

# UNITY



unity

[Unity](#)

[Servicios](#)

[Exhibición](#)

[Conocer](#)

[Comunidad](#)

[Obtener Unity](#)

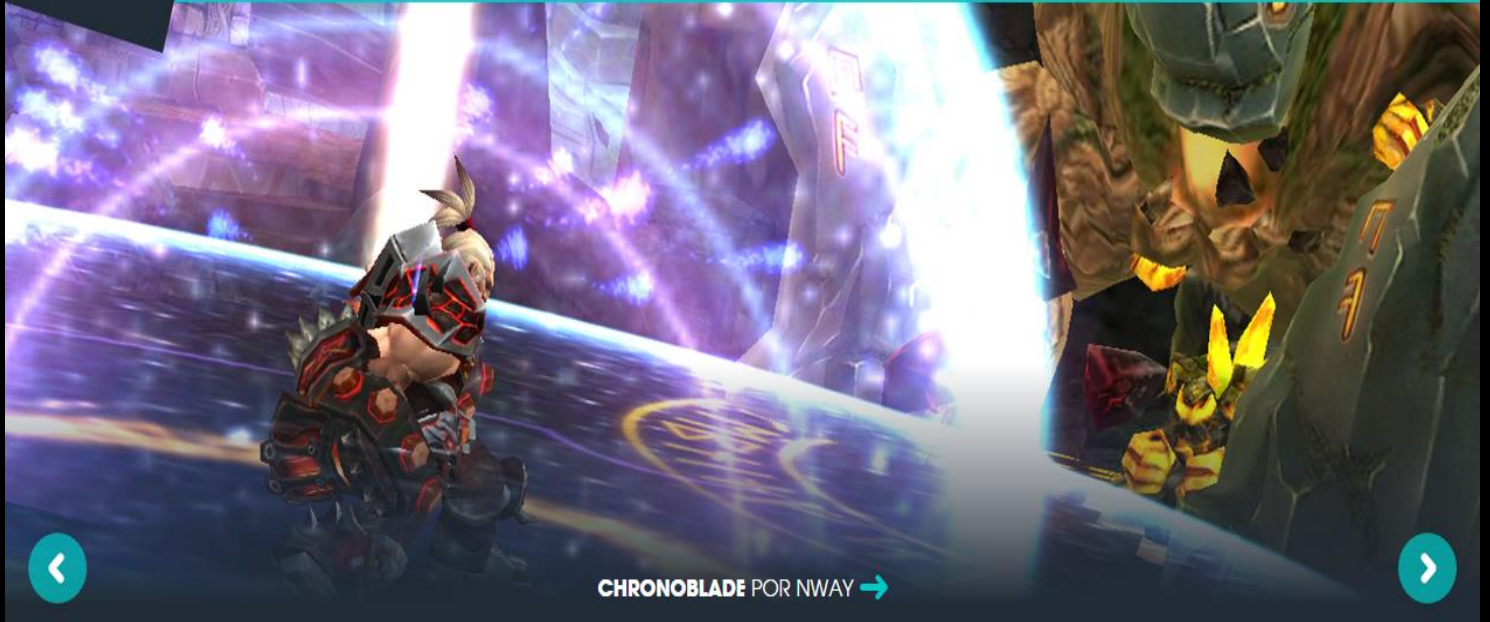
[Asset Store](#)



[Galería](#)

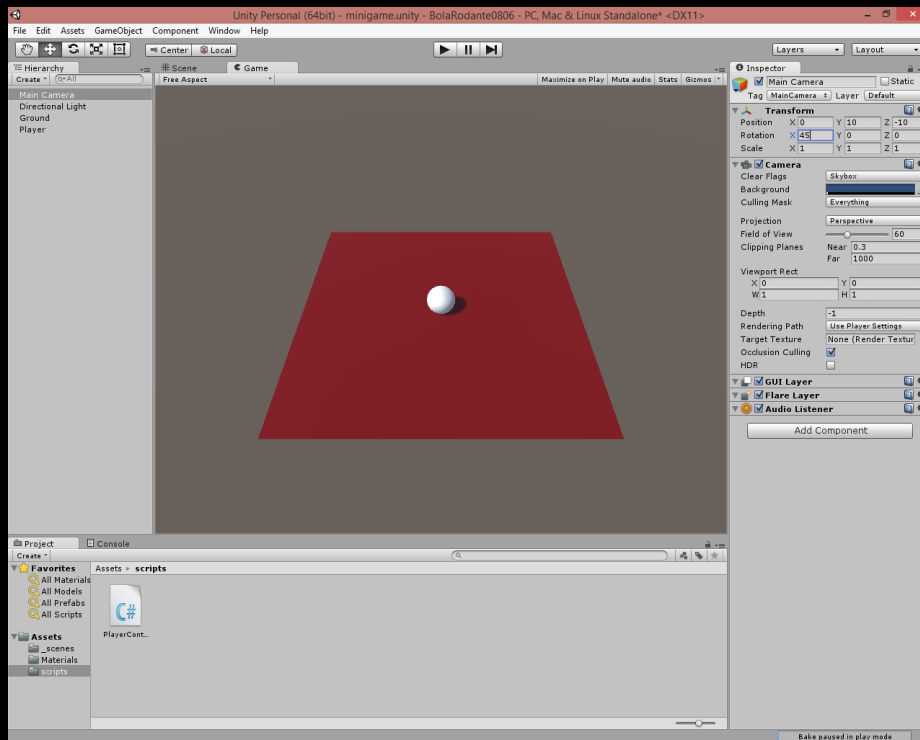
[Game Profiles](#)

[Tráilers](#)

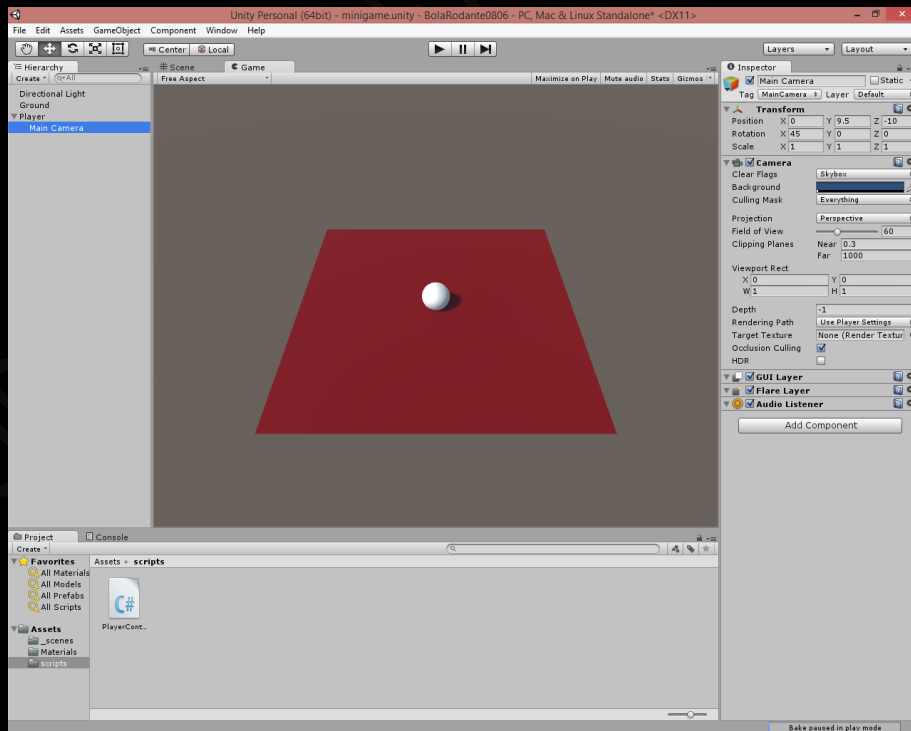


CHRONOBLADE POR NWAY →

# PART III: configurar la cámara



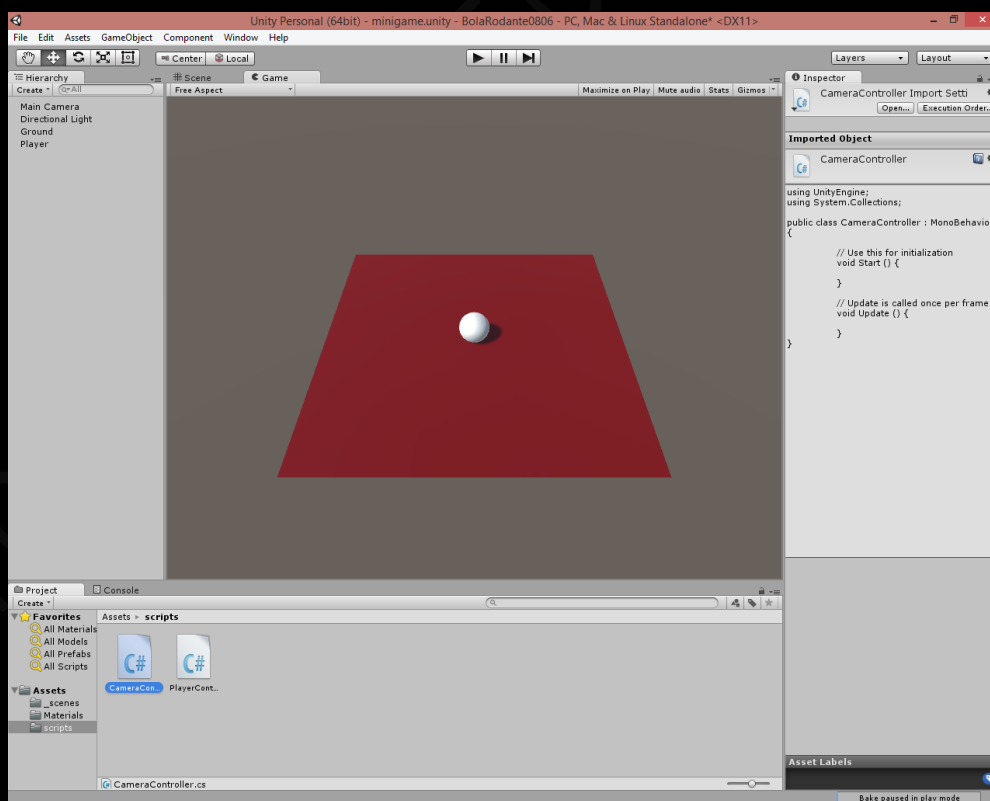
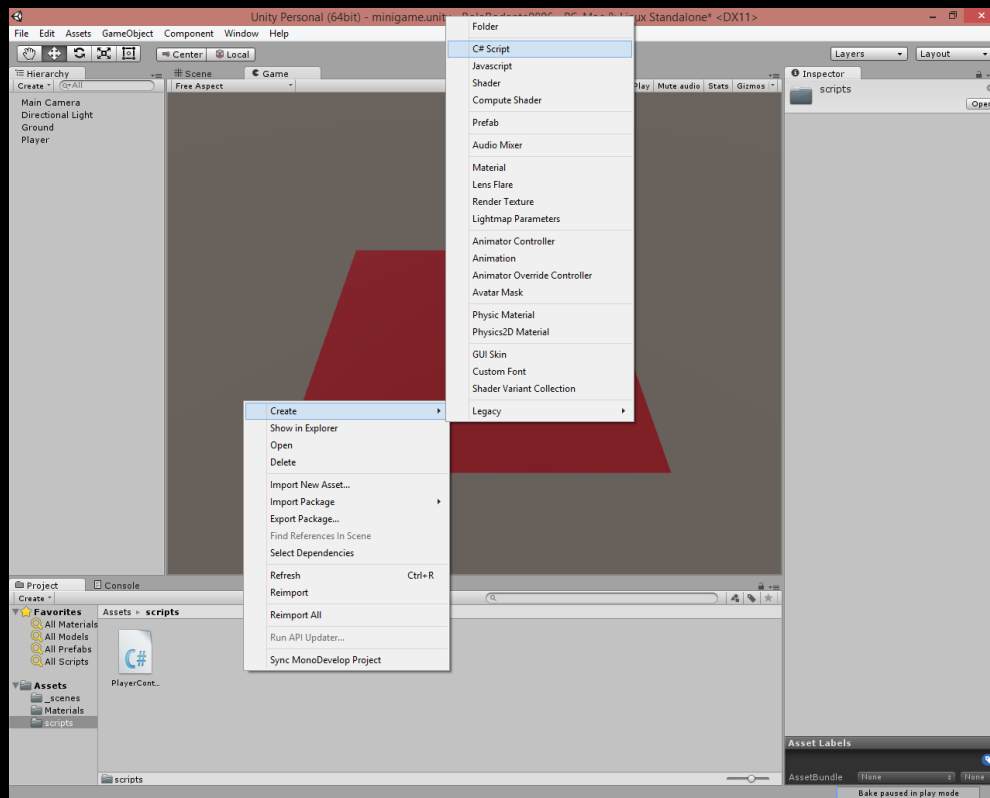
Definir la cámara como “hijo” del jugador



Probar la cámara en el juego: Modo Game

La cámara está asociada al jugador (en este caso es player el padre), como en un juego de tercera persona, en este caso no nos sirve

# Tenemos que generar un script



Vamos a necesitar dos variables:

Una variable Public de tipo `GameObject` referenciada al objeto `Player`

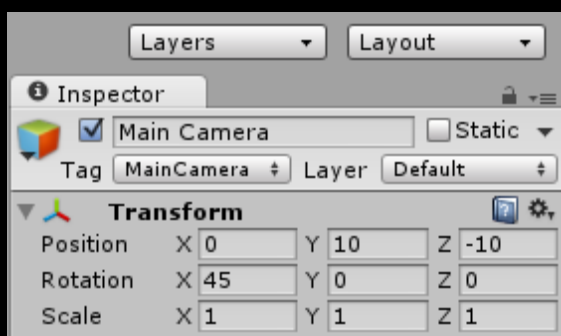
Una variable Private de tipo `Vector3` llamada `offset`

Definamos el valor del offset

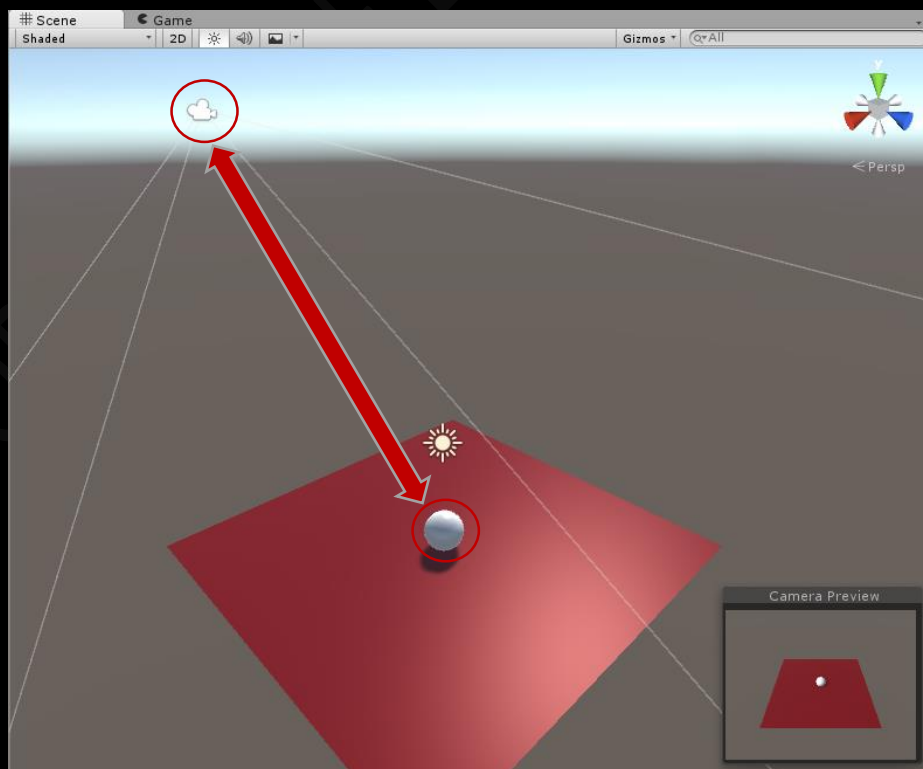
Transform Posición de la cámara

~

Transform Posición del player



para encontrar la diferencia entre los dos



```
CameraController.cs
CameraController ▶ Update ()
1 using UnityEngine;
2 using System.Collections;
3
4 public class CameraController : MonoBehaviour {
5
6     public GameObject player;
7
8     private Vector3 offset;
9
10    // Use this for initialization
11    void Start () {
12        offset = transform.position - player.transform.position;
13    }
14
15    // Update is called once per frame
16    void Update () {
17        transform.position = player.transform.position + offset;
18    }
19 }
20
```

De esta manera cuando nos movamos con los controles del teclado por ejemplo en cada frame, antes que la camera vea lo que está pasando, la movemos a una nueva posición alineada con el objeto Player. Como si fuese un “hijo” del objeto player, pero evitando que ruede con él.

En vez de usar Update (que sí bien se ejecuta en cada frame y permite ubicar al objeto y a la cámara) para seguimiento de cámara, seguir los últimos estados conocidos usamos LateUpdate.

LateUpdate se ejecuta en cada frame como Update, pero garantiza ejecutarse después que todos los ítems fueron procesados con Update.

Cuando seteamos la posición de la cámara sabemos que el jugador se movió a ese frame

```
CameraController.cs
CameraController ▶ LateUpdate ()
1 using UnityEngine;
2 using System.Collections;
3
4 public class CameraController : MonoBehaviour {
5
6     public GameObject player;
7
8     private Vector3 offset;
9
10    // Use this for initialization
11    void Start () {
12        offset = transform.position - player.transform.position;
13    }
14
15    // Update is called once per frame
16    void LateUpdate () {
17        transform.position = player.transform.position + offset;
18    }
19 }
20
```

Agreguemos el script a la cámara



Tenemos que asignar el objeto **Player** a la variable pública del script

