

Laberinto VR

De la práctica que se avanzó la vez pasada del laberinto se agregaron varias funcionalidades que se solicitaron.

Enemigos:

- Para los enemigos se pidió que se instanciaran en varios puntos y que fueran hacia el jugador, para lograr este efecto se crearon varios transforms que son la posición donde se van a instanciar los enemigos. Mediante código con la función instantiate se instancian enemigos, en mi caso se van añadiendo a una lista de enemigos para que nunca aparezcan mas de cierta cantidad en el mapa, además con la función distance y el transform del jugador se pueden aparecer enemigos solo de los spawners que esten cerca del jugador.
- Los enemigos se instancian además de forma controlada con un Timer, de esta forma no se crearán todos los enemigos de forma simultánea.
- Para que los enemigos sigan al jugador se utiliza el NavMesh agent ya que tiene una función que se llama destination, esto nos permite darle un destino a nuestro enemigo que en este caso es el transform del jugador. Esta destination solo se va a asignar si el jugador entra en un trigger que representa la visión del enemigo, de otra forma tomarán rutas al azar hasta poder encontrar al jugador.
- Para que los enemigos puedan dañar al jugador se crearon dos triggers, en este caso como la animación de ataque del enemigo son patadas, se agregaron uno a cada pierna del enemigo, si este trigger toca al jugador se llamará a una función que se llama GetHurt() que disminuye la vida del jugador en 1. El jugador se vuelve invulnerable por 0.5s para evitar ser dañado dos veces por el mismo golpe



Figura 1. Enemigo corriendo hacia el jugador

Balas:

- Para que las balas aparezcan se usa de igual forma instantiate que llevará la misma dirección del jugador para que salgan siempre de la dirección donde estamos viendo, además utilizamos un AddForce impulse para hacer que la bala avance hacia la dirección en la que fue lanzada.
- Si la bala colisiona con cualquier objeto esta se destruye gracias a la función Destroy(gameObject)
- Si la bala colisiona con un enemigo va a destruir al enemigo y luego se destruirá a si misma, además aumentará en 1 el contador de kills



Figura 2. Bala instanciada del arma.

GUI

- Para crear la gui se utilizó un canvas y varios TextMeshPro (uno por cada atributo que se desea mostrar)
- Creamos un script para modificar los valores del texto en tiempo real, para hacer esto creamos varias variables públicas TMP_Text y a estas les asignamos la cantidad de muertes, el tiempo transcurrido y la vida del jugador, estos valores se van obteniendo de cada componente con ayuda de GameObject.FindGameObjectWithTag("nombre_del_objeto")
- Dentro del mismo script podemos agregar además del entero que va a irse modificando, el texto inicial de cada text, esto simplemente igualando el valor del text a "nombre a mostrar" + valor a mostrar;



Figura 3. GUI al inicio de la partida



Figura 4. GUI al recibir daño

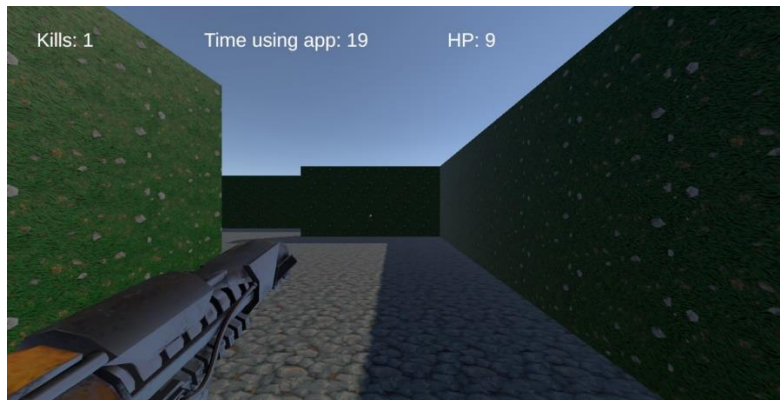


Figura 5. GUI al destruir enemigo e instancia de bala.

GameOver y Win

- Para el GameOver declaramos en el script del movimiento del jugador un entero con 10 vidas, cuando se pierden todas las vidas simplemente se volvera a cargar la escena importando la biblioteca `UnityEngine.SceneManagement`, de esta forma el jugador si

pierde tendrá que volver a recorrer el laberinto, para recargar la escena se crea una variable Scene a la cual, se le asigna la escena actual con ayuda de la función SceneManager.GetActiveScene().

- Para cargar la escena se utiliza la función SceneManager.LoadScene(variable_creada.name)
- Para el Win se creó un gameObject con un boxCollider posicionado justo al final del laberinto, si el jugador entra en el boxCollider quiere decir que ganó la partida así que se crea un botón de replay para que pueda volver a empezar. Para esto se creó un botón que tiene asociado en la función On Click() la misma función que vuelve a cargar la escena.

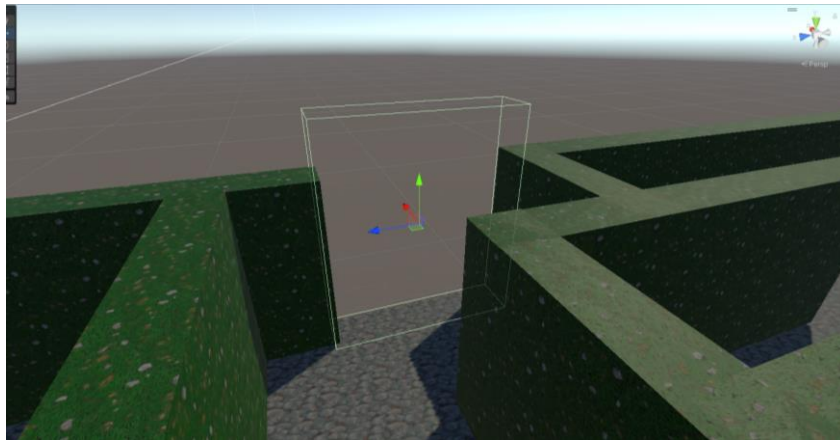


Figura 6. Trigger de victoria.

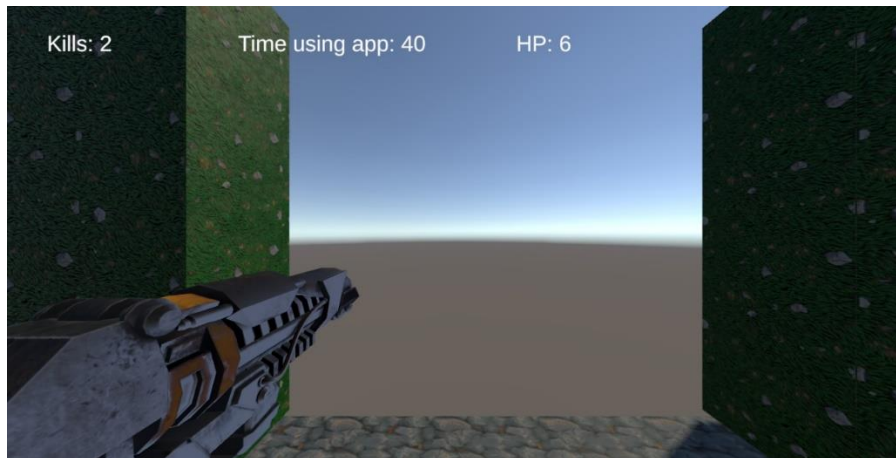


Figura 7. Trigger de victoria (vista del jugador).



Figura n. Pantalla de replay