

Memoria motores Unity

Frías Bernal, Roberto
`roberto.frias@live.u-tad.com`

April 23, 2018

Contents

1	Descripción	2
2	Controles	2
3	Funciones	2

1 Descripción

Me he basado en el juego 1942 para hacer un juego de naves con un enfoque un poco más "bullet hell". El juego consiste en conseguir la máxima puntuación posible con 6 vidas, para ello hay que destruir las naves enemigas que van apareciendo y el boss que aparece cuando han pasado un número concreto de waves.

Si se consigue un nuevo récord de score este se guarda para que el jugador pueda saber cual es, incluso entre partidas distintas. Y para mantener interesante el juego las waves son aleatorias y los fondos que van saliendo también.

2 Controles

- **Teclas direccionales y W/A/S/D:** Controlan el movimiento del player.
- **Espacio:** Disparo del player.
- **Escape:** Menú de pausa, se controla con el ratón.

3 Funciones

- **BackgroundController:** Se encarga de carga y descargar los fondos que van pasando por la pantalla. Los que carga los elige de manera aleatoria de una lista de posibles fondos. Utilizó un fragmento de agua entre fondo y fondo para hacer la union mas sencilla.
 - **Start():** Creó un fondo de barco que siempre es el mismo y el primer fondo aleatorio con su fragmeto de agua que los une.
 - **Update():** Aquí es donde voy descargando los fragmentos pasados y cargo los nuevos de manera aleatoria.
- **BossController:** Este es el controlador del jefe.
 - **Start():** Inicializo todo lo que voy a necesitar.
 - **Update():** Se mueve hasta la posición indicada y entonces dispara, también se encarga de restablecer el color si ha recibido daño.
 - **Shoot():** Esto son los disparos normales que hace en dos líneas en un intervalo indicado.
 - **ShootGuided():** Este es el misil que dispara a donde este el player en ese momento en un intervalo indicado.
 - **Damage():** Cuando le hacen daño le restó vida y le cambio su color a rojo para darle feedback al player, y cuando muere ejecutó la animación de muerte y añado su scoreValue al score del player.

- **WaitForDeathAnimation():** Esto lo hice porque no era capaz de hacer que se destruyera el boss una vez se ejecutara la animación de muerte, así que lo que hice fue que esperara un tiempo y luego ya destruyera.
- **GetHealth():** Devuelvo la vida del boss.
- **OnTriggerEnter2D():** Si el player decide chocarse con el boss le hago daño al player.
- **BulletController:** Controlador de las balas, tanto del player como de los enemigos y el boss.
 - **OnTriggerEnter2D():** Aquí es donde llamo al Damage() según con quien haya chocado, esto funciona en todos los casos porque la matriz de Physic2D están activadas sólo las colisiones necesarias.
 - **OnTriggerExit2D():** Para que las balas no existan fuera de la pantalla cuando pasan de los bounds se eliminan.
- **EnemyController:** Controlador de los enemigos.
 - **Start():** Inicializo todo lo que voy a necesitar.
 - **Update():** Movimiento y el tipo de disparo que realiza el enemigo.
 - **Shoot():** Disparo normal que realizan los enemigos principales, son balas que viajan hacia abajo en la pantalla y que tienen un temporizador mas una probabilidad de ser disparadas.
 - **ShootGuided():** Disparo de misil que va hacia la posición del player en ese instante, lo realizan los enemigos secundarios, al igual que las balas tienen un temporizador y una probabilidad de ser disparados.
 - **Damage():** Le resto vida cuando sufre daño y cuando su vida llega a cero lo elimino y añado su scoreValue al score del player.
 - **OnTriggerEnter2D():** Si se chocan con el player se eliminan y realizan daño al player.
 - **OnTriggerExit2D():** Aquí controlo que no disparen hasta que estén dentro del área de juego y que cuando salgan de esta se eliminan.
- **FadeText:** Función para hacer desaparecer y aparecer el texto.
 - **Start():** Simplemente inicio HideUnhide().
 - **HideUnhide():** Activo o desactivo el texto según el timer que tenga para darle un efecto de "fade".
- **HUDManager:** Singleton del hud del juego en el que muestro la vida, el score y el highscore.
 - **Awake():** Cargo la variable highscore si existe de esta manera aunque se cierre unity el highscore se carga al volver a iniciar.

- **Update():** Actualizo el texto de score y highscore, y si score supera el highscore lo sustituyo.
- **Layers:** Lista de las layers que tengo para acceder de manera más sencilla desde el código.
- **LoadScene:** Pequeño script para cargar una escena, que le indico, cuando se presiona cualquier tecla.
 - **Update():** Si presiona cualquier tecla cargo la escena que me hayan pasado y reinicio el timescale por si esta a 0.
- **LoadUnloadBackgroundController:** Este controlador es el que tienen los fondos para avisar al BackgroundController que tiene que hacer si cargar o descargar.
 - **OnTriggerEnter2D():** Los fondos tiene dos triggers, cuando paso por primera vez por la parte de abajo de los bounds de la escena activo el bool para hacer saber al BackgroundController que tiene que descargar el anterior background, y cuando pasan por la parte de arriba de los bounds de la escena si es el primer trigger no hago nada, pero cuando es el segundo se que tengo que cargar el siguiente fondo así que cambio el bool para que el BackgroundController lo sepa.
- **PauseMenu:** Menú de pausa que sale al presionar escape durante la partida.
 - **Update():** Si se presiona escape entro en el menu y permito que presionen resume o menu.
 - **Resume():** Salgo del menu de pausa, esto es mediante click del botón que hace referencia a esta función o presionando otra vez escape.
 - **Pause():** Pongo el timescale a cero y muestro el menú de pausa.
 - **LoadMenu():** Si presiono el botón menu del menú de pausa me voy a la escena de menu.
- **PlayerController:** Controlador del player.
 - **Start():** Inicializo todo lo que voy a necesitar.
 - **Update():** Si está vivo se puede mover dentro de los bounds de la escena, seteo la animación hacia izquierda, derecha o idle, y le permito disparar si presiona espacio.
 - **Damage():** Si tiene vida le resto cuando sufre daño y actualizo la UI, si no le queda vida llamo a la animación de muerte.

- **WaitForDeathAnimation():** Al igual que en el boss no estaba seguro de como hacer para esperar a que se eliminará después de ejecutar la animación así que lo he hecho con un tiempo que seteo yo aqui tambien.
- **OnDestroy():** Si se destruye el player guardo en las variables de unity el highscore para cargarlo en la próxima partida.
- **Singleton:** Singleton que utilizo con el HUDManager.
- **WaveController:** Controlador de oleadas, cuando pasan la cantidad de waves indicadas hay una wave especial que es el jefe.
 - **Start():** Inicializo todo lo que voy a necesitar.
 - **Update():** Controlo el spawn de las oleadas en la cantidad indicada esperando el tiempo entre ellas indicado también, una vez no quedan más oleadas normales instancio una de jefe y una vez muere el jefe vuelvo a cargar waves normales y así continuamente.