

FUNDAMENTOS DE VIDEOJUEGOS

GCC Minijuego

24/25

Daniel Berdasco González
b190388
d.berdasco@alumnos.upm.es

Álvaro Salamero García
b190364
alvaro.salamero.garcia@alumnos.upm.es

Control de versiones

Identificador	Contenido
MJ V1	Creación de documento

Tabla 1 – control de versión

Índice

Resumen del juego.....	3
Escenas.....	3
Assets y GameObjects.....	3
Mecánicas.....	4
Scripts.....	4
Interacción	4

Resumen del juego

Este minijuego trata de modelar de manera sencilla el popular juego de arcade conocido como Asteroids, lanzado en 1979 por Atari. En nuestro caso, además se han utilizado sprites distintos a los ofrecidos por el profesor, para dar un enfoque más ocurrente y puede que dinámico a este juego.

En la medida de lo posible, se ha tratado de desarrollar un minijuego parecido a una de las batallas más famosas del mundo del anime, pues se intenta recrear la pelea entre Goku y Jiren, de la serie Dragon Ball Super.

Escenas

Este juego contiene una única escena, donde se desarrolla toda la jugabilidad del mismo. Contiene el fondo, que es una captura del espacio donde tiene lugar el Torneo del Poder de la serie en la que se basa el juego, así como el jugador el cual es Goku ultra instinto, Jiren a modo de enemigo, y los kamehameha cuando el jugador dispara.

Assets y GameObjects

Los recursos o assets empleados son:

- Fondos: espacio del Torneo del Poder y pantalla semi transparente.
- Imagen de jugador: Goku ultra instinto.
- Imagen de enemigos: Jiren.
- Imagen de balas: imagen de una onda vital.
- Música: Dragon Ball Super - Ultimate Battle | Ultimate Metal Vocal COVER – Fiedrich Habelter

Los GameObjects utilizados son:

- Jugador: tiene movimiento vertical y horizontal (estilo tanque).
- Enemigos: aparecen desde el borde superior y caen en vertical a modo de meteoritos.
- Balas: onda vital disparada por el jugador.
- Menú de espera: con dos botones, continuar y salir.

Cabe destacar que estos gameobjects tienen detección de colisiones para poder interactuar (tal y como se detalla en el apartado de mecánicas e iteracción) entre ellos.

Mecánicas

Podemos dividir las mecánicas del juego en 3 apartados principales:

- Mecánica del jugador: puede moverse por el espacio infinito, tanto en vertical como en horizontal, si bien para este sigue un movimiento de tanque, es decir, debe de girar primero antes de emplear los botones del movimiento vertical para desplazarse hacia el lado X. Para ello, se utilizan las flechas arriba/abajo para el movimiento vertical, y las de izquierda/derecha para el giro. Asimismo, el jugador puede lanzar kamehamehas pulsando el botón espacio.
- Mecánica de enemigos: un enemigo hace spawn en el límite superior de la pantalla cada 3 segundos, si no es derrotado en 4 segundos, este desaparece. Para derrotarlo basta con que sea alcanzado por un kamehameha del jugador. Si el jugador colisiona con un enemigo, el juego se reinicia. El movimiento de este es en caída desde arriba hacia abajo.
- Mecánicas de balas: un kamehameha aparece cada vez que el jugador pulsa la tecla espacio. Si esta impacta contra un enemigo, este desaparece. Si no impacta contra nada, esta desaparece a los 4 segundos. El movimiento del kamehameha es siempre recto, en la dirección hacia la que esté apuntando el jugador.

Scripts

El juego se compone de 5 scripts, los cuales son:

- Player.cs: el script principal del jugador detalla sus movimientos, así como los límites de este en pantalla y las colisiones con otros elementos.
- EnemySpawn.cs: describe la aparición de enemigos y todo lo relacionado con ello (ubicación, tiempo) así como sus colisiones.
- Kamehameha.cs: indica todos los aspectos relacionados con las “balas” empleadas por el player, en este caso con la forma de un ataque muy famoso.
- MusicPlayer.cs: script para definir la música empleada durante el gameplay.
- GamePause.cs: empleado para poder pausar el juego.

Interacción

El objetivo del jugador es navegar por el espacio en el que se desarrolla el juego, mientras dispara a todos los enemigos posibles y sin que sea tocado por ellos, ya que en ese caso, el juego se reiniciará. El juego cuenta con un contador, SCORE, que cuenta el número de enemigos abatidos en una partida; este volverá a ser 0 en caso de reinicio.