

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**  
**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**  
**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN**  
**ĐỀ TÀI : LẬP TRÌNH GAME RẮN SẴN MỖI**

Giảng viên hướng dẫn: ThS. TRẦN THỊ DUNG

Sinh viên thực hiện: ĐẶNG NGỌC MINH

NGUYỄN VĂN ĐỊA LỢI

LÊ TRUNG KHANG

NGUYỄN NGUYỄN HOÀNG ANH

Lớp: CQ.62.CNTT

Khóa: K62

Tp. Hồ Chí Minh, năm 2022

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**  
**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**  
**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN**  
**ĐỀ TÀI : LẬP TRÌNH GAME RẰN SẰN MỒI**

Giảng viên hướng dẫn: ThS. TRẦN THỊ DUNG

Sinh viên thực hiện: ĐẶNG NGỌC MINH

NGUYỄN VĂN ĐỊA LỢI

LÊ TRUNG KHANG

NGUYỄN NGUYỄN HOÀNG ANH

Lớp: CQ.62.CNTT

Khóa: K62

Tp. Hồ Chí Minh, năm 2022

## NHIỆM VỤ THIẾT KẾ BÀI TẬP LỚN

### BỘ MÔN: CÔNG NGHỆ THÔNG TIN

-----\*\*\*-----

#### 1. Tên đề tài

LẬP TRÌNH GAME RẮN SẴN MỖI

#### 2. Mục đích, yêu cầu

a. Mục đích.

- Giúp người chơi thư giãn sau những giờ làm việc căng thẳng.
- Trải nghiệm và tiếp thu những kiến thức, kinh nghiệm trong quá trình thực hiện bài tập lớn.

b. Yêu cầu.

- Giao diện: Thân thiện, ưa nhìn, dễ sử dụng, cấu hình nhẹ.
- Tổ chức code cấu trúc - bố cục rõ ràng.

#### 3. Nội dung và phạm vi đề tài

a. Nội dung đề tài.

- Tổng quan về ngôn ngữ C.
- Tổng quan về game rắn sắn mỗi.
- Phân tích game rắn sắn mỗi.

+ Hiển thị giao diện game.

+ Thuật toán điều khiển con rắn.

+ Các hàm được sử dụng trong game.

b. Phạm vi đề tài.

- Tạo ra game rắn sắn mỗi với các chức năng đã nêu ra trong phần nội dung.

#### 4. Công nghệ, công cụ và ngôn ngữ lập trình

a. Công nghệ: Github.

b. Công cụ: Dev C, Visual studio code.

c. Ngôn ngữ lập trình: C.

## **5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng**

\* Quyền báo cáo đề tài bài tập lớn được chia thành 4 phần như sau:

### **- Chương 1: Mở đầu**

- Tổng quan về đề tài.
- Mục tiêu nghiên cứu.
- Phạm vi.

### **- Chương 2: Cơ sở lí thuyết.**

- Tổng quan về ngôn ngữ lập trình C.
- Tìm hiểu về Github.
- Tìm hiểu về thư viện đồ họa.

### **- Chương 3: Phân tích đề tài.**

- Giới thiệu về game rắn săn mồi.
- Phân tích các chức năng cần có trong chương trình.
- Hiển thị giao diện game.
- Các hàm được sử dụng trong game.

### **- Chương 4: Kết luận và hướng phát triển.**

- Đưa ra kết quả thực hiện được.
- Đưa ra nhận xét và bổ sung những chức năng và ứng dụng thiếu sót cần phát triển.

## **6. Giáo viên và cán bộ hướng dẫn**

## LỜI CẢM ƠN

Lời đầu tiên em xin phép gửi đến Quý Thầy Cô của **Bộ môn Công Nghệ Thông Tin trường Đại học Giao thông Vận tải - Phân hiệu tại Tp Hồ Chí Minh** lời chúc sức khỏe và lời cảm ơn sâu sắc nhất. Thầy cô đã tận tình hướng dẫn, truyền đạt kiến thức, kinh nghiệm cho em trong suốt quá trình học tập và rèn luyện ở trường đại học giao thông vận tải phân hiệu tại Tp Hồ Chí Minh, chúc thầy cô luôn mạnh khỏe và thu được nhiều thành công hơn nữa trong tương lai.

Đặc biệt, em xin gửi lời cảm ơn đến cô Trần Thị Dung, cô đã tận tình hướng dẫn chỉ bảo để em có thể tiếp cận với nhiều bài toán thực tế để hoàn thành bài tập lớn “**Lập trình game rắn săn mồi**”.

Mặc dù nhóm em đã cố gắng hết sức trong quá trình thực hiện bài tập lớn này, nhưng vì thời gian có hạn và nhóm em cũng chưa có kinh nghiệm làm việc với bài toán thực tế, nên chắc chắn sẽ có nhiều thiếu sót.

Lời sau cùng, em xin kính chúc thầy cô trong bộ môn **Công Nghệ Thông Tin trường Đại học Giao thông Vận tải - Phân hiệu tại Tp Hồ Chí Minh** luôn mạnh khỏe và thu được nhiều thành công hơn nữa trong tương lai.

Em xin chân thành cảm ơn !

## NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày ... tháng ... năm 2022

Giảng viên hướng dẫn

# MỤC LỤC

CHƯƠNG 1: MỞ ĐẦU .....	8
1. Tổng quan về đề tài.....	8
1.1. Mục tiêu nghiên cứu. ....	8
1.2. Phạm vi. ....	9
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	10
1. Tổng quan về ngôn ngữ lập trình C .....	10
1.1. Giới thiệu về ngôn ngữ lập trình C .....	10
1.2. Ưu điểm: .....	10
1.3. Nhược điểm:.....	10
2. Tìm hiểu về Github .....	11
2.1. Khái niệm:.....	11
2.2. Tính năng của Github.....	11
3. Tìm hiểu về thư viện đồ họa .....	11
CHƯƠNG 3: PHÂN TÍCH ĐỀ TÀI.....	12
1. Giới thiệu game rắn săn mồi:.....	12
2. Phân tích các chức năng cần có trong chương trình. ....	12
2.1. Hiện thị giao diện game: .....	12
1.1.1.    Giao diện ban đầu: .....	12
1.1.2.    Giao diện menu:.....	12
Giao diện chơi thường: .....	14
1.1.3.    Giao diện mê cung: .....	14
1.1.4.    Giao diện khi thua:.....	15
2.2. Các hàm được sử dụng trong game: .....	15
1.1.1.    Tạo giao diện khi khởi động game: .....	15
2.3. Khởi tạo trò chơi: .....	18
1.1.1.    Vẽ tường: .....	18

1.1.2.	Về rắn:.....	20
1.1.3.	Xử lí rắn .....	20
1.1.4.	Điều kiện thua: .....	21
1.1.5.	Hướng di chuyển của rắn .....	22
1.1.6.	Điều khiển rắn.....	22
1.1.7.	Tạo quả.....	23
1.1.8.	Rắn đề lên quả.....	23
1.1.9.	Rắn ăn quả.....	24
1.1.10.	Xử lý khi rắn ăn quả.....	24
1.1.11.	Hàm xóa con trỏ: Xóa vị trí con trỏ theo sau rắn.....	24
1.1.12.	Câu lệnh khác.....	25
1.1.13.	Tạo file lưu điểm cao nhất của người chơi: .....	25
1.1.14.	Chia tách file để dễ dàng quản lý:.....	25
CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....		26
1.	Kết luận .....	26
1.1.	Kết quả đạt được .....	26
1.2.	Kiến nghị.....	26
1.3.	Hướng phát triển .....	26
PHỤ LỤC.....		27
1.	Phụ lục 1: Hướng dẫn cài đặt.....	27
2.	Phụ lục 2: Hướng dẫn sử dụng .....	27
TÀI LIỆU THAM KHẢO.....		28



## **DANH MỤC CHỮ VIẾT TẮT**

# CHƯƠNG 1: MỞ ĐẦU

## 1. Tổng quan về đề tài.

Ngày nay công nghệ thông tin trở thành một lĩnh vực mũi nhọn trong công cuộc phát triển kinh tế xã hội. Cùng với công nghệ sinh học và năng lượng mới, công nghệ thông tin vừa là công cụ, vừa là động lực thúc đẩy quá trình công nghiệp hóa, hiện đại hóa đất nước. Và đồng thời công nghệ thông tin cũng đóng góp một phần cực kỳ quan trọng trong ngành công nghiệp game trị giá hàng trăm tỷ đô trên thế giới, và ở Việt Nam ngành công nghiệp game cũng đang cho thấy tiềm năng phát triển vô cùng mạnh mẽ.

Để có được những tựa game hiện đại như ngày nay phải trải qua nhiều lần cải tiến phát triển. Và một trong những tựa game đầu tiên đã đặt nền móng cho sự phát triển của ngành công nghiệp game hiện nay chính là game “Rắn săn mồi”. Qua nghiên cứu và tìm hiểu nhóm em nhận thấy game “Rắn săn mồi” là một tựa game mobile, chơi trên thiết bị di động với nhiệm vụ của người chơi là điều khiển con rắn ăn các khối nhỏ hơn để cơ thể dài ra và tránh va chạm vào tường/vật cản hoặc chính thân thể mình. Nó lần đầu xuất hiện vào năm 1997 trên chiếc điện thoại di động Nokia 6110. Ở phiên bản này, khi đó trò chơi vẫn được hoạt động trên một màn hình đen trắng. Tạo hình con rắn và các vật phẩm trong game đều ở dạng các khối vuông 8 bit. Tựa game tuy đã có tuổi đời lâu năm nhưng luôn để lại nhiều kỷ niệm, đặc biệt là đối với các thế hệ 8x, 9x. Do đó nhóm em quyết định xây dựng chương trình game “Rắn săn mồi”. Trong quá trình phân tích và thiết kế do kỹ năng và kiến thức có hạn nên chương trình không tránh khỏi những thiếu sót. Vì vậy chúng em rất mong sự góp ý của thầy cô.

### 1.1. Mục tiêu nghiên cứu.

- Tổng quan về game rắn săn mồi.
- Nghiên cứu về ngôn ngữ lập trình C, thư viện đồ họa.
- Phân tích và thiết kế: Giúp cho việc xây dựng và thực hiện trở nên dễ dàng và rõ ràng hơn.
- Cấu trúc và bố cục về game rắn săn mồi:

- + Khởi tạo và hiển thị tất cả mọi thứ liên quan về game.
- + Lập trình theo thuật toán điều khiển con rắn.
- + Lập trình theo thuật toán xử lý bài toán về game.

## 1.2. Phạm vi.

Xây dựng game “Rắn săn mồi” bằng ngôn ngữ C có các chức năng đã được nêu ra trong phần nội dung.

Để hiểu rõ hơn về game và một số kiến thức cần thiết về đồ họa trong game, nhóm em đã tham khảo cuốn tài liệu “Lập trình C nâng cao” của Phạm Văn Ất.

Cấu trúc báo cáo bài tập lớn.

### Chương 1: Mở đầu

- Tổng quan về đề tài.
- Mục tiêu nghiên cứu.
- Phạm vi.

### Chương 2: Cơ sở lý thuyết

- Tổng quan về ngôn ngữ lập trình C.
- Tìm hiểu về Github.
- Tìm hiểu về thư viện đồ họa.

### Chương 3: Phân tích đề tài

- Giới thiệu game rắn săn mồi.
- Hiển thị giao diện game.
- Các hàm được sử dụng trong game.

### Chương 4: Kết luận và hướng phát triển.

- Đưa ra kết quả thực hiện được.
- Đưa ra nhận xét và bổ sung chức năng thiếu sót cần phát triển.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 1. Tổng quan về ngôn ngữ lập trình C

#### 1.1. Giới thiệu về ngôn ngữ lập trình C

Ngôn ngữ lập trình C là một ngôn ngữ mệnh lệnh được phát triển từ đầu thập niên 1970 bởi Dennis Ritchie để dùng trong hệ điều hành UNIX.

Ngôn ngữ C có những đặc điểm cơ bản sau:

- Tính cô đọng (compact): C chỉ có 32 từ khóa chuẩn và 40 toán tử chuẩn, nhưng hầu hết đều được biểu diễn bằng những chuỗi ký tự ngắn gọn.
- Tính cấu trúc (structured): C có tập hợp những chỉ thị của lập trình như cấu trúc lựa chọn, lặp... Từ đó các chương trình viết bằng C được tổ chức rõ ràng, dễ hiểu.
- Tính tương thích (compatible): C có bộ tiền xử lý và một thư viện chuẩn vô cùng phong phú nên khi chuyển từ máy tính này sang máy tính khác các chương trình viết bằng C vẫn hoàn toàn tương thích.
- Tính linh động (flexible): C là một ngôn ngữ rất uyển chuyển và cú pháp, chấp nhận nhiều cách thể hiện, có thể thu gọn kích thước của các mã lệnh làm chương trình chạy nhanh hơn.
- Biên dịch (compile): C cho phép biên dịch nhiều tập tin chương trình riêng rẽ thành các tập tin đối tượng (object) là liên kết (link) các đối tượng đó lại với nhau thành một chương trình có thể thực thi được (executable) thống nhất.

#### 1.2. Ưu điểm:

- Ngôn ngữ mạnh, mềm dẻo và có thể truy nhập vào hệ thống, nên thường được sử dụng để viết hệ điều hành, các trình điều khiển thiết bị, đồ họa,...
- Có cấu trúc module, từ đó ta có thể phân hoạch hay chia nhỏ chương trình để tăng tính hiệu quả, rõ ràng, dễ kiểm tra trong chương trình.

#### 1.3. Nhược điểm:

Vì là một ngôn ngữ mềm dẻo, đó là do việc truy nhập tự do vào dữ liệu, trộn lẫn các dữ liệu,... Từ đó, dẫn đến sự lạm dụng và sự bất ổn của chương trình.

## **2. Tìm hiểu về Github**

### **2.1. Khái niệm:**

GitHub là một hệ thống quản lý dự án và phiên bản code, hoạt động như một mạng xã hội cho lập trình viên. Các lập trình viên có thể clone lại mã nguồn từ một repository và Github chính là một dịch vụ máy chủ repository công cộng, mỗi người có thể tạo tài khoản trên đó để tạo ra các kho chứa của riêng mình để có thể làm việc.

### **2.2. Tính năng của Github**

Github được coi là một mạng xã hội dành cho lập trình viên lớn nhất và dễ dùng nhất với các tính năng cốt lõi như:

- Wiki, issue, thống kê, đổi tên project, project được đặt vào namespace là user.

- Watch project: theo dõi hoạt động của project của người khác. Xem quá trình người ta phát triển phần mềm thế nào.

- Follow user: theo dõi hoạt động của người khác.

Lợi ích của Github đối với lập trình viên

- Quản lý source code dễ dàng.

- Tracking sự thay đổi qua các version.

- Giúp cải thiện kỹ năng code, thậm chí là tracking bug.

- Là một kho tài nguyên tuyệt vời.

## **3. Tìm hiểu về thư viện đồ họa**

Thư viện đồ họa chứa các hàm, các câu lệnh về màu nền, màu chữ như hàm SetCoLoR, TextColor. Ngoài ra còn có các hàm như là gotoXY() để dịch con trỏ hiện tại đến điểm có tọa độ (x, y); ShowCur() làm ẩn trỏ chuột; inputKey() trả về mã phím người dùng bấm.

Nhờ có thư viện đó nên việc code game dễ dàng, thuận tiện hơn.

## CHƯƠNG 3: PHÂN TÍCH ĐỀ TÀI

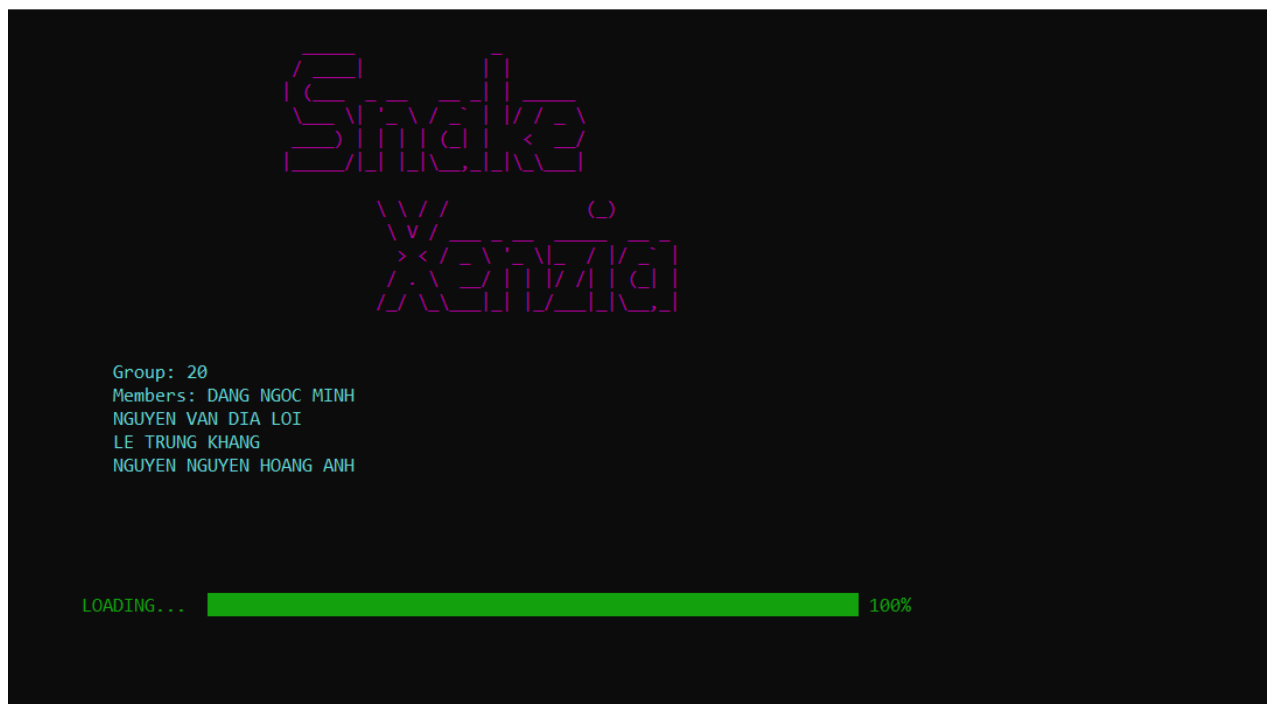
### 1. Giới thiệu game rắn săn mồi:

Rắn săn mồi là một tựa game chơi trên thiết bị di động. Với nhiệm vụ của người chơi là điều khiển con rắn ăn các khối thức ăn để cơ thể dài ra và tránh va chạm vào tường hoặc vật cản hoặc chính thân thể của mình. Đây là một trò chơi được ghi nhận thành tích qua số điểm và ai đạt điểm càng cao thì sẽ có thứ hạng càng cao.

### 2. Phân tích các chức năng cần có trong chương trình.

#### 2.1. Hiển thị giao diện game:

##### 1.1.1. Giao diện ban đầu:



##### 1.1.2. Giao diện menu:

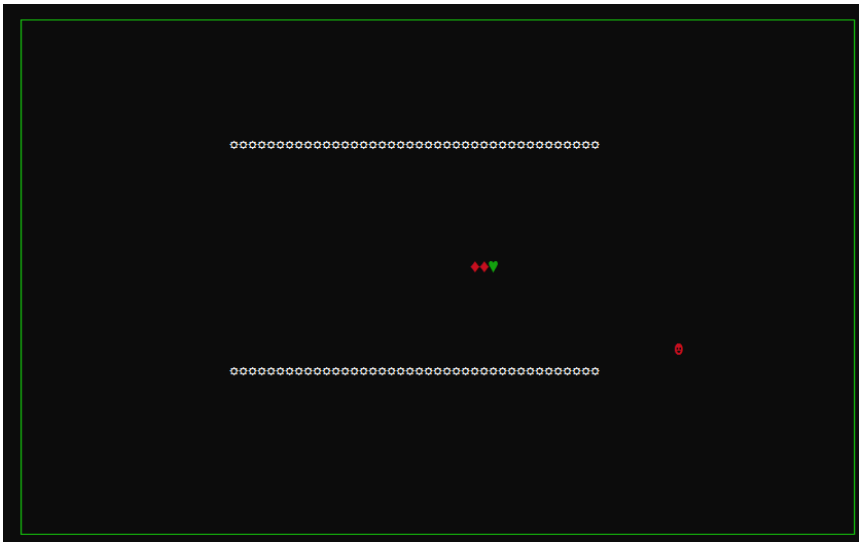
CHOI THUONG  
ME CUNG

GAME MENU  
TUTORIAL

1.1.3. Giao diện chơi thường:

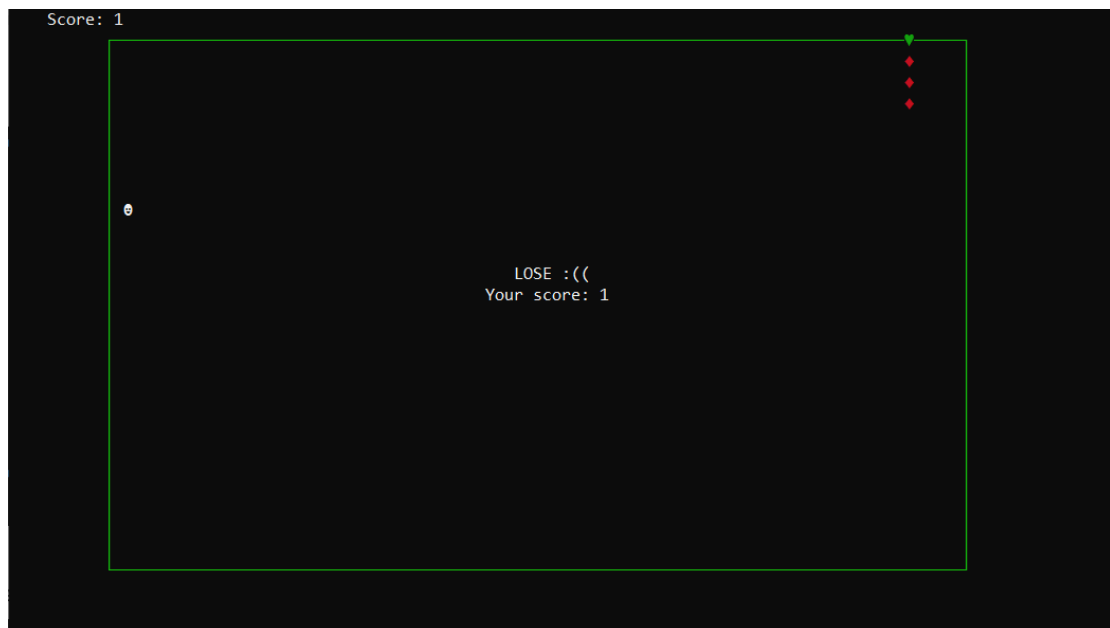


1.1.4. Giao diện mê cung:





### 1.1.5. Giao diện khi thua:



### 2.2. Các hàm được sử dụng trong game:

#### 1.1.1. Tạo giao diện khi khởi động game:

- Hàm printWelcome():

Sử dụng công cụ chuyển đổi chữ “SNAKE XENZIA” sang mã ASCII, ta dùng hàm rand() để random màu tạo ra chữ nhấp nháy.

```
char a = _getch();  
if (a == 13)  
    break;
```

Câu lệnh ngắt chữ nhấp  
nháy nếu nhấn phím  
enter

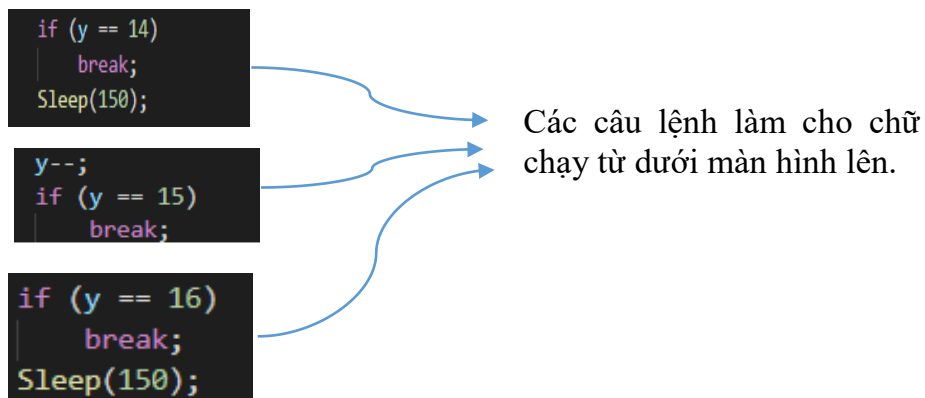
```

int i;
while (true)
{
    i = rand() % 15 + 1 + 1;
    if (i % 16 == 0)
    {
        i = 1;
        SetColor(i);
        gotoXY(25, 1);
        printf("      _      ");
        gotoXY(25, 2);
        printf(" / ____ |      | |      ");
        gotoXY(25, 3);
        printf(" | (___ _ _ _ _ | | ____ ");
        gotoXY(25, 4);
        printf(" \\__ \\| ' _ \\ / _ ' | / / _ \\ ");
        gotoXY(25, 5);
        printf(" ____ ) | | | ( _ | < _ / ");
        gotoXY(25, 6);
        printf(" | ____ / | | | _ \\ \\ _ _ | ");

        gotoXY(35, 8);
        printf(" \\ \\ / /      ( _      ");
        gotoXY(35, 9);
        printf(" \\ \\ v / _ _ _ _ _ _ _ ");
        gotoXY(35, 10);
        printf(" > < / _ \\ ' _ \\ | / _ ' | ");
        gotoXY(35, 11);
        printf(" / . \\ _ / | | / / | ( _ | ");
        gotoXY(35, 12);
        printf(" / / \\ \\ \\ _ | | / _ | _ \\ _ | ");
        gotoXY(35, 13);
        Sleep(500);
        if (_kbhit() == true)
        {
            char a = _getch();
            if (a == 13)
            {
                break;
            }
        }
    }
}

```

- Hàm printInfo(): trong hàm này ta dùng các câu lệnh để in ra tên của từng thành viên trong nhóm.



```

int x = 10, y = 22;
int xcu = -1, ycu = -1;
SetColor(11);
while (true)
{
    // xoa cu
    gotoXY(xcu, ycu);
    printf("                ");
    xcu = x;
    ycu = y;
    gotoXY(x, y);
    printf("Group: 20");
    y--;
    if (y == 14)
        break;
    Sleep(150);
}
x = 10, y = 22;
xcu = -1, ycu = -1;
while (true)
{
    // xoa cu
    gotoXY(xcu, ycu);
    printf("                ");
    xcu = x;
    ycu = y;
    gotoXY(x, y);
    printf("Members: DANG NGOC MINH");
    y--;
    if (y == 15)
        break;
}

```

```

xcu = -1, ycu = -1;
while (true)
{
    // xoa cu
    gotoXY(xcu, ycu);
    printf("                ");
    xcu = x;
    ycu = y;
    gotoXY(x, y);
    printf("NGUYEN VAN DIA LOI");
    y--;
    if (y == 16)
        break;
    Sleep(150);
}
x = 10, y = 22;
xcu = -1, ycu = -1;
while (true)
{
    // xoa cu
    gotoXY(xcu, ycu);
    printf("                ");
    xcu = x;
    ycu = y;
    gotoXY(x, y);
    printf("LE TRUNG KHANG");
    y--;
    if (y == 17)
        break;
    Sleep(150);
}
x = 10, y = 22;
xcu = -1, ycu = -1;
while (true)
{
    // xoa cu
    gotoXY(xcu, ycu);
    printf("                ");
    xcu = x;
    ycu = y;
    gotoXY(x, y);
    printf("NGUYEN NGUYEN HOANG ANH");
    y--;
    if (y == 18)
        break;
}

```

- Hàm printLoadingBar():

Tạo ra thanh loading: “x = 219” trong bảng mã ASCII là thanh chữ nhật.



```

if (i <= 44)
    Sleep(5);
else
    Sleep(2);

```

→ Dùng Sleep tăng dần tốc độ chạy.

```

SetColor(2);
gotoXY(7, 25);
printf("LOADING..");
char x = 219;
int r = 0;
for (int i = 19; i <= 80; i++)
{
    gotoXY(16, 25);
    printf(".");
    Sleep(80);
    gotoXY(16, 25);
    printf(" ");
    gotoXY(i, 25);
    if (i <= 44)
        Sleep(5);
    else
        Sleep(2);
    printf("%c", x);
    gotoXY(82, 25);
    if (i == 80)
    {
        cout << 100 << "%";
        gotoXY(16, 25);
        printf(".");
        break;
    }
    else
    {
        cout << r << "%";
        r++;
    }
}
//_getch();
system("cls");

```

## 2.3. Khởi tạo trò chơi:

### 1.1.1. Vẽ tường:

- Để vẽ tường ta cần có hàm gotoXY

```

7 void gotoXY(int x, int y)
8 {
9     static HANDLE h = NULL;
10    if(!h)
11        h = GetStdHandle(STD_OUTPUT_HANDLE);
12    COORD c = {x,y};
13    SetConsoleCursorPosition(h,c);
14 }
15 void SetTeColor(WORD color)

```

Hàm này dùng để dịch con trỏ hiện tại đến điểm có tọa độ (x,y).

Tiếp theo ta sẽ vẽ tường trên và tường dưới

```

28 void ve_tuong_tren(){
29     int x=10,y=1;
30     while(x<=100){
31         gotoXY(x,y);
32         printf("+");
33         x++;
34     }
35 }
36 void ve_tuong_duoi(){
37     int x=10,y=26;
38     while(x<=100){
39         gotoXY(x,y);
40         printf("+");
41         x++;
42     }
43 }

```

- Vẽ tường trên: Ta sẽ cho chạy từ điểm có tọa độ (10,1) đến (100,1) và dùng vòng lặp, với mỗi điểm sẽ in ra một dấu “+”
- Vẽ tường dưới: Tương tự với tường trên nhưng ta sẽ cho tọa độ y = 26
- Vẽ tường trái và tường phải:

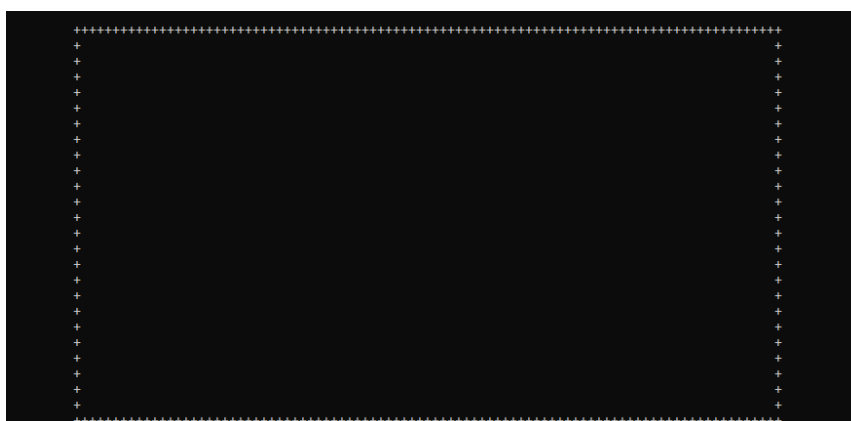
```

44 void ve_tuong_trai(){
45     int x=10,y=1;
46     while(y<=26){
47         gotoXY(x,y);
48         printf("+");
49         y++;
50     }
51 }
52 void ve_tuong_phai(){
53     int x=100,y=1;
54     while(y<=26){
55         gotoXY(x,y);
56         printf("+");
57         y++;
58     }
59 }

```

Cũng giống với vẽ tường trên và dưới nhưng ta sẽ giữ nguyên x và cho y chạy từ 1 đến 26, với mỗi điểm sẽ in ra một dấu “+”

Và ta sẽ có được tường như hình



### 1.1.2. Vẽ rắn:

- Đầu tiên ta sẽ đưa con trỏ tới vị trí con rắn sẽ xuất hiện.

```
void khoitaoran(int toadox[], int toadoy[]){  
    int x=40,y=15;  
    for(int i = 0;i < sl;i++){  
        toadox[i] = x;  
        toadoy[i] = y;  
        x--;  
    }  
}
```

- Sau đó sẽ vẽ rắn ở vị trí con trỏ đã được đưa tới.

```
void ve_ran(int toadox[], int toadoy[]){  
    SetColor(2);  
    for(int i = 0; i < sl; i++){  
        gotoXY(toadox[i],toadoy[i]);  
        if(i == 0){  
            printf("%c",3);  
        }  
        else printf("%c",4);  
    }  
}
```

### 1.1.3. Xử lí rắn

- Đầu tiên ta sẽ tăng độ dài của rắn.

```
void them(int a[], int x){  
    for(int i = sl; i > 0; i--){  
        a[i] = a[i - 1];  
    }  
    a[0] = x;  
    sl++;  
}
```

- Đồng thời ta sẽ xóa đi những vị trí cuối cùng để trả lại độ dài ban đầu của rắn.

```

void xoa(int a[], int vt){
    for(int i = vt; i < sl; i++){
        a[i] = a[i + 1];
    }
    sl--;
}

```

1.1.4. Điều kiện thua:

- a. Rắn chạm tường
  - Ta sẽ so sánh vị trí của đầu rắn với vị trí của bức tường

```

bool kt_ran_cham_tuong(int x0, int y0){
    // ran cham tuong tren
    if(y0 == 1 && (x0 >= 10 && x0 <= 100)){
        return true;
    }
    // ran cham tuong duoi
    else if(y0 == 26 && (x0 >= 10 && x0 <= 100))
        return true;
    // ran cham tuong phai
    else if(x0 == 100 && (y0 >= 1 && y0 <= 26))
        return true;
    // ran cham tuong trai
    else if(x0 == 10 && (y0 >= 1 && y0 <= 26))
        return true;
    return false;
}

```

- Khi vị trí đầu rắn bằng vị trí bức tường, hàm sẽ trả về giá trị true. Khi đó trò chơi sẽ kết thúc.

**bool gameover = false;**

- b. Rắn chạm đuôi
  - Khi vị trí đầu rắn bằng vị trí của thân rắn, hàm cũng sẽ trả về giá trị true, kết thúc trò chơi.

```

bool kt_ran_cham_duoi(int toadox[], int toadoy[]){
    for(int i = 1; i < sl; i++){
        if((toadox[0] == toadox[i]) && (toadoy[0] == toadoy[i]))
            return true;
    }
    return false;
}

```

#### 1.1.5. Hướng di chuyển của rắn

```

int check = 0;
while (true)
{ //kiemtrahuong

    if (check == 0)
    {
        y++; //di xuong
    }
    else if (check == 1)
    {
        y--; // di len
    }
    else if (check == 2)
    {
        x--; // qua trai
    }
    else if (check == 3)
    {
        x++; // qua phai
    }
    xu_ly_ran(toadox, toadoy, x, y);
    sleep(300);
}
_getch();

```

#### 1.1.6. Điều khiển rắn

- Ở đây cho rắn di chuyển bằng hướng mũi tên mà phím mũi tên là phím kép vì vậy cần phải có “kitu” lưu lại `char kitu = _getch();` rồi sau đó bắt đầu kiểm tra kí tự.
- Điều hướng con rắn đi trong trục tọa độ (x,y).



```

if (_kbhit())
{
    char kitu = _getch();
    if (kitu == -32)
    {
        kitu = _getch();
        if (kitu == 72) // di len
        {
            check = 1;
        }
        else if (kitu == 80) // di xuống
        {
            check = 0;
        }
        else if (kitu == 75) //di trai
        {
            check = 2;
        }
        else if (kitu == 77) // di phai
        {
            check = 3;
        }
    }
}
}

```

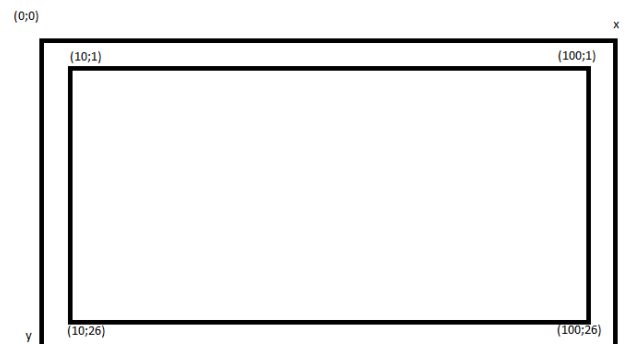
#### 1.1.7. Tạo quả

- Sử dụng công thức random để tạo ra quả: **rand()%(b-a+1)+a**; random 1 số từ (a;b) như hình vẽ thì ta thấy x đi từ (11 < x < 99) y đi từ (2 < y < 25).

```

void tao_qua(int &xqua,int &yqua, int toadox[], int toadoy[]) //random
{
    do
    {
        //11<x<99
        xqua = rand() % (99 - 11 + 1) + 11;
        //2<y<25
        yqua = rand() % (25 - 2 + 1) + 2;
    } while (kt_ran_de_qua(xqua, yqua, toadox, toadoy) == true);
    gotoXY(xqua, yqua);
    cout << "o"
}

```



#### 1.1.8. Rắn đề lên quả

- Tiếp theo, ta sẽ tạo hàm để trái cây xuất hiện một cách ngẫu nhiên trong trục tọa độ xy, với điều kiện trái cây xuất hiện không được trùng với các vị trí xuất hiện con rắn. Khi rắn đề lên quả thì quay lại vòng lặp tạo quả.

```

bool kt_ran_de_qua(int xqua, int yqua, int toadox[],int toadoy[])
{
    for (int i = 0; i < sl; i++)
    {
        if ((xqua == toadox[i])&&(yqua == toadoy[i]))
        {
            return true; // ran de len qua
        }
    }
    return false;
}

```

#### 1.1.9. Rắn ăn quả

```
bool kt_ran_an_qua(int xqua, int yqua, int x0, int y0)
{
    if ((x0 == xqua) && (y0 == yqua))
    {
        return true; // ran an qua
    }
    return false;
}
```

#### 1.1.10. Xử lý khi rắn ăn quả

```
void xu_ly_ran(int toadox[], int toadoy[], int x, int y, int& xqua, int& yqua)
{
    //them toa do moi vao dau mang
    them(toadox, x);
    them(toadoy, y);

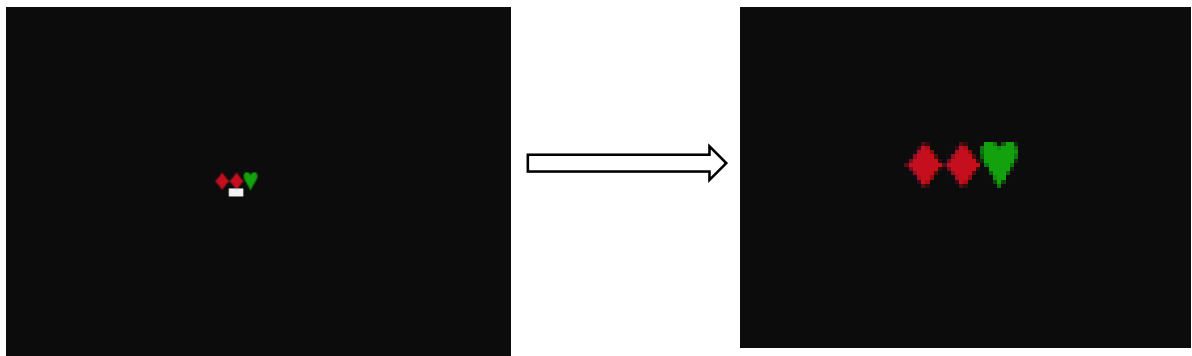
    if (kt_ran_an_qua(xqua, yqua, toadox[0], toadoy[0]) == false)
    {
        //b2 xoa toa do cuoi mang
        xoa(toadox, sl - 1);
        xoa(toadoy, sl - 1);
    }
    else
    {
        tao_qua(xqua, yqua, toadox, toadoy);
    }

    ve_ran(toadox, toadoy);
}
```

Khi rắn ăn quả thì sẽ không xóa tọa độ cuối của mảng và ngược lại sau mỗi lần ăn quả thì rắn sẽ dài ra một đơn vị.

#### 1.1.11. Hàm xóa con trỏ: Xóa vị trí con trỏ theo sau rắn

```
void xoacontro()
{
    CONSOLE_CURSOR_INFO Info;
    Info.bVisible = FALSE;
    Info.dwSize = 20;
    SetConsoleCursorInfo(GetStdHandle(STD_OUTPUT_HANDLE), &Info);
}
```



#### 1.1.12. Câu lệnh khác

- Sleep giúp căn chỉnh tốc độ phụ hợp để rắn di chuyển và là lệnh để dừng màn hình.
- Câu lệnh srand giúp random quả ngẫu nhiên

```
sleep(300);
```

```
srand(time(NULL));
int xqua, yqua;
//taoqua
tao_qua(xqua, yqua, toadox, toadoy);
```

#### 1.1.13. Tạo file lưu điểm cao nhất của người chơi:

```
void xuatVaoFile(int score){
    FILE *f;
    f = fopen("E:/C-CPP/snake/Diem.txt", "w");
    fprintf(f, "Diem cao nhat: %d", score);
    fclose(f);
}
```

```
if(score>maxscore){
    maxscore = score;
    xuatVaoFile(maxscore);
}
```

#### 1.1.14. Chia tách file để dễ dàng quản lý:

Gồm 4 file:

- “mylib.h”: chứa thư viện đồ họa
- “function.h”: chứa các hàm khởi tạo trò chơi con rắn
- “function2.h”: chứa các hàm khởi tạo chế độ trò chơi, giao diện khi game khởi động.
- “main.cpp”: gồm các hàm để chạy được game.

## CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 1. Kết luận

#### 1.1. Kết quả đạt được

Sau quá trình tìm hiểu và thực hiện, đề tài đã đáp ứng được những yêu cầu ban đầu được đưa ra. Xây dựng game rắn săn mồi với đầy đủ các chức năng cơ bản:

- Hiện thị giao diện game: Giao diện ban đầu, giao diện khi chơi, giao diện khi thua.

- Các hàm được sử dụng trong game: Khởi tạo khung giao diện của trò chơi, khởi tạo hàm oxy ban đầu in vị trí rắn và trái cây không trùng nhau. Khởi tạo hàm tọa độ x,y để điều khiển rắn lên xuống; khởi tạo hàm tính điểm khi tọa độ đầu rắn trùng tọa độ trái cây; khởi tạo hàm để các đốm rắn di chuyển; tạo file để lưu điểm cao nhất.

#### 1.2. Kiến nghị

Do thời gian tìm hiểu, phân tích, thiết kế và hiện thực game “Rắn săn mồi” còn hạn chế nên game vẫn còn một số thiếu sót như:

- Giao diện chưa đáp ứng về mặt thẩm mỹ
- Chưa có nhiều tính năng mới, còn sót một vài lỗi.

#### 1.3. Hướng phát triển

Tìm hiểu thêm về một số thư viện, nền tảng hỗ trợ giao diện game trên ngôn ngữ C như unity, unreal, cocos.

Xây dựng thêm chức năng mới như chơi cùng bạn bè bằng mạng Lan qua Wifi.

## PHỤ LỤC

### 1. Phụ lục 1: Hướng dẫn cài đặt

- Cài đặt thư viện “mylib.h”, “functon.h”, “function2.h” và file “main.cpp” vào trong project.

### 2. Phụ lục 2: Hướng dẫn sử dụng

- Người dùng truy cập vào đường link sau để tải các file và trải nghiệm game.

[https://github.com/DanMinnn/big\\_exercise/tree/main](https://github.com/DanMinnn/big_exercise/tree/main)

## TÀI LIỆU THAM KHẢO

- [1]. [Hướng dẫn làm game rắn săn mồi bằng C/C++ | Thien Tam Nguyen](#)  
[Accessed 01-04-2022].
- [2]. [Lập trình game rắn săn mồi bằng C++](#) [Accessed 01-04-2022].
- [3]. [Tổng quan về Github](#) [Accessed 24-04-2022].
- [4]. [Tổng quan về ngôn ngữ lập trình C](#) [Accessed 24-04-2022].