

Simple XR Rig Setup

Project Setup

Launch **Unity Hub**.

Click **New project**.

Select **3D Cross-Platform (URP)**.

Why URP?

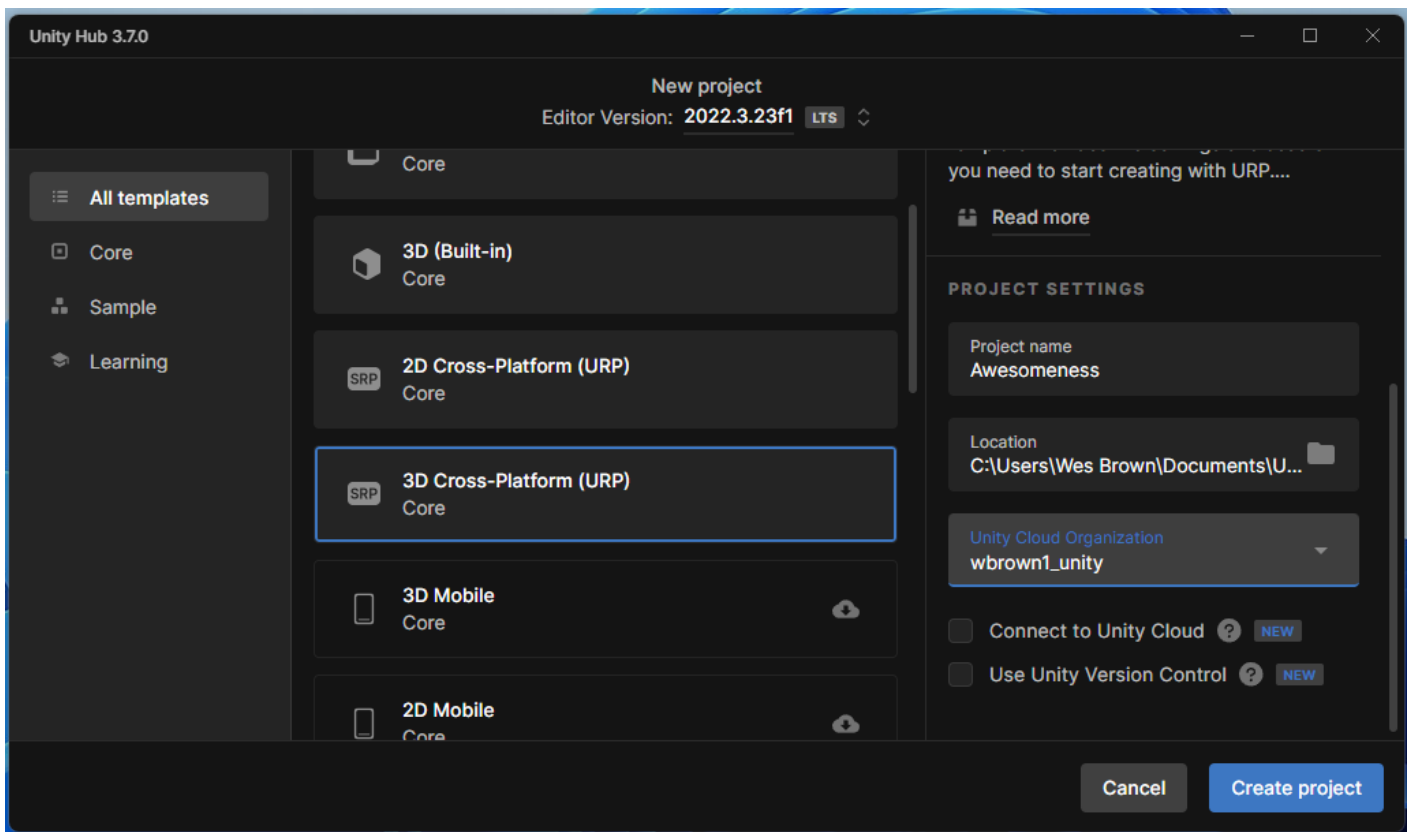
Unlike the standard or built-in 3D pipeline for Unity, the universal rendering pipeline (URP) gives scalability for different systems out of the box as well as node-based shader creation.

See more: <https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@17.0/manual/index.html>

Give your project a name and choose a file destination under **Project settings**.

Uncheck **Connect to Unity Cloud** and **Use Unity Version Control**. You can connect to these services later, if you wish.

Click **Create project**.



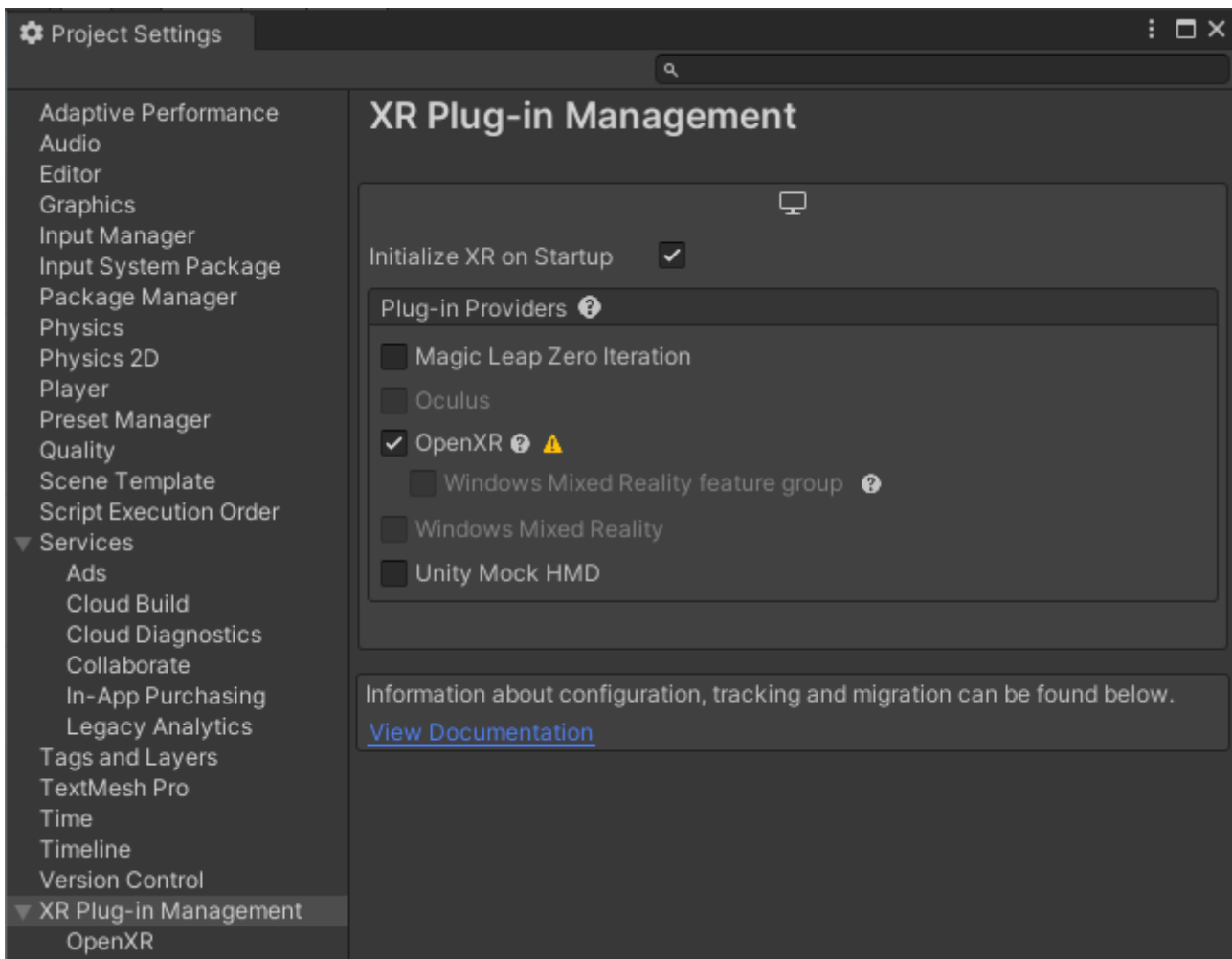
The project will be built. Take a few moments and get acquainted with the layout of the Unity Editor. Take a particular look at the **Project Window**, where your assets will live. Check out the **URP Readme** file. When you are ready, you may click the **Remove Readme Assets** button in the **Inspector**.

Enabling OpenXR

Go to Edit > Project Settings. Click on **XR Plugin Management** on the left and install the XR Plugin Management.

Check **OpenXR** under Plug-in Providers. Click **yes** when it prompts you to enable backends for the new Input System. Unity will restart. This switches Unity to use the new Input System (which is better and more scalable).

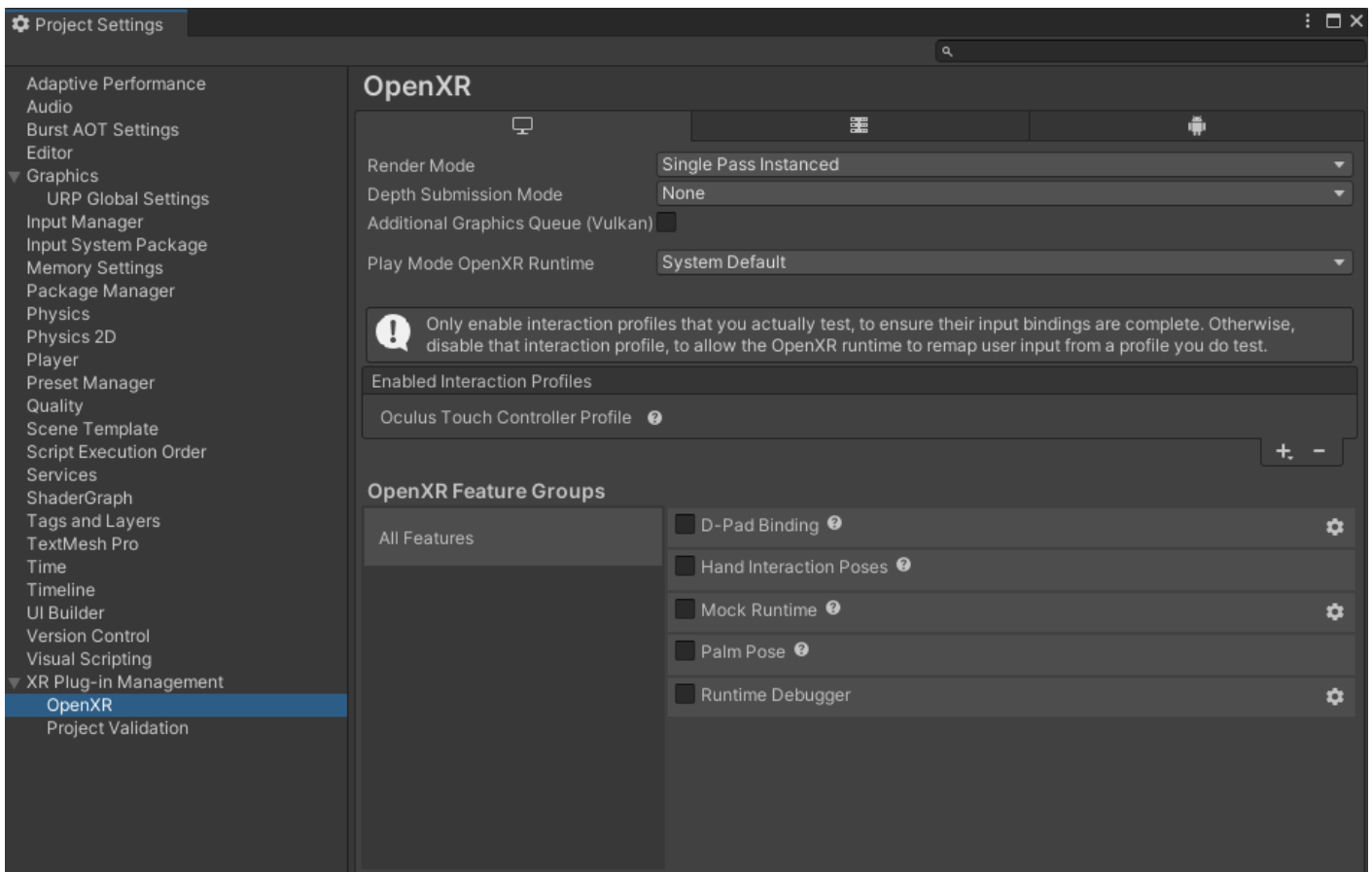
Click the yellow warning sign next to OpenXR. Click **fix all** in the upper right corner of the window that appears.



You'll notice not all of the warnings went away. What is left is adding controller support in something called the Interaction Profile. Before you leave, sometimes it is good to come back here to Project Validation to verify warnings are properly dealt with.

Navigate to the **OpenXR** tab under **XR Plug-in Management** on the left.

Under **Interaction Profiles** click the + button and add **Oculus Touch Controller Profile**. You may add more support for other devices if you wish.



Close **Project Settings** window.

Very Simple XR Rig

In the **Hierarchy** window, click on the Main Camera and reset its position and rotation to all zeros.

Right click on the **Main Camera** and select XR > Convert Main Camera to XR Rig.

Test in VR!!! I find it helpful here to create a temporary plane object (at position 0,0,0) for a floor and make sure my XRRig's Y-position is at zero, or matching the Y-position of the floor.

Optional: Adding Controllers

Expand the XRRig if not already and right click on **Camera Offset**. Select **Create Empty**. Name it **Left Controller**.

Select Left Controller add the component **Tracked Pose Driver**.

What is Tracked Pose Drive (Input System)???

Some of you may have noticed the other component, the **Tracked Pose Drive (Input System)**. If you had selected this as the component, you will notice there are some features to be added before getting this to work. These features are part of the newer Input System (and the component we chose is using the old system).

We are not covering the new Input System now in this tutorial. These controller game objects will be replaced when we install the XR Interaction Toolkit (XRIT) in a future tutorial.

Change device dropdown to **Generic XR Controller**.

Right click on Left Controller and add a 3D cube (or other 3D object according to what shape is desired to represent the controller) and resize it.

Do the same for the right controller.

For the right controller, change pose source in the component to **Right Controller**.

This tutorial is now complete.

DON'T FORGET TO TEST IN VR!

*We are not covering the new Input System now in this tutorial. **These controller game objects will be replaced** when we install the XR Interaction Toolkit (XRIT) in a future tutorial.*

Revision #6

Created 7 November 2023 15:16:37 by BaggoNotes

Updated 12 July 2024 13:49:13 by Wes