VR Hello World Program

Using the Meta Quest Pro

Chansu Yu and Felicia Wenner

Washkewicz College of Engineering

Last revision: 8/18/2023

Table of Contents

Introduction

This is a beginner's guide for those wanting to learn more about VR programming. It'll take you through the basics of creating a project, installing packages, using and modifying GameObjects in Unity, writing simple code in C#, and more.

NOTE: This guide is specific to the Meta Quest Pro, but may work with other Oculus devices.

What you'll need:

- ☐ Meta Quest Pro (VR headset and controllers)
- ☐ The Link Cable (or similar high-quality USB 3 cable)
- ☐ Your computer (Windows or Mac OS)

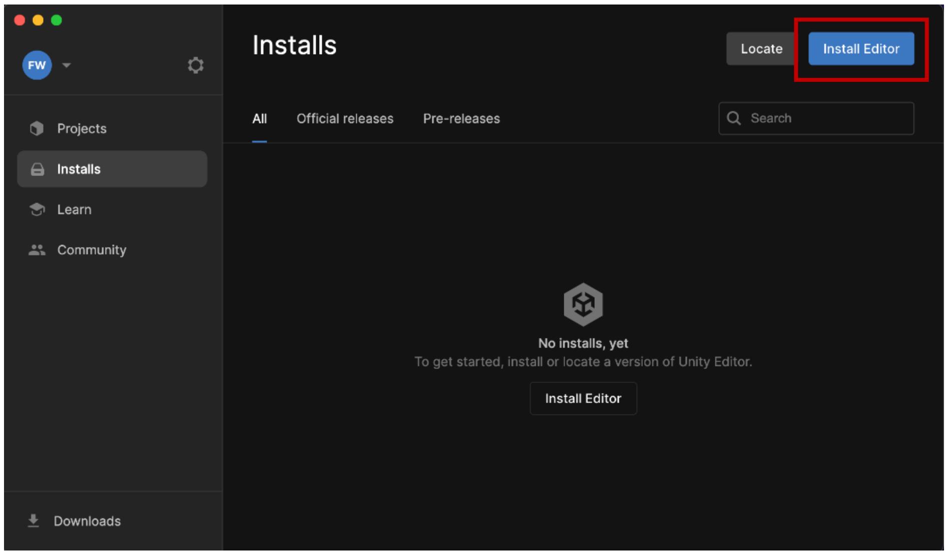
IMPORTANT: A USB-C cable capable of high-speed data transfer is required for the software to recognize your device.

For additional assistance and questions, you can contact us through our Discord server "CSU MakerSpace VR AR": https://discord.gg/s3j4WY8uk3.

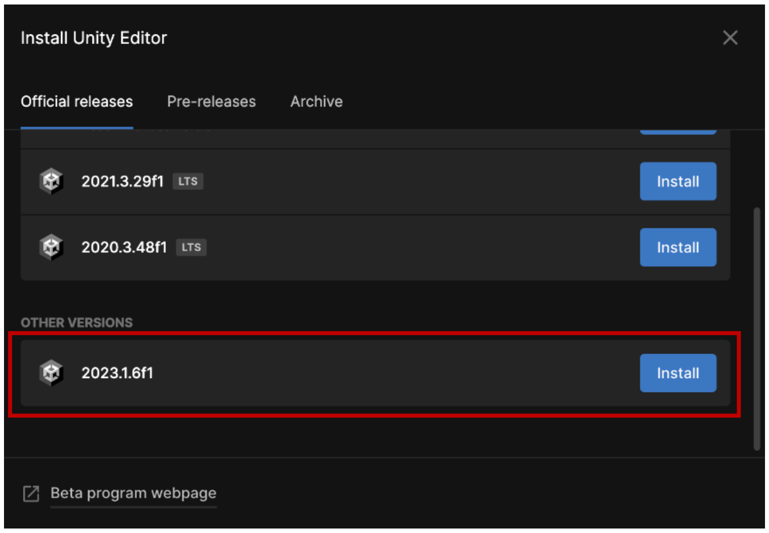
Install and Set Up Tools

Unity

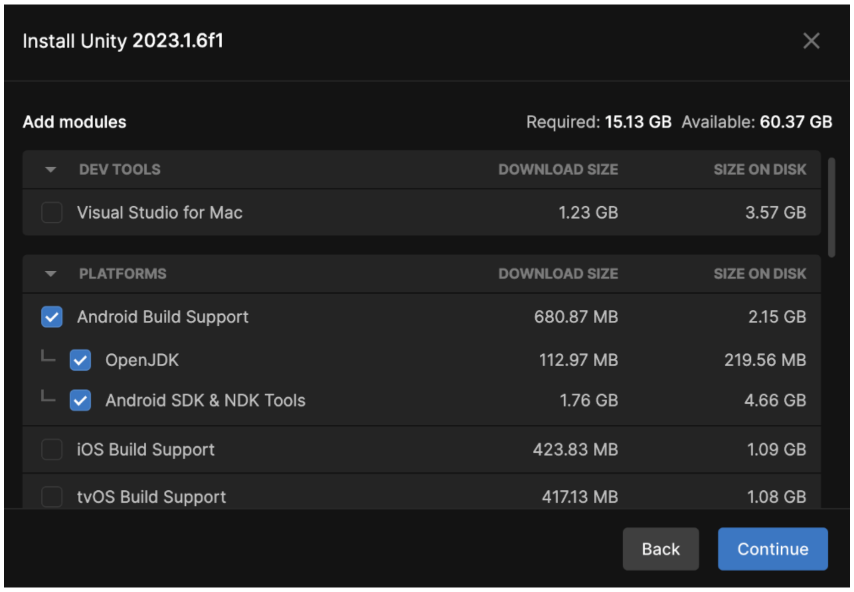
- ☐ Download and install Unity Hub.
  - https://unity.com/download
- ☐ Launch the Unity Hub application.
- ☐ Log in with your Unity account (or create one if needed).
- ☐ If this is your first time running Unity Hub, prompts for initial setup will appear after logging in. Follow the prompts along with these steps:
  - When prompted to "Install Unity Editor", click "**Skip installation"** in the bottom right.
  - When prompted for a license, choose the personal edition license.
    - You can apply for a student license here: https://unity.com/products/unity-student. For this project, a personal license is sufficient.
- ☐ Go to **Installs** on the left-side panel, then click "**Install Editor**".



- ☐ Install the latest 2023 Unity Editor.
  - **IMPORTANT:** Do **NOT** install the LTS version. This causes templates used later to be different than the one used in this guide.



- ☐ **Add modules** screen:
  - **NOTE:** This screen appears during the Editor install steps, but can also be accessed in the **Installs** window by selecting the cog wheel next to the Editor version you'd like to add modules to.
  - Uncheck "**Visual Studio**" under the "Dev Tools" section.
  - Check "**Android Build Support**" and "**IL2CPP**" under the "Platforms" section.
    - Depending on OS, the "IL2CPP" module will either say "Windows Build Support (IL2CPP)" or "Mac Build Support (IL2CPP)".
  - Continue and install.



Install Unity 2023.1.6f1

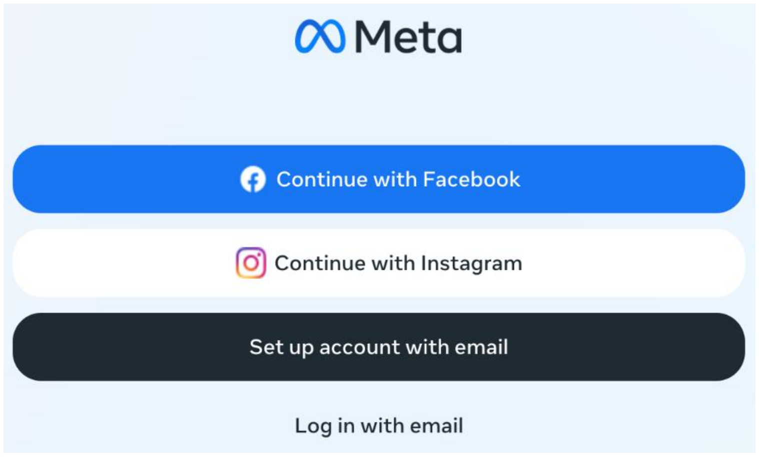
Add modules — Required: 15.13 GB Available: 60.37 GB

DEV TOOLS	DOWNLOAD SIZE	SIZE ON DISK
☐ Visual Studio for Mac	1.23 GB	3.57 GB

PLATFORMS	DOWNLOAD SIZE	SIZE ON DISK
✓ Android Build Support	680.87 MB	2.15 GB
└ ✓ OpenJDK	112.97 MB	219.56 MB
└ ✓ Android SDK & NDK Tools	1.76 GB	4.66 GB
☐ iOS Build Support	423.83 MB	1.09 GB
☐ tvOS Build Support	417.13 MB	1.08 GB

Meta Quest Mobile App

- ☐ Download the Meta Quest app on your smartphone.
- ☐ Choose one of the following login options:
  - "**Continue with Facebook**" or "**Continue with Instagram**" if you have an account with either one and would like to use it as your Meta login.
  - "**Set up account with email**" to create a Meta account.
  - "**Log in with email**" if you already have a Meta account.



Visual Studio Code

- ☐ Install VS Code (or skip this step if you have a preferred C# code editor)
  - o https://code.visualstudio.com/download

VR Headset Setup

IMPORTANT: In order to enable developer mode on the headset, your Meta account needs to be verified as a developer account. Follow this section to verify your account and learn how to enable developer mode.

- ☐ Follow the Meta Quest Device Setup guide using the following link:
  - o https://developer.oculus.com/documentation/native/android/mobile-device-setup/
- ☐ Open the Meta Quest mobile app and pair your headset.
- ☐ Enable developer mode.
  - o After verifying your developer account, developer mode can be enabled a few ways:
    - ▪ On the headset as detailed in the Device Setup guide.
    - ▪ In the Meta Quest mobile app.
      - • Go to **Menu** -> **Devices** -> select your device -> **Headset Settings** -> **Developer Mode**. Toggle on developer mode.
    - ▪ During MQDH device setup (refer to the Meta Quest Developer Hub (Mac only) section).

Meta Application

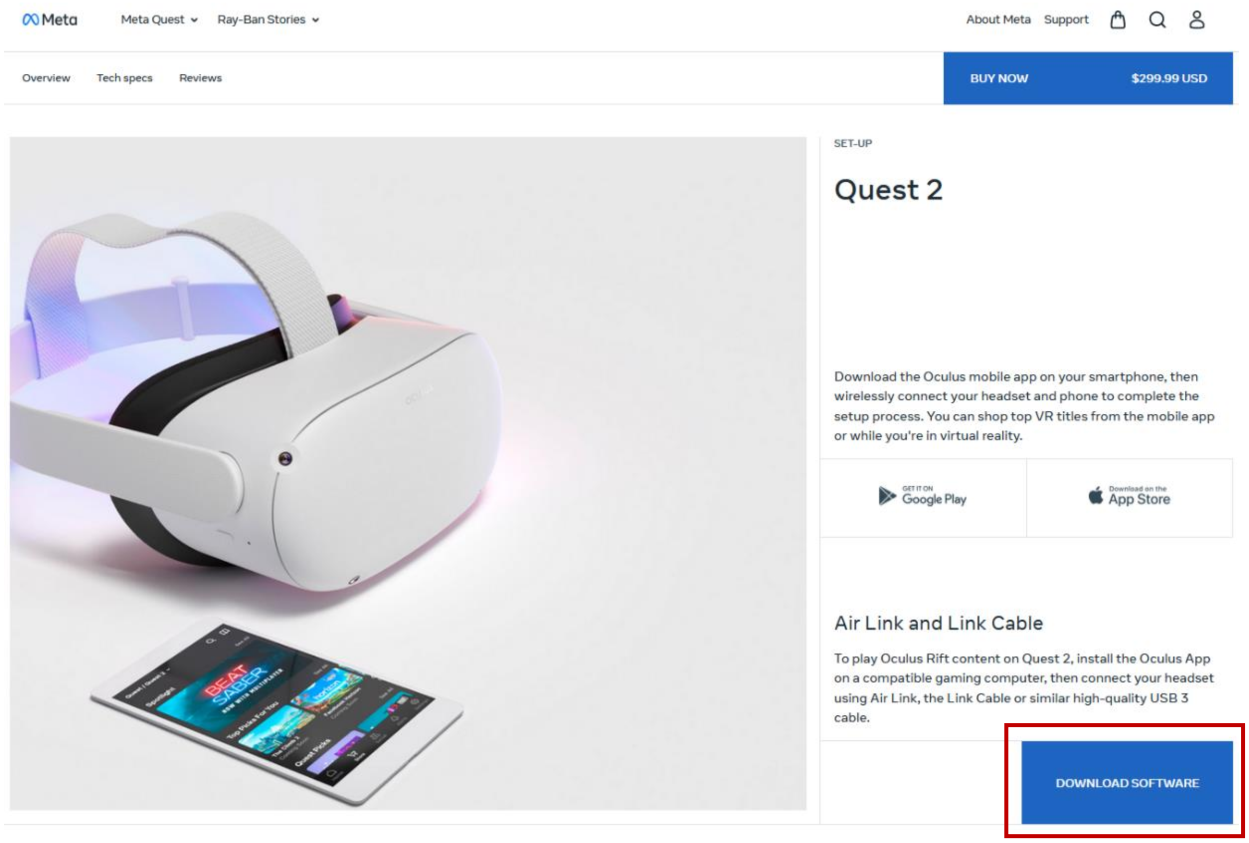
IMPORTANT: This section contains steps for two different applications: one for Windows users, one for Mac users. Follow the section that corresponds with your PC's OS.

Oculus App (Windows only)

NOTE: This app is only offered on Windows. Throughout creation of the Unity project, this application allows the user to test their project without having to build the apk file.

- ☐ Use this link to go to the Oculus app download page: https://www.meta.com/quest/setup/?utm_source=www.meta.com&utm_medium=dollyredirect

- ☐ Scroll until you find **Quest 2**, click "**Download Software**", then complete installation.



Quest 2

Download the Oculus mobile app on your smartphone, then wirelessly connect your headset and phone to complete the setup process. You can shop top VR titles from the mobile app or while you're in virtual reality.

Air Link and Link Cable

To play Oculus Rift content on Quest 2, install the Oculus App on a compatible gaming computer, then connect your headset using Air Link, the Link Cable or similar high-quality USB 3 cable.

- ☐ Launch the Oculus application.
- ☐ Log in with your Meta account (use the same account from the Meta Quest Mobile App section).
- ☐ Navigate to **Devices**, plug-in the headset and follow instructions to connect the device.
- ☐ Go to **Settings** -> **General** -> enable the "**Unknown Sources"** setting, then confirm to allow content from unknown sources.
- ☐ Make sure developer mode is enabled (refer to the VR Headset Setup section)

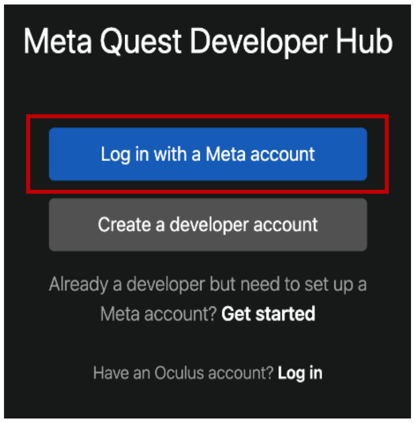
NOTE: In order to test play your project, have this app open on your PC, plug in the headset, then open "Quest Link" on the headset. With your project open in Unity Editor, press the play button at the top of the screen, then put on the headset to see what the player sees.

Meta Quest Developer Hub (Mac only)

NOTE: This application works with Windows, but isn't necessary for this project since Windows users will be using the Oculus App.

- ☐ Download and install MQDH for Mac.
    - Mac: https://developer.oculus.com/downloads/package/oculus-developer-hub-mac/
    - Windows: https://developer.oculus.com/downloads/package/oculus-developer-hub-win/
- ☐ Launch the MQDH application.

☐ Click the option “**Log in with a Meta account**”.

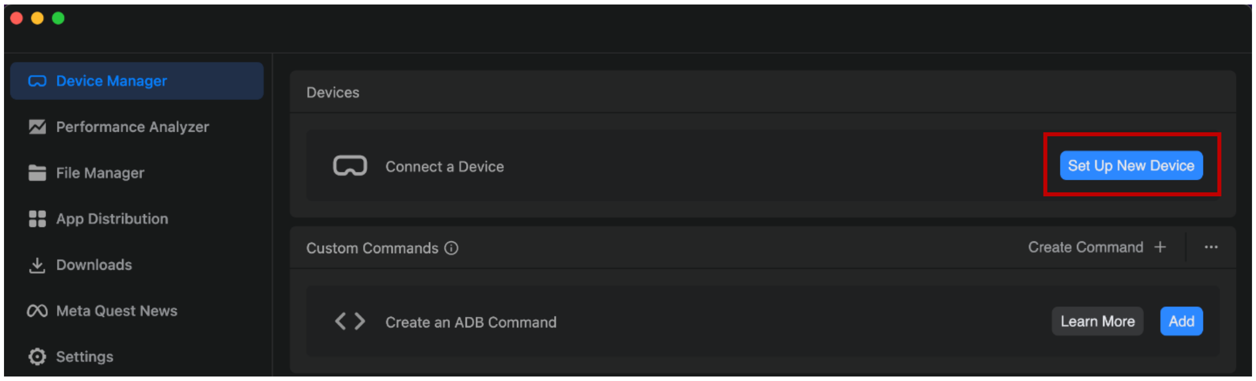

Meta Quest Developer Hub
Log in with a Meta account
Create a developer account
Already a developer but need to set up a Meta account? Get started
Have an Oculus account? Log in

☐ Log in with your Meta account (use the same account from the Meta Quest Mobile App section).

☐ Navigate to **Device Manager** and plug in your headset.

☐ Click “**Set Up New Device**”. Follow prompts and instructions.

- Enable developer mode when prompted.

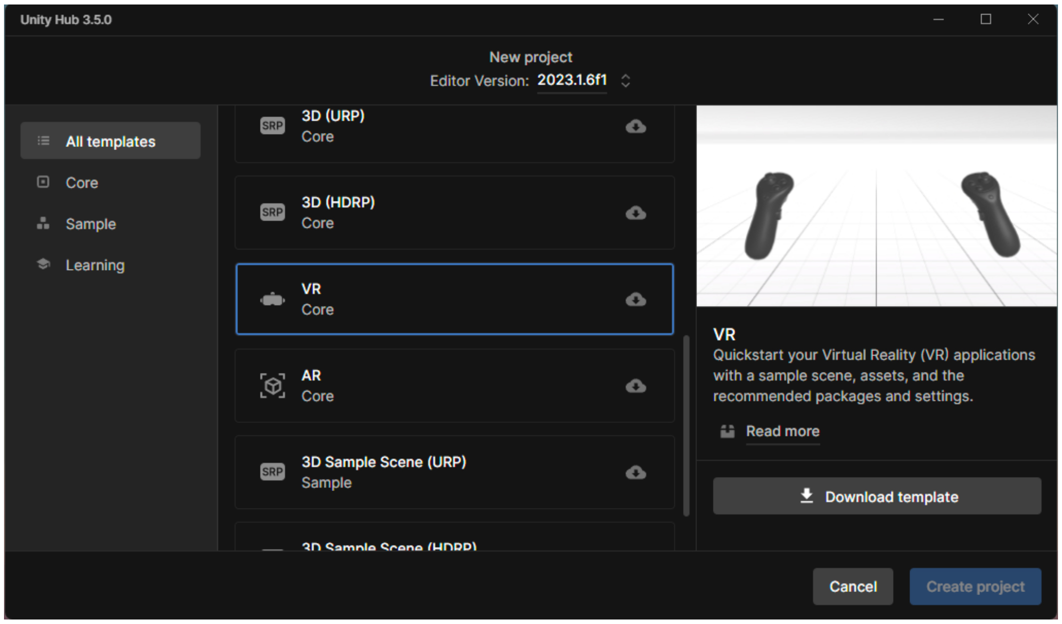


☐ Put on the headset. Allow access to the prompts that appear.

Create a Hello World Program for VR

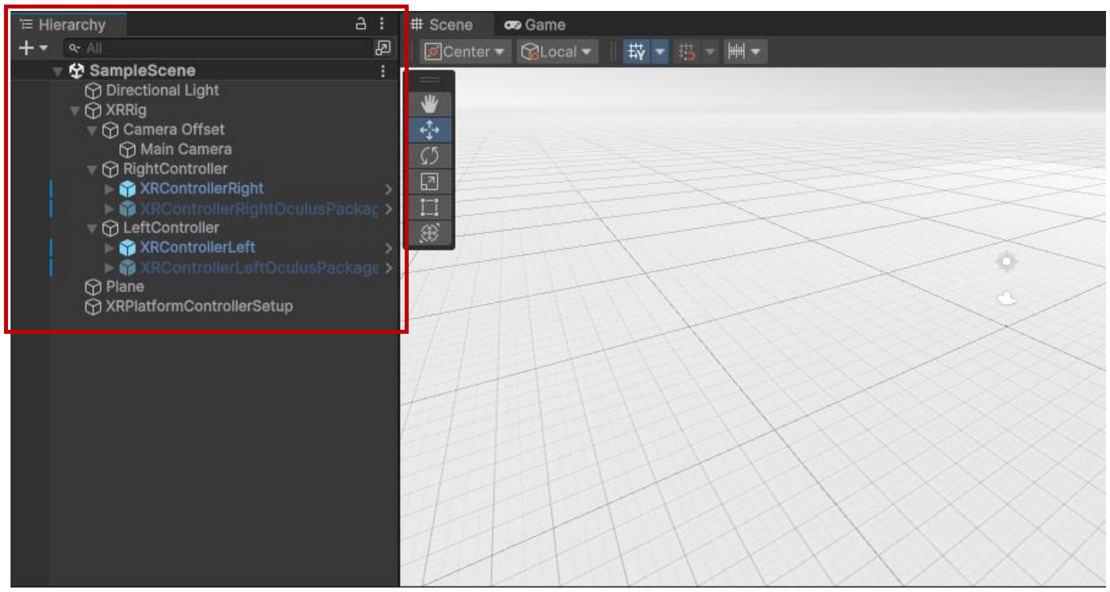
Creating a Project

- ☐ Launch Unity Hub.
- ☐ Go to **Projects**, then select “**New Project**”.
- ☐ Scroll through the templates and choose the VR template.
    - ○ Click “**Download template**”, if needed.

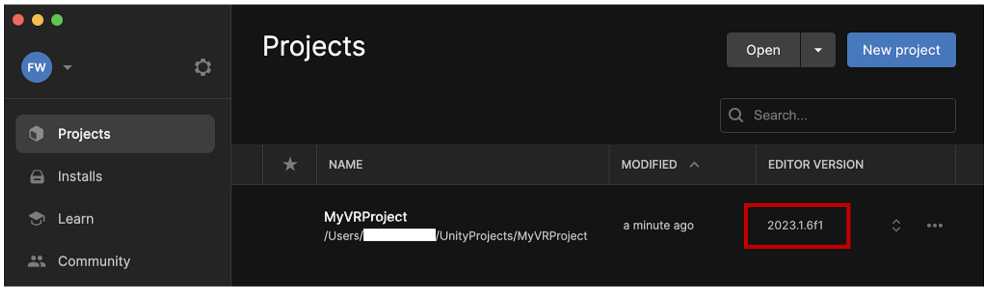


Quickstart your Virtual Reality (VR) applications with a sample scene, assets, and the recommended packages and settings.

- ☐ Name the project, select a file location, ensure the correct version of Editor is selected, then click “**Create project”**.
    - ○ Ex: C:\Users\johnsmith\UnityProjects
- ☐ **IMPORTANT:** After the project is created and it’s opened in Unity Editor, find the **Hierarchy** panel. Check to make sure yours matches the screenshot below by expanding the same dropdowns as shown.



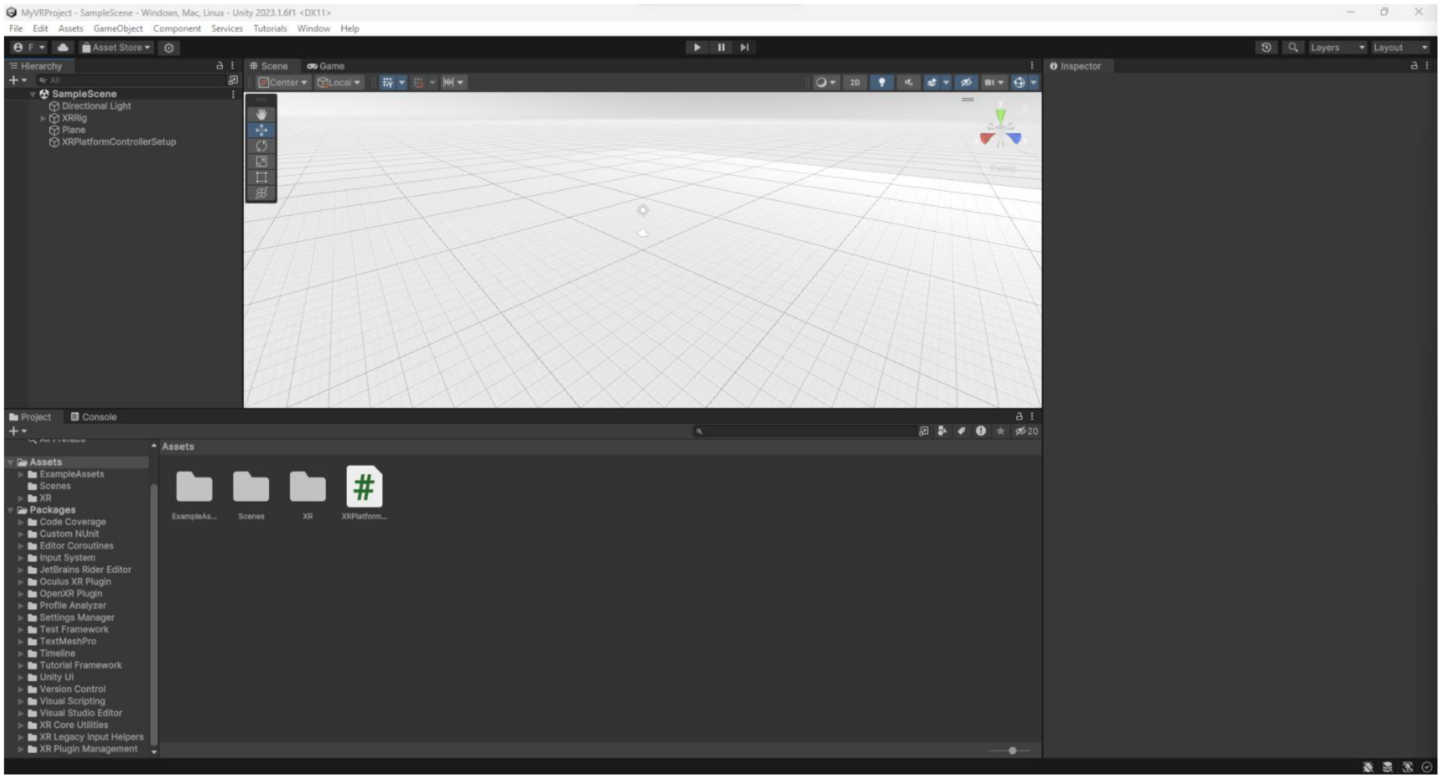
- If yours does not match:
  - Check that the correct version of Editor was selected for the project.
    - In the Unity Hub **Projects** screen, find your project and make sure the latest non-LTS version of Editor is selected.
  - Check that the correct version of Unity Editor was installed (refer to the Unity section).



	NAME	MODIFIED	EDITOR VERSION
	MyVRProject /Users/[redacted]/UnityProjects/MyVRProject	a minute ago	2023.1.6f1

☐ With your project open in Editor, take note of the following four panes:

- ***Hierarchy*** pane
- ***Project*** pane
- ***Inspector***
- Scene ***Visualizer***

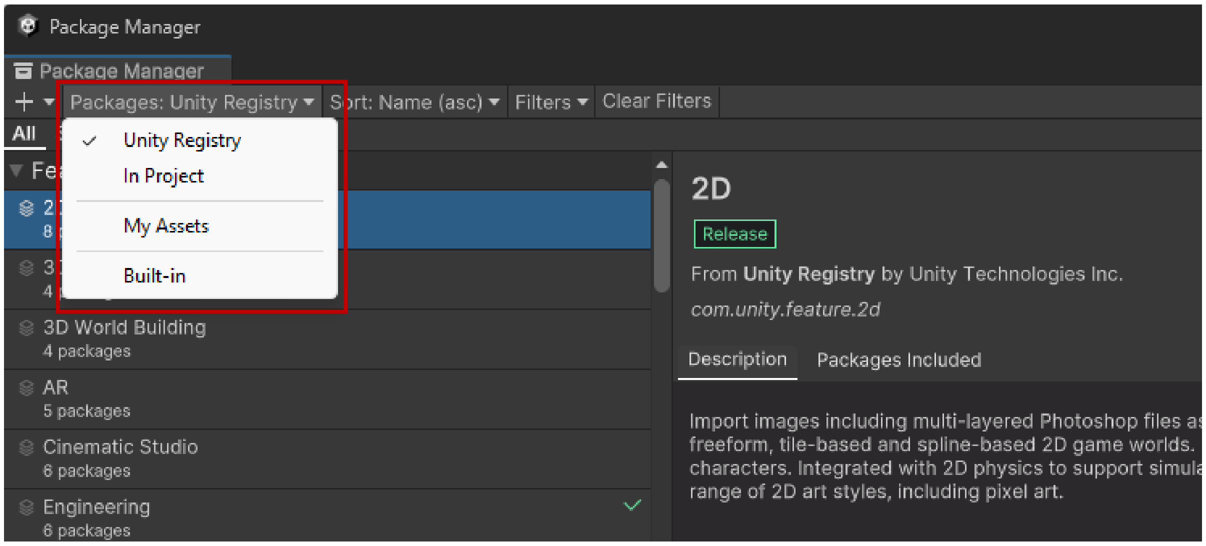


Packages and Initial Setup

Package Manager

☐ Go to *Window* -> ***Package Manager***.

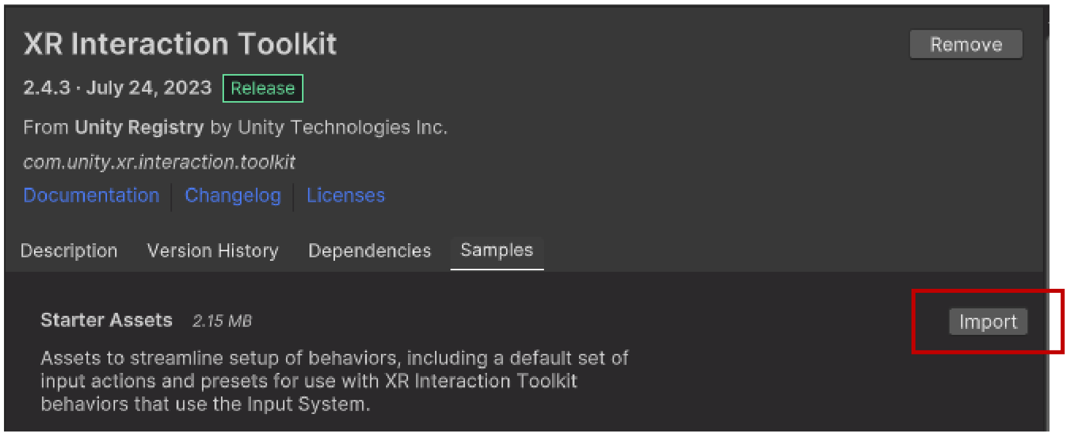
☐ In **Package Manager**, click the "Packages" dropdown and select "Unity Registry".



Import images including multi-layered Photoshop files as
freeform, tile-based and spline-based 2D game worlds.
characters. Integrated with 2D physics to support simula
range of 2D art styles, including pixel art.

☐ Search for and install the following packages:

- XR Plugin Management
- XR Interaction Toolkit
    - In the screen for this package, click the "Samples" tab, then import "Start Assets".



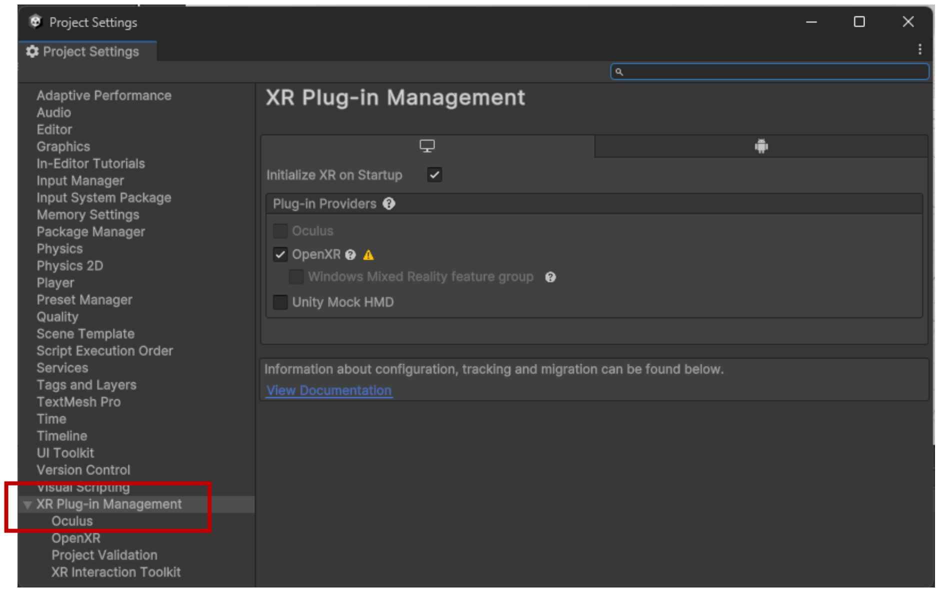
Assets to streamline setup of behaviors, including a default set of
input actions and presets for use with XR Interaction Toolkit
behaviors that use the Input System.

☐ Close **Package Manager**.

Project Settings

☐ Go to *Edit* -> ***Project Settings***.

☐ Find the **XR Plug-in Management** settings.



Project Settings

Adaptive Performance
Audio
Editor
Graphics
In-Editor Tutorials
Input Manager
Input System Package
Memory Settings
Package Manager
Physics
Physics 2D
Player
Preset Manager
Quality
Scene Template
Script Execution Order
Services
Tags and Layers
TextMesh Pro
Time
Timeline
UI Toolkit
Version Control
Visual Scripting
XR Plug-in Management
Oculus
OpenXR
Project Validation
XR Interaction Toolkit

XR Plug-in Management

Initialize XR on Startup ✓

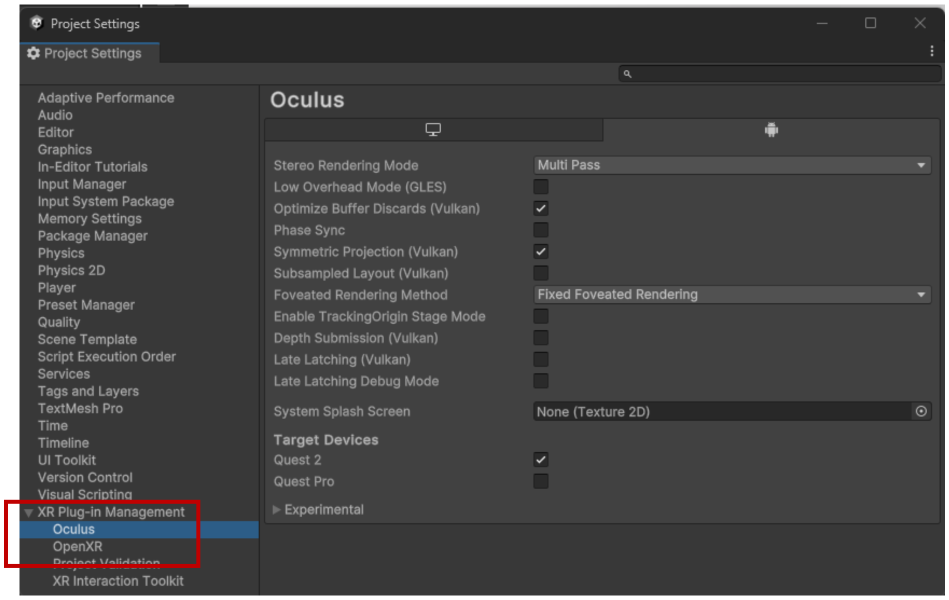
Plug-in Providers

- ☐ Oculus
- ✓ OpenXR
- ☐ Windows Mixed Reality feature group
- ☐ Unity Mock HMD

Information about configuration, tracking and migration can be found below.
View Documentation

☐ Click the **Android** tab. Check the “Oculus” box under **Plug-in Providers**.

☐ On the left under XR Plug-in Management, open the **Oculus** settings.



Project Settings

Adaptive Performance
Audio
Editor
Graphics
In-Editor Tutorials
Input Manager
Input System Package
Memory Settings
Package Manager
Physics
Physics 2D
Player
Preset Manager
Quality
Scene Template
Script Execution Order
Services
Tags and Layers
TextMesh Pro
Time
Timeline
UI Toolkit
Version Control
Visual Scripting
XR Plug-in Management
Oculus
OpenXR
Project Validation
XR Interaction Toolkit

Oculus

Stereo Rendering Mode	Multi Pass
Low Overhead Mode (GLES)	☐
Optimize Buffer Discards (Vulkan)	✓
Phase Sync	☐
Symmetric Projection (Vulkan)	✓
Subsampled Layout (Vulkan)	☐
Foveated Rendering Method	Fixed Foveated Rendering
Enable TrackingOrigin Stage Mode	☐
Depth Submission (Vulkan)	☐
Late Latching (Vulkan)	☐
Late Latching Debug Mode	☐
System Splash Screen	None (Texture 2D)
Target Devices	
Quest 2	✓
Quest Pro	☐

Experimental

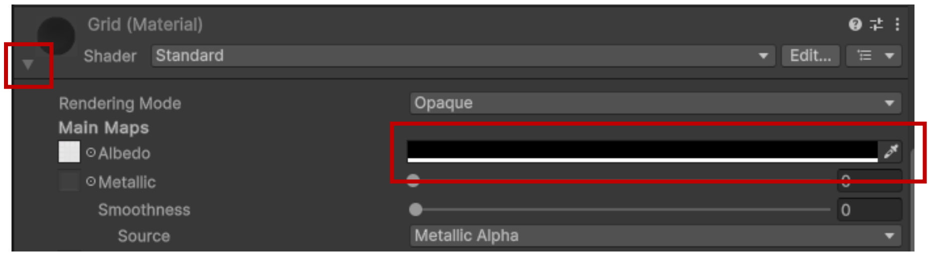
☐ In the **Android** tab, under the **Target Devices** section, check the “Quest Pro” box.

☐ Close **Project Settings**.

Plane Setup

☐ In the *Hierarchy* pane, under SampleScene, click the **Plane** object.

☐ In the *Inspector* window, expand **Grid (Material)**, then change the material color to black.



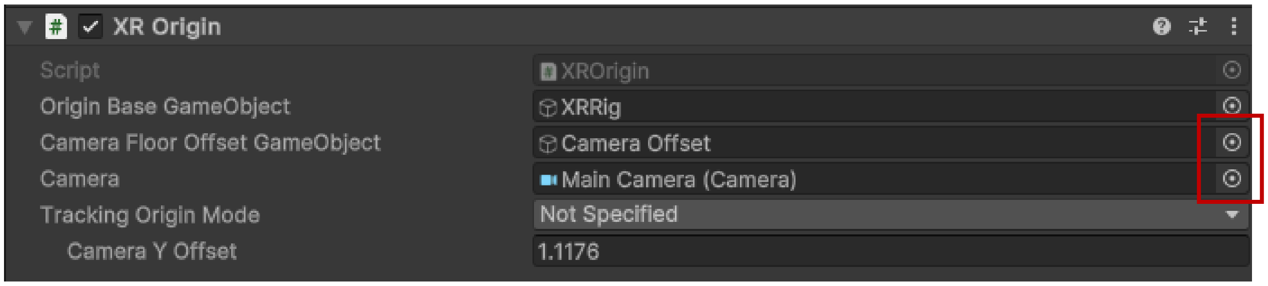
Grid (Material)

Shader: Standard   Edit...

Rendering Mode	Opaque
Main Maps	
Albedo	
Metallic	0
Smoothness	0
Source	Metallic Alpha

Main Player Camera (Rig)

- ☐ In the *Hierarchy* pane, expand **XRRig**, then **Camera Offset** until you see **Main Camera**.
- ☐ Click the **XRRig** object from the *Hierarchy* pane.
- ☐ In the *Inspector* for XRRig, click "**Add Component**".
- ☐ Search for "XR Origin" and click it.
- ☐ Under the **XR Origin** component, click the bullseye icon for:
    - "Camera Floor Offset GameObject". Find "Camera Offset" under the Scene tab and select it.
    - "Camera". Find "Main Camera" under the Scene tab and select it.


XR Origin

Script	XROrigin
Origin Base GameObject	XRRig
Camera Floor Offset GameObject	Camera Offset
Camera	Main Camera (Camera)
Tracking Origin Mode	Not Specified
Camera Y Offset	1.1176

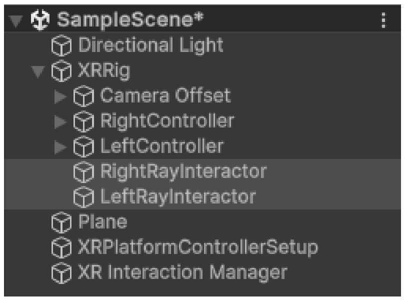
- ☐ Click the **Main Camera** object from the *Hierarchy* pane.
- ☐ In the *Inspector* for Main Camera, under the **Camera** component, change "Clipping Planes" value for *Near* to 0.1.

Controllers

- ☐ In the *Hierarchy* pane, expand **XRRig** until you see **RightController** and **LeftController**.
- ☐ Select both **RightController** and **LeftController** using shift.
- ☐ In the *Inspector*, click "**Add Component**".
- ☐ Search for "XR Controller (Device-based)" and click it.
- ☐ Select only **LeftController**.
- ☐ In the *Inspector*, under the recently added component, change "Controller Node" to "Left Hand".
- ☐ Do the same for **RightController** and make sure it's set to "Right Hand".

Rays

- ☐ Right-click **XRRig** from the *Hierarchy*. Select *XR -> Device-based -> Ray Interactor*.
    - Do this twice to create one for each hand.
- ☐ Name one "RightRayInteractor" and the other "LeftRayInteractor".

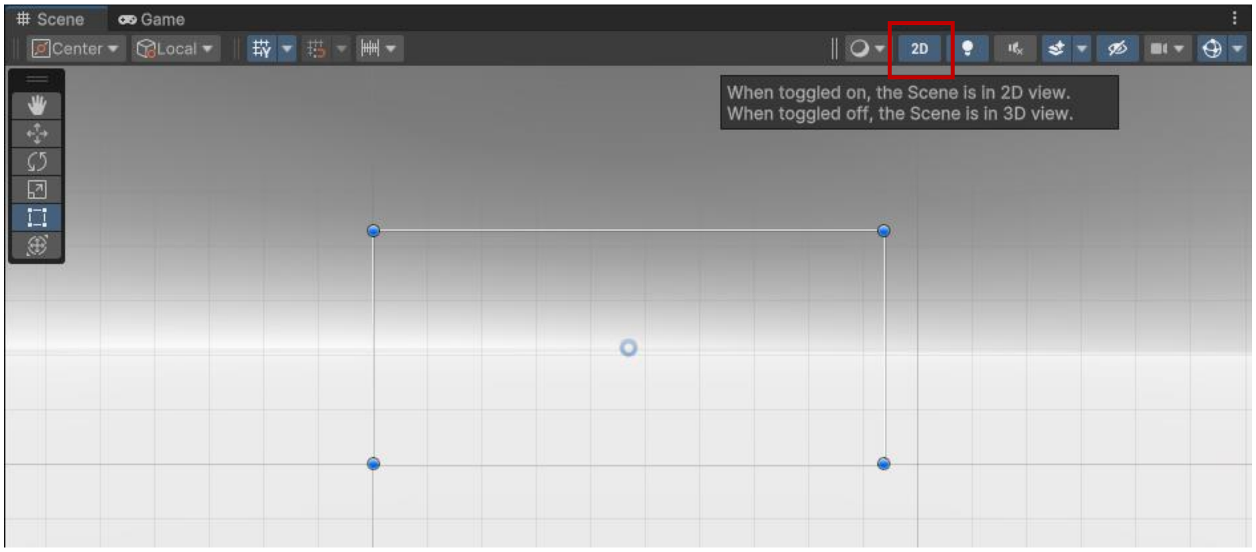

```
SampleScene*
  Directional Light
  XRRig
    Camera Offset
    RightController
    LeftController
    RightRayInteractor
    LeftRayInteractor
  Plane
  XRPlatformControllerSetup
  XR Interaction Manager
```

- ☐ In the *Inspector* for each ray interactor, under the **XR Controller** component, change "Controller Node" to right or left hand, respectively.
- ☐ Select both **RightRayInteractor** and **LeftRayInteractor** using shift.

- ☐ In the *Inspector*, under the **XR Ray Interactor** component, find the "Raycast Configuration" section.
- ☐ Click the "Raycast Mask" dropdown, select "Nothing", then select "UI" (this way only "UI" is checked).

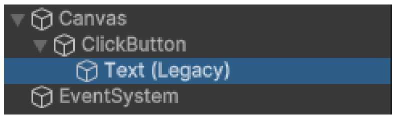
Canvas

- ☐ Right-click in the *Hierarchy* pane. Select *UI -> Canvas*.
- ☐ Double-click the **Canvas** object, then select the **2D** option in the *Visualizer* window.



When toggled on, the Scene is in 2D view.
When toggled off, the Scene is in 3D view.

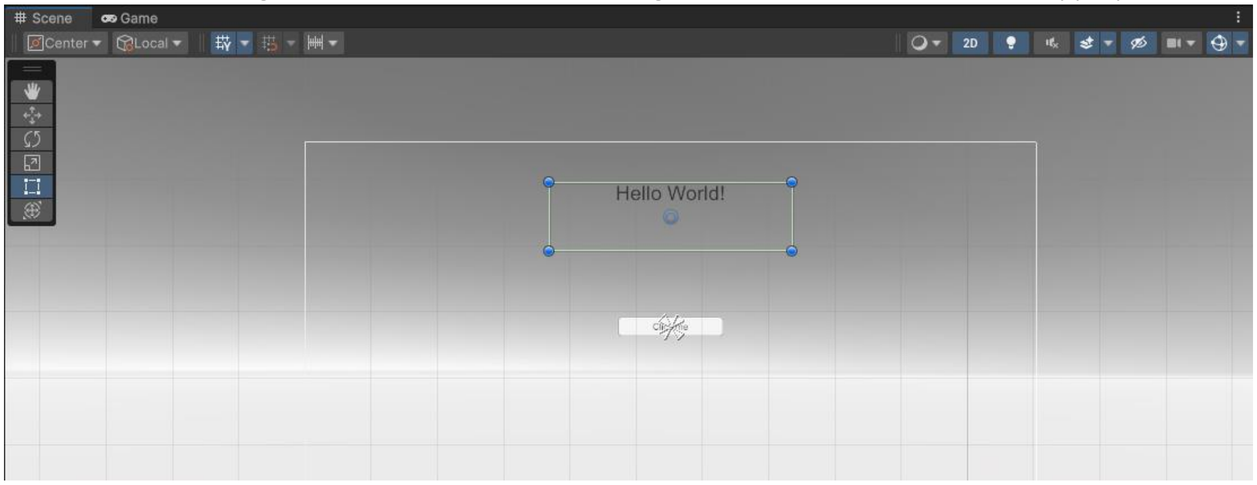
- ☐ In the *Hierarchy* pane, right-click *Canvas -> UI -> Legacy -> Button*. Name it "ClickButton".
- ☐ Expand **ClickButton**. Select the "Text" object.



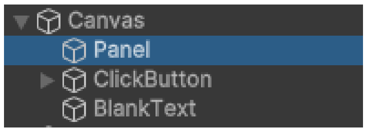
Canvas
ClickButton
Text (Legacy)
EventSystem

- ☐ In the *Inspector*, under the **Text** component, change the text to say "Click me".
- ☐ In the *Hierarchy* pane, right-click *Canvas -> UI -> Legacy -> Text*. Name it "BlankText".
- ☐ In the *Visualizer* window, re-arrange the objects.
  - **NOTE:** Since we double-clicked Canvas, the *Visualizer* window centered on it and Canvas remains highlighted. Clicking the objects under Canvas in the *Hierarchy* pane will highlight them. They can be moved and re-sized here.
  - Drag the corner of the **BlankText** textbox to make it bigger.
  - Move **BlankText** towards the middle-top of Canvas. Leave **ClickButton** in the center.
- ☐ Select **BlankText** from the *Hierarchy* pane.
- ☐ In the *Inspector*, under the **Text** component, change the settings to center-align and font 30.
- ☐ Test that the **BlankText** textbox is big enough:
  - **IMPORTANT:** If the textbox is not large enough for all the text to show, it will run into issues with our script and appear blank.

- Change the text to “Hello World!”. Drag the corner of the textbox to an appropriate size.

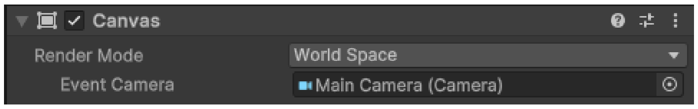


- ☐ Change the text for **BlankText** to be blank.
- ☐ In the *Hierarhcy* pane, right-click *Canvas -> UI -> Panel*.
- ☐ Drag the **Panel** object to the top of the hierarchy within **Canvas**.
    - **NOTE:** This layers the objects so that the panel is behind the button and text.



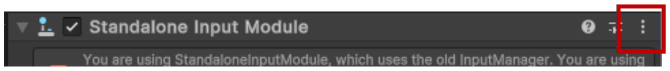
Canvas
Panel
ClickButton
BlankText

- ☐ Select **Canvas** from the *Hierarchy* pane.
- ☐ In the *Inspector* for Canvas:
    - Under the **Canvas** component:
        - Change “Render Mode” to “World Space”.
        - Click the bullseye icon for “Event Camera”, then select “Main Camera” under the “Scene” tab.



Canvas
Render Mode: World Space
Event Camera: Main Camera (Camera)

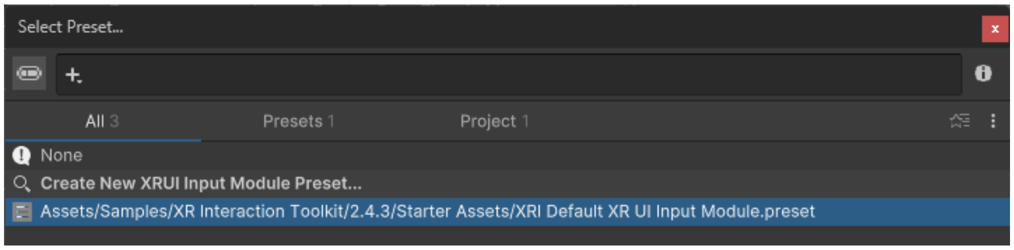
    - Under the **Rect Transform** component:
        - Set “Scale” for X, Y, and Z = 0.001
        - Set Pos Y and Pos Z = 1
        - Set Pos X = 0
    - Click “**Add Component**”. Find “Tracked Device Graphic Raycaster” and click it.
- ☐ Select **EventSystem** from the *Hierarchy* pane.
- ☐ In the *Inspector* for EventSystem:
    - Remove the “Standalone Input Module” component by clicking the 3-dot icon, then “Remove Component”.



Standalone Input Module
You are using StandaloneInputModule, which uses the old InputManager. You are using

    - Click “**Add Component**”. Find “XR UI Input Module” and click it.
    - On this new component, click the icon that looks like sliders (to the left of the 3-dot icon).

- Select “XRI Default XR UI Input Module”.



Assets/Samples/XR Interaction Toolkit/2.4.3/Starter Assets/XRI Default XR UI Input Module.preset

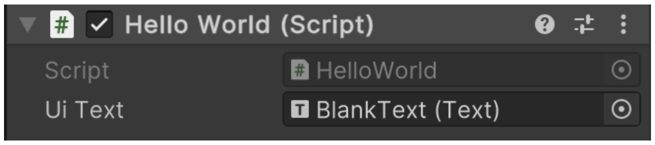
Script

- ☐ In the *Project* pane, open the “Assets” folder.
- ☐ Right-click in the “Assets” folder -> *Create* -> *C# Script*. Name it “HelloWorld”.
- ☐ Open the “HelloWorld.cs” file in VS Code.
- ☐ Edit the code to match the following:

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

public class NewBehaviourScript : MonoBehaviour
{
    public Text uiText;

    // Start is called before the first frame update
    void Start()
    {

    }

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

    }

    public void SayHello()
    {
        uiText.text = "Hello World!";
    }
}
```

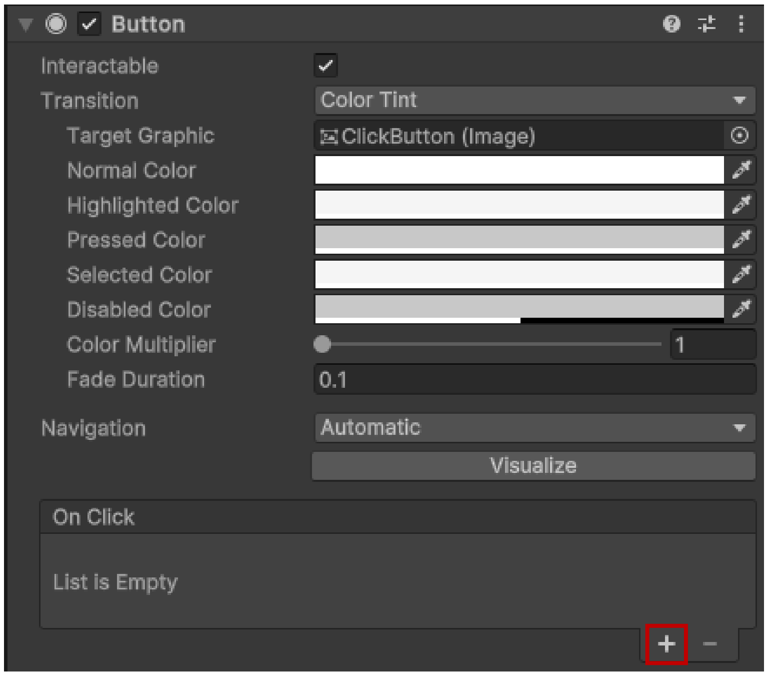
- ☐ Be sure to save, then close VS Code.
- ☐ In the *Hierarchy* pane, select the **Canvas** object.
- ☐ In the *Inspector* for Canvas, click “**Add Component**”. Find and click the “Hello World” script.

☐ Under this component, click the bullseye icon for “Ui Text”. Choose the “BlankText” object under the Scene tab.


Hello World (Script)
Script: HelloWorld
Ui Text: BlankText (Text)

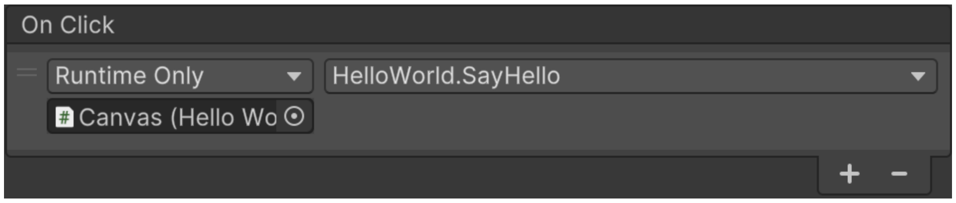
☐ In the *Hierarchy* pane, select the **ClickButton** object.

☐ In the *Inspector* for ClickButton, find the **Button** component. In the box labeled “**On Click**”, press the “**+**”.


Button
Interactable: ✓
Transition: Color Tint
Target Graphic: ClickButton (Image)
Normal Color
Highlighted Color
Pressed Color
Selected Color
Disabled Color
Color Multiplier: 1
Fade Duration: 0.1
Navigation: Automatic
Visualize
On Click
List is Empty

☐ Click the bullseye icon and set the object to the “Canvas” object (which is found under the Scene tab).

☐ Set the function to “HelloWorld.SayHello” by clicking the function dropdown.


On Click
Runtime Only
HelloWorld.SayHello
Canvas (Hello Wo

Build and Run

☐ With the VR headset turned on and logged into your Meta account, use the link cable to plug it into your PC.

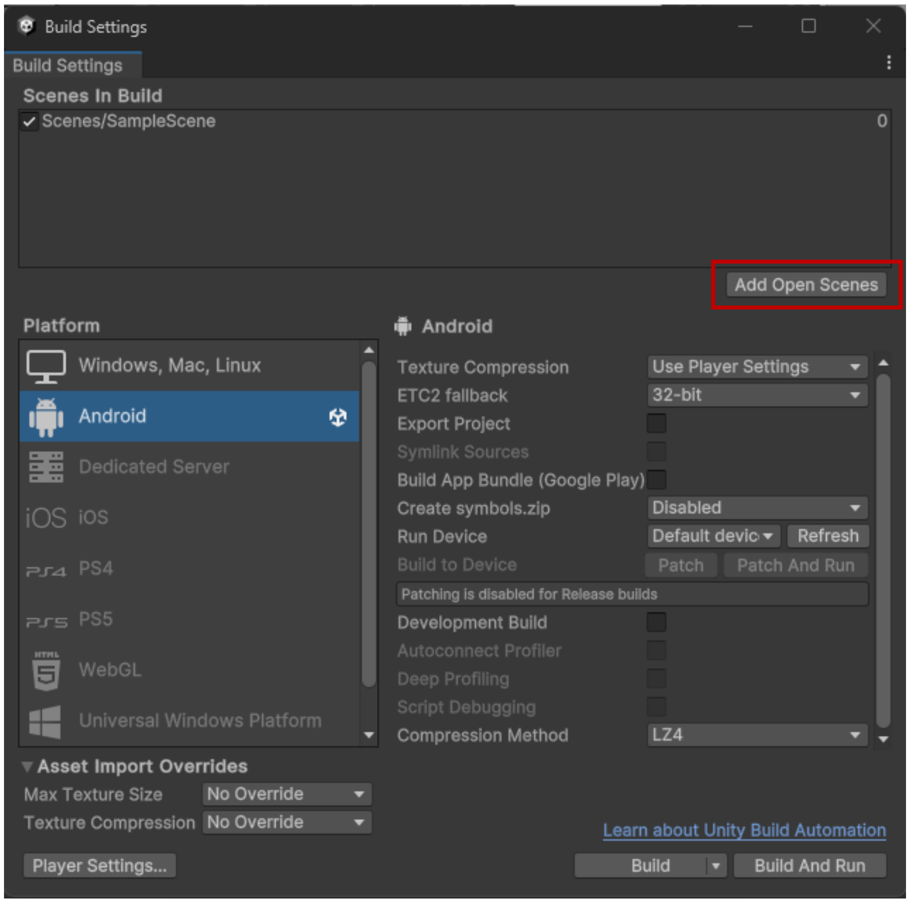
  - Make sure the Meta application is open and running on your PC (refer to Meta Application).

☐ Put on the headset. Based on which Meta application is being used, prompts on the headset will be different. Accept any prompts and follow instructions if they appear.

☐ In Unity, go to *File* -> ***Build Settings***.

☐ Under **Platform**, select **Android**, then click the "**Switch Platform**" button on the bottom right.

☐ If the nothing is listed in the **Scenes In Build** box, add your scene by clicking the "**Add Open Scenes**" button.


Build Settings

Scenes In Build

✓ Scenes/SampleScene 0

Platform: Android

Texture Compression	Use Player Settings
ETC2 fallback	32-bit
Export Project	☐
Symlink Sources	☐
Build App Bundle (Google Play)	☐
Create symbols.zip	Disabled
Run Device	Default devic
Build to Device	Patch / Patch And Run
Development Build	☐
Autoconnect Profiler	☐
Deep Profiling	☐
Script Debugging	☐
Compression Method	LZ4

Patching is disabled for Release builds

Asset Import Overrides

Max Texture Size	No Override
Texture Compression	No Override

☐ Click "**Build**" on the bottom right.

☐ Choose where builds are to be located:

- File Explorer should pop-up and show your project directory.
- In your project directory, create a folder named "Builds". Go into the "Builds" folder.
- Name the file.
    - Ex: TestBuild1
- Press "Save" to continue.
- The project will begin to build after choosing the file location. This may take a bit of time for the first build.

☐ Click "**Build And Run**" on the bottom right. Put on your headset and test your project!

- **NOTE:** Every time you rebuild the project, File Explorer will appear. You can overwrite the previous build file by keeping the same file name. Or you can change the name to create a new build file. The project will automatically open and start on your headset after the build is completed.
- **IMPORTANT:** This will only work if developer mode is enabled on your headset. Refer to the VR Headset Setup section for steps on how to enable developer mode.

Resources

The AR VR Guy. (2023, February 12). *Complete 2023 Guide: Setting up Meta Quest Pro/Quest 2 and Unity for VR Development* [Video]. YouTube. https://youtu.be/xX7NwVChGMM?si=wq7FlvnHRY818Zg6

Meta. *Meta Quest Developer Resources*. https://developer.oculus.com/resources/

RCode. (2022, May 29). *Unity 3D & Quest 2 : Programming Basics | hello world!* [Video]. YouTube. https://youtu.be/1h13LkdyUyY?si=x9fuhJ5ExXKF2s1z

ThirteeNov Coding Vlog. (2022, November 5). *Make your very first Hello World program for Oculus using Unity* [Video]. YouTube. https://youtu.be/_8taIktRffs?si=KmzxOBYod0X47t2N

Valem. (2020, April 8). *Introduction to VR in Unity - PART 1 : VR SETUP* [Video]. YouTube. https://youtu.be/gGYtahQjmWQ?si=H7vL3k2GE077IDBB

Valem Tutorials. (2022, September 4). *How to Make a VR Game in Unity 2022 - PART 7 - User Interface* [Video]. YouTube. https://youtu.be/yhB921bDLYA?si=iSLW60LTmIaFsed