Escuela de computación

Curso: IC 2001 Estructura de datos

Profesor: Víctor Garro Abarca

Estudiantes:

Bryan Daniel Barquero Barrantes

José Andrés Quesada Artavia

Proyecto 1A – Shotter Espacial

Manual técnico

Segundo semestre 2021

3 de noviembre 2021

Tabla de contenidos

Estructuras de datos utilizadas

Para la implantación del programa se utilizó diferentes mecanismos para realizar la ejecución de cada una de las funciones de los objetos que se muestran en pantalla.

Para realizar un programa de una mejor calidad primeramente se utilizó la creación diferentes clases y de esa forma tener un control de cada estructura de datos que se va a utiliza en pantalla asignándole sus propias funciones y atributos internos. Estas clases son: BackGround, Disparo, Enemigo, Global, Highscores, Nave y Test.

La ejecución de cada una de las acciones que se realizan en tiempo real será llevada a cabo mediante una cola de eventos la cual registrará cada acción que se realice en la ejecución de los eventos, haciendo que de esa forma se tengan diferentes contadores según los que se esté realizando en orden de la entrada de estructuras.

```
TestQueue = al_create_event_queue();
al_register_event_source(TestQueue, al_get_keyboard_event_source());
al_register_event_source(TestQueue, al_get_display_event_source(DISPLAY));
al_register_event_source(TestQueue, al_get_timer_event_source(timer));
```

Los enemigos son agregados dentro de un array el cual consta de una matriz de 6 por 4, esto permitirá que las oleadas salgan en un orden predeterminado para luego cada uno de los enemigos tenga un movimiento hacia la izquierda y de arriba abajo de forma aleatoria hasta recorrer la pantalla o ser eliminados.

Principales Subrutinas

El algoritmo consta de 4 faces de ejecución las cuales determinan que funciones se van a ejercen en el momento. La primera fase consta de la pantalla inicial en la cual se presenta el movimiento de la pantalla de fondo además de los textos con las opciones del juego.

La fase dos y principal, consiste en la ejecución del juego y que lleva todos los llamado de las funciones del programa, esta fase lleva a cabo la creación de la nave de juego con sus atributos y características visuales, fondo de pantalla el cual consta de 3 imágenes donde están distribuidas por capas, la primera es un fondo fijo con estrellas, el segundo es un fondo que está en movimiento a una velocidad baja sobre la posición en el eje x y el tercer es también un fonde con estrellas pero que va a mayor velocidad de igual forma sobre el eje x, esto de forma repetida para simular el movimiento de la nave del jugador a través del espacio.

En esta fase también se maneja la creación de naves enemigas lo que costa de crear cada uno de los objetos con sus respectivos atributos, colocándolos en la matriz de distribución y siendo representados en pantalla.

También se tiene representaciones fijas para la puntuación del jugador y cantidad de enemigos destruidos. La representación de las vidas consta de un arreglo de imágenes que van cambiando según la vida que posea la nave del jugador.

Las naves enemigas poseen la capacidad de disparar de forma aleatoria cuando se es llamada la función que así lo realiza.

Tanto para la nave del jugador como para las naves enemigas verifica en cada llamado de funciones si las balas colisionan ya sea con la nave le jugador o si una de las balas disparadas por el jugador impacta a alguna nave enemiga, esto se implementa mediante la constante verificación de coordenadas do los objetos en pantalla, ya que si una bala tiene la misma ubicación de coordenadas que de una nave y si hitbox esta lo tomara como un impacto.

```
void colisionJugador()
{
    if (enemigoDisparo.status)
    {
        if (enemigoDisparo.y_pos  < jugador.y_pos + jugador.boxheight && enemigoDisparo.y_pos > jugador.y_pos - jugador.boxheight
            && enemigoDisparo.x_pos < jugador.x_pos + jugador.boxright && enemigoDisparo.x_pos > jugador.x_pos - jugador.boxleft)
        {
            enemigoDisparo.status = 0;
            jugador.health -= 10;
        }
    }

    if (jugador.health == 0)
    {
        jugador.active = false;
        estadoJuego = 3;
    }
}
```

Las naves enemigas al estar en movimiento constante y ser eliminadas por las colisiones requieren ser nuevamente colocadas en la pantalla, es por eso que una vez que las naves salen de pantalla o reinician el ciclo de movimiento vuelven a ser “activadas”.

```
void Reactivar_Enemigos()
{
    for (int i = 0; i < NUM_COLUMNAS; i++)
    {
        for (int j = 0; j < NUM_FILAS; j++)
        {
            arrEnem[i][j].active = true;
        }
    }
}
```

La fase dos finaliza una vez que la nave del jugador recibe seis impactos lo que hace que se ingrese al a fase 3 la cual se representa por una pantalla que presenta un texto con el puntaje final de a ver destruido las naves enemigas, el tener el resultado final del “marcador” se hará un llamado a la función encargada de leer el documento de texto que almacena los puntajes anteriores y se hará una comparación de estos puntajes para verificar si se debe ingresar el puntaje en el archivo,

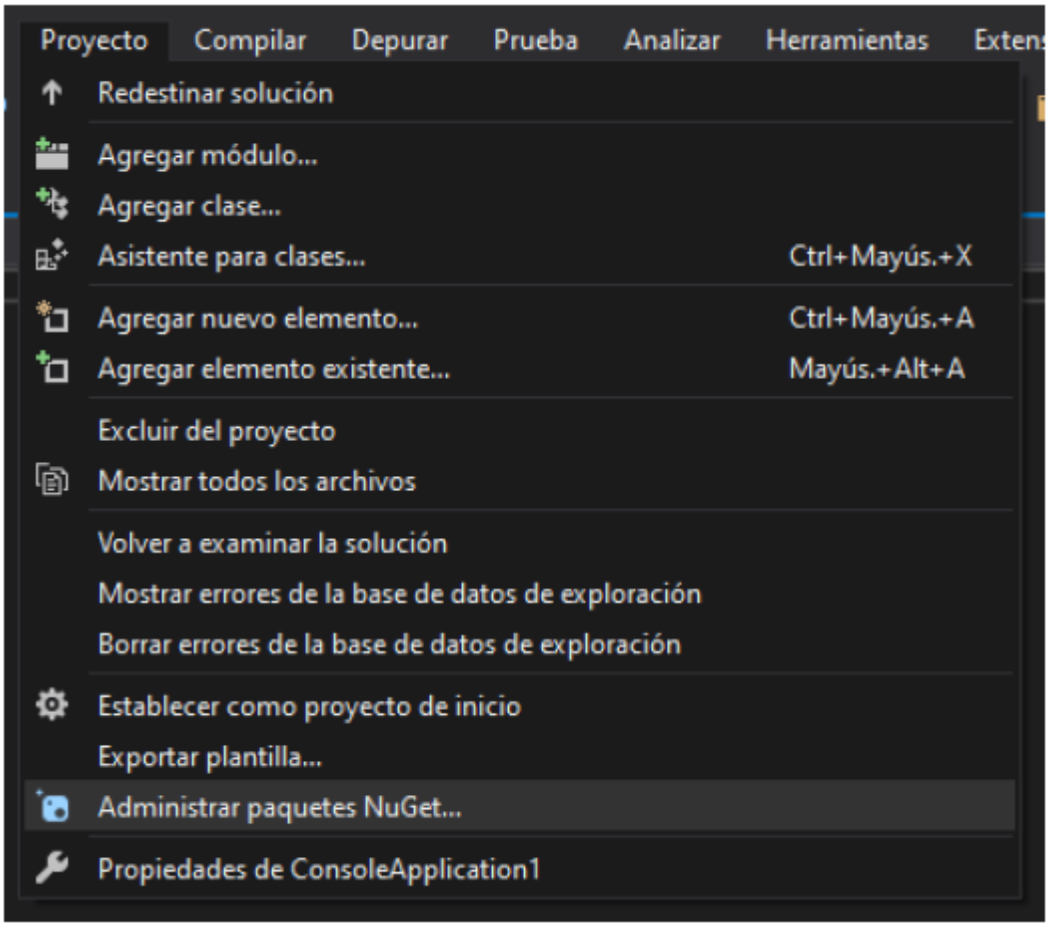
de ser así se le solicita al usuario que ingrese su nombre para ser guardado, de no hacerlo se almacenara el puntaje sin un nombre. Además de una simulación de una nave recorriendo de izquierda a derecha de forma infinita hasta accionar la tecla entre lo que hará que se pase a la fase 4 de ejecución.

La fase 4 consta de mostrar la pantalla de puntuaciones mas altas, con lo que abrirá el documento que las almacena y las mostrará en pantalla. Presionando ENTER se regresa a la primera fase para continuar con el ciclo de forma constante.

Sección de Instalación

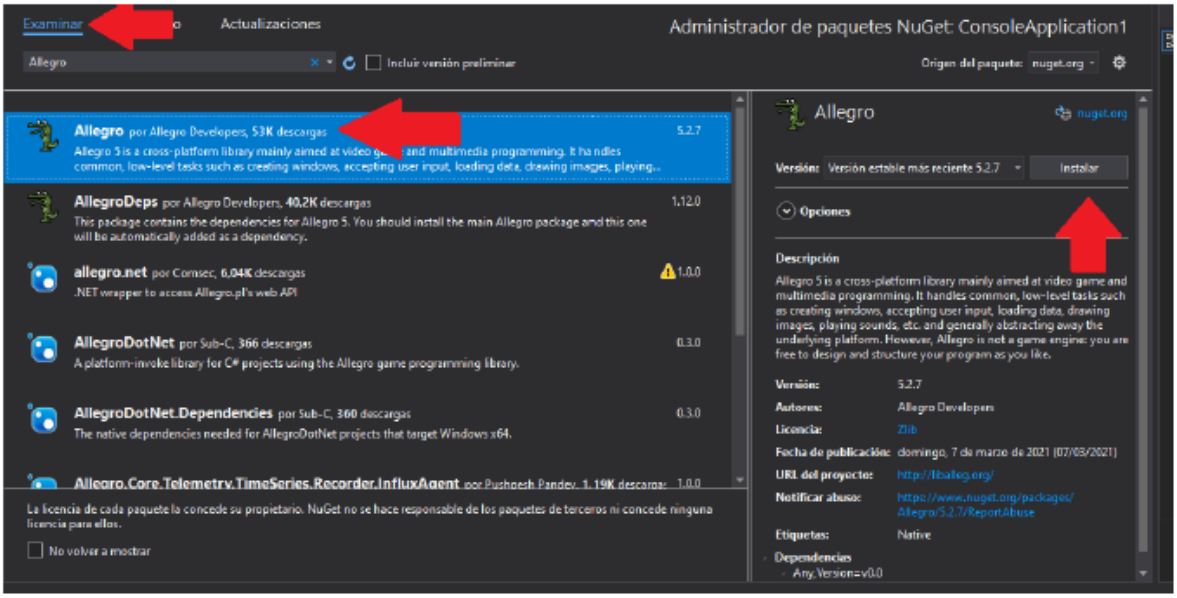
Para la ejecución de este programa es necesario realizar la instalación de la biblioteca ALLEGRO, la cual va a permitir la programación de videojuegos y multimedia. Además de maneja tareas comunes de bajo nivel, como crear ventanas, aceptar la entrada del usuario, cargar datos, dibujar imágenes, reproducir sonidos, etc.

Para realizar esta instalación primeramente se debe de ingresar desde el programa de desarrollo de software Visual Studio 2019, una vez dentro se debe ingresar a la sección de Administrar paquetes Nuget… donde se podrá realizar la instalación de la librería que se requieran en diversos programas.



Una vez dentro se procede a la sección de Examinar en donde en el área de búsqueda se debe de escribir el nombre de la librería Allegro, al encontrar en las búsquedas se presiona la primera opción para luego al lado derecho de la pantalla presionar el botón de Instalar,

seguidamente se debe de aceptar los valores de la instalación e iniciara el proceso de descarga e instalación de la librería. Esto proceso puede tardar algunos minutos dependiendo del terminal que se esté utilizando. Una vez finalizado se podrá ejecutar el programa.



Es importante verificar constantemente este apartado debido a que pueden surgir futuras actualizaciones de esta librería.