



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Fundamentos de programación

Proyecto Final

2022-1

GRUPO 04

CARRERA

Ingeniería Industrial

ESTUDIANTE

Garcia Perez Diana Paola

PROFESOR

Marco Antonio Martinez

Quintana



Fecha de elaboración y entrega 10/12/21

1.Introducción

1.1 Contenido y contexto del trabajo

El proyecto consiste en realizar y desarrollar el juego de Snake con total funcionalidad a través de un compilador en lenguaje c++ para esto, es necesario tener una selección específica de elementos de software y hardware que buscará cubrir las necesidades para desarrollar el juego es necesario que previo a esto se realice un estudio para conocer los comandos y funciones que se utilizarán ubicando todas las posibilidades y seleccionar la mejor con el menor número de funciones posibles.

La mecánica del juego consistirá en en guiar una serpiente en un determinado margen continuamente y sin parar con el objetivo de alimentarse con los rombos que aparecen en la pantalla, los movimientos se podrán realizar con las flechas del teclado, el juego termina cuando la serpiente choque contra los bordes que se encuentran como margen rectangular, en la pantalla siempre aparecerá la puntuación, esta aumentará según la cantidad de alimento que consuma la serpiente.

En el caso de este proyecto usaremos la estrategia de utilizar un proyecto que ya existe y adaptarlo a las nuevas necesidades dividiré las tareas en 3 fases :

1. Documentacion y analisis

- Estudio sobre las herramientas a utilizar
- Investigar qué librerías utilizar
- Inicio de la estructura
- Desarrollo de diagrama de flujo

2.Desarrollo y pruebas

- Desarrollo de pseudocódigo
- Codificación y pruebas de funcionalidad
- Interfaz amigable

3.Desarrollar las tareas relacionadas con la presentación

Se presenta un diagrama de gantt que tiene como función la organización del tiempo y de las tareas todo esto debido a que el proyecto tiene como fecha de entrega el día 10 de diciembre del 2021, además antes de realizar el código final se realizará un diagrama de

flujo como guía y seguidamente un pseudocódigo todo esto para que se mantenga un objetivo fijo y no existan topes que generen contratiempos en la realización de este proyecto.

1.2 Tema que aborda el proyecto final

Snake nació como Blockade en octubre de 1976, un juego arcade tipo laberinto que consistía en maniobrar una línea que no paraba de moverse. En el juego original te las veías con otros rivales. El objetivo era que tus enemigos chocasen contigo o consigo mismos mientras tú seguías en pie. Sólo podías moverte 90 grados en cada movimiento y para ello contaban con los clásicos botones direccionales.

El origen del juego snake es un juego desarrollado por Gremlin en 1976 que fue un juego para dos jugadores que consiste en el manejo de una flecha mediante el uso de botones que direccionan la serpiente el objetivo era sobrevivir consiguiendo la estela más larga sin chocar con los bordes o del mismo u otros jugadores el juego terminaba cuando el mismo jugador lograba ganar 6 veces.

En la variante Snake, el enemigo somos nosotros mismos, ya que podemos chocar contra nosotros si no vamos con cuidado. Versiones de Blockade y Snake hubo muchas. Atari creó dos versiones para Atari 2600, Dominos y Surround. Por su parte, una versión llamada Worm fue programada para computadoras Commodore y Apple II. Y en 1982 salía a la venta un juego titulado Nibbler, protagonizado por una serpiente en un escenario que recuerda al laberinto de Pac-Man (1980).

El impacto de Snake en la memoria colectiva ha sido tal que el MoMA o Museo de Arte Moderno de Nueva York se planteó incluir este juego en una de sus exposiciones. Titulada Applied Design o Diseño Aplicado, estuvo a la vista de los visitantes durante 2013 y 2014 y, entre otras cosas, mostró juegos como Pac-Man (1980), Tetris (1984), Another World (1991), Myst (1993), SimCity 2000 (1994), The Sims (2000), EVE Online (2003) o Portal (2007).

A lo largo del tiempo se realizan múltiples versiones del juego snake en móviles y en consolas hoy en día es posible encontrar nuevas versiones en google app store o en línea además de opciones en tres dimensiones.

El objetivo de la realización del proyecto es adquirir conocimientos respecto a los sistemas de compilación de códigos en lenguaje c++ y al tener la oportunidad de realizar un proyecto de esta índole con ayuda de un juego me parece muy interesante y divertido.

2.Desarrollo

2.1 Descripción general del proyecto (idea con enfoque de emprendimiento)

El método pomodoro es un método en el que se busca tener una mayor concentración esto con ayuda de lapsos de descanso entre cada lapso de estudio o trabajo, siendo éstos mayores a los de descanso, así la mente estará mayormente enfocada en la actividad principal poniendo toda su atención en esta ya que durante el periodo de estudio o trabajo no se puede hacer uso de dispositivos tecnológicos y tampoco pueden existir distracciones lo que se busca con la realización de este proyecto es crear un juego que pueda ser utilizado durante los lapsos de descanso que además de activar la mente en cada usuario aumente la concentración y al mismo tiempo se utilice como un distractor que cumpla la función de relajar y desestresar es por eso que este proyecto se puede vender como una nueva actualización y función.

2.2 Algoritmo completo de la solución

Problema: seguir el algoritmo para obtener el juego snake

Entrada: Compilador en c++

Salida: Compilación en c++

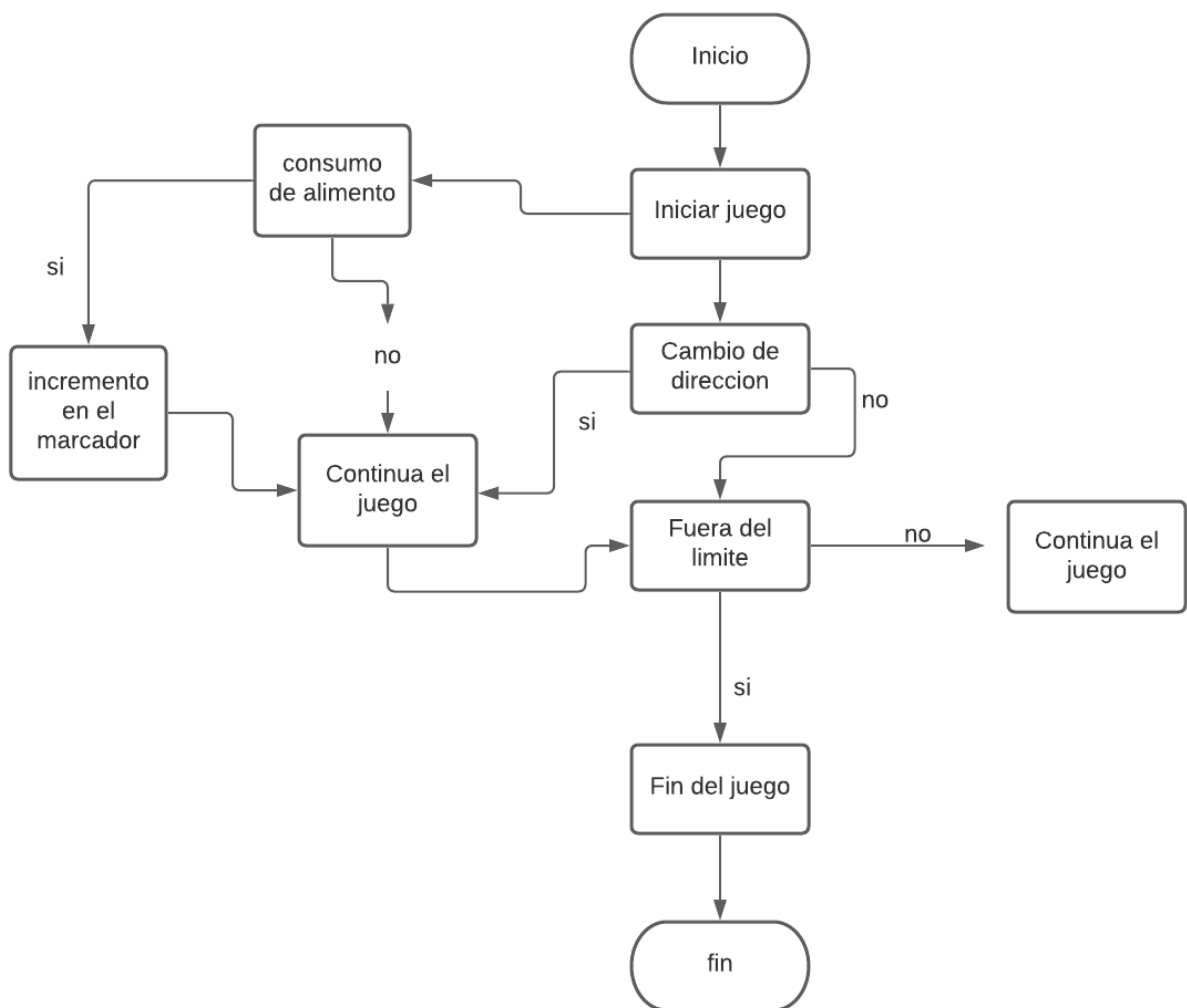
1. Empezamos delimitando un margen que servirá de límite permitido posteriormente (lados y esquinas)
2. Definimos el tamaño de la serpiente que tendrá al comienzo del juego
3. Definimos la variación de la posición de la comida
4. Relacionamos los movimientos de la serpiente (numero con teclas y direcciones)
5. Buscamos la forma de que cada parte de la serpiente siga a la primera y según lo que se consuma su tamaño aumentará
6. Seleccionamos la velocidad a la que la serpiente se moverá alrededor de nuestro panel de juego
7. Proponemos un valor para cada alimento que consuma
8. Se agrega un marcador que irá aumentando de puntaje de acuerdo a lo que la serpiente consuma

9. Marcamos el margen anterior como un obstáculo y si este es tocado la partida termina

10. Nos aseguramos de que cada una de estas partes corra de manera funcional y que si se llega a perder el marcador no siga corriendo y la partida termine de manera correcta

11. Has terminado de desarrollar el juego snake

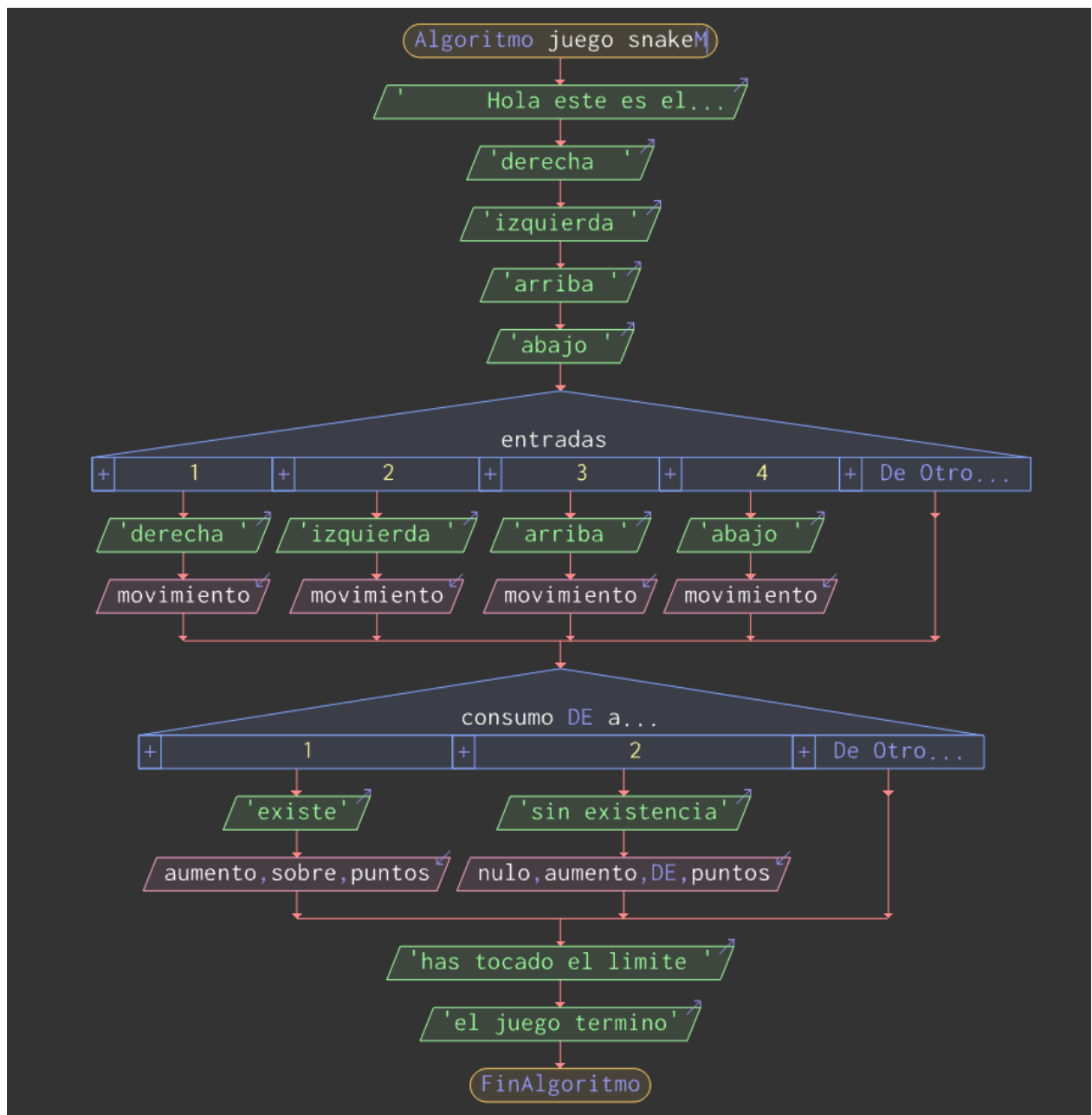
2.3 Diagrama de flujo y pseudocodigo



```

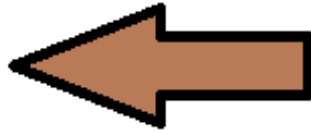
1  Algoritmo juegosnake
2      Escribir '      Hola este es el juego de PAOLA'
3      Escribir 'derecha '
4      Escribir 'izquierda '
5      Escribir 'arriba '
6      Escribir 'abajo '
7      Segun entradas Hacer
8          1:
9              Escribir 'derecha '
10             Leer movimiento
11          2:
12             Escribir 'izquierda '
13             Leer movimiento
14          3:
15             Escribir 'arriba '
16             Leer movimiento
17          4:
18             Escribir 'abajo '
19             Leer movimiento
20      FinSegun
21      Segun consumo DE alimentos Hacer
22          1:
23             Escribir 'existe'
24             Leer aumento,sobre,puntos
25          2:
26             Escribir 'sin existencia'
27             Leer nulo,aumento,DE,puntos
28      FinSegun
29      Escribir 'has tocado el limite '
30      Escribir 'el juego termino'
31  FinAlgoritmo
32

```



2.4 Código fuente

```
#include <windows.h>
#include <iostream>
#include <stdlib.h>
#include <conio.h>
```



se estan incluyendo las
declaraciones de otro fichero en
la compilacion

```
#define ARRIBA 72
#define IZQUIERDA 75
#define DERECHA 77
#define ABAJO 80
#define ESC 27
```



esta instruccion nos permite declarar
constantes de manera mas rapida

```
int cuerpo[200][2];
int n = 1;
int tam = 5;
int dir = 3;
int x = 10, y = 12;
int xc = 30, yc = 15;
int velocidad = 60, h=1;
char tecla;
int SCORE =0;
```



la palabra int define los tipos de datos enteros de
manera que int nombre_variable= valor y no es
necesario que la variable tenga un valor
predeterminado

```
void gotoxy(int x, int y)
```



void: se refiere a que la funcion declarada no devolvera
ningun valor

```
{
    HANDLE hCon;
    COORD dwPos;
```

void gotoxy (int x,inty): se refiere al movimiento del cursos
de la ventana de texto segun las coordenadas especificadas
por los argumentos x y y

```
    dwPos.X = x;
    dwPos.Y = y;
    hCon = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleCursorPosition(hCon,dwPos);
}
```

se utiliza ocultarcursos para que durante la compilacion la
flecha del cursor no aparezca

```
void OcultaCursor() {
    CONSOLE_CURSOR_INFO cci = {100, FALSE};
```

```
    SetConsoleCursorInfo(GetStdHandle(STD_OUTPUT_HANDLE), &cci);
}
```

```
void pintar(){
    for(int i=2; i < 78; i++){
        gotoxy (i, 3); printf ("%c", 205);
        gotoxy(i, 23); printf ("%c", 205);
    }
```

```
    for(int v=4; v < 23; v++){
        gotoxy (2,v); printf ("%c", 186);
        gotoxy(77,v); printf ("%c", 186);
    }
```

```
    gotoxy (2,3); printf ("%c", 201);
    gotoxy (2,23); printf ("%c", 200);
    gotoxy (77,3); printf ("%c", 187);
    gotoxy(77,23); printf ("%c", 188);
}
```



void pintar se utiliza en este codigo para que
se pinte un margen con las coordenadas que
se estan dando este margen tendra lineas
rectas y esquinas los numeros que aprecen
seguidamente de las coordenadas son los
valores de estas lineas


```
void guardar_posicion(){
    cuerpo[n][0] = x;
    cuerpo[n][1] = y;
    n++;
    if(n == tam) n = 1;
}
```



ubicacion del cuerpo
a principio del juego

```
void dibujar_cuerpo(){
    for(int i = 1; i < tam; i++){
        gotoxy(cuerpo[i][0] , cuerpo[i][1]); printf("*");
    }
}
```



dibujo de los puntos que
el cuerpo tendra con *

```
void borrar_cuerpo(){
    gotoxy(cuerpo[n][0] , cuerpo[n][1]); printf(" ");
}
```



cuando se borra el cuerpo
utilizar un espacio

```
void teclar(){
    if(kbhit()){
        tecla = getch();
        switch(tecla){
            case ARRIBA : if(dir != 2) dir = 1; break;
            case ABAJO : if(dir != 1) dir = 2; break;
            case DERECHA : if(dir != 4) dir = 3; break;
            case IZQUIERDA : if(dir != 3) dir = 4; break;
        }
    }
}
```



este sera nuestro control, los numeros
son las direcciones de las flechas del
teclado nuestro personaje se movera
segun los movimientos que realizemos

```
void comida()
{
    if(x == xc && y == yc)
    {
        xc = (rand() % 73) + 4;
        yc = (rand() % 19) + 4;
    }
}
```



la ubicacion de la comida sera
aleatoria

```
    tam++;
    SCORE+=10;
    gotoxy(xc, yc); printf("%c", 4);
}
```

```
bool game_over(){
    if(y == 3 || y == 23 || x == 2 || x == 77) return false;
    for(int j = tam - 1; j > 0; j--){
        if(cuerpo[j][0] == x && cuerpo[j][1] == y)
            return false;
    }
    return true;
}
```



aqui se marcan las
ubicaciones del limite y
como el juego termina
cuando nuestro personaje lo
toca

```
void puntos(){
    gotoxy (3,1);printf("SCORE %d",SCORE);
}

void cambiar_velocidad(){
```



se encuentra el codigo que
marca la obtencion de puntos

```
int main(){
    printf("    Hola este es el juego de PAOLA");
    OcultaCursor();
```



cuando el juego se compile
aparecera este mensaje

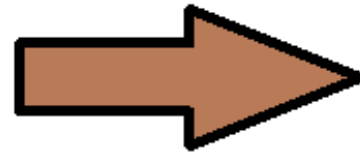
```
    pintar();
    gotoxy(xc, yc); printf("%c", 4);

    while(tecla != ESC && game_over())
    {
        borrar_cuerpo();
        guardar_posicion();
        dibujar_cuerpo();
```

```
        if(kbhit()){
            tecla=getch();
            switch(tecla){
                case ARRIBA:
                    if(dir!=2)
                        dir=1;
                    break;
                case ABAJO:
                    if(dir!=1)
                        dir=2;
                    break;
                case DERECHA:
                    if(dir!=4)
                        dir=3;
                    break;
                case IZQUIERDA:
                    if(dir!=3)
                        dir=4;
                    break;
            }
        }
        comida();
        puntos();
        cambiar_velocidad();
        teclear();
        teclear();
```



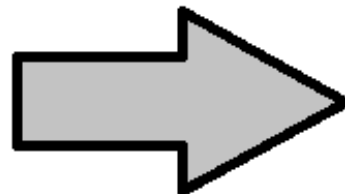
ubicacion del final del
cuerpo



ubicaciones que el cuerpo
seguira, para que todo el
cuerpo siga la misma direccion
y se mueva seguidamente de la
otra parte

```
        if(dir == 1) y--;
        if(dir == 2) y++;
        if(dir == 3) x++;
        if(dir == 4) x--;

        Sleep(velocidad);
    }
    pintar();
    return 0;
}
```

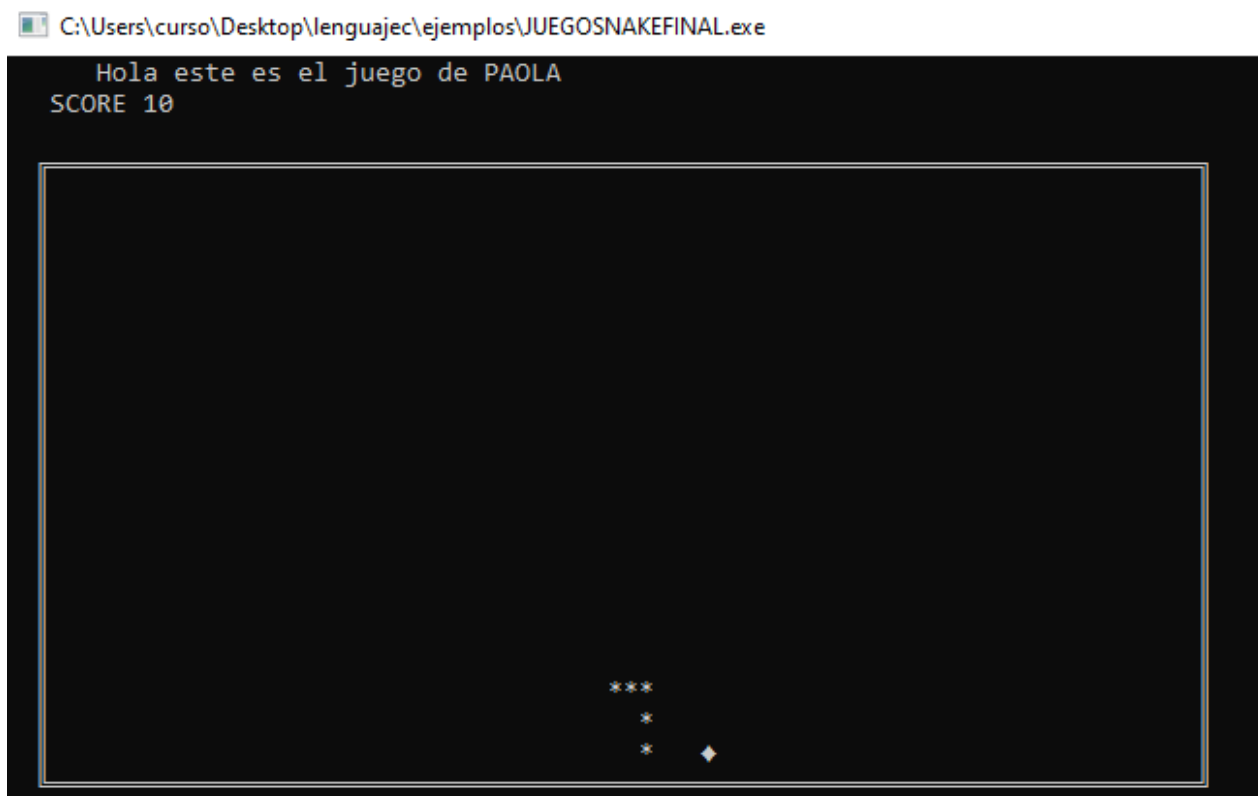
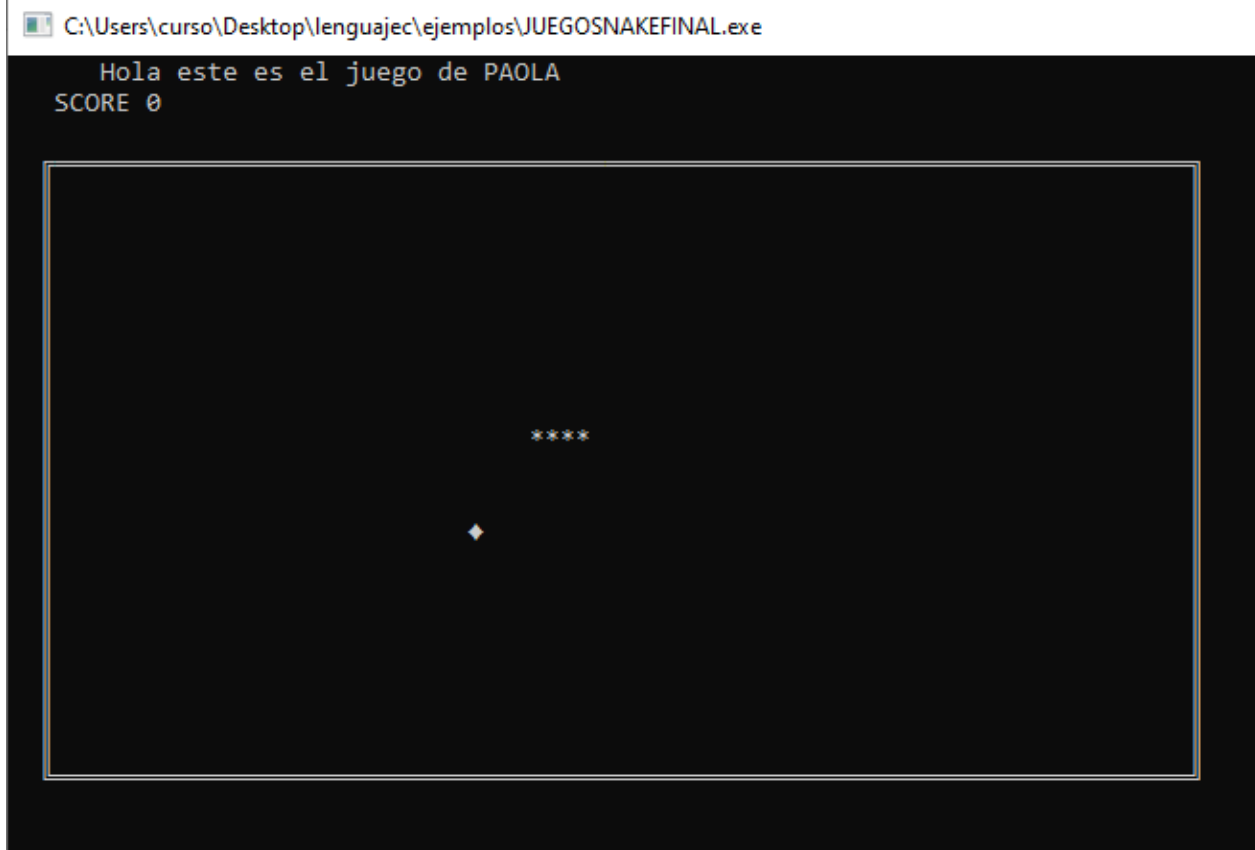


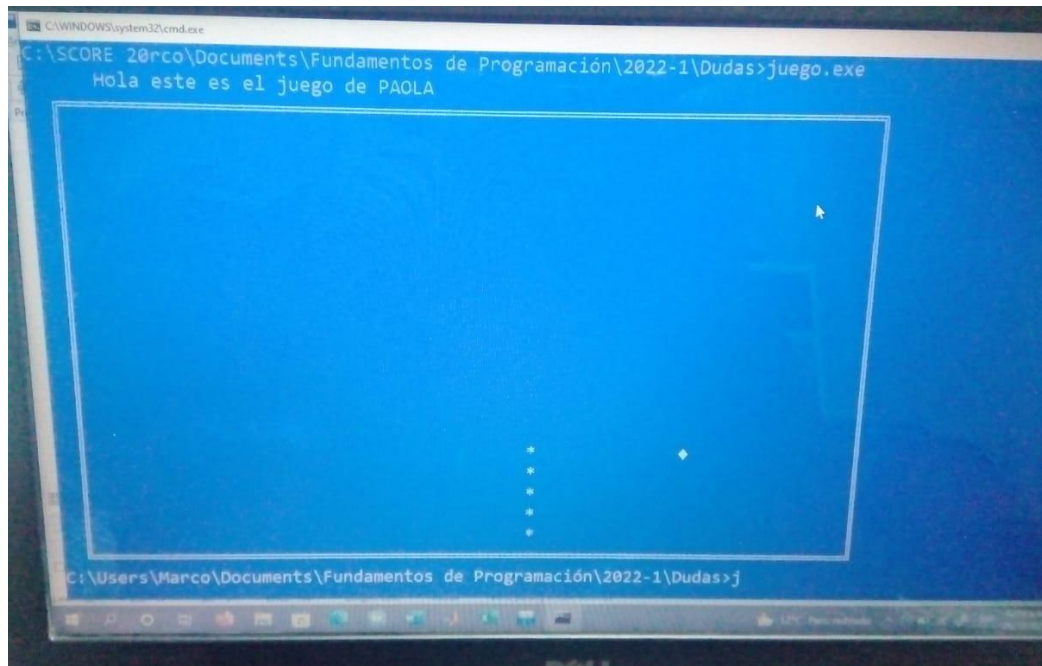
dir esta siendo utilizada como
direccion y no como accion



variables que permiten cambiar
la velocidad

3.1 Capturas de pantalla del funcionamiento de su proyecto





3.2 Tabla de recursos informáticos

software	hardware
recursos ofimáticos	computadora de escritorio marca DELL
páginas web de consulta	auriculares
devc++	camara
SO Android	teclado

3.3 Tabla de costos asociados al proyecto

diagrama	1000
algoritmo	2000
pseudocódigo	5000
código	8000
proyecto	10000
repositorio de github	1500

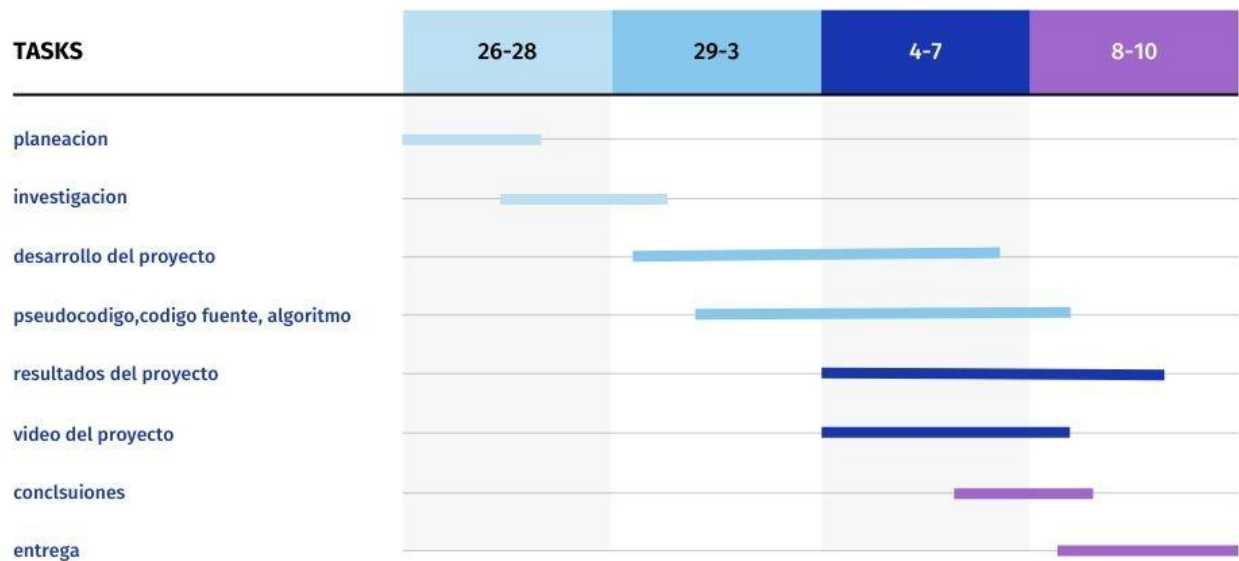
total 27500 pesos

3.4 Diagrama de Gantt para la elaboración del proyecto



juego de snake

DIAGRAMA DE GANTT

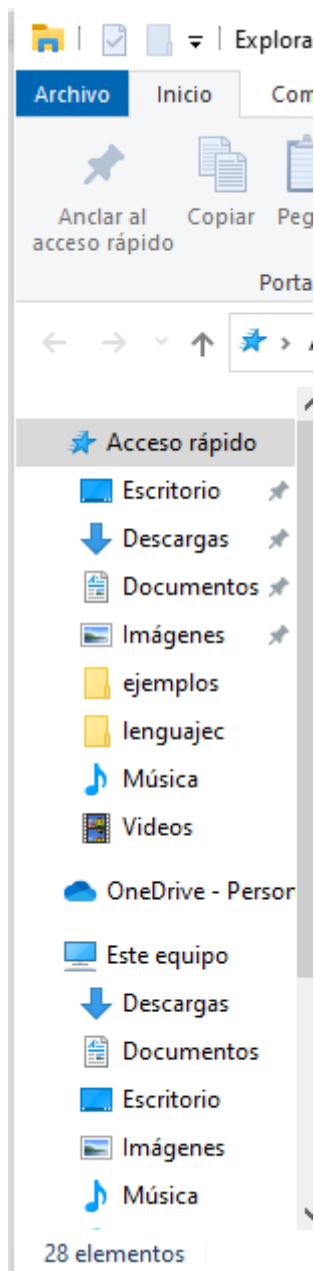


3.5 Manual de usuario

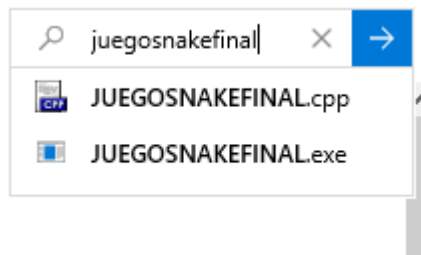
1. buscamos nuestros archivos y lo abrimos



2. seleccionamos nuestra carpeta ejemplos



3.en el buscador ponemos JUEGOFINALSNAKE

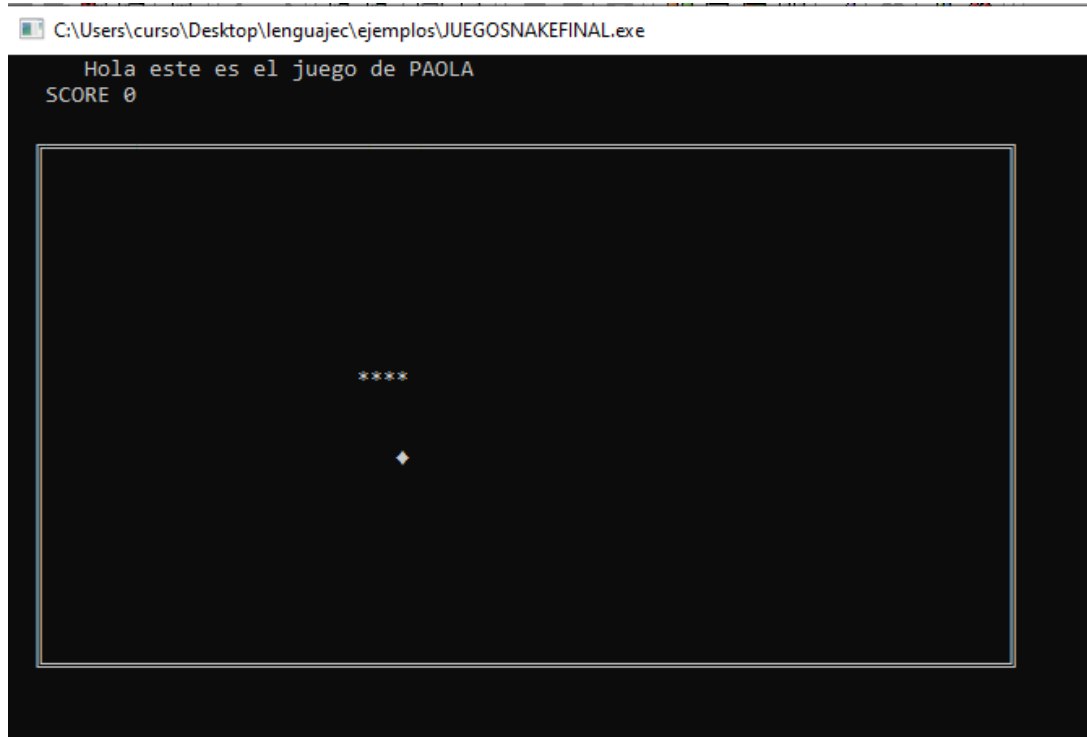


4.En este punto tenemos dos opciones el primero abrir el codigo realizar su compilación y comenzar a jugar o segundo abrir directamente nuestro juego y comenzar la partida

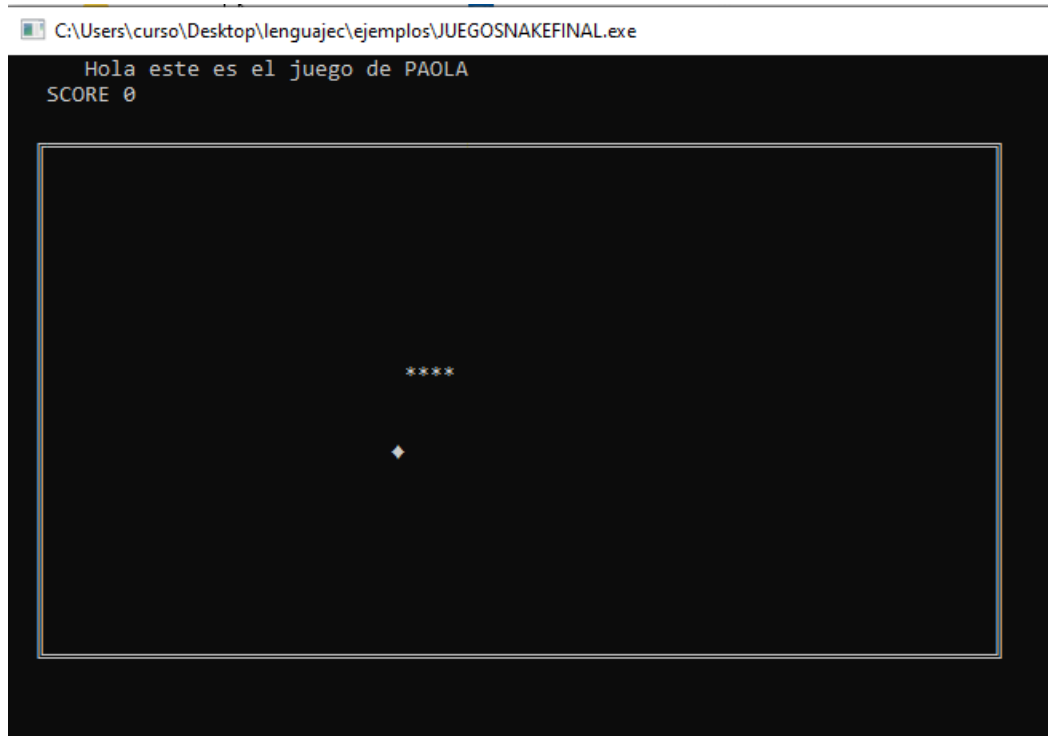
- si se escoge la primera opción al abrirlo lo tendremos de esta manera

```
enguajec\ejemplos\JUEGOSNAKEFINAL.cpp - Dev-C++ 5.11
Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
JUEGOSNAKEFINAL.cpp
1 #include <windows.h>
2 #include <iostream>
3 #include <stdlib.h>
4 #include <conio.h>
5
6 #define ARRIBA 72
7 #define IZQUIERDA 75
8 #define DERECHA 77
9 #define ABAJO 80
10 #define ESC 27
11
12 int cuerpo[200][2];
13 int n = 1;
14 int tam = 5;
15 int dir = 3;
16 int x = 10, y = 12;
17 int xc = 30, yc = 15;
18 int velocidad = 80, h=1;
19 char tecla;
20 int SCORE =0;
21
22 void gotoxy(int x, int y)
23 {
24     HANDLE hCon;
25     COORD dwPos;
26
27     dwPos.X = x;
28     dwPos.Y = y;
29     hCon = GetStdHandle(STD_OUTPUT_HANDLE);
30     SetConsoleCursorPosition(hCon,dwPos);
31 }
32 void OcultaCursor() {
33     CONSOLE_CURSOR_INFO cci = {100, FALSE};
```

- presionamos la pequeña libreta que se encuentra en la parte superior y es cuando se compila nuestro juego viéndose de la siguiente manera



- y listo puedes empezar a jugar
- y si decides abrir directamente el juego lo que te apareciera sera lo siguiente



4.Aplicación de los temas vistos en clase en proyecto final

En cuanto a los temas vistos durante la clase fundamentos de programación se utilizaron en su mayoría todos, algunos de estos son la correcta creación de un pseudocódigo para posteriormente realizar un correcto diagrama de flujo esto es importante debido a que es como un manual a seguir de los que posteriormente se realizará en el código que recopilaremos, en el código encontramos funciones que vimos durante las clases tales como el uso de las librerías, el uso de variables como int y char y palabras que nos dan para realizar acciones como while, else e if además del uso de la impresión o print en la pantalla que nos ayudó además aplicamos nuevos comandos como el uso de las coordenadas que permiten localizar el margen en un determinado lugar otro comando usado que utilizamos durante la realización de las prácticas es la creación de un ciclo que permite que nuestro personaje al moverse no pare hasta que se cambie la dirección que debe tomar y también aprendí uno nuevo siendo este el de la ubicación aleatoria de un símbolo en la pantalla de juego que nosotros estamos utilizando como un tipo de alimento para nuestro personaje el tema que no fue utilizado fue el uso de arreglos uni y multidimensionales esto debido a que nuestro programa no necesitaba la impresión de una matriz en su compilación.

4.1 Conclusiones personales

La programación es importante porque con ella nosotros podemos cambiar el mundo, lo innovamos. Además programando nosotros salimos adelante, es un bien para mí y para la sociedad, el programador es quien hace los programas que hacen facilitar la vida del ser humano, en un principio la programación me parecía que era aburrida pero todo esto por que no conocía como hacerlo no sabía lo fácil que podía llegar a ser además de entretenido y divertido es posible realizar todo lo que te imagines con la programación además de que esta facilita la vida.

Considero que es importante que se empiece a implementar el gusto por la programación desde pequeños para que no crezcan con ideas sobre el nivel de dificultad de esto.

5.Referencias

Programación de juegos - Snake en c++ #1 (Tutoriales Español y HD). (2012, 25 junio). [Video]. YouTube. <https://www.youtube.com/watch?v=gwfkWpjORRw>

Programación de juegos - Snake en c++ # 2 (Tutoriales Español). (2012, 27 junio). [Video]. YouTube. <https://www.youtube.com/watch?v=vBC3XEhd4Ic>

Programación de juegos - Snake en c++ # 3 (Tutoriales Español). (2012, 27 junio). [Video]. YouTube. <https://www.youtube.com/watch?v=aoBB7hUaPdw>

Programación de juegos - Snake en c++ #4 (Tutoriales Español). (2012b, julio 2). [Video]. YouTube. <https://www.youtube.com/watch?v=kaZzmwbRa2Y>

Restrepo, D. (2021, 8 noviembre). Mejores skills de programación con un juego snake en C#. Doctus / Web/Mobile Development.
<https://doctus.com.co/es/blog/mejora-las-skills-de-programacion-creando-un-juego-de-snake-en-c/>

T., & Perfil, V. T. M. (2013). Juego Snake en C++. blog.
<https://tillocodes.blogspot.com/2014/12/juego-snake-en-c.html>

UNAM. (2015). Laboratorios Salas A y B. Obtenido de <http://lcp02.fi-b.unam.mx/>