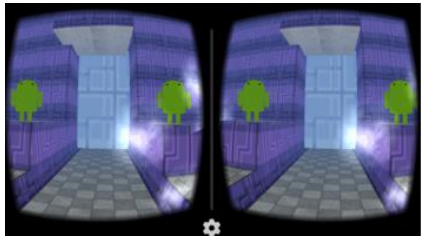
天津GDG社区，2016年3月12日

Cardboard Unity Codelab

师文轩，13920561100



Overview

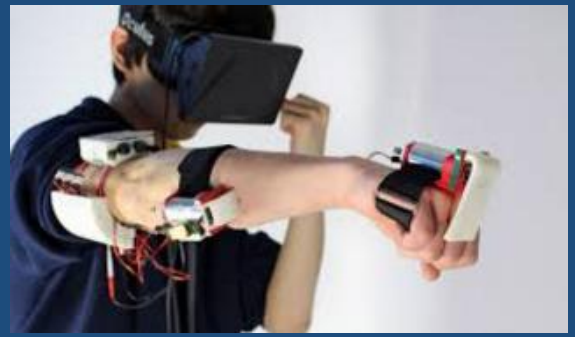
This codelab is an introduction to how to take a **first person** game made in **Unity** and enhance it to have a **Virtual Reality** mode using the Google **Cardboard** plugin powered by Unity.

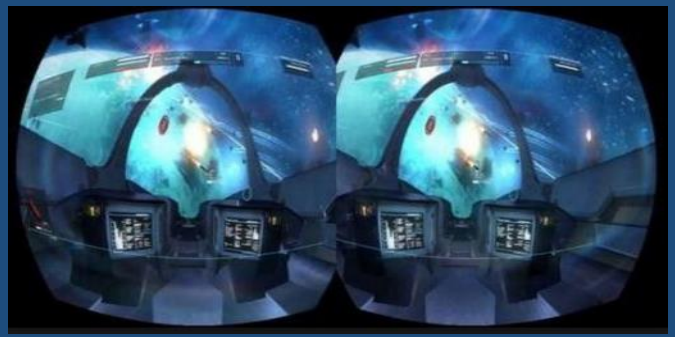
What you'll learn

- Development tools and configuration to build Unity projects targeted for Android.
- How to design a game that can be used in Cardboard and out of Cardboard.

What you'll learn

- Knowledge of best-practices for mobile VR.
- Links to key information for developing and publishing great Cardboard applications.





What you'll need

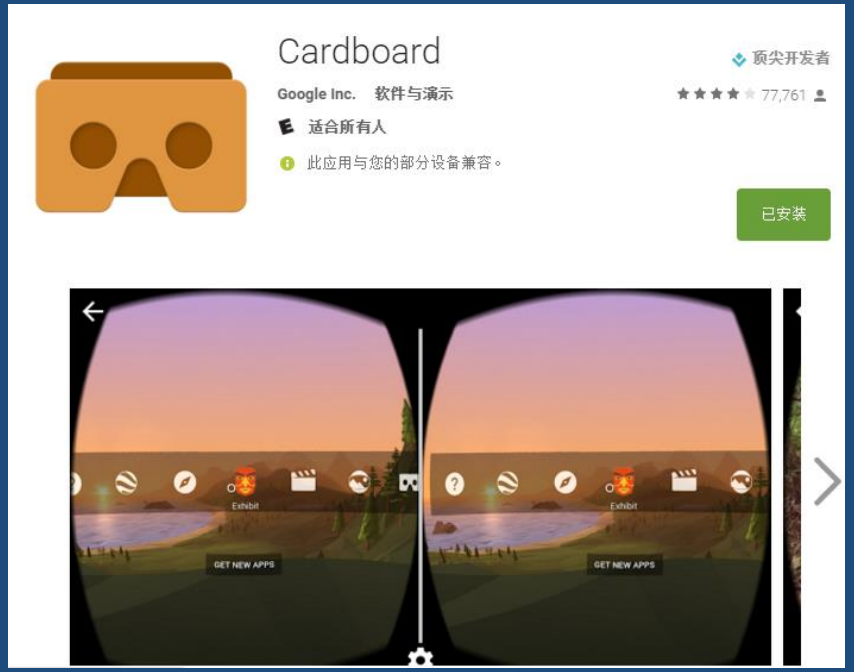
- Cardboard VR Viewer
- Device compatible with Cardboard
  - such as Nexus 5, Samsung Galaxy S5...
- Unity 4.6.1 or greater
- Android SDK

What you'll need

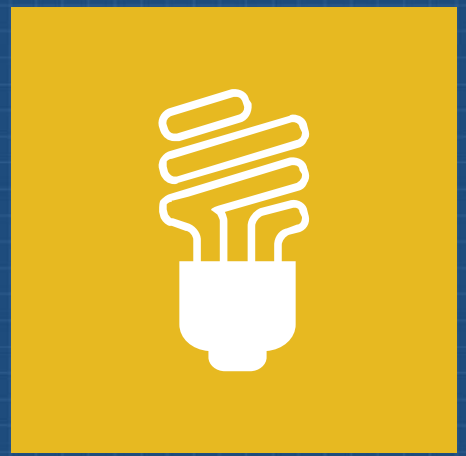
- The cardboard plugin (286M)
  - https://github.com/googlesamples/cardboard-unity/archive/master.zip
- Game assets
  - https://github.com/googlesamples/io2015-codelabs/tree/master/cardboard-unity

What you'll need

- The Cardboard Application (Optional).
  - https://play.google.com/store/apps/details?id=com.google.samples.apps.cardboarddemo



Game Summary



"Lollygagger“: The main game play is to travel along a path, atop a robot, shooting Lollipop "cannonballs" at the Androids that are lollygagging around.

Lollygagger

- Two different Unity scenes, one for the main menu, and one for game play.
- Ability to play in “2D” mode as well as be able to start in Cardboard mode.

Lollygagger

- Flinging lollipops to the targets, triggered by touch in 2D mode, or by the trigger on the side of Cardboard in VR mode.

Get the sample code

1. First you need the Cardboard SDK for unity.
2. Next you need the unity asset package containing the sample game.

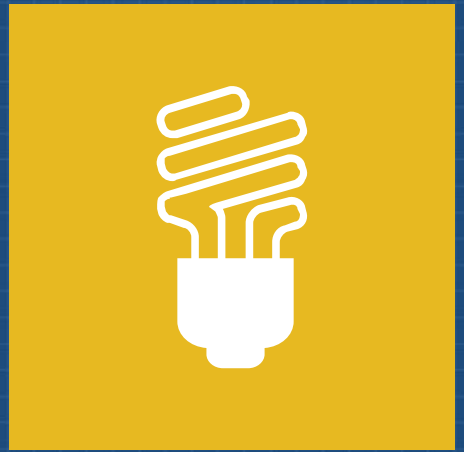
Get the sample code

- Cardboard SDK for unity
  - Download:
    - https://github.com/googlesamples/cardboard-unity/archive/master.zip
  - Or clone：
    - $ git clone https://github.com/googlesamples/cardboard-unity.git

Get the sample code

- The unity asset package containing the sample game
  - lollygagger_step0.unitypackage
    - https://github.com/googlesamples/io2015-codelabs/blob/master/cardboard-unity/assets/lollygagger_step0.unitypackage?raw=true

Run the starter game

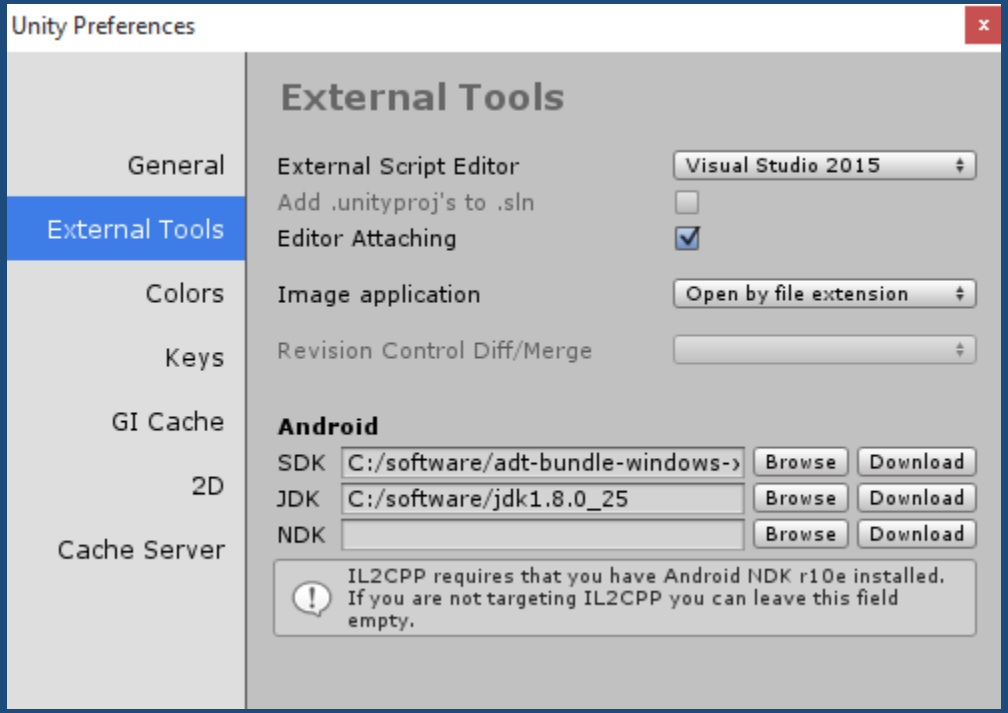


1. First, let's build and run the non-Cardboard version of the game.
2. From there, we'll add Cardboard support.

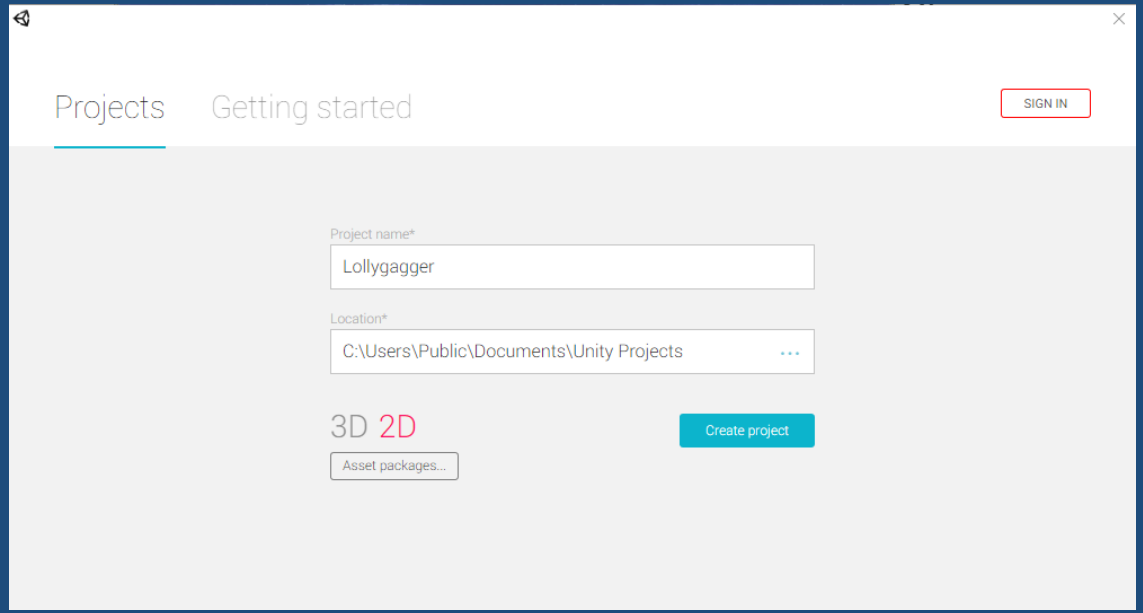
Configure Unity

- Start Unity.
- In Unity, click **Edit** > **Preferences** > **External Tools** > **Android SDK Location**, then select the folder where you downloaded and unzipped the Android SDK.

Android SDK Location



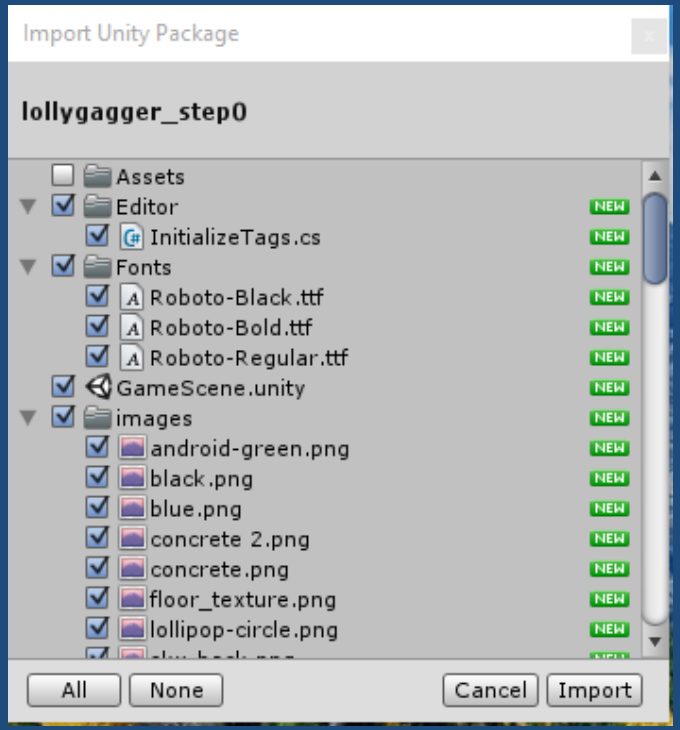
- Click **File** > **New Project** and create a new project named '**Lollygagger**'.

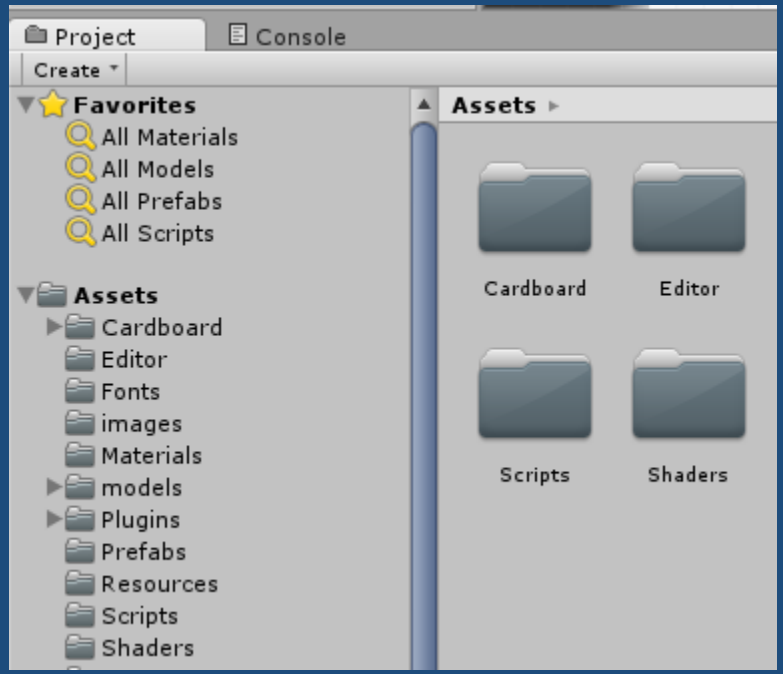


Import the game assets

- Import the sample game scenes and assets into your project:
  - Click **Assets** > **Import Package** > **Custom Package**.
  - Select the **lollygagger_step0.unitypackage** file you downloaded

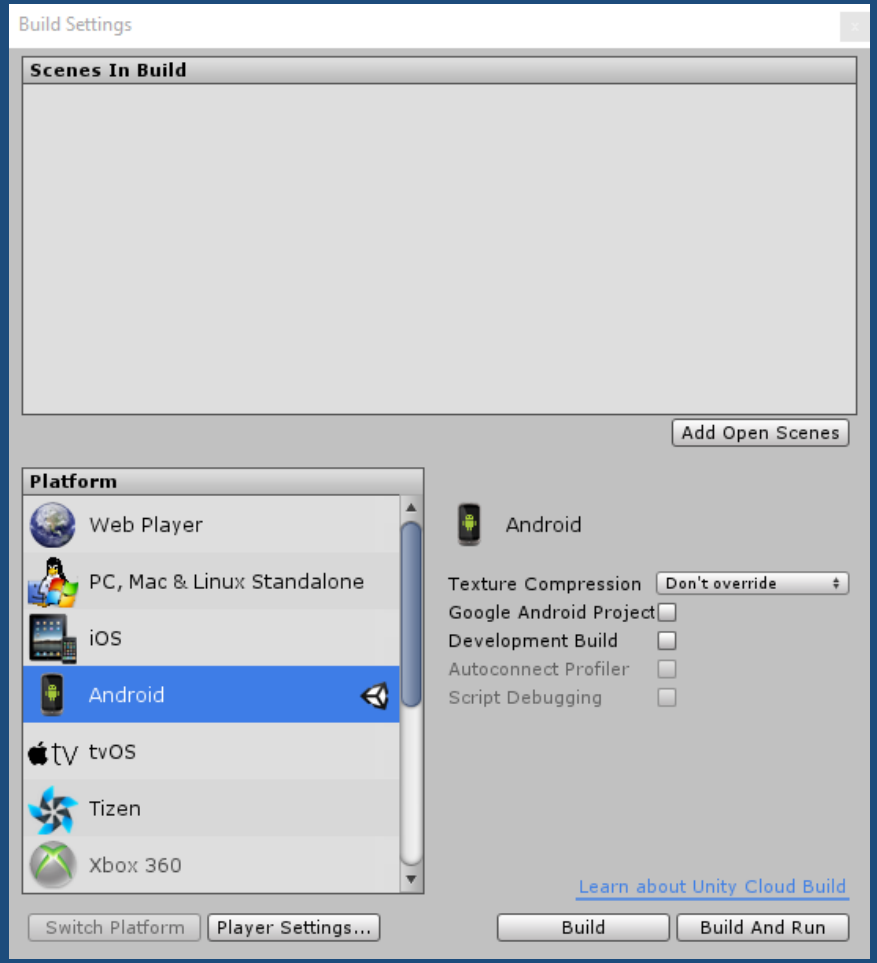
Click Import.



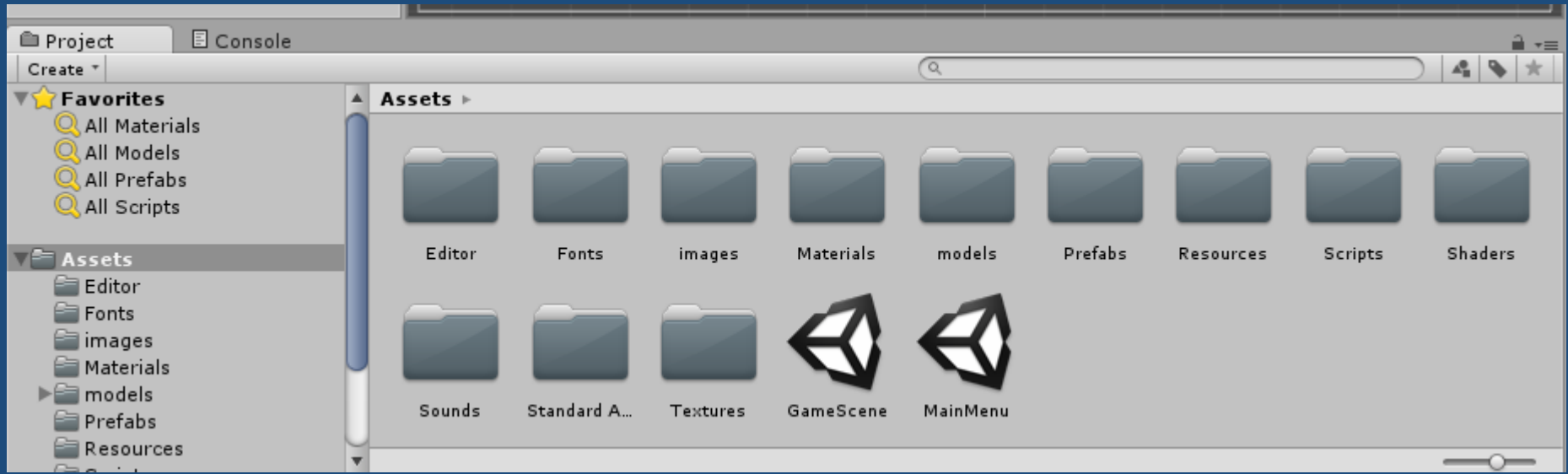


Configure the Android player

- Configure Unity to build an Android application
- From the menu: click **File** > **Build Settings**...
- Select **Android** in the platform list
- Click **Switch Platform**

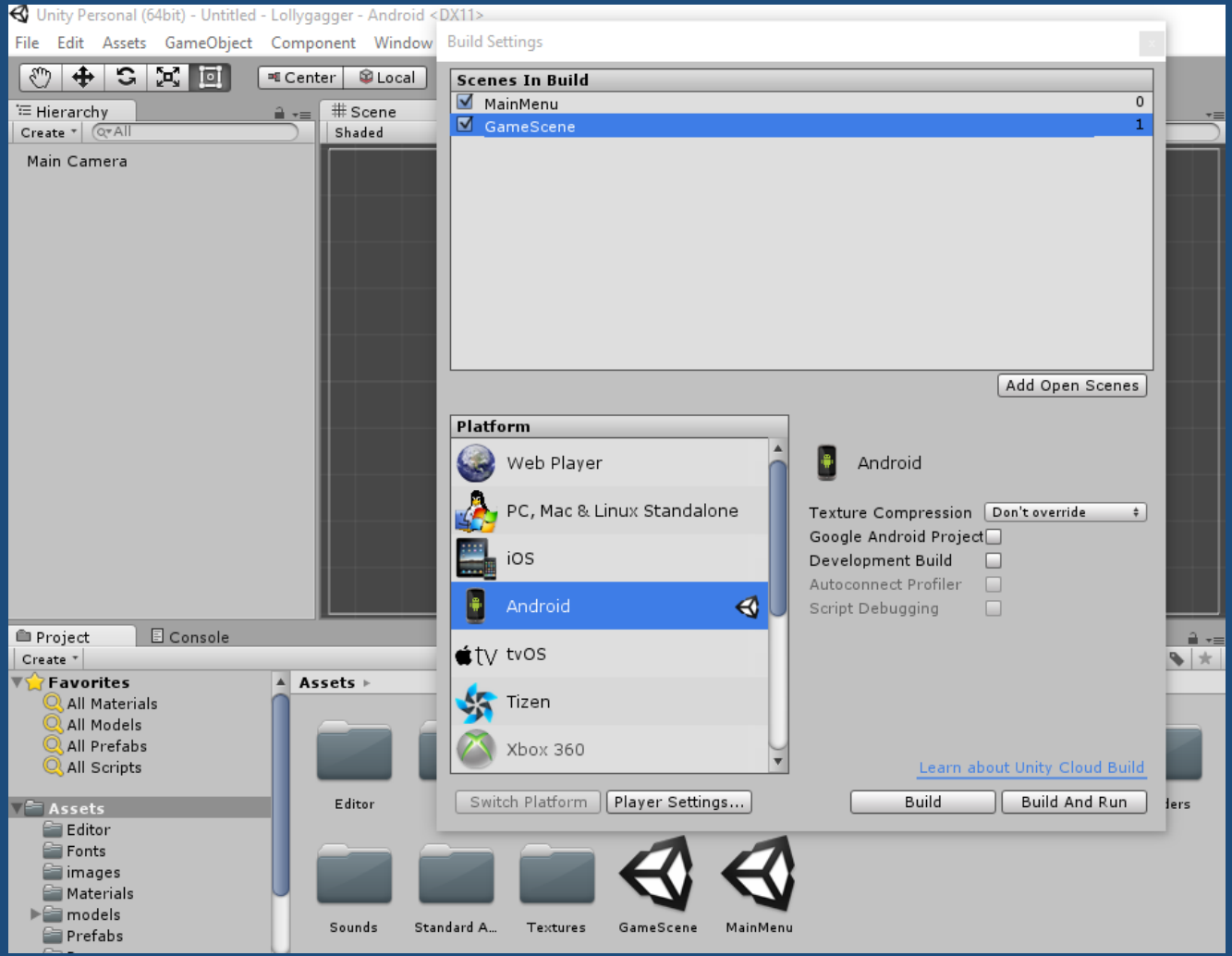

Build Settings
Scenes In Build
Add Open Scenes
Platform
Web Player
PC, Mac & Linux Standalone
iOS
Android
tvOS
Tizen
Xbox 360
Android
Texture Compression
Don't override
Google Android Project
Development Build
Autoconnect Profiler
Script Debugging
Learn about Unity Cloud Build
Switch Platform
Player Settings...
Build
Build And Run

In the project explorer, select Assets



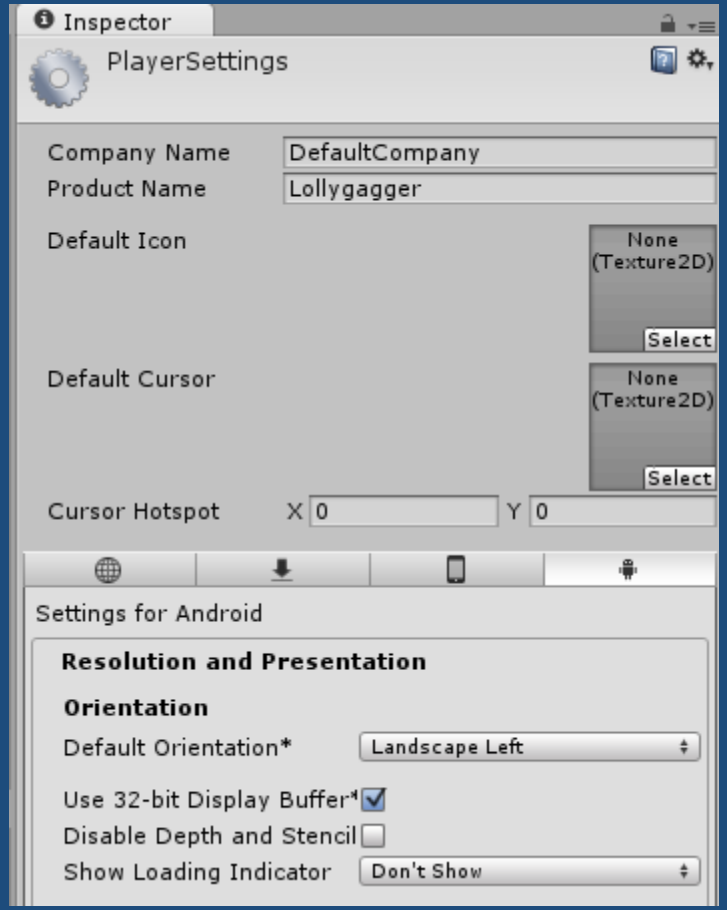
Add the game scenes to the build.

- Drag and Drop Assets/MainMenu scene to the "Scenes In Build" list within the Build Settings dialog.
- Then drag and drop Assets/GameScene scene into the list under MainMenu.

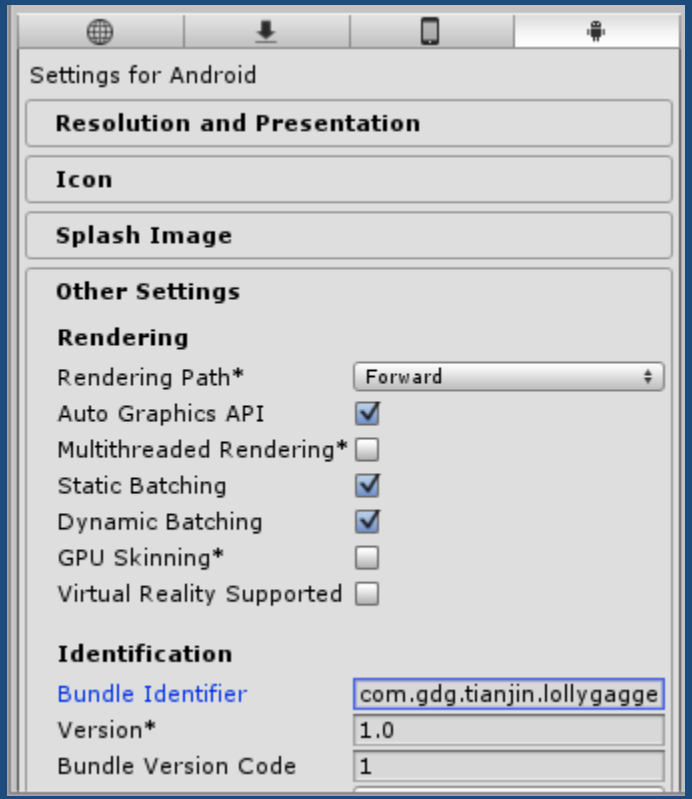

Unity Personal (64bit) - Untitled - Lollygagger - Android <DX11>
File Edit Assets GameObject Component Window
Center Local
Hierarchy
Create All
Main Camera
Scene
Shaded
Project Console
Create
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
Editor
Fonts
images
Materials
models
Prefabs
Assets
Editor
Sounds
Standard A...
Textures
GameScene
MainMenu
Build Settings
Scenes In Build
MainMenu 0
GameScene 1
Add Open Scenes
Platform
Web Player
PC, Mac & Linux Standalone
iOS
Android
tvOS
Tizen
Xbox 360
Android
Texture Compression Don't override
Google Android Project
Development Build
Autoconnect Profiler
Script Debugging
Learn about Unity Cloud Build
Switch Platform
Player Settings...
Build
Build And Run

Configure the Unity player settings

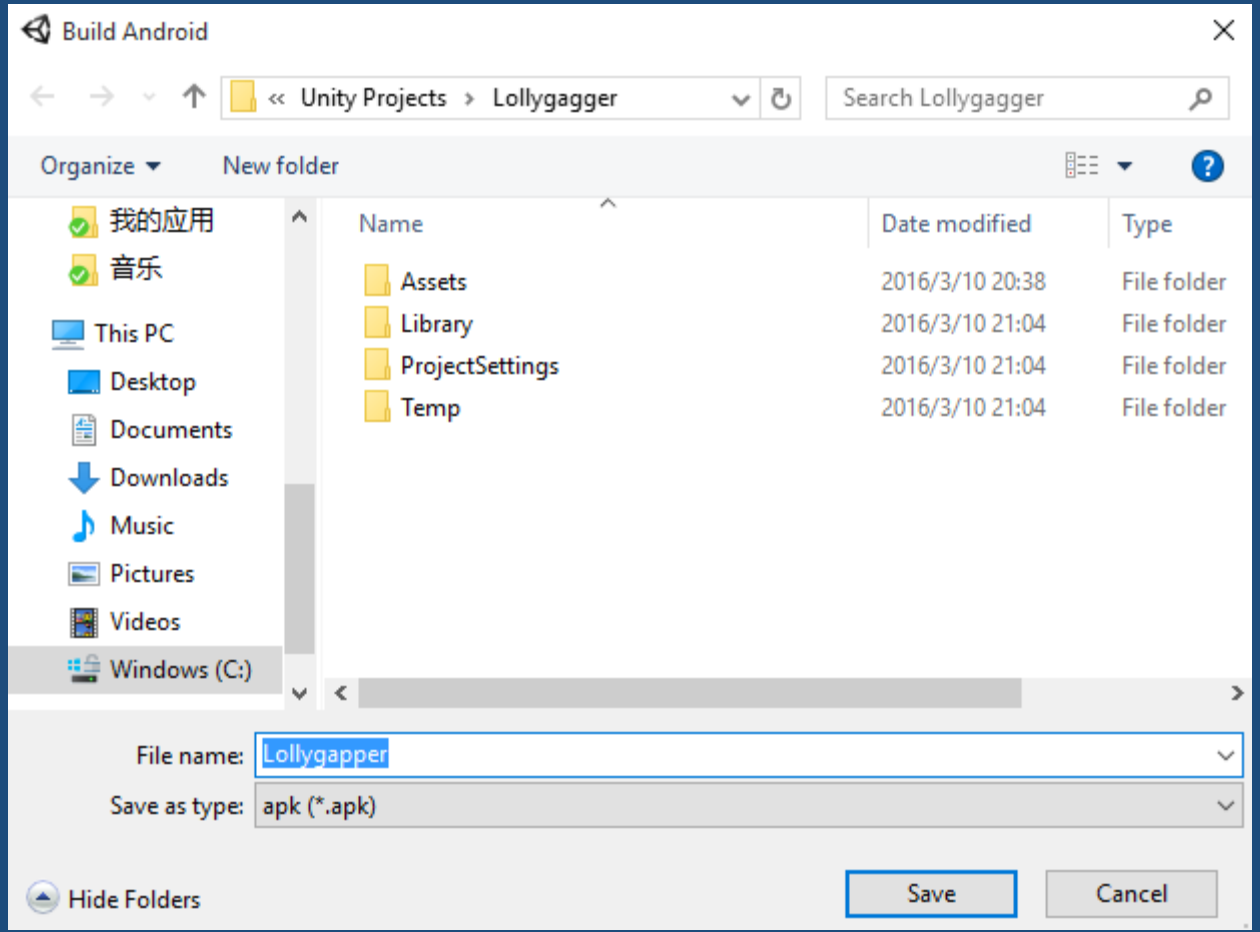
- Click **Player Settings**... in the Build Settings dialog
- Then expand "**Resolution and Presentation**" section by clicking on it.
- Set the application **default orientation** to **Landscape Left**.


Inspector
PlayerSettings
Company Name
DefaultCompany
Product Name
Lollygagger
Default Icon
None (Texture2D)
Select
Default Cursor
None (Texture2D)
Select
Cursor Hotspot
X 0
Y 0
Settings for Android
Resolution and Presentation
Orientation
Default Orientation*
Landscape Left
Use 32-bit Display Buffer*
Disable Depth and Stencil
Show Loading Indicator
Don't Show

- Expand the “Other Settings” section by clicking on it.
- Enter a unique Bundle ID
  - Typically this is of the form: com.<yourcompany>.<appname>

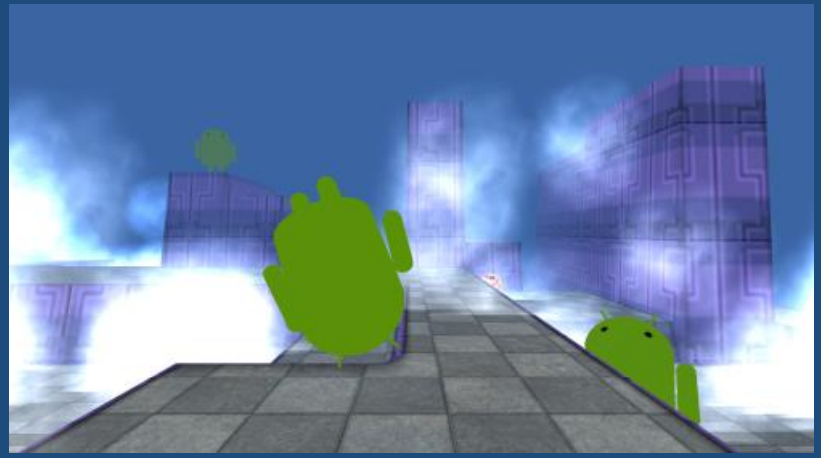

Settings for Android
Resolution and Presentation
Icon
Splash Image
Other Settings
Rendering
Rendering Path*
Forward
Auto Graphics API
Multithreaded Rendering*
Static Batching
Dynamic Batching
GPU Skinning*
Virtual Reality Supported
Identification
Bundle Identifier
com.gdg.tianjin.lollygagge
Version*
1.0
Bundle Version Code
1

- **Save** your project! Better safe than sorry :)
- Connect a device **build and run**!
- When prompted for a name for the APK file, enter **Lollygagger.apk**.


Build Android
« Unity Projects › Lollygagger
Search Lollygagger
Organize
New folder
我的应用
音乐
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
Name
Date modified
Type
Assets
2016/3/10 20:38
File folder
Library
2016/3/10 21:04
File folder
ProjectSettings
2016/3/10 21:04
File folder
Temp
2016/3/10 21:04
File folder
File name:
Lollygapper
Save as type:
apk (*.apk)
Hide Folders
Save
Cancel

Congratulations!

- If you can play the starter project game on your Android device, you are now ready to add Cardboard!

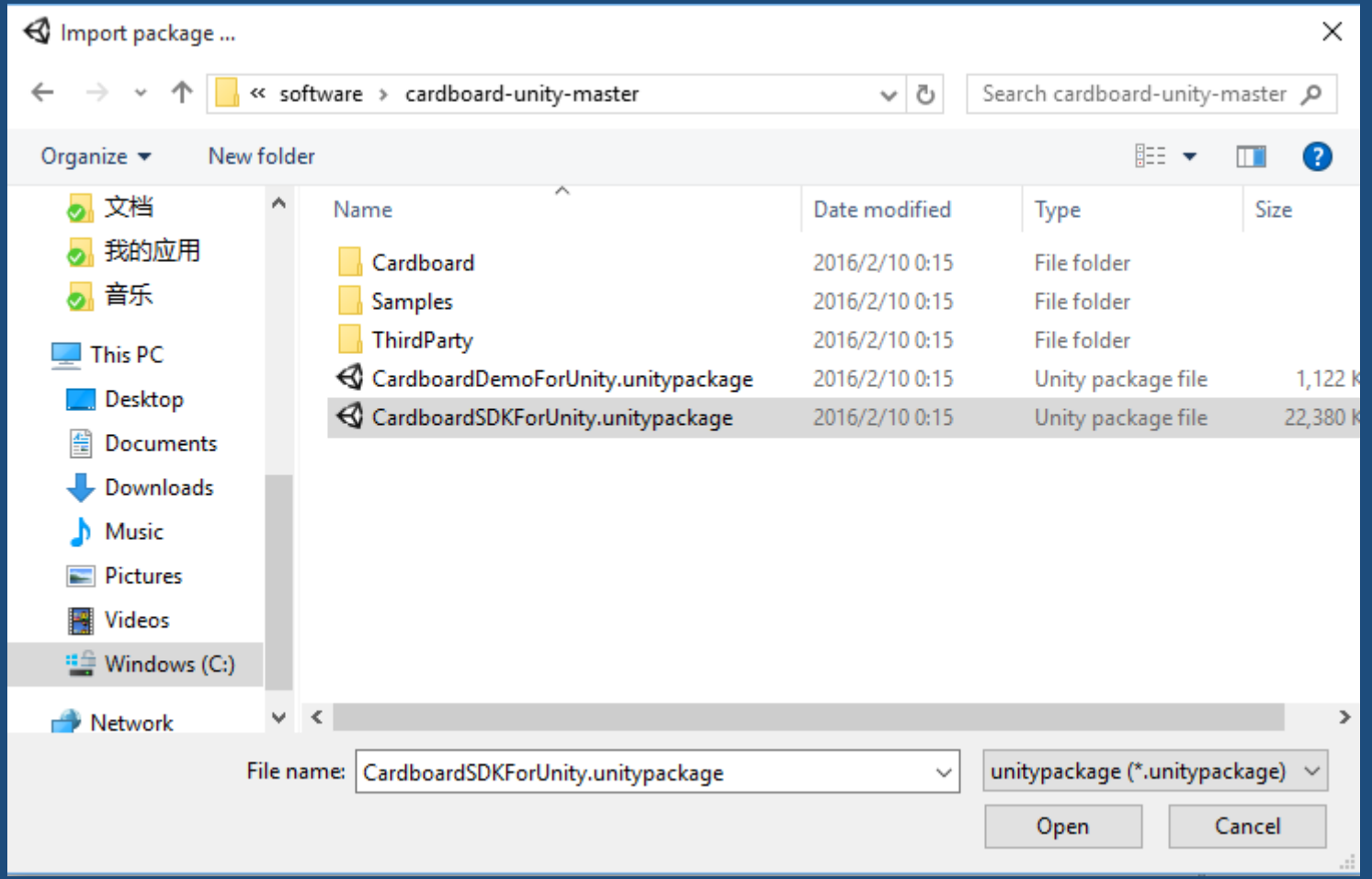


Add cardboard to the main menu scene

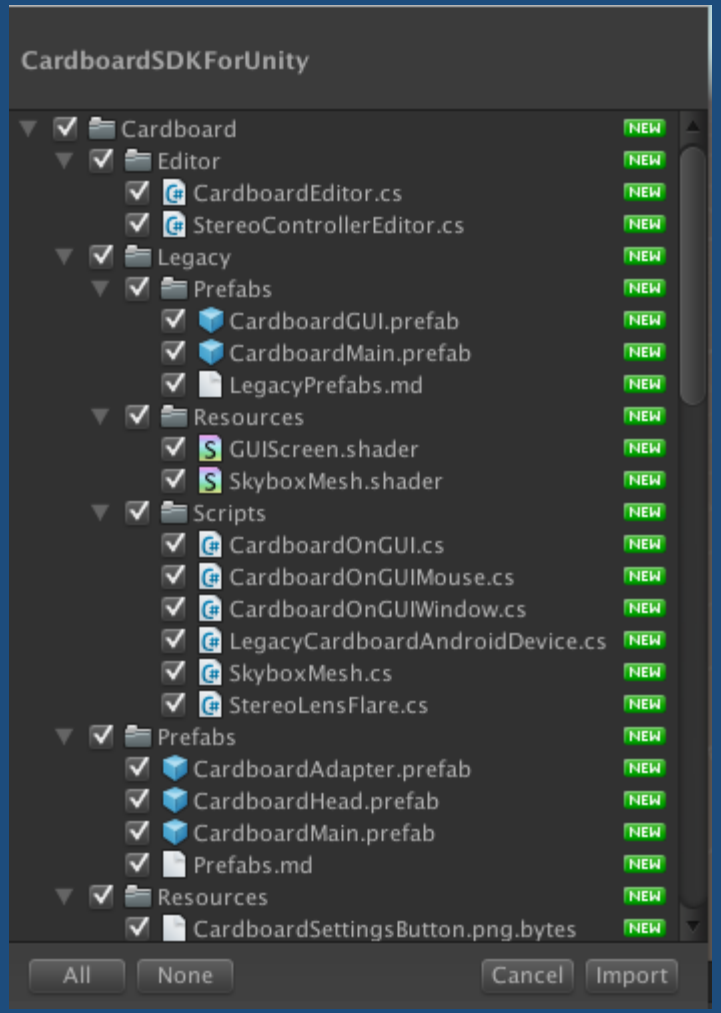
In this lesson, you'll import the Cardboard SDK into the Unity project.

Add the Cardboard SDK to the project

- From the menu in Unity: click Assets > Import Package > Custom package
- Select CardboardSDKForUnity.unitypackage
- Click “Open"


Import package ...
« software › cardboard-unity-master
Search cardboard-unity-master
Organize
New folder
文档
我的应用
音乐
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
Network
Name
Date modified
Type
Size
Cardboard
2016/2/10 0:15
File folder
Samples
2016/2/10 0:15
File folder
ThirdParty
2016/2/10 0:15
File folder
CardboardDemoForUnity.unitypackage
2016/2/10 0:15
Unity package file
1,122 K
CardboardSDKForUnity.unitypackage
2016/2/10 0:15
Unity package file
22,380 K
File name:
CardboardSDKForUnity.unitypackage
unitypackage (*.unitypackage)
Open
Cancel

- Once the package is opened, Click "Import"

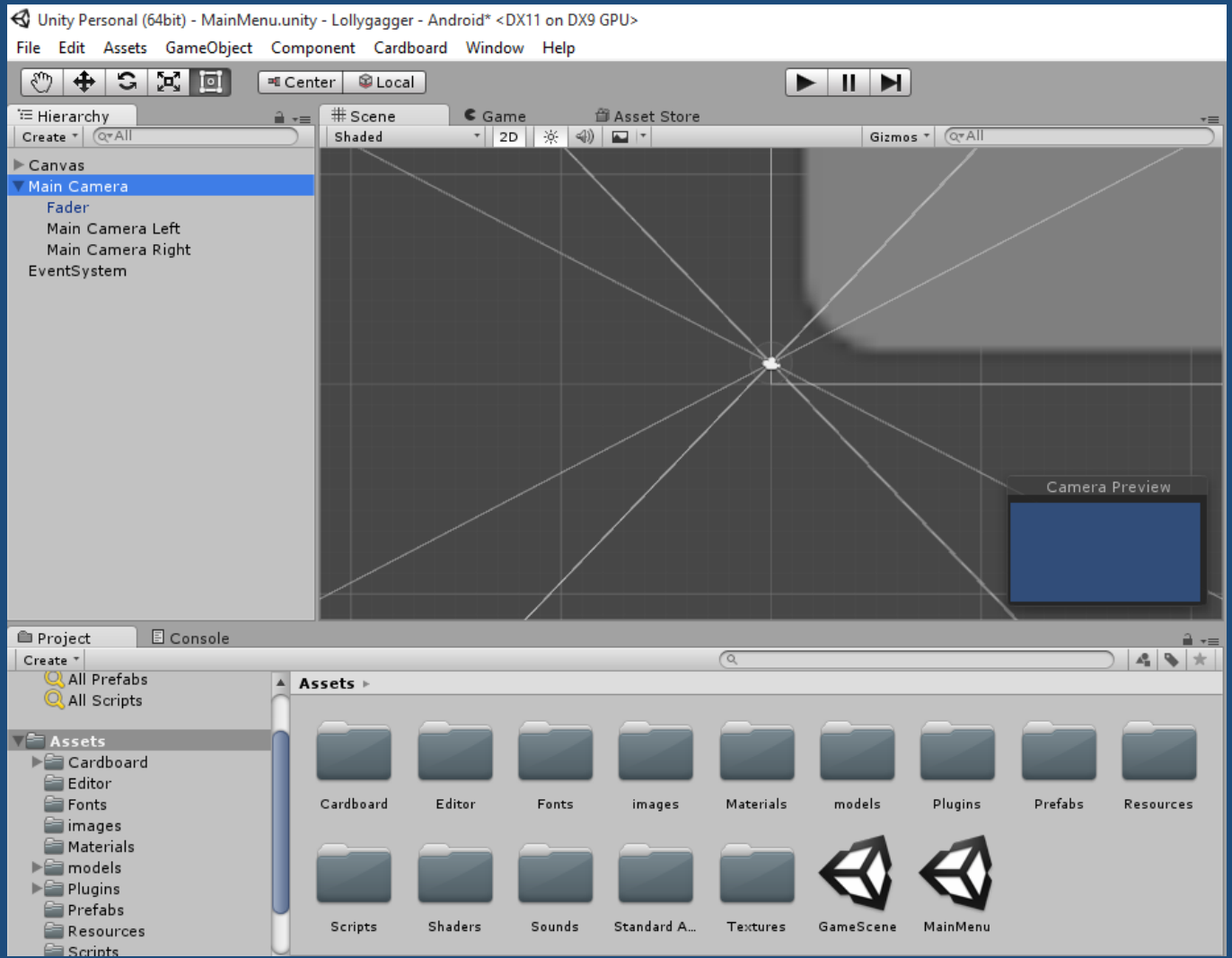


Add cardboard mode to the main menu scene

- This step adds the stereo cameras and related scripts which are used by the Cardboard SDK to render the stereoscopic view needed to use Cardboard.

Open the main menu scene

- In the project explorer, select the Assets folder, and then double click the MainMenu scene.
- Select the Main Camera in the Scene hierarchy.
- From the menu, click Component > Cardboard > Update Stereo Cameras.

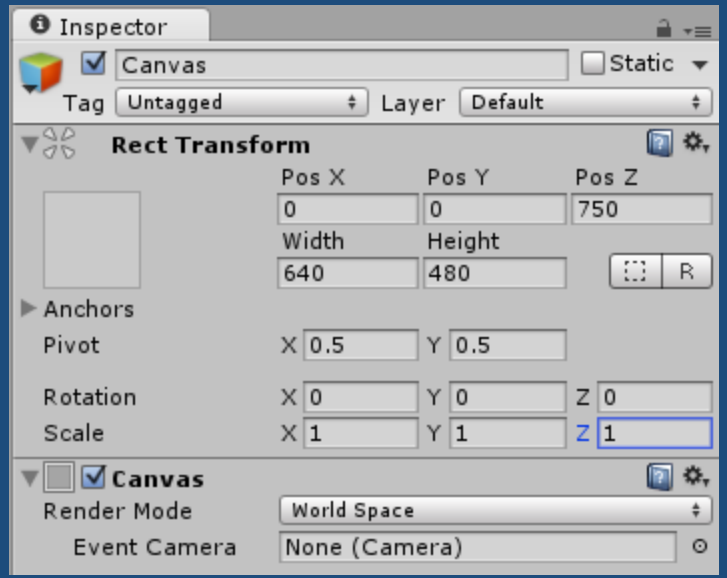

Unity Personal (64bit) - MainMenu.unity - Lollygagger - Android* <DX11 on DX9 GPU>
File Edit Assets GameObject Component Cardboard Window Help
Center Local
Hierarchy
Create All
Canvas
Main Camera
Fader
Main Camera Left
Main Camera Right
EventSystem
Scene Game Asset Store
Shaded 2D Gizmos All
Camera Preview
Project Console
Create
All Prefabs
All Scripts
Assets
Cardboard
Editor
Fonts
images
Materials
models
Plugins
Prefabs
Resources
Scripts
Assets
Cardboard Editor Fonts images Materials models Plugins Prefabs Resources
Scripts Shaders Sounds Standard A... Textures GameScene MainMenu

- **Note**: If the menu item is not there, you'll need to re-import the assets by clicking **Assets** > **Reimport All**.

Change the UI to be in world space

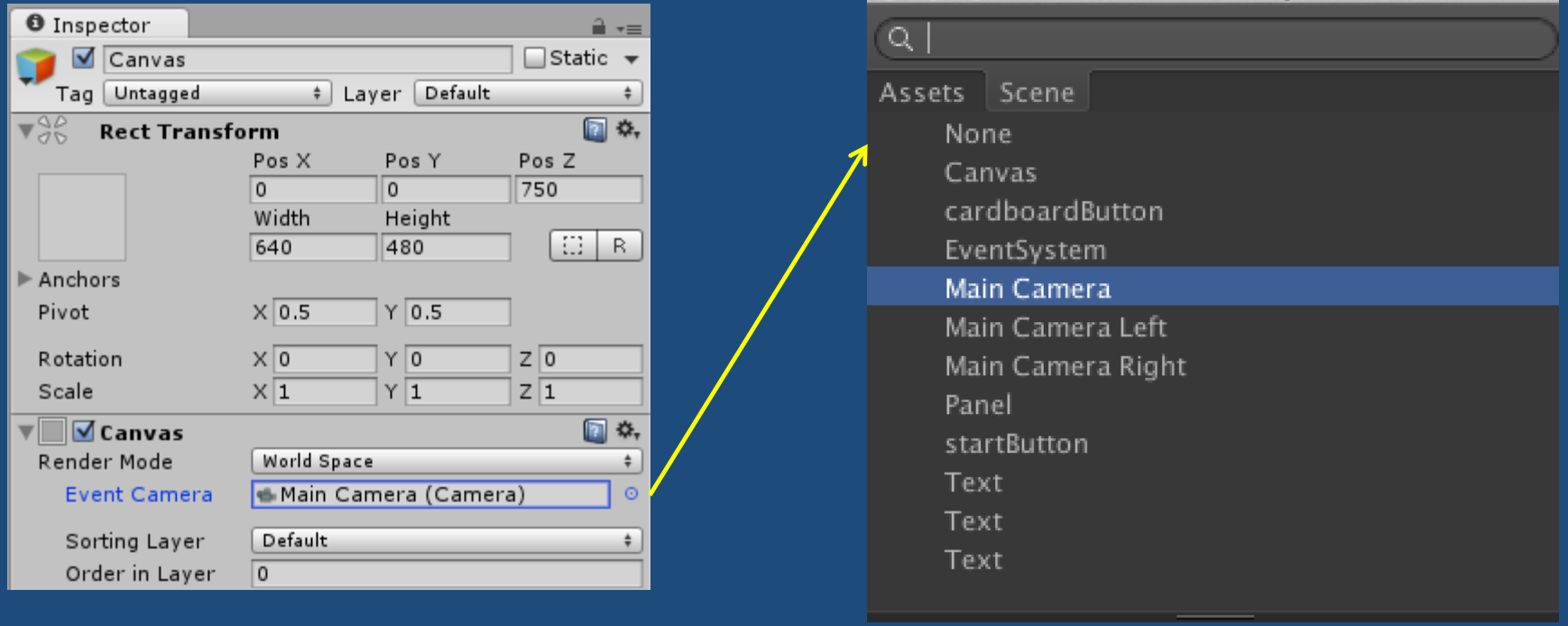
This step converts the main menu from being mapped to the **Screen Space** to be mapped in the **World Space** .

1. Select Canvas from the Hierarchy window, then in the inspector panel
   - Change Render Mode to World space
   - Change position to (0, 0, 750)
   - Change Width and Height to 640 and 480
   - pivot = (.5, .5)
   - Change scale to (1, 1, 1)


Inspector
Canvas
Static
Tag
Untagged
Layer
Default
Rect Transform
Pos X
Pos Y
Pos Z
0
0
750
Width
Height
640
480
R
Anchors
Pivot
X 0.5
Y 0.5
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Canvas
Render Mode
World Space
Event Camera
None (Camera)

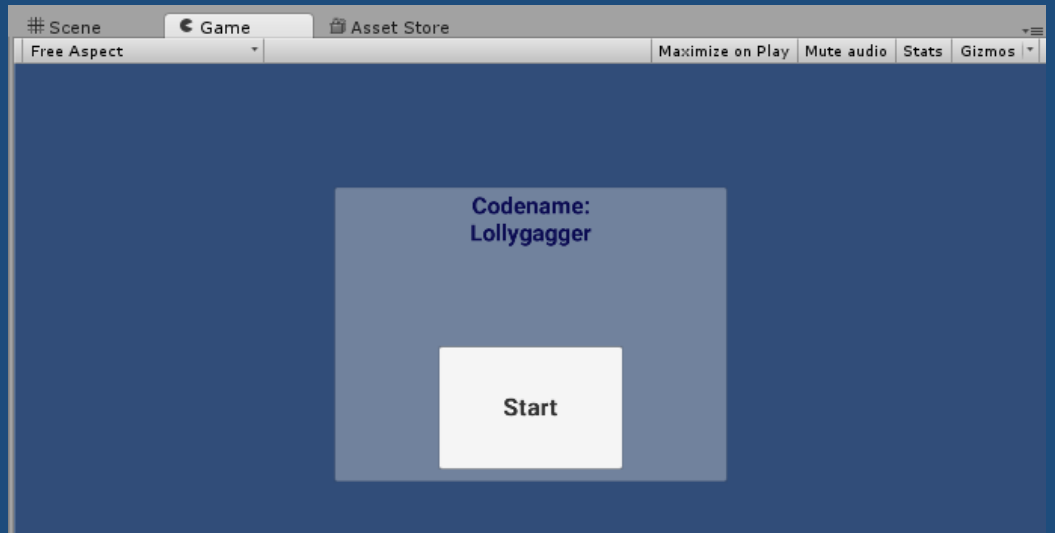
2. Associate the Main Camera with the Canvas.
   - This is done so that UI events can be translated from the screen coordinates to the world coordinates.

- Click on the Canvas
- In the properties inspector, click on the circle control next to the Event Camera which will bring up the object selection dialog.
- Select the Scene tab, then Main Camera object.


Inspector
Canvas
Static
Tag
Untagged
Layer
Default
Rect Transform
Pos X
Pos Y
Pos Z
0
0
750
Width
Height
640
480
R
Anchors
Pivot
X 0.5
Y 0.5
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Canvas
Render Mode
World Space
Event Camera
Main Camera (Camera)
Sorting Layer
Default
Order in Layer
0
Select GameObject
Assets
Scene
None
Canvas
cardboardButton
EventSystem
Main Camera
Main Camera Left
Main Camera Right
Panel
startButton
Text
Text
Text

3. Check the menu location and size.

- You can do this by clicking on the Game tab in the scene editor.

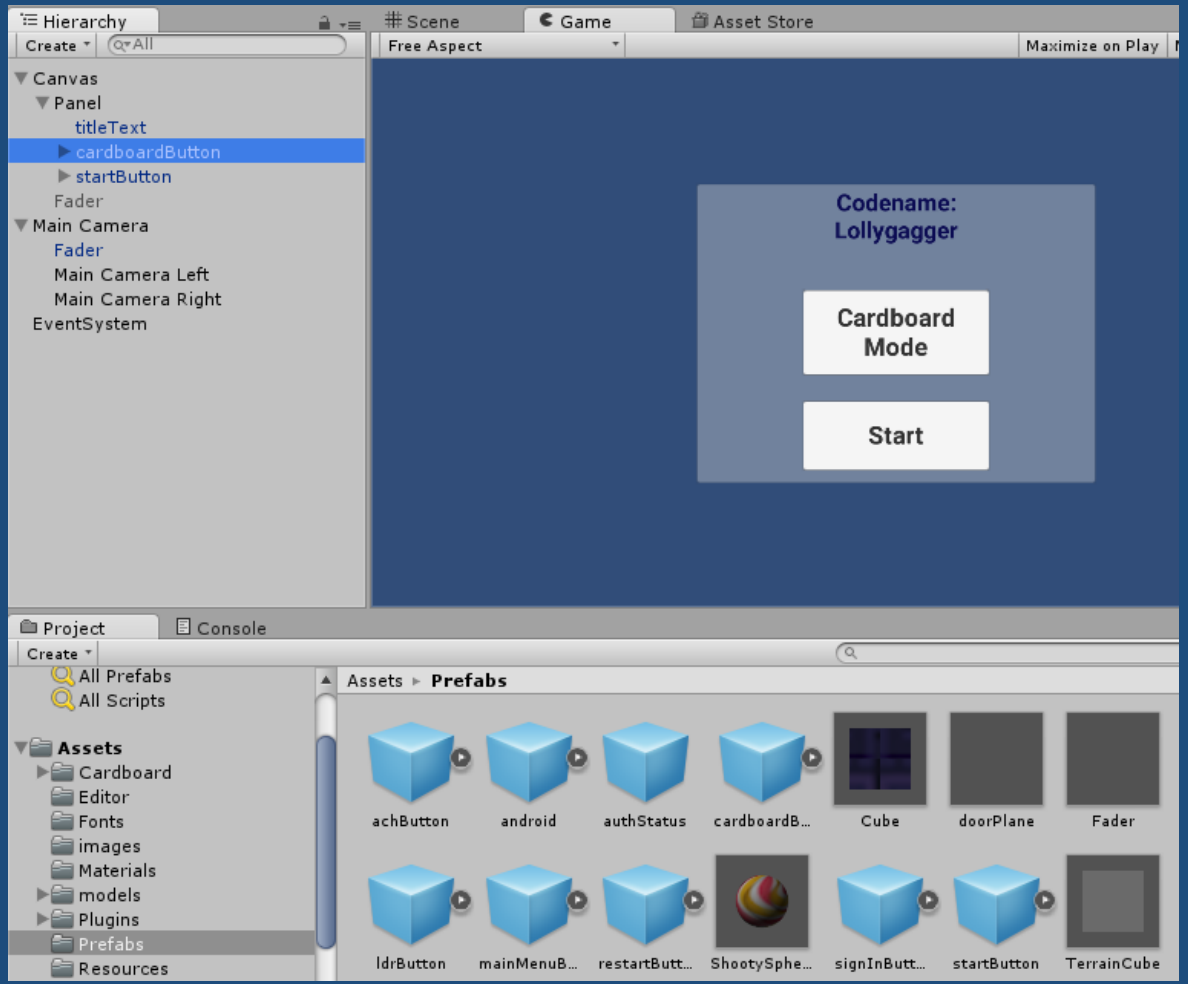


Add the Cardboard mode button to the menu

In this step, we add a button to the menu that will **turn on** and **off "Cardboard Mode"**.

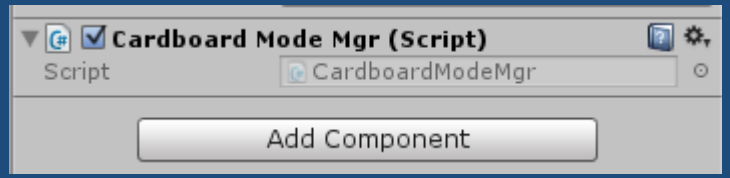
The prefab button

- Find the cardboardButton (Assets/Prefabs) prefab in the project explorer.
- In the Hierarchy panel, expand the Canvas object and the Panel object.
- Drag and drop the cardboardButton prefab onto the Panel object in the hierarchy.


Hierarchy
Create
All
Canvas
Panel
titleText
cardboardButton
startButton
Fader
Main Camera
Fader
Main Camera Left
Main Camera Right
EventSystem
Scene
Game
Asset Store
Free Aspect
Maximize on Play
Codename:
Lollygagger
Cardboard Mode
Start
Project
Console
Create
All Prefabs
All Scripts
Assets
Cardboard
Editor
Fonts
images
Materials
models
Plugins
Prefabs
Resources
Assets ▸ Prefabs
achButton
android
authStatus
cardboardB...
Cube
doorPlane
Fader
ldrButton
mainMenuB...
restartButt...
ShootySphe...
signInButt...
startButton
TerrainCube

Write the Cardboard Mode Script

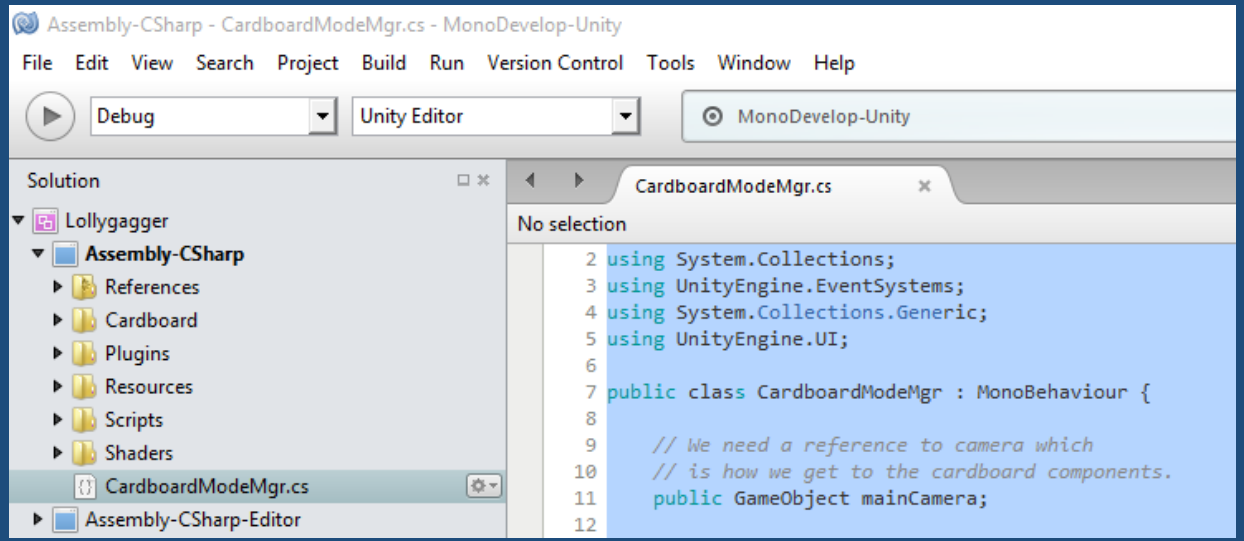
1. Create a new script component on the Canvas
   - Select the Canvas object in the hierarchy.
   - In the properties panel, scroll to the bottom and click Add Component



- In the list of component types, scroll to the bottom and click **New Script**.
- Name the script **CardboardModeMgr**, and keep the language **CSharp**.
- Click **Create and Add**.

Handle Cardboard mode

- Open **CardboardModeMgr** script by double-clicking on it, delete the entire contents and paste in the following code.



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

public class CardboardModeMgr : MonoBehaviour {

    // We need a reference to camera which
    // is how we get to the cardboard components.
    public GameObject mainCamera;
```

```
    public void Start()
    {
        // Save a flag in the local player preferences to init
ialize VR mode
        // This way when the app is restarted, it is in the m
ode that was last used.
        int doVR = PlayerPrefs.GetInt("VREnabled");
        Cardboard.SDK.VRModeEnabled = doVR == 1;
        CardboardHead head = mainCamera.GetCompone
nt<CardboardHead>();
        head.enabled  = Cardboard.SDK.VRModeEnabled;
        Cardboard.SDK.TapIsTrigger = true;
    }
```

```
// The event handler to call to toggle Cardboard mode.
    public void ChangeCardboardMode ()
    {
        CardboardHead head = mainCamera.GetComponent<CardboardHead>();
        if (Cardboard.SDK.VRModeEnabled) {
            // disabling.  rotate back to the original rotation.
            head.transform.localRotation = Quaternion.identity;
        }
        Cardboard.SDK.VRModeEnabled = !Cardboard.SDK.VRModeEnabled;
        head.enabled  = Cardboard.SDK.VRModeEnabled;
        PlayerPrefs.SetInt("VREnabled", Cardboard.SDK.VRModeEnabled?1:0);
        PlayerPrefs.Save();
    }
}
```

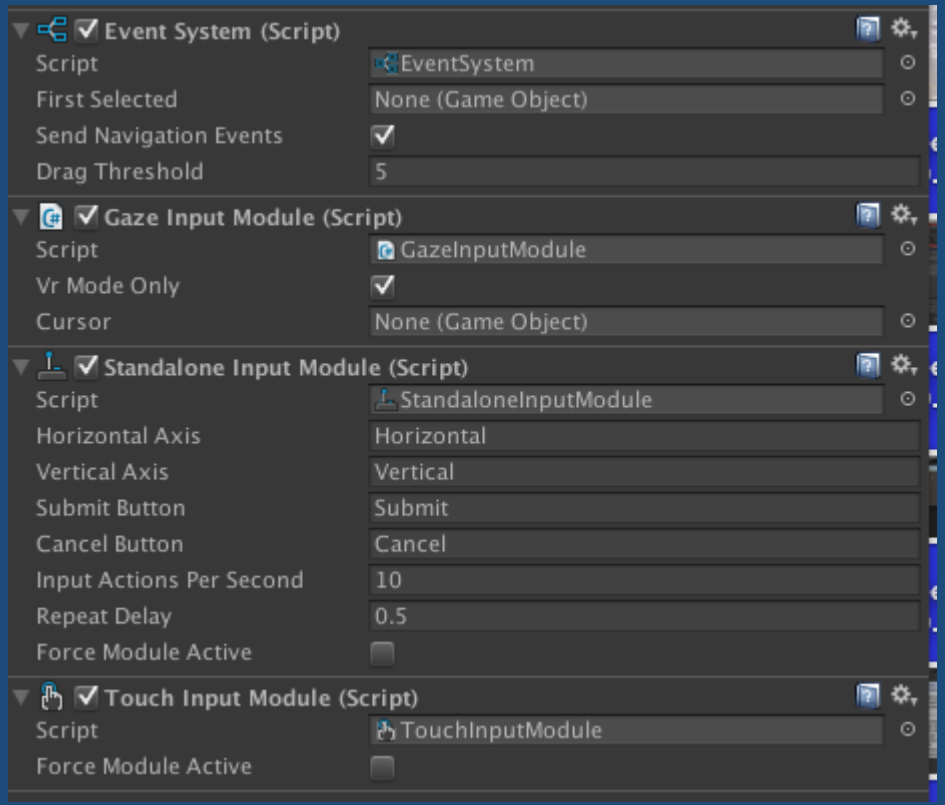
Pro Tip

- When pasting code into the editor, the script editor sometimes gets out of sync with the Unity project.
  - This is fixed by selecting **Assets** > **Sync Monodevelop Project** (or **Assets** > **Open C# Project**) from the Unity main menu.

Handle VR pointing and clicking

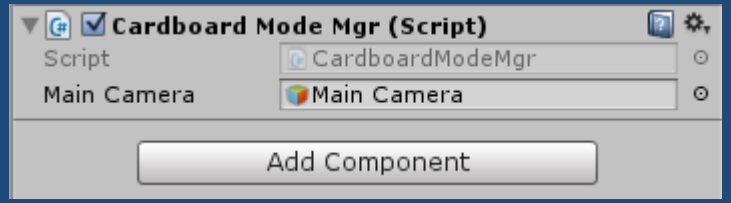
In order to control the game play while in **cardboard mode**, we need to add an input module to the event system that uses the **player's gaze** to select or point to objects and use the **cardboard trigger** as an input event.

- In the scene hierarchy, select the EventSystem object.
- From the menu, select Component > Scripts > Gaze Input Module.
- Make sure the Gaze Input Module is the first input module in the list.
- Check the box Vr Mode Only in the properties of the Gaze Input Module.


Event System (Script)
Script
EventSystem
First Selected
None (Game Object)
Send Navigation Events
Drag Threshold
5
Gaze Input Module (Script)
Script
GazeInputModule
Vr Mode Only
Cursor
None (Game Object)
Standalone Input Module (Script)
Script
StandaloneInputModule
Horizontal Axis
Horizontal
Vertical Axis
Vertical
Submit Button
Submit
Cancel Button
Cancel
Input Actions Per Second
10
Repeat Delay
0.5
Force Module Active
Touch Input Module (Script)
Script
TouchInputModule
Force Module Active

Associate the camera with the other game objects

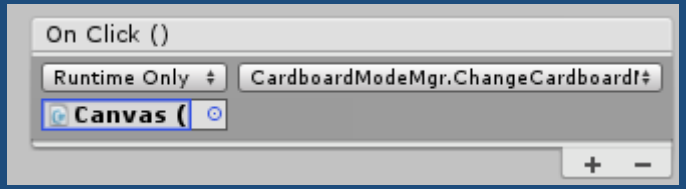
- In order to gain access to the user's direction of view, or gaze, the main camera needs to be made available to these components.
  - Select the Canvas object in the hierarchy.
  - Set Main Camera for the Cardboard Mode Mgr component.



Associate the click event with the script

- In the scene hierarchy, select the buttonCardboard object(under Canvas/Panel) .
- In the properties inspector, scroll down to the On Click () section.
- Click the + to add another handler.

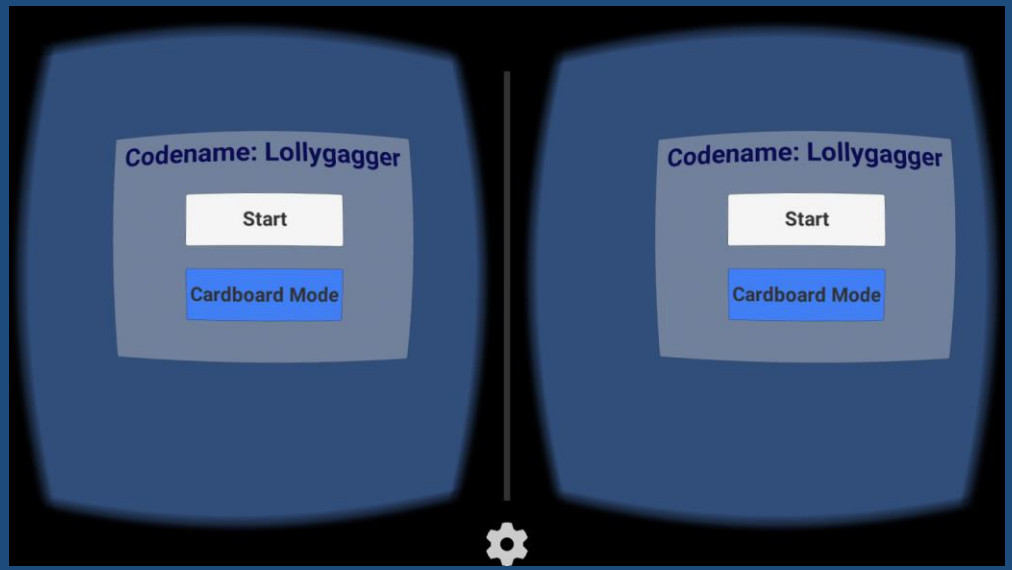
- Make sure **Runtime Only** is selected
- For the game object, click the circle, and under the scene tab, select **Canvas** by double clicking.
- Drop down the function selection and select **CardboardModeMgr.ChangeCardboardMode()**
- Save the scene, and Save the project!



Testout the main menu!

- But enough sanity! Let's **build** the project again and **run** the app.
- Click on the **Cardboard Button** to go into VR mode, and put your phone in the Cardboard device.

- Look around, see how the buttons become selected by just looking at them!



Add Cardboard to the Game Scene

1. This lesson updates the main game scene to be VR enabled.
2. Specifically adding the stereoscopic rendering, updating menu UIs and adding trigger support.

Add Cardboard view to the Game Scene

- Open the game scene.
  - In the project explorer, select the Assets folder, and the double click the GameScene scene.
- Select the Main Camera in the Scene hierarchy.
  - Since the game play is in first person, the main camera is a child object of the player.

- From the menu, click **Component** > **Cardboard** > **Update Stereo Cameras**. This does the following:
  - Adds StereoController
  - Adds CardboardHead
  - Creates Left and Right child cameras
  - Also adds SkyboxMesh

- **Note**: If the menu item is not there, you'll need to re-import the assets by clicking **Assets** > **Reimport All**.

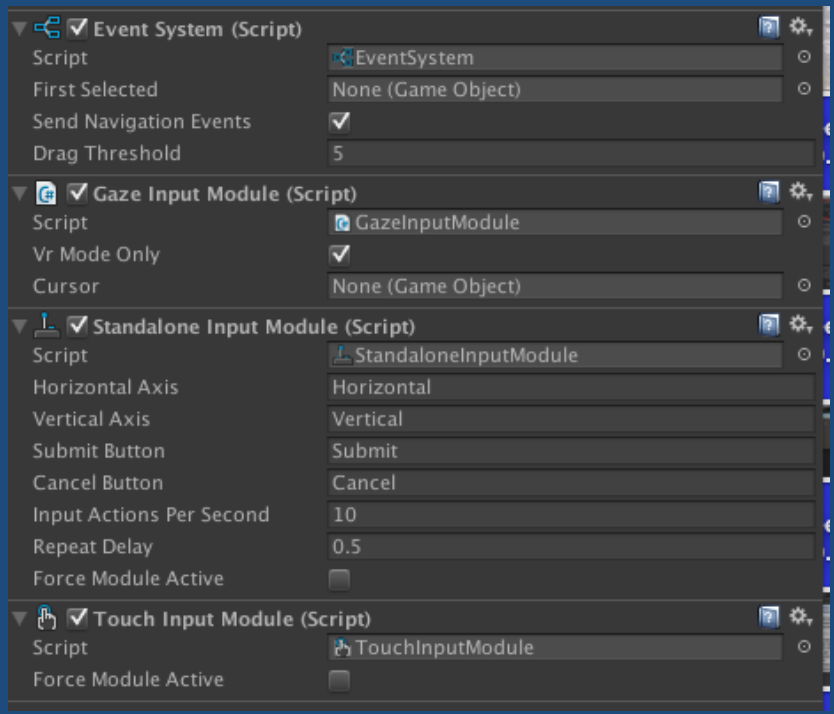
Add Cardboard Mode Manager script

- Adding this script will initialize cardboard based on the mode selected in the main menu.
  - In the scene hierarchy, select the Canvas object.
  - From the menu, select Component > Scripts > Cardboard Mode Mgr.
  - Set the Main Camera property in the script to the Main Camera.

Add the GazeInputModule script to the EventSystem

- The GazeInputModule handles input events, so it needs to be on every scene.
  - In the scene hierarchy, select the EventSystem object.
  - From the menu, select Component > Scripts > Gaze Input Module.

- Make sure the **Gaze Input Module** is the **first** input module in the list.
- Check the box **Vr Mode Only** in the properties of the **Gaze Input Module**.



Save and Run!

- File > Save scene
- File > Save project
- File > Build and run

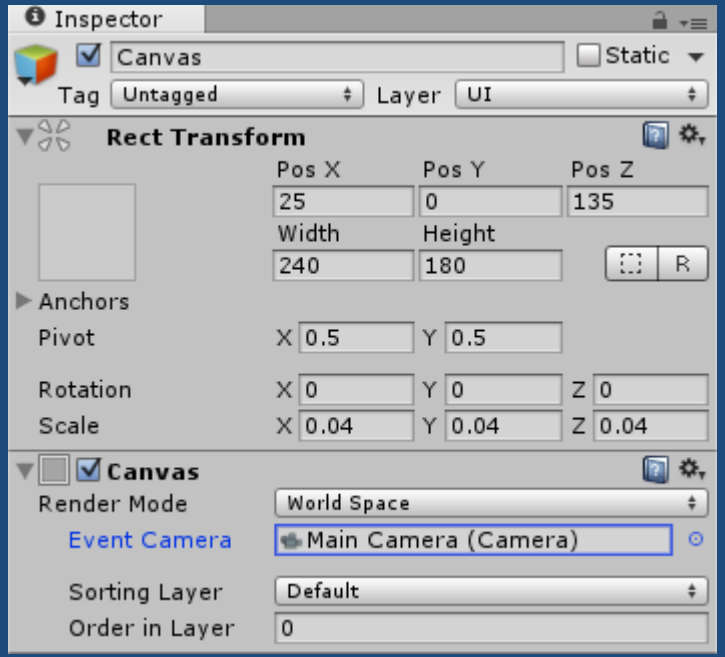
Change Aiming to be based on the gaze of the player

- In the 2D world, the finger touch is used to rotate the player in order to aim and shoot.
- In VR mode, the rotation is done by the actual user, and the Cardboard SDK handles that for you!

Make the ending menu VR compatible

- Select Canvas from the Hierarchy window, then in the inspector panel:
  - Change **Render Mode** to **World space**
  - Change **position** to **(25, 0, 135)**
  - Change **Width** and **Height** to **240** and **180**
  - **pivot (.5, .5)**
  - Change **scale** to **(0.04, 0.04, 0.04)**

- Set the **Event Camera** to be the **Main Camera**
- **Save** the project
- **Build & run** and be amazed!



Congratulations!!!

- You've just taken a giant step to the exciting world of Cardboard using Unity.
- But as you can imagine, there are so many more facets to creating awesome Cardboard games.

New to Cardboard?

If you are new to Cardboard, or want to explore it some more, your can create a new Unity project and import the **CardboardDemoForUnity** package and **run it!**

天津GDG社区，2016年3月12日

谢 谢！

师文轩，13920561100