Actividad 1: creación de un laberinto

**Objetivos**

A través de esta actividad, adquirirás soltura en dos de las tareas fundamentales del desarrollo de videojuegos con Unity: la construcción de escenarios, usando el editor, y la elaboración de *scripts* en C#.

**Pautas de elaboración**

La actividad consiste en elaborar un videojuego en primera persona en el que el jugador debe escapar de un laberinto, el cual estará formado por una plataforma y un conjunto de muros.

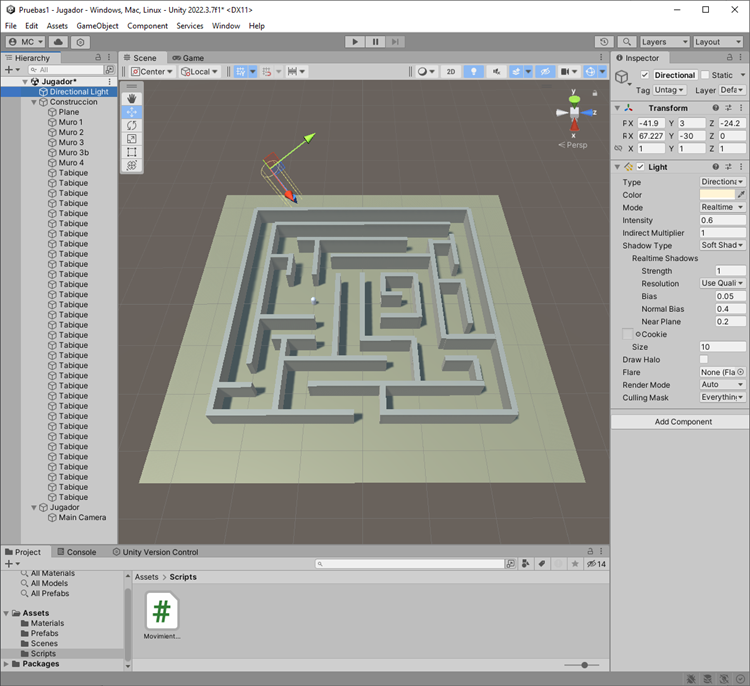


Figura 1. Pautas de elaboración actividad 1. Figura: elaboración propia.

El *game object* que representa el avatar del jugador ha de ser una cápsula con la *main camera* dentro.

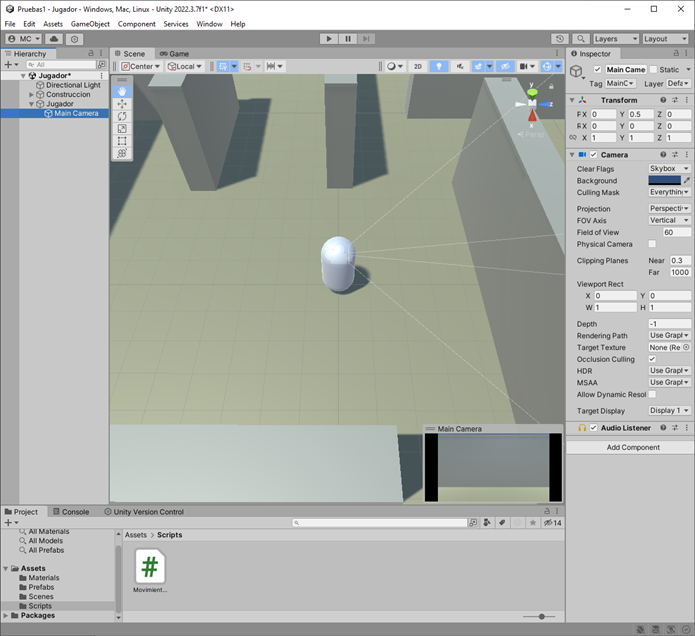


Figura 2. Pautas de elaboración actividad 1. Figura: elaboración propia.

Los pasos que hay que dar para construir el juego son:

1. Preparar dos materiales para dar color, uno al suelo y otro a los muros. Se acepta que sean colores básicos, como los de la imagen, aunque también puede ser interesante buscar otros más elaborados.
2. Preparar un *prefab* para los muros.
3. Crear la cápsula para el personaje y colocar la cámara como *game object* hijo.
4. Sustituir el *capsule collider* de la cápsula por un *carácter controller,* que facilita el control y el movimiento del personaje.
5. Agregar un *script* al personaje para implementar su movimiento.
6. Construir el laberinto usando los muros, tratando de generar una dificultad media, ni muy fácil ni demasiado difícil.

Algunas orientaciones sobre la programación del *script:*

* El jugador debe controlar el giro y el desplazamiento del personaje. Para el giro, se usan los cursores izquierda y derecha o las teclas A y D. Para el desplazamiento, se usan los cursores arriba y abajo o las teclas W y S.
* El *script* obtiene del jugador el valor del giro y el valor del desplazamiento. Esto se hace utilizando los métodos:
  + Input.GetAxis("Horizontal").
  + Input.GetAxis("Vertical").
* Para que el jugador configure la velocidad del giro y la velocidad de avance (o retroceso), se deben declarar dos variables, que deben aparecer en el inspector de Unity. Estas variables son **factores** que afectan (multiplicando) al valor del giro y del avance.
* Primero se gira al personaje, usando el método Transform.Rotate. Tras girar, el avance debe producirse sobre la posición girada. Es decir, el personaje avanza siempre en la dirección en la que está mirando.
* Para evitar que los FPS *(frames per second),* que no son constantes, afecten a la velocidad de avance, se puede utilizar Time.deltaTime. Se usa como otro factor del avance (multiplicando).
* El avance calculado (en un vector) debe transformarse del local *space* (espacio local) al *world space* (espacio de la escena). Para eso, se usa el método transform.TransformDirection.
* El avance efectivo debe realizase sobre el componente CharacterController de la cápsula, usando el método Move.
* Obviamente, toda esta operación se realiza en la función Update del *script.*

**Extensión y formato**

La entrega consiste en el propio proyecto de Unity creado. Ha de ser posible jugarlo directamente, dándole a *play* nada más abrirlo, y el *script* debe ser claro y estar bien comentado.

**Rúbrica**

|  |  |  |  |
| --- | --- | --- | --- |
| Creación de un laberinto | Descripción | Puntuación máxima  (puntos) | Peso  % |
| Criterio 1 | La entrega se adecúa a lo que se pide | 2 | 20 % |
| Criterio 2 | El script compila y el movimiento del jugador es correcto | 2 | 20 % |
| Criterio 3 | El laberinto está completo y bien construido | 3 | 30 % |
| Criterio 4 | El script está bien elaborado, con un código solvente y comentarios donde resulte adecuado | 3 | 30 % |
|  |  | **10** | **100 %** |