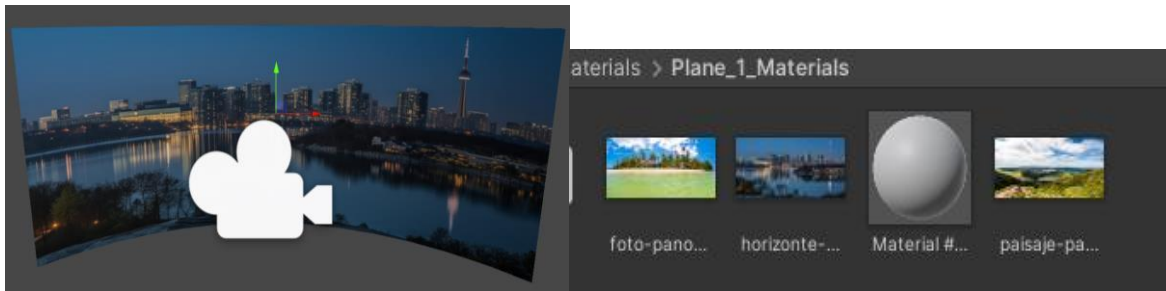


Carrusel VR

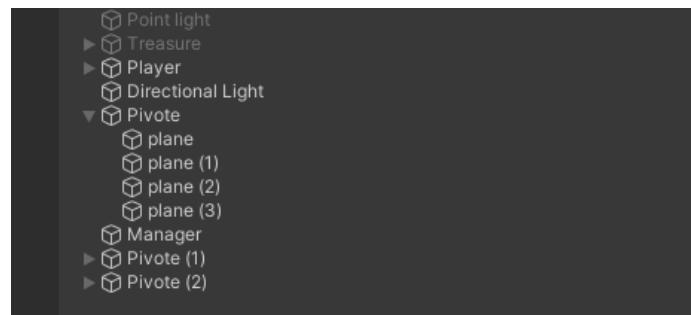
Se creó con ayuda de 3ds Max un plano al cuál se le agregó una textura y se deformó con el modificador blend para que tuviera una curvatura. De esta forma se exportó como fbx para ser utilizado posteriormente en Unity.

En Unity se importo el plano y se modificó el tamaño, también se extrajeron las texturas y los materiales.

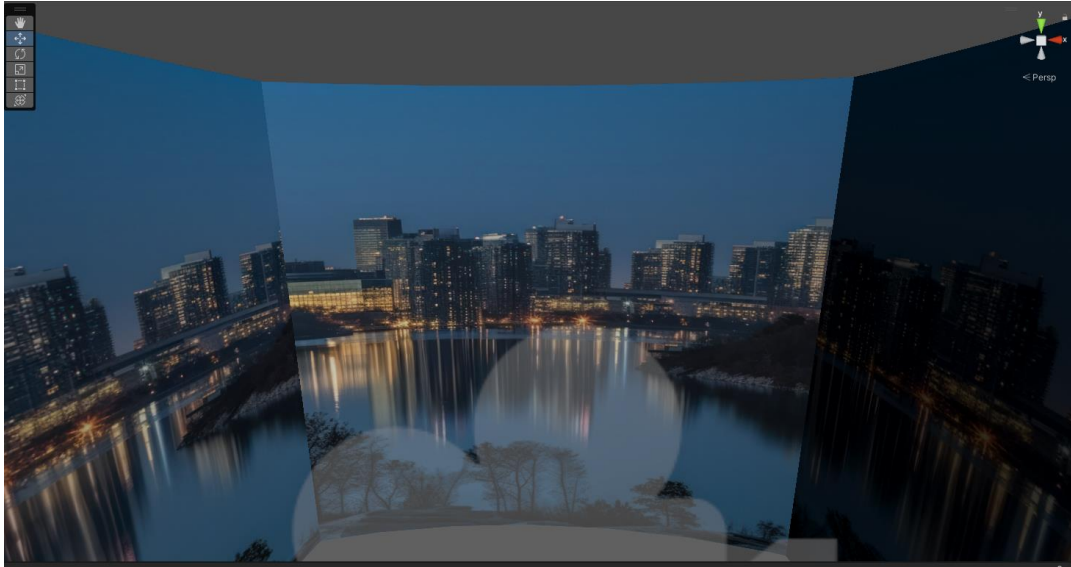


Para crear planos con fondos distintos simplemente se descargaron más imágenes, se agregaron en la carpeta donde estaba la textura base y simplemente arrastrándolo a la copia del plano se cambia la textura.

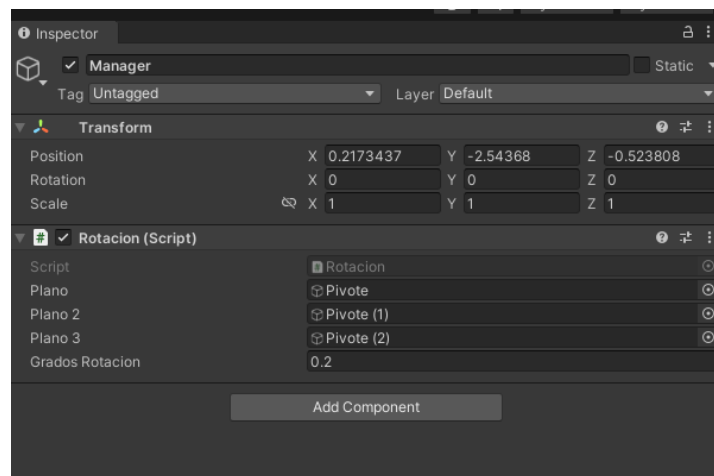
Una vez que se tienen los distintos planos para armar el carrusel se crearon gameObjects vacíos con el propósito de actuar como pivote de giro. Se deben reiniciar los transforms para que todos los atributos sean inicializados en cero.



Cada plano con imagen distinta se debe copiar cuatro veces para el carrusel, entonces cada carrusel se compone de cuatro planos, una vez que se tienen los planos en posición alrededor de la cámara principal se ajustan los demás carruseles para que queden en la misma posición que el carrusel inicial.



Se creó un gameObject llamado manager que, como indica su nombre es el que se encargará de la rotación de los distintos carruseles, así como de irlos alternando. Para esto se creó un script que recibirá los tres carruseles, tendrá un contador y una variable que administra que carruseles activar y desactivar.



El giro se controla con un Rotate en el plano que se encuentra activo y los grados de rotación en el update se controlan con una variable llamada gradosRotacion.

Cada determinado tiempo (10 segundos para este caso) se revisará la variable state para saber que carrusel activar o desactivar.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Rotacion : MonoBehaviour
6  {
7      public GameObject plano, plano2, plano3;
8      private Transform plano_transform;
9      private float changeSceneCD;
10     private int state = 0;
11     public float gradosRotacion = 1;
12     // Start is called before the first frame update
13     void Start()
14     {
15         plano_transform = plano.transform;
16         changeSceneCD = 0;
17         plano2.SetActive(false);
18         plano3.SetActive(false);
19     }
20
21     // Update is called once per frame
22     void Update()
23     {
24         plano_transform.Rotate(new Vector3(0, gradosRotacion, 0));
25         CambiaPlano();
26         Debug.Log(Time.time);
27         Debug.Log(changeSceneCD);
28         Debug.Log(state);
29     }
30
31     public void CambiaPlano()
32     {
33         if (changeSceneCD < Time.time && state == 0)
34         {
35             plano.SetActive(false);
36             plano2.SetActive(true);
37             state = 1;
38             plano_transform = plano2.transform;
39             changeSceneCD = Time.time + 10.0f;
40         }
41         else
42         {
43             if (changeSceneCD < Time.time && state == 1)
44             {
45                 plano2.SetActive(false);
46                 plano3.SetActive(true);
47                 state = 2;
48                 plano_transform = plano3.transform;
49                 changeSceneCD = Time.time + 10.0f;
50             }
51             else
52             {
53                 if (changeSceneCD < Time.time && state == 2)
54                 {
55                     plano3.SetActive(false);
56                     plano.SetActive(true);
57                     state = 0;
58                     plano_transform = plano.transform;
59                     changeSceneCD = Time.time + 10.0f;
60                 }
61             }
62         }
63     }
64 }
65

```

