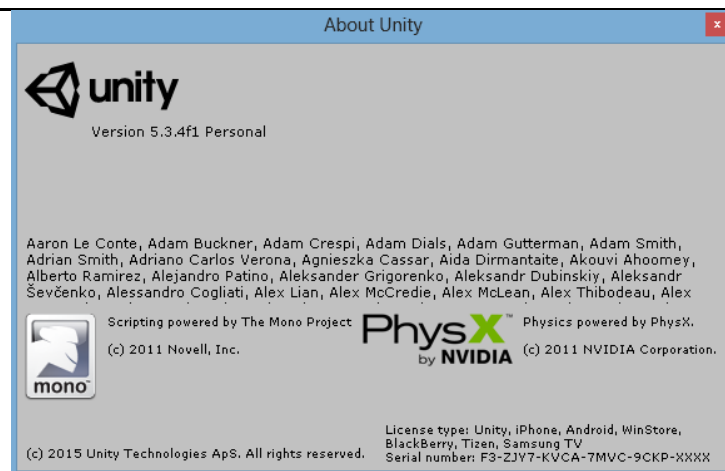


Cardboard VR Gallery Stereo Image Viewer



1. GENERAL INFORMATION

DATE OF DOCUMENT	20/04/2016
NAME OF THE PROJECT	Cardboard VR Gallery Stereo Image Viewer
AUTHOR	Michael Soler
UNITY VERSION	5.3.4.F1 PERSONAL



Index

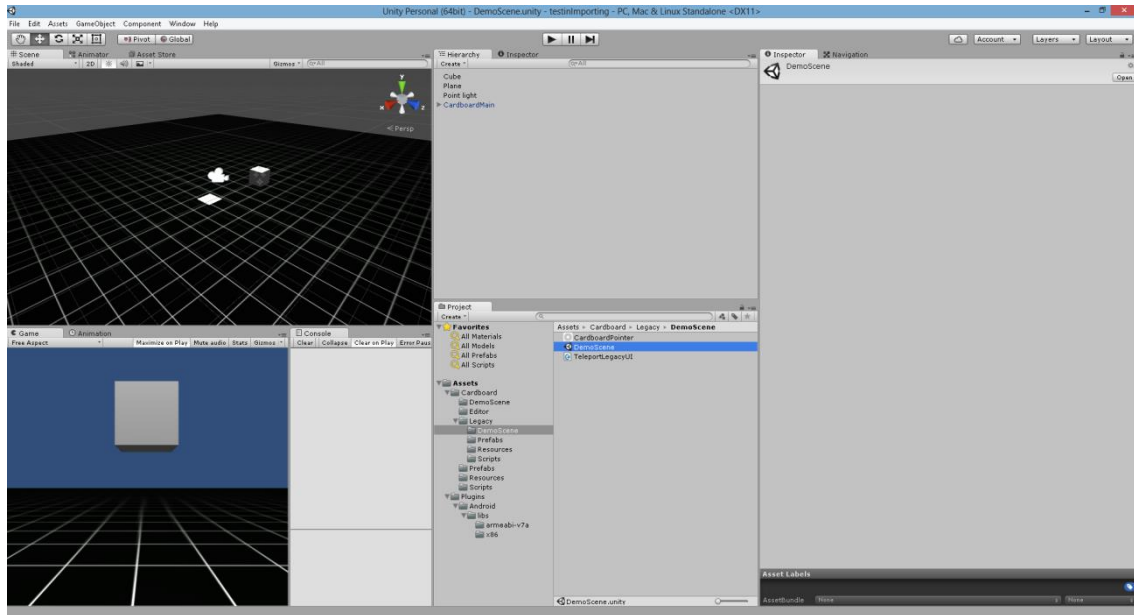
1. GENERAL INFORMATION.....	1
2. IMPORTING INFORMATION	2
3. PROJECT DESCRIPTION	4
4. LAYERS, TAGS AND COLLIDERS.....	5
5. SCRIPTING INFORMATION.....	6
6.MOVEMENT AND SETUP ON ANDROID DEVICE	¡Error! Marcador no definido.

2. IMPORTING INFORMATION

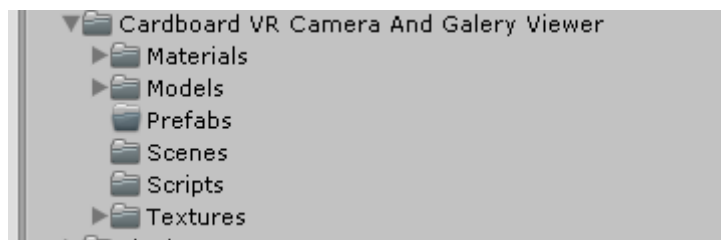
This package works with the “google cardboard” for UNITY that must be downloaded first using the following link:

<https://developers.google.com/cardboard/unity/?hl=en>

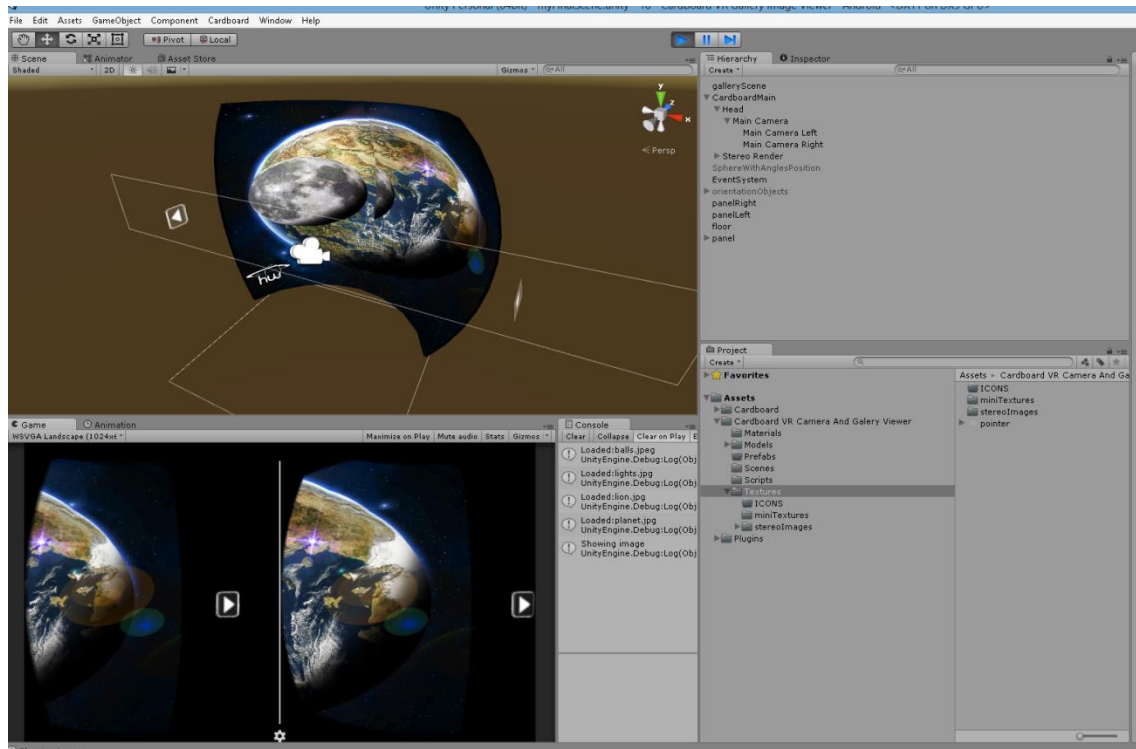
Once downloaded and imported to unity, your project should look like this:



Then, import our package to the project, which will leave you the following folder configuration:



The gaze input collider must be disabled. There is no need to change collider or other game objects.



3. PROJECT DESCRIPTION

This package allows the user to create your own VR gallery using stereo images. You will be able to plot single and separated stereo images and loop images in a gallery.

This package includes:

- Two “curved” panels used for VR images.
- Stereo Scripts that automatically update the number of pictures on your folders.
- Simple and Commented code.
- Functional demo version for PC.
- Example textures of both types.

This package works with the “google cardboard” for UNITY that must be downloaded first using the following link:

<https://developers.google.com/cardboard/unity/?hl=en>

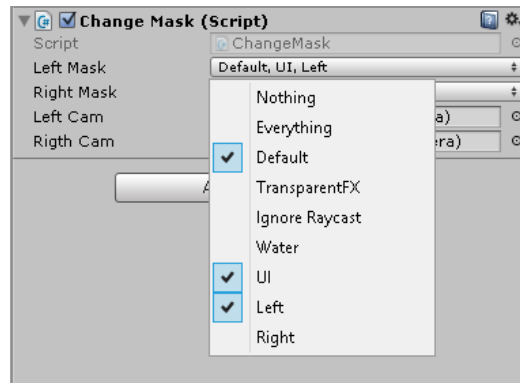
On Android you have to change and adapt the directory reference on “LoadGallery.cs”:

“DirectoryInfo info = new DirectoryInfo(Application.dataPath + ...)” and use an absolute reference to your internal or SD memory. Example line of code is on the documentation.

4. LAYERS, TAGS AND COLLIDERS

LAYERS

It is important to notice the different layer that we are using:



- Default will render in both cameras.
- UI will render in both cameras
- Left will only render o left camera.
- Right will only render on right camera.

TAGS:

Gameobjects are untagged.

COLLIDERS

No colliders are used. Only graphical-raycaster.

Check if the “gaze pointer cursor” collider is disabled:

It is important to import the cardboard package correctly, and check if these colliders are working properly.

5. SCRIPTING INFORMATION



We explain each script with some detail in the following table:

- **LoadGallery:**

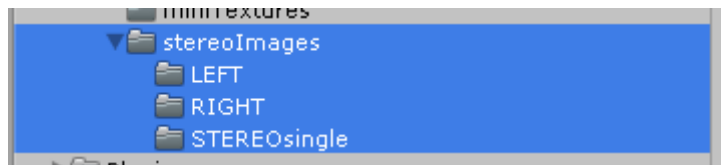
This script controls the generation of the gallery and it is the main asset of the package:

IMPORTANT VARIABLES
Public string path → it is the path of the files on PC. Has to be changed on Android. public Material panell,panelR; → Both curved panels for Stereo Pair. public TypeOfStereo imageType; → Used to determine whether the image is a single file or two files. public int Imageindex=0; → current image. public int maxImageNumber; → max number of images on gallery. public Texture2D[] textL,textR; → array of stereo images.
IMPORTANT FUNCTIONS
public void loadImageStereo(int x) → Loads the Texture “x” of the arrays textL and textR. nextImage() → increments x+1 previousImage() → decrements x-1

All scripts are commented on the “*.cs” files.

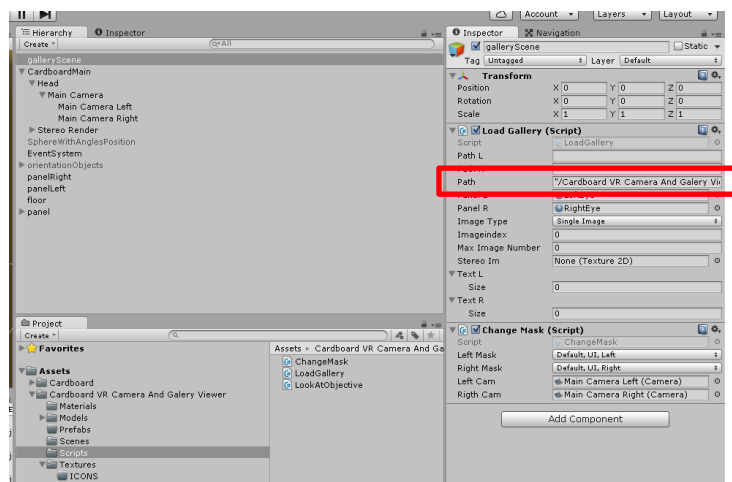
7.ANDROID INTEGRATION

You will have to create the following folders on your SD or internal phone memory:

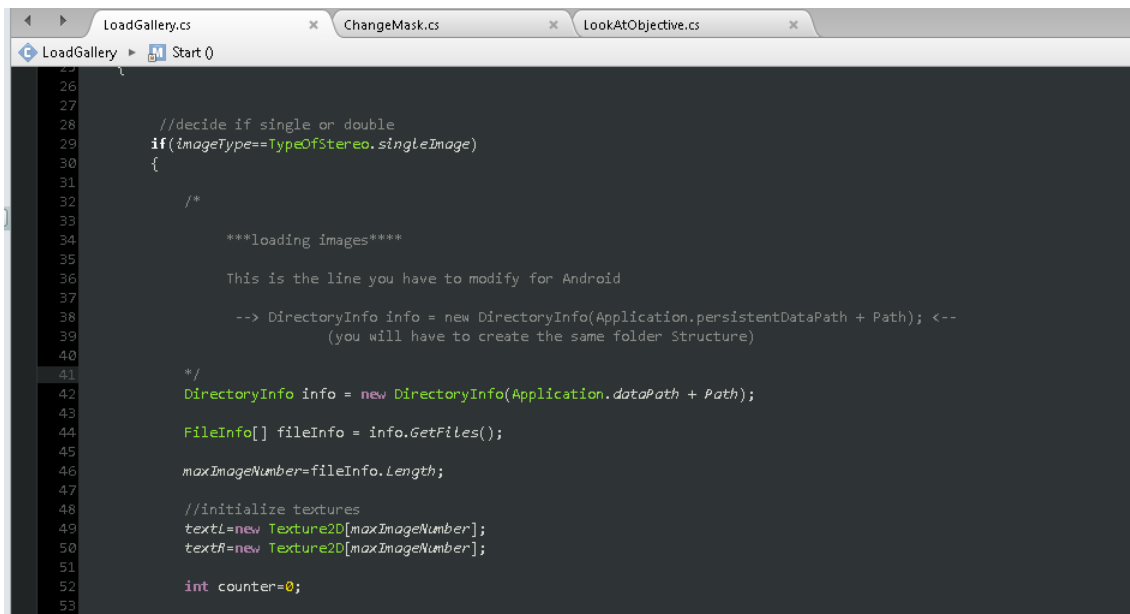


Then refer to your memory allocation (change folder path) as:

```
"/storage/extSdCard/stereolImages" → LEFT
                                     → RIGHT
                                     → STEREOsingle
```



Finally you will have to change the line of code in LoadGallery.cs



```
26
27
28 //decide if single or double
29 if (imageType == typeof(Stereo.SingleImage))
30 {
31
32     /*
33     ***loading images***
34
35     This is the line you have to modify for Android
36
37     --> DirectoryInfo info = new DirectoryInfo(Application.persistentDataPath + Path); <--
38         (you will have to create the same folder Structure)
39
40
41     */
42     DirectoryInfo info = new DirectoryInfo(Application.dataPath + Path);
43     FileInfo[] fileInfo = info.GetFiles();
44
45     maxImageNumber = fileInfo.Length;
46
47     //initialize textures
48     textL = new Texture2D(maxImageNumber);
49     textR = new Texture2D(maxImageNumber);
50
51     int counter = 0;
52
53 }
```

Then, insert your images in the proper folders (on your device).