TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN**

**ĐỒ HỌA MÁY TÍNH**

**ĐỀ TÀI:**

**GAME RẮN SĂN MỒI**

|  |  |
| --- | --- |
|  |  |
| |  |  | | --- | --- | | **Sinh viên thực hiện** | **: NGUYỄN VĂN DŨNG**  **NGUYỄN HOÀNG ĐẠT** | | **Giảng viên hướng dẫn** | **: CÙ VIỆT DŨNG** | | | **Ngành** | **: CÔNG NGHỆ THÔNG TIN** | | | **Chuyên ngành** | **: CÔNG NGHỆ PHẦN MỀM** | | | **Lớp** | **: D11CNPM** | | | **Khóa** | **: D11** | | |  |
|  |  |

***Hà Nội, ngày 09 tháng 06 năm 2018***

**PHIẾU CHẤM ĐIỂM**

Sinh viên thực hiện:

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Chữ ký** | **Ghi chú** |
| Nguyễn Văn Dũng  (1681310021) |  |  |
| Nguyễn Hoàng Đạt  (1681310014) |  |  |
|  |  |  |
|  |  |  |

Giảng viên chấm:

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Chữ ký** | **Ghi chú** |
| Giảng viên chấm 1 : |  |  |
| Giảng viên chấm 2 : |  |  |

**Góp ý của giảng viên**

**MỤC LỤC**

Lời mở đầu

Trang

[PHẦN 1: GIỚI THIỆU VỀ ĐỒ HỌA C/C++ 1](#_Toc516240285)

[1.1 Đồ họa máy tính là gì? 1](#_Toc516240286)

[1.2 Đồ họa trong ngôn ngữ lập trình C/C++ 1](#_Toc516240287)

[1.3 Màn hình đồ họa trong C/C++ 2](#_Toc516240288)

[PHẦN 2: GIỚI THIỆU ĐỀ TÀI 3](#_Toc516240289)

[2.1 Tên đề tài 3](#_Toc516240290)

[2.2 Công cụ thực hiện 3](#_Toc516240291)

[2.3 Mô tả đề tài 3](#_Toc516240292)

[PHẦN 3: PHÂN TÍCH THUẬT TOÁN 4](#_Toc516240293)

[3.1 Phân tích 4](#_Toc516240294)

[3.1.1 Đối tượng ToaDo (Tọa độ) 4](#_Toc516240295)

[3.1.2 Đối tượng Snake (Rắn) 4](#_Toc516240296)

[3.1.3 Đối tượng HoaQua (Hoa quả) 5](#_Toc516240297)

[PHẦN 4: CÁC HÀM CHÍNH SỬ DỤNG TRONG GAME 7](#_Toc516240298)

[4.1 Hàm DrawCircle (Vẽ hình tròn) 7](#_Toc516240299)

[4.2 Hàm KhoiTao (Khởi tạo) 8](#_Toc516240300)

[4.3 Hàm VeRan (Vẽ rắn) 8](#_Toc516240301)

[4.4 Hàm VeHoaQua (Vẽ hoa quả) 8](#_Toc516240302)

[4.5 Hàm HienThi (Hiển thị) 9](#_Toc516240303)

[4.6 Hàm DieuKhien (Điều khiển) 9](#_Toc516240304)

[4.7 Hàm GetHighScore 11](#_Toc516240305)

[4.8 Hàm CheckHighScore 12](#_Toc516240306)

[4.9 Hàm XuLy (Xử lý) 13](#_Toc516240307)

[4.10 Hàm ShowHighScore 14](#_Toc516240308)

[4.11 Hàm Run 14](#_Toc516240309)

[PHẦN 5: MỘT SỐ HÌNH ẢNH TRONG GAME 15](#_Toc516240310)

[KẾT LUẬN 17](#_Toc516240311)

[TÀI LIÊU THAM KHẢO 18](#_Toc516240312)

LỜI MỞ ĐẦU

*Lời đầu tiên, chúng em xin chân thành gửi lời cảm ơn tới các thầy cô giáo trong Trường Đại học Điện Lực nói chung và các thầy cô giáo trong Khoa Công nghệ thông tin nói riêng đã tận tình giảng dạy, truyền đạt cho chúng tôi những kiến thức cũng như kinh nghiệm quý báu trong suốt quá trình học hành.*

*Đặc biệt, chúng em gửi lời cảm ơn đến Giáo viên hướng dẫn Cù Việt Dũng, cô đã tận tình theo sát giúp đỡ, trực tiếp chỉ bảo, hướng dẫn trong suốt quá trình nghiên cứu và học tập của chúng em. Trong thời gian học tập với cô, nhóm chúng em không những tiếp thu thêm nhiều kiến thức bổ ích mà còn học tập được tinh thần làm việc, thái độ nghiên cứu khoa học nghiêm túc, hiệu quả. Đây là những điều rất cần thiết cho chúng em trong quá trình học tập và công tác sau này.*

*Hà Nội, ngày 09 tháng 06 năm 2018*

*Người thực hiện:*

*Nguyễn Văn Dũng*

*Nguyễn Hoàng Đạt*

**PHẦN 1: GIỚI THIỆU VỀ ĐỒ HỌA C/C++**

**1.1 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...

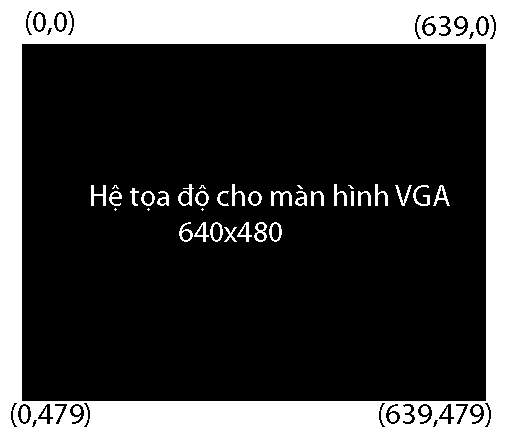
**1.2 Đồ họa trong ngôn ngữ lập trình C/C++**

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++.

**1.3 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 ĐỀ TÀI**

**2.1 Tên đề tài**

Tên đề tài: Trò chơi “Rắn Săn Mồi”.

**2.2 Công cụ thực hiện**

Ngôn ngữ lập trình: C/C++

IDE: DEV-C++ 4.9.9.2

**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, nhóm 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 khung màn hình đã lập trình sẵn (500,300) theo 4 hướng (trên, dưới, trái, phải) và một đối tượng **HoaQua** (Hoa quả) được xuất hiện ngẫu nhiên trên khung màn hình đã lập trình.

***3.1.1 Đối tượng ToaDo (Tọa độ)***

Như vậy, chúng ta cần tạo ra một cấu trúc ToaDo để lưu tọa độ (x, y) của đối tượng Snake và HoaQua.

struct ToaDo {

int x, y;

};

**3.1.2 Đối tượng Snake (Rắn)**

struct Snake {

ToaDo dot[31];

int n;

TrangThai tt;

};

Snake sẽ được cấu tạo bởi nhiều đốt (dot[31]) và tối đa là 31 đốt và có biến trạng thái (tt) với kiểu dữ liệu TrangThai được lập trình gồm 4 trạng thái: UP, DOWN, LEFT, RIGHT.

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 **ToaDo** để xác định hướng theo tọa độ (x, y).

(0, -10)

Direction

(-10, 0)

(0, 10)

(10, 0)

Thực chất để di chuyển con rắn di chuyển, ta chỉ cần di chuyển đốt đầu con rắn và những đốt sau di chuyển theo đốt phía trước của nó.

for(int i = snake.n - 1; i > 0; i--){

snake.dot[i] = snake.dot[i - 1];

}

**3.1.3 Đối tượng HoaQua (Hoa quả)**

struct HoaQua {

ToaDo td

};

Để hoa quả có thể xuất hiện trên màn hình game ta cần lưu vị trí của nó, ở đây ta sẽ dùng biến ToaDo để lưu (x, y).

Rắn sẽ ăn được Thức ăn khi tọa độ đầu Snake.dot[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 thân thì Game kết thúc. Khi trò chơi kết thúc lưu lại điểm cao và tên người chơi.

**PHẦN 4: CÁC HÀM CHÍNH SỬ DỤNG TRONG GAME**

**4.1 Hàm DrawCircle (Vẽ hình tròn)**

Sử dụng thuật toán MidPoint ta có hàm như sau.

void drawcircle(int x0, int y0, int radius)

{

int x = radius-1;

int y = 0;

int dx = 1;

int dy = 1;

int err = dx - (radius << 1);

while (x >= y){

putpixel(x0 + x, y0 + y, WHITE);

putpixel(x0 + y, y0 + x, WHITE);

putpixel(x0 - y, y0 + x, WHITE);

putpixel(x0 - x, y0 + y, WHITE);

putpixel(x0 - x, y0 - y, WHITE);

putpixel(x0 - y, y0 - x, WHITE);

putpixel(x0 + y, y0 - x, WHITE);

putpixel(x0 + x, y0 - y, WHITE);

if (err <= 0){

y++;

err += dy;

dy += 2;

}

if (err > 0){

x--;

dx += 2;

err += dx - (radius << 1);

}

}

}

**4.2 Hàm KhoiTao (Khởi tạo)**

Hàm có chức năng:

+ Khởi tạo snake gồm 3 đốt và gán sẵn vị trị 3 đốt cho snake, gán sẵn trạng thái cho snake**.**

+ Khởi tạo vị trí cho hoa quả**.**

void KhoiTao(Snake &snake, HoaQua &hq){

snake.n = 3;

snake.dot[0].x = 20;

snake.dot[0].y = 20;

snake.dot[1].x = 10;

snake.dot[1].y = 20;

snake.dot[2].x = 10;

snake.dot[2].y = 20;

snake.tt = RIGHT;

hq.td.x = 40;

hq.td.y = 40;

}

**4.3 Hàm VeRan (Vẽ rắn)**

Dựa vào độ dài của snake ta vẽ từng đốt của nó, mỗi đốt ta sử dụng hàm DrawCircle để vẽ.

void VeRan(Snake snake){

for(int i = 0; i < snake.n; i++){

setcolor(WHITE);

DrawCircle(snake.dot[i].x, snake.dot[i].y, 3);

}

}

**4.4 Hàm VeHoaQua (Vẽ hoa quả)**

Ta dùng hàm DrawCircle để vẽ hoa quả ra màn hình game.

void VeHoaQua(HoaQua hd){

setcolor(YELLOW);

DrawCircle(hq.td.x, hq.td.y, 3);

}

**4.5 Hàm HienThi (Hiển thị)**

Hàm hiển thị để hiển thị ra snake, hoa ta vừa vẽ và vẽ thêm giao diện.

void HienThi(Snake snake, HoaQua hq){

cleardevice();

VeRan(snake);

VeGiaoDien(snake);

VeHoaQua(hq);

}

**4.6 Hàm DieuKhien (Điều khiển)**

void DieuKhien(Snake &snake){

for(int i = snake.n - 1; i > 0; i--){

snake.dot[i] = snake.dot[i - 1];

}

if(snake.tt == UP){

snake.dot[0].y -= 10;

if(kbhit()){

int key = getch();

if(key == 'A' || key == 'a')

snake.tt = LEFT;

if(key == 'D' || key == 'd')

snake.tt = RIGHT;

if (key == 32){

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

key = getch();

while (key != 32){

key = getch();

delay(0);

}

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

delay(1000);

key = 'W';

}

}

}

else if(snake.tt == DOWN){

snake.dot[0].y += 10;

snake.dot[0].y += 10;

if(kbhit()){

int key = getch();

if(key == 'A' || key == 'a')

snake.tt = LEFT;

snake.tt = RIGHT;

if (key == 32){

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

key = getch();

while (key != 32){

key = getch();

delay(0);

}

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

delay(1000);

key = 'S';

}

}

}

else if(snake.tt == LEFT){

snake.dot[0].x -= 10;

if(kbhit()){

int key = getch();

if(key == 'W' || key == 'w')

snake.tt = UP;

while (key != 32){

key = getch();

delay(0);

}

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

delay(1000);

key = 'A';

}

}

}

else if(snake.tt == RIGHT){

snake.dot[0].x += 10;

if(kbhit()){

int key = getch();

if(key == 'S' || key == 's')

snake.tt = DOWN;

if(key == 'W' || key == 'w')

snake.tt = UP;

if (key == 32){

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

key = getch();

while (key != 32){

key = getch();

delay(0);

}

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

delay(1000);

key = 'D';

}

}

}

}

**4.7 Hàm GetHighScore**

Hàm này lấy ra điểm cao nhất trong file “ListHighScore.txt”.

int GetHightScore(){

FILE \*fp;

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

int temp = 0;

char \_temp[25];

fscanf(fp, "%s %d", \_temp, &temp);

return temp;

}

**4.8 Hàm CheckHighScore**

Hàm này sẽ kiểm tra xem số điểm người đang chơi có lớn hơn số điểm cao nhất trong file "ListHighScore.txt" hay không? Nếu có người chơi sẽ được nhập tên. Số điểm và tên người chơi sẽ được lưu thay thế vào file "ListHighScore.txt"

void CheckHighScore(int \_score){

if(\_score > getHightScore()){

FILE \*fp = fopen("ListHighScore.txt", "w+");

outtextxy(540,130,"Nhap ten: ");

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

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);

if(ch1 == 13){

fputs(\_name, fp);

fprintf(fp, " %d", \_score);

exit(0);

}

}

}

}

**4.9 Hàm XuLy (Xử lý)**

***Hàm này gồm 3 phần:***

Khi rắn đâm vào tường đầu rắn sẽ xuất hiện ở phía tường đối diện.

if (snake.dot[0].x >= 500) snake.dot[0].x = 10;

else if (snake.dot[0].x < 10) snake.dot[0].x = 500 - 10;

else if (snake.dot[0].y >= 300) snake.dot[0].y = 10;

else if (snake.dot[0].y < 10) snake.dot[0].y = 300 - 10;

Khi rắn cắn vào thân thì sẽ gọi đến hàm CheckHighScore và game kết thúc.

for (int i = 1; i < snake.n; i++)

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

outtextxy(250,250,"Thua roi!");

CheckHighScore((snake.n\*10 - 30));

while (getch() != 13);

}

Khi rắn ăn được hoa quả thì hoa quả sẽ xuất hiện ở vị trí khác.

if(snake.dot[0].x == hq.td.x && snake.dot[0].y == hq.td.y){

for(int i = snake.n; i > 0; i--)

snake.dot[i] = snake.dot[i - 1];

snake.n++;

if(snake.tt == UP)

snake.dot[0].y -= 10;

else if(snake.tt == DOWN)

snake.dot[0].y += 10;

else if(snake.tt == LEFT)

snake.dot[0].x -= 10;

else if(snake.tt == RIGHT)

snake.dot[0].x += 10;

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

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

}

**4.10 Hàm ShowHighScore**

Hàm có chức năng hiển ra ra người chơi điểm cao nhất.

void ShowHighScore(){

cleardevice();

int temp\_score = 0;

char key, score\_str[10], temp\_name[25];

FILE \*fp = fopen("ListHighScore.txt", "r");

fscanf(fp, "%s %d", temp\_name, &temp\_score);

sprintf(score\_str,"%d",temp\_score);

outtextxy(10,10, "Nhan ESC de tro lai menu!");

setcolor(YELLOW);

settextstyle(1,0,2);

outtextxy(250,200, "Ten");

outtextxy(370,200, "Diem");

outtextxy(250,250, temp\_name);

outtextxy(370,250, score\_str);

key = getch();

if(key == 27){

Menu();

}

}

**4.11 Hàm Run**

Trong vòng lặp while sẽ gọi ra hàm HienThi, DieuKhien, XuLy lần lượt để người chơi có thể chơi. Mỗi lần vòng lặp lặp lại hàm HienThi sẽ xóa đi màn hình cũ và HienThi mới sẽ được hiện ra với vị trí mới của snake, hoaqua và giao diện.

void run(){

Snake snake;

HoaQua hq;

KhoiTao(snake, hq);

while(1){

HienThi(snake, hq);

DieuKhien(snake);

XuLy(snake, hq);

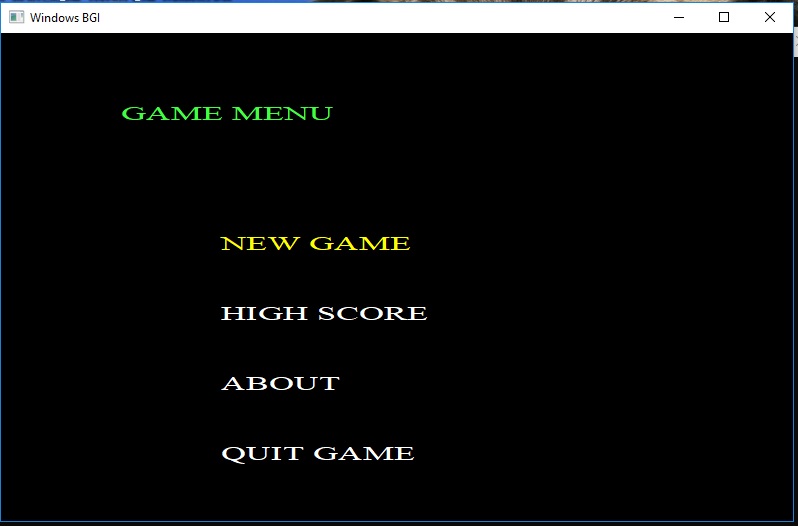
delay(100);

}

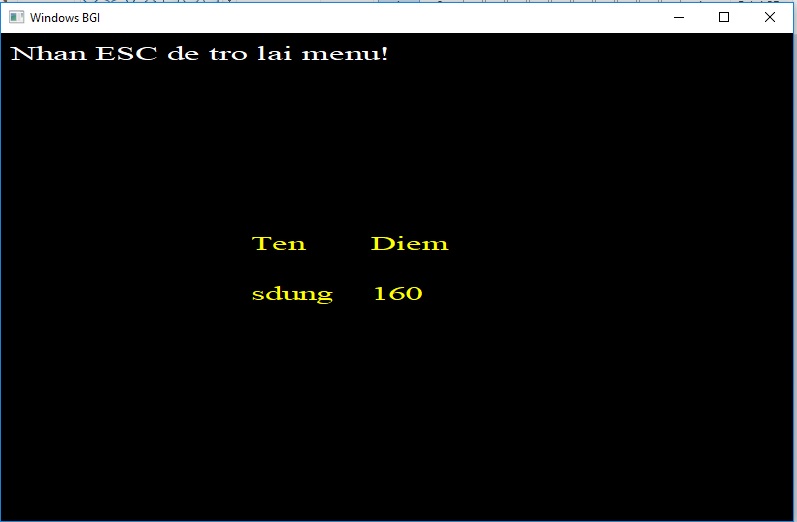
}

**PHẦN 5: MỘT SỐ HÌNH ẢNH TRONG GAME**

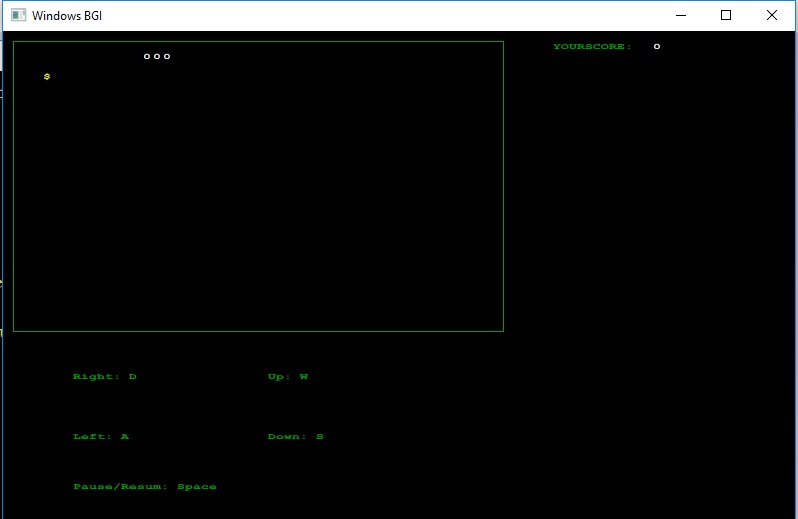
Menu Start Game



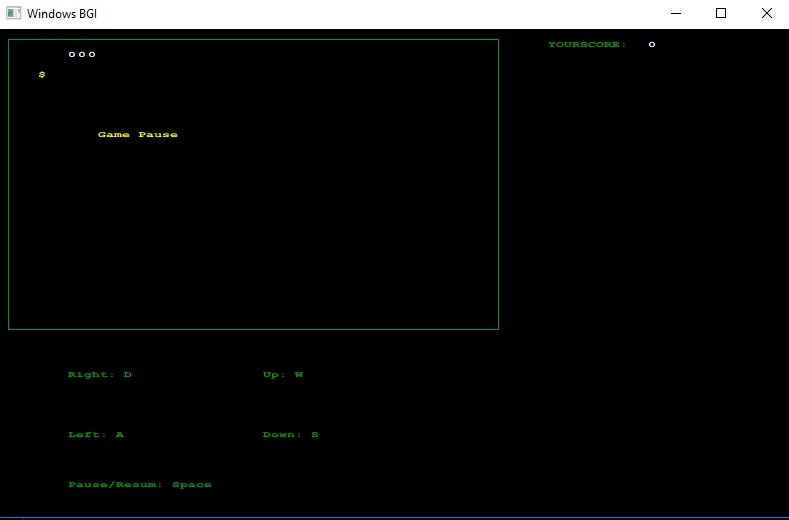
High Score



Màn hình game



Pause/ReSume



**KẾT LUẬN**

**Kết quả đạt được:**

Qua thời gian nghiên cứu và học tập, nhìn chung đã hoàn thành các mục tiêu, nhiệm vụ đề ra của trò chơi rắn săn mồi.

Xây dựng được một trò chơi hoàn chỉnh có tính giải trí cao.

**Ưu và nhược điểm của hệ thống:**

1. Ưu điểm:

- Game dễ sử dụng.

- Đầy đủ chức năng như game snake xenzia.

2. Nhược điểm:

- Màu sắc chưa bắt mắt.

- Chưa có âm thanh cho người chơi.

- Kĩ năng lập trình chưa cao.

**TÀI LIÊU THAM KHẢO**

1. Slide giáo trình môn Đồ Họa Máy Tính của trường ĐHĐL.

2. Sách Phạm Văn Ất; *Lập trình C/C++*; NXB Thống Kê; Hà Nội - 2004.

3. Các trang Web: Google, stackoverflow, daynhauhoc, youtube….