

HDRI - Skybox

José Luis Carreño Arteaga
jcarreno53@yahoo.com.mx

Materiales y Recursos utilizados

- Los materiales y recursos aquí presentados son propiedad intelectual de sus autores. Presentamos estos para apoyar a la educación y sin fines de lucro.
- Los alumnos deberán construir sus propios materiales y recursos que necesiten.



HDRI de estudio

https://polyhaven.com/a/arboretum

Arboretum HDRI • Poly Haven

polyhaven.com/a/arboretum

Poly Haven HDRIs / **Arboretum** Assets Add-on Vaults Gallery Support Us About/Contact

4K EXR Download 29.56 MB

Author: **Sergej Majboroda**

CC0 License 16 EVs

Downloads: 28346

Sponsored by: **Bernd Schmidt**

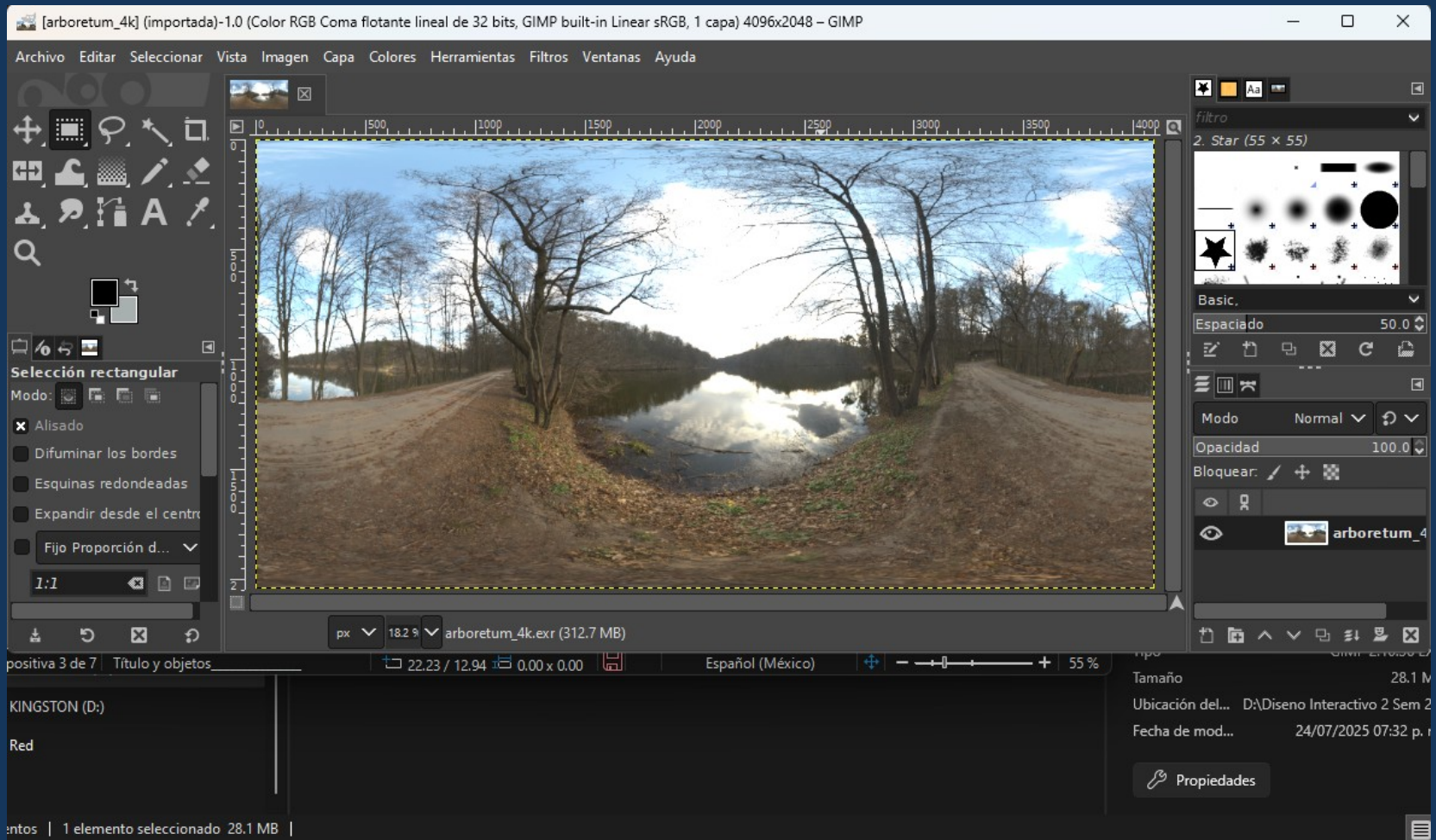
Categories: outdoor nature sunrise/sunset

Tags: lake water path dam bridge tree

Released: 5 years ago

Arobretum del artista Sergej Majboroda es un proyecto 4K

4096x2048



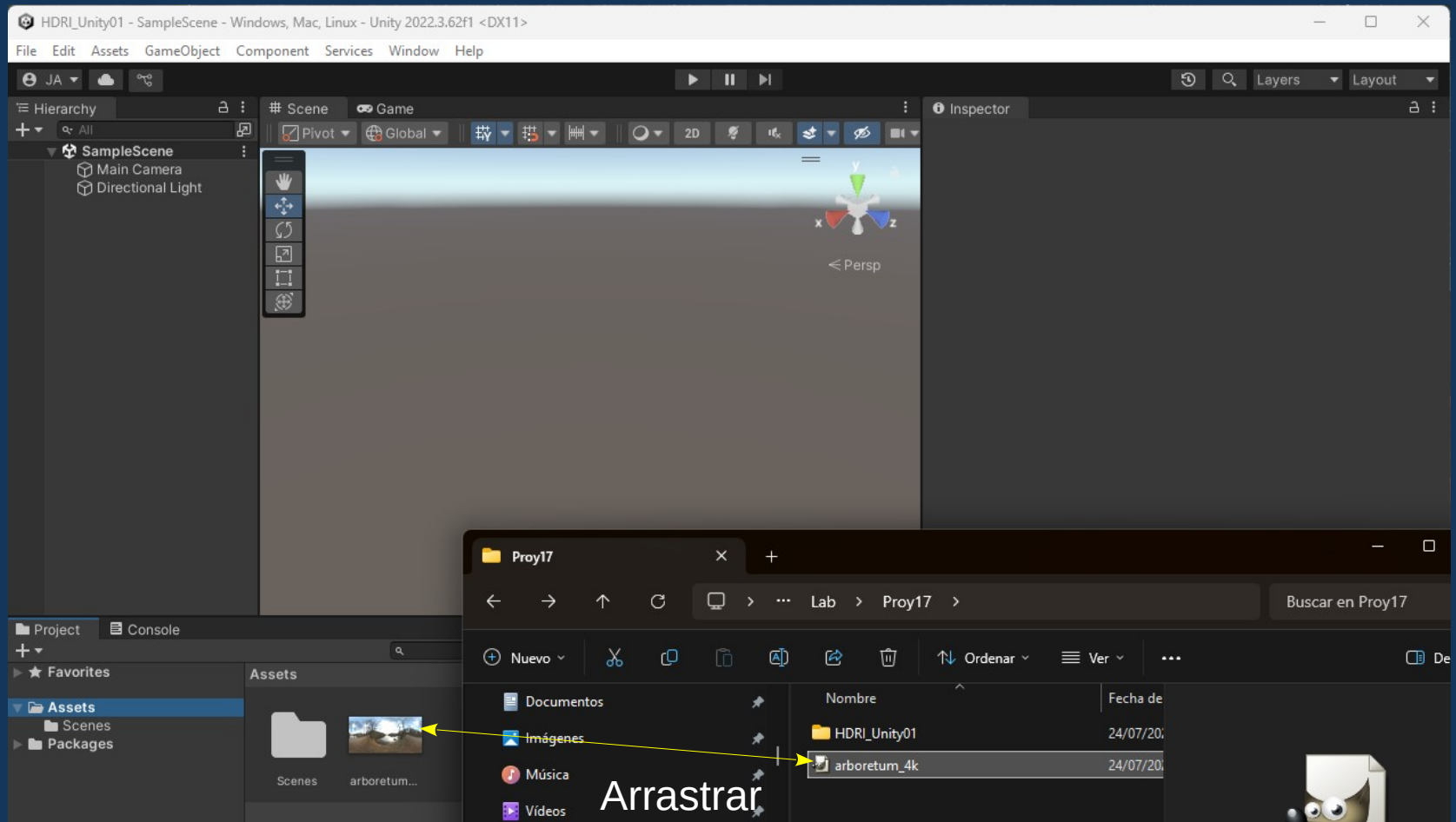
Paso1 Construcción de un nuevo proyecto en Unity

- Lo llamaré HDRI_Unity01



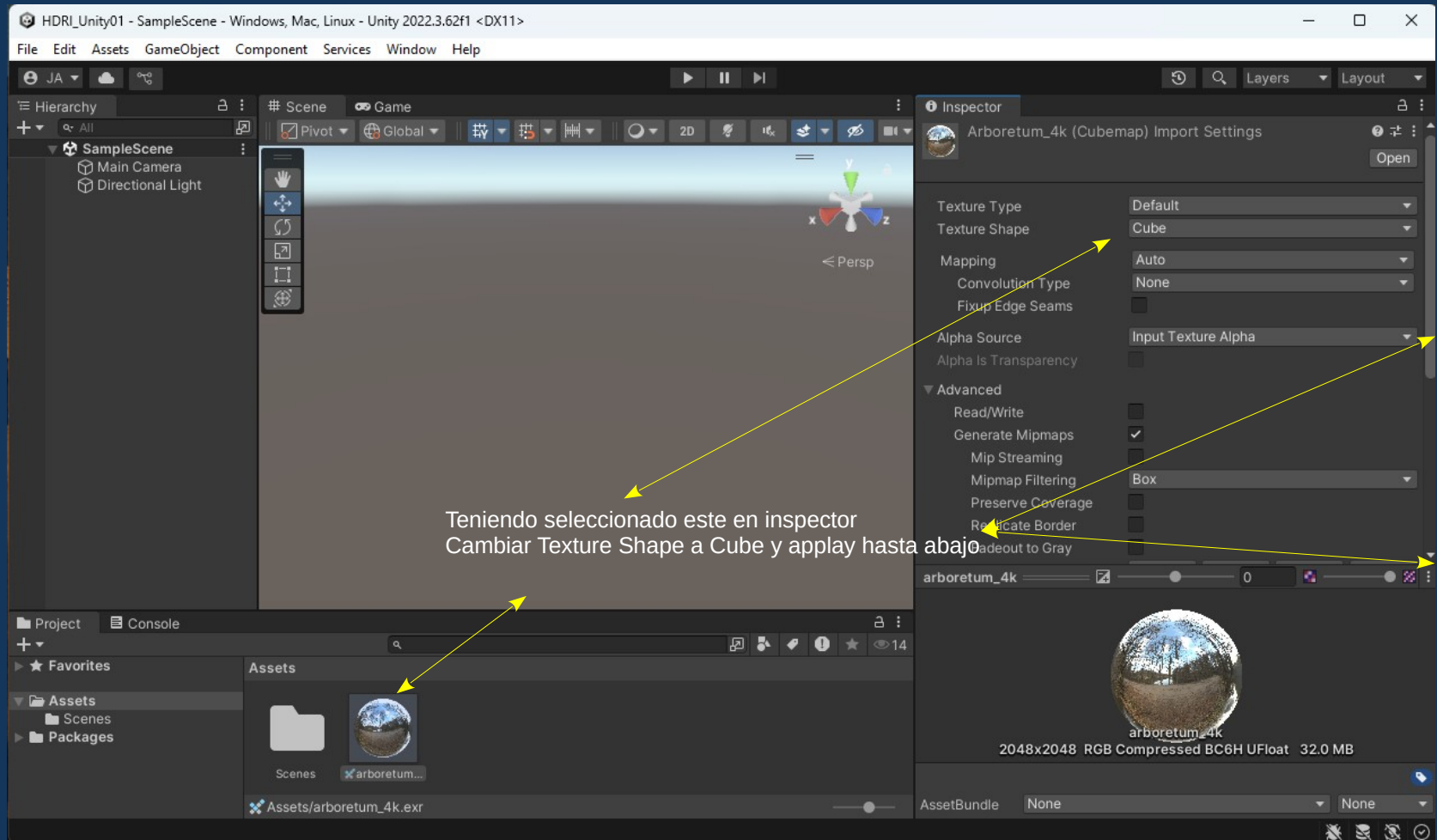
Colocar el HDRI en Assets

● Colocar



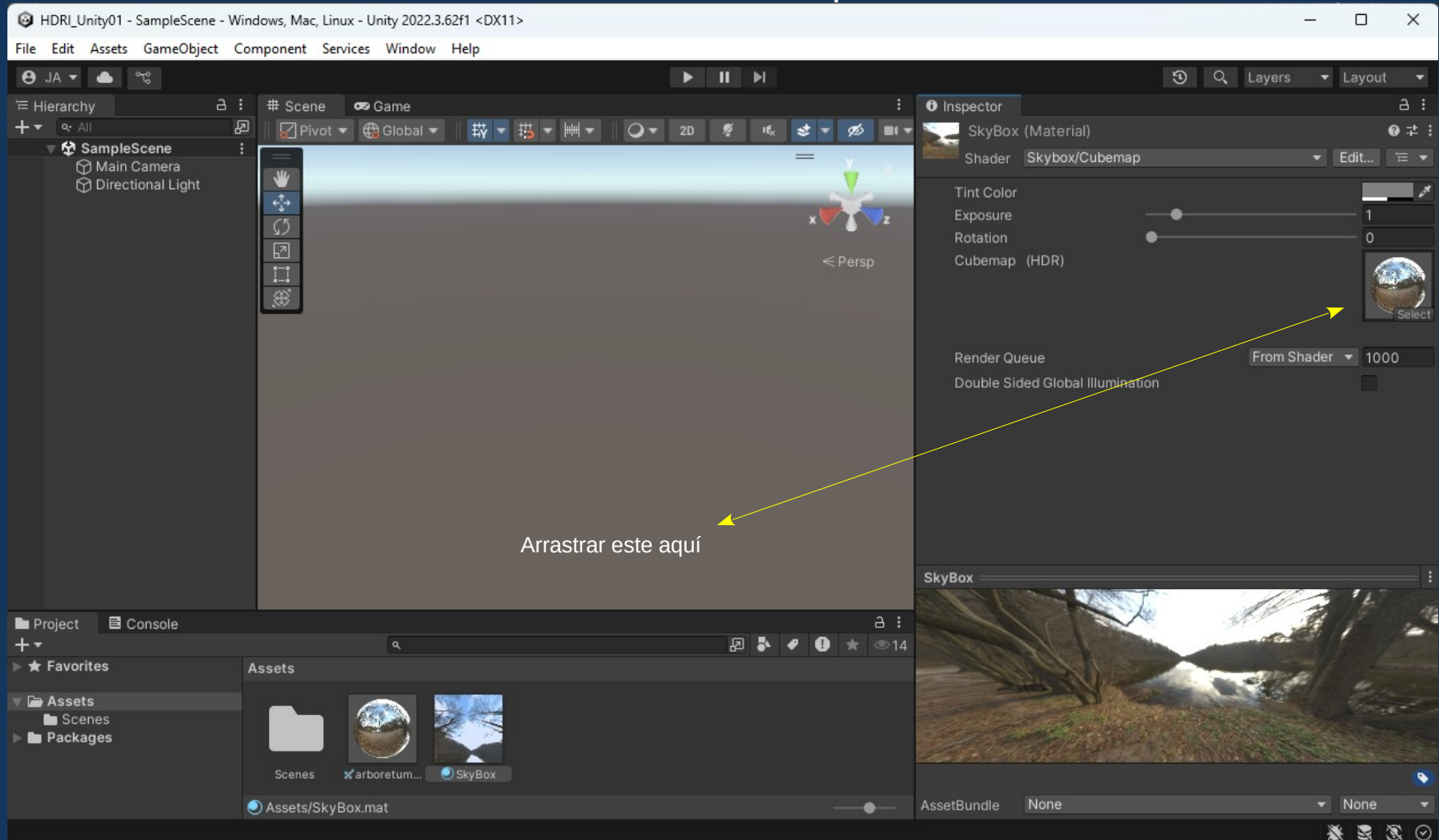
Cambiando el Texture Shape a cube

- Teniendo seleccionado el HDRI, en Inspector cambiar el Texture Shape a cube y hasta abajo “applay” (desplazar la barra hasta abajo)



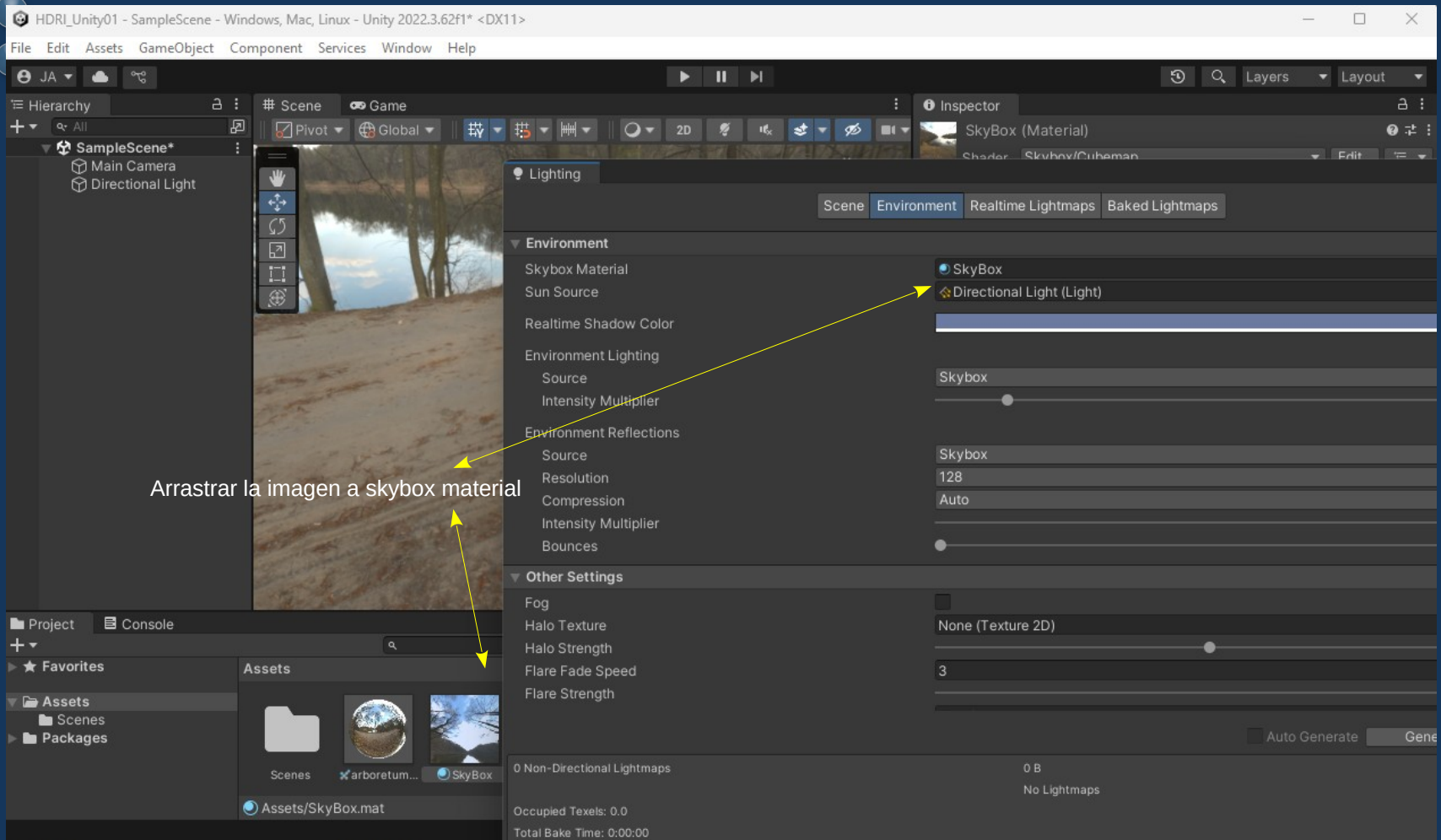
Crear material skybox

- En el material creado, en shader seleccionar “Skybox” y luego “Cubemap”
- Arrastrar el HDRI modificado a Cube Map HDR



Cambiando el Rendering

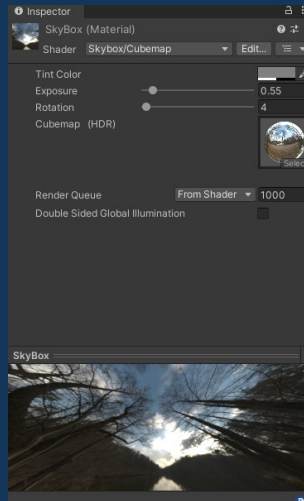
- En el menu en “Window” → “Rendering” → Lighting en la pestaña Environment arrastrar a “Skybox Material” el material antes generado
- Arrastrar el material a “skybox Material”



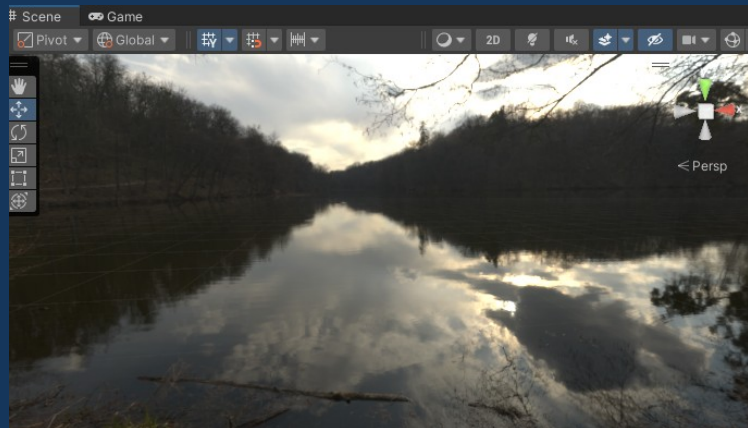
Arrastrar la imagen a skybox material

Recordando algo de Unity en el material.

- Seleccionando el material skybox que creamos, podemos usar “Exposure” o “Rotation”

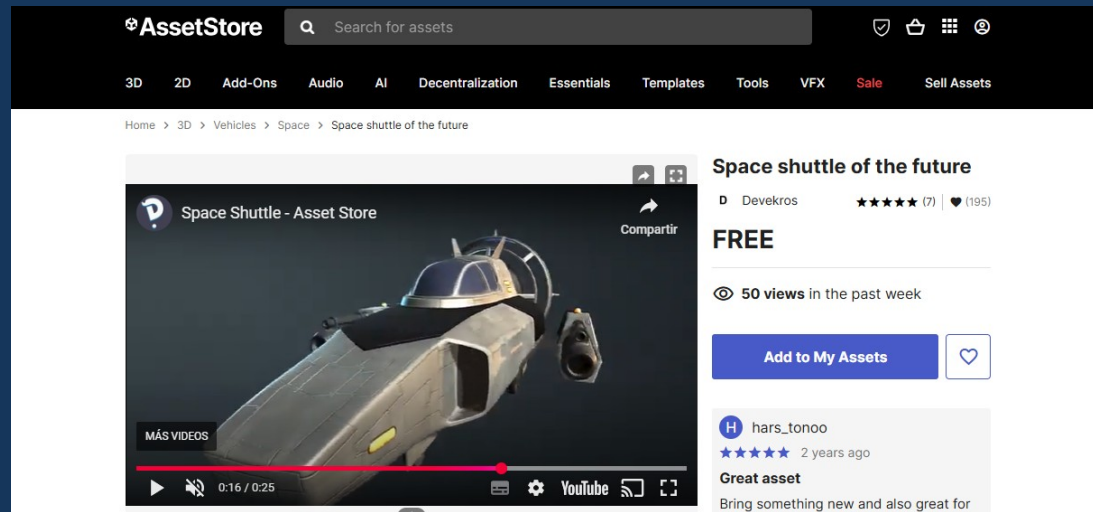


- O en Scene también podemos rotar y manipular la escena con el botón derecho del mouse



Asset agregado.

Asset

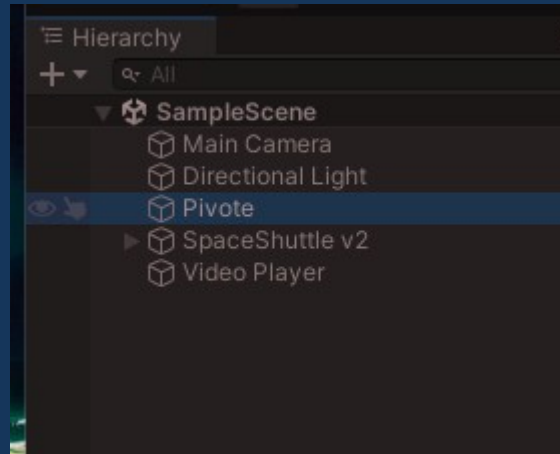


En escena

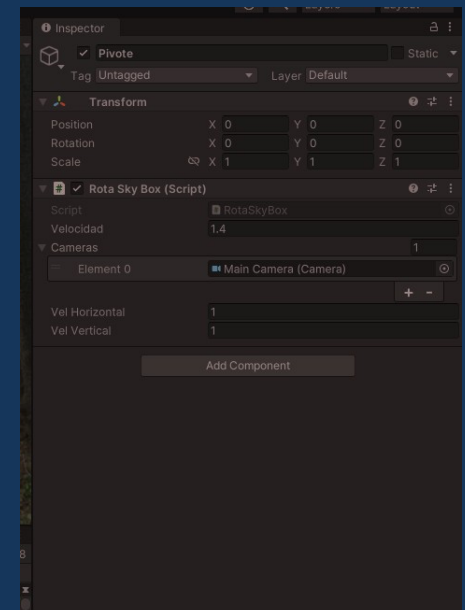


Agregando objeto vacío

- Agregar un objeto vacío quedando la jerarquía de los objetos

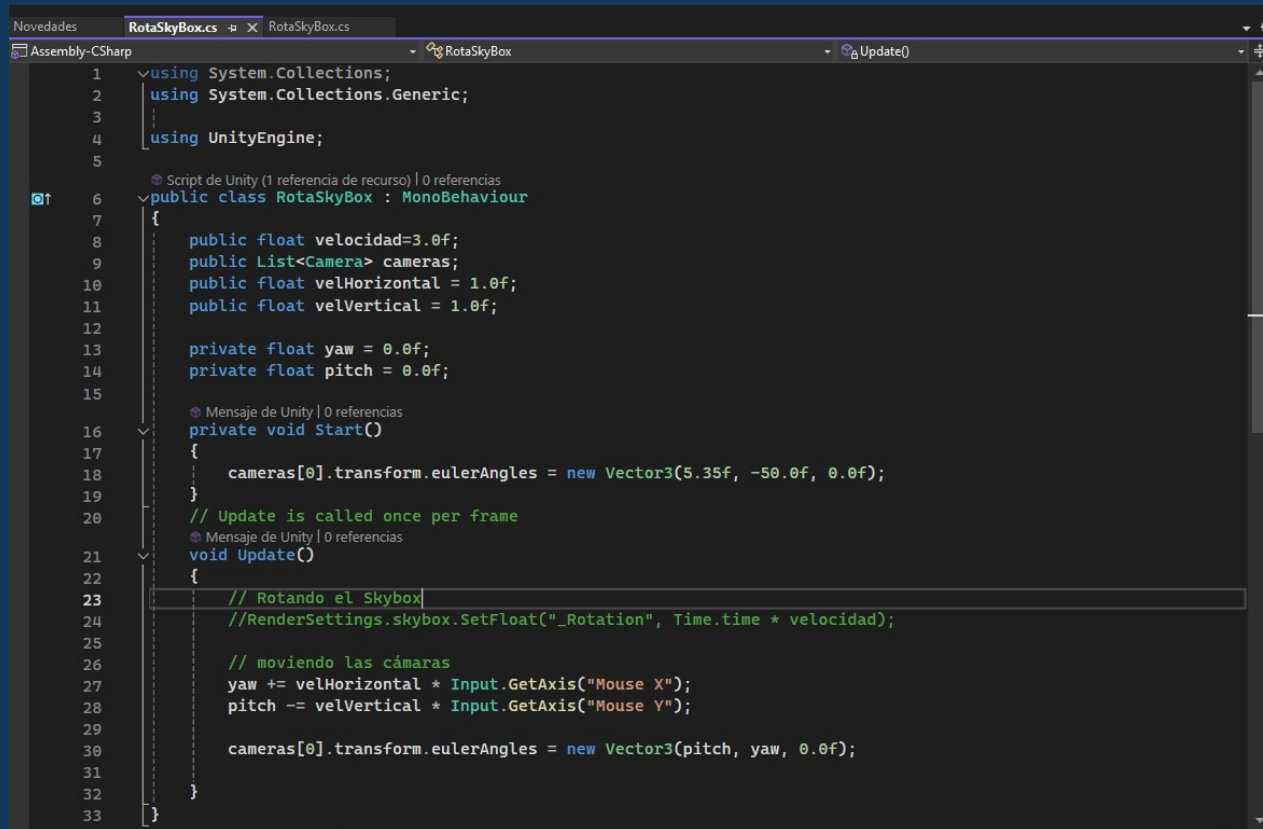


- Pivote es el objeto vacío. Colocar el script RotaSkyBox



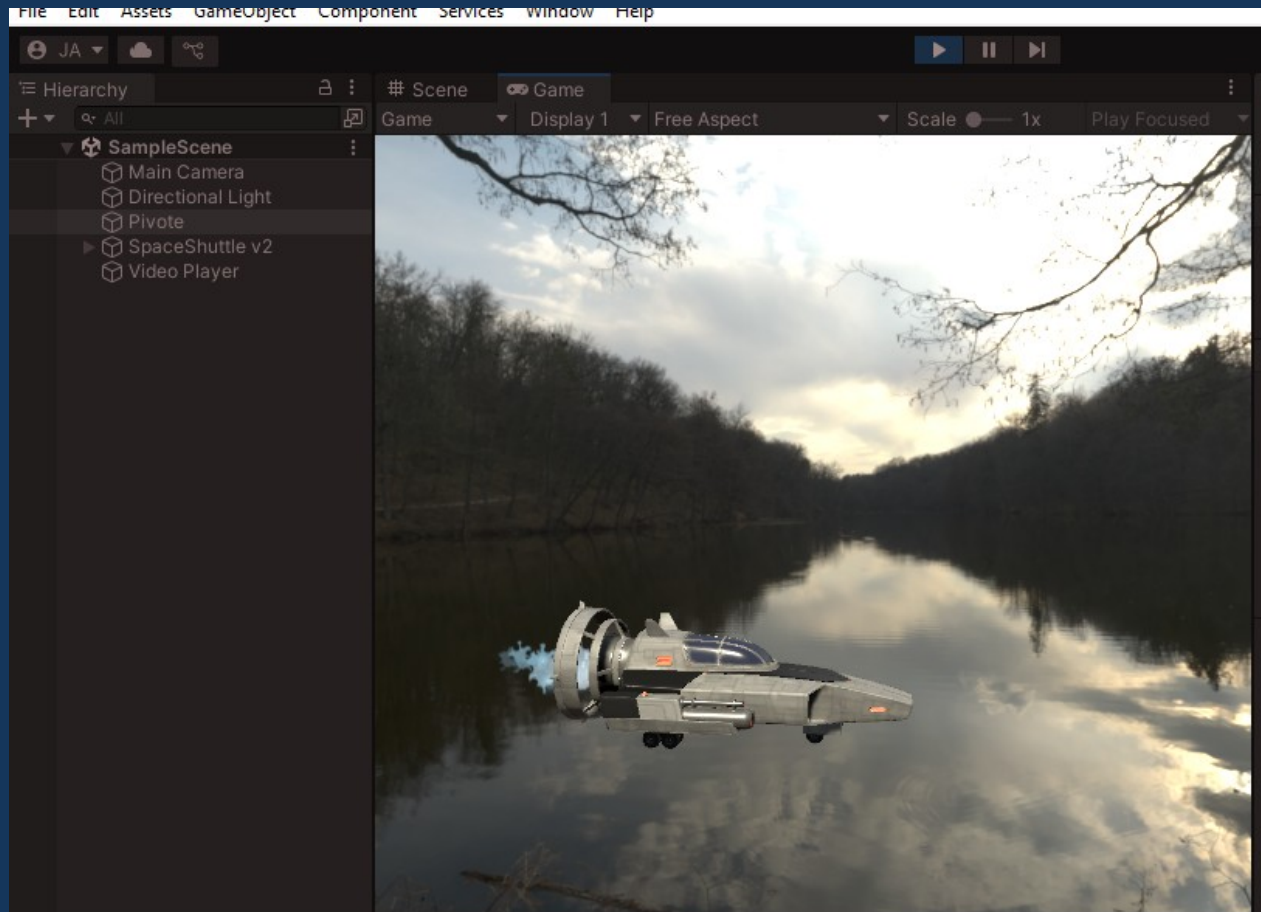
Script RotaSkyBox → Controlar con el mouse el movimiento de la cámara.

- Agregando un poco de más código se puede controlar varias cámaras



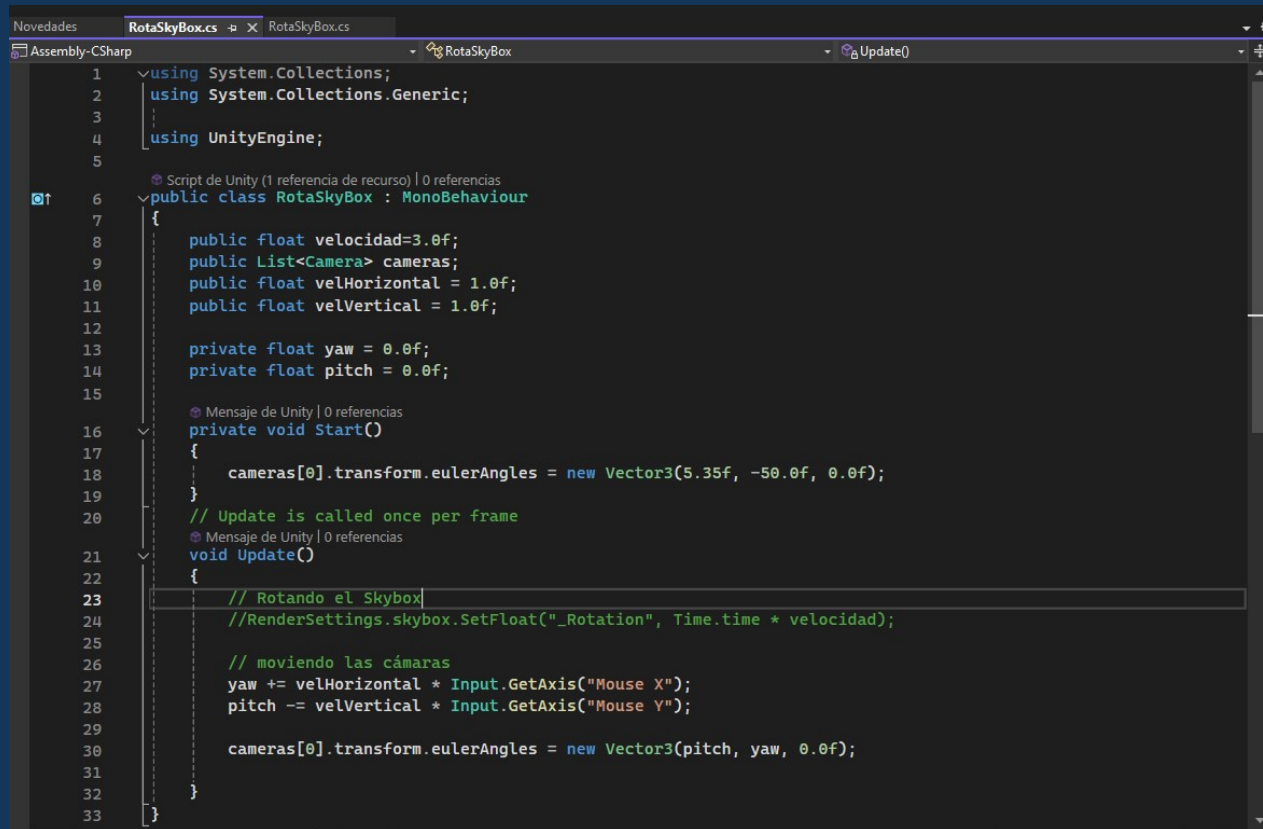
```
1 using System.Collections;
2 using System.Collections.Generic;
3
4 using UnityEngine;
5
6 public class RotaSkyBox : MonoBehaviour
7 {
8     public float velocidad=3.0f;
9     public List<Camera> cameras;
10    public float velHorizontal = 1.0f;
11    public float velVertical = 1.0f;
12
13    private float yaw = 0.0f;
14    private float pitch = 0.0f;
15
16    private void Start()
17    {
18        cameras[0].transform.eulerAngles = new Vector3(5.35f, -50.0f, 0.0f);
19    }
20    // Update is called once per frame
21    void Update()
22    {
23        // Rotando el Skybox
24        //RenderSettings.skybox.SetFloat("_Rotation", Time.time * velocidad);
25
26        // moviendo las cámaras
27        yaw += velHorizontal * Input.GetAxis("Mouse X");
28        pitch -= velVertical * Input.GetAxis("Mouse Y");
29
30        cameras[0].transform.eulerAngles = new Vector3(pitch, yaw, 0.0f);
31    }
32 }
33
```

● Ejecutando la escena



Script RotaSkyBox → Controlar con el mouse el movimiento de la cámara.

- Agregando un poco de más código se puede controlar varias cámaras



```
1 using System.Collections;
2 using System.Collections.Generic;
3
4 using UnityEngine;
5
6 public class RotaSkyBox : MonoBehaviour
7 {
8     public float velocidad=3.0f;
9     public List<Camera> cameras;
10    public float velHorizontal = 1.0f;
11    public float velVertical = 1.0f;
12
13    private float yaw = 0.0f;
14    private float pitch = 0.0f;
15
16    private void Start()
17    {
18        cameras[0].transform.eulerAngles = new Vector3(5.35f, -50.0f, 0.0f);
19    }
20    // Update is called once per frame
21    void Update()
22    {
23        // Rotando el Skybox
24        //RenderSettings.skybox.SetFloat("_Rotation", Time.time * velocidad);
25
26        // moviendo las cámaras
27        yaw += velHorizontal * Input.GetAxis("Mouse X");
28        pitch -= velVertical * Input.GetAxis("Mouse Y");
29
30        cameras[0].transform.eulerAngles = new Vector3(pitch, yaw, 0.0f);
31    }
32 }
33
```

Referencias Bibliográficas