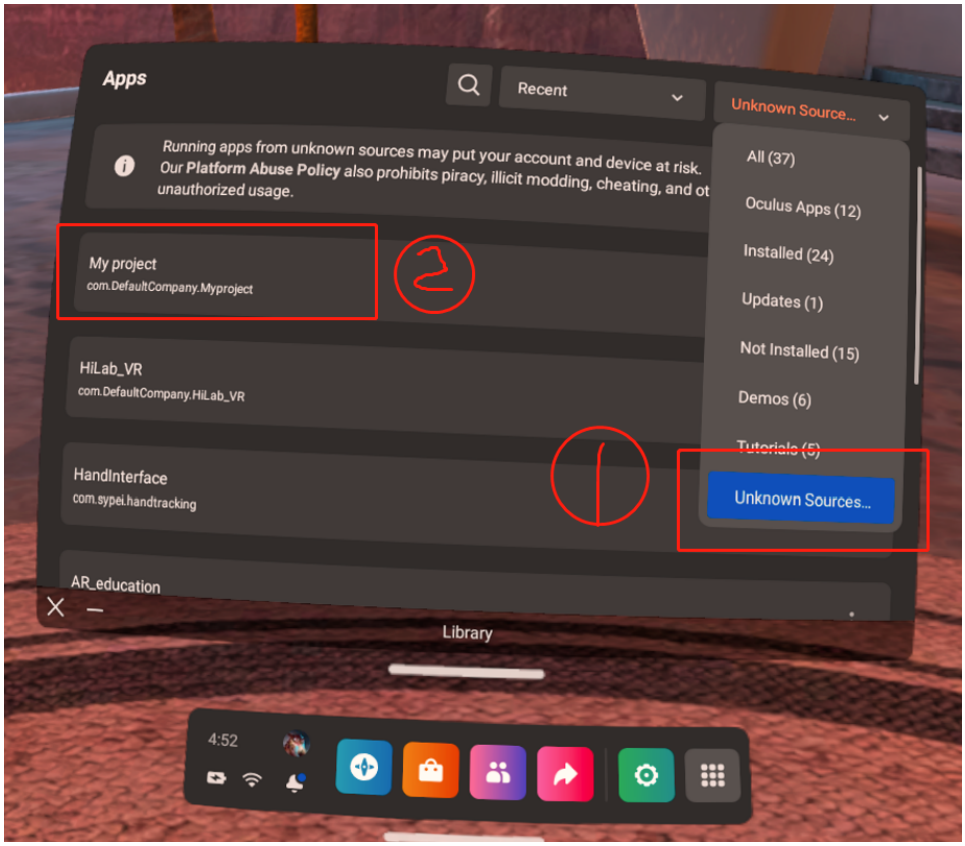
5. After the build, we open the ODH, and drag the apk file to the region



Headset:

1. Click **Settings (the nine-grid button) > unknown sources > my project**

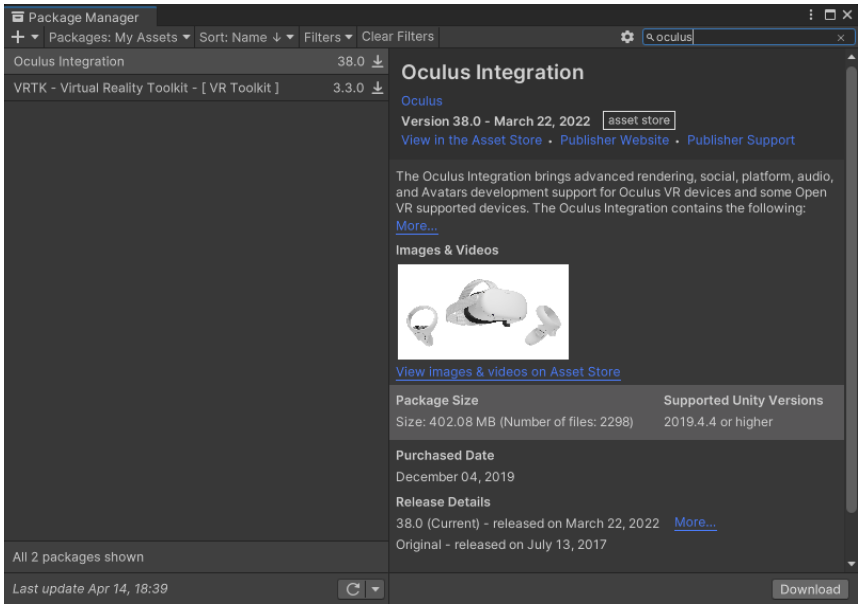




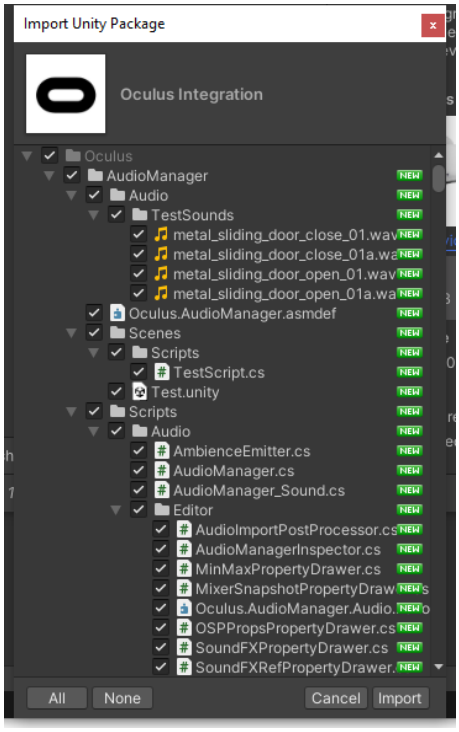
2. Then we will see the app like this, the unity scene (blue sky and brown ground) shows up. Wait, it's not a VR app! And I cannot interact at all! Alright you are right. But up to now we already know how to export an app from unity to oculus! All the rest is to do configuration inside Unity, which will be covered in the next sessions.

Import Oculus Integration Package

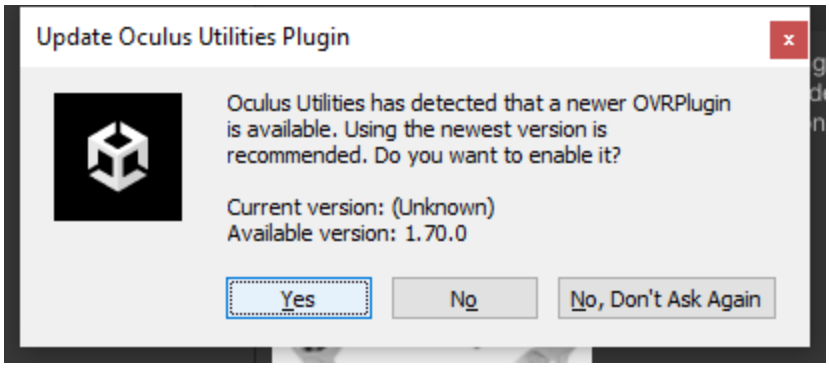
1. Go to the Unity Asset Store, sign in using your Unity account (register one if you don't have one), and then click Open in Unity.
2. Unity opens the Package Manager window in the Unity editor.
3. In the Package Manager window, choose “my asset” in the upper left corner, then search for **Oculus Integration**, and then on the bottom-right side of the window, click download/Upgrade, then click “import” to import it to “my project”. (If you create a new project, you will need to do step 1,2,3 again)



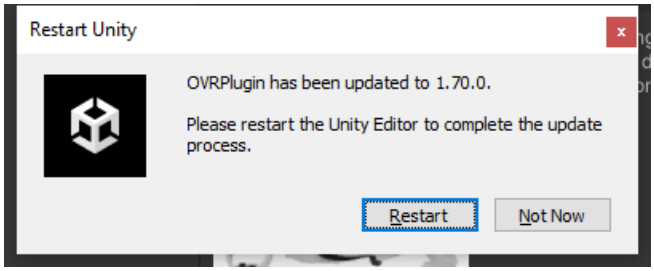
4. In the Import Unity Package window, leave all the files and folders selected, and click Import. It takes some time.



5. When prompted to update the Oculus Utilities plugin, click Yes



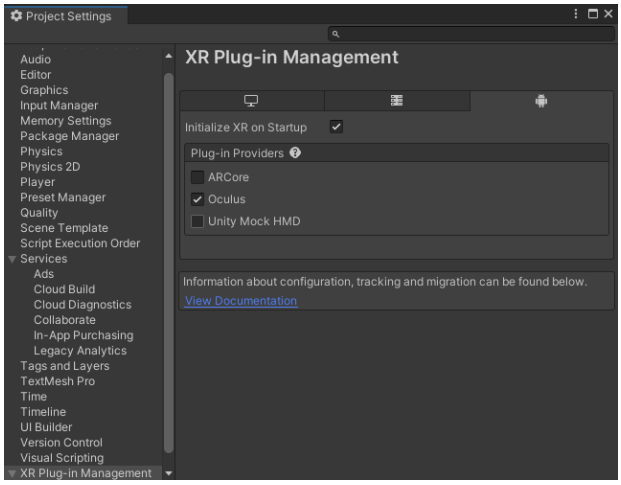
6. ‘Use openXR’ is optional, depending on your needs. You can switch it back later https://developer.oculus.com/documentation/unity/unity-openxr/
7. Click Restart



8. When prompted to update Spatializer plugins, click Upgrade, and then click Restart.

Configure Unity Settings

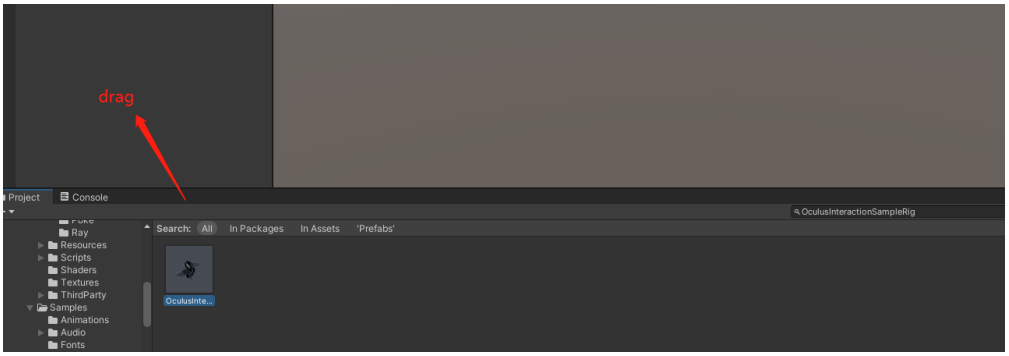
1. Go to **Edit > Project Settings > Player**, and then expand the **Other Settings** tab >
    a. Rendering >
        i. Color space - linear
        ii. Uncheck Auto Graphic API
        iii. Remove Vulkan, keep OPENGLES3
        iv. Check multi threaded-rendering
        v. No need to modify the others
    b. identification >
        i. Minimum API Level, set to Android 6.0 Marshmallow (API level 23)
        ii. Target API Level, select Automatic (highest installed)
    c. Configuration >
        i. Scripting Backend- IL2CPP
        ii. Clear the ARMv7 checkbox and instead select the ARM64 checkbox
2. Go to **Edit > Project Settings > XR Plugin Management (at the bottom left, scroll down)**, click Install XR Plugin Management. Click the Android tab and select Oculus to install the Oculus XR plugin, which enables the VR support.



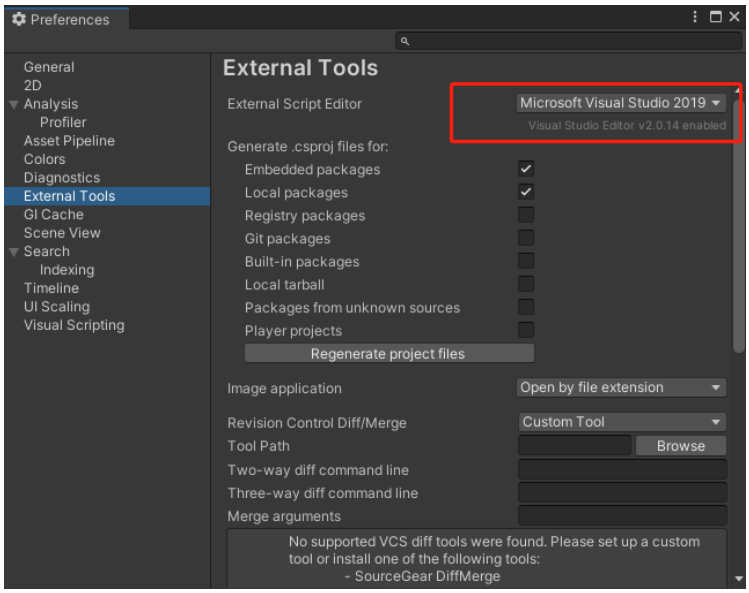
3. In the menu, go to **Oculus > Tools > Create store-compatible AndroidManifest.xml**.

Build Your App in VR

1. We finished all the configuration, now we need to develop the app.
2. Set up the camera. Camera represents your eyes in the VR world.
    a. Delete the mainCamera in the hierarchy.
    b. On the Project tab, search for ***OculusInteractionSampleRig***, and then drag the prefab into the hierarchy or scene.



    c. On the Hierarchy tab, expand ***OculusInteractionSampleRig*** and select OVRCameraRig to open the Inspector tab (on the right).
    d. On the Inspector tab, go to **OVR Manager > Tracking**. If your app is to be used while standing, set “Tracking Origin Type” to Floor Level. If otherwise, leave it at Eye Level. Also, set “Display - color gamut” to be Quest.
3. We will use hand tracking in the workshop too. Let’s set up hands.
    a. Still in OVRCameraRig’s inspector, go to **OVR Manager > Quest Features**, and then in the Hand Tracking Support list, select **Controllers and Hands** or **Hands only**.
4. For future development (in the workshop), let’s check our IDE for C# coding. Go to **Preference > External Tools >** selected preferred script editor.



Ref link: https://developer.oculus.com/documentation/unity/unity-handtracking/

After configuring all these stuff for a LONG day, we are all set to build and run. Go to **File > Build and Run**. The first attempt will take some time. Be patient, we are almost there :)

After it finishes, put on your headset. You are IN the scene now!

(Your hands might be light blue or transparent)

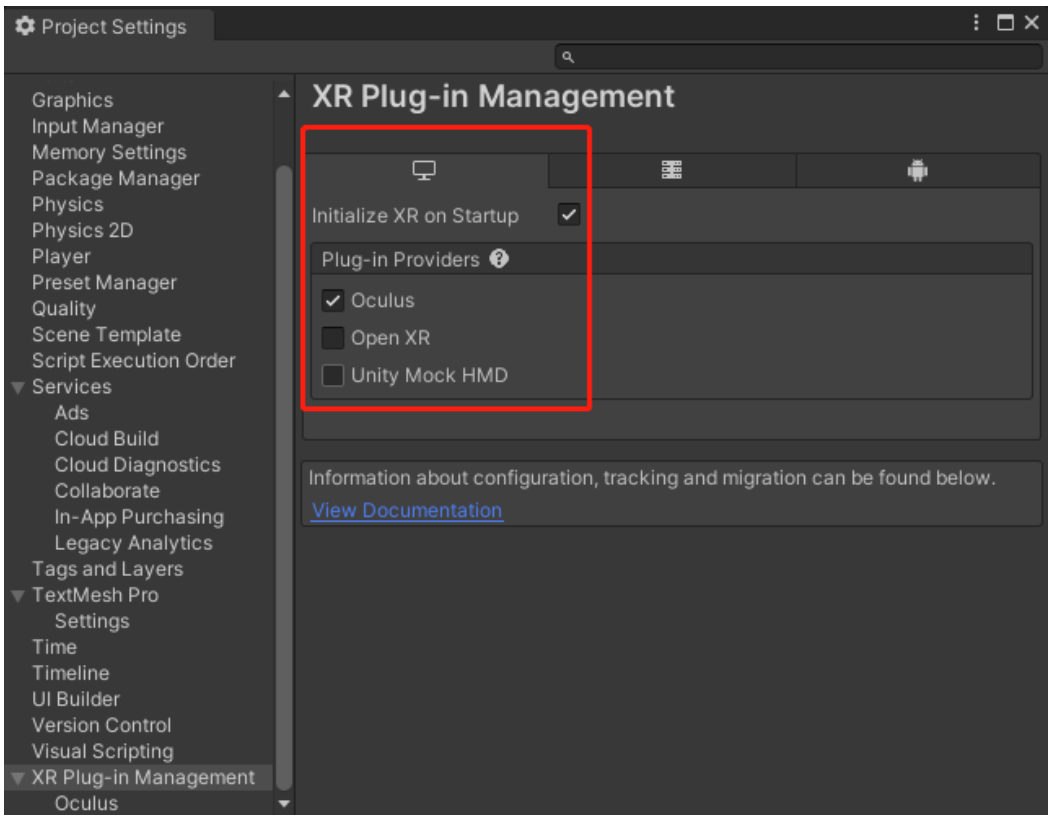
Okay, but the world is still empty. How can we create things in this world? How can we interact with them? **Let's learn how to do it in the workshop.**

Ref link: https://developer.oculus.com/documentation/unity/unity-gs-overview/

==========

If you still have spare energy, I highly recommend configuring your quest for play-mode in unity. That will save a lot of time from build and deploy! Steps:

1. Download the PC APP https://www.oculus.com/setup/ connect Quest to PC
2. Go to Unity Edit > Project Settings > XR Plug-in Management. Switch to PC - Oculus.



3. Follow this https://www.youtube.com/watch?v=ngXRqeF0igA Click “play” in unity and enjoy the view in quest!