

Informe sobre la práctica final parte 2

La práctica se ha realizado en parejas:

Alumno 1: José Miguel Márquez Herreros 77023123X

Alumno 2: Antonio Haro Arriaza 77799905J

Clases Particula y ConjuntoParticulas

En cuanto a la clase Particula, la implementación de los operadores que se pedían fue bastante sencilla y no ocasionó ningún problema. También se añadió el operador >> que se pedía de forma adicional en las instrucciones de entrega y se le añadió un test en pruebaConjuntoFinal.cpp (test 8).

En la Clase ConjuntoParticulas se implementó todo lo que se pidió de la parte obligatoria sin mayores dificultades. Todo funcionaba sin problemas pero nos dimos cuenta al pasar valgrind que había pérdidas de memoria en el operador de asignación, esto sucedía debido a que no se liberaba la memoria antes de hacer la reserva previa a copiar los datos. Nada que no se solucione con una llamada al método liberarMemoria antes de llamar el método privado copiar.

Minijuego

Para construir el minijuego los datos se leen de un archivo de texto externo, tuvimos un problema el cual leemos bien el archivo pero no lo situabamos en la carpeta correcta. No le encontramos mucha dificultad a esta parte salió todo bastante bien y rápido. No creo que le haya faltado nada, era bastante completa y tampoco se complicaba mucho. Hicimos pruebas adicionales para probar lo que teníamos nuevo.

Tareas adicionales

1- Encapsulamos todos los objetos y lógica del juego en una clase Game que tenga como métodos Init() y Update(). También le añadimos dos constructores, uno por defecto (juego a 5 rondas con 3 vidas) y otro que recibe como parámetro el nombre de un fichero que contiene las características del juego.

2- La características del minijuego son dadas por la lectura de un archivo de texto. La estructura del texto es:

- **ancho:** Este número es el ancho en píxeles de la pantalla del minijuego.
- **alto:** Este número es el alto en píxeles de la pantalla del minijuego.
- **ovnis:** Es el número de ovnis inicial en la primera ronda, en las rondas siguientes el número de ovnis será $ovnis * \text{número de la ronda actual}$.
- **primer color:** Es el color de los ovnis. Debe ser un color admitido por raylib.
- **segundo color:** Es el color de los proyectiles. Debe ser un color admitido por raylib.
- **tercer color:** Es el color de la nave. Debe ser un color admitido por raylib.
- **FPS:** Son los Frames por segundo del minijuego.
- **vidas:** Es el número de vidas del jugador.
- **rondas:** número de oleadas de ovnis, el juego no acaba hasta que se terminan las rondas o el jugador muere antes.

3- Se le ha añadido un disparo especial que hace que la nave dispare 5 proyectiles en cono. Para lograr este efecto hemos usado funciones trigonométricas (seno y coseno) para calcular las velocidades en los ejes xy a partir del ángulo que le quisiéramos dar a cada proyectil. Para esto fue necesario incluir la biblioteca cmath. Este disparo se activa mediante la tecla ALT izquierdo a diferencia del disparo normal que se activa con la barra espaciadora y solo se puede utilizar una vez cada 5 segundos. Para indicar cuando este disparo especial está disponible se ha añadido arriba a la izquierda de la interfaz un contador regresivo de 5 segundos, cuando este sea 0 el disparo está disponible y al utilizarse vuelve a ser 5.

4- La nave tiene una inmortalidad de 3 segundos cuando los ovnis le dan, para dar mayor facilidad al jugador y que no pierda tan rápido. El cálculo de los segundos tanto para el disparo especial y el tiempo de inmortalidad viene dado por $FPS * tiempo \text{ en segundos}$. Así se obtienen los frames que tienes que pasar desde la última vez que se disparó o el jugador perdió una vida.

5- La nave se puede mover hacia todas las direcciones. Esto se corrigió añadiendo a la nave velocidad en el eje y para que se pueda mover arriba y abajo.

6- Hemos puesto pantallas de victoria y derrota dependiendo del resultado de la partida. Se hace con una variable booleana GameOver para saber si se ha acabado el juego y otra ganador para saber si mostrar la pantalla de victoria o derrota.

7- El juego se puede pausar con la tecla P y el juego lleva registros internos sobre el número de ovnis destruidos por el jugador para que en la pantalla de victoria o derrota le muestre al jugador su puntuación. También muestra el número de ovnis restantes en una ronda.