#define \_CRT\_SECURE\_NO\_WARNINGS//防止不必要的警告

#include<iostream>

using namespace std;//std::cout<<

#include<windows.h>

#include<time.h>

#include<stdlib.h>

#include<conio.h>

#include <mmsystem.h>

#pragma comment(lib, "WINMM.LIB")

//int r;// 星星周围装甲层数，一发炮弹击穿一层，fire，star中的参数

void gotoxy(int x, int y)//光标位置函数

{

COORD pos;

pos.X = 2 \* x;

pos.Y = y;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), pos);

}

void color(int a)//颜色函数

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), a);

}

void tank(int tank[2], char c, int n)

{

color(n);

gotoxy(tank[0], tank[1]);

if (n == 4 || n == 6 || n == 1)

printf("★");

else

printf("■");

gotoxy(tank[0] - 1, tank[1]);

printf("■");

gotoxy(tank[0] + 1, tank[1]);

printf("■");

gotoxy(tank[0], tank[1] - 1);

printf("■");

gotoxy(tank[0], tank[1] + 1);

printf("■");

switch (c)

{

case 'w':

gotoxy(tank[0] + 1, tank[1] + 1);

printf("■");

gotoxy(tank[0] - 1, tank[1] + 1);

printf("■");

break;

case 's':

gotoxy(tank[0] - 1, tank[1] - 1);

printf("■");

gotoxy(tank[0] + 1, tank[1] - 1);

printf("■");

break;

case 'a':

gotoxy(tank[0] + 1, tank[1] - 1);

printf("■");

gotoxy(tank[0] + 1, tank[1] + 1);

printf("■");

break;

case 'd':

gotoxy(tank[0] - 1, tank[1] - 1);

printf("■");

gotoxy(tank[0] - 1, tank[1] + 1);

printf("■");

break;

}

}

void desk(int N)//初始化函数（初始化围墙、显示信息，）

{

int n = 10, m = 0;

int i, j;//初始化围墙

for (i = 0; i <= N + 1; i++)

{

gotoxy(i, 1);

color(n);

printf("□");

}

putchar('\n');

for (j = 2; j <= N + 1; j++)

{

color(n);

printf("□");

for (i = 1; i <= N; i++)

{

color(m);

printf("■");

}

color(n);

printf("□");

putchar('\n');

}

for (i = 0; i <= N + 1; i++)

{

gotoxy(i, N + 2);

color(n);

printf("□");

}

color(13);

gotoxy(N + 3, 1);//显示信息

printf("W S A D 移动");

gotoxy(N + 3, 2);//显示信息

printf("空格开炮");

gotoxy(N + 3, 3);

printf("生命:");

gotoxy(N + 3, 4);

printf("得分：");

}

void star(int s, int N)//星星函数

{

int q = 12;

int i;

int k;

for (i = 1; i <= 3; i++)

{

color(q);

for (k = 1; k <= s + 1; k++)

{

gotoxy(N / 2 - 1 + i, N + 2 - k);

printf("■");

}

}

gotoxy(N / 2 + 1, N + 1);

color(q);

printf("★");

}

void wall(int n, int N)//墙

{

int i, c = 20;

color(c);

for (i = 1; i <= 3; i++)

{

gotoxy(3 + i, 5);

printf("■");

gotoxy(N - 6 + i, 5);

printf("■");

gotoxy(3 + i, N - 2);

printf("■");

gotoxy(N - 6 + i, N - 2);

printf("■");

}

for (i = 1; i <= 2; i++)

{

gotoxy(4, 5 + i);

printf("■");

gotoxy(N - 3, 5 + i);

printf("■");

gotoxy(4, N - 2 - i);

printf("■");

gotoxy(N - 3, N - 2 - i);

printf("■");

}

if (n == 3)

{

for (i = 1; i <= N / 5; i++)

{

gotoxy(2 \* N / 5 + i, 5);

printf("■");

gotoxy(4, 2 \* N / 5 + 1 + i);

printf("■");

gotoxy(2 \* N / 5 + i, N - 2);

printf("■");

gotoxy(N - 3, 2 \* N / 5 + 1 + i);

printf("■");

}

}

}

int fire(int firetank[2], int tank1[2], int tank2[2], char c, char e1, char e2, int z, int N, int r) //敌方坦克开火函数 其中返回值是3表示敌方打中敌方 ,tank[1]是我方坦克

{

int n = 0, m = 9;

int p;

switch (c)

{

case 'w':

p = firetank[1] - 2;

if (p == 1)

;

else

{

for (p; p > 1; p--)

{

gotoxy(firetank[0], p);

color(m);

printf("●");

Sleep(30);

gotoxy(firetank[0], p);

color(n);

printf("■");

if (p == tank1[1] + 1 && (firetank[0] == tank1[0] || firetank[0] == tank1[0] + 1 || firetank[0] == tank1[0] - 1))

{

tank(tank1, e1, n);

Sleep(100);

mciSendString(TEXT("play x.wma wait"), 0, 0, 0);

return 1;

}

if (p == tank2[1] + 1 && (firetank[0] == tank2[0] || firetank[0] == tank2[0] + 1 || firetank[0] == tank2[0] - 1))

{

tank(tank2, e2, n);

Sleep(100);

mciSendString(TEXT("play x.wma wait"), 0, 0, 0);

return 3;

}

}

break;

}

case 's':

p = firetank[1] + 2;

if (p == N + 2)

;

else

{

for (p; p < N + 2; p++)

{

if ((p == N + 1 - z) && (firetank[0] == N / 2 || firetank[0] == N / 2 + 2)) break;// 左右上角装甲打不穿

gotoxy(firetank[0], p);

color(m);

printf("●");

Sleep(50);

gotoxy(firetank[0], p);

color(n);

printf("■");

if (p == tank1[1] - 1 && (firetank[0] == tank1[0] || firetank[0] == tank1[0] + 1 || firetank[0] == tank1[0] - 1))

{

tank(tank1, e1, n);

Sleep(100);

mciSendString(TEXT("play x.wma wait"), 0, 0, 0);

return 1;

}

if (p == tank2[1] - 1 && (firetank[0] == tank2[0] || firetank[0] == tank2[0] + 1 || firetank[0] == tank2[0] - 1))

{

tank(tank2, e2, n);

Sleep(100);

mciSendString(TEXT("play x.wma wait"), 0, 0, 0);

return 3;

}

if (firetank[0] == N / 2 + 1 && p == N + 1 - r)//一次只能击穿一层中间的装甲。

{

return 2;

}

}

}

break;

case 'a':

p = firetank[0] - 2;

if (p == 0)

;

else

{

for (p; p > 0; p--)

{

if ((firetank[1] == N + 1 - z || firetank[1] == N + 2 - z || firetank[1] == N + 3 - z || firetank[1] == N + 4 - z) && (p == N / 2 + 2)) break;//右侧不被击穿

gotoxy(p, firetank[1]);

color(m);

printf("●");

Sleep(50);

gotoxy(p, firetank[1]);

color(n);

printf("■");

if (p == tank1[0] + 1 && (firetank[1] == tank1[1] || firetank[1] == tank1[1] + 1 || firetank[1] == tank1[1] - 1))

{

tank(tank1, e1, n);

Sleep(100);

mciSendString(TEXT("play x.wma wait"), 0, 0, 0);

return 1;

}

if (p == tank2[0] + 1 && (firetank[1] == tank2[1] || firetank[1] == tank2[1] + 1 || firetank[1] == tank2[1] - 1))

{

tank(tank2, e2, n);

Sleep(100);

mciSendString(TEXT("play x.wma wait"), 0, 0, 0);

return 3;

}

}

}

break;

case 'd':

p = firetank[0] + 2;

if (p == N + 1)

;

else

{

for (p; p < N + 1; p++)

{

if ((firetank[1] == N + 1 - z || firetank[1] == N + 2 - z || firetank[1] == N + 3 - z || firetank[1] == N + 4 - z) && (p == N / 2)) break;//左侧不被打穿

gotoxy(p, firetank[1]);

color(m);

printf("●");

Sleep(50);

gotoxy(p, firetank[1]);

color(n);

printf("■");

if (p == tank1[0] - 1 && (firetank[1] == tank1[1] || firetank[1] == tank1[1] + 1 || firetank[1] == tank1[1] - 1))

{

tank(tank1, e1, n);

Sleep(50);

mciSendString(TEXT("play x.wma wait"), 0, 0, 0);

return 1;

}

if (p == tank2[0] - 1 && (firetank[1] == tank2[1] || firetank[1] == tank2[1] + 1 || firetank[1] == tank2[1] - 1))

{

tank(tank2, e2, n);

Sleep(50);

mciSendString(TEXT("play x.wma wait"), 0, 0, 0);

return 3;

}

}

}

}

return 0;

}

int starfire(int y, int tank1[2], char e, int N)//苹果开火

{

int n = 0, m = 9;

int p;

if (y == 0)//当且仅当苹果正面装甲全部被击穿，并且只能开一炮

{

y = y - 1;//防止无限制开炮

p = N;

for (p; p > 1; p--)

{

gotoxy(N / 2 + 1, p);

color(m);

printf("●");

Sleep(30);

gotoxy(N / 2 + 1, p);

color(n);

printf("■");

if (p == tank1[1] + 1)

{

tank(tank1, e, n);

Sleep(100);

mciSendString(TEXT("play x.wma wait"), 0, 0, 0);

return 1;

}

}

}

return 0;

}

int Move(int tank0[2], int tank1[2], int tank2[2], char c, char b, int n, int z, int N)//移动坦克中心位置，与另外两辆坦克不能相撞，并且不能撞star的装甲

{

tank(tank0, b, 0);

if (b == c)//坦克前进一格

{

switch (c)

{

case 'a':

if ((tank0[0] == N / 2 + 4 && (tank0[1] == N - z || tank0[1] == N + 1 - z || tank0[1] == N + 2 - z)) || tank0[0] - 1 < 2 || (abs(tank0[1] - tank1[1])<3 && tank0[0] - 2 == tank1[0] + 1) || (abs(tank0[1] - tank2[1])<3 && tank0[0] - 2 == tank2[0] + 1))//防止出界和相撞

;

else

tank0[0] = tank0[0] - 1;

break;

case 'd':

if ((tank0[0] == N / 2 - 2 && (tank0[1] == N - z || tank0[1] == N + 1 - z || tank0[1] == N + 2 - z)) || tank0[0] + 1 > N - 1 || (abs(tank0[1] - tank1[1])<3 && tank0[0] + 2 == tank1[0] - 1) || (abs(tank0[1] - tank2[1])<3 && tank0[0] + 2 == tank2[0] - 1))//防止坦克与star的左侧装甲相撞

;

else

tank0[0] = tank0[0] + 1;

break;

case 's':

if ((tank0[1] == N - z - 1 && (tank0[0] == N / 2 - 1 || tank0[0] == N / 2 || tank0[0] == N / 2 + 1 || tank0[0] == N / 2 + 2 || tank0[0] == N / 2 + 3)) || tank0[1] + 1 > N - 1 || (abs(tank0[0] - tank1[0])<3 && tank0[1] + 2 == tank1[1] - 1) || (abs(tank0[0] - tank2[0])<3 && tank0[1] + 2 == tank2[1] - 1))

;

else

tank0[1] = tank0[1] + 1;

break;

case 'w':

if (tank0[1] - 1 < 3 || (abs(tank0[0] - tank1[0])<3 && tank0[1] - 2 == tank1[1] + 1) || (abs(tank0[0] - tank2[0])<3 && tank0[1] - 2 == tank2[1] + 1))

;

else

tank0[1] = tank0[1] - 1;

break;

}

}

else; //坦克不移动仅仅改变方向

color(n);

tank(tank0, c, n);

return tank0[2];

}

int Ptank(int tank[2], int n, int N)

{

int i;

if (n == 0)

{

tank[0] = N / 2 + 1;//我方坦克位置

tank[1] = N / 2 + 1;

}

else//敌方坦克位置

{

srand((unsigned)(time(NULL) + 1));

i = 1 + rand() % 3;

if (i == 1)

{

tank[0] = 2;

tank[1] = 3;

}

if (i == 2)

{

tank[0] = N - 1;

tank[1] = 3;

}

if (i == 3)

{

tank[0] = 2;

tank[1] = N;

}

if (i == 4)

{

tank[0] = N - 1;

tank[1] = N;

}

}

return tank[2];

}

void brand()

{

int brandtank1[2] = { 10,10 };

int brandtank2[2] = { 17,4 };

int i;

PlaySound(TEXT("tankmove.wav"), NULL, SND\_ASYNC | SND\_NODEFAULT);

tank(brandtank1, 'd', 1);

tank(brandtank2, 's', 11);

Sleep(500);

for (i = 1; i <= 8; i++)

{

if (i == 8)

{

brandtank1[2] = Move(brandtank1, brandtank2, brandtank2, 'w', 'd', 1, 0, 30);

}

else

{

brandtank1[2] = Move(brandtank1, brandtank2, brandtank2, 'd', 'd', 1, 0, 30);

}

Sleep(300);

}

PlaySound(NULL, NULL, NULL);

Sleep(100);

fire(brandtank1, brandtank2, brandtank2, 'w', 's', 's', 3, 30, 3);

Sleep(1000);

PlaySound(TEXT("background1.wav"), NULL, SND\_ASYNC | SND\_NODEFAULT);

color(15);

for (i = 1; i <= 9; i++)//T

{

if (i <= 5)

{

gotoxy(i, 12);

printf("■");

}

else

{

gotoxy(3, 12 + i - 5);

printf("■");

}

Sleep(75);

}

color(12);

for (i = 1; i <= 9; i++)//a

{

if (i <= 3)

{

gotoxy(6 + i, 14);

printf("■");

}

if (i == 4)

{

gotoxy(7, 15);

printf("■");

}

if (i == 5)

{

gotoxy(9, 15);

printf("■");

}

if (i <= 9 && i >= 6)

{

gotoxy(6 + i - 5, 16);

printf("■");

}

Sleep(75);

}

color(11);

for (i = 1; i <= 7; i++)//n

{

if (i <= 3)

{

gotoxy(11 + i, 14);

printf("■");

}

if (i <= 5 && i >= 4)

{

gotoxy(12, 14 + i - 3);

printf("■");

}

if (i <= 7 && i >= 6)

{

gotoxy(14, 14 + i - 5);

printf("■");

}

Sleep(75);

}

color(13);

for (i = 1; i <= 8; i++)//k

{

if (i <= 5)

{

gotoxy(16, 12 + i - 1);

printf("■");

}

if (i == 6)

{

gotoxy(18, 14);

printf("■");

}

if (i == 7)

{

gotoxy(17, 15);

printf("■");

}

if (i == 8)

{

gotoxy(18, 16);

printf("■");

}

Sleep(75);

}

color(5);

for (i = 1; i <= 9; i++)//w

{

if (i <= 3)

{

gotoxy(19 + i, 13 + i);

printf("■");

}

if (i == 4 || i == 5)

{

gotoxy(19 + i, 16 - (i - 3));

printf("■");

}

if (i == 6 || i == 7)

{

gotoxy(19 + i, 14 + (i - 5));

printf("■");

}

if (i == 8 || i == 9)

{

gotoxy(19 + i, 16 - (i - 7));

printf("■");

}

Sleep(75);

}

color(10);

for (i = 1; i <= 9; i++)//a

{

if (i <= 3)

{

gotoxy(29 + i, 14);

printf("■");

}

if (i == 4)

{

gotoxy(30, 15);

printf("■");

}

if (i == 5)

{

gotoxy(32, 15);

printf("■");

}

if (i <= 9 && i >= 6)

{

gotoxy(29 + i - 5, 16);

printf("■");

}

Sleep(75);

}

color(3);

for (i = 1; i <= 5; i++)//r

{

if (i <= 3)

{

gotoxy(35, 13 + i);

printf("■");

}

if (i == 4 || i == 5)

{

gotoxy(35 + (i - 3), 16 - (i - 3));

printf("■");

}

Sleep(75);

}

Sleep(1000);

system("cls");

}

void end(int n)

{

int i;

if (n == 1)

{

PlaySound(TEXT("win.wav"), NULL, SND\_ASYNC | SND\_NODEFAULT);

color(12);

for (i = 0; i <3; i++)

{

gotoxy(7, 10 + i);

printf("■");

Sleep(100);

}

for (i = 0; i <2; i++)

{

gotoxy(8, 13 + i);

printf("■");

Sleep(100);

}

gotoxy(9, 15);

printf("■");

Sleep(100);

for (i = 0; i <2; i++)

{

gotoxy(10, 14 - i);

printf("■");

Sleep(100);

}

for (i = 0; i <2; i++)

{

gotoxy(11, 12 - i);

printf("■");

Sleep(100);

}

gotoxy(12, 10);

printf("■");

Sleep(100);

for (i = 0; i <2; i++)

{

gotoxy(13, 11 + i);

printf("■");

Sleep(100);

}

for (i = 0; i <2; i++)

{

gotoxy(14, 13 + i);

printf("■");

Sleep(100);

}

gotoxy(15, 15);

printf("■");

Sleep(100);

for (i = 0; i <2; i++)

{

gotoxy(16, 14 - i);

printf("■");

Sleep(100);

}

for (i = 0; i <3; i++)

{

gotoxy(17, 12 - i);

printf("■");

Sleep(100);

}

for (i = 21; i <30; i++) //I

{

gotoxy(i + 1, 10);

printf("■");

Sleep(100);

}

for (i = 1; i <6; i++)

{

gotoxy(25 + 1, 10 + i);

printf("■");

Sleep(100);

}

for (i = 21; i <30; i++) {

gotoxy(i + 1, 15);

printf("■");

Sleep(100);

}

for (i = 0; i <6; i++) //N

{

gotoxy(35, 10 + i);

printf("■");

Sleep(100);

}

for (i = 35; i <41; i++) {

gotoxy(i + 1, i - 25);

printf("■");

Sleep(100);

}

for (i = 0; i <6; i++) {

gotoxy(42, 15 - i);

printf("■");

Sleep(100);

}

for (i = 0; i <4; i++) //!

{

gotoxy(47, 10 + i);

printf("■");

Sleep(100);

}

gotoxy(47, 15);

printf("■");

Sleep(100);

}

if (n == 2)

{

PlaySound(TEXT("fail.wav"), NULL, SND\_ASYNC | SND\_NODEFAULT);

color(11);

for (i = 0; i < 6; i++)//L

{

gotoxy(7, 10 + i);

printf("■");

Sleep(100);

}

for (i = 0; i < 3; i++)

{

gotoxy(8 + i, 15);

printf("■");

Sleep(100);

}

for (i = 0; i < 6; i++)//O

{

gotoxy(14, 10 + i);

printf("■");

Sleep(100);

}

for (i = 0; i < 3; i++)

{

gotoxy(15 + i, 15);

printf("■");

Sleep(100);

}

for (i = 0; i < 6; i++)

{

gotoxy(18, 15 - i);

printf("■");

Sleep(100);

}

for (i = 0; i < 4; i++)

{

gotoxy(18 - i, 10);

printf("■");

Sleep(100);

}

for (i = 0; i < 5; i++)//S

{

gotoxy(26 - i, 10);

printf("■");

Sleep(100);

}

gotoxy(22, 11);

printf("■");

Sleep(100);

for (i = 0; i < 5; i++)

{

gotoxy(22 + i, 12);

printf("■");

Sleep(100);

}

for (i = 0; i < 2; i++)

{

gotoxy(26, 13 + i);

printf("■");

Sleep(100);

}

for (i = 0; i < 5; i++)

{

gotoxy(26 - i, 15);

printf("■");

Sleep(100);

}

for (i = 0; i < 5; i++)//E

{

gotoxy(34 - i, 10);

printf("■");

Sleep(100);

}

for (i = 0; i < 6; i++)

{

gotoxy(30, 10 + i);

printf("■");

Sleep(100);

}

for (i = 0; i < 4; i++)

{

gotoxy(31 + i, 12);

printf("■");

Sleep(100);

}

for (i = 0; i < 4; i++)

{

gotoxy(31 + i, 15);

printf("■");

Sleep(100);

}

for (i = 0; i < 4; i++) //!

{

gotoxy(37, 10 + i);

printf("■");

Sleep(100);

}

gotoxy(37, 15);

printf("■");

Sleep(100);

for (i = 0; i < 4; i++) //!

{

gotoxy(40, 10 + i);

printf("■");

Sleep(100);

}

gotoxy(40, 15);

printf("■");

Sleep(100);

for (i = 0; i < 4; i++) //!

{

gotoxy(43, 10 + i);

printf("■");

Sleep(100);

}

gotoxy(43, 15);

printf("■");

Sleep(100);

}

Sleep(1000);

system("cls");

}

int game()

{

int mytank[2];//我方坦克位置

int enemytank1[2];//敌方坦克位置

int enemytank2[2];//敌方坦克位置

char c, b, e1, e2, f, o;//方向

int i, t, n, q, z, y, N;//q是starfire返回值 ，z第一，第三列装甲，值不变，y控制star开火

int r;// 苹果正中心装甲层数，一发炮弹击穿一层，fire，star中的参数

int c0, c1 = 9;//坦克颜色

int score = 0;//初始化得分

int goal;//目标得分

int life;//生命值

color(2);

printf("输入玩家名字:");

char name[10];

scanf("%s", name);

PlaySound(TEXT("button.wav"), NULL, SND\_ASYNC | SND\_NODEFAULT);

printf("选择难度:1.一般 2.困难 3.地狱 0.退出\n");

int j;

scanf("%d", &j);

PlaySound(TEXT("button.wav"), NULL, SND\_ASYNC | SND\_NODEFAULT);

printf("选择坦克颜色:1.红 2.黄 3.蓝 0.退出\n");

int p;

scanf("%d", &p);

printf("选择界面大小:1.大 2.中 3.小 0.退出\n");

int m;

scanf("%d", &m);

PlaySound(TEXT("button.wav"), NULL, SND\_ASYNC | SND\_NODEFAULT);

switch (m)

{

case 1:

N = 30;

break;

case 2:

N = 25;

break;

case 3:

N = 20;

break;

case 0:

return 0;

default:

{

gotoxy(15, 15);

printf("操作错误 ，请重新操作");

system("pause");

}

}

switch (p)

{

case 1:

c0 = 4;

break;

case 2:

c0 = 6;

break;

case 3:

c0 = 1;

break;

case 0:

return 0;

default:

{

gotoxy(N + 3, N + 3);

printf("操作错误 ，请重新操作");

system("pause");

}

}

system("clc");

PlaySound(TEXT("background.wav"), NULL, SND\_ASYNC | SND\_NODEFAULT);//播放后台音乐

gotoxy(N + 7, 4);

color(12);

printf("%d", score);

desk(N);

b = c = o = e1 = e2 = f = 'w';//初始化坦克形状

mytank[2] = Ptank(mytank, 0, N);

tank(mytank, c, c0);

switch (j)

{

case 0:

return 0;

case 1:

{

enemytank1[2] = Ptank(enemytank1, 1, N);

tank(enemytank1, e1, c1);

enemytank2[0] = N + 4;

enemytank2[1] = N + 4;

r = 3;// 星星四周装甲层数

z = 3;// 星星两侧装甲无法被打穿

y = 3;// 控制星星前方炮台什么时候开火

star(r, N);

wall(j, N);

life = 3;

goal = 10;

break;

}

case 2:

{

enemytank1[2] = Ptank(enemytank1, 1, N);

tank(enemytank1, e1, c1);

enemytank2[0] = N + 4;

enemytank2[1] = N + 4;

y = 2;// 控制星星前方炮台什么时候开火

r = 2;// 星星四周装甲层数

z = 2;

star(r, N);

wall(j, N);

life = 2;

goal = 15;

break;

}

case 3:

{

enemytank1[2] = Ptank(enemytank1, 1, N);

tank(enemytank1, e1, c1);

enemytank2[2] = Ptank(enemytank2, 2, N);

tank(enemytank2, e2, c1);

y = 1;// 控制星星前方炮台什么时候开火

r = 1;

z = 1;

star(r, N);

wall(j, N);

life = 1;

goal = 20;

break;

}

default:

{

gotoxy(N + 3, N + 3);

printf("操作错误 ，请重新操作");

system("pause");

break;

}

}

color(12);

gotoxy(N + 7, 3);

printf("%d", life);

while (1)

{

if (!\_kbhit())//当没有键按下时，坦克随机移动

{

srand((unsigned)(time(NULL) + 1));

if (j <= 2)

t = 1;

else

{

t = rand() % 2;

}

if (t == 1)//移动enemytank1

{

srand((unsigned)(time(NULL) + 1));

int t2, ti = 10, t3; //根据游戏难度调整敌方坦克性能

t2 = rand() % (6 - j); //控制敌方坦克追踪我方坦克的概率

t3 = rand() % (5 - j); //控制敌方开火的概率

int judgedx, judgedy, judgesx, judgesy, abc;

judgedx = (mytank[0] - enemytank1[0]); //敌我双方横坐标之差

judgedy = (mytank[1] - enemytank1[1]); //敌我双方纵坐标之差

judgesx = abs(judgedx); //敌我双方横向距离

judgesy = abs(judgedy); //敌我双方纵向距离

if (t2 == 0 || t2 == 1)

{

if (judgesx == 0) //如果敌我双方在同一竖直线上

{

abc = 0; ti = t3;

}

else if (judgesy == 0) //如果敌我双方在同一水平线上

{

abc = 1; ti = t3;

}

else if (judgesx <= 3)//防止敌我双方坦克在水平方向卡住

abc = 0;

else if (judgesy <= 3)//防止敌我双方坦克在竖直方向卡住

abc = 1;

else if (judgesx > judgesy)

abc = 1;

else if (judgesx <= judgesy)

abc = 0;

if (abc == 1) //敌方坦克进行水平的方向运动

{

if (judgedx > 0) //我方在敌方右侧

e1 = 'd';

else e1 = 'a'; //我方在敌方左侧

}

if (abc == 0) //敌方坦克进行竖直方向运动

{

if (judgedy > 0)

e1 = 's';

else e1 = 'w';

}

}

else

{

i = rand() % 4;

switch (i) //敌方坦克进行随机运动

{

case 0:

e1 = 'w';

break;

case 1:

e1 = 's';

break;

case 2:

e1 = 'a';

break;

case 3:

e1 = 'd';

break;

case 4:

e1 = f;

break;

}

}

if (ti == 0)

{

n = fire(enemytank1, mytank, enemytank2, e1, c, e2, z, N, r);

if (n == 1)

{

life--;

gotoxy(N + 7, 3);

color(12);

printf("%d", life);

mytank[2] = Ptank(mytank, 0, N);

tank(mytank, c, c0);

}

if (n == 2)

{

y = y - 1;

r = r - 1;

q = starfire(y, enemytank1, e1, N);

if (q == 1)//苹果开炮了

{

enemytank1[2] = Ptank(enemytank1, 1, N);

tank(enemytank1, e1, c1);

}

if (r == -1)

{

mciSendString(TEXT("play x.wma wait"), 0, 0, 0);

gotoxy(N / 2, N / 2);

PlaySound(NULL, NULL, NULL);//背景音乐停止

return 2;//lose

}

}

if (j == 3 && n == 3)

{

enemytank2[2] = Ptank(enemytank2, 1, N);

tank(enemytank2, e2, c1);

}

if (life == 0)

{

gotoxy(N / 2, N / 2);

PlaySound(NULL, NULL, NULL);

return 2;//lose

}

}

else

Sleep(200);

enemytank1[2] = Move(enemytank1, mytank, enemytank2, e1, f, c1, z, N);

f = e1;

}

if (j == 3)//移动第二辆坦克

{

srand((unsigned)(time(NULL) + 10));

int t22, ts = 10, t32; //根据游戏难度调整敌方坦克性能

t22 = rand() % (6 - j); //控制敌方坦克追踪我方坦克的概率

t32 = rand() % (4 - j); //控制敌方开火的概率

int judgedx2, judgedy2, judgesx2, judgesy2, abc2;

judgedx2 = (mytank[0] - enemytank2[0]); //敌我双方横坐标之差

judgedy2 = (mytank[1] - enemytank2[1]); //敌我双方纵坐标之差

judgesx2 = abs(judgedx2); //敌我双方横向距离

judgesy2 = abs(judgedy2); //敌我双方纵向距离

if (t22 == 0 || t22 == 1)

{

if (judgesx2 == 0) //敌我双方在同一竖直线上

{

abc2 = 0; ts = t32;

}

else if (judgesy2 == 0) //敌我双方在同一水平线上

{

abc2 = 1; ts = t32;

}

else if (judgesx2 <= 3)//防止敌我双方坦克在水平方向卡住

abc2 = 0;

else if (judgesy2 <= 3)//防止敌我双方坦克在竖直方向卡住

abc2 = 1;

else if (judgesx2 > judgesy2)

abc2 = 1;

else if (judgesx2 <= judgesy2)

abc2 = 0;

if (abc2 == 1) //敌方坦克进行水平的方向运动

{

if (judgedx2 > 0) //我方在敌方右侧

e2 = 'd';

else e2 = 'a'; //我方在敌方左侧

}

if (abc2 == 0) //敌方坦克进行竖直方向运动

{

if (judgedy2 > 0)

e2 = 's';

else e2 = 'w';

}

}

else

{

i = rand() % 5;

switch (i) //敌方坦克进行随机运动

{

case 0:

e2 = 'w';

break;

case 1:

e2 = 's';

break;

case 2:

e2 = 'a';

break;

case 3:

e2 = 'd';

break;

case 4:

e2 = 'o';

break;

}

}

if (ts == 0)

{

n = fire(enemytank2, mytank, enemytank1, e2, c, e1, z, N, r);

if (n == 1)

{

life--;

gotoxy(N + 7, 3);

color(12);

printf("%d", life);

mytank[2] = Ptank(mytank, 0, N);

tank(mytank, c, c0);

}

if (n == 2)

{

y = y - 1;

r = r - 1;

q = starfire(y, enemytank1, e1, N);

if (q == 1)//星星前方炮台开炮

{

enemytank2[2] = Ptank(enemytank2, 1, N);

tank(enemytank2, e2, c1);

}

if (r == -1)

{

mciSendString(TEXT("play x.wma wait"), 0, 0, 0);

gotoxy(N / 2, N / 2);

PlaySound(NULL, NULL, NULL);//背景音乐停止

return 2;//lose

}

}

if (n == 3)

{

enemytank1[2] = Ptank(enemytank1, 1, N);

tank(enemytank1, e1, c1);

}

if (life == 0)

{

gotoxy(N / 2, N / 2);

PlaySound(NULL, NULL, NULL);

return 2;//lose

}

}

else

Sleep(100);

enemytank2[2] = Move(enemytank2, mytank, enemytank1, e2, o, c1, z, N);

o = e2;

}

}

else//移动己方坦克

{

c = \_getch();

if (c == 'a' || c == 's' || c == 'd' || c == 'w')

{

mytank[2] = Move(mytank, enemytank1, enemytank2, c, b, c0, z, N);

b = c;

}

if (c == ' ')

{

c = b;

n = fire(mytank, enemytank1, enemytank2, c, e1, e2, z, N, r);

if (n == 1)

{

score++;

gotoxy(N + 7, 4);

color(12);

printf("%d", score);

enemytank1[2] = Ptank(enemytank1, 1, N);

tank(enemytank1, e1, c1);

}

if (j == 3 && n == 3)

{

score++;

gotoxy(N + 7, 4);

color(12);

printf("%d", score);

enemytank2[2] = Ptank(enemytank2, 1, N);

tank(enemytank2, e2, c1);

}

if (n == 2)

{

y = y - 1;

r = r - 1;

q = starfire(y, mytank, c, N);

if (q == 1)//苹果开炮了

{

life--;

gotoxy(N + 7, 3);

color(12);

printf("%d", life);

mytank[2] = Ptank(mytank, 0, N);

tank(mytank, c, c0);

}

if (life == 0)

{

mciSendString(TEXT("play x.wma wait"), 0, 0, 0);

gotoxy(N / 2, N / 2);

PlaySound(NULL, NULL, NULL);

return 2;//lose

}

}

if (r == -1)

{

gotoxy(N / 2, N / 2);

PlaySound(NULL, NULL, NULL);

return 2;//lose

}

}

if (score == goal)

{

gotoxy(N / 2, N / 2);

PlaySound(NULL, NULL, NULL);

return 1;//win

}

}

if (c == 27)

return 0;

}

}

void main() //主函数

{

int i, n;

brand(); //播放开场动画

while (1) //通过循环运行游戏

{

gotoxy(15, 10);

color(10);

printf("1.新游戏 2.退出\n");

scanf("%d", &i);

PlaySound(TEXT("button.wav"), NULL, SND\_ASYNC | SND\_NODEFAULT);

system("cls");

switch (i)

{

case 1:

n = game();

system("cls");

end(n);

//system("pause");

break;

case 2:

return;

default:

{

printf("操作错误 ，请重新操作");

system("pause");

}

}

}

}