Nhận xét, đánh giá của giảng viên hướng dẫn:

Hà nội, ngày…., tháng 01, năm 2017

*ThS.Ninh Thị Thanh Tâm*

Nhận xét, đánh giá của giảng viên chấm bài:

Hà nội, ngày…., tháng 01, năm 2017

Giảng viên:

**Mục lục**

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 3](#_Toc502563465)

[1.1 Giới thiệu đề tài 3](#_Toc502563466)

[1.2 Mục đích, ý nghĩa của đề tài 3](#_Toc502563467)

[1.3 Phương án triển khai đề tài 3](#_Toc502563468)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 4](#_Toc502563469)

[2.1 Giới thiệu công cụ viết chương trình 4](#_Toc502563470)

[2.2 Giới thiệu về console 4](#_Toc502563471)

[2.3 Ngôn ngữ lập trình hướng đối tượng C++ cơ bản 5](#_Toc502563472)

[2.4 Tuần tự viết một game console cơ bản 6](#_Toc502563473)

[CHƯƠNG 3: PHÂN TÍCH CHƯƠNG TRÌNH 7](#_Toc502563474)

[3.1 Ngôn ngữ lập trình và các thành phần trong chương trình 7](#_Toc502563475)

[3.2 Các thành phần trong chương trình 7](#_Toc502563476)

[3.3 Cấu trúc của một game console cơ bản 8](#_Toc502563477)

[CHƯƠNG 4: QUÁ TRÌNH LẬP TRÌNH 11](#_Toc502563478)

[4.1 Thiết kế chương trình 11](#_Toc502563479)

[CHƯƠNG 5: KẾT QUẢ VÀ DEMO CỦA CHƯƠNG TRÌNH 15](#_Toc502563480)

[5.1 Kết quả demo 15](#_Toc502563481)

[CHƯƠNG 6: KẾT LUẬN 16](#_Toc502563482)

[6.1 Kết quả đạt được 16](#_Toc502563483)

[6.2 Những vấn đề còn tồn đọng 17](#_Toc502563484)

[6.3 Hướng phát triển của đề tài 17](#_Toc502563485)

# CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

## 1.1 Giới thiệu đề tài

* Hiện nay, công nghệ đang từng ngày phát triển, nhu cầu giải trí trên các thiết bị di động, laptop, PC tăng cao. Với mục đích, hướng đi trong tương lai của chúng em là lập trình ứng dụng di động, PC. Vậy nên, nhóm em chọn đề tài này với mục đích nghiên cứu về lập trình ứng dụng và game cơ bản, làm nền cho quá trình nghiên cứu và công việc tương lại sau này.
* Đề tài gồm 2 phần chính:
* Hỗ trợ học toán cơ bản:
* Giải nhanh chu vi và diện tích của một số hình học hai chiều.
* Dữ liệu được nhập vào từ bàn phím và kết quả hiển thị trên màn hình.
* Giải trí:
* Demo mini game rắn săn mồi.

## 1.2 Mục đích, ý nghĩa của đề tài

* Củng cố kiến thức đã học và nghiên cứu: tìm hiểu về màn hình console, lập trình game cơ bản.
* Xây dựng trò chơi rắn săn mồi (snake) trên môi trường Windows.

## 1.3 Phương án triển khai đề tài

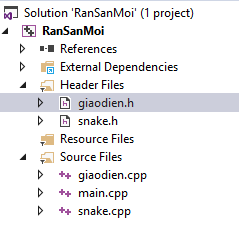
* Để hoàn thành đề tài này, nhóm em đã làm thao các bước:
* Lập kế hoạch thực hiện đề tài.
* Xây dựng chương trình (chia nhỏ từng chức năng của chương trình và hoàn thiện từng phần).
* Củng cố kết quả đạt được (bao gồm kiểm tra chương trình khi chạy, các chức năng của chương trình, chỉnh sửa khi xuất hiện lỗi và hoàn thiện chương trình ở mức có thể).
* Song song với quá trình trên là quá trình học hỏi, tìm kiếm, tích góp tài liệu và các kiến thức về các công cụ liên quan nhằm phục vụ xây dựng hoàn thiện đề tài.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1 Giới thiệu công cụ viết chương trình

***2.1.1 Phần mềm visual stdio 2017***

Microsoft **Visual Studio là** *một môi trường phát triển tích hợp (IDE) từ Microsoft. Nó được sử dụng để phát triển chương trình máy tính cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web*

**

## 2.2 Giới thiệu về console

***2.2.1 Tọa độ trong console***

Màn hình console là màn hình có thể hiển thị các ký tự dạng 2D, trên phông nền đen, chữ trắng mặc định. Màn hình console là tập hợp những ô nhỏ, mỗi ô có thể hiển thị tối đa một ký tự, và chúng cũng có một tọa độ riêng.

Consol chỉ hiển thị được các ký tự dạng 2D nên sẽ có một hệ trục tọa độ Oxy giống trong tọa độ đề các hình hai chiều toán học. Tuy nhiên gốc tọa độ trong console là ô đầu tiên trên cùng phía bên trái màn hình. Các ô càng nằm về bên phải thì x càng lớn (Ox có chiều tăng dần từ trái sang phải, và ngược lại). Các ô càng nằm về phía dưới màn hình thì y càng lớn (Oy tăng dần từ trên xuống dưới).

***2.2.2 Dịch chuyển tọa độ trên console***

Như đã nói, mỗi ô trong console đều có một tọa độ riêng, vậy nên ta có thể dịch chuyển từ ô này sang ô khác bằng cách thay đổi tọa độ.

Việc dịch tọa độ này giúp chúng ta có thể dễ dàng in ra ký tự ở những nơi mà chúng ta mong muốn.

***2.2.3 In màu sắc trên console***

Khi chạy chương trình, màu mặc định của console là ký tự trắng hiển thị trên nền đen. Tuy nhiên, để không bị đơn điệu, ta có thế sử dụng hàm màu tô màu cho các ký tự và cả màu nền của console.

## 2.3 Ngôn ngữ lập trình hướng đối tượng C++ cơ bản

***2.3.1 Class***

Mục đích chính của C++ là bổ sung thêm tính hướng đối tượng (object) cho ngôn ngữ lập trình C và các lớp (Class) là tính năng trung tâm của C++.

Một lớp (Class) được sử dụng để xác định form của một đối tượng và nó kết nối sự biểu diễn dữ liệu và các phương thức để thao tác dữ liệu đó vào một package gọn gàng. Dữ liệu và hàm bên trong một lớp được gọi là các thành viên của lớp đó.

***2.3.2 Hàm bạn, lớp bạn***

Hàm bạn (hàm friend) trong C++ của một lớp được định nghĩa bên ngoài phạm vi của lớp đó. Nhưng nó có quyền truy cập tất cả các thành viên trong private của lớp đó. Hàm bạn có thể là hàm thành phần của một lớp khác, cũng có thể là hàm tự do không thuộc lớp nào cả.

Lớp bạn, Khi lớp A khai báo lớp B là lớp bạn của lớp A thì tất cả các hàm thành phần trong lớp B được quyền tự do truy cập tất cả các thành viên trong private của lớp A. Nhưng lớp A không có quyền truy cập các thành viên trong private của lớp B, trừ khi lớp B cũng khai báo lớp A là bạn của nó.

***2.3.3 Chồng toán tử***

C++ cho phép bạn xác định nhiều hơn một định nghĩa cho một tên hàm hoặc một toán tử trong cùng phạm vi (scope), được gọi tương ứng là nạp chồng hàm (function overloading) và nạp chồng toán tử (operator overloading) trong C++.

Một khai báo nạp chồng là một khai báo mà đã được khai báo cùng tên như một khai báo được khai báo trước đó trong cùng phạm vi, ngoại trừ rằng: Cả hai khai báo có các tham số khác nhau và định nghĩa khác nhau.

Nạp chồng toán tử trong C++ là các hàm với tên đặc biệt: Tên hàm là từ khóa operator theo sau là ký hiệu của toán tử đang được định nghĩa. Giống như bất kỳ hàm khác, một toán tử nạp chồng có một kiểu trả về là một danh sách tham số.

***2.3.4 Tính kế thừa***

Một trong những khái niệm quan trọng nhất trong lập trình hướng đối tượng là tính kế thừa (inheritance). Tính kế thừa cho phép chúng ta định nghĩa một lớp trong điều kiện một lớp khác, mà làm cho nó dễ dàng hơn để tạo và duy trì một ứng dụng. Điều này cũng cung cấp một cơ hội để tái sử dụng tính năng code và thời gian thực thi nhanh hơn.

Khi tạo một lớp, thay vì viết toàn bộ các thành viên dữ liệu và các hàm thành viên mới, lập trình viên có thể nên kế thừa các thành viên của một lớp đang tồn tại. Lớp đang tồn tại nàu gọi là Base Class-lớp cơ sở, và lớp mới được xem như là Deriver Class-lớp thừa kế.

***2.3.5 Hàm ảo, hàm ảo thuần túy***

Hàm ảo là cơ chế của C++ cho phép cài đặt đa cấu hình động và được khai báo với từ khóa Virtual.

Hàm ảo thuần túy là hàm ảo Virtual với khai báo “=0” trong định nghĩa của nó.

Một lớp có chứa ít nhất một hàm thành viên là hàm ảo thuần túy được gọi là lớp trừu tượng.

## 2.4 Tuần tự viết một game console cơ bản

* Để viết được một game console cơ bản, phải làm theo tuần tự:
* Xây dựng ý tưởng viết game.
* Xác định các đối tượng trong game.
* Khởi tạo các đối tượng.
* Xây dựng vòng lặp main game.
* Điều khiển.
* Xử lý.
* Kết thúc game (thắng, thua).

# CHƯƠNG 3: PHÂN TÍCH CHƯƠNG TRÌNH

## 3.1 Ngôn ngữ lập trình và các thành phần trong chương trình

***3.1.1 Ngôn ngữ C/C++ và thư viện***

Để đạt được yêu cầu là thể hiện rõ nội dung môn học. Ngôn ngữ sử dụng trong chương trình là ngôn ngữ lập trình hướng đối tượng C++.

Đề tài là viết chương trình xây dựng phần mềm trên Windows. Windows đã được tích hợp sẵn môi trường lập trình trên ngôn ngữ C++.

Các thư viện sử dụng chính trong quá trình viết chương trình:

* #include <iostream> (thư viện nhập xuất trong C++)
* #include <connio.h> (thư viện chứa hàm kbhit() )
* #include <windows.h> (thư viện chứa hàm PlaySound() )
* #include <stdio.h> (thư viện chứa hàm putchar(), xuất các ký tự bảng mã ASCII)
* #include<ctime> (thư viện lấy time hiện tại của máy)
* #include <iomanip> ( Định nghĩa nhiều phép điều chỉnh mà nó lấy vào một đối số. )
* #include<string> (Định nghĩa thùng chứa lớp tiêu bản basic\_string và nhiều hỗ trợ cho các tiêu bản.khung)

## 3.2 Các thành phần trong chương trình

Chương trình gồm 1 đối tượng và được viết trong 1 class.

SNACK

1. **enum** Trangthai { UP, DOWN, LEFT, RIGHT };
2. **struct** Toado  // cấu trúc tạo độ gôm trục ox, oy
3. {
4. **int** x, y;
5. };
6. **struct** Hoaqua  // hoa quả con rắn sẽ ăn có tọa độ
7. {
8. Toado td;
9. };
10. **struct** Snake  // cấu trúc của con rắn
11. {
12. Toado dot[31];
13. **int** n;
14. **int** diem = 0;
15. Trangthai tt;
16. };

Các Phương thức

1. **void** vaogame();  // Vào game
2. **void** khoitao(Snake &p, Hoaqua &hq);  // Khởi tạo con rắn
3. **void** hienthi(Snake p, Hoaqua hq);  // Hiển thị
4. **void** Dieukhuyen(Snake &p);  // Điều khiển con rắn
5. **int** Xuly(Snake &p, Hoaqua &hq, **int** &ThoiGianSleep); // xử lý thắng thua

## 3.3 Cấu trúc của một game console cơ bản

***3.2.1 Vòng lặp game console***

Trong chương trình chúng em sử dụng vòng lặp “while()” với điều kiện là một hằng số (while(const)) Như vậy, vòng lặp này sẽ lặp vô hạn, cho đến khi chúng ta cho dừng bằng lệnh “break;” (khi thắng, thua hoặc thoát chương trình).

* Cấu trúc vòng lặp:

1. **while** (1)
2. {
3. AnConTro(); // Ẩn con trỏ ở màn hình console
4. hienthi(p, hq); //Hiển thị
5. Dieukhuyen(p); // Điều khiển con rắn
6. {
7. ma = Xuly(p, hq, ThoiGianSleep); // xử lí nếu con rắn đâm vào thành
8. hoặc vào đuôi
10. **if** (ma == 1)
11. {
12. ve();
13. gotoXY(32, 1);
14. TextColor(14);
15. cout << "Score:" << p.diem;  // Trả lại điểm
16. **while** (\_getch() != 13);
17. **break**;  // thoát khỏi vòng lặp while
18. }
19. Sleep(ThoiGianSleep);
20. }
21. }

Với vòng lặp này chúng ta cần sử dụng hàm xóa màn hình để mỗi lần lặp, màn hình console được xóa rồi hiển thị lại đối tượng ở vị trí khác.

***3.2.2 Cấu trúc game snake***

Cấu trúc game snake nhóm e xây dựng có sử dụng vòng lặp game console:

* Cấu trúc chương trình chính game snake:

1. **while** (**true**)
2. {
3. // Hiện thời gian
4. //Hiện khung game
5. **while** (**true**)
6. {
7. **if** (GetAsyncKeyState(VK\_UP) != 0)
8. {
9. //Bắt phím lên
10. }
11. **else** **if** (GetAsyncKeyState(VK\_DOWN) != 0)
12. {
13. //Bắt phím Xuống
14. }
15. **else** **if** (GetAsyncKeyState(VK\_RETURN) != 0)
16. ////Bắt phím enter
17. {
18. **switch** (contro)
19. {
20. **case** 0:
21. {
22. //chọn vào chơi game
23. }
24. } **break**;
25. **case** 1:
26. {
27. //Xem Thông tin game
28. } **break**;
29. **case** 2:
30. {
31. //Thoát game
32. **return** 0;
33. } **break**;
34. }
35. **break**;
36. }
37. }
38. }

Hàm main cho phép người chơi chọn menu có 3 lựa chọn là

* Vào chơi game
* Xem Thông tin
* Thoát game

***3.2.3 Nội dung game snake***

Trò chơi mô phỏng quá trình tìm đồ ăn của một con rắn. Trò chơi chỉ có 1 chế độ chơi duy nhất là tính điểm. Người chơi khi bắt đầu chỉ có duy nhất một mạng để chơi, và con rắn chỉ có 1 đốt. Đồ ăn sẽ xuất hiện randun trong màn hình game, nhiệm vụ là phải ăn chúng và người chơi sẽ dành được 5 điểm khi ăn được một đồ ăn, con rắn sẽ dài thêm một đốt.

Độ khó của trò chơi: Khi độ dài con rắn dài 1 đốt thì biến TGSleep sẽ giảm 20 (tốc độ game sẽ nhanh dần).Cho đến khi >30 thì không giảm nữa

Trò chơi kết thúc: Khi người chơi điều khiển con rắn đâm vào tường (viền của màn hình game) hoặc đâm vào một trong các đốt của con rắn (trừ đốt đầu) thì trò chơi kết thúc, người chơi thua.

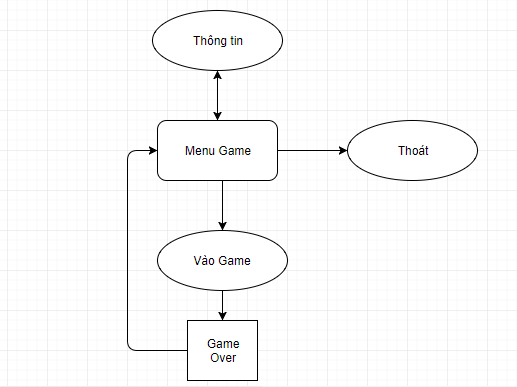
***3.2.4 Âm thanh***

Trương trình sử dụng các đoạn code âm thanh có sẵn (tiếng bip).

# CHƯƠNG 4: QUÁ TRÌNH LẬP TRÌNH

## 4.1 Thiết kế chương trình

Ý tưởng khung của toàn bộ ương trình:



* Đây là bộ khung cần thiết để chúng ta xây dựng các phần trong chương trình.

**4.2 Vòng lặp chương trình**

Trong toàn bộ chương trình có 2 vòng lặp, những vòng lặp này có cấu trúc như phần vòng lặp game.

* Vòng lặp chính:
* Là vòng lặp tại menu vào game và game over
* vòng xem thông tin
* Là những vòng lặp khi ấn vào xem thông tin và back lại

**4.3 Nội dung chương trình**

***4.3.1 Hỗ trợ học toán cơ bản***

*4.3.1.1 Vị trí nhập và xuất dữ liệu trên giao diện console*

* Giao diện chương trình và vị trí nhập xuất.

**Vị trí hiển thị phần chơi game**

**Thông báo game over và back**

* Hàm di chuyển tọa độ (hàm có sẵn)

1. **void** gotoXY(**int** x, **int** y)
2. {
3. COORD coord;
4. coord.X = x;
5. coord.Y = y;
6. SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coord);
7. }

***4.3.2 Mini game snake console***

*4.3.2.1 Khởi tạo vị trí của các đối tượng trong game*

* Khởi tạo con rắn:

Vì con rắn có nhiều đốt nên mỗi đốt sẽ có một tọa độ, vậy nên phải khai báo tọa độ dưới dạng struct. Ngoài ra còn một số biến độ dài, điểm và biến trạng thái.

* Ta có các thành viên trong private:

1. **enum** Trangthai { UP, DOWN, LEFT, RIGHT };
2. **struct** Toado
3. {
4. **int** x, y;
5. };
6. **struct** Hoaqua
7. {
8. Toado td;
9. };
10. **struct** Snake
11. {
12. Toado dot[31];
13. **int** n;
14. **int** diem = 0;
15. Trangthai tt;
16. };

* Hàm khởi tạo: con rắn và đồ ăn

1. p.n = 1;
2. p.dot[0].x = 0;
3. p.dot[0].y = 0;
4. p.tt = RIGHT;
6. hq.td.x = rand() % consoleWidth / 4;
7. hq.td.y = rand() % consoleHeight;

*4.3.2.2 Cập nhật trạng thái của các đối tượng trong game*

* Cập nhật trạng thái con rắn

Con rắn được hiển thị dạng mảng 1 chiều, khi di chuyển các phần tử phía sau sẽ đi theo phần tử đầu tiên. Vậy cập nhật trạng thái ta chỉ cần cập nhật đốt đầu tiên.

* Hàm điều khiển và cập nhật:

1. **void** SNAKE::DieuKhien(Snake &p)
2. {
3. **for** (**int** i = p.n - 1; i > 0; i--)
4. p.dot[i] = p.dot[i - 1];
6. **if** (GetAsyncKeyState(VK\_LEFT))
7. p.tt = LEFT;
8. **else** **if** (GetAsyncKeyState(VK\_RIGHT))
9. p.tt = RIGHT;
10. **else** **if** (GetAsyncKeyState(VK\_UP))
11. p.tt = UP;
12. **else** **if** (GetAsyncKeyState(VK\_DOWN))
13. p.tt = DOWN;
15. **if** (p.tt == UP)
16. p.dot[0].y--;
17. **else** **if** (p.tt == DOWN)
18. p.dot[0].y++;
19. **else** **if** (p.tt == LEFT)
20. p.dot[0].x--;
21. **else** **if** (p.tt == RIGHT)
22. p.dot[0].x++;
23. }

*4.3.2.3 Kiểm tra rắn ăn được trái cây không*

Khi con rắn ăn được đồ ăn tức là tọa độ đầu của con rắn bằng

tọa độ hiện tại của đồ ăn.

Sau khi con rắn ăn được đồ ăn ngay lập tức đồ ăn sẽ xuất hiện randun ở vị trí mới. Để làm việc này, ta sử dụng hàm rand() cho hai biến tọa dộ của đồ ăn. Và đồng thời độ dài con rắn tăng lên 1.

* Hàm kiểm tra

1. **int** SNAKE::Xuly(Snake &p, Hoaqua &hq, **int** &ThoiGianSleep)
2. {
3. **if** (p.dot[0].x < 0 || p.dot[0].x >= (consoleWidth / 4) ||
4. p.dot[0].y < 0 || p.dot[0].y >= consoleHeight)
5. **return** 1;
7. **for** (**int** i = 1; i < p.n; i++)
9. **if** (p.dot[0].x == p.dot[i].x &&
10. p.dot[0].y == p.dot[i].y)
11. **return** 1;

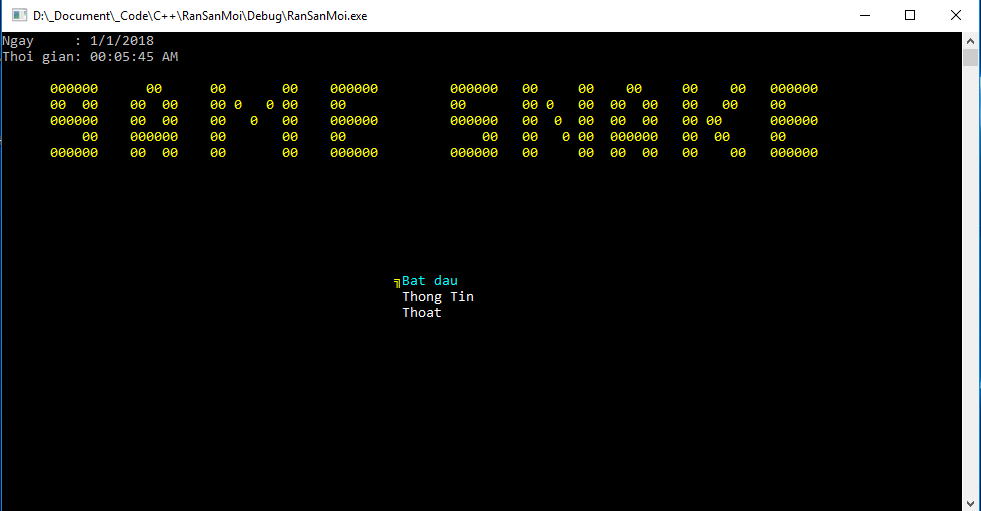

15. **if** (p.dot[0].x == hq.td.x && p.dot[0].y == hq.td.y)
16. {
17. Beep(2300, 200);
18. **for** (**int** i = p.n; i > 0; i--)
19. p.dot[i] = p.dot[i - 1];
21. p.n++;
22. p.diem += 5;
24. **if** (p.tt == UP)
25. p.dot[0].y--;
26. **else** **if** (p.tt == DOWN)
27. p.dot[0].y++;
28. **else** **if** (p.tt == LEFT)
29. p.dot[0].x--;
30. **else** **if** (p.tt == RIGHT)
31. p.dot[0].x++;
33. hq.td.x = rand() % (consoleWidth / 4);
34. hq.td.y = rand() % consoleHeight;
35. **if** (ThoiGianSleep > 30)
36. {
37. ThoiGianSleep -= 20;
38. }
40. }
41. TextColor(14);
42. gotoXY(32, 1);
43. cout << "Score: " << p.diem;
45. **return** 0;
46. }

Hàm này có giá trị trả về là 0 hoặc 1, nếu trả về 1 tức là con rắn đã bị đâm vào tường hoặc vào thân mình lúc đó vòng while ở trên sẽ bị break và kết thúc game

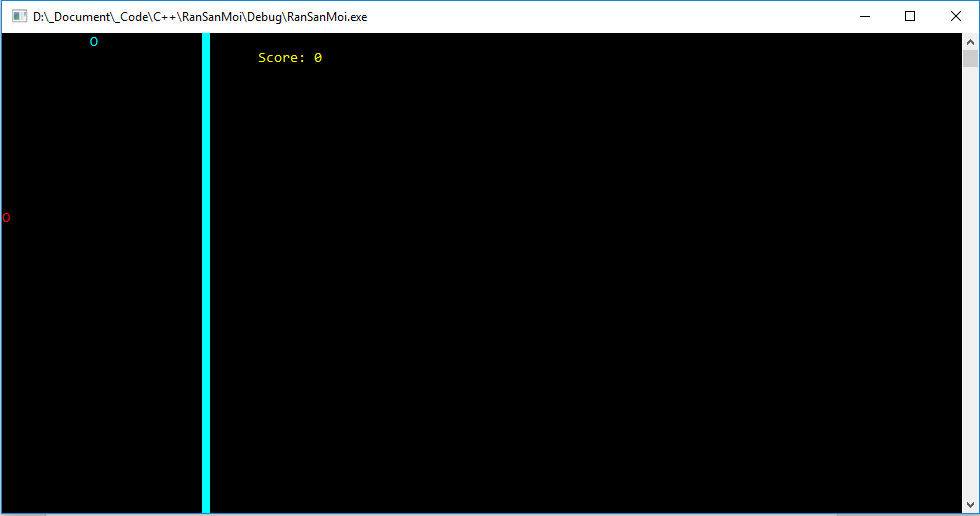
# CHƯƠNG 5: KẾT QUẢ VÀ DEMO CỦA CHƯƠNG TRÌNH

## 5.1 Kết quả demo

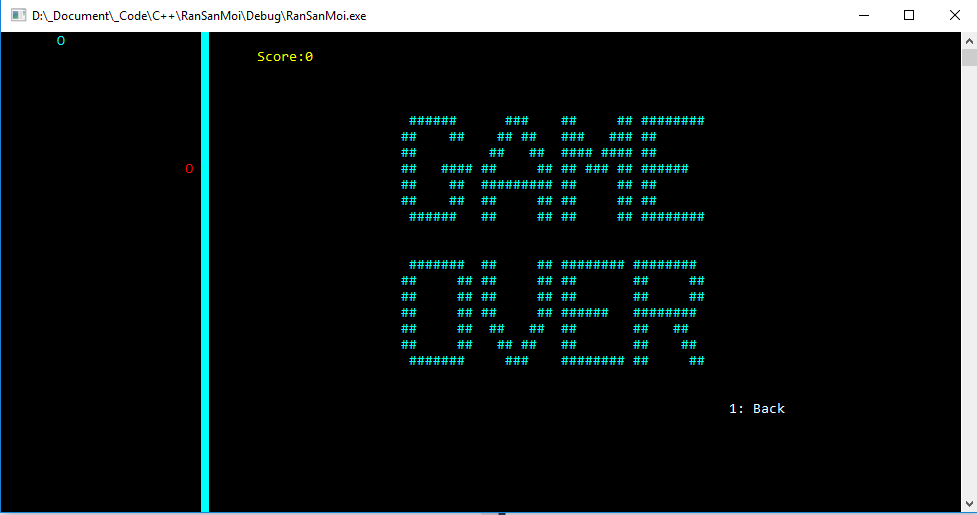
* *Giao diện menu lựa chọn:*



* *Giao diện game snake*



* Giao diện Game Over



# CHƯƠNG 6: KẾT LUẬN

## 6.1 Kết quả đạt được

* Tìm hiểu và xây dựng được game cơ bản trên console
* Biết được cấu trúc của game
* Tìm hiểu thêm được nhiều thư viện, hàm, câu lệnh cho việc xây dựng chương trình.
* Hiểu rõ hơn về môn lập trình hướng đối tượng C++.

## 6.2 Những vấn đề còn tồn đọng

* Các tính năng của chương trình còn khá đơn giản.
* Chưa sử dụng đồ họa vào chương trình.
* Chưa tìm hiểu sâu được hàm phát nhạc, chưa phát lại khi file kết thúc.

## 6.3 Hướng phát triển của đề tài

* Xây dựng thêm tính năng mới của game như nhập, xuất tên người chơi, lưu danh sách người chơi có điểm cao nhất. Sử dụng công cụ lập trình có thể thiết kế đồ họa cho game. Mở rộng lập trình sang các hệ điều hành khác như Android, Ios, Linux và nhiều hệ điều hành khác nữa.
* **Lời cảm ơn:**
* Chúng em xin chân thành cảm ơn các thầy cô trong khoa CNTT
* Xin chân thành cảm ơn cô: Ninh Thị Thanh Tâm đã tận tình chỉ bảo chúng em trong suốt thời gian học môn C++.
* Cuối cùng xin cảm ơn các bạn trong khoa CNTT, những người giúp đỡ, chia sẽ kiến thức, kinh nghiệm, tài liệu… trong suốt quá trình nghiên cứu thực hiện đề tài.
* **Lời cam đoan:**
* Những nội dung trong báo cáo này là do nhóm chúng em thực hiện.
* Mọi tham khảo dùng trong báo cáo này đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.
* Mọi sao chép không hợp lệ, vi phạm quy chế đào tạo, hay gian trá chúng em xin hoàn toàn chịu trách nhiệm.

*Sinh viên thực hiện:*

**Dương thị thu hiền - 1512102117**

**Doãn văn dũng - 1412102227**

**Nguyễn hữu tiến - 1412102208**

* **Tài liệu tham khảo:**

1. Trang web:<https://www.stdio.vn/articles/read/91/chay-file-wav-voi-windowsh>
2. Trang web: <http://www.cplusplus.com/forum/beginner/89434/>
3. Trang web: <http://diendan.congdongcviet.com/threads/t5823::xuat-am-thanh-trong-cpp.cpp>
4. TS.Nguyễn Mạnh Hùng: Bài giảng Ngôn ngữ lập trình C++.
5. Nicolai M.Josuttis: The C++ Standard Library: A Tutorial and Reference (2nd Edition).
6. Một số tài liệu khác và kinh nghiệm thực tiễn.