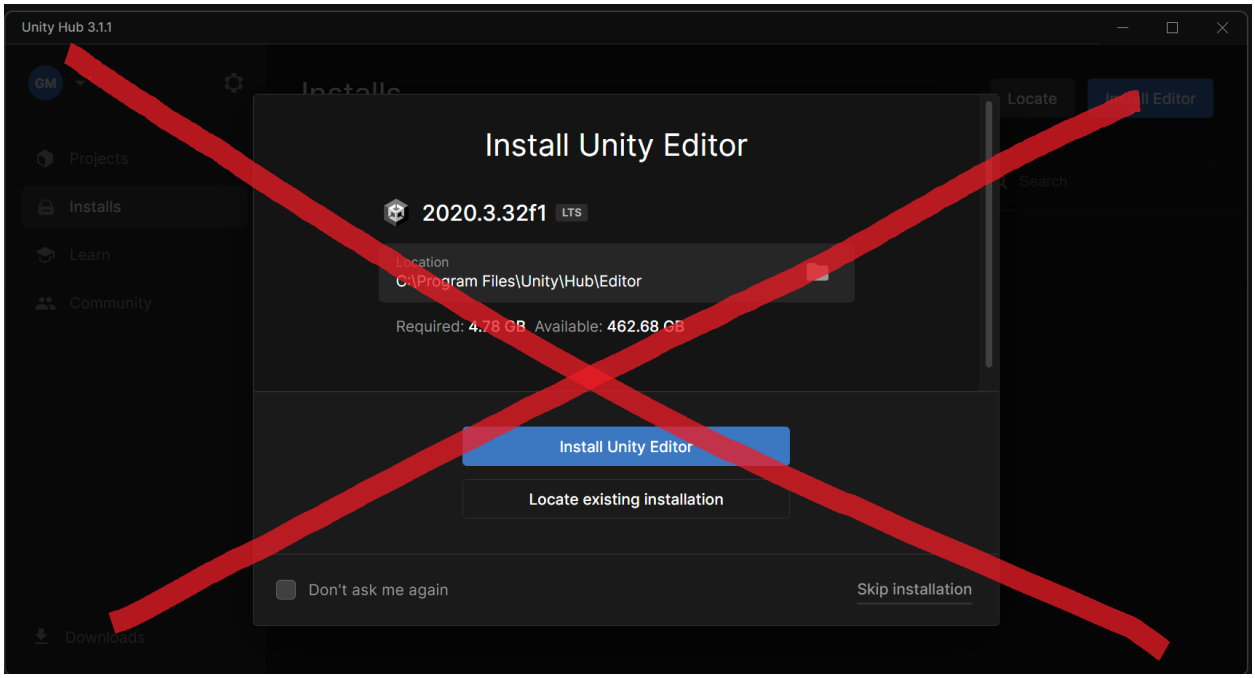
HW 0: Unity Installation

Created: Mar 28, 2022
Updated: Jan 11, 2023

1. Install Unity with Android

1. Install Unity Hub which lets us manage different projects and versions of Unity.
2. When you open Hub, don't install the version of Unity it prompts you to use, click **Skip installation** instead.

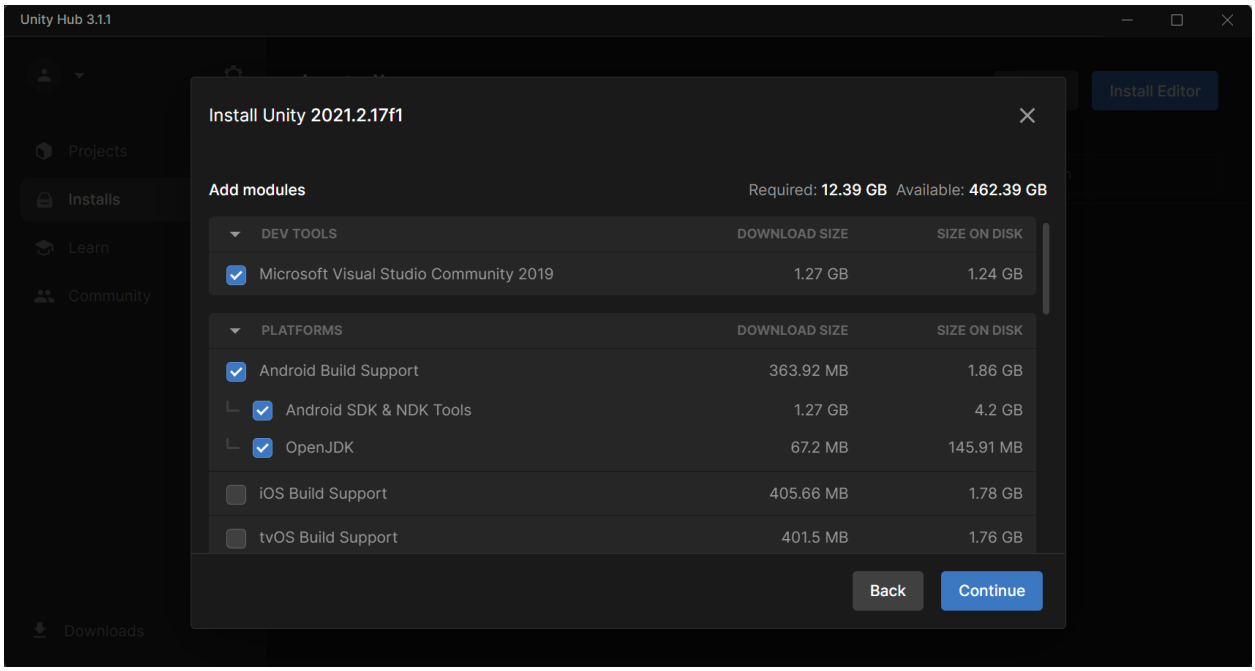


3. Click the **Installs** tab, then in the top right click **Install Editor**, then scroll down and click the **Install** button next to the latest version of Unity 2021 (2021.2.19f1 at the time of writing).

Note: If using an M1 Mac computer, the Intel editor version is probably a little more stable, but the Silicon version might be a bit faster so feel free to try it.

4. Under **Add modules**, definitely select **Android Build Support**, **Android SDK & NDK Tools**, and **OpenJDK**. Optionally select **Microsoft Visual Studio Community 2019** (you can install Visual Studio 2022 instead or use another IDE, though VS has very good Unity autocomplete

support (if manually installing VS, be sure to check the **Game development with Unity** workload in the VS Installer)). Optionally scroll down and uncheck **Documentation** to save some space (you can look the docs up online). With all of that checked, you'll need 13 GB of storage.



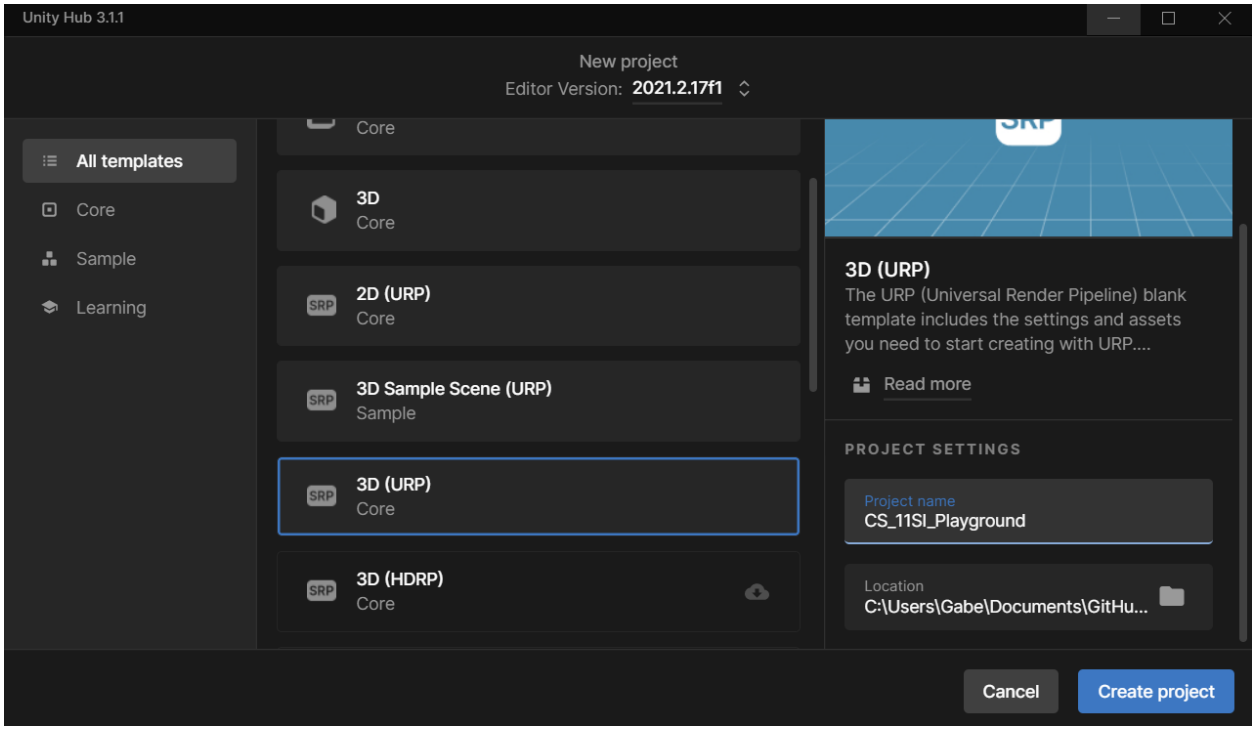
Note: In case you've already installed Unity without Android support, you can still add Android tools from Unity to any existing Unity versions by selecting the gear icon next to the Unity version and selecting Add modules.

5. While waiting for the Unity editor to install, create a Unity ID so you can log in. Then select the profile icon in the top left of Hub and select **Sign in** to sign in.
6. Also in the top-left of Hub, select your profile, select **Manage licenses**, select **Add**, then select **Get a free personal license** and follow the instructions.

2. Create a Project

1. With Unity 2021 installed, open the **Projects** tab in Hub then select **New project**.

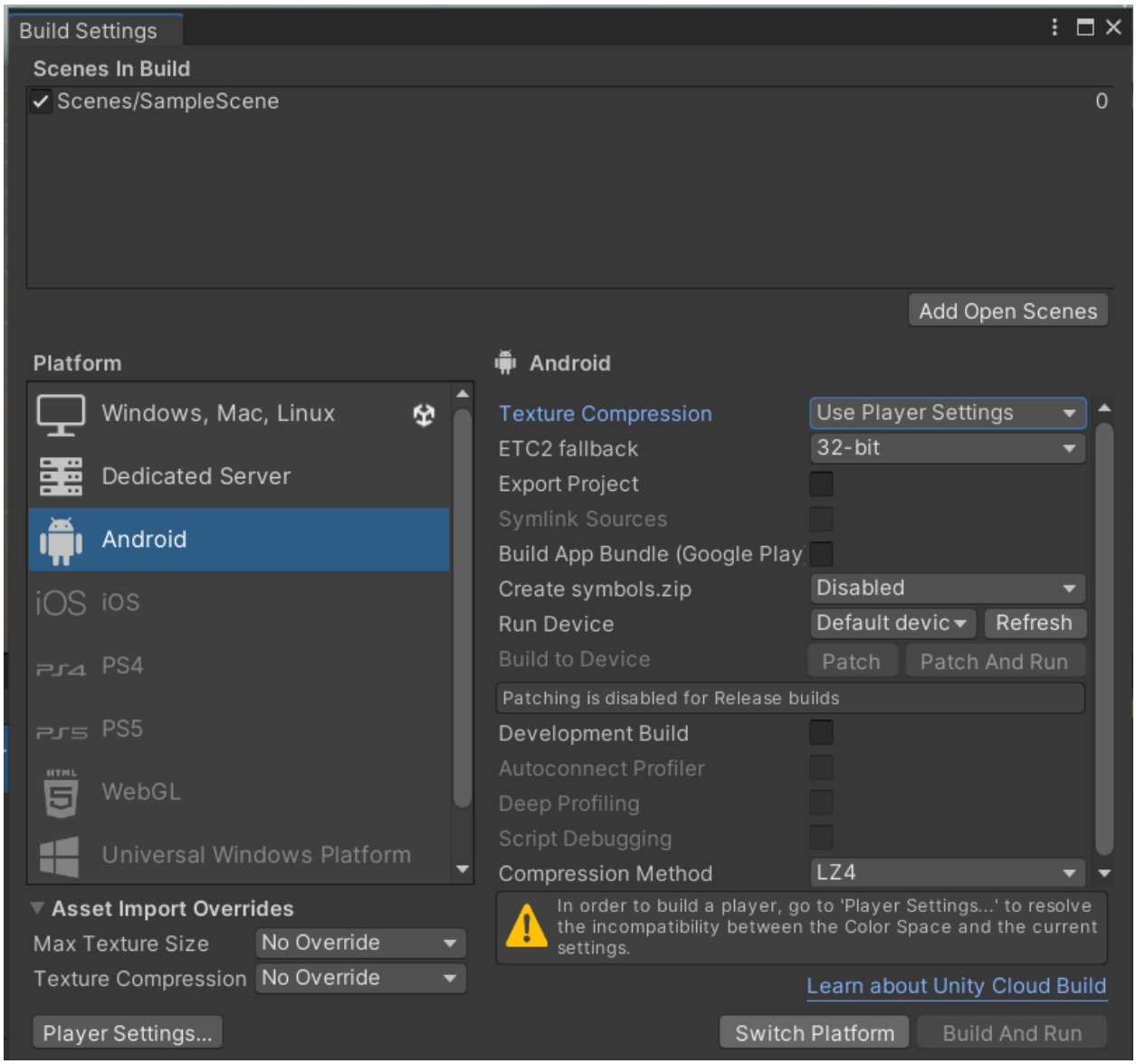
2. Click the **3D (URP)** template (not the VR template or 3D template) and click **Download template** on the right.
3. Under **Project Settings** on the right, give your first project a creative name (doesn't have to mean anything) but don't put spaces in the name. Choose a folder in which to create it, then select **Create project**.



3. Configure Your Project for VR

Here, we'll set up our project for mobile (Android) VR development which is what the Quest 2 platform is. If you have a Windows computer with a good graphics card and want to test PCVR with a Quest or another headset plugged into the computer (faster iteration time, but requires a better computer), you'll just need to follow the optional PCVR instructions and switch to the **Windows, Mac, Linux** platform in **Build Settings** when you're ready to test (probably don't worry about this yet).

1. With your Unity project open, select **File->Build Settings**, select the **Android** tab under **Platform**, and click **Switch Platform**.



2. Go to **Edit->Project Settings->XR Plugin Management** and select **Install XR Plugin Management**.
3. [PCVR Optional] Within XR Plugin Management, click the **Windows, Mac, Linux** tab (it's a computer monitor icon) then check the **Open XR** plug-in provider to install the OpenXR package. Select **Yes** when it prompts you about the new input system and let it restart the editor. Then, open **Edit->Project Settings->XR Plugin Management** and navigate to the **Windows, Mac, Linux** tab if it's not already there. Click

the red exclamation point next to Open XR and follow the instructions to fix the issues with it.

4. Within XR Plugin Management, select the **Android** tab (it's an Android icon) if it isn't already selected. Select the **Oculus** plug-in provider to install the Oculus package.

Note: If you installed OpenXR for desktop platforms in the previous steps, you'll also see an Open XR checkbox here (it won't show up if you didn't, but that's okay). OpenXR is the future standard for VR plugins within game engines and apps, but Unity is currently missing several features from its OpenXR implementation, the most important for us being Fixed-Foveated Rendering (FFR) which is a very useful and probably mandatory performance optimization for mobile VR. If you want to try OpenXR on Quest or you're from the future and want to test with a non-Oculus mobile VR headset (e.g. Pico Neo I think) or Foveated Rendering is supported in OpenXR, click that plug-in provider instead of Oculus and follow similar instructions as the previous step. For the purposes of our class, these different providers won't change our platform-agnostic VR code using Unity's XR Interaction Toolkit, so feel free to change which provider is checked to test performance and visual differences between them.

5. In **Edit->Project Settings->Player**, set the **Company Name**, **Product Name**, and **Version** to something reasonable and without spaces. E.g. you might set the Company Name to your SUNet ID, leave the Product Name as your project name, and set the Version to 1.0.0. Note that your Company Name and Product Name cannot start with a number.
6. Still in the **Player** settings, scroll down and click on the **Android** tab (for optional PCVR, you don't need to change any additional Player settings yet), expand **Other Settings**, and ensure the following settings are set appropriately:
    a. **Color Space**: Linear
    b. **Auto Graphics API**: False
    c. **Graphics APIs**: Use the +/- icons to add Vulkan and remove all OpenGL versions
    d. **Multithreaded Rendering**: True

    e. **Texture compression format**: ASTC
    f. **Minimum API Level**: Android 6.0 ‘Marshmallow’
    g. **Scripting Backend**: IL2CPP
    h. **Target Architecture**: Uncheck ARMv7, check ARM64
    i. **Active Input Handling**: Input System Package (New)
7. Finally, we’ll install the XR Interaction Toolkit. Select **Window->Package Manager**, select the **+** icon in the top left, select **Add package from git URL**, enter **com.unity.xr.interaction.toolkit**, and select **Add**. Give it some time to install the package, then select **I Made a Backup, Go Ahead** when prompted.