

Nhóm 11

Đồ án Kỹ thuật lập trình

Game rắn săn mồi



Nhóm 11

Đồ án Kỹ thuật lập trình



Game rắn săn mồi

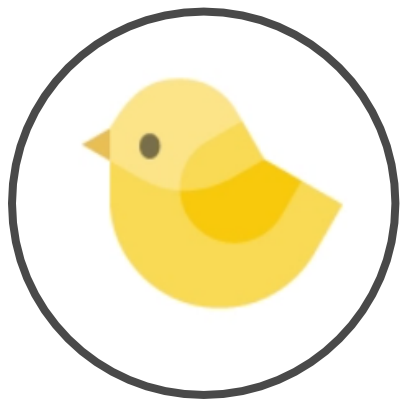


THÀNH VIÊN NHÓM

OUR TEAM



Trịnh Hạnh
21127040



Trần Đỗ Anh Khoa
21127321



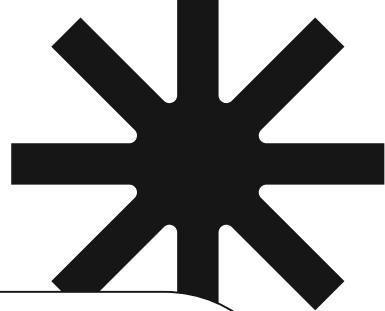
Nguyễn Thiện Nhân
21127380



Nguyễn Khánh Hoàng
21127612

OUR TEAM OUR

Mục lục



01

Giới thiệu

Giới thiệu tổng quát về game rắn sẵn môi

02

Xây dựng game

Xây dựng game ở mức cơ bản nhất

03

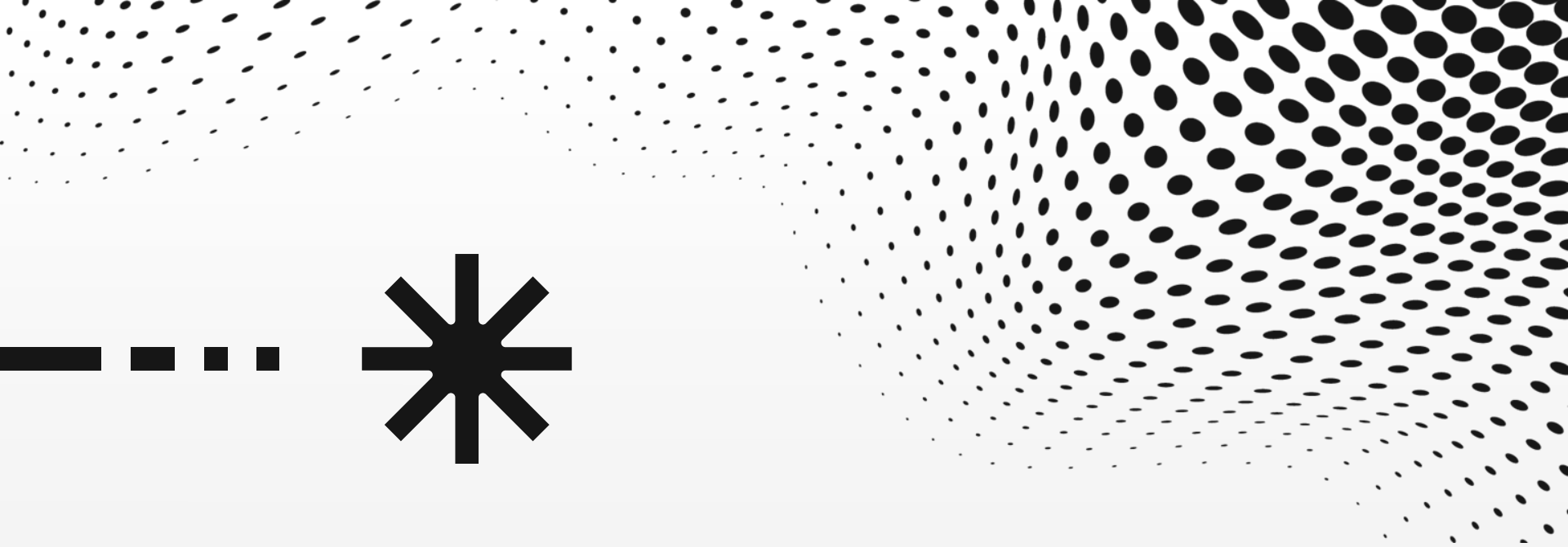
Các tính năng mở rộng

Xây dựng các cơ chế hỗ trợ cho game

04

Demo

Demo sản phẩm

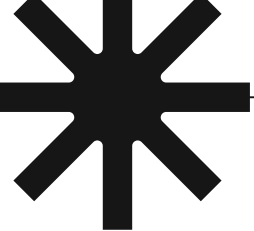


01

Giới thiệu tổng quát

Rắn sẵn mỗi

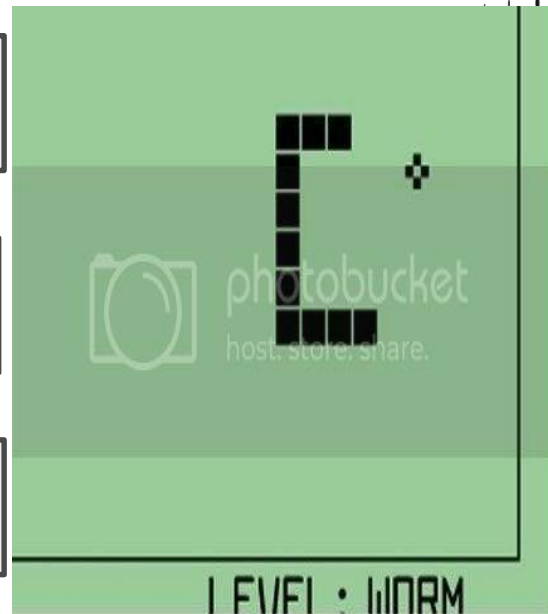




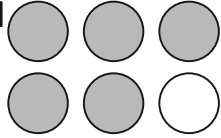
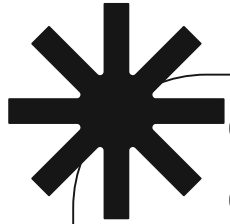
Xuất hiện vào cuối thế kỉ 20 trên các điện thoại Nokia

Được xem là thủy tổ trong ngành game trên di động

Lối chơi đơn giản, có thể chơi trong các khoảng thời gian khá ngắn

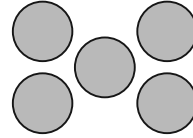
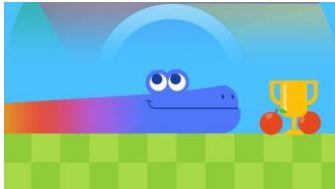


Cơ chế chơi



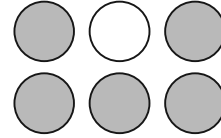
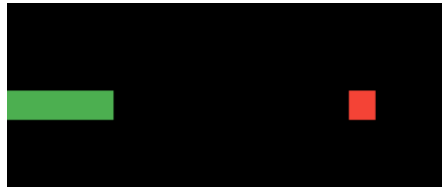
Điều khiển rắn

Người chơi sẽ điều khiển rắn ăn mồi, cố gắng đạt được điểm cao nhất



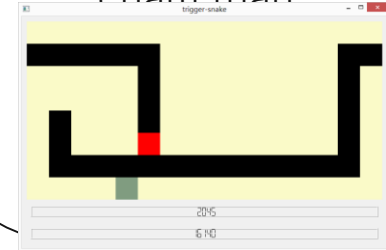
Rắn ăn mồi

Mỗi khi ăn mồi, rắn sẽ dài ra



Rắn thua

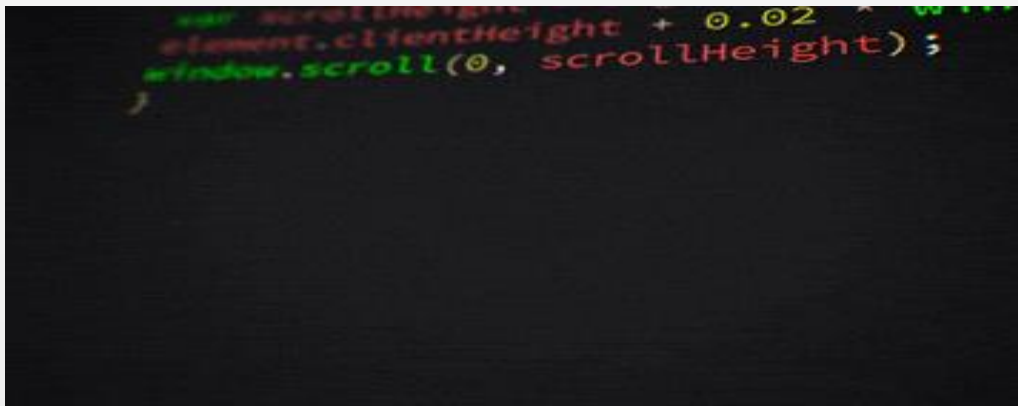
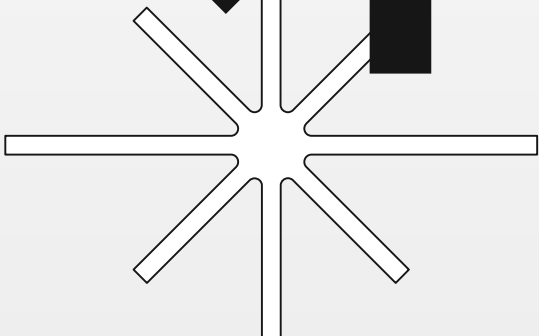
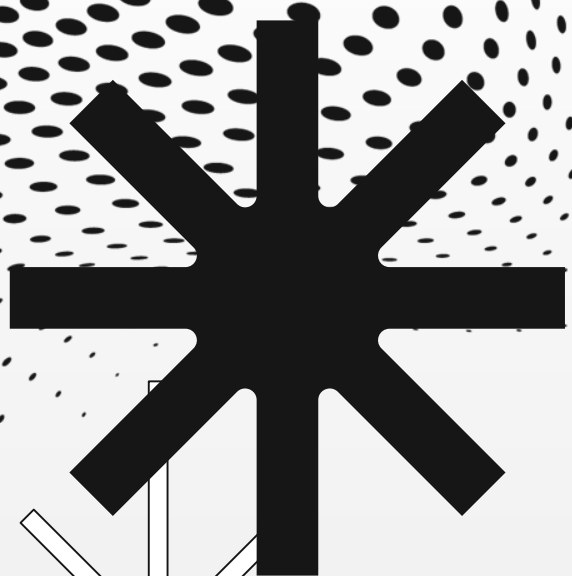
Trò chơi sẽ kết thúc khi rắn chạm tường/ chạm thân

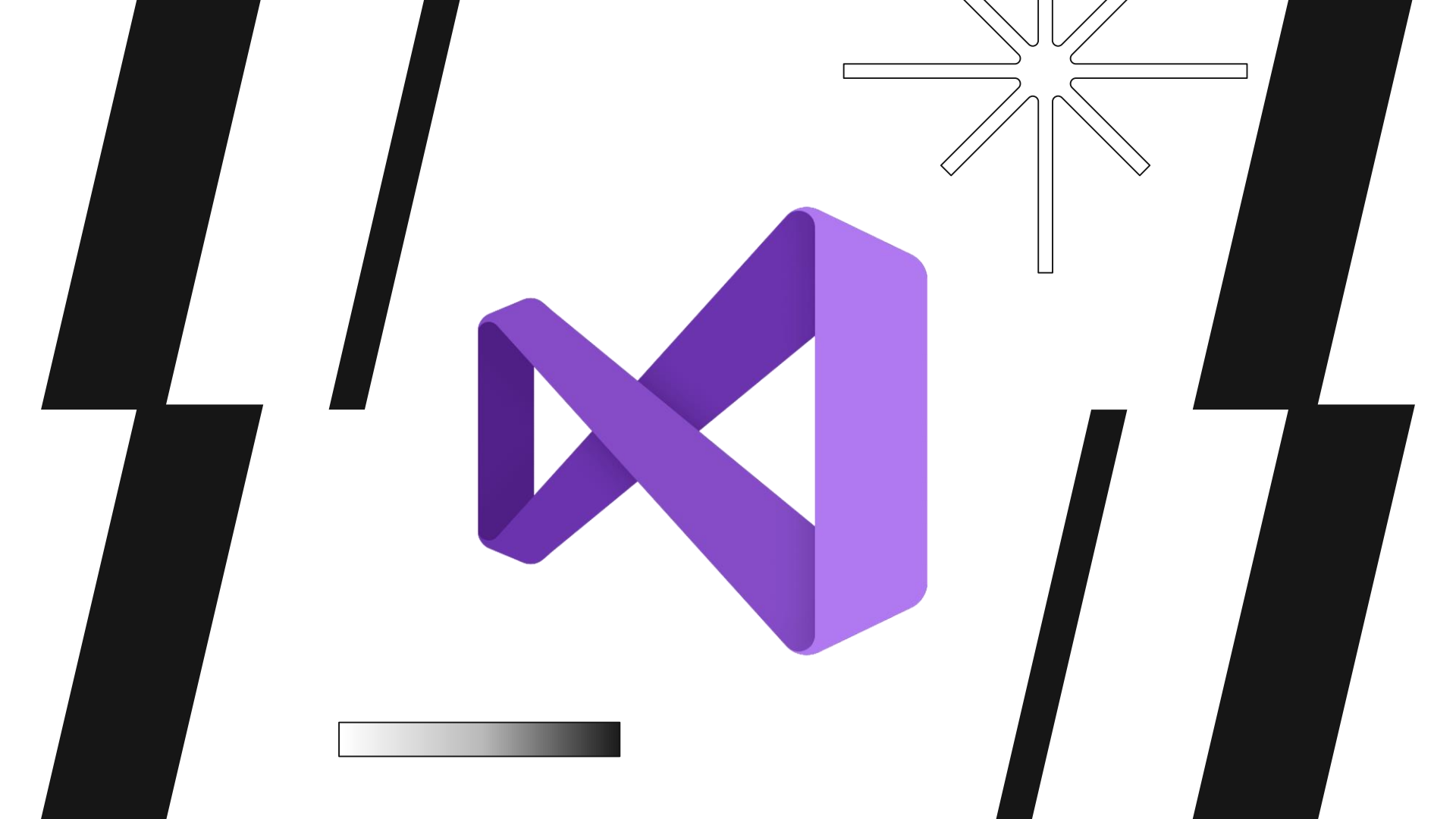


02



Xây dựng game cơ bản





Xây dựng game rắn săn mồi



01

Tạo rắn

Thiết lập tọa độ và
vẽ rắn

02

Di chuyển rắn

Di chuyển rắn trên
hệ trục tọa độ

03

Mồi và sự kiện mồi

Tạo mồi, xử lý sự
kiện khi rắn ăn mồi

04

Xử lý thua game

Kết thúc hành trình
bằng cái chết



Xử lý trên hệ tọa độ XoY

```
void gotoXY(int x, int y)
{
    COORD coord;
    coord.X = x;
    coord.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
}
```



2.1 Tạo rắn

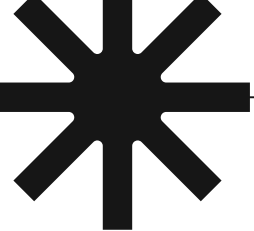
PRESS START

Tạo tọa độ ban đầu cho rắn

```
//Ve ran
void VeRan(ToaDo ran[], int ran_dot)
{
    int a[100] = { 2,1,1,2,7,0,4,0,2,1,1,2,7,3,2,1
, 2, 1, 1, 2, 7, 3, 8, 0, 2, 1, 1, 2, 7};
    for (int i = 0; i < ran_dot; i++)
    {
        gotoXY(ran[i].x, ran[i].y);
        cout << a[i];
    }
}
```

```
//Khoi tao ran
void KhoiTaoRan(ToaDo ran[])
{
    ran[0].x = 7;
    ran[0].y = 15;
}
```

Hiển thị rắn ra console



2.2 Di chuyển rắn

1

Cập nhật hướng

- Trong mỗi vòng game loop, nếu người dùng nhập hướng khác, chương trình sẽ cập nhật hướng mới cho rắn

2

Khóa hướng ngược

- Khóa hướng ngược lại so với hướng trước đó

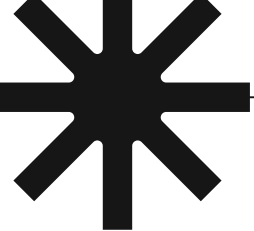
3

Cập nhật tọa độ mới toàn bộ các đốt

- Chương trình sẽ cập nhật tọa độ của từng đốt mỗi lần chạy. Sau đó, chương trình sẽ hiển thị rắn với tọa độ mới cập nhật

```
void BatSukien(int& huong)
{
    int key = inputKey();
    if (key == 'z')
    {
        huong = -1;
        return;
    }
    if (huong == 3)
    {
        if (key == 'w' || key == 'W')
        {
            huong = LEN;
        }
        else if (key == 's' || key == 'S')
        {
            huong = XUONG;
        }
    }
    else if (huong == 1)
    {
        if (key == 'a' || key == 'A')
        {
            huong = TRAI;
        }
        else if (key == 'd' || key == 'D')
        {
            huong = PHAI;
        }
    }
    else if (huong == 4)
    {
        if (key == 'w' || key == 'W')
        {
            huong = LEN;
        }
        else if (key == 's' || key == 'S')
        {
            huong = XUONG;
        }
    }
    else if (huong == 2)
    {
        if (key == 'a' || key == 'A')
        {
            huong = TRAI;
        }
        else if (key == 'd' || key == 'D')
        {
            huong = PHAI;
        }
    }
}
```





2.2 Di chuyển rắn

1

Cập nhật hướng

- Trong mỗi vòng game loop, nếu người dùng nhập hướng khác, chương trình sẽ cập nhật hướng mới cho rắn

2

Khóa hướng ngược

- Khóa hướng ngược lại so với hướng trước đó

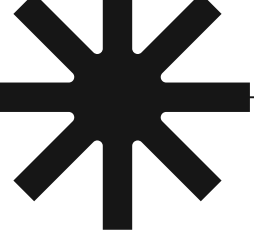
3

Cập nhật tọa độ mới toàn bộ các đốt

- Chương trình sẽ cập nhật tọa độ của từng đốt mỗi lần chạy. Sau đó, chương trình sẽ hiển thị rắn với tọa độ mới cập nhật

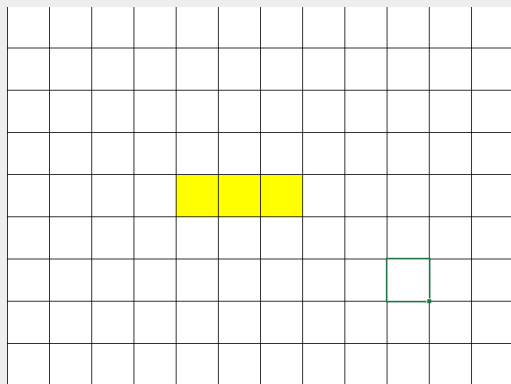
```
ToDo DiChuyen(ToDo ran[], int huong, int ran_dot)
{
    ToDo dot_cuoi_cu = ran[ran_dot - 1];
    for (int i = ran_dot - 1; i >= 1; i--)
    {
        ran[i] = ran[i - 1];
    }
    switch (huong)
    {
        case 1:
        {
            ran[0].y--;
            break;
        }
        case 2:
        {
            ran[0].y++;
            break;
        }
        case 3:
        {
            ran[0].x--;
            break;
        }
        case 4:
        {
            ran[0].x++;
            break;
        }
    }
    return dot_cuoi_cu;
}
```

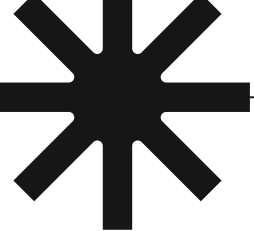




2.2 Di chuyển rắn

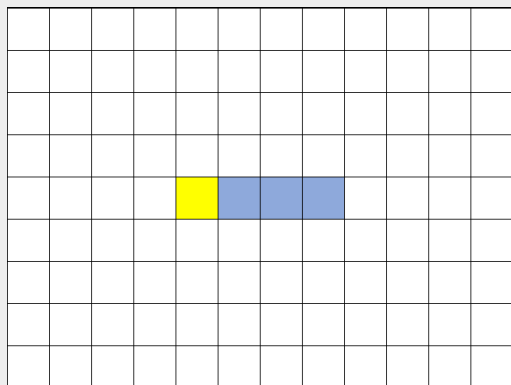
Ý tưởng





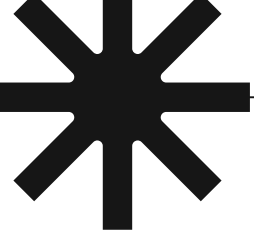
2.2 Di chuyển rắn

Ý tưởng



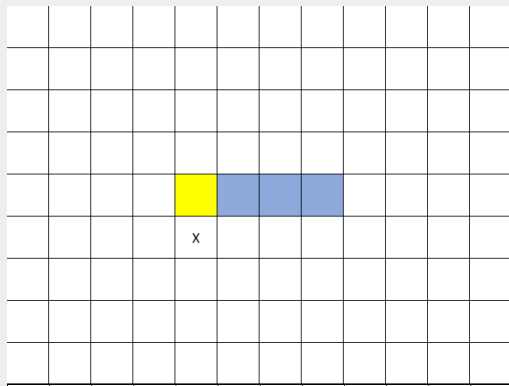
+Vẽ rắn ở tọa độ mới





2.2 Di chuyển rắn

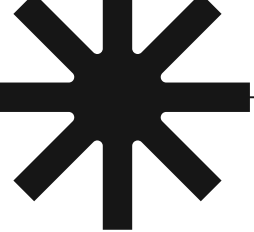
Ý tưởng



+Vẽ rắn ở tọa độ mới

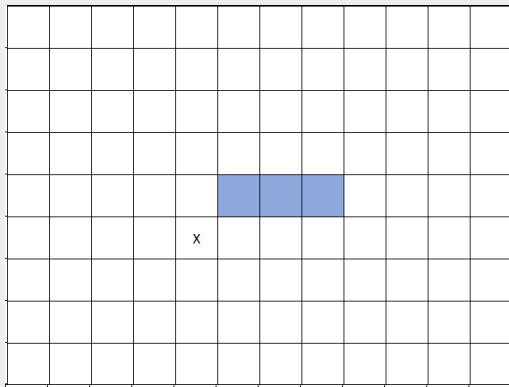
+Đưa con trỏ về đầu cuối cũ





2.2 Di chuyển rắn

Ý tưởng

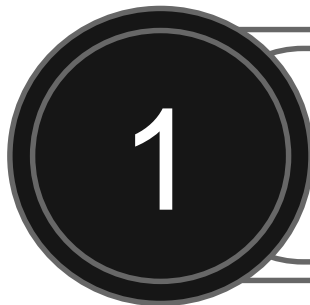
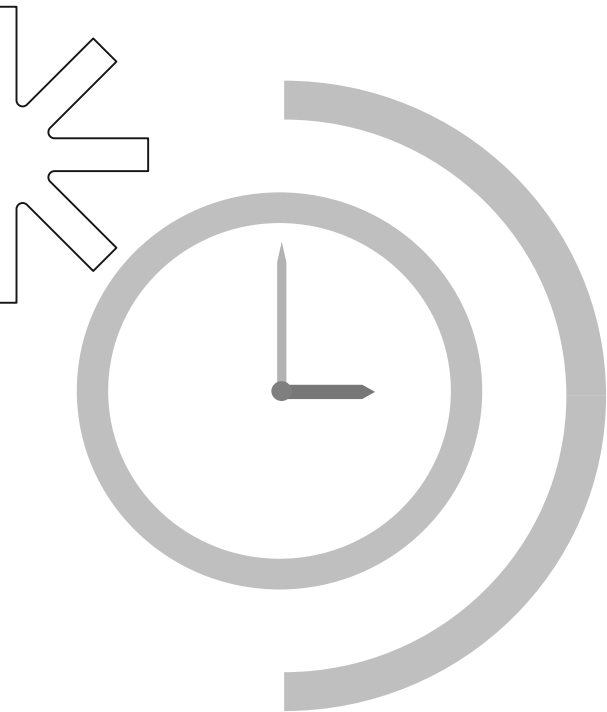


```
void HienThiRan(ToaDo ran[], ToaDo dot_cuoi_cu, int ran_dot)
{
    VeRan(ran, ran_dot);
    gotoXY(dot_cuoi_cu.x, dot_cuoi_cu.y);
    cout << " ";
}
```

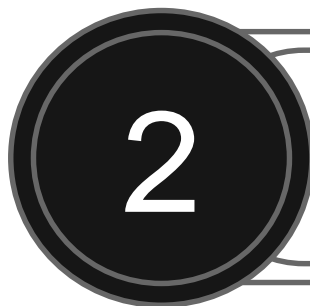
- +Vẽ rắn ở tọa độ mới
- +Đưa con trỏ về đốt cuối cũ
- +Xuất kí tự "space"(xóa đốt đó)



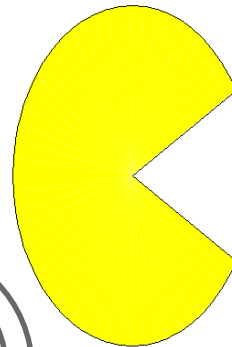
2.3 Tạo môi và xử lý môi



Môi phải nằm trong vùng
chơi, phải tạo trên tọa độ
trống



Duy nhất một môi tồn tại
trên màn hình



...

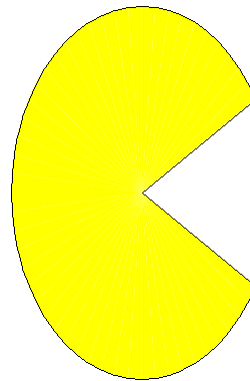


3. Tạo môi và xử lý môi

```
bool KiemTraToaDoQuaKhaThi(int x, int y, ToDo vat_can[], int so_o_vat_can,ToDo ran[],int ran_dot)
{
    for (int i = 0; i < so_o_vat_can; i++)
    {
        if (x == vat_can[i].x && y == vat_can[i].y)
        {
            return false;
        }
    }
    for (int i = 0; i < ran_dot; i++)
    {
        if (x == ran[i].x && y == ran[i].y)
        {
            return false;
        }
    }
    return true;
}
```

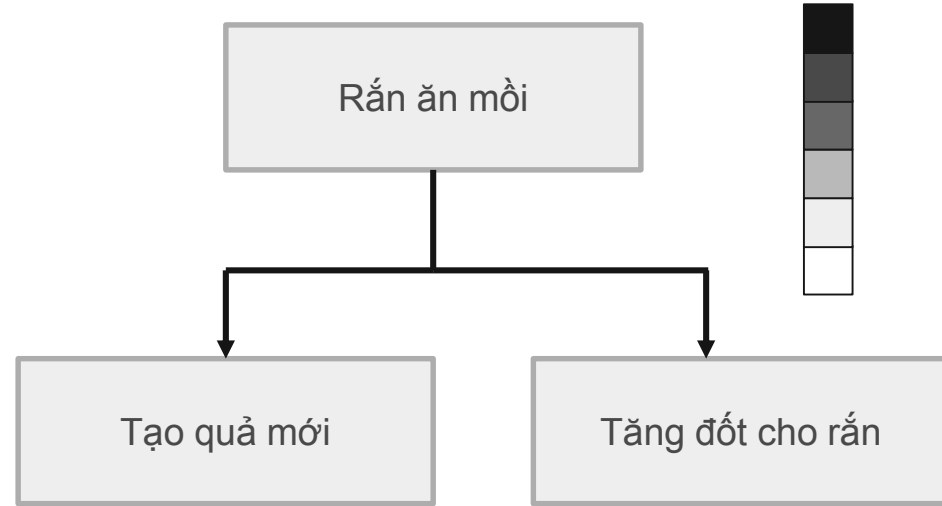
```
//Tao qua 1
void TaoQua1(QUA& food, ToDo vat_can[], int so_o_vat_can) // 4 < xqua < 110; 3 < yqua < 35
{
    do
    {
        food.qua1.x = rand() % (109 - 5 + 1) + 5;
        food.qua1.y = rand() % (34 - 4 + 1) + 4;
    } while (KiemTraToaDoQuaKhaThi(food.qua1.x, food.qua1.y, vat_can, so_o_vat_can) == false);
    gotoXY(food.qua1.x, food.qua1.y);
    cout << "0";
}
```

```
cout << "0";
gotoXY(x_leap.boot, y_leap.boot);
//Tao ran
void TaoRan(RAN& ran, ToDo vat_can[], int so_o_vat_can) // 4 < xran < 110; 3 < yran < 35
{
    do
    {
        ran.ran1.x = rand() % (109 - 5 + 1) + 5;
        ran.ran1.y = rand() % (34 - 4 + 1) + 4;
    } while (KiemTraToaDoQuaKhaThi(ran.ran1.x, ran.ran1.y, vat_can, so_o_vat_can) == false);
    gotoXY(ran.ran1.x, ran.ran1.y);
    cout << "1";
}
```



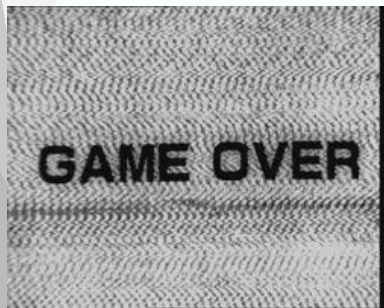
...





```
void AnQua2(ToaDo ran[], QUA& food, int& ran_dot, int& SPEED, int& diem, ToaDo vat_can[], int so_o_vat_can)
{
    if (ran[0].x == food.qua1.x && ran[0].y == food.qua1.y) // +1 point , +1 dot
    {
        ran_dot++;
        diem += 3;
        SPEED -= 2;
        FoodRound2(food, vat_can, so_o_vat_can);
    }
    else if (ran[0].x == food.qua10.x && ran[0].y == food.qua10.y) { ... }
}
```

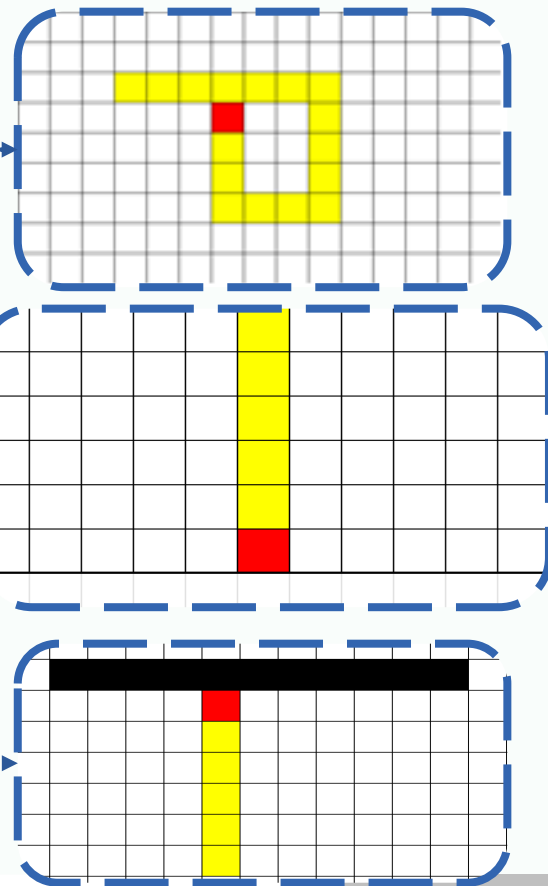
4. Sự kiện thua



Rắn chạm thân

Rắn chạm tường

Rắn chạm vật cản



```

bool RanChamThan(ToaDo ran[], int ran_dot)
{
    for (int i = 1; i < ran_dot; i++)
    {
        if (ran[0].x == ran[i].x && ran[0].y == ran[i].y)
        {
            return true;
        }
    }
    return false;
}

```

//Kiem tra thua khi ran cham vao khung tro choi

```

bool RanChamTuong(ToaDo ran[])

```

```

{
    if (ran[0].y == TUONG_TREN)
    {
        return true;
    }
    if (ran[0].y == TUONG_DUOI)
    {
        return true;
    }
    if (ran[0].x == TUONG_TRAI)
    {
        return true;
    }
    if (ran[0].x == TUONG_PHAI)
    {
        return true;
    }
    return false;
}

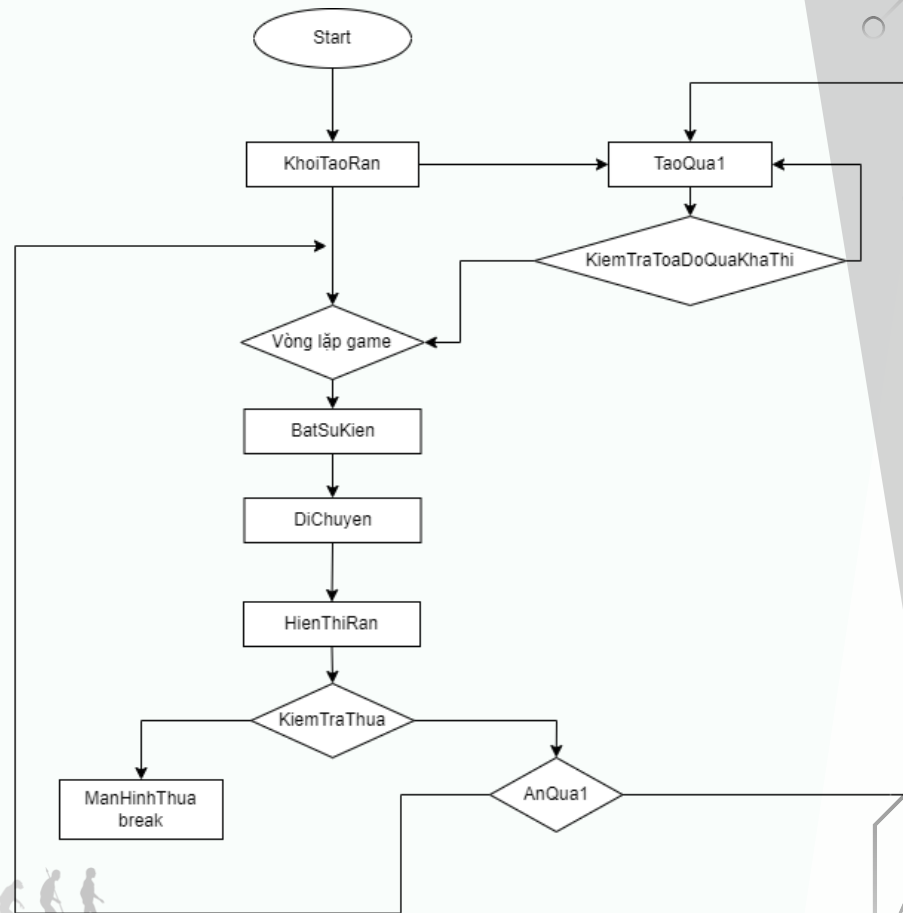
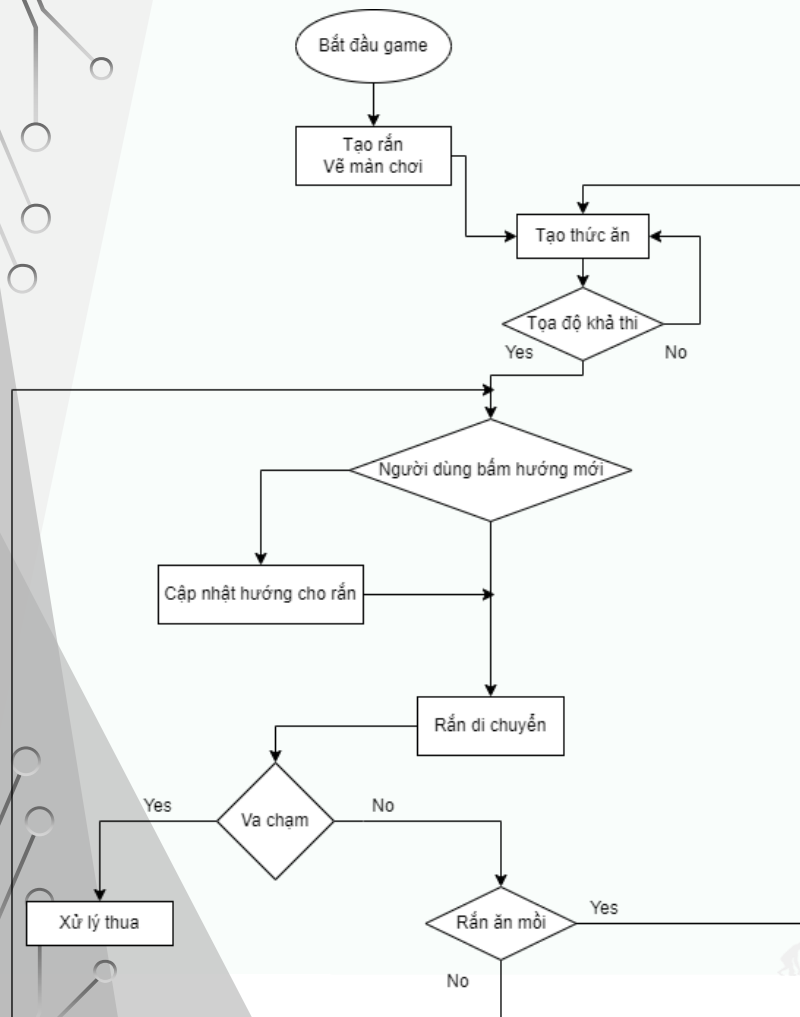
```

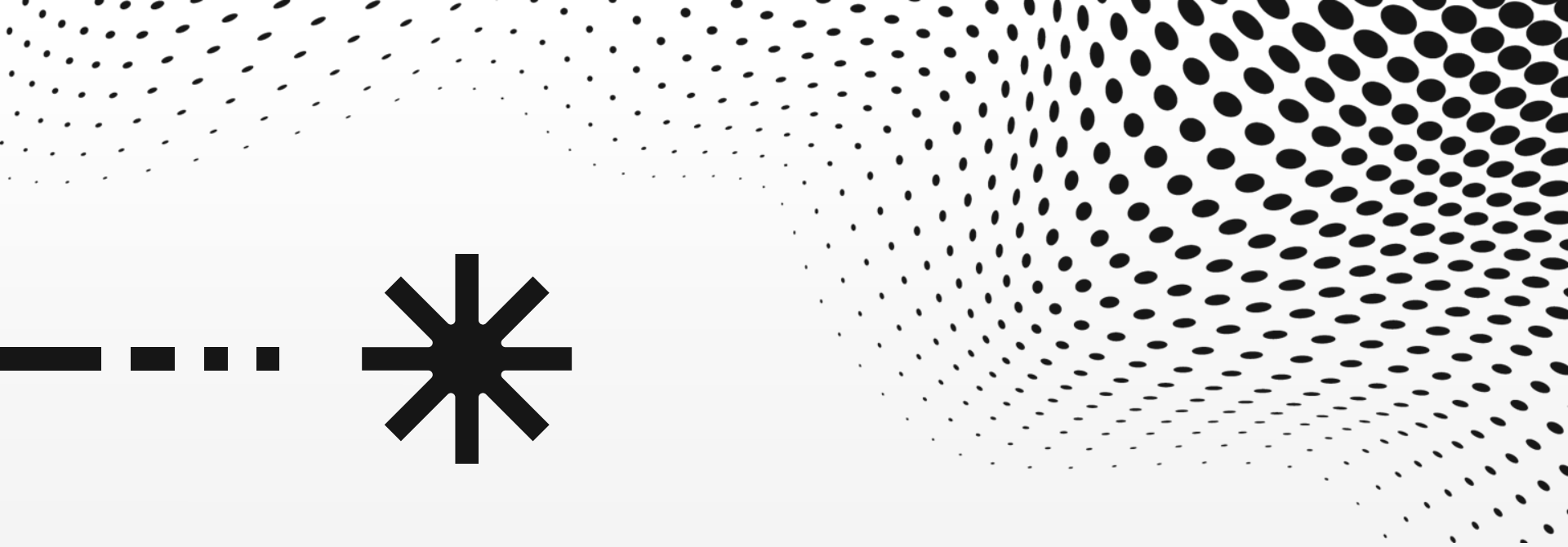
```

bool KiemTraThuaVatCan2(ToaDo ran[], ToaDo vat_can[], int so_o_vat_can)
{
    for (int i = 0; i < so_o_vat_can; i++)
    {
        if (ran[0].x == vat_can[i].x && ran[0].y == vat_can[i].y)
        {
            return true;
        }
    }
    return false;
}

```







03

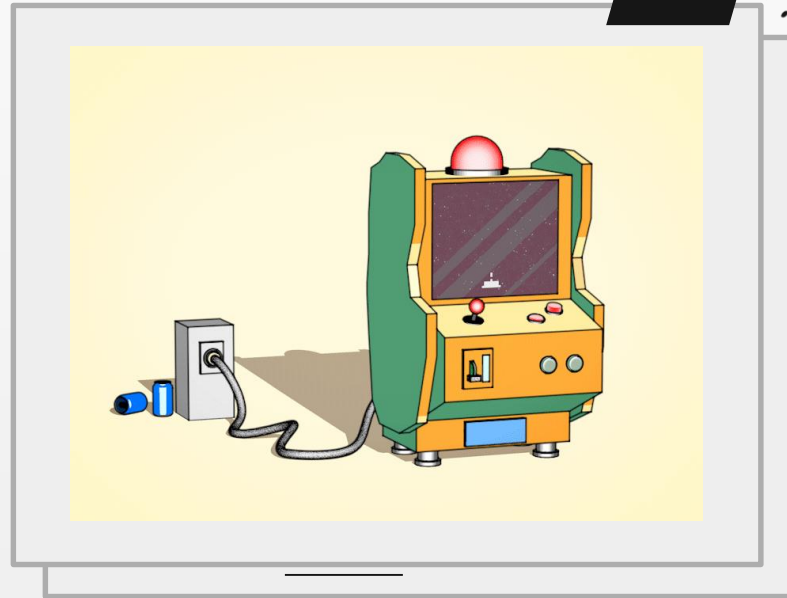
Các tính năng mở rộng

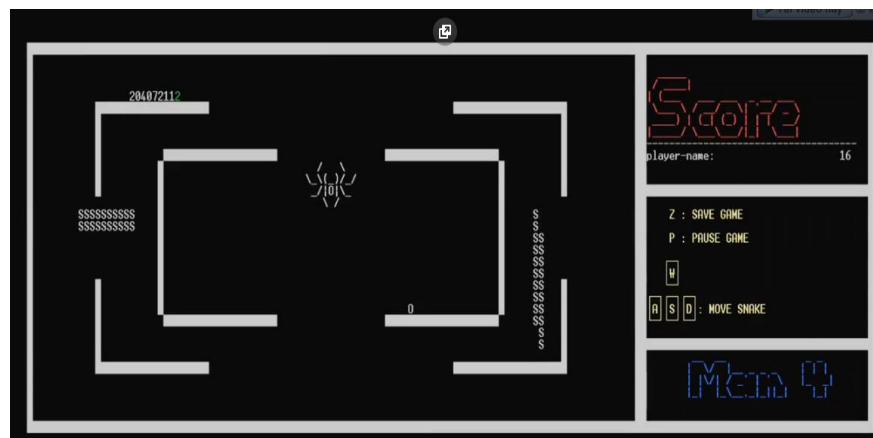
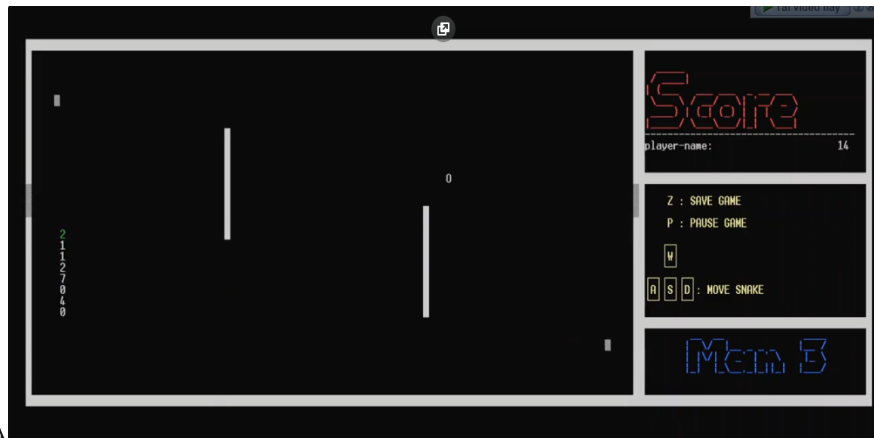




Màn chơi

- Game sẽ có 4 màn chơi và một màn trùm
- Nếu đủ một số điểm nhất định, người chơi sẽ đi qua cổng để qua màn tiếp theo

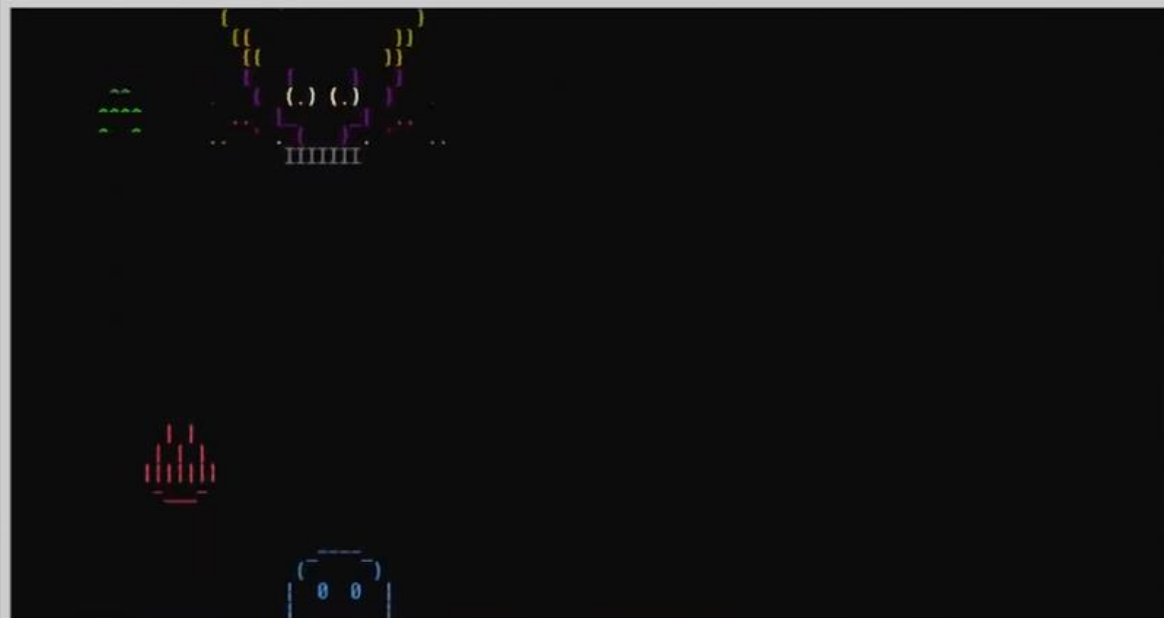




Dragon 's Life:



fat video nay



Score

nguyen van a: 0

Z : SAVE GAME

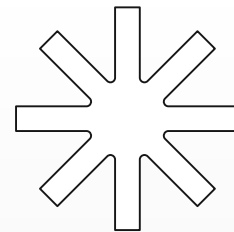
P : PAUSE GAME

W

A S D : MOVE SNAKE

Menu B

Hệ thống quả



Quả 1

Ăn vào giúp tăng 1
đốt và 5 điểm



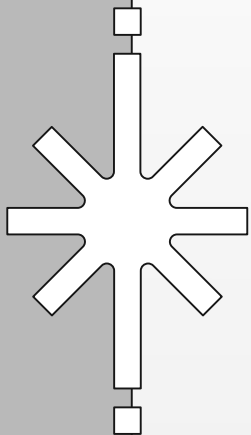
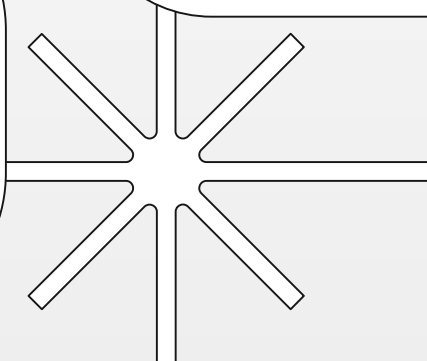
Quả 10

Ăn vào giúp tăng 1
đốt và 10 điểm



Quả độc

Ăn vào sẽ khiến rắn từ
từ mất đốt. Khi mất hết
đốt, game sẽ thua. Ăn
quả khác để thoát
trạng thái



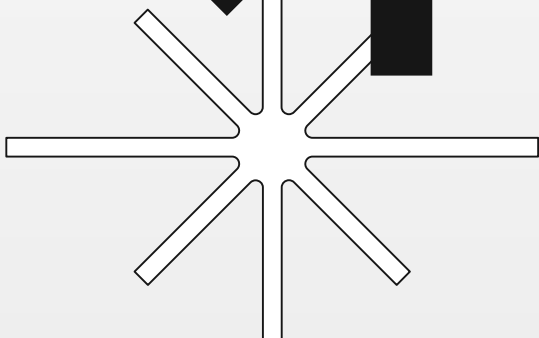
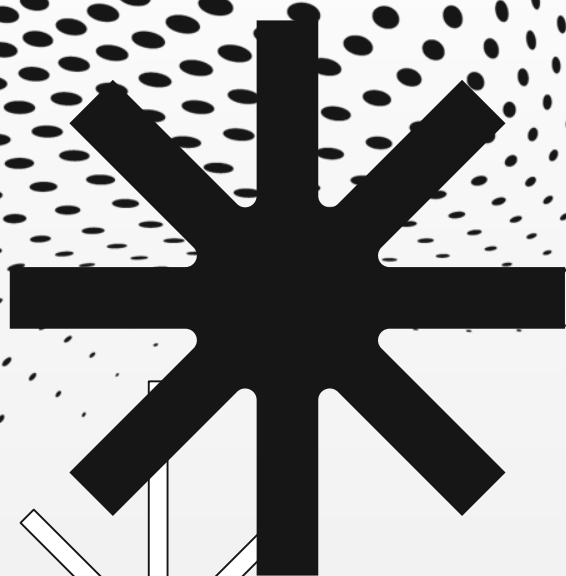
Một vài chức năng khác



04



Demo



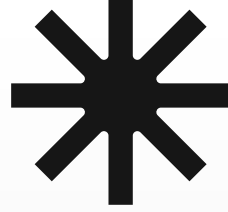


→	BẮT ĐẦU
	Tiếp tục
	Bảng điểm
	Hướng dẫn
	Nhạc: Tắt
	Thoát

Nhấn bắt đầu để bắt đầu chơi từ màn 1



KAPWING



Does anyone have
any questions?

**Thank
you!**

