Google Street View App

Google's updated app now supports Cardboard, meaning we can take a virtual tour of just about anywhere.

Cardboard, of course, is the cheap virtual-reality headset that works with smartphone. And Street View is the mobile version of Google's ground-level Maps feature. Put the two together and presto:

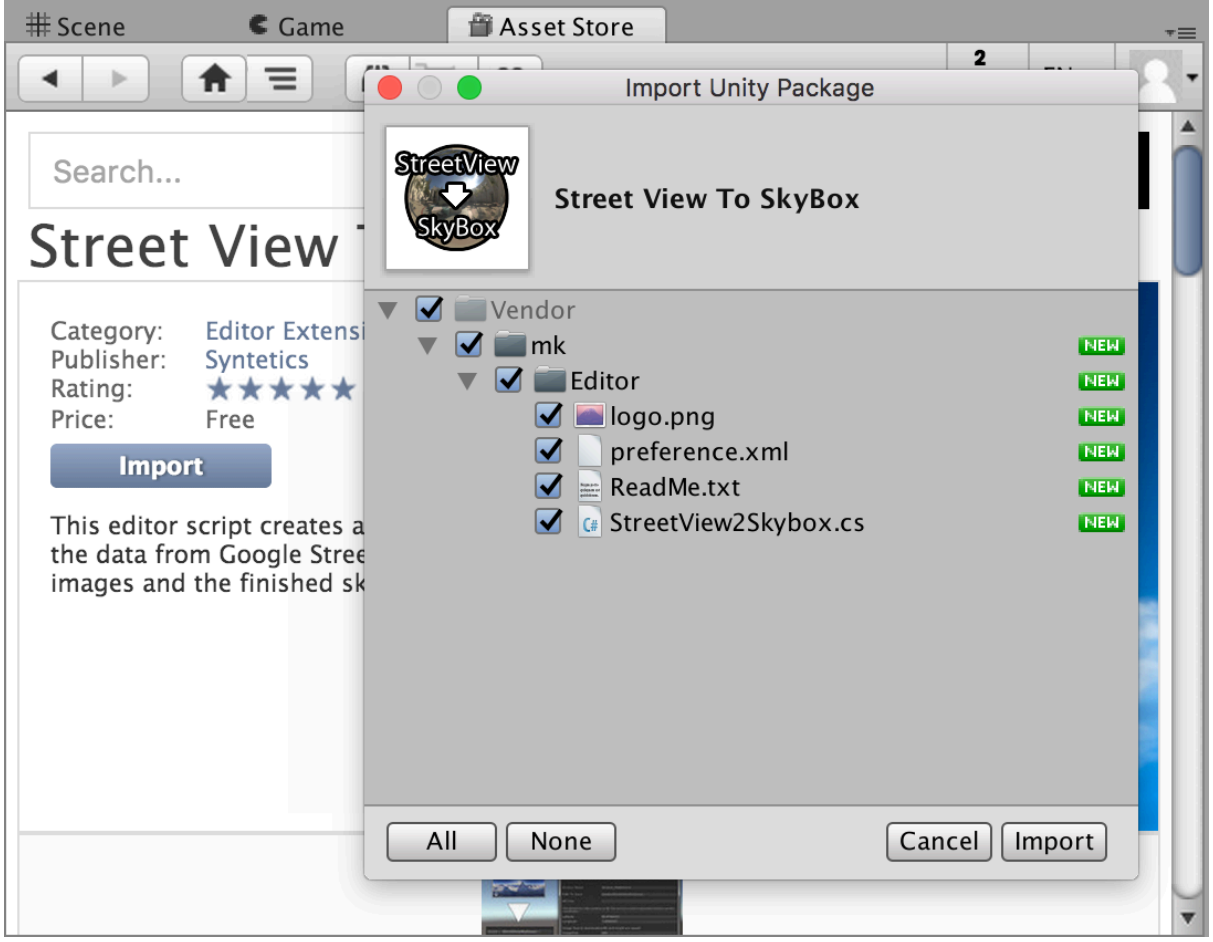
`Now we get a virtual-reality view of anywhere we want to visit in Street View.`

Street View to Skybox

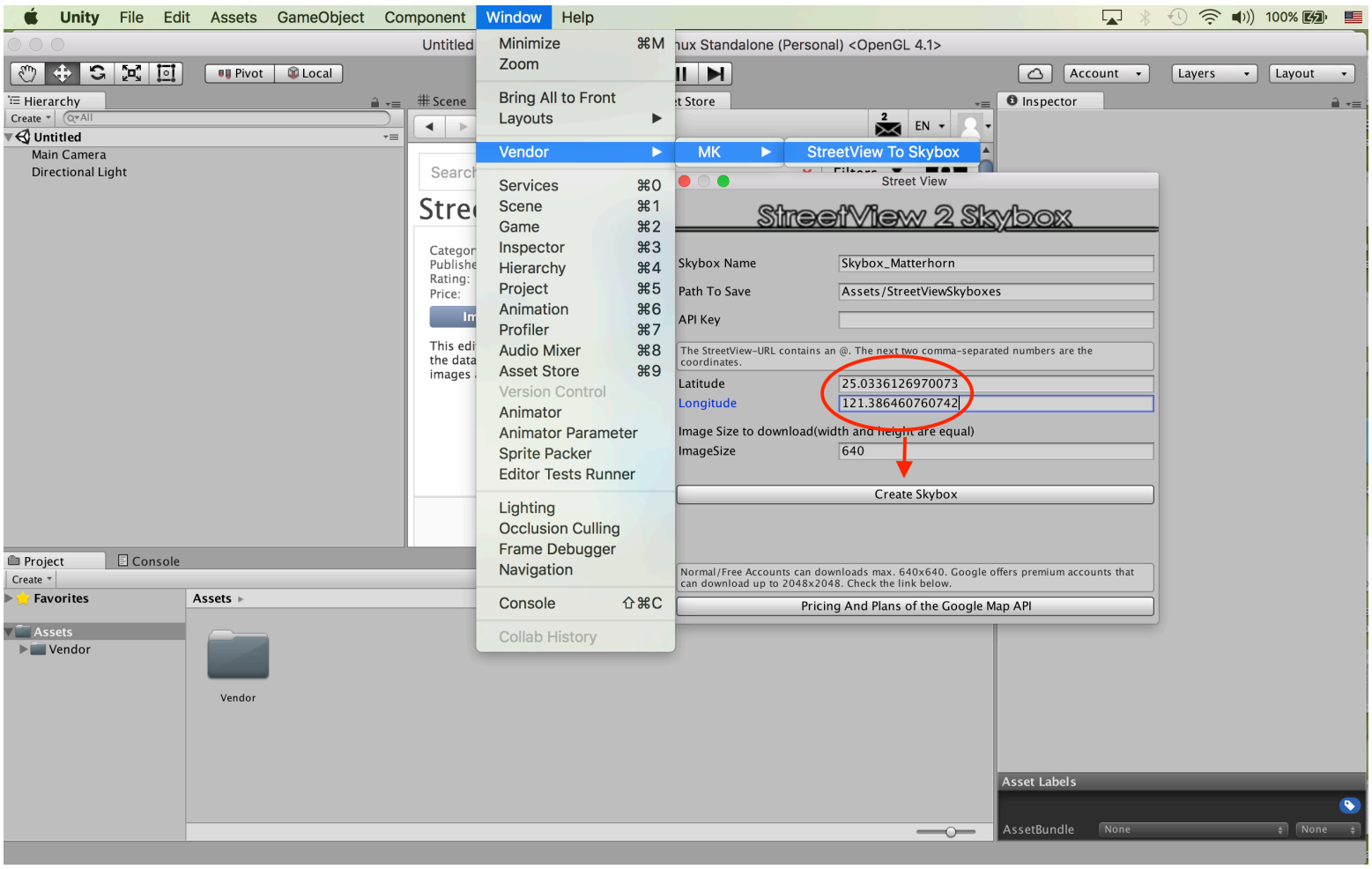
Download and import package of `Street View to Skybox` , (having Problems)

1. Find it and open in Unity

2. Login unity and import it:



3. Select **[Window]->[Vendor]->[Mk]->[StreetView To Skybox]** input the latitude/longitude of center, found from address (http://diffusion.cgu.edu.tw/2014/computer/2014-2/GPSCoord-4.html) and create the skybox:

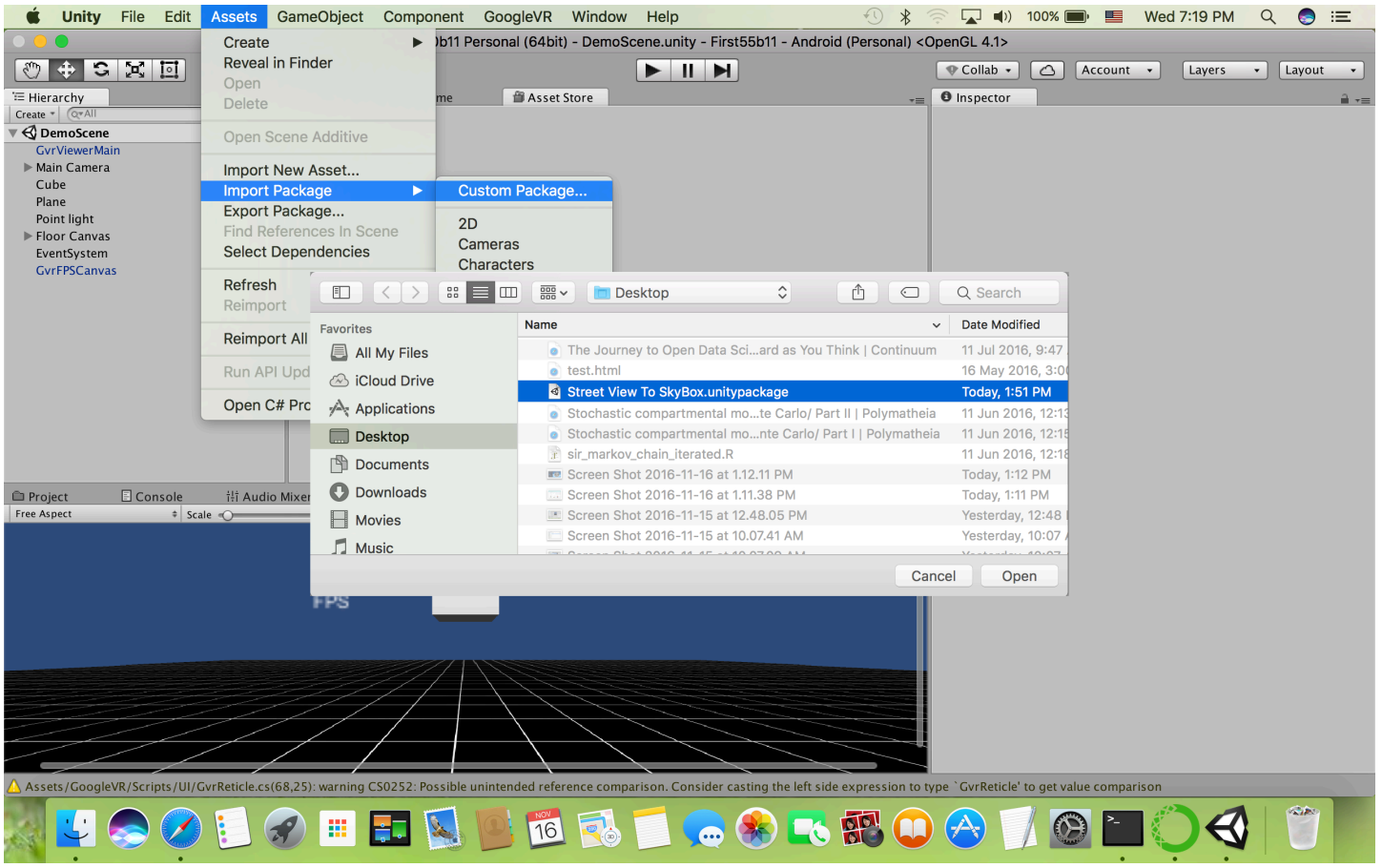


Note

- ~~While updated to last Unity-5.5.0b11, the unit package can not be download again~~. Find the download package from

`[user/Library/Unity/Asset Store-5.x/Syntetics/Extensions/Street View To SkyBox.unitypackage]`

and import from unity main menu:



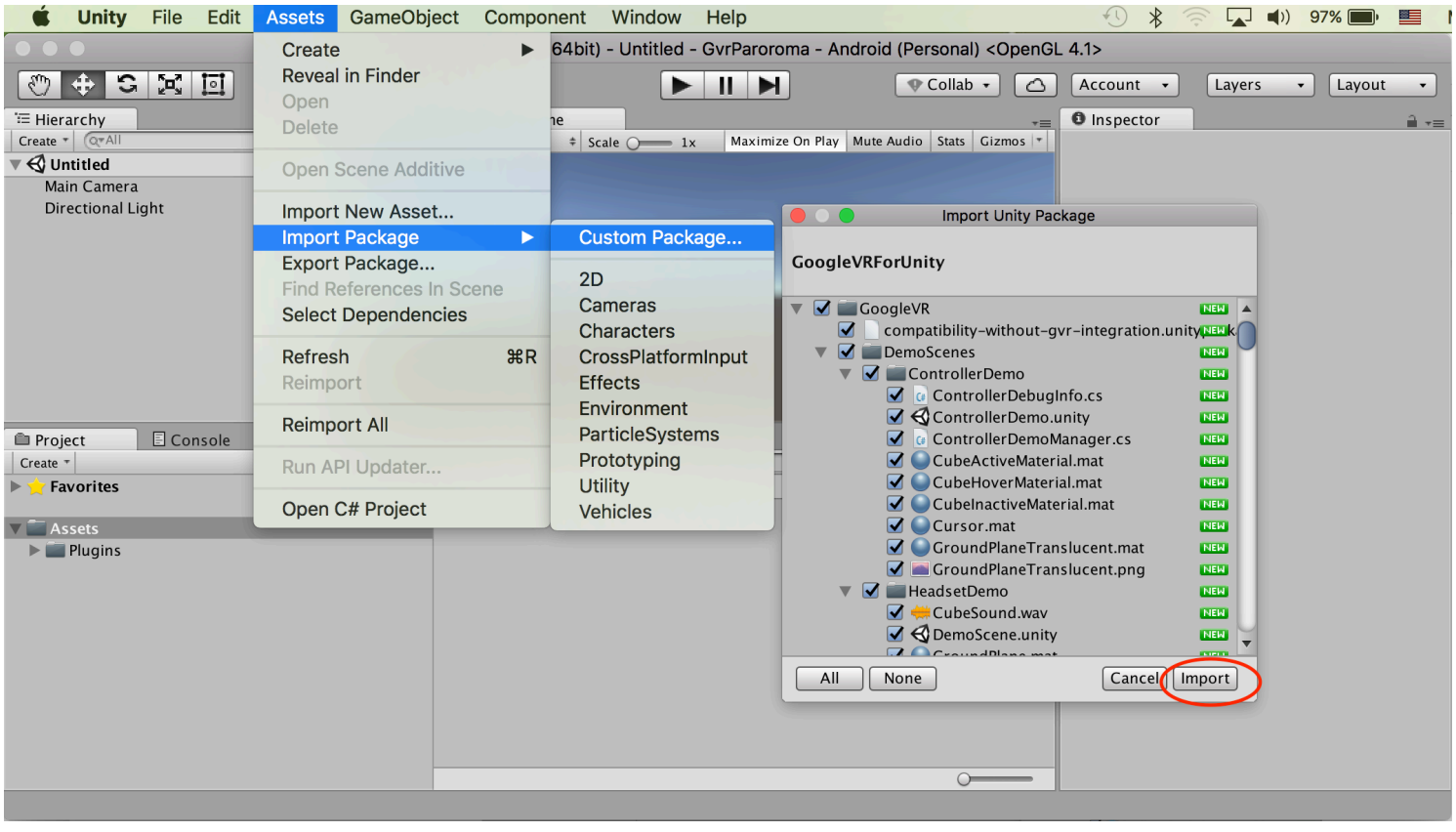
- but Unity-5.5f1 works normally again.

4. import the CardboardSDK (v 1.0.3) for Unity package by going to:

- **Assests > Import Package > Custom Package**

select the Google Unity Plugin:

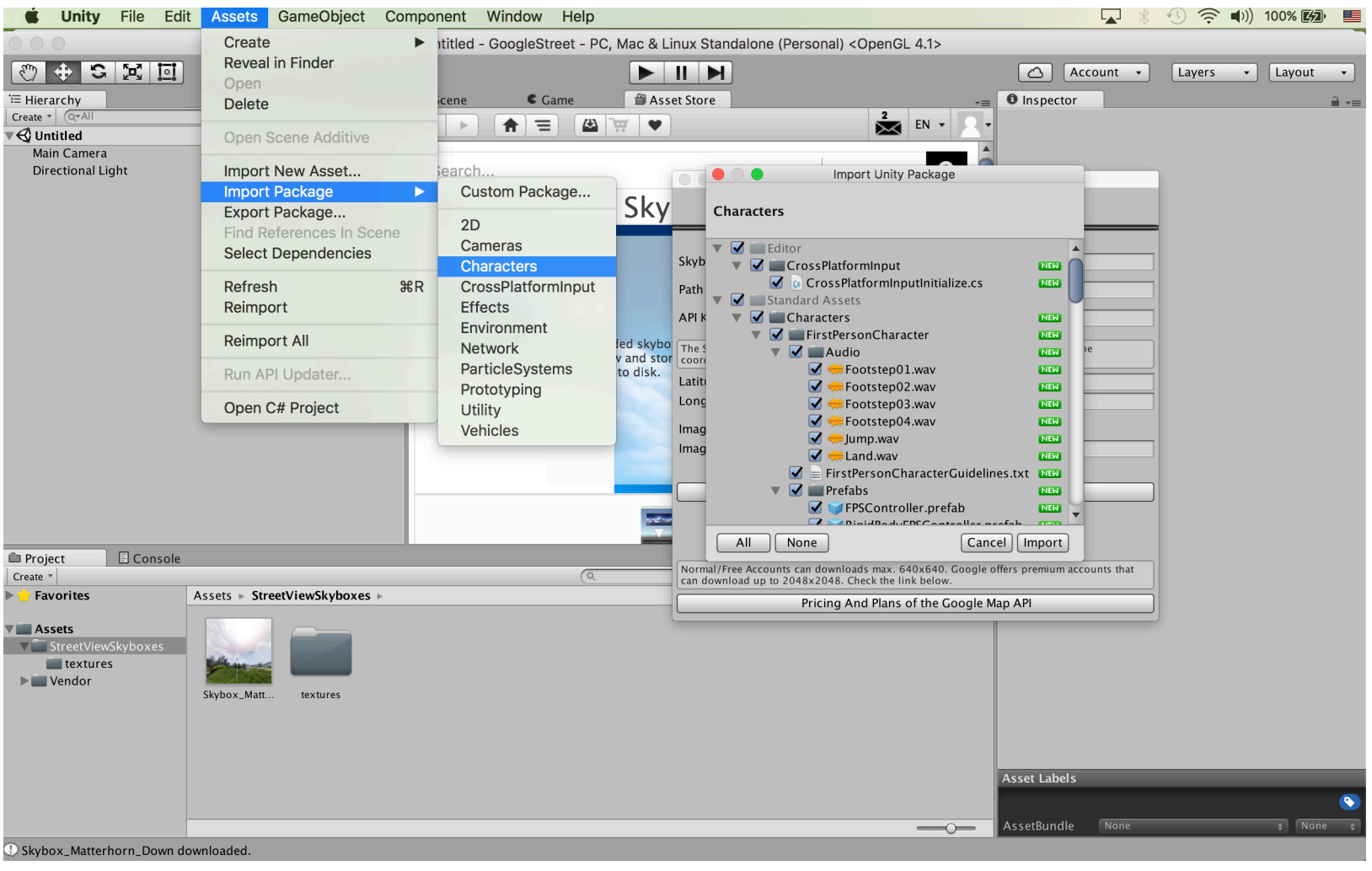
- GoogleVRForUnity.unitypackage
- ~~CardboardSDKForUnity.unitypackage (pre-0.8)~~ and
- ~~CardboardDemoForUnity.unitypackage (pre-0.8)~~;



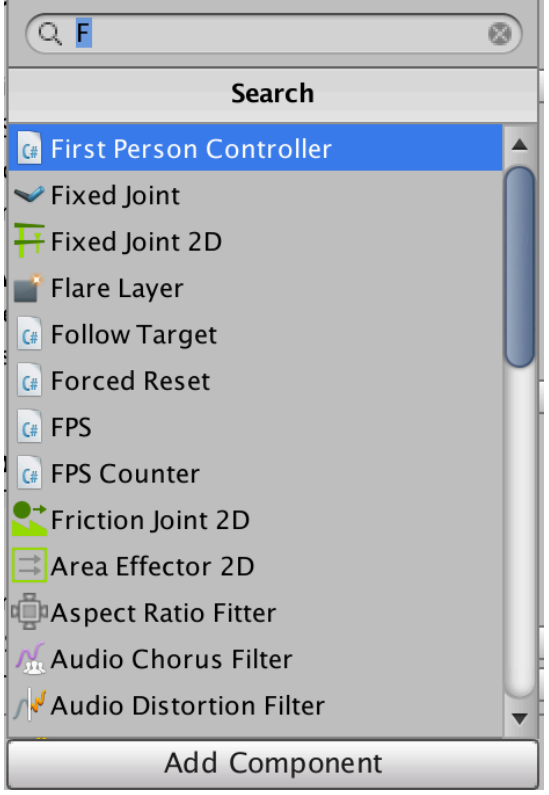
  - Open the **GoogleVR/Prefabs** folder. Drag CardboardMain.prefabs into the **Hierachy Windows**

~~If nothing option was found in last step, drag `Standard Assets` into [Project] Assets.~~

5. Import package from **[Assets]->[Import package]->[Characters]** and

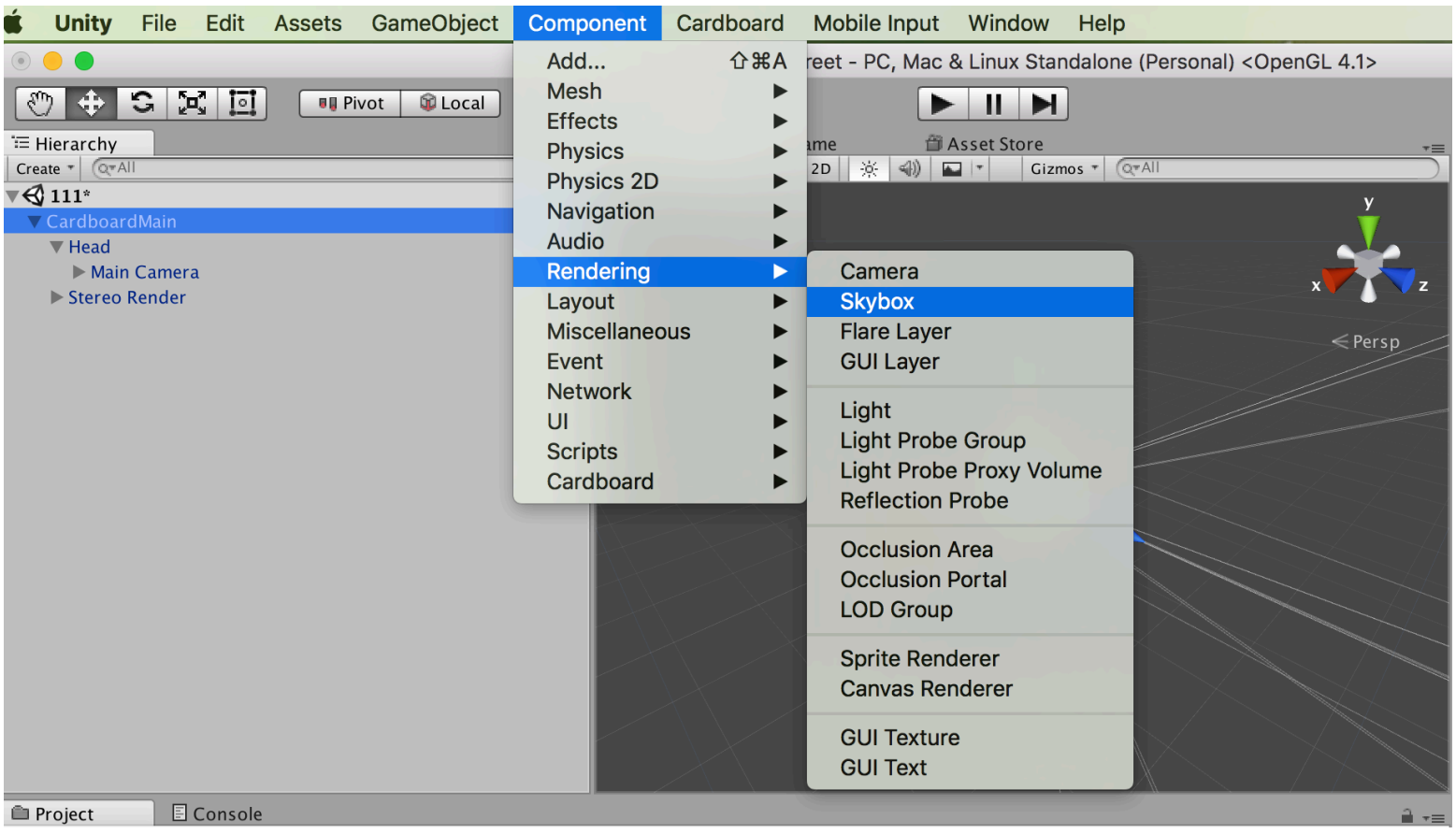


5. From right column, add component, **[First Person Controller]**, for *CardboardMain* in Hierarchy Windows:

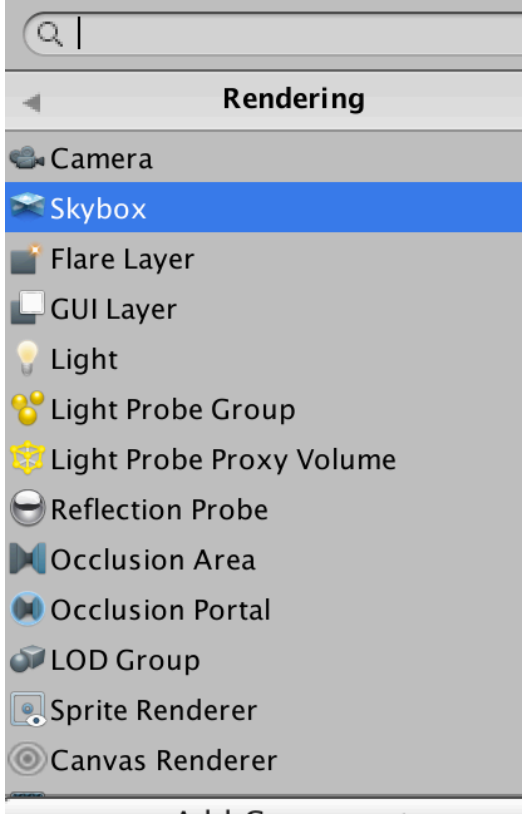


Skybox

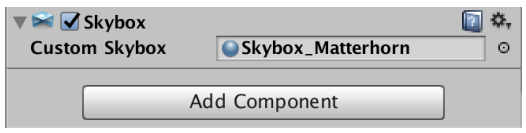
1. Add Skybox by **[Component]->[Rendering]->[Skybox]**



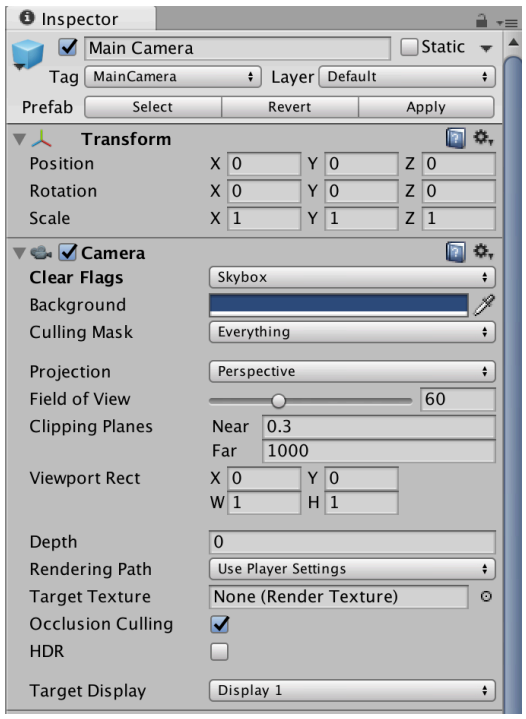
2. In **Inspector Windows**, [Add Component] for `CardboardMain -> Main Camera` in **Hierachy Windows**,



also change the property of `Skybox`,



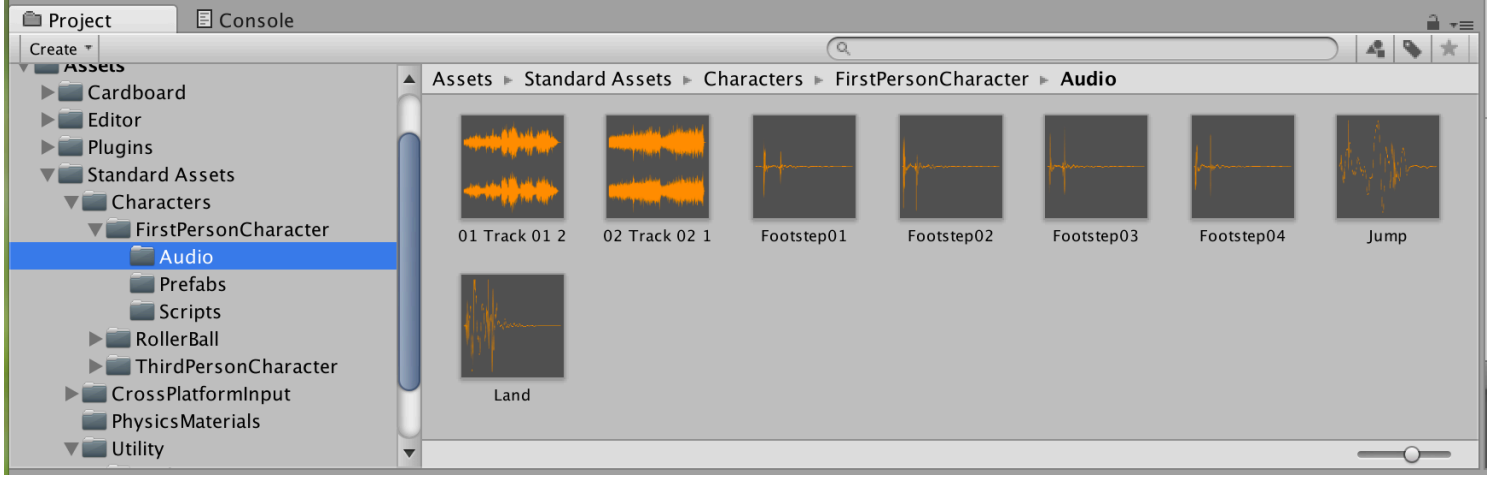
and the property of `clear flags` to *Skybox* in **Camera**,



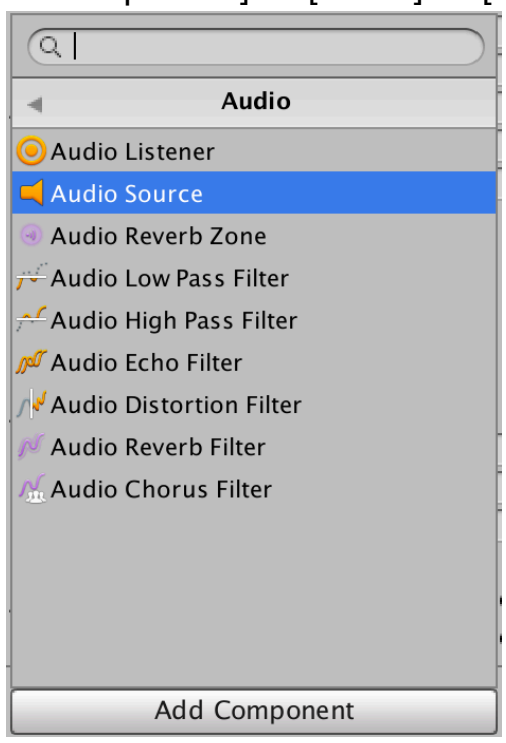
Now we can test it or build the app.

Note

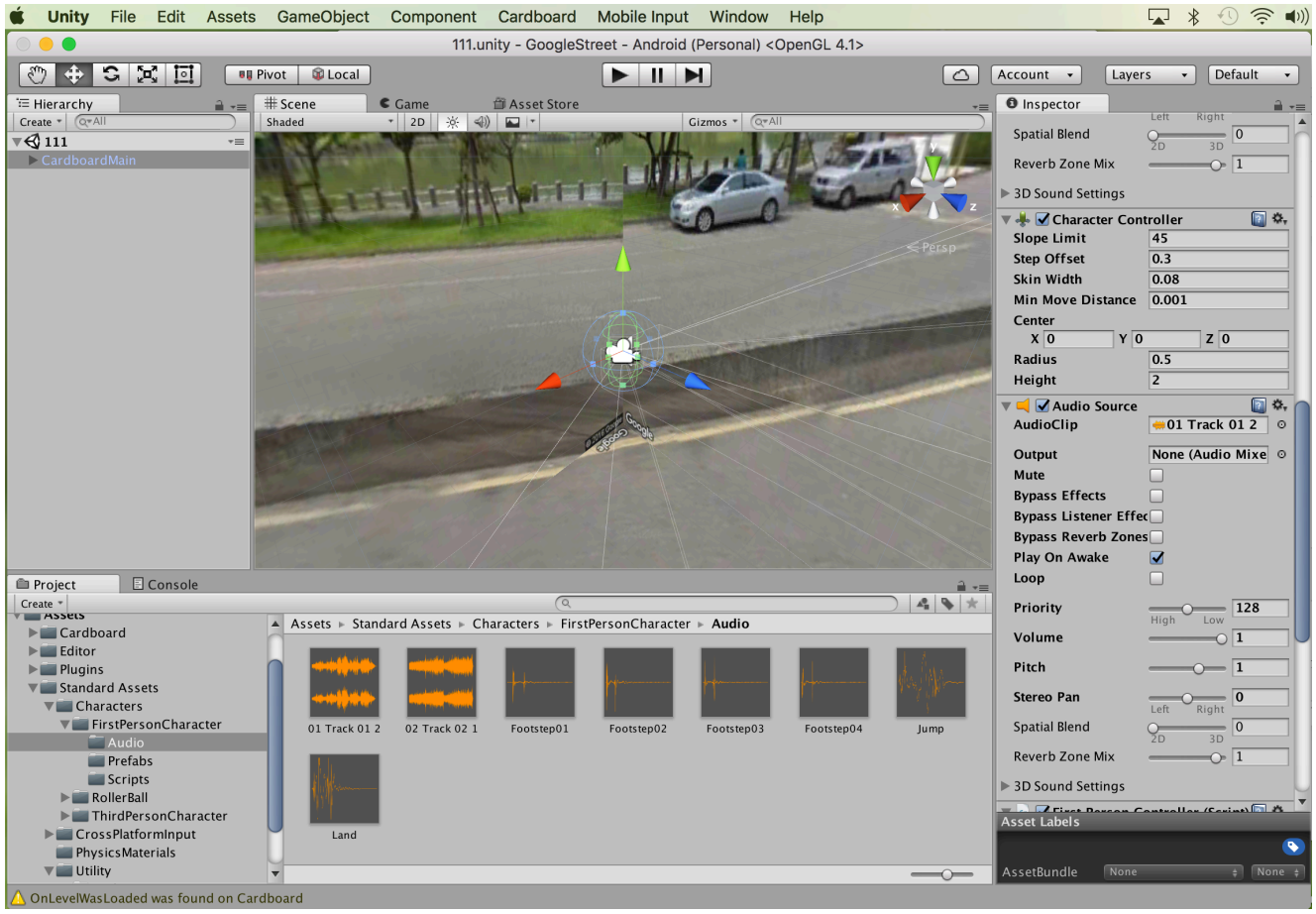
1. Sound Effect:
    - add the sound file by importing [new Asset] or dragging the file into certain folder:



- Add Sount for CardboardMain, [Add Component] -> [Audio] -> [Audio Source]:



- Drag the sound asset into [Audio Source] -> [AudioClip]:



2. UI: to test the app, (Apple Case):
    - hold [control] key to rotate the scence;
    - hold [option] key to look around.

3. HTML/Javascript codes to get latitude/logitude of position:

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, minimum-scale=1, i
nitial-scale=1, user-scalable=no">
    <title></title>
    <script type="text/javascript"
       src="http://maps.google.com/maps/api/js?sensor=true&v=3&libraries=
geometry"></script>
</head>


<body>

    <div id="map_canvas" style="width: 600px;height: 500px;"></div>
    <br />
    <label for="latitude">Latitude (緯度):</label>
    <input id="latitude" type="text" value="" />
    <label for="longitude">Longitude (經度):</label>
    <input id="longitude" type="text" value="" />

    <br>
    <!--
    <label>
    The distance from Chang-Gung University to Destination: <div id="dist
anceAB"></div>
    <label>
    -->
<script type="text/javascript">
   var myZoom = 16;
   var myMarkerIsDraggable = true;
   var myCoordsLenght = 12;
   var defaultLat = 25.01743594338596;
   var defaultLng = 121.540956;
   var loc1 = new google.maps.LatLng(25.033515487308918, 121.386525133759
);
   var loc2 = new google.maps.LatLng(25.033515487308918, 121.386525133759
);

   function initialize() {
      var mapOptions = {
          center: new google.maps.LatLng(defaultLat,defaultLng),
          zoom: myZoom,
          mapTypeId: google.maps.MapTypeId.ROADP
      };
      var map = new google.maps.Map(document.getElementById("map_canvas")
,mapOptions);
```

```
        // creates a draggable marker to the given coords
        var myMarker = new google.maps.Marker({
            position: new google.maps.LatLng(defaultLat, defaultLng),
            draggable: myMarkerIsDraggable
        });


        google.maps.event.addListener(myMarker, 'dragend', function(evt){
          var newLat=evt.latLng.lat();
          var newLng=evt.latLng.lng().toFixed(myCoordsLenght);
          document.getElementById('latitude').value = newLat;
          document.getElementById('longitude').value = newLng;
          var loc2 = new google.maps.LatLng(newLat, newLng);
        });

        // centers the map on markers coords
        map.setCenter(myMarker.position);

        // adds the marker on the map
        myMarker.setMap(map);
    }
    google.maps.event.addDomListener(window, 'load', initialize);
</script>

    </body>
    </html>
```

```
How to Create Create Panorama Maually
---
- Got an image;
  - Have better to take the image in HDR mode;
  - From my Asus padfone, took a snapshoot, connect with Mac botebook pro; draw
it from <font color="blue">/HOME/PHONE_MEMORY/DCIM/Camera</font> to Mac;
- draw  the image into certain folder in Project Windows and rename it if
possible;
- In [Inspector] Windows,  set the option of [Texture shape] as  <font
color="brown">cube</font>,  [mapping] as <font color="brown">Six Frames Layout
(Cubic Environment)</font> or  <font color="brown">Lattitude-Longtinude
Layout(Cylindricl)</font>; then  click [Apply] on the bottom to convert it. Test
the result by pulling the output on the bottom.
- in the same folder of image placed, create a material by right click the mouse
botton;
  - in [Inspector] Windows, set the [Shader] option as <font color="brown">
[skybox -&gt;Cubemap]</font>;
  - in [Cubemap (HDR], select the image we jjust converted;
- Replace the [Main Camera] Skybox material.
Now try the panorama image you had made.
```

In [ ]:

In [1]:

```
!jupyter nbconvert Unity-Google-Street-View-2.ipynb
```

```
[NbConvertApp] Converting notebook Unity-Google-Street-View-2.ipynb
to html
[NbConvertApp] Writing 265613 bytes to Unity-Google-Street-View-2.ht
ml
```

In [ ]: