

HLC\codigosUnity\CONTROLJUGADOR.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class CONTROLJUGADOR : MonoBehaviour
6 {
7     // Variable para almacenar el desplazamiento horizontal del jugador
8     public float desplazamientoHorizontal;
9
10    // Velocidad de movimiento del jugador
11    public float velocidad = 10.0f;
12
13    // Rango máximo de movimiento en el eje X para evitar que el jugador salga de la pantalla
14    public float rangoX = 24.0f;
15
16    // Prefab del proyectil que se disparará al presionar la tecla Espacio
17    public GameObject proyectilPrefab;
18
19    // Prefab del segundo proyectil que se disparará al presionar la tecla Enter
20    public GameObject proyectil2Prefab;
21
22    // Método Start, se ejecuta una vez al iniciar el juego
23    void Start()
24    {
25        // No hay inicialización en este caso
26    }
27
28    // Método Update, se ejecuta una vez por frame
29    void Update()
30    {
31        // Si el jugador presiona la tecla Espacio, se instancia el proyectil 1 en la posición del jugador
32        if (Input.GetKeyDown(KeyCode.Space)){
33            Instantiate(proyectilPrefab, transform.position, proyectilPrefab.transform.rotation);
34        }
35
36        // Si el jugador presiona la tecla Enter, se instancia el proyectil 2 en la posición del jugador
37        if (Input.GetKeyDown(KeyCode.Return)){
38            /*
39            proyectilPrefab → Es el objeto (Prefab) que se quiere instanciar. Debe estar asignado en el Inspector de Unity.
```

```
40     transform.position → Es la posición en la que se generará el nuevo objeto, en este caso, la misma posición del jugador (transform.position).
41     proyectilPrefab.transform.rotation → Se usa la rotación predeterminada del prefab para que el proyectil tenga la orientación correcta.
42     */
43     Instantiate(proyectil2Prefab, transform.position, proyectil2Prefab.transform.rotation);
44 }
45
46 // Evita que el jugador se salga por la izquierda del rango permitido
47 if(transform.position.x < -rangoX){
48     transform.position = new Vector3 (-rangoX, transform.position.y, transform.position.z);
49 }
50
51 // Evita que el jugador se salga por la derecha del rango permitido
52 if (transform.position.x > rangoX) {
53     transform.position = new Vector3 (rangoX, transform.position.y, transform.position.z);
54 }
55
56 // Captura el input del eje horizontal ("A" o "D" en teclado, o joystick en un gamepad)
57 desplazamientoHorizontal = Input.GetAxis("Horizontal"); //Se usa Input.GetAxis("Horizontal")
58                                     // para detectar la entrada del usuario
59                                     // y mover al jugador en el eje X.
60
61 // Mueve el jugador horizontalmente basado en el input, la velocidad y el tiempo transcurrido
62 transform.Translate(Vector3.right * desplazamientoHorizontal * Time.deltaTime * velocidad);
63 /*
64 transform.Translate(...)
65 Este método mueve el objeto en función de un vector de desplazamiento.
66
67 Vector3.right
68 Es un vector unitario ((1, 0, 0)) que representa el eje X positivo en Unity.
69 Mover en Vector3.right desplaza el objeto hacia la derecha y en -Vector3.right hacia la izquierda.
70
71 desplazamientoHorizontal
72 Es un valor que proviene de Input.GetAxis("Horizontal"), que puede ser:
73 -1 cuando se presiona "A" (o la flecha izquierda) → Movimiento a la izquierda.
74 1 cuando se presiona "D" (o la flecha derecha) → Movimiento a la derecha.
75 0 si no hay entrada del jugador.
76
77 Time.deltaTime
78 Representa el tiempo transcurrido desde el último frame.
79 Se usa para que el movimiento sea independiente de la velocidad de los fotogramas (frame rate).
80 Sin Time.deltaTime, el objeto se movería a diferentes velocidades según el rendimiento del dispositivo.
```

```
81
82     velocidad
83     Es una variable pública que define qué tan rápido se mueve el objeto.
84     */
85
86     /*
87     ¿Por qué se usa Time.deltaTime en transform.Translate() si ya estamos usando Input.GetAxis("Horizontal")?
88     es para que el movimiento sea consistente y no dependa de la velocidad de fotogramas (FPS).
89
90     Input.GetAxis("Horizontal") devuelve un valor entre -1 y 1, dependiendo de la entrada del usuario:
91     -1 → si presionas "A" o flecha izquierda (mover a la izquierda).
92     1 → si presionas "D" o flecha derecha (mover a la derecha).
93     Valores intermedios (0 a ±1) → si usas un joystick o presionas ligeramente una tecla.
94     🚩 Importante: Input.GetAxis() no tiene en cuenta el tiempo transcurrido entre frames.
95
96
97     Time.deltaTime representa el tiempo transcurrido desde el último frame.
98     En PCs rápidos con alto FPS, los frames se procesan más rápido, por lo que sin Time.deltaTime, el objeto se movería más rápido de lo esperado.
99     En PCs lentos con bajo FPS, los frames se procesan más lento, y sin Time.deltaTime, el objeto se movería más despacio de lo esperado.
100     ✅ Multiplicar por Time.deltaTime hace que el movimiento sea uniforme en todas las computadoras y dispositivos, sin importar el rendimiento.
101     */
102 }
103 }
104
```