

Quickstart for Google Cardboard for Unity

10–12 Minuten

[Skip to main content](#)

- [Develop](#)
- [Reference](#)
- [Manufacturers](#)
- [Community](#)
- [Overview](#)
- [Branding guidelines](#)
- Android NDK
- [Quickstart](#)
- [Generate Vulkan files](#)
- iOS
- [Quickstart](#)
- [Generate Protocol buffer files](#)
- [Update screen params from new iOS devices](#)
- Unity
- [Quickstart](#)
- [Migrate from GVR](#)

Quickstart for Google Cardboard for Unity

bookmark_border Stay organized with collections Save and categorize content based on your preferences.

- On this page
- [Set up your development environment](#)
- [Import the SDK and create a new project](#)
- [Configure HelloCardboard scene](#)
- [Configure Android build settings](#)
- [Player Settings](#)
- [Build your project](#)
- [Configure iOS project settings](#)
- [Player Settings](#)
- [Build your project](#)
- [Recenter](#)
- [Turn VR mode on and off](#)
- [Next steps](#)
-

This guide shows you how to use the [Google Cardboard XR Plugin for Unity](#) for Unity to create your own Virtual Reality (VR) experiences.

You can use the Cardboard SDK to turn a mobile phone into a VR platform. A mobile device can display 3D scenes with stereoscopic rendering, track and react to head movements, and interact with apps by detecting when the user presses the viewer button.

To get started, you'll use **HelloCardboard**, a demo game that demonstrates the core features of the Cardboard SDK. In the game, users look around a virtual world to find and collect objects. It shows you how to:

- Set up your development environment
- Download and build the demo app
- Scan the QR code of a Cardboard viewer to save its parameters
- Track the user's head movements

- Render stereoscopic images by setting the correct distortion for each eye
- Turn VR mode on and off

Set up your development environment

Software requirements:

- [Unity 2021.3.44f1](#) or later
- Make sure to include Android and iOS Build Support during installation.
- Make sure to install patch version 44f1 or later.
- [Git](#) must be installed and the `git` executable must be on the PATH environment variable. See [Unity's package manager git support](#) docs for more details.

Import the SDK and create a new project

Follow these steps to import the Unity SDK and create a new project.

1. Open Unity and create a new **3D** project.
2. In Unity, go to **Window > Package Manager**.
3. Click **+** and select **Add package from git URL**.
4. Paste `https://github.com/googlevr/cardboard-xr-plugin.git` into the text entry field.
The package should be added to the installed packages.
5. Navigate to the **Google Cardboard XR Plugin for Unity** package. In the **Samples** section, choose **Import into Project**.
The sample assets should be loaded into `Assets/Samples/Google Cardboard/<version>/Hello Cardboard`.

Configure HelloCardboard scene

1. Navigate to `Assets/Samples/Google Cardboard/<version>/Hello Cardboard/Scenes`, select **Add Open Scenes**, and choose **HelloCardboard** to open the sample scene.
2. Open the **Layers** menu and select **Edit Layers....**

3. Define a new layer called "Interactive".
4. Click the **Treasure** GameObject to open the Inspector window. Set its layer to be "Interactive". If a pop up window appears asking if you want to set layer to Interactive for all child objects as well, click "Yes, change children".
5. Click the **Player > Camera > CardboardReticlePointer** GameObject to open the Inspector window. In the "Cardboard reticle pointer" script, select "Interactive" as the **Reticle Interaction Layer Mask**.

Configure Android build settings

Navigate to **File > Build Settings**.

1. Select **Android** and choose **Switch Platform**.
2. Select **Add Open Scenes** and choose **HelloCardboard**.

Player Settings

Navigate to **Edit > Project Settings....**

Configure Resolution and Presentation settings

Navigate **Player > Resolution and Presentation**.

1. Set the **Default Orientation** to **Landscape Left** or **Landscape Right**.
2. Disable **Optimized Frame Pacing**.

Configure Other settings

Navigate to **Player > Other Settings**.

1. Choose OpenGL ES2, OpenGL ES3, or Vulkan, or any combination of them in **Graphics APIs**.
2. Select Android 8.0 'Oreo' (API level 26) or higher in **Minimum API Level**.
3. Select API level 33 or higher in **Target API Level**.
4. Select IL2CPP in **Scripting Backend**.

5. Select wanted architectures by choosing ARMv7, ARM64, or both in **Target Architectures**.
6. Select Require in **Internet Access**.
7. Select Input System Package (New) in **Active Input Handling**.
8. Specify your company domain under **Package Name**.
9. If Vulkan was selected as **Graphics API**:
 - Clear **Apply display rotation during rendering** checkbox in **Vulkan Settings**.
 - If the Unity version is 2021.2 or later, Select ETC2 in **Texture compression format**.
10. If the Unity version is 2023.1 or later, select Activity and clear GameActivity in **Application Entry Point**.

Configure Publishing settings

Navigate to **Player > Publishing Settings**.

1. In the **Build** section, select Custom Main Gradle Template and Custom Gradle Properties Template.
2. Add the following lines to the dependencies section of Assets/Plugins/Android/mainTemplate.gradle:

```
implementation 'androidx.appcompat:appcompat:1.6.1'
implementation 'com.google.android.gms:play-services-
vision:20.1.3'
implementation
'com.google.android.material:material:1.12.0'
implementation 'com.google.protobuf:protobuf-
javalite:3.19.4'
```

3. Add the following lines to Assets/Plugins/Android/gradleTemplate.properties:

```
android.enableJetifier=true
android.useAndroidX=true
```

Configure XR Plug-in Management settings

Navigate to **XR Plug-in Management**.

1. Select **Cardboard XR Plugin** under **Plug-in Providers**.

Build your project

Navigate to **File > Build Settings**.

1. Select **Build**, or choose a device and select **Build and Run**.

Configure iOS project settings

Navigate to **File > Build Settings**.

1. Select **iOS** and choose **Switch Platform**.
2. Select **Add Open Scenes** and choose **HelloCardboard**.

Player Settings

Navigate to **Edit > Project Settings...**

Configure Resolution and Presentation settings

Navigate to **Player > Resolution and Presentation**.

1. Set the **Default Orientation** to **Landscape Left** or **Landscape Right**.

Configure Other settings

Navigate to **Player > Other Settings**.

1. In **Camera Usage Description**, write Cardboard SDK requires camera permission to read the QR code (required to get the encoded device parameters)..
2. In **Target minimum iOS Version**, write 12.0.
3. Specify your company domain under **Package Name**.
4. Check **Top Edge**, **Left Edge** and **Right Edge** in **Defer system gestures on edges**.

Configure XR Plug-in Management settings

Navigate to **XR Plug-in Management**.

1. Select Cardboard XR Plugin under **Plug-in Providers**.

Build your project

Navigate to **File > Build Settings**.

1. Select **Build** or **Build and Run**.

Recenter

The [Cardboard SDK](#) lets you to recenter the head tracker using [Recenter\(\)](#).

Follow these steps to try it out using the sample application:

1. Move the device to the position you want to recenter (use as new looking forward head pose).
2. Hold the trigger of your Cardboard device active for at least three seconds.
3. Release the trigger.
4. The initial pose is now in the direction the camera is pointing.

Turn VR mode on and off

The [Unity XR Plugin Management API](#) lets you turn VR mode on or off for the [Google Cardboard XR Plugin for Unity](#). End-user documentation and usage examples are available in Unity's [End-user documentation](#).

The **VrMode** scene in HelloCardboard sample shows a basic usage of the aforementioned API. In this scene, VR mode can be turned off by tapping **exit**



, and can be turned on again just by tapping anywhere on the screen.

Check [VrModeController.cs](#) for details about how this is performed.

Next steps

- Review the [Cardboard branding guidelines](#).

All rights reserved. Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2025-06-10 UTC.