

# XR Support

---

This github repository includes a full unity project that is setup and ready to go for XR Management for use with Oculus. However, it is advised to instead use the unity package provided in releases and follow the 'XR Management in a new project' instructions below.

**Latest Unity version tested: 2019.3.13f1**

**Devices tested: Oculus Quest**

**Platforms tested: Android (Quest), Windows 64 (Virtual Desktop)**

## Features

---

### Prefabs

---

#### XR Player

- Drag and drop fully setup VR player
- Controller and headset tracking with XR Management
- Unity events for button presses (currently on primary and secondary buttons but more coming soon)

#### XR Player Oculus

- Hand tracking (including bones) using OVR
- Everything from XR Player Prefabs

#### InteractableCube

Example interactable object using 'XR Grab Interactable' script. To interact with this object use the XR Player and point a controller at the object. This will cast a ray to the object, use the trigger to pull the object towards you.

#### XRInputEventController

Example object using XR Input events to trigger spawning interactable cubes.

## Scenes

---

Provided in the package are example scenes for using the Prefabs.

#### InteractableCubeScene

This scene has a pink interactable cube that can be picked up by pointing a controller at the object and pressing the trigger.

More interactable cubes can be spawned by pressing the primary or secondary buttons on the controller. The color of the cube is determined by which button was pressed.

## XRSupportBuilder

---

XRSupportBuilder is included in this package. This includes quick and pain free building of a VR game for multi platform support.

Platforms currently supported: - Android (Oculus Quest) - PC (Windows 64 only) **Note: This is Windows 64 only as MacOS and Linux do not support the Vulkan or Direct3D11 graphics API's required for Oculus**

This also provides building of a Windows Installer using Inno Setup.

Start a build by going to File -> XRSupport -> Build All

This will build an Android APK, Windows 64 application, and a Windows 64 Installer application.

### Requirements

The XRSupportBuilder uses Inno Setup Tool to build a windows installer. As Unity Assets cannot contain applications this must be installed manually. Download it from [here](#).

By default this should install to 'C:\Users\USERNAME\AppData\Local\Programs\Inno Setup 6\'. If this is not the case then you will need to edit the builder script parameter in XRSupport/Editor/XRSupportBuilder.cs

```
private static string innoSetupFolder = System.Environment.GetFolderPath(System.Environment.SpecialFolder.ApplicationData) + @"..\Lo
```

The icon for the installer is assumed to be in Assets/XRSupport/logo.icon

The license is assumed to be in Assets/XRSupport/LICENSE

If you would these to be different for your project edit these script parameters:

```
private static string logoFolder = "Assets/XRSupport/Icon/";
private static string logoFile = "logo.ico";
private static string licenseFile = "Assets/XRSupport/LICENSE";
```

You can also change the build folder locations:

```
private static string buildFolderAndroid = "Build/Android/";
private static string buildFolderWin64 = "Build/Win64/";
```

And the scene that will be built:

```
private static string[] scenes = { "Assets/XRSupport/Scenes/InteractableCubeScene.unity" };
```

## XR Management with XR Support in a new project

---

Install XR Management Toolkit & XR Plugin Management from the package manager (Window -> Package Manager)

Make sure to show preview packages (Advanced -> Show preview packages)

Go to XR Management (Edit -> Project Settings -> XR Management)

Make sure to add Oculus SDK by pressing '+' (Do this for both android and PC)

Import 'Oculus Integration' from the Asset Store (See [here](#) for details on importing 'Oculus Integration'). It is advised to not import the SampleFramework from this package as it just has example scenes and dramatically increases build times.

Import XR Support Package from the releases section of this repository

This also brings a reduced version of the Oculus Integration package with to use the OVR Hand Prefab (this will show up as the folder 'Oculus' in your project)

Add 'XR Player Oculus' prefab to scene (XRSupport/Prefabs/XR Player Oculus)

Build all (File -> XRSupport -> Build All)

## Run in Virtual Desktop

---

Virtual Desktop allows you to run almost any game from Oculus Store or Steam VR via WiFi on the Oculus Quest.

Download the Virtual Desktop Streamer application on your computer from [here](#)

Install Virtual Desktop app using SideQuest [download sidequest Virtual Desktop App](#)

To run a Unity application that has been built for Oculus:

Startup the Virtual Desktop steamer (on computer)

Start the Virtual Desktop app (on Quest)

Select the PC in the app

Go to the taskbar icon for Virtual Desktop Streamer

Right click the icon and select 'inject game...'

Choose the exe in the build directory of this project

This should run the application inside the Quest!

**Note:** *Virtual Desktop cannot pass hand tracking events as it only simulates the controllers. For hand tracking on the PC you should use Oculus Link*

## Useful guides

---

Unity provides guide on using XR Management Toolkit [here](#)

## Unity SteamVR support

---

If you would like to use SteamVR in your unity project rather than XR Management then use this [github repository](#) instead.

This is necessary as Unity XR Management does not support SteamVR. When and if this is added, these two will be joined.