Set up your development environment

Software requirements:

- Unity 2020.3.36f1 or later
  - Make sure to include Android and iOS Build Support during installation.
- 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. Download Source from https://github.com/googlevr/cardboard-xr-plugin.git
3. Extract your download to specified folder.
4. In Unity, go to **Window** > **Package Manager**.
5. Add package from disk, find “Package.json” and select.
6. 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.
7. Navigate to Assets/Samples/Google Cardboard/<version>/Hello Cardboard/Scenes, select **Add Open Scenes**, and choose **HelloCardboard** to open the sample scene.

Note: <version> is the **X.Y.Z** semantic version number of the released package (for example, **1.1.0**).

Configuring Android project settings

Navigate to **File** > **Build Settings**.

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

Player Settings

Resolution and Presentation

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

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

Note: While supported by the Cardboard XR plugin, the **Portrait** and **Portrait Upside Down** orientations may not provide enough room for eye rendering on devices.

Other Settings

Navigate to **Project Settings** > **Player** > **Other Settings**.

1. Choose OpenGLES2, OpenGLES3, or Vulkan, or any combination of them in **Graphics APIs**.
2. Select Android 7.0 'Nougat' (API level 24) or higher in **Minimum API Level**.
3. Select API level 31 or higher in **Target API Level**.
4. Select IL2CPP in **Scripting Backend**.
5. Select desired architectures by choosing ARMv7, ARM64, or both in **Target Architectures**.
6. Select Require in **Internet Access**.
7. Specify your company domain under **Package Name**.
8. If Vulkan was selected as **Graphics API**:
    - Uncheck **Apply display rotation during rendering** checkbox in **Vulkan Settings**.
    - If the Unity version is 2021.2 or above, Select ETC2 in **Texture compression format**.

Note: It's possible to use a lower minimum API level by changing rendering API compatibility. For more information, see the SDK's **build.gradle**.**Note:** In case you are experiencing issues when selecting Vulkan as the graphics API, check the **Development Build** box in **Build Settings** and analyze the runtime logs looking for driver compatibility errors.

Publishing Settings

Navigate to **Project Settings** > **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:
3. implementation 'androidx.appcompat:appcompat:1.4.2'
4. implementation 'com.google.android.gms:play-services-vision:20.1.3'
5. implementation 'com.google.android.material:material:1.6.1'
   implementation 'com.google.protobuf:protobuf-javalite:3.19.4'
6. Add the following lines to Assets/Plugins/Android/gradleTemplate.properties:
7. android.enableJetifier=true
8. android.useAndroidX=true

Note: The dependencies needed may change between versions. If you want to use a version different from the most recent release, take a look at the history of the dependencies section in sdk/build.gradle of the Cardboard SDK repository.

XR Plug-in Management Settings

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

Configuring 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

Resolution and Presentation

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

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

Note: While supported by the Cardboard XR plugin, the **Portrait** and **Portrait Upside Down** orientations may not provide enough room for eye rendering on devices.

Other Settings

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

Note: If using an iPhone X, select the **Hide home button on iPhone X** option.

XR Plug-in Management Settings

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

Recentering

The Cardboard SDK allows 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.

Turning 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.

Install Android SDK

1. Download Android SDK API level 31 or newer.

   note: You can use Android Studio SDK manager to download this SDK.

2. Open download directory and copy the SDK to Unity3D SDK Path.
3. In Unity, go to **Edit**> **Preference>External Tools**.
4. Copy Android SDK Tools Installed with Unity Path.
5. Go to Unity SDK path and copy SDK from your Download folder to this Path.