PROYECTO DE PRÁCTICAS: VIDEOJUEGO

Apellidos: Gil Planes

Nombre: Germán

D.N.I.: 49246858-V

Asignatura: Tecnología de la Programación

Grupo: PCEO

Convocatoria: Junio

Año académico: 2020-2021

Profesor: Juan Antonio Sánchez Laguna

Índice de contenidos:

- ✓ flor_fuego.bmp
- ✓ fondo_mario.bmp
- ✓ fuego.bmp
- ✓ gafas.bmp
- ✓ tux.bmp
- ✓ vidas.bmp

- **Estructuras de datos**
  - ✓ heroe
  - ✓ enemigo
  - ✓ Bala
  - ✓ Rafaga
  - ✓ Ejercito
- **Conclusiones**

Descripción de la aplicación

Objetivo.

El objetivo ha sido lograr asimilar los conocimientos aprendidos en esta asignatura a través de un proyecto práctico, el cual me ha servido para familiarizarme y acercarme a la programación en C.

Funcionamiento general.

La aplicación consta de un código principal archivado en el fichero *main.c*. En él va incluida la declaración del programa principal y, desde ahí, se va ejecutando toda la aplicación. Desde la función *main* se va llamando a otras funciones, guardadas en otras librerías. Estas han sido creadas para clasificar los procedimientos en módulos de acuerdo a su funcionalidad. Al ejecutarse el programa se genera la pantalla sobre la cual se desarrolla el videojuego. Se crea una variable *(var_menu)* que tomará un valor entero comprendido entre 0 y 3:

- Si su valor es 0, entonces se ejecutará la función correspondiente al menú principal de la aplicación.
- Si su valor es 1, se ejecutará el código de la función *jugar*, y se reproducirá entonces el juego diseñado.
- Si su valor es 2, se abre una pantalla de ayuda, explicando brevemente los controles y reglas de juego.

- Si su valor es 3, finalizará la ejecución del programa.

Cabe mencionar que el valor que toma la variable, inicialmente, depende del botón que pulse el usuario de entre los siguientes: “Jugar”, “Ayuda” y “Salir”.

Una vez que pulsa “Jugar”, la variable *var_menu* toma el valor de 1. Volverá a ser 0 una vez que el jugador haya perdido en la partida y regresará al menú.

Lo mismo sucede si el jugador decide entrar en la pantalla de ayuda. El valor de *var_menu* será 2 y volverá a ser 0 cuando pulse sobre el aspa de cerrar la pantalla de ayuda. Es entonces cuando regresará al menú principal.

Manual de usuario

Al ejecutar la aplicación se abrirá el menú principal del juego son tres botones: “Jugar”, “Ayuda” y “Salir.

1.- Si pulsa el primer botón, empezará la partida. El héroe del videojuego es un pingüino que se desplaza lateralmente a lo largo de la parte inferior de la pantalla. Para poder manejarlo solamente hay que usar las flechas que marcan la dirección de izquierda y derecha.

El botón del espacio permitirá al personaje poder disparar una ráfaga de balas. Sin embargo, el número de balas está limitado y, es posible que haya momentos en los que el héroe no disponga de munición. Si esto ocurre saltará la señal de “SIN BALAS” y se verá como la barra, que muestra la cantidad de balas disponibles en el cargador, estará vacía. Las balas se van regenerando automáticamente conforme se van liberando al salir por la parte superior de la pantalla.

Los enemigos son los calamares del Super Mario Bros. Estos se moverán independientemente por la pantalla y tratarán de alcanzar al pingüino. Si lo consiguen, el jugador perderá una de las tres vidas que tiene al comienzo de la partida. En ese momento, al personaje se le pondrán unas gafas de bucear y un escudo. Además, saltará por pantalla un mensaje diciendo “Invulnerabilidad activada” y durante un tiempo el héroe será inmune a los ataques enemigos.

También se podrá ver, en todo momento, la puntuación (esquina superior izquierda, junto con la barra de las balas que quedan) y las vidas restantes (esquina superior derecha).

En el momento en el que el jugador pierda las tres vidas, regresará automáticamente al menú principal. Si ha habido un nuevo récord, se modificará en el archivo “record.txt”.

2.- Si pulsa el segundo botón, se mostrará la pantalla de ayuda. Para volver, el usuario sólo debe pulsar el aspa situada en la esquina superior derecha de esa pantalla de ayuda.

3.- Si pulsa el tercer botón, saldrá de la aplicación.

Organización del proyecto

- Código.cbp - Fichero de proyecto Code::Blocks.
- main.c – Código principal del programa. Requiere de la inclusión de los módulos pantalla, ejercito, colisiones y heroe.
- heroe.c – Implementación de las funciones relacionadas con el héroe. Requiere de la inclusión del módulo pantalla.
- heroe.h – Interfaz del módulo heroe.
- enemigo.c – Implementación de las funciones relacionadas con el héroe. Requiere de la inclusión del módulo pantalla.
- enemigo.h – Interfaz del módulo enemigo.
- bala.c – Implementación de las funciones relacionadas con el héroe. Requiere de la inclusión del módulo pantalla.
- bala.h – Interfaz del módulo bala.
- colisiones.c – Implementación de las funciones relacionadas con las colisiones.
- colisiones.h – Interfaz del módulo colisiones.
- pantalla.c – Implementación de las funciones relacionadas con las colisiones.
- pantalla.h – Interfaz del módulo pantalla.
- rafaga.c – Implementación de las funciones relacionadas con la ráfaga.
- rafaga.h – Interfaz del módulo rafaga. Requiere de la inclusión del módulo bala.
- ejercito.c – Implementación de las funciones relacionadas con ejército. Requiere de la inclusión de los módulos pantalla, enemigo, colisiones y rafaga.
- ejercito.h – Interfaz del módulo ejercito.
- Requiere de la inclusión del módulo rafaga.

Además, también hay incluidas imágenes en formato .bmp, que sirven de complemento para el diseño del programa: aspa_blanca.bmp, aspa_roja.bmp, ayuda.bmp, bola_fuego.bmp, calamar.bmp, escudo.bmp, explosion.bmp, flor_fuego.bmp, fondo_mario.bmp, fuego.bmp, gafas.bmp, tux.bmp y vida.bmp. Así como un fichero record.txt para almacenar el récord de puntuación.

Estructuras de datos

Los TDA que se han definido hacen la función de punteros que apuntan a una estructura. Esa estructura será de la forma adecuada para representar a un personaje u objeto del vídeo. Los Tipos de Datos Abstractos que se han diseñado son los siguientes:

- heroe – Apunta a struct heroeRep, que es una estructura capaz de almacenar valores cómo para representar al héroe del videojuego: coordenada x (número

real), coordenada y (número real), altura (número real), anchura (número real), velocidad horizontal (número real), velocidad vertical (número real) y una imagen.

- enemigo – Apunta a struct enemigoRep, capaz de almacenar los mismos datos que la estructura heroeRep.
- Bala – Apunta a struct BalaRep, también capaz de almacenar los mismos datos que las dos estructuras anteriores.
- Rafaga – Apunta a struct Nodo. Esta estructura almacena dos valores: El primero es una bala de tipo Bala (TDA anterior) y el segundo es otro nodo del tipo Struct Nodo.
- Ejercito – Apunta a struct EjercitoRep. Esta almacena un array de elementos del tipo enemigo (segundo TDA) y un entero, que representa la cantidad de enemigos que hay incluidos en el array. Ese número será útil de saber ya que la cantidad de elementos va a estar limitada.

Conclusiones

En cuanto al grado de satisfacción con la asignatura, en la medida en que constituye un instrumento práctico de aplicación de los conocimientos teóricos que se obtienen simultáneamente con el desarrollo del trabajo, ha resultado evidentemente productivo. Bajo mi punto de vista es un buen método, si no el mejor, para asimilar los conceptos desarrollados en la asignatura.