**BÁO CÁO BÀI TẬP LỚN NHÓM 20**

**Nhóm sinh viên thực hiện:**

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **MSSV** | **Lớp** |
|  |  |  |
| Nguyễn Hưng Thịnh**(nhóm trưởng)** | 6151071105 | CNTT.K61 |
| Nguyễn Minh Triều | 6151071025 | CNTT.K61 |
| Đặng Đăng Trình | 6151071026 | CNTT.K61 |

**Ý tưởng của nhóm:**

Tạo ra một chương trình game đơn giản dựa theo tựa vượt chướng ngại vật, người chơi sẽ điều khiển xe của mình tránh các vật cản để được cộng điểm. Ngoài ra, xe của họ còn được trang bị súng để bắn các vật cản. Khi đạt được số điểm nhất định thì độ khó của trò chơi sẽ tăng lên.

**Phương án thực hiện:**

Sử dụng ngôn ngữ C, tạo ra các đối tượng **xe, vatcan, dan** và thao tác đối với nó. Thiết kế hình dạng của **xe, vatcan, dan** và in ra màn hình console. Dùng hàm **buffer()** quét màn hình mỗi vòng lặp để chương trình chạy mượt hơn. Xử lý các sự kiện khi **xe** va chạm với **vatcan → gameover()**, **dan** va chạm với **vatcan**.

**Các kiến thức được sử dụng trong chương trình:**

* Gọi hàm
* If – else
* Vòng lặp ( for, while, do while )
* Struct

**Tiến hành viết chương trình:**

Bài làm của nhóm: [**bewtmf/BAITAPLON (github.com)**](https://github.com/bewtmf/BAITAPLON)

Chương trình gồm 3 đối tượng chính : **xe, vatcan, dan**

Giới hạn console:

#define consoleWidth 120

#define consoleHeight 30

#define Road 50

Tạo struct: **Toado, Hinhdang, Vehicle, Dan**

struct Toado{

    int x, y;

};

struct Hinhdang{

    char a[3][3];

};

typedef struct Vehicle{

    Toado td;

    Hinhdang hd;

    int diem= 0;

    trangthai tt;

}Veh;

struct Dan{

    Toado td;

    trangthai tt;

    Hinhdang hd;

};

Khởi tạo đối tượng **xe** là một mảng 3x3

    #define HD xe.hd.a

    xe.td.y = consoleHeight -2;

    xe.td.x = Road / 2;

    //--Vẽ xe

    HD[0][0] = HD[0][2] = HD[2][0] = HD[2][2] = 'o';

    HD[0][1] = '^';

    HD[1][0] = HD[1][2] = '|'; HD[1][1] = 'X';

    HD[2][1]='-';

Khởi tạo đối tượng **vatcan** gồm **6 vatcan**

    for(int i=0 ;i < 6; i++)

    {

        vc[i].td.x = 3 + rand() % (Road - 5); // x[2, 27];

        vc[i].td.y = rand() % (-2 + 6) - 2; // y[-2, -6]; rand() % (max - min) + min;

        vc[i].tt = DOWN;

        for(int dong = 0;dong < 3; dong++)

        {

            for(int cot = 0; cot < 3; cot++)

            {

                if(i < 3)

                    vc[i].hd.a[dong][cot] = 19; // tạo hình cho vatcan 1 -> 3

                else if( i >= 3)

                    vc[i].hd.a[dong][cot] = 207; // tạo hình cho vatcan 4, 5

            }

        }

    }

Khởi tạo hàm buffer quét màn hình console mỗi vòng lặp:

**Tài liệu hướng dẫn buffer:** [Phần 4 - Vấn đề mở rộng [Lập trình game cơ bản] - YouTube](https://www.youtube.com/watch?v=hsvzJlxG2LY) **(phút 49:00)**

struct A{

    char kytu;

    int mau;

};

A buffer[consoleHeight][Road];

void Draw(int dong, int cot, int kytu, int mau){

    buffer[dong][cot].kytu = kytu;

    buffer[dong][cot].mau = mau;

}

// Khởi tạo buffer

for(int i=0;i< consoleHeight; i++)

        for(int j=0;j<Road;j++)

        {

            buffer[i][j].kytu = ' ';

            buffer[i][j].mau = 7;

        }

Vẽ các đối tượng ra màn console sau khi khởi tạo:

//------Vẽ đường biên

    for(int i= 0;i< consoleHeight; i++)

    {

        Draw(i, 1, 219, 1);

        Draw(i, Road, 219, 1);

    }

    //-----Vẽ xe

    for(int dong = -1; dong <= 1; dong++)

    {

        for(int cot = -1; cot <= 1; cot++)

            {

                int x = cot + xe.td.x;

                int y = dong + xe.td.y;

                Draw(y, x, xe.hd.a[dong+1][cot+1], 11);

            }

    }

    //---Vẽ vật cản

    for(int i =0 ;i < 3;i ++)

    {

        for(int dong = -1; dong <= 1; dong++)

        {

            for(int cot = -1; cot <= 1; cot ++)

            {

                int x = cot + vc[i].td.x;

                int y = dong + vc[i].td.y;

                    if(y >= 0 && y < consoleHeight)

                        Draw(y, x, vc[i].hd.a[dong+1][cot+1], 4);

            }

        }

    }

    if(xe.diem >= 50)

    {

        for(int i = 3 ;i < 6;i ++)

        {

            for(int dong = -1; dong <= 1; dong++)

            {

                for(int cot = -1; cot <= 1; cot ++)

                {

                    int x = cot + vc[i].td.x;

                    int y = dong + vc[i].td.y;

                        if(y >= 0 && y < consoleHeight)

                            Draw(y, x, vc[i].hd.a[dong+1][cot+1], 4);

                }

            }

        }

    }

//\*\* Dua buffer ra console \*\*

    gotoxy(0, 0);

    for(int i = 0; i < consoleHeight; i++)

    {

        for(int j = 0;j < Road + 1; j++)

        {

            TextColor(buffer[i][j].mau);

            putchar(buffer[i][j].kytu);

            buffer[i][j].kytu = ' ';

        }

        if(i < consoleHeight - 1)

            putchar('\n');

    }

Điều khiển **xe**:

Hàm **kbhit()**: để kiểm tra có sự kiện nhập phím của người dùng hay không. Nếu có, dùng getch() để tóm phím đó.

if(\_kbhit())

    {

        int key = getch();

        if(key == 75 && xe.td.x > 3)

            xe.td.x -= 2;

        else if(key == 77 && xe.td.x < Road -3)

            xe.td.x += 2;

        if(key == 32)

        {

            dan.td.x = xe.td.x;

            dan.td.y = xe.td.y - 2;

            dan.tt = UP;

        }

    }

Khi có va chạm:

//----Vật cản va chạm xe

    int dX[5], dY[5];

    for(int i = 0; i < 3;i++)

    {

        dX[i] = abs(xe.td.x - vc[i].td.x);

        dY[i] = abs(xe.td.y - vc[i].td.y);

        if(dX[i] < 3 && dY[i] < 3)

                return -1;

    }

    for(int i = 3; i < 6; i++)

        {

            dX[i] = abs (xe.td.x - vc[i].td.x);

            dY[i] = abs (xe.td.y - (vc[i].td.y - 1) );

            if(dX[i] < 3 && dY[i] < 3)

                return -1;

        }