

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



FACULTAD DE INGENIERÍA

Fundamentos de programación 2022-1

GRUPO 04

Proyecto final
"Snake"

ESTUDIANTES

Basurto Díaz Carla Maricela

PROFESOR

Mat. Marco Antonio Martínez Quintana



Ciudad Universitaria, Cd. Mx., 2021

Índice.

1. [Resumen](#)
2. [Introducción](#)
3. [Desarrollo del proyecto](#)
4. [Algoritmo](#)
5. [Diagrama de flujo](#)
6. [Pseudocódigo](#)
7. [Resultados del proyecto](#)
8. [Tabla de recursos informáticos](#)
9. [Diagrama de Gantt](#)
10. [Links](#)
11. [Manual de usuario](#)
12. [Conclusiones](#)
13. [Referencias](#)

Resumen:

Este proyecto está enfocado a al antaño, en donde solíamos usar teléfonos celulares que solo tenían la capacidad de reflejar “juegos” de 8 bits, a lo largo de este archivo podremos ver un poco de la historia del juego y como su servidor recreo a su manera el juego de “snake” en el compilador DevCpp++ para el fácil desarrollo de esta, así como también se desarrolló el pseudocodigo para que cualquier persona ya sea a fin o no de la programación pueda entender que es lo que se realizo a lo largo de esto, por otra parte tenemos un diagrama de flujo de como funcionan los videojuegos y el diagrama de flujo de nuestro programa.

Aunado a esto tendremos nuestras pruebas acerca del proyecto, del como funcionan y se verán ilustradas, al igual de una tabla de costos asociados lo cual nos daría un aproximado de cuanto ingreso deberíamos recibir por la creación de un “sencillo” código como el que se presenta.

Con un diagrama sobre como el estudiante tuvo un avance en su proyecto, también se encontrarán los links hacia el repertorio y el canal de YT donde el alumno subirá su video.

Tambien se encuentra el manual de usuario sobre como se usará el “snake” explicado paso a paso, terminando con las conclusiones, referencias usadas y la explicación de algunos términos usados a lo largo de este archivo. 😊

Introducción

En este proyecto, he decidido optar por recrear el antiguo juego “Snake” un antiguo juego que todos en su infancia nos paramos a jugar en teléfonos celulares antiguos, el enfoque que le he dado ha sido mejorar la habilidad motora y reflejos en el uso de los periféricos como lo es el teclado, de cierta forma enfocadas a personas que juegan con las flechas del teclado.

A esto sabemos que Snake es un videojuego lanzado en la época de los 70 que ha mantenido su popularidad desde entonces coma convirtiéndose en un clásico. en 1998, él Snake obtuvo la audiencia masiva tras convertirse en un juego estándar pregrabado en los teléfonos Nokia.

Desarrollo

Varios juegos Snake se originaron con el juego arcade Blockade, desarrollado por Gremlin en 1976.² La primera versión conocida para microordenador, titulada Worm(Gusano), fue programada en 1978 por P. Trefonas desde Estados Unidos en los ordenadores TRS-80, y fue publicado por CLOAD magazine en el mismo año. Este fue seguido poco después por una versión del mismo autor para los ordenadores Commodore PET y Apple II. P. Trefonas escribió por primera vez una versión de Hustle para microordenadores en 1979, publicada posteriormente por CLOAD magazine.³ Esta fue luego desarrollada por Milton Bradley para los TI-99/4A en 1980.⁴

Algunas versiones mejor conocidas incluyen el ejemplo de Neopets, que es conocido como "Meerca Chase". Su versión revisada es conocida como "Meerca Chase II". durante un tiempo se incluyó en MS-DOS una variante popular denominada Nibbles.

En TimeSplitters 2 se incluyó, como juego oculto, una variante del snake controlado por un joystick analógico denominada Anaconda.

La versión incluida en el Nokia N70 y otros modelos posteriores de los móviles Nokia es una versión 3D, con metas por niveles. La versión de los Nokia también tiene una serpiente.

Desarrollo del proyecto.

Descripción de mi proyecto.

El juego snake, un programa que necesita conocimiento en algunas estructuras como lo pueden ser if, else al igual que matrices entre otras, pero entraremos en esos puntos más adelante.

La descripción del juego es que sea difícil con el tiempo y confuso para el jugador a diferencia del snake tradicional, utilizando las teclas wasd siendo una ayuda de reflejos (en mi opinión y mi objetivo) para personas jugadoras en dispositivos computacionales.

Además de ser parte de la nostalgia como lo pueden ser igual otros juegos como "pong" o "buscaminas" que seguramente no faltarán en los proyectos de demás compañeros.

En general es un programa sencillo, con “errores” a propósito para generar la confusión de el espectador o jugador (en este caso poniéndome a mí como público y otros tipos de jugadores.)

Algoritmo:

```
Snake.cpp
1  #include<iostream>
2  #include<windows.h>
3  #include<conio.h>
4  #include<stdlib.h>
5  #include<stdio.h>
6
7  #define ARRIBA 72
8  #define IZQUIERDA 75
9  #define DERECHA 77
10 #define ABAJO 80
11 #define ESC 27
12
13 int cuerpo [200][2];
14 int n=1;
15 int tam=5;
16 int x=10,y=12;
17 int dir=3;
18 int xc=30,yc=15;
19 int velocidad=150,h=1;
20 int score=0;
21 int w=500;
22
23 char tecla;
24
25
26
27 void gotoxy(int x, int y) {
28     HANDLE hCon;
29     COORD dwPos;
30
31     dwPos.X = x;
32     dwPos.Y = y;
33     hCon = GetStdHandle(STD_OUTPUT_HANDLE);
34     SetConsoleCursorPosition(hCon,dwPos);
35 }
```

```

35 }
36 void pintar(){
37     // Líneas horizontales
38     for(int i=2; i < 78; i++){
39         gotoxy(i, 3); printf("%c", 205); // los números hacen referencia al código ascii
40         gotoxy(i, 23); printf("%c", 205);
41     }
42     // Líneas verticales
43     for(int v=4; v < 23; v++){
44         gotoxy(2,v); printf("%c", 186);
45         gotoxy(77,v); printf("%c", 186);
46     }
47     // Esquinas
48     gotoxy(2,3); printf("%c", 201);
49     gotoxy(2,23); printf("%c", 200);
50     gotoxy(77,3); printf("%c", 187);
51     gotoxy(77,23); printf("%c", 188);
52 }
53
54 void guardar_posicion(){
55     cuerpo[n][0]=x;
56     cuerpo[n][1]=y;
57     n++;
58     if(n==tam)n=1;
59 }
60
61 void dibujar_cuerpo(){
62     for(int i=1; i<tam; i++){
63         gotoxy(cuerpo[i][0], cuerpo[i][1]); printf("%c", 2);
64     }
65 }
66
67 void borrar_cuerpo(){
68     gotoxy(cuerpo[n][0], cuerpo[n][1]); printf(" ");
69 }
70 }

```

```

71 void teclar(){
72     if(kbhit()){
73         tecla=getch();
74         switch(tecla){
75             case ARRIBA:
76                 if(dir!=2){
77                     dir=1;
78                     break;
79             case ABAJO:
80                 if(dir!=1){
81                     dir=2;
82                     break;
83             case DERECHA:
84                 if(dir!=4){
85                     dir=3;
86                     break;
87             case IZQUIERDA:
88                 if(dir!=3){
89                     dir=4;
90                     break;
91             }
92         }
93     }
94 }
95
96 void cambiar_velocidad(){
97     if(score==h*50){
98         velocidad-=15;
99         h++;
100     }
101 }
102
103 void comida(){
104     if(x==xc&&y==yc){
105         xc=(rand()%73)+4;
106         yc=(rand()%19)+4;
107         tam++;
108         score+=10;
109         gotoxy(xc,yc); printf("%c", 4);
110     }
111 }
112
113 }

```

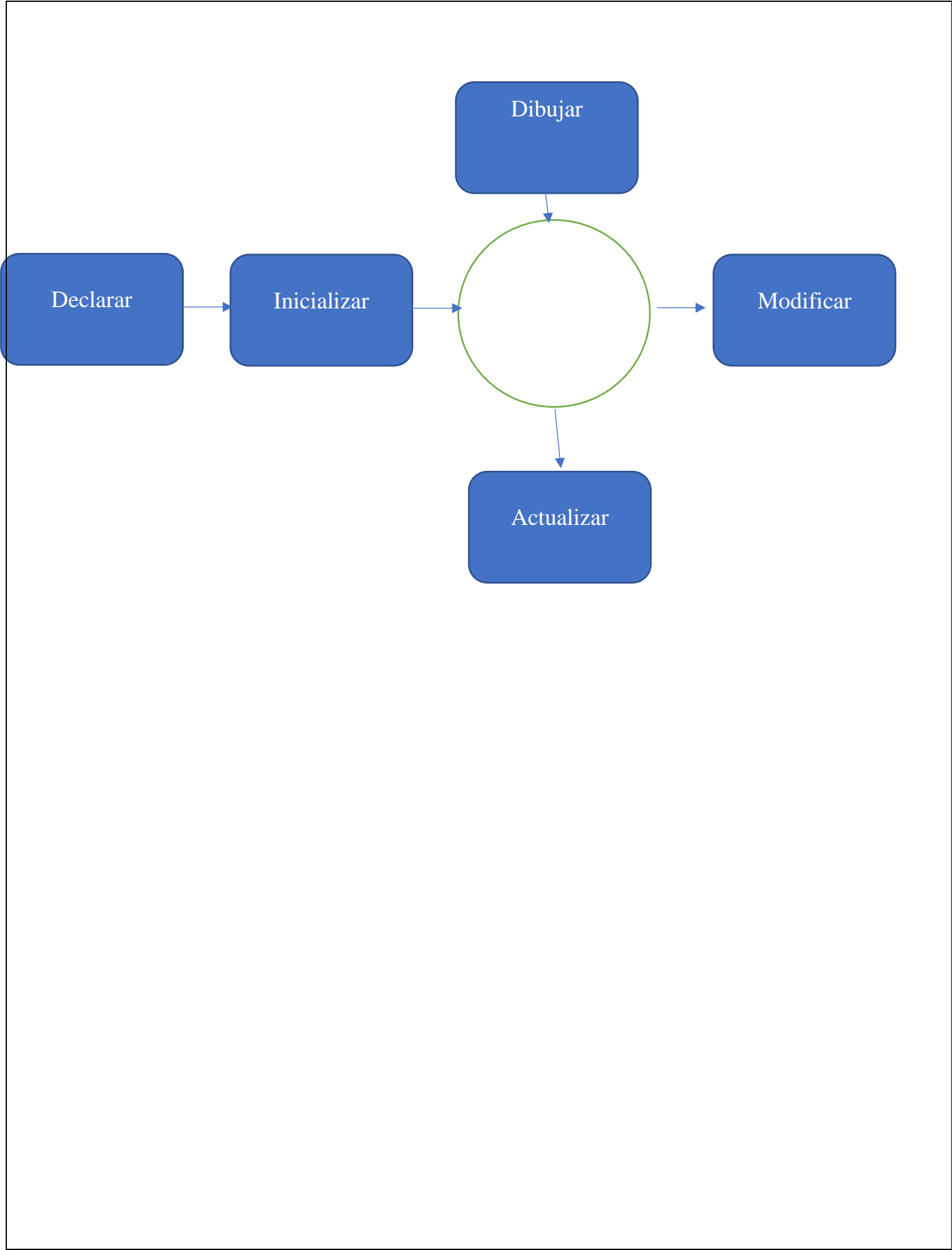
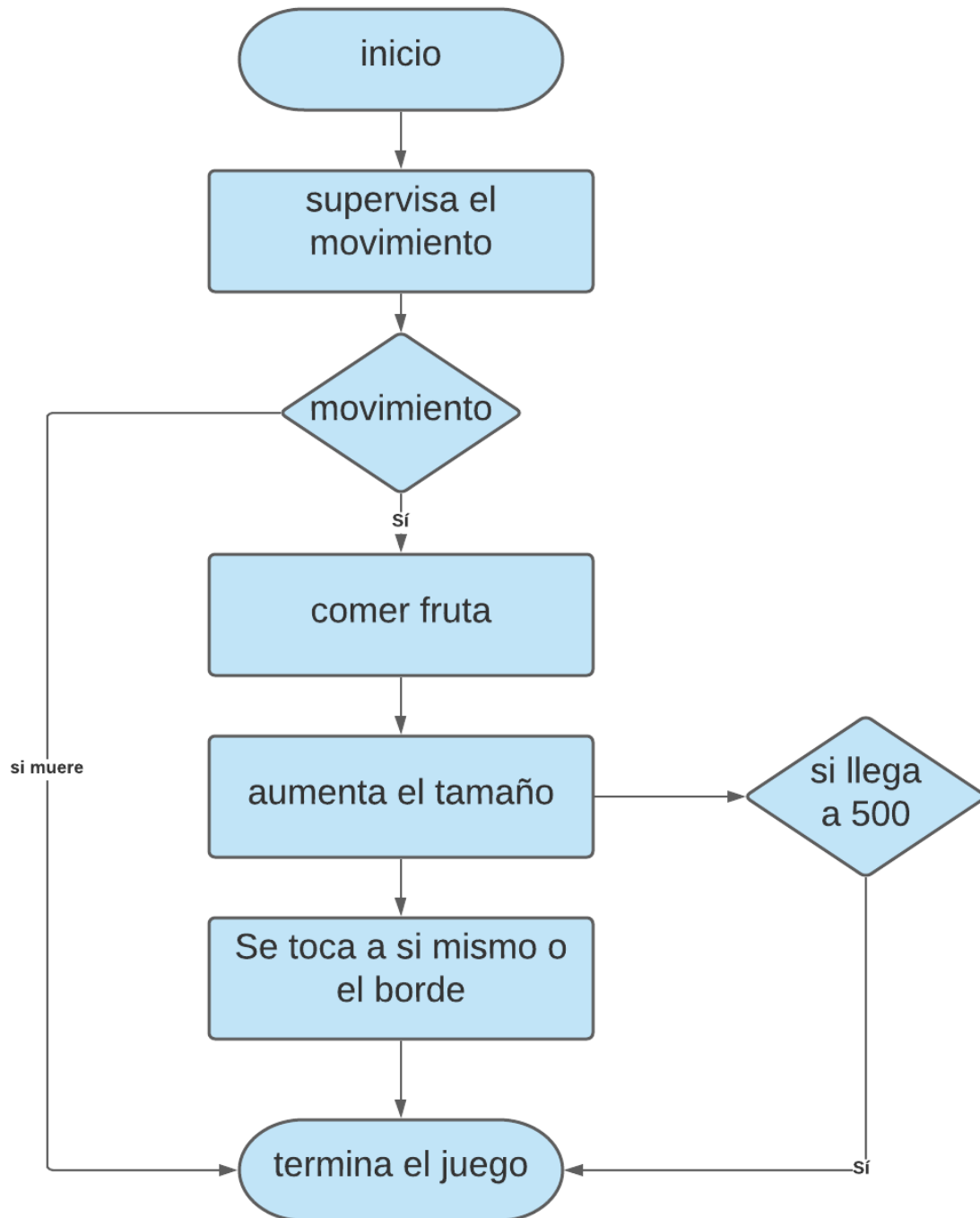



Diagrama de flujo.



Pseudocodigo.

Este programa tiene como propósito ejecutar el juego “snake”.

→Empezamos con las librerías a usar.

→Definimos los valores que vamos a usar.

→Declaramos la función principal para los valores del cuerpo, el tamaño, la velocidad coma la puntuación y a cuántos puntos vamos a ganar (en este caso 500).

→Creamos un **Void Gotoxy**, que mover el cursor de la ventana del texto. Según las coordenadas indicadas en este caso x, y.

→Creamos otro **void** pero esta vez para pintar el delimitado del cuadro, en este caso con una matriz, en este caso con un código acci por mera estética.

→Pintamos /ponemos las líneas verticales y horizontales incluyendo las esquinas.

→Hacemos otro **void** para mantener la posición de nuestra viborita en la matriz.

→Siguiendo con otro **void** para dibujar el cuerpo y el tamaño inicial que tendrá.

→Seguimos con otro **void** para borrar el cuerpo al finalizar.

→Creamos un **void** para dar el orden que tendrán las teclas que usaremos para el movimiento de la serpiente, dando un break después de cada declaración.

→Hacemos otro **void** para que cada vez que se llegue X cantidad de puntos aumente la velocidad.

→**Void** para la comida, dando localizaciones aleatorias en cada una y diciendo cuántos puntos aumentará al consumir una fruta.

→Un **bool** para la lógica, en donde si es falso terminará el juego, mientras que si sigue siendo verdadero el juego continuará.

→Un **void** para el contador de puntos.

→Un bool para la declaración de una victoria si los puntos llegan X cantidad.

→Hacemos un **void** para la pantalla de inicio imprimiendo el mensaje, en este caso las instrucciones y una vez continuado se limpiará la pantalla.

→**Void** para el juego que recibirá todas las declaraciones anteriores para así después.

→**If** si perdemos, imprimirá un “game over” y limpiará la pantalla o.

→**If** si ganamos imprimirá un “win” y limpiará la pantalla.

→**Int main** para las funciones título, pintar, gotoxy y el game.

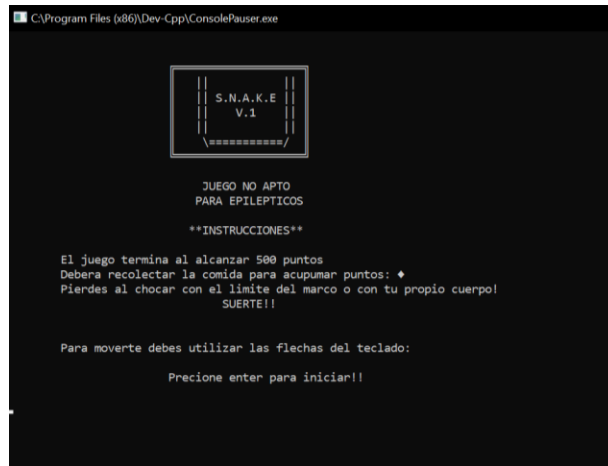
→**System null**

→**Fin/return.**

Resultados del Proyecto.

Capturas de pantalla:

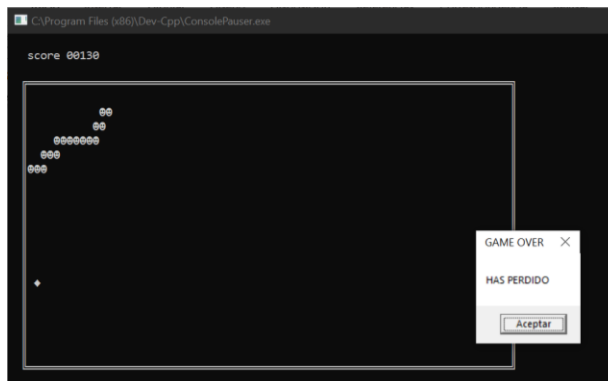
Pantalla de inicio.



Mientras jugamos:



Cuando perdemos



Cuando ganamos:

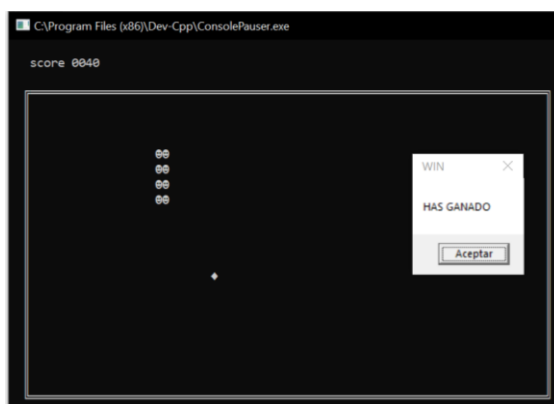


Tabla de recursos informáticos.

Programa	Uso (gratis/paga)	Uso
DecCpp++	Descarga gratuita en la página del laboratorio	Editor de texto y de código fuente libre con soporte para varios lenguajes de programación.
Windows (a partir de la 7ma edición)	Programa que viene incluido con cualquier pc.	
También puede usarse IOS en caso de tener MAC	Programa que viene incluido con cualquier dispositivo electrónico de apple	

Diagrama de Gantt

Iniciado	Proceso	Acabado

[illegible]

Tabla de costos.

Programa usado	Actividad	Tiempo empleado	costo
DevCpp++	Creación del Código	5hrs	\$141 x h = \$750
Lucidchart	Creación del diagrama de flujo	10 minutos	\$23
-	Creación del pseudocodigo	1 hora	\$141
			Total: \$914

El costo total sería de \$914, pero lo dejaré como un programa de uso libre así que tendría un costo de cero pesos. 😊

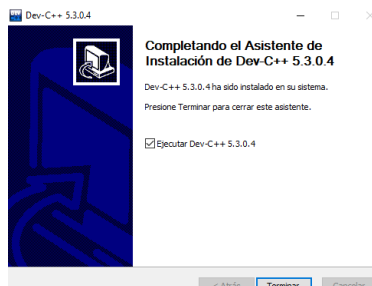
Canal de yt: [Carla Basurto Díaz - YouTube](#)

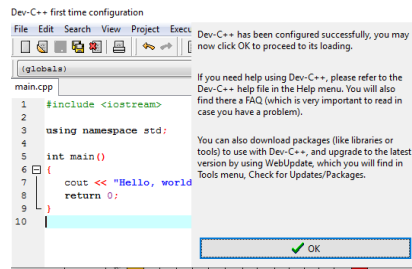
Repositorio de GitHub: [Yu3san/LabFundamentosdeProgramaci-n: Aquí subo mis trabajos ¿Esperabas algo más?, lo siento. \(github.com\)](#)

Manual de usuario

¿Cómo usar Snake?

Bueno, tenemos que empezar por tener un programa que pueda leer el archivo, en este caso sino tenemos uno podemos instalar el DevCpp++, una vez descargado de la pagina [Laboratorio Salas A y B \(unam.mx\)](#) lo instalamos.

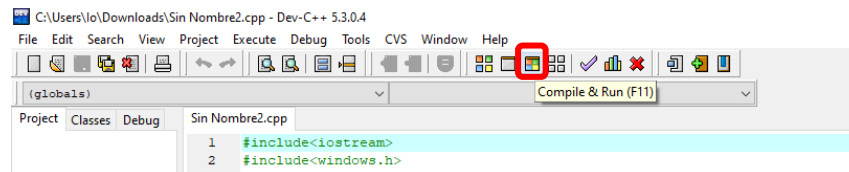
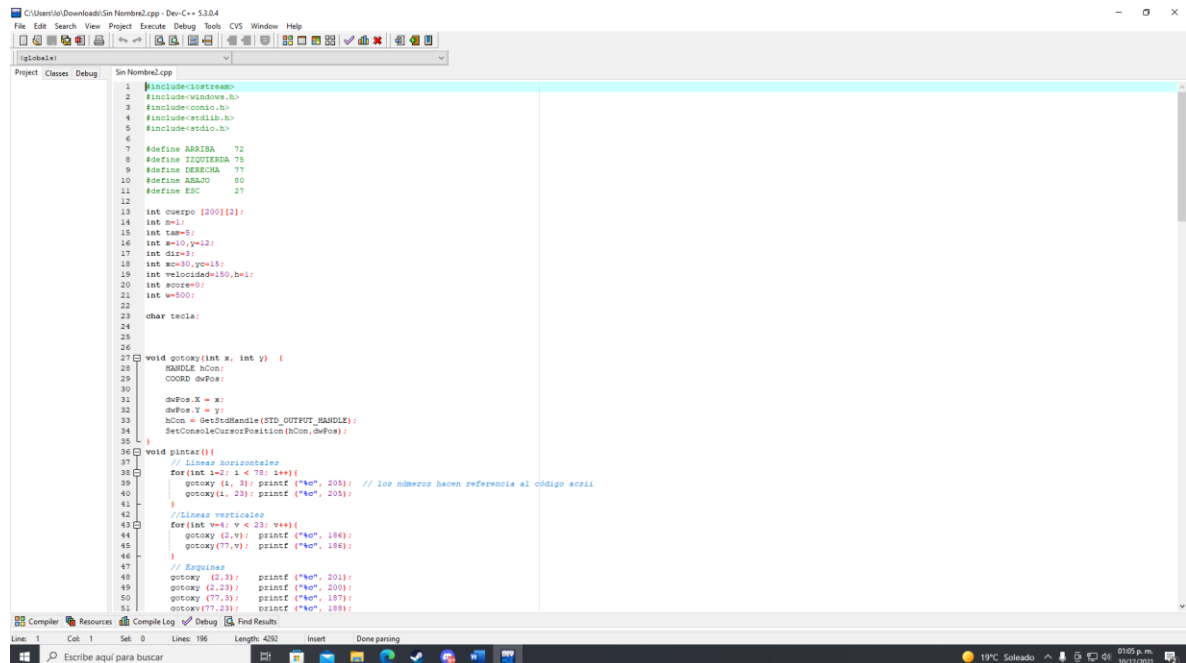




Descargamos el archivo de “snake” que estará disponible en la siguiente liga:

<https://drive.google.com/drive/folders/1LWbgeq9HHGicFcHIBYhqFY-Up6T3VaTy?usp=sharing>

Una vez lo abrimos y nos aparecerá tal que así:

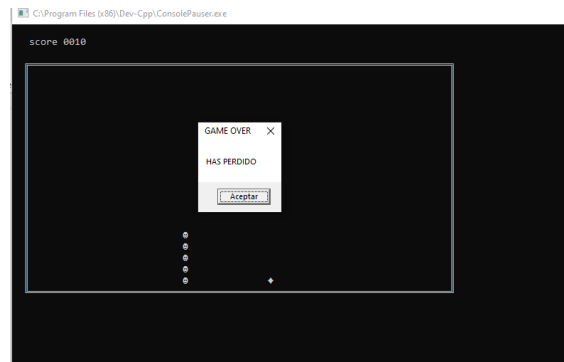


Podemos darle a ese botón o simplemente pulsar F11 (sí estas en laptop presiona FN + F11)

Nos aparecerá el siguiente cuadro:



Seguimos la instrucción de presionar una tecla cualquiera y podremos empezar a jugar, una vez perdemos solo presionamos otra tecla y el juego se cerrará.



Sí llegas a 500 puntos ganarás el juego, aunque sinceramente es algo difícil 😊

Conclusiones

Con este proyecto pude aplicar los conocimientos adquiridos a lo largo del semestre, claro con ayuda de algunos manuales y videos de yt como retroalimentación con las sentencias switch que me permite ejecutar varios bloques de instrucciones según el valor de la variable, o void para indicar que la función no devolverá el valor/no tiene parámetros, la ausencia de un valor básicamente, entre otras como do while o if, aun así siento satisfacción con el trabajo que hice y con los conocimientos que me proporcionó mi profesor.

No solo este proyecto me permitió desarrollar mis conocimientos en lenguaje c, sino que a lo largo del semestre fue grato ir desarrollando ser autodidacta para seguir investigando cada una de las practicas y las cosas que podría aplicar en mi proyecto, espero que este juego haya sido de su agrado, aunque fue un poco problemático, además de que a lo largo del semestre se me hizo muchísimo más cómodo y practico usar el DevCpp++ porque la compilación es muchísimo más sencilla.

Referencias

Metodología de la programación. Osvaldo Cairó, tercera edición, México D.F., Alfaomega 2005.

Metodología de la programación a través de pseudocódigo. Miguel Ángel Rodríguez Almeida, primera edición, McGraw Hill

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.

Bloque 3.3 Juegos Simples en C: Snake. (2013, 15 noviembre). [Vídeo]. YouTube.

<https://www.youtube.com/watch?v=F-daVcDp2iI&list=LL&index=1>