

1 Virtual Reality

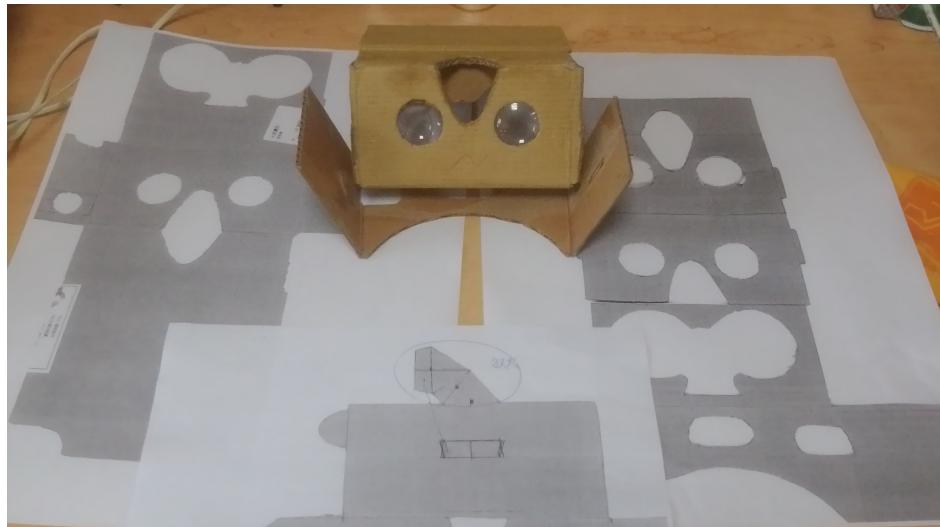
1.1 Unity (2019-1, 2019.1.4f1) with GVR Unity SDK v1.200.0

(2019/5/25)

- [Welcome Unity 2019.1.4f1 \(<https://unity3d.com>\)](https://unity3d.com)
- [GVR Unity SDK, 1.200.1 \(<https://github.com/googlevr/gvr-unity-sdk/releases>\)](https://github.com/googlevr/gvr-unity-sdk/releases)

No Problem, **Virtual Reality (VR)** is the hottest spot this year, 2016. Before we continue, there are some necessary requirement for implementing the work:

- [BluePrint of Cardboard \(<https://www.google.com/get/cardboard/developers/>\)](https://www.google.com/get/cardboard/developers/)

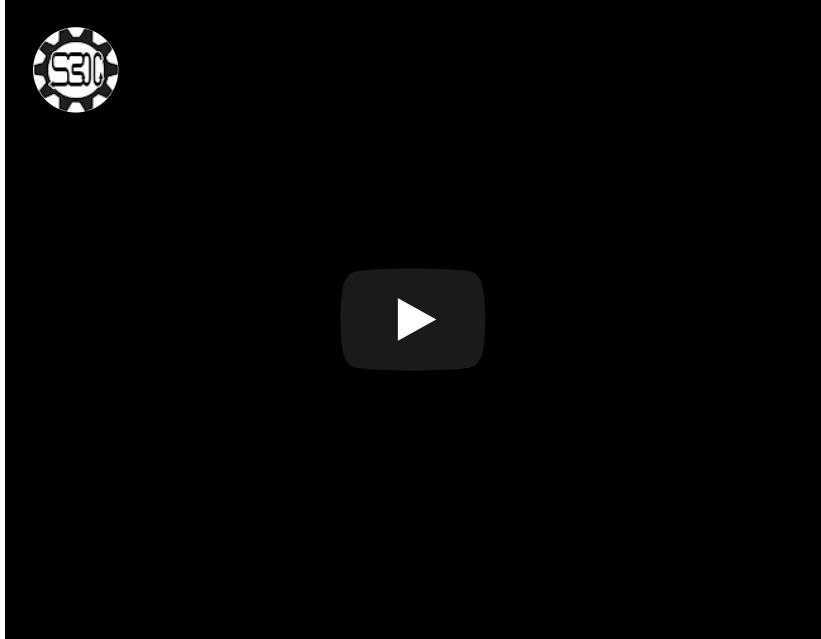


- [Unity \(<http://unity3d.com>\)](http://unity3d.com), Get the last release, 2019-1-4f1 (2019/05/25);
- [Cardboard SDK for Unity \(<https://github.com/googlesamples/cardboard-unity>\)](https://github.com/googlesamples/cardboard-unity), 1.200.0 (till 2019/05/25).
- a little idea, :-)

1.2 News about Google VR SDK (re-named GVR-unity-sdk since 2016)

2016, Google announced [Google VR SDK in Google IO \(GoogleVRSDK.ipynb\)](#)

1.3 Assemble Cardboard v2 in Two minutes

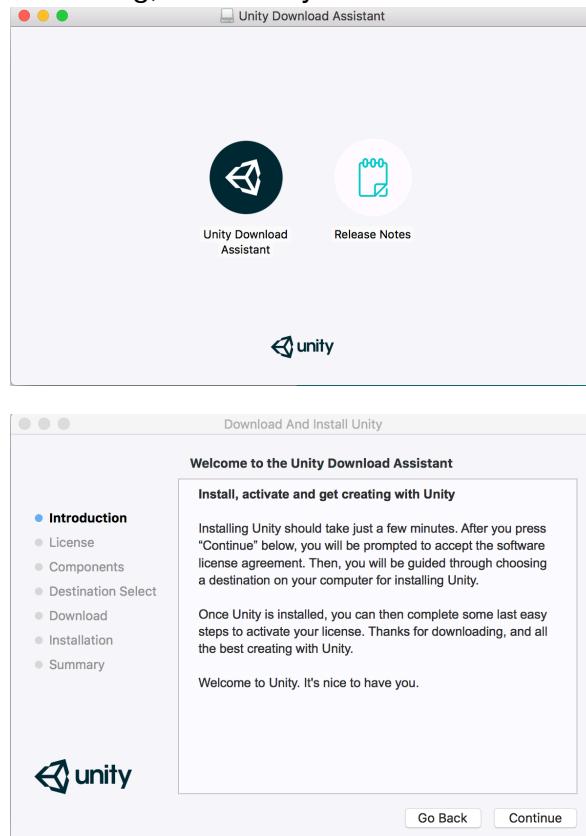


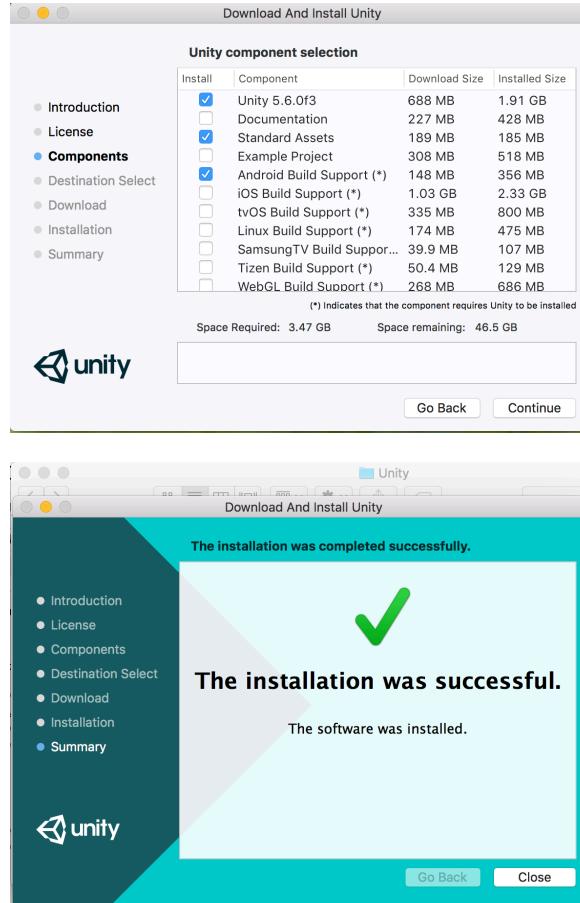
2 Unity

Unity is a game development platform that can create both 2D and 3D games. It's cross platform, so it can be used to develop apps for PC, Mac OS X, consoles, iOS, Android, Windows Phone 8, the web, and more.

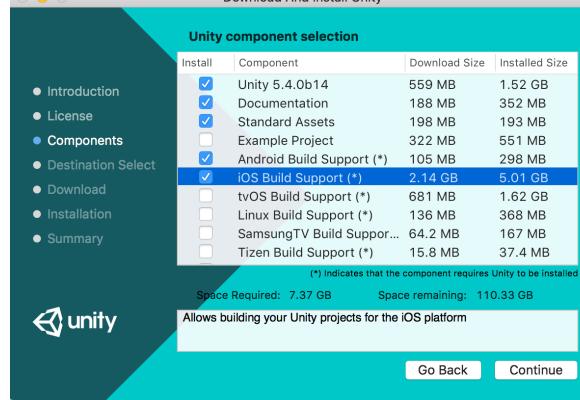
2.1 Unity installation Step by Step

1. Download [UnityHubSetup \(unity.org\)](https://unity3d.com/get-unity/download) from official site, for instance 2019.1f3 (2019/05/25);
2. As the assistant's default setting, install unity:





While installing Unity via its "Unity Download Assistant", also install the platform supports you want to make, for instance ios for Apple mobile devices:



2.2 Make a Virtual Reality Game with Unity For Google Cardboard

Prerequisites

- [Cardboard Unity Plugin \(<https://github.com/googlesamples/cardboard-unity>\)](https://github.com/googlesamples/cardboard-unity) v.0.6 (01/01/2016)
 - Modify the file, [\[cardboard-unity/Cardboard/Scripts/\]Cardboard.cs](#) as follows:

```
```C#
public RenderTexture StereoScreen {
get {
...
}
return stereoScreen;
}
set {
/// For 5.3.x
if (stereoScreen != null && !stereoScreen.IsCreated()) {
stereoScreen.Create();
}
///
if (value == stereoScreen) {
return;
}
...
}
````
```

- [\[Cardboard Unity Plugin\] v.0.7 \(04/15/2016\)](#)
- [\[Cardboard Unity Plugin\] v.0.8 \(05/20/2016\)](#)

2.3 GVR Unity SDK

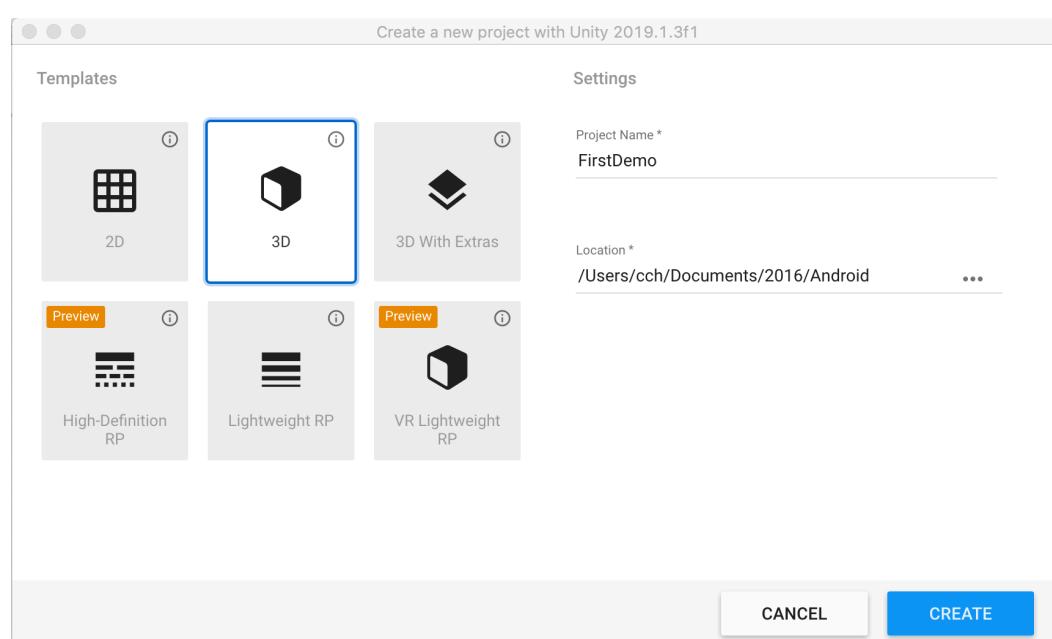
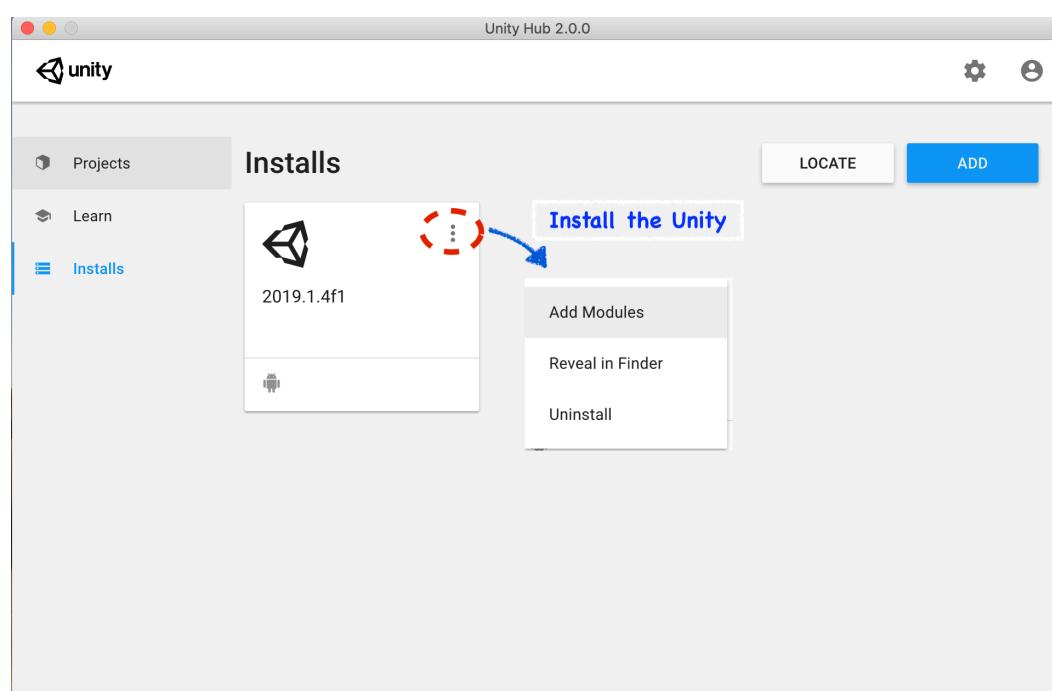
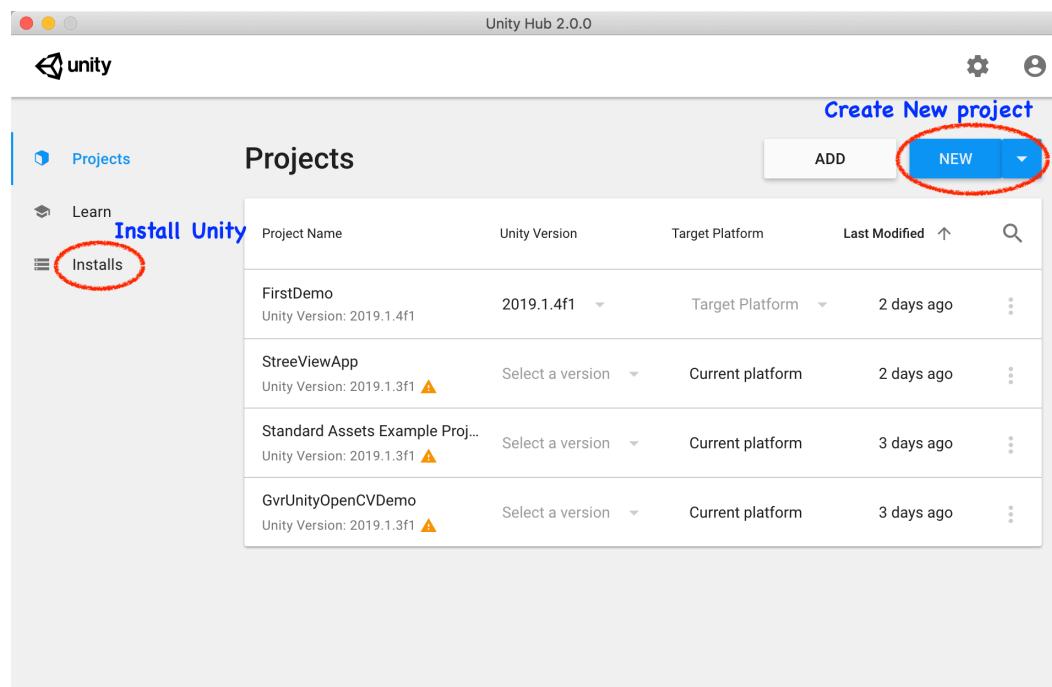
The new name for Google VR SDK for plugin,

- [Version 1.0.3 / 2016/11/21 \(<https://github.com/googlevr/gvr-unity-sdk>\)](#)
- [Version 1.4.0 / 2017/4/20 \(<https://github.com/googlevr/gvr-unity-sdk>\)](#)
- [Version v1.110.0 2017/11/20 \(\[https://github.com/googlevr/gvr-unity-sdk/releases/download/v1.110.0/GoogleVRForUnity_1.110.0.unitypackage\]\(https://github.com/googlevr/gvr-unity-sdk/releases/download/v1.110.0/GoogleVRForUnity_1.110.0.unitypackage\)\)](#) for Unity-6 or newer, size about 34 mega bytes
- [GVR SDK for Unity, 1.130.1 \(<https://github.com/googlevr/gvr-unity-sdk/releases>\)](#), 2018-05-08

2.4 Brief of Steps of Using Unity with Sample Project

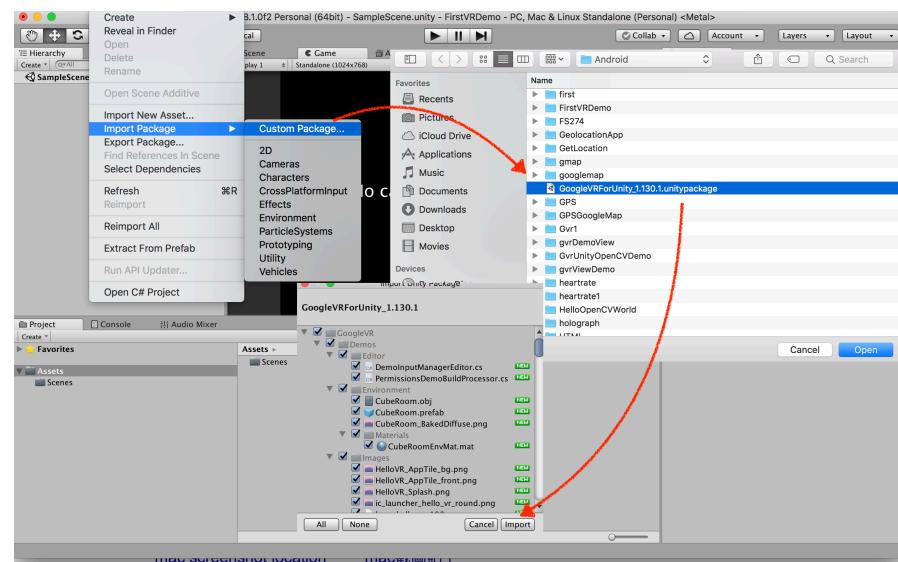
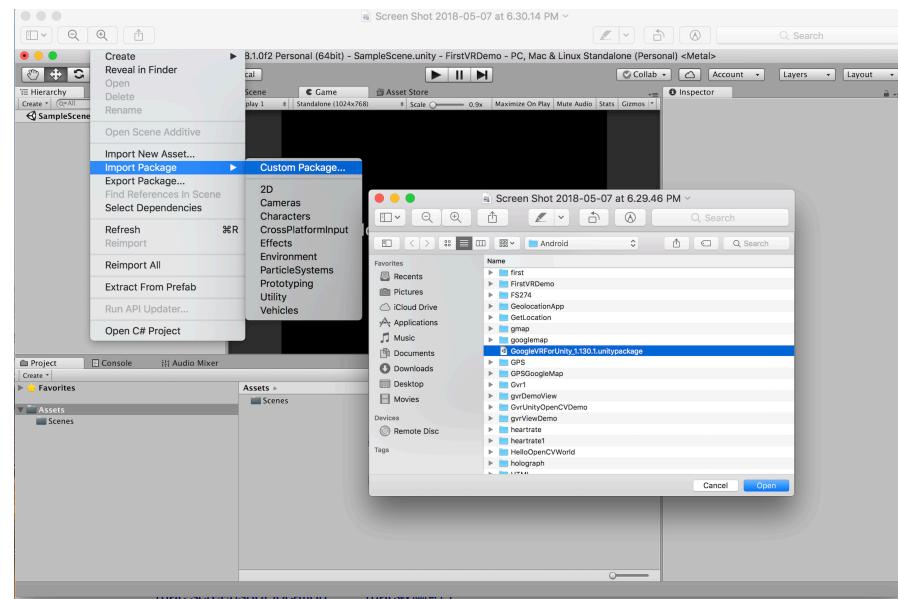
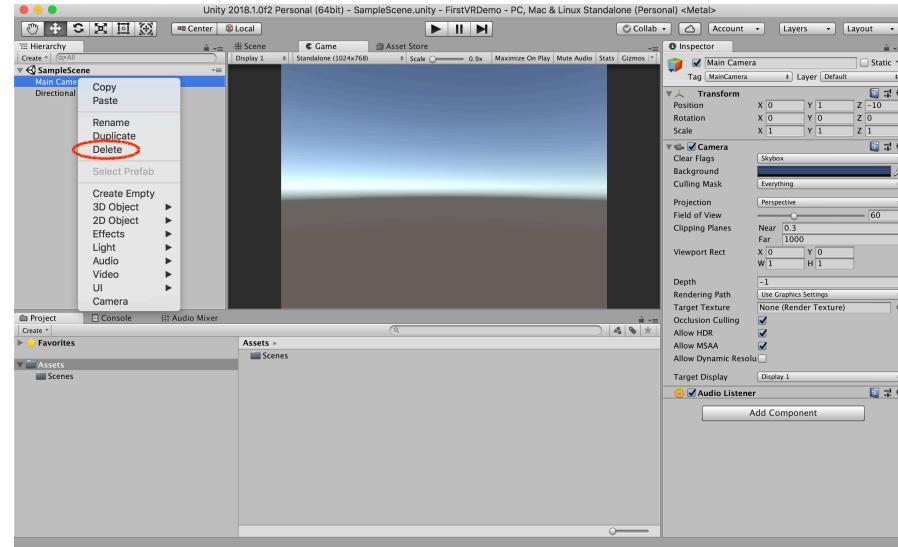
Reference the [official documentation \(<https://developers.google.com/vr/unity/get-started-android>\)](https://developers.google.com/vr/unity/get-started-android), let us to explore the first **Cardboard GVR app** by unity:

1. Open Unity, install last unity and create a new project.

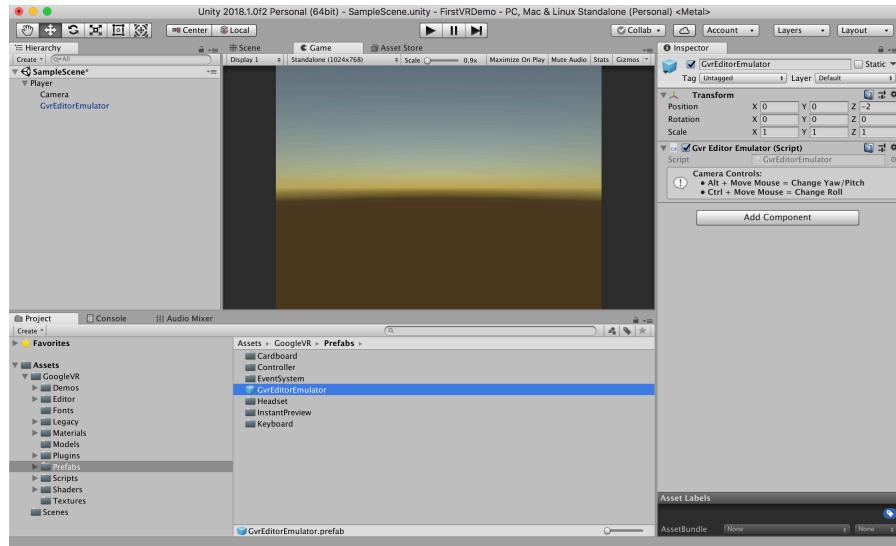
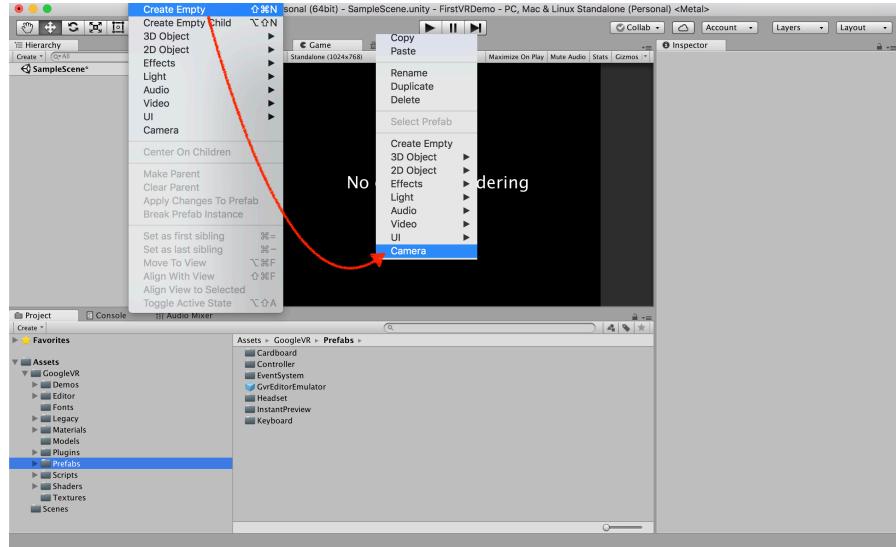


2. Import the SDK package: **Assets > Import Package > Custom Package**. Select the **GoogleVRSDKForUnity** (**CardboardSDKForUnity**, pre 0.8) unitypackage where you downloaded it and click **Open**. Make sure all the boxes are checked in the Importing Package dialog and click **Import**. **Note:** If you are using Unity 5, you may be warned that the APIs will be automatically upgraded. Accept it and continue if it happens.

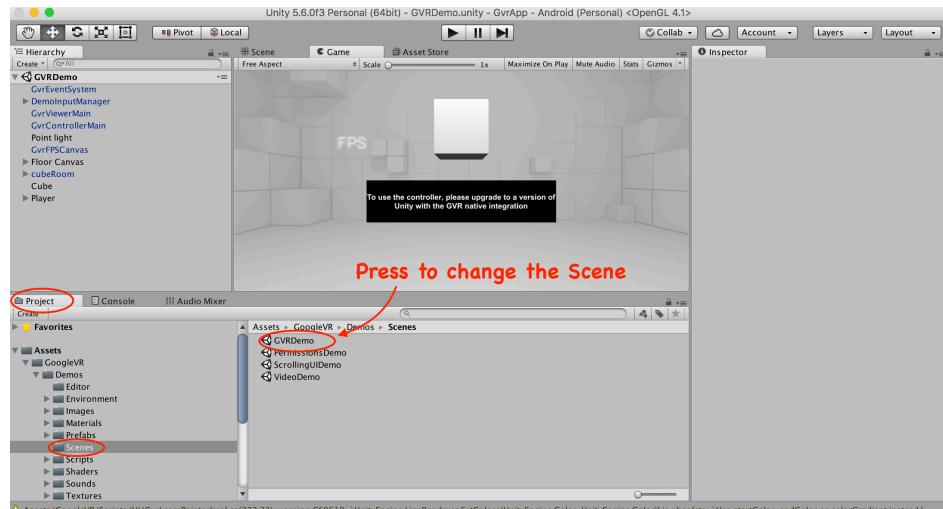
- delete any object in Hierarchy Windows, Main Camera and Directional Light ;
- Import the last GVR SDK for Unity, 1.30.1 (2018/05/08):



- create new GameObject, renamed as Player , add Camera and GVR SDK → Prefabs → GvrEditorEmulator under Player object.



3. In the editor's project panel, navigate to **Assets > GoogleVR > Demos > Scenes** folder and open **GvrDemo**. You should see a scene in which a floating cube is above a plane. Also an new option, GoogleVR, appears at the top menu bar.



4. Press **Play**, and you should see the game view showing a stereo rendering of a red cube. Press the **Control** key on your keyboard and move your mouse back and forth to tilt your view. You can also press the **option** button (**Alt** button for Windows) on your keyboard and move your mouse to pan around your view of the VR environment.

2.5 Build and Run APP

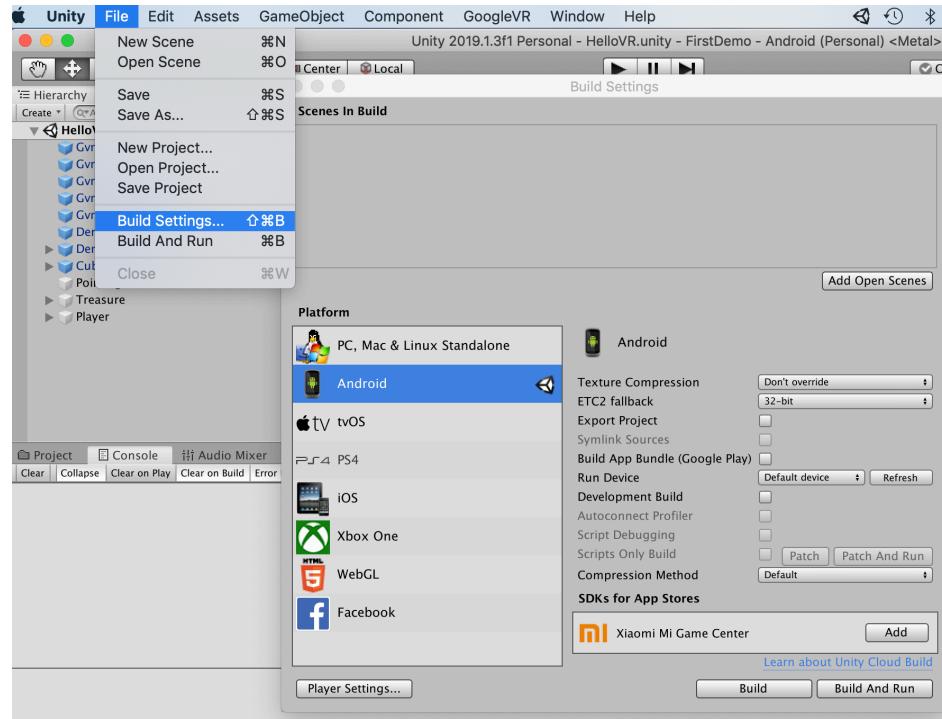
Building and deploying above demo to mobile app, for instance Android:

For Windows Users:

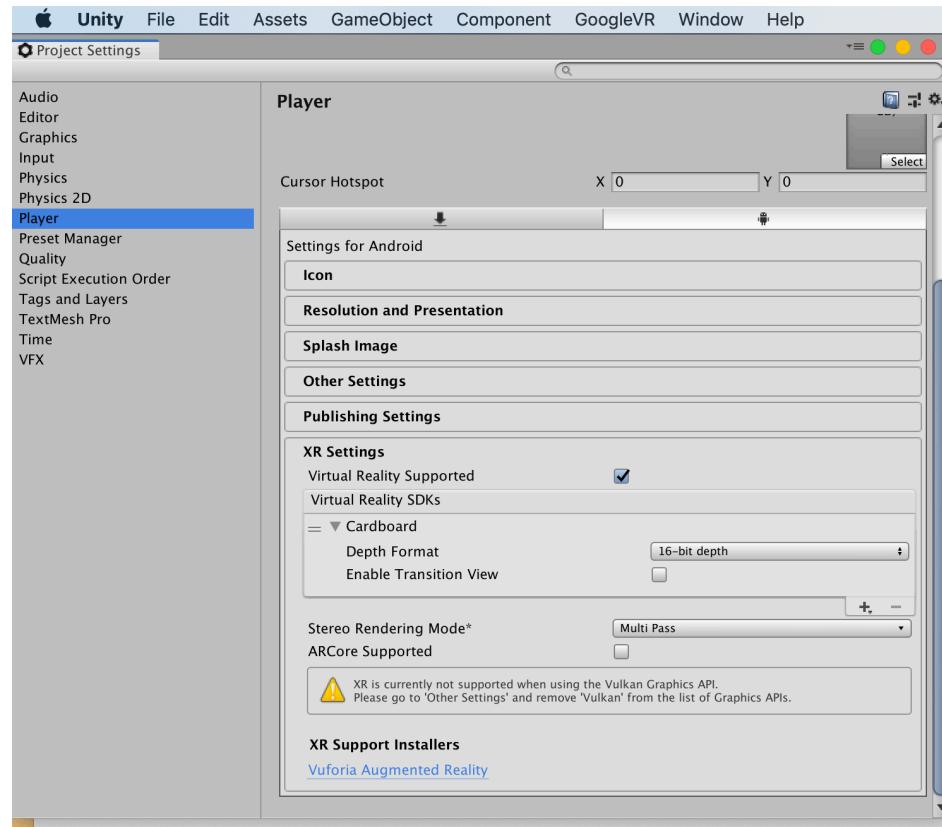
Before setup the "build setting", have to complete the settings (follows [Preferences/External Tools])

1. install "WinMerge"
2. set the path variables where Android SDK and JDK located;
3. ...

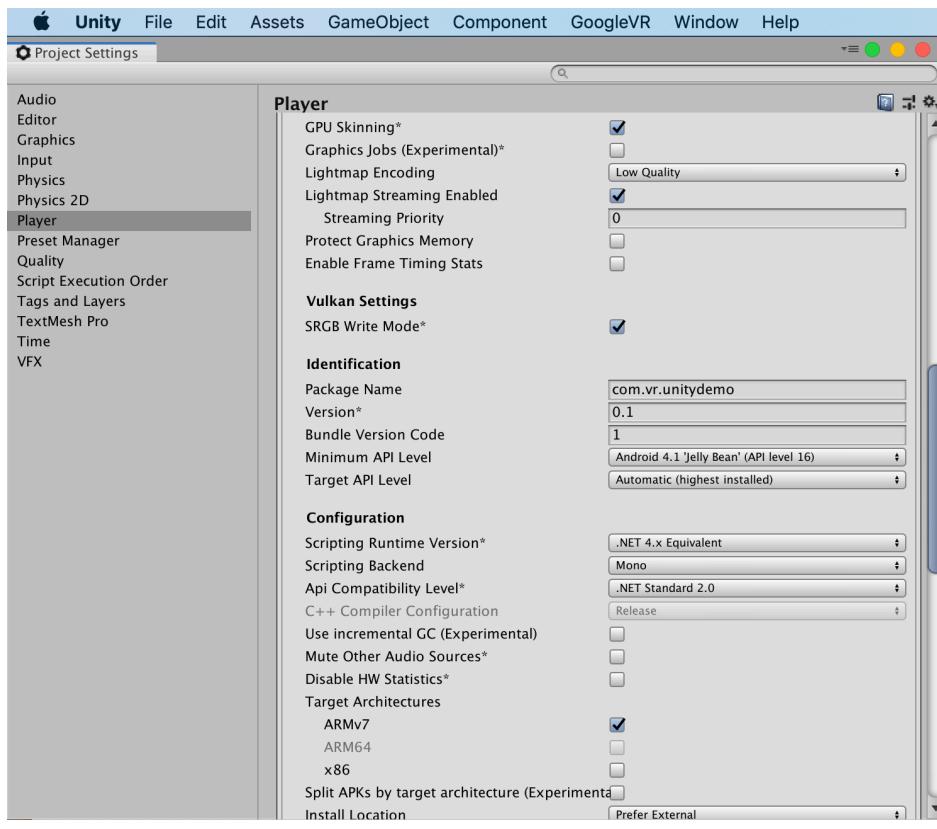
1. From **File > Build Settings**, select **Android** as follows:



2. Click **Player Settings**, in **XR Settings**, enable **Virtual Reality Supported**, choose **Cardboard** or Daydream, for instance, in **Virtual Reality SDKs**.



3. Change the [Package Name], `com.vr.unitydemo` for instance, in [Player Settings], [Player], [Other Settings]:



4. Click **Build and Run**. If prompted for location of the Android SDK, select unzipped SDK at which it had been installed. If not worked for JDK not found, check Unity preference. If more higher version of SDK is not found, update should be automatically prompted to progress from [from higher version] enabled.

2.6 Demo for GvrSDK(200)

GvrSDK provides plenty of interactive functions; here, we introduce how to its `Event` functions.

1. create new project, [GVR200App](#),
2. import gvr SDK-200, by [Assets] ➔ [Import Package] ➔ [Custom Package]
3. switch platform to android, [build setting] ➔ [Android], change app's name, build platform, enable cardboard support etc as above;
4. Create an empty GameObject, name it as **Player**, [GameObject] ➔ [Create Empty],
5. drag [Main Camera] into {Player},
6. GVR part: from left bottom [Assets] widows, add some stuffs of GVR
 - add a child object of [Main Camera] from [GoogleVR ➔ Prefabs ➔ Cardboard ➔ GvrReticlePointer] which is the pointer that enables the interaction with the UI system and 3D objects; it's in Position: [0, 0, 0], no rotations, and a scale of 1.
 - add [GoogleVR ➔ Prefabs ➔ GvrEditorEmulator] in [GvrStuff] which allow for the emulation of the Cardboard in the Editor.
 - add [GoogleVR ➔ Prefabs ➔ Controllers ➔ GvrControllerMain] in [GvrStuff] which allows for the cardboard to actually work.
 - add [GoogleVR ➔ Prefabs ➔ EventSystem ➔ GvrEventSystem]
7. Interacting with User Interface [UI],
 - Create [GameObject ➔ UI ➔ Button]
 - Go to the canvas and remove the [Graphics Raycaster] component and add the Gvr Pointer Graphics Raycaster, from right button [add Component];
 - Set the Canvas's Render Mode to World Space
 - Set the Canvas scale to (0.01, 0.01, 0.01)
 - Set the rotation to (0, 30, 0)
 - Set Position to (2, 0, 2)

- Set the width to 200 and height to 100
- Set the button position to (0, 0, 0)

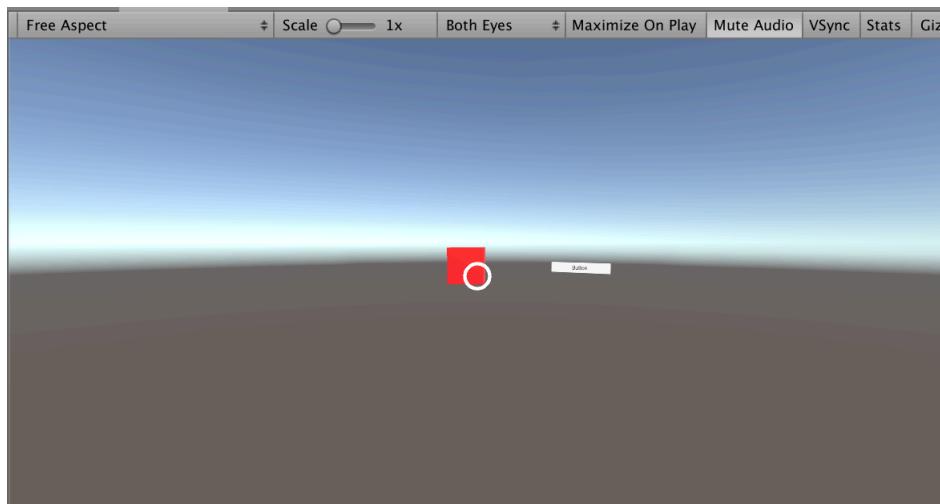
8. Create a cube for test, by ['GameObject'→'3D Object'→'Cube'],

- set position (-1, 0, 3) with no rotation and scale,
- Create a material in [Assets] and name it as Red , by [create>material],
- drag [Red] to [No material]
- In the [OnClick()] of the button, drag the [Cube] into [None Object], and choose function, [MeshRenderer→Material material]
- Now test the game.

9. Interacting with 3D Objects: we can also interact with object without button by changing its material using EventTriggers:

- Go to the [Main Camera] and add a component called GvrPointerPhysicsRaycaster , searched by phys ; To interact with object, we needs to have a [Collider] and an [EventTrigger] components.
- [Cube]: add component, [Event Trigger], Press Add New Event and select PointerEnter, PointerExit, PointClick.
- Just like we did with the button On Click, we will duplicate the material 3 times and add 3 different materials to test it.

Test it all now.



Note To make game more interesting, we can add some effect, audio for instance:

- Sound Effect:
 - Create new folder within [Assets], name Audio , and drag the sound file in it:
 - Add Sound for **GvrEditorEmulator**, [Add Component] -> [Audio] -> [Audio Source]:
 - Drag the sound asset into [Audio Source] -> [AudioClip]:

Et Voila, enjoy!

2.7 Second Demo

Implement yacht navigating on the water.

1. delete [Main Camera]; later, we add the stero Camera, come from Gvr SDK. [Edit, delete]
2. select [Directional Light], and change the setting:

```
Transform
Position: X=0, Y=20, Z=0
Rotation: X=80, Y=18, Z=18
```

```
Light
Mode: Mixed
Intensity: 3.5
Indirect Multiplier: 0
```

3. import Gvr SDK,

- drag [GVR SDK → Prefabs → GvrEditorEmulator] to **Hierarchy Windows**, change Position: X=0, Y=2, Z=-20 , Add Component. Select [Nav Mesh Agent]

```
Agent Size
Base offset: 1
Steering
Speed: 1
Angular speed: 0
Acceleration: 1
```

- Add Component, select [New Script]. Leave the name as NewBehaviourScript and language C#. Press [Create and Add]

```
--config=<Unicode> (JupyterApp.config_file)
Default: ''
Full path of a config file.
--to=<Unicode> (NbConvertApp.export_format)
Default: 'html'
The export format to be used, either one of the built-in formats
['asciidoc', 'custom', 'html', 'html_ch', 'html_embed',
 'html_toc',
 'html_with_lenvs', 'html_with_toclenvs', 'latex', 'latex_with_lenvs',
 'markdown', 'notebook', 'pdf', 'python', 'rst', 'script',
 'selectLanguage',
 'slides'] or a dotted object name that represents the import path for an
`Exporter` class
--template=<Unicode> (TemplateExporter.template_file)
Default: ''
Name of the template file to use
--writer=<DottedObjectName> (NbConvertApp.writer_class)
```

2.8 Google Street View App

1. Add Street View to Skybox,
2. input Latitude/longitude
3. get the skybox pictures
4. Import FPSCo ntrol, Assets -> import package -> Characters
5. drag the pictures into Scence