***ÍNDICE***

ÍNDICE…………………………………………………………………………………………… 1

INTRODUCCION…………………………………………………………………………. 2

OBJETIVOS………………………………………………………………………………….. 2

DESARROLLO………………………………………………………………………………… 3

RESULTADOS……………………………………………………………………………….. 10

CONCLUSIONES…………………………………………………………………………… 15

BIBLIOGRAFIA…………………………………………………………………………….. 15

introduccion

Las consolas de juego han sido una constante en la sociedad moderna. Desde el atari, pasando por el sega, hasta llegar al PS3, el x-box360 y aparatos tan novedosos como el wii o el kinect.

Son aparatos muy útiles para la fortificacion de habilidades motrices, logicas, de tiempo de reaccion, de analisis, de solucion de problemas; aparte de ser un medio de liberar el estrés del día a día.

Las consolas de juego, lejos de lo que se puede pensar, no estan dirigidas exclusivamente para los jovenes sino para todo publico en general.

Este proyecto pretende emular una consola de juegos clasica, para la PC. Los juegos utilizados son viejos conocidos del mundo de los videojuegos, sin embargo cada uno tiene nuestro toque personal.

A lo largo de este escrito, mostraremos varios puntos importantes sobre el funcionamiento del programa asi como las complicaciones que se nos presentaron en su creación. Asi mismo, se reflejan los resultados obtenidos y las conclusiones a las cuales llegamos.

objetivos

* Diseño de una interfaz que facilite elegir entre varios usuarios para la utilizacion de la consola de juegos. Asi mismo dicha interfaz tendra que guardar y leer dichos datos para llevar un registro.
* Diseñar una interfaz que te permita elegir entre diferentes juegos, estos deberán de estar en relacion al perfil seleccionado.
* Crear tres juegos útiles, interactivos y atractivos.

desarrollo

La consola de juegos esta compuesto de diez archivos de texto, tres archivos ejecutables, un archivo de proyecto, cinco archivos de cabecera y tres archivos de código fuente.

**Archivos de texto:**

En estos archivos se guarda la informacion, nueve de ellos estan destinados a guardar la informacion de los nueve perfiles de jugadores y el ultimo guarda el usuario activo.

**Archivos ejecutables:**

En estos archivos se encuentran los progrmas ya compilados. Uno de ellos es el juego “NAVE”, que se ejecuta en la pantalla consola. El segundo corresponde al juego “TETRIS”, que igualmente se ejecuta en la pantalla consola. El ultimo corresponde a los menus y al juego “SIGUEME”, que se ejecuta en la pantalla grafica. El tercer ejecutable tiene dentro de su codigo la referencia para abrir los otros dos ejecutables.

*Asi se mandan llamar a los programas de juegos externos:*

* WinExec("tetris.exe",SW\_SHOW);
* WinExec("nave.exe",SW\_SHOW);

**Archivos de proyecto:**

Es un proyecto gráfico que contiene cuatro de los archivos de cabecera y un archivo de código fuente (main) que corresponde a el tercer ejecutable.

**Archivos de cabecera:**

* *menu.h*

En este se concentran los graficos y la logica del menu principal. Posee varias funciones, generalmente de botones o animaciones. La funcion mas importante de esta librería es “*void menu\_principal(int perfil)”.*

A continuacion se coloca el código de un botón, este codigo es similar en cualquier botón ya sea de esta librería o de las librerias “sigueme.h” y “usuarios.h”.

void boton\_nave(int x,int y,int estado){

setcolor(0);

switch (estado){

case 0: //APAGADO

setbkcolor(9);

setfillstyle(1,9);

break;

case 1: //PRENDIDO

setbkcolor(COLOR(238,134,0));

setfillstyle(1,COLOR(238,134,0));

break;

default:

cout<<"Este mensaje nunca deberia aparecer (boton\_nave)" <<endl;

}

bar3d(x,y,x+200,y+50,5,1);

settextstyle(10,0,3);

outtextxy(x+100-textwidth("Nave")/2,y+10,"Nave");

}

* *sigueme.h*

En esta librería se encuentra el menu del juego sigueme, asi como la logica del juego. Se muestra a continuacion la forma en la que se genera la siguiente posición, como se guardan en una matriz y se reproduce en pantalla (el vector “p” es de 200 y la matriz “posiciones” 9x200).

p[nivel]=1+rand()%9; //Genera la siguiente posicion.

posicion[nivel][p[nivel]-1]=1; //Guarda la posicion.

//Reproduce la secuencia hasta el momento.

for(int i=0;i<=nivel;i++){

boton(270,420,1,posicion[i][0]); //Los primeros dos parametros corresponden a las coordenadas

boton(400,420,2,posicion[i][1]); //El tercer parámetro corresponde al numero y nota.

boton(530,420,3,posicion[i][2]); //El cuarto parametro corresponde a el estado (0/1).

boton(270,290,4,posicion[i][3]);

boton(400,290,5,posicion[i][4]);

boton(530,290,6,posicion[i][5]);

boton(270,160,7,posicion[i][6]);

boton(400,160,8,posicion[i][7]);

boton(530,160,9,posicion[i][8]);

}

* *sonidos.h*

Contiene las definiciones de las notas musicales (escritas en cifrado y correspondientes a su frecuencia en Hertz) y de la duracion de cada tipo de nota.

Destaca la precencia de la funcion nota que recibe parametros de nota, tempo y duracion correspondientes a la notacion de cualquier partitura musical. Cabe mencionar que hay muchas intenciones musicales que no se reflejan en esta funcion puesto a que no son necesarias para nuestros fines. Esta funcion es usada en el juego sigueme para generar los sonidos de cada numero.

void nota(int nota,int tempo,int duracion){

if(duracion==d2|duracion==d3|duracion==d4|duracion==d8|duracion==d16){

Beep(nota,(600000/(tempo\*duracion)));

}

else{

if(duracion==p1){Beep(nota,(6000/tempo)\*duracion);}

else{Beep(nota,(60000/tempo)\*duracion);}

}

}

* *usuarios.h*

Es el archivo mas complejo y extenso ya que este no solo genera el menu de usuarios sino tambien todos los metodos de una clase llamada usuario. Varios de estos metodos leen y/o guardan datos de los archivos.

*class usuario {*

*//Datos:*

*char \*nombre;*

*char \*archivo;*

*int total;*

*//Juegos:*

*int tetris[8];*

*int sigueme[8];*

*int nave[8];*

*public:*

*usuario(char \*file);*

*void actualizar();*

*void dibujar(int x,int y);*

*void dibujar\_nombre(int x,int y,int estado);*

*void cambiar\_nombre(char \*n);*

*int puntos(int juego);*

*int partidas(int juego);*

*void fin(int juego,int pts);*

*int logros(int juego);*

*void logros(int juego,int L);*

*};*

Presentamos el fragmento de código correspondiente a la clase usuario (izquierda) y el fragmento de la funcion constructora (abajo) que lee el archivo de texto y guarda la informacion linea a linea en cadenas de caracteres para su posterior utilización.

file=fopen(archivo,"r");

if (file!=NULL){

while(!feof(file)){

fgets(info[a],50,file);

fflush(stdin);

if(!feof(file)){info[a][strlen(info[a])-1]='\0';}

a++;

}

}

fclose(file);

A continuacion se presenta el fragmento de la funcion actualizar que guarda los datos del usuario en el archivo correspondiente.

file=fopen(archivo,"w");

if (file!=NULL){

fprintf(file,"----------------------------------------\n");

fprintf(file,"NOMBRE\n");

fputs(info[2],file);

fprintf(file,"----------------------------------------\n");

fprintf(file,"PUNTACION TOTAL:\n");

fprintf(file,"%i\n",total);

fprintf(file,"TETRIS:\n");

fprintf(file,"%i\n",tetris[0]);

fprintf(file,"%i\n",tetris[1]);

fprintf(file,"%i%i%i%i%i\n",tetris[2],tetris[3],tetris[4],tetris[5],tetris[6]);

fprintf(file,"SIGUEME:\n");

fprintf(file,"%i\n",sigueme[0]);

fprintf(file,"%i\n",sigueme[1]);

fprintf(file,"%i%i%i%i%i\n",sigueme[2],sigueme[3],sigueme[4],sigueme[5],sigueme[6]);

fprintf(file,"NAVE:\n");

fprintf(file,"%i\n",nave[0]);

fprintf(file,"%i\n",nave[1]);

fprintf(file,"%i%i%i%i%i\n",nave[2],nave[3],nave[4],nave[5],nave[6]);

//Agregar nuevos juegos:

fprintf(file,"----------------------------------------");

}

else {cout<<"Error al abrir el archivo" <<endl;}

fclose(file);

* *usuario\_consola.h*

Esta librería es la unica que no esta contenida en el proyecto gráfico. Se creó por la necesidad de guardar los datos de los usuarios al usar juegos que no utilizan a la interfaz gráfica. La clase usuario consola es la principal y esta hereda a usuario\_tetris y usuario\_nave.

*class usuario\_consola {*

*protected:*

*//Datos:*

*char \*nombre;*

*char \*archivo;*

*int total;*

*//Juegos:*

*int tetris[8];*

*int sigueme[8];*

*int nave[8];*

*public:*

*usuario\_consola();*

*usuario\_consola(char \*file);*

*void actualizar();*

*};*

*class usuario\_nave:public usuario\_consola {*

*public:*

*usuario\_nave();*

*usuario\_nave(char \*file);*

*int puntos();*

*void fin(int pts);*

*int logros();*

*void logros(int L);*

*};*

**Arcihivos de código fuente:**

*#include "usuarios.h"*

*#include "menu.h"*

*int main(){*

*int perfil;*

*initwindow(900,600);*

*perfil=menu\_usuarios();*

*menu\_principal(perfil);*

*closegraph();*

*return 0;*

*}*

Main.cpp

* *main.cpp*

Inicia la pantalla grafica, manda llamar a los metodos principales de cada una de las librerias y cierra la pantalla grafica.

* *tetris.cpp*

Tres temas de gran relevancia para el desarrollo del “tetris” fueron sentencias de control, funciones y matrices.

Las sentencias de control fueron sumamente importantes para el desarrollo del “tetris” ya que debido a que se requiere la simulación del movimiento de las figuras era necesario el imprimirlas en una posición inicial, borrarlas, reimprimirlas en la próxima posición y continuar sucesivamente con este paso, para esta simulación se utilizo un ciclo for; también era necesario que las piezas detuvieran su movimiento cuando se encontraran en la parte más alta o se encontraran con otra figura previamente puesta; para este efecto se utilizo la sentencia if; finalmente para la aplicación del menú, se utilizo la sentencia do-while.

Funciones: Se utilizaron para evitar la repetición de líneas de código como ejemplo al especificar las coordenadas de cada figura; en total son 19 figuras, 7 bases y el resto se obtienen al girar las anteriores, un ejemplo el cuadrado:

//FIGURAS:

void cuadrado(int r,int s){

r1=r;s1=s;

r2=r+2;s2=s;

r3=r;s3=s+1;

r4=r+2;s4=s+1;

}

Otra función de gran relevancia fue la llamada “gotoxy” ya que gracias a esta es posible desplazarse por la pantalla e imprimir en el punto exacto que se desea hacer:

void gotoxy(int x, int y)

{

HANDLE hCon;

COORD dwPos;

dwPos.X = x;

dwPos.Y =y;

hCon= GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleCursorPosition(hCon,dwPos);

}

gotoxy(r1,s1);puts(p1);

Matrices básicamente se utilizo para la comprobación al momento de eliminar una fila que se encuentre llena, y para poner los límites de desplazamiento de las figuras en vista de las piezas previamente colocadas:

void eliminar(int nf)

{

suma=0;

for(int k=0;k<=10;k++){

suma=suma+mcomprobar[nf][k];

}

if(suma==11){

for(int j=nf;j<=36;j++){

for(int k=0;k<=10;k++){

mcomprobar[j][k]=mcomprobar[j+1][k];

}

}

for(int j=nf;j<=36;j++){

for(int k=0;k<=10;k++){

gotoxy((k+1)\*2,j);puts(b1);

if(mcomprobar[j][k]==1){

gotoxy((k+1)\*2,j);puts(p1);

}

}

}

puntos=puntos+11;

}

}

El tetris cuenta con movimientos a través de las teclas de dirección () y la tecla enter (); esto se hizo mediante interrupciones y select case:

#define ARRIBA 72

#define IZQUIERDA 75

#define ABAJO 80

#define DERECHA 77

#define ENTER 13

if(kbhit())

{

unsigned char tecla=getch();

switch(tecla)

{

case IZQUIERDA:

…

* *nave.cpp*

En el juego de la nave los asteroides caen sobre el eje de las y, se utiliza un random para que caigan en forma aleatoria en la pantalla, tamien se comprueba que el eje de cordenadas no sea igual al de la nave, en dado caso el asteroide golpea al la nave.

gotoxy(x,y); printf("%c",2);

gotoxy(xx,yy); printf("%c",2);

gotoxy(x1,y1); printf("%c",2);

gotoxy(x2,y2); printf("%c",2);

Sleep(70-(nivel\*2));

gotoxy(x,y); printf(" ");

gotoxy(xx,yy); printf(" ");

gotoxy(x1,y1); printf(" ");

gotoxy(x2,y2); printf(" ");

if(y>20)

{

y=4;

x=(rand()%70)+6;

if(y==4)

condicion= false;

}

if(yy>20)

{

yy=4;

xx=(rand()%70)+6;

}

if(y1>20)

{

y1=4;

x1=(rand()%70)+6;

}

if(y2>20)

{

y2=4;

x2=(rand()%70)+6;

}

//golpes de asteroides

if( (x > ix && x<ix+6 && y==iy-1) || (xx>ix && xx<ix+6 && yy==iy-1)||

(x1>ix && x1<ix+6 && y1==iy-1)||(x2>ix && x2<ix+6 && y2==iy-1))

{

corazones--;

Barra\_salud(corazones);

printf("\a");//sonido cada golpe

}

resultados

Se han superado las espectativas en la mayoria de los juegos. Estos funcionan en su totalidad. Asi mismo se han obtenido buenos resultados en la utilizacion de archivos y en la generacion de graficos.

A continuacion presentamos varias impresiones de pantalla con los resultados obtenidos.

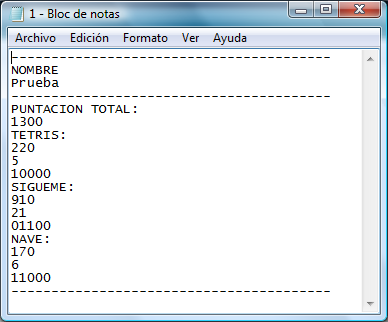
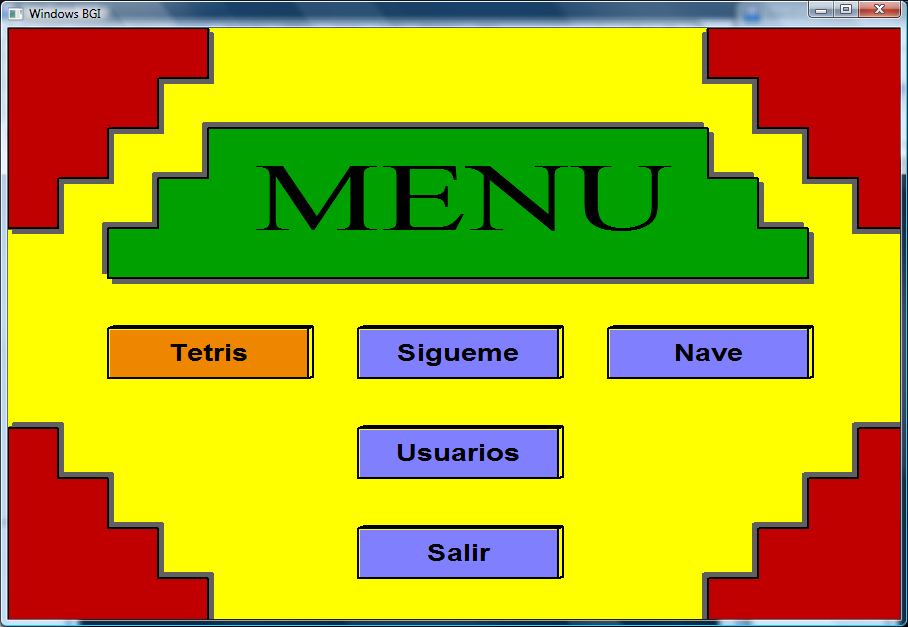
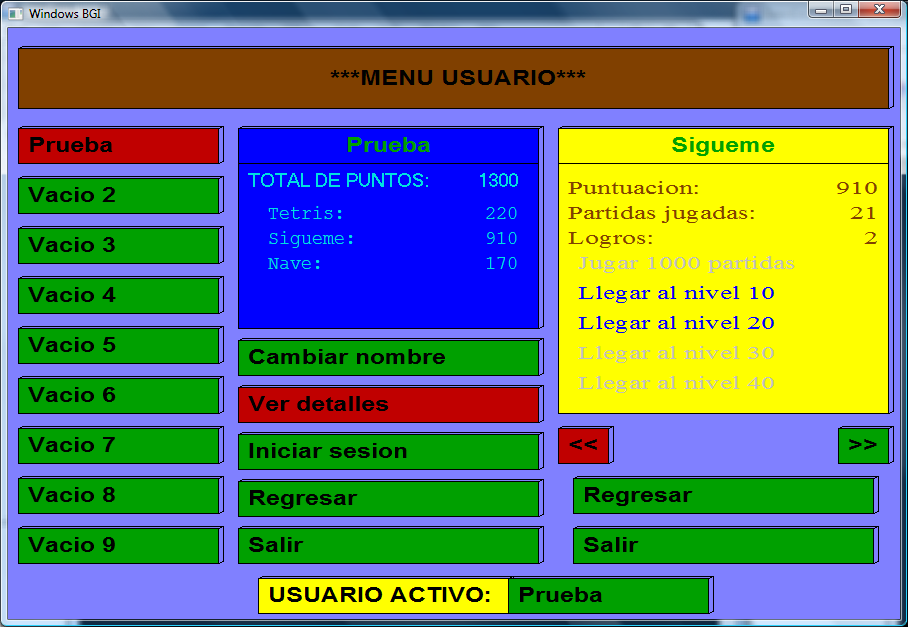


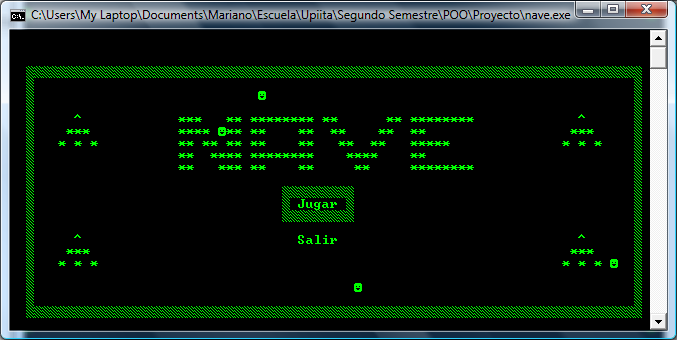
Fig 1. Archivo de texto con la informacion del usuario.



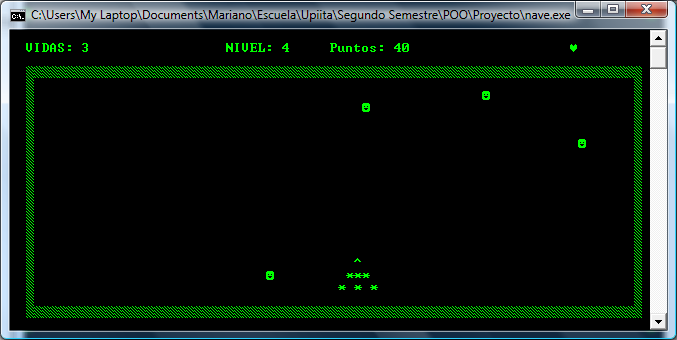
*Fig 2. Apariencia del menu principal*

**

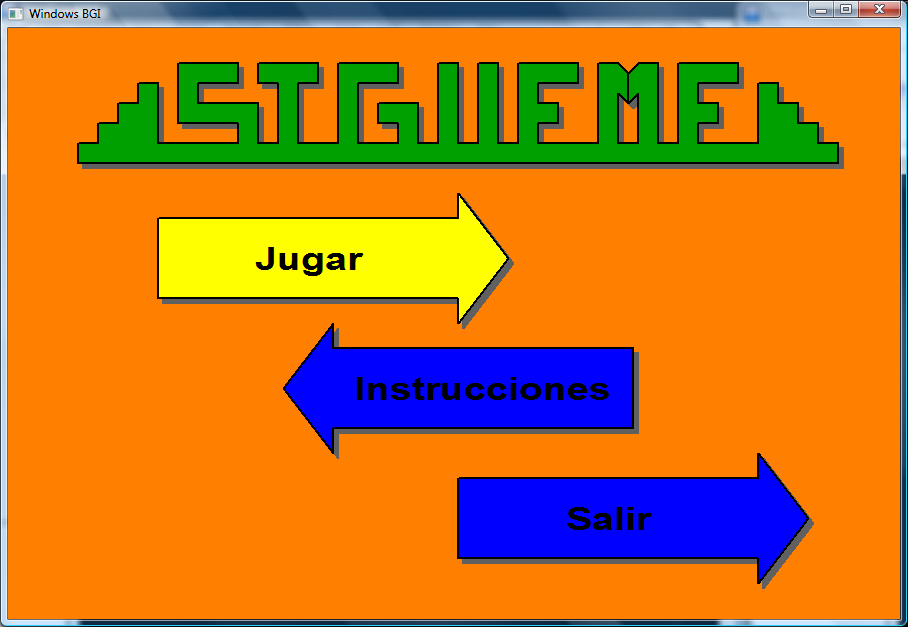
*Fig 3. Menu usuarios*

**

*Fig 4. Menu de juego nave*

**

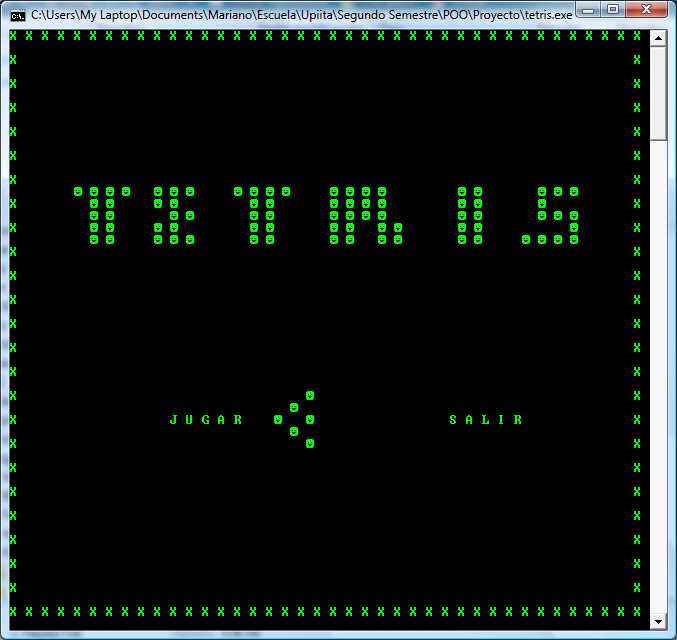
*Fig 5. Juego nave.*

**

*Fig 6. Menu de juego sigueme.*

**

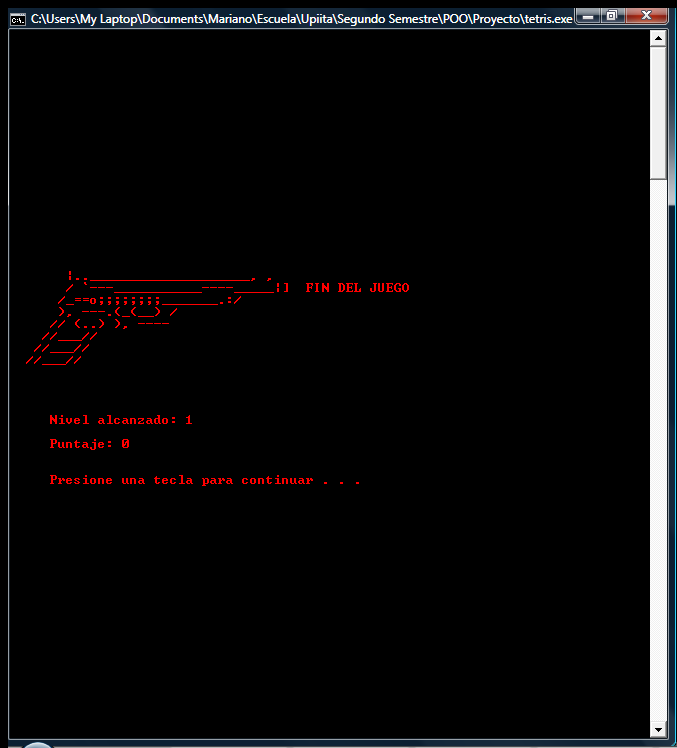
*Fig 7. Juego sigueme.*

**

*Fig 8. Menu juego tetris.*

**

*Fig. 9 Juego tetris.*

**

*Fig 10. Fin de juego.*

Conclusiones

Se logro la elaboración de un paquete de tres juegos con una interfaz amigable con el usuario, una pequeña base de datos donde se guardan los perfiles de los jugadores con datos como su nombre y sus logros en los distintos juegos, lo cual era el objetivo planteado inicialmente y se logro cumplir satisfactoriamente.

Durante la elaboración del proyecto se obtuvieron conocimientos que tal vez fueron vistos en el curso pero se reafirmaron aun más en el desarrollo como la utilización de librería “winbgim” y el desarrollo de gráficos, las interrupciones durante la ejecución del programa mediante el teclado con la sentencia *“if(kbhit())”;* el uso de sonidos con la función Beep(int Hz) y el uso de la función gotoxy. El manejo de cadenas, apuntadores, las conversiones de tipos de dato asi como la utilizacion de menus y submenus gráficos tambien fueron recursos altamente utilizados.

En resumen se utilizaron aproximadamente 4400 líneas de código que integraron los temas antes mencionados así como otros más básicos pero sin los cuales la realización de este proyecto no hubiera podido llevarse a cabo; desafortunadamente aun persisten ciertos problemas que tal vez con el conocimiento de temas de mayor nivel podrían quitarse.

Fue un proyecto que presento varios retos y nos ayudo a aprender cosas nuevas. Consideramos que el trabajo es bueno y se llego a un resultado exitoso. Por ahora solo queda seguir jugando.