**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**CÔNG NGHỆ THÔNG TIN**

**

**BÁO CÁO BÀI TẬP LỚN**

LẬP TRÌNH SỬ DỤNG API

*ĐỀ TÀI : LẬP TRÌNH GAME RẮN SĂN MỒI BẰNG NGÔN NGỮ C/C++*

SINH VIÊN: MÃ SINH VIÊN

TRỊNH XUÂN TÙNG 171203068

NGUYỄN TIẾN DŨNG

NGUYỄN NGỌC KHÁNH

LỚP : CNTT1-K58

Giới thiệu về game rắn săn mồi

Trò chơi rắn săn mồi quá quen thuộc với các bạn rồi, mình sẽ không trình bày đó là trò chơi gì. Nhưng mình cần trình bày lại trò chơi này cho các bạn ở khía cạnh kỹ thuật, để từ đó ta tìm ra giải pháp xây dựng trò chơi:

#### **Con rắn thực tế và rắn trong game**

Con rắn săn mồi của chúng ta sẽ là **một chuỗi các hình vuồn nhỏ** (các đốt của con rắn) nối lại với nhau (số hình vuông nhỏ chính là độ dài của con rắn). Khởi tạo trò chơi, chúng ta có thể đặt độ dài ban đầu của nó là 3 chẳng hạn. Trong quá trình trò chơi diễn ra, ta phải lưu vết được tọa độ của từng hình vuông đó.

Tại mỗi bước dịch chuyển của rắn, mỗi đốt thân của rắn sẽ dịch chuyển đi 1 đơn vị độ dài bằng nhau. Trong đó, đốt thân đầu tiên (đầu của rắn) sẽ tiến lên theo hướng dịch chuyển, các  đốt thân phía sau di chuyển đến vị trí cũ của đốt thân phía trước nó. Ví dụ:

Giả sử con rắn có 3 đốt và tọa độ của nó hiện tại là: x1(3,0) – đầu, x2(2,0) và x3(1,0) và đang đi theo hướng trục Ox. Bây giờ rắn đổi hướng di chuyển sang bên trái. Khi đó tọa độ mới của từng đốt là: x1(3,-1), x2 sẽ là tọa độ của x1 cũ (3, 0) và x3 chuyển sang vị trí của x2 cũ là (2, 0).

#### **Đưa các đối tượng của game lên ngôn ngữ lập trình**

Như vậy, ta sẽ xây dựng một đối tượng **tọa độ**. Đối tượng này giúp ta lưu được tọa độ của một điểm trên trục tọa độ 2 chiều Oxy. Ta có cấu trúc **tọa độ**như sau:

struct toaDo {

int x;

int y;

};

Khi đó:

* Một con rắn sẽ là 1 mảng các đối tượng toado tương ứng với các đốt của rắn. Tại mỗi đốt, cặp tọa độ (x,y) sẽ lưu vị trí đốt hiện tại và cặp tọa độ (x0,y0) sẽ lưu ví trí trước đó của đốt hiện tại để các đốt sau đó của con rắn có thể sử dụng.
* Mỗi đối tượng thức ăn sẽ là 1 đối tượng toado. Ta chỉ cần sử dụng cặp biến (x, y) để lưu 1 đối tượng thức ăn. Tại một thời điểm trên màn hình chỉ có 1 thức ăn, và thức ăn đó xuất hiện ở 1 vị trí ngẫu nhiên bất kỳ.
* Để rắn di chuyển được 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). Ví dụ nếu rắn đang di chuyển theo hướng trái sang phải, như vậy đối tượng của ta sẽ là (1, 0). Tức là ở mỗi bước đi, tọa độ x sẽ tăng thêm 1 đơn vị và tọa độ y không đổi. Khi ta thay đổi hướng đi, ta chỉ cần thay đổi giá trị (x, y) của đối tượng

#### **Mô hình hóa cách di chuyển của rắn**

Khai báo mảng chứa chiều dài của rắn và một biến hiện chiều dài ban đầu của con rắn :

toaDo snake[Max];

int soDot = 3;

Đầu tiên khởi tạo 3 đốt của con rắn ở vị trí ban đầu :

snake[0].x = tuongTrai+3;

snake[1].x = tuongTrai+2;

snake[2].x = tuongTrai+1;

snake[0].y = tuongTren+2;

snake[1].y = tuongTren+2;

snake[2].y = tuongTren+2;

Khi rắn di chuyển , tọa độ của đốt sau sẽ bằng tọa độ của đốt trước trước đó , tọa độ của đốt cuối sẽ được xóa đi bằng cách thêm vào vị trí đốt cuối một khoảng trắng

gotoXY(dotCuoicu.x, dotCuoicu.y);

cout << " ";

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

1. Tăng chiều dài của rắn
2. Khởi tạo ngẫu nhiên tọa độ thức ăn mới – Lưu ý không khởi tạo trùng vào các đốt thân của rắn.
3. Tăng điểm số (game phải có điểm chứ)

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)

## Các hàm trong game rắn săn mồi

### **Các biến toàn cục trong trò chơi**

typedef char str[31];

str thaotac[4] = { "NEW GAME","HIGHSCORE","ABOUT","QUIT"};//khai báo mảng thaotac chứa 4 lựa chọn cua menu chinh

str thaotac8[2] = { "MORDEN" ,"CLASSIC" };// 2 lua chon phu

str level[5] = { "LEVEL 1","LEVEL 2","LEVEL 3","LEVEL 4", "LEVEL 5" };

str yesno[2] = { "YES","No" };

struct toaDo {

int x;

int y;

};

toaDo snake[Max]; // tọa độ của các đốt rắn

int soDot = 3;// so lượng đốt rắn khởi tạo

Danh sách các hàm xây dựng cho trò chơi rắn săn mồi này:

void khoitaoRan();

void hienthiRan(toaDo dotCuoicu);

toaDo diChuyen(int huong);

void batsuKien(int& huong);

void veTuong();

bool kiemtraThua();

void xulyThua();

toaDo hienthiMoi();

bool kiemtradaanMoi(toaDo moi);

void themDot();

void diemCao();

void classic();

void morden();

void inmenu();

void menu();

void menu8();

void inmenu8();

void mnu();

void about();

int docdiemCao();

void run();

void menuyesno();

void inmenuyesno(int vt);

void mnuyn();

int clock1();

### **Các hàm và vai trò của nó**

void khoitaoRan();

. Khởi tạo vị trí ban đầu của con rắn

void hienthiRan(toaDo dotCuoicu);

.Hiển thị hình vuông của mỗi đốt rắn lên các vị trí đã khởi tạo

.Ghi nhớ đốt đuôi của con rắn

toaDo diChuyen(int huong);

.In đốt mới theo hướng mỗi lần rắn di chuyển

void batsuKien(int& huong);

. Bắt sự kiện di chuyển lên xuống trái phải cho nút

void veTuong();

.Vẽ tường

bool kiemtraThua();

.kiểm tra thua khi rắn chạm vào tường hoặc cắn vào thân

. Dừng game khi giá trị trả về là true

void xulyThua();

. in ra game over

toaDo hienthiMoi();

. Hiển thị tọa độ ngẫu nhiên của mồi trong tọa độ 4 bức tường

bool kiemtradaanMoi(toaDo moi);

. Kiểm tra xem rắn đã ăn mồi chưa

void themDot();

.Thêm đốt cho rắn mỗi khi rắn ăn mồi

void diemCao();

. Ghi lại điểm cao

void classic();

.Chế độ chơi cổ điển cho phép rắn đi xuyên tường

void morden();

.Chế độ chơi hiện đại không cho phép rắn đi xuyên tường

### **Game rắn của bọn mình có tính năng gì?**

 Play/Pause trong khi chơi game

 Lưu điểm cao của người chơi ra tệp

 Điều khiển bàn phím

 Âm thanh sống động