**TRƯỜNG ĐẠI HỌC ĐIỆN LỰC**

**KHOA CÔNG NGHỆ THÔNG TIN**

**\*\*\***

****

**BÀI TẬP LỚN**

**MÔN ĐỒ HỌA MÁY TÍNH**

**Đề Tài**

**TRÒ CHƠI RẮN SĂN MỒI**

**Giảng viên hướng dẫn: THẦY NGÔ NGỌC THÀNH**

**Lớp : D9CNPM**

**Chuyên ngành : CÔNG NGHỆ PHẦN MỀM**

**Sinh viên thực hiện: NGUYỄN VĂN HIẾU**

**Hà Nội, ngày 25/06/2016**

**TRƯỜNG ĐẠI HỌC ĐIỆN LỰC**

**KHOA CÔNG NGHỆ THÔNG TIN**

**\*\*\***

****

**BÀI TẬP LỚN**

**MÔN ĐỒ HỌA MÁY TÍNH**

**Đề Tài**

**TRÒ CHƠI RẮN SĂN MỒI**

**Giảng viên hướng dẫn: THẦY NGÔ NGỌC THÀNH**

**Lớp : D9CNPM**

**Chuyên ngành : CÔNG NGHỆ PHẦN MỀM**

**Sinh viên thực hiện: NGUYỄN VĂN HIẾU**

**Hà Nội, ngày 25/06/2016**

# LỜI NÓI ĐẦU

Đồ họa máy tính là một trong những lĩnh vực lí thú nhất và phát triển nhanh nhất của tin học. Ngay từ khi xuất hiện, đồ họa máy tính đã có sức lôi cuốn mãnh liệt, cuốn hút rất nhiều người và được sử dụng ở nhiều lĩnh vực khác nhau như : khoa học, nghệ thuật, kinh doanh, thương mại, công nghiệp, quản lí, giáo dục, giải trí, … Số lượng các chương trình đồ họa ứng dụng thật khổng lồ và phát triểnliên tục không ngừng.

Cùng với sự phát triển của tin học, kỹ thuật đồ họa trên máy vi tính, ngày càng trở nên hiện đại hơn. Giao diện các phần mềm ngày nay trở nên thân thiện, đẹp mắt nhờ các thể hiện đồ họa. Sự hổ trợ của tin học cho các ngành khác trở nên đắc lực hơn nhờ khả năng đồ họa vi tính. Và thế giới thực thì được biểu diễn một cách sinh động, linh hoạt, đầy màu sắc bằng không gian ba chiều.

Trong thành công của kỹ thuật đồ họa ngày nay không thể không nói đến sự phát triển vượt bậc của tốc độ phần cứng lẫn hệ điều hành. Song song với đó, các phần mềm ứng dụng đồ họa mạnh mẽ ra đời hỗ trợ rất nhiều và dễ dàng cho người dùng. Nhưng tiền thân và bản chất của chúng đều xuất phát từ những thuật toán trong đồ họa C++.

Mục tiêu của bài tập lớn này là tìm hiểu các thuật toán, các hàm, … trong đồ họa C++ để tạo ra một ứng dụng hoàn chỉnh có giá trị sử dụng.

Em xin gửi lời cảm ơn chân thành tới thầy giáo: *“****Ngô Ngọc Thành****”* đã tận tâm chỉ bảo và giúp đỡ em hoàn thành đề tài này. Trong quá trình làm đề tài sẽ không tránh khỏi những sai sót, khuyết điểm. Vì vậy, em hy vọng nhận được sự đánh giá và đóng góp nhiệt tình từ phía thầy cô và các bạn để tài của em được hoàn thiện hơn.

***Em xin chân thành cảm ơn!***

# PHẦN 1 : GIỚI THIỆU VỀ ĐỒ HỌA C/C++

## Đồ họa máy tính là gì ?

Đồ họa máy tính là một lĩnh vực của khoa học máy tính nghiên cứu về cơ sở toán học, các thuật toán cũng như các kĩ thuật để cho phép tạo, hiển thị và điều khiển hình ảnh trên màn hình máy tính. Đồ họa máy tính có liên quan ít nhiều đến một số lĩnh vực như đại số, hình học giải tích, hình học họa hình, quang học,... và kĩ thuật máy tính, đặc biệt là chế tạo phần cứng (các loại màn hình, các thiết bị xuất, nhập, các vỉ mạch đồ họa...).

Theo nghĩa rộng hơn, đồ họa máy tính là phương pháp và công nghệ dùng trong việc chuyển đổi qua lại giữa dữ liệu và hình ảnh trên màn hình bằng máy tính. Đồ họa máy tính hay kĩ thuật đồ họa máy tính còn được hiểu dưới dạng phương pháp và kĩ thuật tạo hình ảnh từ các mô hình toán học mô tả các đối tượng hay dữ liệu lấy được từ các đối tượng trong thực tế. Thuật ngữ "đồ họa máy tính" (computer graphics) được đề xuất bởi một chuyên gia người Mĩ tên là William Fetter vào năm 1960. Khi đó ông đang nghiên cứu xây dựng mô hình buồng lái máy bay cho hãng Boeing. William Fetter đã dựa trên các hình ảnh 3 chiều của mô hình người phi công trong buồng lái để xây dựng nên mô hình buồng lái tối ưu cho máy bay Boeing. Đây là phương pháp nghiên cứu rất mới vào thời kì đó. Phương pháp này cho phép các nhà thiết kế quan sát một cách trực quan vị trí của người lái trong khoang buồng lái. William Fetter đã đặt tên cho phương pháp của mình là computer graphics...

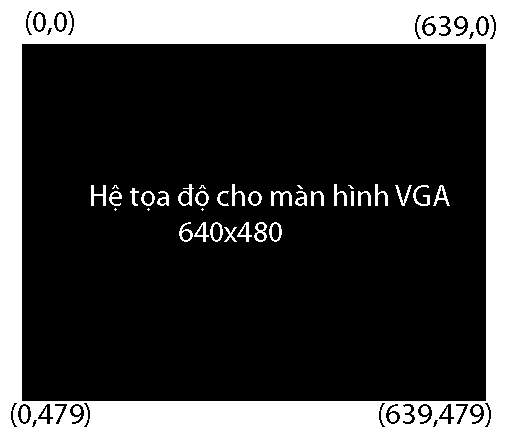
## Đồ họa trong ngôn ngữ lập trình C/C++

Như chúng ta đã biết ,bất kì một chương trình nào khi được đưa vào sử dụng ngoài những tính năng độc đáo mà chương trình đó đem lại thì phần đồ họa hay thiết kế giao diện người dùng (graphic user interface\_GUI) cũng không kém phần quan trọng.

C/C++ là một ngôn ngữ có hỗ trợ thư viện đồ họa là “graphics.h” hoặc “winbgim.h”. Đây là một trong hai thư viện bạn cần thêm vào nếu muốn sử dụng đồ họa trong C/C++

## Màn hình đồ họa trong C/C++

Dưới đây là một minh họa cho màn hình đồ họa có kích thước 640x480. Trong màn hình đồ họa bạn có thể xử lý tới từng điểm pixel. Mỗi pixel có một tọa độ riêng (x,y) với gốc tọa độ là góc trên bên trái (Ảnh), chiều trục Oy hướng từ trên xuống, chiều trục Ox hướng từ trái sang.



Nhờ hệ trục tọa độ này ta có thể tham chiếu tới tới bất kỳ điểm ảnh nào trên màn hình đồ họa.

Các bước thao tác với chế độ đồ họa trong C/C++ :

* Khởi động chế độ đồ họa
* Kiểm tra lỗi đồ họa
* Thao tác trên màn hình đồ họa
* Đóng chế độ đồ họa (Về chế độ màn hình văn bản )

# PHẦN 2 : GIỚI THIỆU VỀ ĐỀ TÀI

## 2.1 Tên đề tài

Tên đề tài : Trò chơi “Rắn Săn Mồi”

## 2.2. Môi trường lập trình

* Ngôn ngữ lập trình : C/C++
* IDE : Dev C++ 5.9.2

## 2.3. Mô tả đề tài

Giải trí là một nhu cầu không thể thiếu của mỗi con người. Trong quá trình lao động, học tập và làm việc, chúng ta thường dễ bị căng thẳng và giải trí chính là một phương thuốc giúp cho tinh thần bạn trở nên ổn định và thư giãn.

Nhìn từ thực tế mà nói ngày nay việc giải trí bằng hình thức chơi game là rất phổ biến. Trong mỗi chúng ta chắc rằng ai cũng đã từng một lần chơi game, thậm chí là chơi game hàng ngày. Nó đã trở thành món ăn tinh thần không thể thiếu của rất nhiều người sau những giờ học tập và làm việc căng thẳng. Cũng xuất phát từ thực tế này mà em đã quyết định lựa chọn đề tài làm một mini game nho nhỏ.

Trước khi smartphone phổ biến như hiện nay thì nokia từng là một hãng điện thoại tiếng tăm và một trò chơi kinh điển của Nokia mà ai ai cũng biết tới “Snake Xenzia”. Đây là một tựa game tuy không có màu mè gì và giao diện cũng hết sức đơn giản được tích hợp trong các dòng máy của Nokia. Thế nhưng nó lại là một tựa game huyền thoại cho đến hiện tại. Minh chứng là hiện này có rất nhiều tựa game cải tiến từ dòng game này được ra đời và phát triển trên nhiều nền tảng, hệ điều hành khác nhau.

Bằng kiến thức đồ họa em được học và tìm hiểu thêm trên internet, em đã quyết định thực hiện đề tài “Game Rắn Săn Mồi” sử dụng ngôn ngữ lập trình C/C++ chạy trên hệ điều hành windows. Với mong muốn sản phẩm này ra đời sẽ là một trò chơi giúp mọi người giải trí, giúp bản thân củng cố thêm kiến thức được học, là nền tảng cho các môn học tiếp về sau. Hi vọng mini game sẽ được sự ủng hộ và đóng góp của thầy và các bạn !

# PHẦN 3 : PHÂN TÍCH THUẬT TOÁN

## 3.1. Phân tích

Trò chơi **Rắn Săn Mồi** gồm một đối tượng **Snake** (Rắn) di chuyển tự do trên màn hình theo 4 hướng (trên, dưới, trái, phải) và một đối tượng **Food** (Thức ăn) được được random ngẫu nhiên trên màn hình chơi.

### 3.1.1 Đối tượng **Point**

Như vậy chúng ta có thể tạo ra một đối tưởng **Point** để lưu tọa độ mỗi điểm trên màn hình đồ họa

struct Point {

int x, y;

int x0, y0;

};

Mỗi đối tượng **Food** sẽ có một tọa độ (x,y)

Đối tượng **Snake** sẽ được tạo bởi một mảng các đối tượng **Point** xếp cạnh nhau

Cặp tọa độ (x,y) sẽ lưu vị trí Point hiện tại.

Cặp tọa độ (x0,y0) sẽ lưu ví trí trước đó của Point hiện tại để các đốt sau đó của con rắn có thể sử dụng.

Để **Snake** có thể di chuyển trên màn hình thì ta cần thêm một biến lưu hướng đi của nó. Ta sẽ tận dụng luôn đối tượng Point để xác định hướng theo tọa độ (x,y).

(0, -10)

direction

(-10, 0)

(0, 10)

(10, 0)

Vậy để di chuyển Snake. Ta cần di chuyển đầu của con Rắn trước tiên.

snake[0].x += direction.x;

snake[0].y += direction.y;

Nhưng trước đó ta phải lưu lại tọa độ cũ của nó để đốt kế sau nó lấy làm tọa độ mới

snake[0].x0 = snake[0].x;

snake[0].y0 = snake[0].y;

Để di chuyển phần thân rắn ta sẽ chạy một vòng lặp hết phần thân rắn

snake[i].x = snake[i-1].x0;

snake[i].y = snake[i-1].y0;

Rắn sẽ ăn được Thức ăn khi tọa độ đầu Snake[0] trùng với tọa độ thức ăn. Khi đó ta cần :

* + - Tăng chiều dài của rắn
    - Random lại thức ăn
    - Tăng điểm số

Nếu đầu Rắn chạm tường hoặc chạm thân thì Game kết thúc ( Chế độ chơi hiện đại) và chỉ chết khi chạm vào thân (chế độ chơi cổ điển)

Khi trò chơi kết thúc lưu lại điểm cao và tên người chơi.

### 3.1.2 Đối tượng HighScore

Để tăng tính hấp dẫn cho game, ta tạo thêm một đối tượng **HighScore** lưu thông tin Tên người chơi và Số điểm đạt được nếu người chơi đó đạt điểm cao

struct HighScore {

int score;

char name[30];

};

Thông tin điểm cao sẽ được ghi ra một file txt và lưu lại để so sánh với các người chơi sau đó.

Nếu người chơi mới đạt điểm cao hơn trong danh sách đã lưu thì sẽ thay thế thành tích của người chơi cũ tương ứng bởi thành tích của người chơi mới

# PHẦN 4 : CÁC HÀM CHÍNH SỬ DỤNG TRONG GAME

## 4.1 Hàm **void initGame ();**

Hàm này có các chức năng sau :

* Vẽ khung của trò chơi
* Khởi tạo tọa độ ban đầu cho Snake (Gồm 3 Point) và Food
* Khởi tạo hướng đi ban đầu
* Vẽ backgound và các hiển thị không lặp

void initGame(){

setbkcolor (15);

cleardevice ();

setwindowtitle ("SNAKE-Playing....");

endGame = false;

setfillstyle (1,5);

bar (MINX,MINY,MAXX,MINY+5);

bar (MAXX,MINY,MAXX-5,MAXY);

bar (MAXX,MAXY,MINX,MAXY-5);

bar (MINX,MINY,MINX+5,MAXY);

setcolor (0);

int bk = getbkcolor();

setfillstyle (SOLID\_FILL,0);

int a[] = {15,350,30,350,30,345,50,355,30,365,30,360,15,360,15,350};

fillpoly (8,a);

settextstyle (4,0,3); outtextxy (70,340,"Right");

int b[] = {15,410,30,400,30,405,45,405,45,415,30,415,30,420,15,410};

fillpoly (8,b);outtextxy (70,400,"Left");

int c[] = {235,370,235,355,230,355,240,340,250,355,245,355,245,370,235,370};

fillpoly (8,c);outtextxy (265,340,"Up");

int d[] = {235,400,235,415,230,415,240,430,250,415,245,415,245,400,235,400};

fillpoly (8,d);outtextxy (265,400,"Down");

bar (50,450,170,480);setcolor (15);settextstyle(1,0,1);setbkcolor(0);outtextxy(60,455,"SPACE");setbkcolor(bk);

setcolor(0);settextstyle (4,0,3);outtextxy(200,450,"Pause/Resum");

bar (50,500,170,530);setcolor (15);settextstyle(1,0,1);setbkcolor(0);outtextxy(80,505,"ESC");setbkcolor(bk);

setcolor(0);settextstyle (4,0,3);outtextxy(200,500,"End Game");

snake[0].x = 60;snake[0].y = 50;

snake[1].x = 50;snake[1].y = 50;

snake[2].x = 40;snake[2].y = 50;

snakeLength = 3;

setfillstyle (1,BACKGROUND);

bar (25,25,415,215);

direction.x = 10;direction.y = 0; // Khoi tao di theo huong trai qua phai

setcolor (2);

srand ( time(NULL));

do{

food.x = (rand() % (39) + 3)\*10;

food.y = (rand() % (19) + 3)\*10;

}while (checkPoint() == false);

}

## 4.2. Hàm **bool checkPoint ();**

Hàm này có chức năng kiểm tra trùng lặp khi tạo mới đối tượng Food. Kiểm tra chắc chắn rằng tọa độ thức ăn không trùng với thân Rắn

bool checkPoint (){

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

if (food.x == snake[i].x && food.y == snake[i].y)

return false;

}

return true;

}

## 4.3. Hàm **void drawPoint (int x,int y,int r)**

Hàm này có chức năng vẽ một đối tượng Point (hình tròn) lên màn hình đồ họa có tọa độ tâm (x,y) và có bán kính r

void drawPoint (int x,int y,int r){

circle (x,y,r);

floodfill (x,y,getcolor());

}

## 4.4. Hàm **void drawSnake ()**

Hàm này gọi làm hàm drawPoint () có các chức năng :

* Vẽ đối tượng Snake lên màn hình đồ họa
* Xóa bỏ đốt cuối cùng của thân rắn mỗi khi rắn di chuyển được thêm một bước

void drawSnake (){

setfillstyle (1,10);

int count = 0;

for (int i = 0;count < snakeLength;i++){

drawPoint(snake[i].x, snake[i].y,5);

count++;

}

//delete point when snake move

int x = getcolor();

setfillstyle (1,BACKGROUND);

drawPoint(snake[snakeLength-1].x0,snake[snakeLength-1].y0,5);

setcolor (BACKGROUND);

circle (snake[snakeLength-1].x0,snake[snakeLength-1].y0,5);

setcolor(x);

setfillstyle (1,5);

}

4.5. Hàm **void drawFood ()**

Hàm này gọi lại hàm drawPoint () và chỉ thực hiện nhiệm vụ hiển thị thức ăn lên màn hình đồ họa

void drawFood (){

int x = getcolor();

int c = rand() % 14 + 1;

setcolor(c);setfillstyle (SOLID\_FILL, c);

drawPoint(food.x, food.y,5);

setcolor(x);

}

## 4.6. Hàm **void drawGame ()**

Hàm này gọi lại hai hàm **drawSnake()** và hàm **drawFood()** và thực hiện thêm công việc hiển thị điểm của người chơi lên màn hình

void drawGame (){

drawSnake();

drawFood();

int x = getcolor ();

setcolor (2);

sprintf(score\_str,"%d",snakeLength\*level\*10-level\*30); //show score

settextstyle(4,0,3);

outtextxy(450,20,"SCORE:");

setcolor(12);

outtextxy(600,20,score\_str);

setcolor (x);

}

## 4.7. Hàm **void classic()**

Hàm này thực hiện chức năng chơi cổ điển. Người chơi có thể đi xuyên tường.

Chức năng của hàm

* Di chuyển Rắn theo điều khiển hướng đi
* Nếu Rắn đi ra ngoài mép tường – cho rắn xuất hiện lại ở phía bên kia
* Kiểm tra rắn ăn mồi : tăng chiều dài, điểm số và random lại tọa độ thức ăn
* Kiểm tra game kết thúc

void classic(){

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

if (i == 0) {

snake[0].x0 = snake[0].x;snake[0].y0 = snake[0].y;

snake[0].x += direction.x;

snake[0].y += direction.y;

}else{

snake[i].x0 = snake[i].x;snake[i].y0 = snake[i].y;

snake[i].x = snake[i-1].x0;snake[i].y = snake[i-1].y0;

}

if (snake[i].x >= MAXX) snake[i].x = MINX + 10;

if (snake[i].x <= MINX) snake[i].x = MAXX - 10;

if (snake[i].y >= MAXY) snake[i].y = MINY + 10;

if (snake[i].y <= MINY) snake[i].y = MAXY - 10;

if (i != 0 && (snake[0].x == snake[i].x && snake[0].y == snake[i].y)) endGame = true;

}

if (snake[0].x == food.x && snake[0].y == food.y){

snake[snakeLength].x = snake[snakeLength-1].x0;snake[snakeLength].y = snake[snakeLength-1].y0;

snakeLength++;

PlaySound(TEXT("eatFood.wav"), NULL, SND\_ASYNC);

//Random again food if sanke ate food

srand ( time(NULL));

do{

food.x = (rand() % (39) + 3)\*10;

food.y = (rand() % (19) + 3)\*10;

}while (checkPoint() == false);

}

}

## 4.8. Hàm **void modern()**

Hàm này tương tự hàm classic() nhưng thay vào đó rắn không thể đi xuyên tường và đó là một trong những điều kiện bổ sung vào kiểm tra game kết thúc của hàm này

void modern(){

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

if (i == 0){

snake[0].x0 = snake[0].x;snake[0].y0 = snake[0].y;

snake[0].x += direction.x;

snake[0].y += direction.y;

}

else {

snake[i].x0 = snake[i].x;snake[i].y0 = snake[i].y;

snake[i].x = snake[i-1].x0;snake[i].y = snake[i-1].y0;

}

if (snake[0].x < MINX+5 || snake[0].y < MINY+5 || snake[0].x > MAXX-5 || snake[0].y > MAXY - 5 )

endGame = true;

if (i != 0 && snake[0].x == snake[i].x && snake[0].y == snake[i].y) endGame = true;

}

if (snake[0].x == food.x && snake[0].y == food.y){

snake[snakeLength].x = snake[snakeLength-1].x0;snake[snakeLength].y = snake[snakeLength-1].y0;

snakeLength++;

PlaySound(TEXT("eatFood.wav"), NULL, SND\_ASYNC);

//Random again food if sanke ate food

srand ( time(NULL));

do{

food.x = (rand() % (39) + 3)\*10;

food.y = (rand() % (19) + 3)\*10;

}while (checkPoint() == false);

}

}

## 4.9. Hàm **void changeDirecton (int x)**

x là biến kiểu nguyên lưu giữ giá trị tương ứng của của phím chuyển sang kiểu int

Đây là hàm thực thi việc nhận lệnh từ bàn phím, cụ thể là các phím mũi tên điều hướng và phím ESC

Hàm sẽ nhận vào mã phím nhập từ bàn phím và thực hiện thay đổi hướng đi theo mã phím tương ứng.

Nếu người dùng nhấn phím ESC – Game sẽ dừng lại

void changeDirecton (int x){

switch(x){

case 72:

if (direction.y != DIRECTION) {

PlaySound(TEXT("beep.wav"), NULL, SND\_ASYNC);

direction.y = -DIRECTION; direction.x = 0;

}

break;

case 80:

if (direction.y != -DIRECTION) {

PlaySound(TEXT("beep.wav"), NULL, SND\_ASYNC);

direction.y = DIRECTION; direction.x = 0;

}

break;

case 77:

if (direction.x != -DIRECTION) {

PlaySound(TEXT("beep.wav"), NULL, SND\_ASYNC);

direction.x = DIRECTION; direction.y = 0;

}

break;

case 75:

if (direction.x != DIRECTION) {

PlaySound(TEXT("beep.wav"), NULL, SND\_ASYNC);

direction.x = -DIRECTION; direction.y = 0;

}

break;

case 27:

endGame = true;

break;

}

}

## 4.10. Hàm **void mainLoop (void (\*gloop)())**

Hàm này chính là hàm lặp cho phép game chạy và thực thi toàn bộ các hành động trong khi game chạy

Tại đây, em sử dụng đệ quy để hàm gọi lại chính nó.

Tham số truyền vào là một trong hai hàm classic() hoặc hàm modern() tùy theo lựa chọn của người chơi

Hàm này còn có chức năng Pause/Resum game khi đang chơi nếu người dùng ấn phím SPACE

void mainLoop (void (\*xxx)()){

int x = getcolor();

if (!kbhit()){

xxx();

}

else{

char ch;

ch = getch();

if (ch == -32) ch = getch();

if ( ch == 32){

setcolor(0);outtextxy (100,100,"Game Pause");setcolor(x);

ch = getch();

while (ch != 32){

ch = getch();

delay(0);

}

setcolor (BACKGROUND);

outtextxy (100,100,"Game Pause");

setcolor (x);

delay(1000);

}

changeDirecton(ch);

mainLoop (xxx);

}

}

## 4.11. Hàm **bool isEmpty()**

Hàm này kiểm tra file lưu điểm cao. Nếu file là rỗng thì trả về giá trị true

bool isEmpty(){

FILE \*fp;

long size;

fp = fopen("highscore.txt","r");

if(fp){

fseek (fp, 0, SEEK\_END);

size = ftell(fp);

fclose(fp);

}

return(size == 0);

}

## 4.12. Hàm initScore()

void initScore(){

if (isEmpty()){

for (int i = 0;i < 5;i++){

strcpy(highscore[i].name,"PLAYER");

highscore[i].score = 0;

}

getHighScore();

}else{

char arr[20];

int count = 0;

FILE \*f;

f = fopen("highscore.txt", "r");

for (int i = 0;i < 5;){

while (!feof(f)){

count++;

fscanf(f, "%s", arr);

if (count%2 == 1){

strcpy(highscore[i].name,arr);

//printf ("%s\t",ch);

}

else{

highscore[i].score = atoi(arr);

//printf ("%d\n",atoi(arr));

i++;

}

}

}

}

}

Hàm này kiểm tra file lưu điểm cao có rỗng hay không. Nếu file rỗng thì hàm sẽ khở tạo 5 giá trị mặc định có tên PLAYER và điểm cao là 0.

Nếu hàm có chứa thông tin hàm sẽ đọc thông tin điểm cao lưu vào mảng kiểu HighScore

## 4.13 Hàm void showHighScore()

Hàm này có chức năng đọc thông tin của file điểm cao và xuất ra thông tin khi được gọi

void showHighScore(){

FILE \*f;

f = fopen("highscore.txt", "r");

char ch[20];

settextstyle (1,0,5);

setcolor (15);

outtextxy (150,50,"HIGH SCORE");

settextstyle (1,0,4);

int y = 150,count = 0;

while (!feof(f)){

if (count == 10) break;

count++;

fscanf(f, "%s", ch);

if (count%2 == 1){

setcolor (10);

outtextxy (180,y,ch);

y+=50;

}else{

setcolor (12);

outtextxy (500,y-50,ch);

}

}

fclose(f);

}

## 4.14 Hàm **void getHighScore ()**

Hàm này có chức năng ghi lại thông tin điểm cao của người chơi và lưu vào file

void getHighScore (){

FILE \*f;

f = fopen("highscore.txt", "w");

for (int i = 0;i < 5;i++){

fputs(highscore[i].name,f);

fputs(" ",f);

fprintf(f,"%d",highscore[i].score);

fputs("\n",f);

}

fclose(f);

}

## 4.15 Hàm **checkHighScore (int \_score)**

Hàm này thực hiện nhiệm vụ nhận điểm cao của người chơi, lấy thông tin tên người chơi nếu người chơi đó đạt thành tích cao.

Hàm sẽ lưu giá trị vào mảng kiểu HighScore và gọi lại hàm **getHighScore()** để lưu lại thông tin

void checkHighScore (int \_score){

char \_name[20]={""};

for (int i = 0;i < 5;i++){

if (\_score > highscore[i].score){

//to do sth

settextstyle(1,0,3);

for (int j = 0;j < 50;j++){

if (j%2 == 0){

setcolor (14);

if (i == 0)

outtextxy(460,100,"BEST SCORE");

else

outtextxy(460,100,"HIGH SCORE");

delay(100);

}else{

setcolor (9);

if (i == 0)

outtextxy(460,100,"BEST SCORE");

else

outtextxy(460,100,"HIGH SCORE");

delay(100);

}

}

settextstyle(1,0,2);

setcolor (4);outtextxy(430,150,"Player:");

delay(100);

char ch1;

int x = 0;

char str[2];

str[1] = 0;

while (ch1 != 13 && x < 10){

do{

ch1 = getch();

}while (ch1 < 65 && ch1 >90 || ch1 < 97 && ch1 > 132);

x++;

str[0] = ch1;

strcat(\_name,str);

outtextxy(540,150,\_name);

}

for (int j = 4;j > i;j--){

strcpy(highscore[j].name,highscore[j-1].name);

highscore[j].score = highscore[j-1].score;

}

strcpy(highscore[i].name,\_name);

highscore[i].score = \_score;

break;

}

}

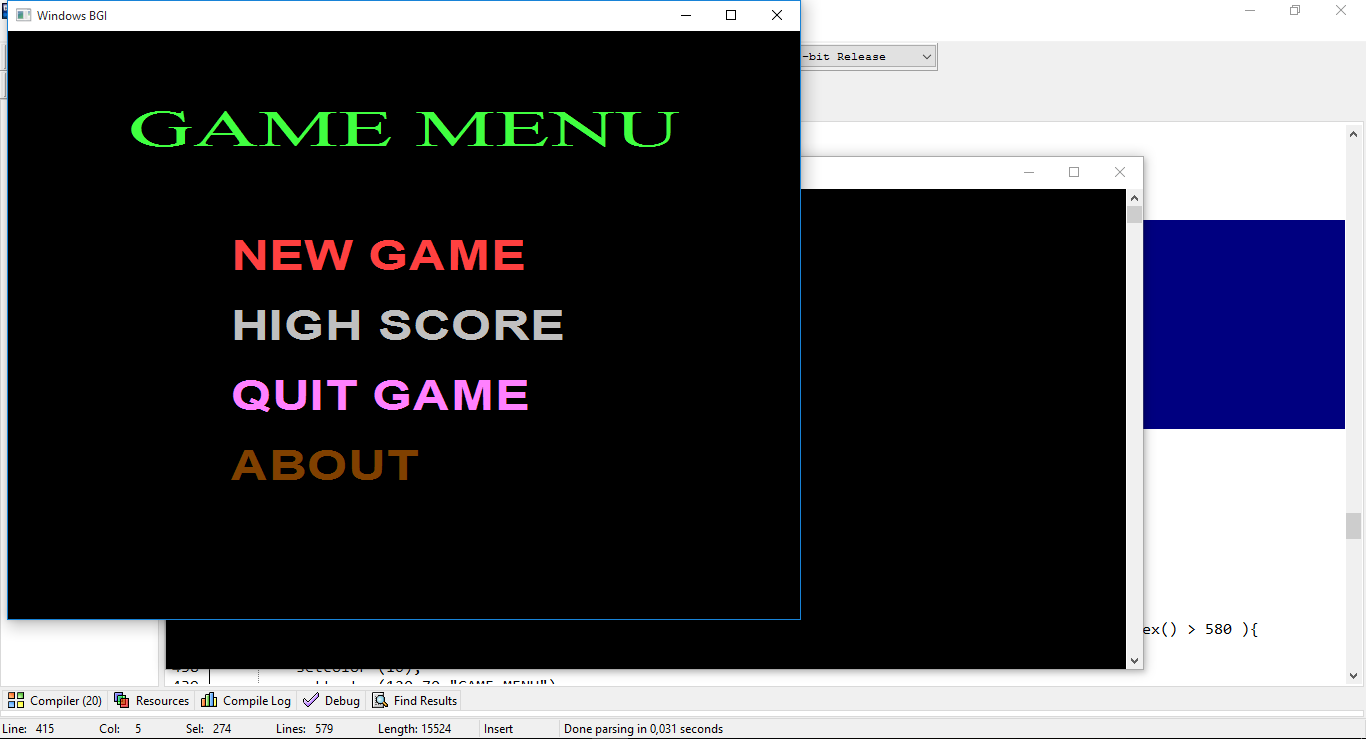
getHighScore();

}

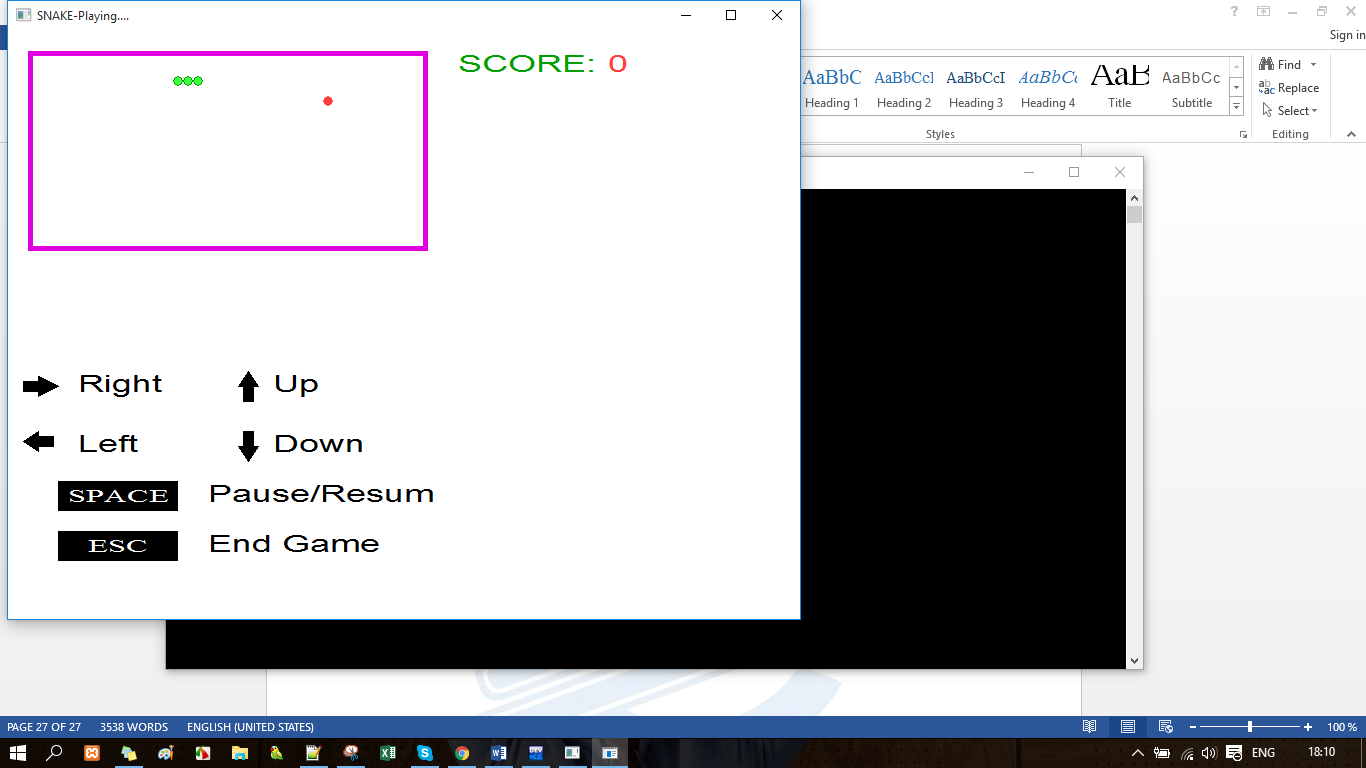
* Ngoài ta chương trình còn có một số hàm nhỏ thực hiện một số chức năng nhất định.
* Trong quá trình thao tác em có sử dụng tính này click chuột thay cho nhấn bàn phím
* Thêm nhạc cho game trở nên sống động

# PHẦN 5 : MỘT SỐ HÌNH ẢNH TRONG GAME

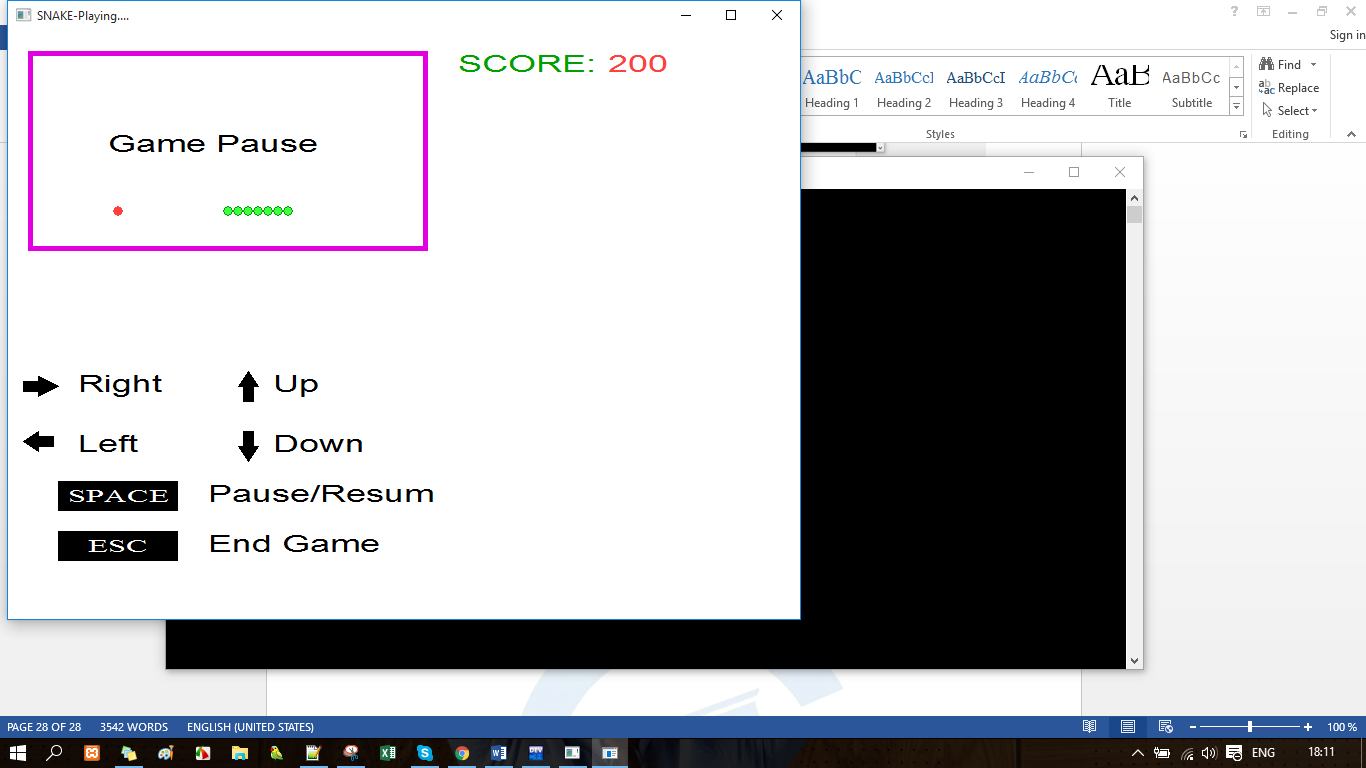
## Menu Start Game



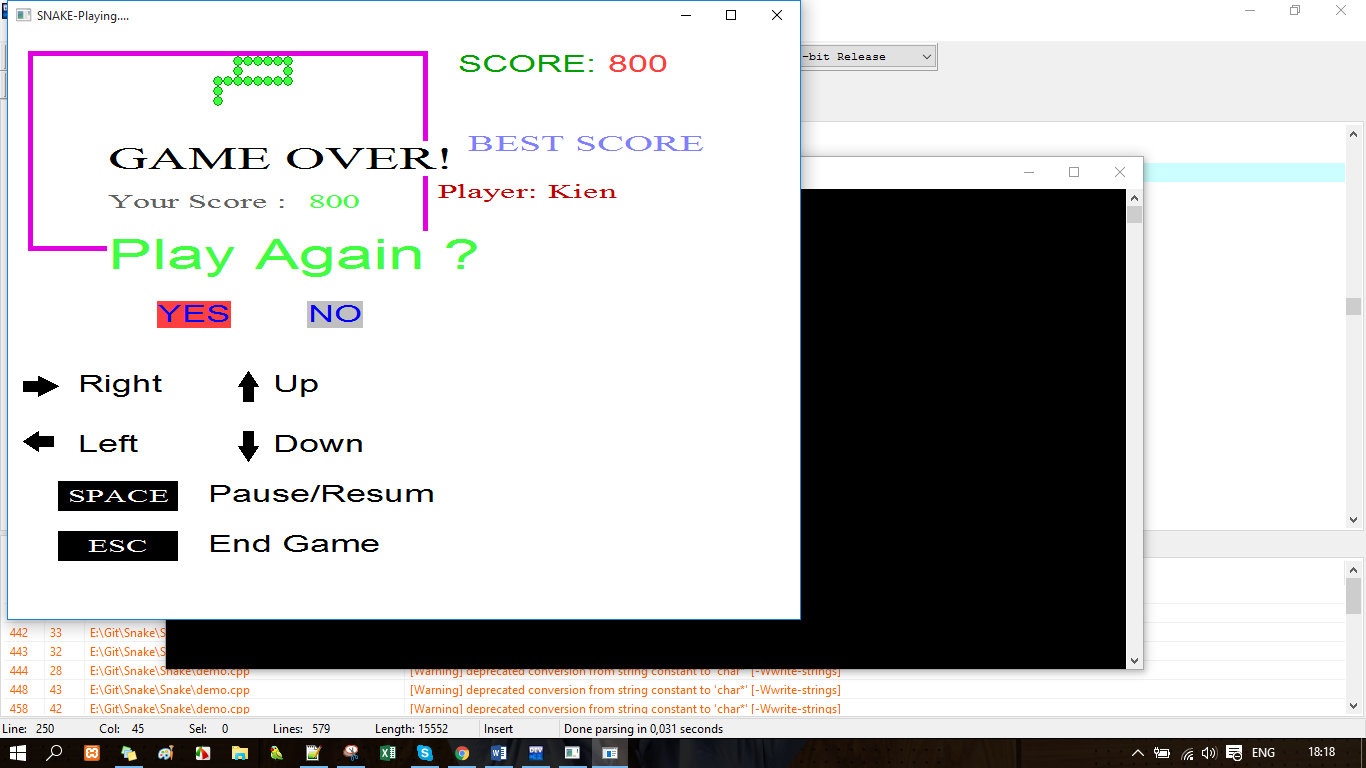
## Game Playing



## Pause/Resum Game



## End Game



# TỔNG KẾT

Trên đây là báo cáo của em về đề tài “Game Rắn Săn Mồi”.

Qua việc làm bài tập lớn và các kiến thức mà thầy dạy về môn học đồ họa máy tính đã cho em nhiều kiến thức bổ ích áp dụng vào thực tiễn cuộc sống. Đồ họa máy tính còn có nhiều ứng dụng để áp dụng vào giáo dục,xây dựng,thiết kế,..v..v vẫn đang chờ chúng ta khai phá…

# TÀI LIỆU THAM KHẢO

* Giáo trình Lập trình C++ của Thầy Phạm Văn Ất
* Các tài liệu có liên quan trên Google

Source Code Full :