

PRÀCTICA 2: "CITY"

REQUISITS DE LA PRÀCTICA

Realitzar un videojoc des de zero, on el jugador en una càmera de primera persona, ha de recollir totes les monedes possibles pels carrer d'una ciutat on mai s'acaba i evitar ser atropellat pels cotxes que circulin per ella. Cada cop que sigui atropellat perd una vida i en el marge inferior dret s'elimina un dels tres cotxes. La puntuació de les monedes quan les recull és en el marge superior dret. Al cap de 30 segons tornen aparèixer en el joc.

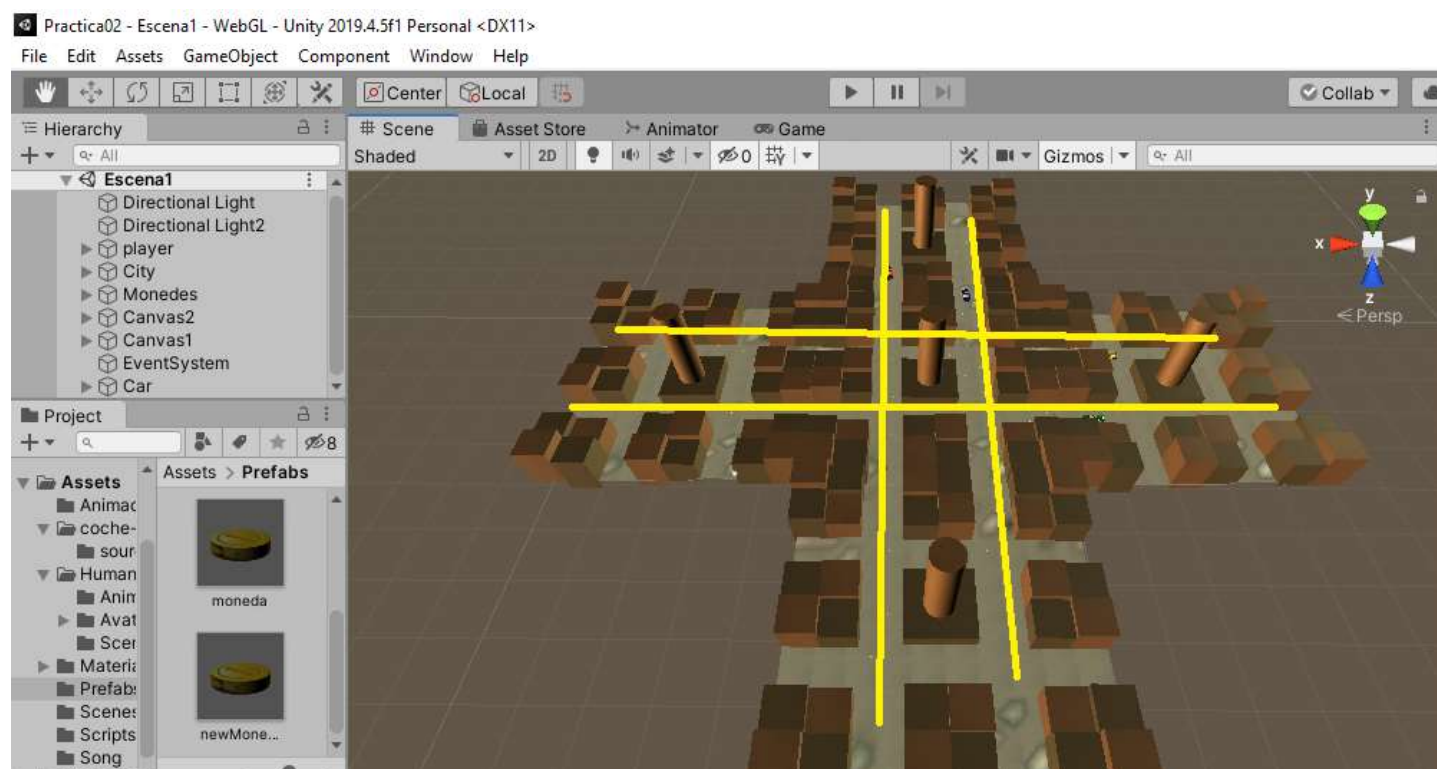


S'ha afegit una segona càmera per jugar en tercera persona, amb possibilitat de canviar de càmera amb la tecla F2, l'indicatiu està en el marge superior esquerra com veiem en la captura de pantalla a dalt.

També s'ha afegit possibilitat de parar la partida amb la tecla P, com els cops que perdem una vida es parar el joc a l'espera d'entrada de la tecla ENTER des de teclat.



Com s'aprecia en les dues il·lustracions s'ha construït quatre cotxes, un per cada carrer principals de la ciutat a baix queden indicats en groc els quatre carrers.



També he fet un *Prefab*, de la moneda amb la seva rotació amb *animator*, tenim 8 monedes per cada carrer principal.

FITXA TÈCNICA

Tenim 4 *gameObject* principals i són els següents:

1. **City**: s'ha construït a partir d'un terreny i quatre més als seus costats, cada terreny consta de 9 illes d'edificis distribuïdes equidistants entre elles elaborats amb dos *Prefab* diferents. El paviment he emprat un material amb una textura de ciments.
2. **Player**: els seus components són *Character Collider* i el script principal **PlayerMove.cs**, on tenim el moviment del personatge, la lògica del vídeo joc, la coalició amb el cotxes com les dues funcions per activar la seva càmera corresponent amb un *canvas* específic per ella. Aquest es compon de quatre *gameObject* fills el personatge un fitxer FBX, les dues càmeres i l'últim el *GroundCheck*, encarregat de detectar la coalició del *Character Collider* amb el terreny.

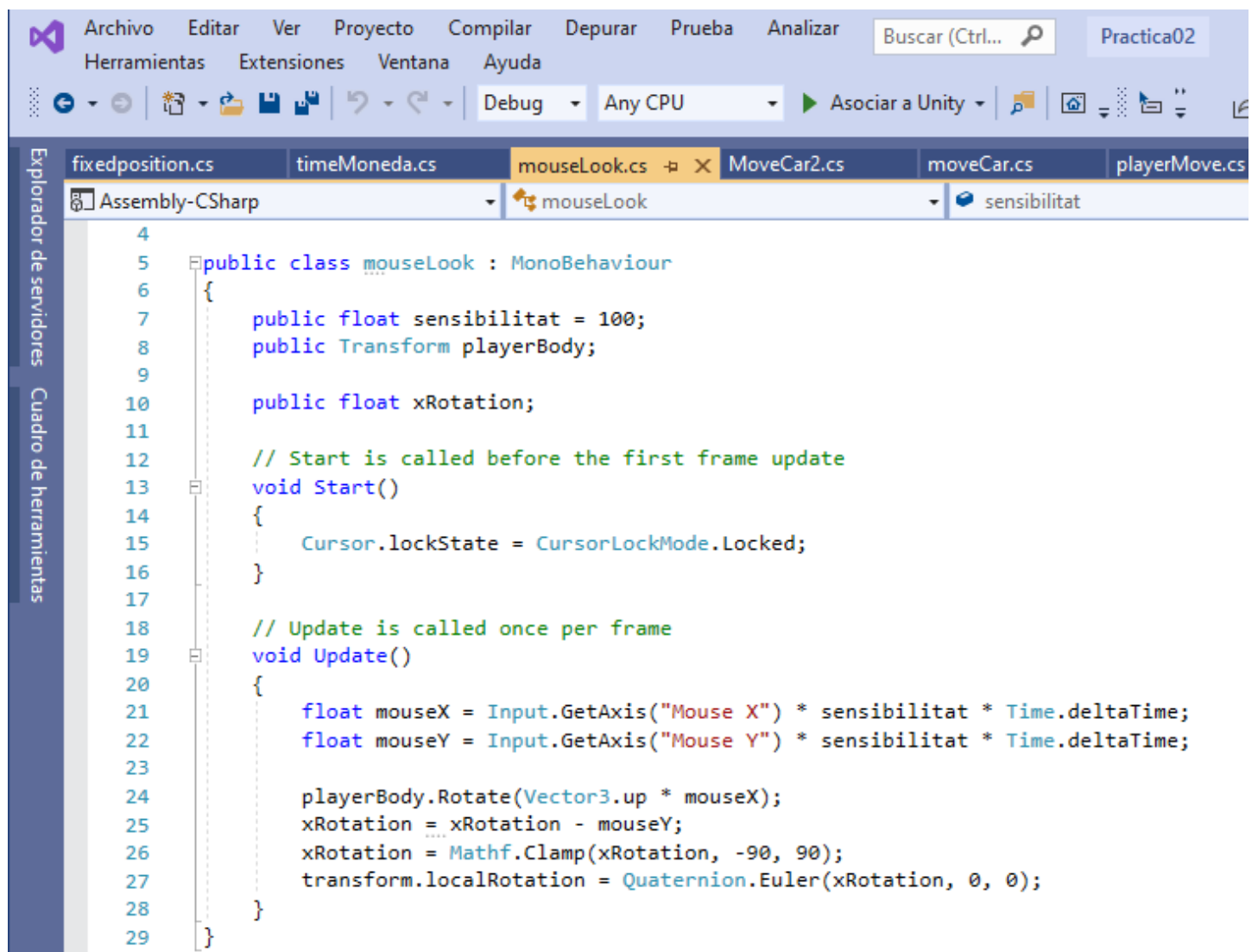


3. **Monedes:** compost de 24 *gameObject* fills que són Prefabs de la moneda, aquesta amb un script secundari anomenat **timeMoneda.cs**, on ens detectarà la coalició amb el Player i la seva posterior conta de 30 segons per tornar-se a visualitzar.

```
15 void Update()
16 {
17     if (GetComponent<BoxCollider>().enabled == false)
18     {
19         time += Time.deltaTime;
20         if (time >= 3 && time <= 4)
21             GetComponentInChildren<AudioSource>().enabled = false;
22         if (time >= 30)
23         {
24             GetComponent<MeshRenderer>().enabled = true;
25             GetComponent<BoxCollider>().enabled = true;
26             time = 0.0f;
27         }
28     }
29 }
30 private void OnTriggerEnter(Collider other)
31 {
32     if (other.tag == "Player")
33     {
34         movePl.score += 10;
35         GetComponent<MeshRenderer>().enabled = false;
36         GetComponent<BoxCollider>().enabled = false;
37         GetComponentInChildren<AudioSource>().enabled = true;
38         GetComponentInChildren<ParticleSystem>().Play();
39     }
```

4. **Car:** per últim el cotxes són quatre FBX amb tres components cada un; el rigidBody, BoxCollider i dos scrips vinculats els dos cotxes verticals i l'altre dos els horitzontals. Són el **moveCar.cs** i el **moveCar2.cs**.

Per acabar cal anomenar el script per moure les dues càmeres amb el mouse, el **mouseLook.cs**, com s'aprecia en la següent il·lustració és un moviment de 180° en els dos eixos X i Y, a partir de l'entrada dels mètode Input.GetAxis("Mouse X") i l'equivalent per les Y.

The image shows a screenshot of the Unity IDE interface. The top menu bar includes 'Archivo', 'Editar', 'Ver', 'Proyecto', 'Compilar', 'Depurar', 'Prueba', and 'Analizar'. Below the menu is a toolbar with icons for file operations and a search bar labeled 'Buscar (Ctrl...)'. The 'Practica02' tab is active. The 'Explorador de servidores' and 'Cuadro de herramientas' are visible on the left. The 'Herramientas' and 'Extensiones' tabs are also present. The 'Ventana' and 'Ayuda' tabs are at the bottom. The 'Debug' dropdown is set to 'Any CPU'. The 'Asociar a Unity' button is visible. The 'fixedposition.cs', 'timeMoneda.cs', 'mouseLook.cs', 'MoveCar2.cs', 'moveCar.cs', and 'playerMove.cs' files are open in the 'Herramientas' tab. The 'mouseLook.cs' file is selected, and its content is displayed in the main editor. The code is a C# script for a 'MonoBehaviour' class, implementing a mouse look functionality. It includes a 'Start' method to lock the cursor and an 'Update' method to calculate the rotation based on mouse input and a sensitivity factor. The 'sensibilitat' variable is set to 100.0f. The 'playerBody' variable is of type 'Transform'. The 'xRotation' variable is of type 'float'. The 'Update' method uses 'Input.GetAxis' to get the mouse X and Y axis values, multiplies them by the sensitivity and 'Time.deltaTime', and then uses 'Vector3.up' to rotate the 'playerBody' around the Y-axis. The rotation is clamped between -90 and 90 degrees using 'Mathf.Clamp'. The final rotation is converted to a 'Quaternion' using 'Quaternion.Euler'.

```
4
5 public class mouseLook : MonoBehaviour
6 {
7     public float sensibilitat = 100;
8     public Transform playerBody;
9
10    public float xRotation;
11
12    // Start is called before the first frame update
13    void Start()
14    {
15        Cursor.lockState = CursorLockMode.Locked;
16    }
17
18    // Update is called once per frame
19    void Update()
20    {
21        float mouseX = Input.GetAxis("Mouse X") * sensibilitat * Time.deltaTime;
22        float mouseY = Input.GetAxis("Mouse Y") * sensibilitat * Time.deltaTime;
23
24        playerBody.Rotate(Vector3.up * mouseX);
25        xRotation = xRotation - mouseY;
26        xRotation = Mathf.Clamp(xRotation, -90, 90);
27        transform.localRotation = Quaternion.Euler(xRotation, 0, 0);
28    }
29 }
```

REALITZACIÓ

Construcció d'una ciutat en un terreny, amb quatre carrers principals, realitzat amb *Prefab* d'illes d'edificis, la illa central és la representació del centre de la ciutat. Quatre terrenys més clonats amb la mateixa distribució i adjunts al primer per cada costat d'aquest, tan sols serveixen de visualització com una ciutat que no s'acaba, ja que el personatge, farem posicionar-lo quan surti d'un extrem aparegui en el seu extrem oposat.

Construïrem el jugador amb el component *character controller* interactuat amb el teclat del seus moviments i doble salts,

relacionats a aquest moviments s'ha creat una animació amb tres estats diferents repòs, corre i saltar, tots tres enllaçats entre ells.

Construir les monedes en un *Prefab* amb dos scripts un per la rotació sobre ella i l'altre per establir els 30 segons d'invisibilitat en cas de coalició amb el jugador. El disseny de la moneda a partir del cilindre, *gameobject* en 3D per defecte d'*Unity*, modificant les seves dimensions i el material.

Construir una segona càmera on veiem el personatge des d'una posició a distància d'ell, lògica d'interacció de càmeres i els seus *Canves* respectius des de el script principal del joc.

Finalment s'han construït quatre cotxes a partir d'un cotxe amb format FBX textura d'ells materials de la carrosseria groc, blau, verd i vermell. Lògica de coalició del personatges i vides d'ell, amb el cotxe des del script principal.

Enllaç web on hi ha el projecte amb *Unity*, amb el nom de **Practica02.zip** :

https://drive.google.com/drive/folders/1vnAV03N9x0FoKUMTeugfMcEGPhO_3ZpA?usp=sharing

Alumne: **Gilbert Viader Sauret**

Curs: **Disseny de vídeojocs amb Unity**

Barcelona, 5 de març 2021