#include <stdio.h>

#include <windows.h>

#include <conio.h>

#include <stdlib.h>

#include <list>

using namespace std;

#define ARRIBA 72

#define IZQUIERDA 75

#define DERECHA 77

#define ABAJO 80

void gotoxy(int x, int y){

HANDLE hCon;

hCon = GetStdHandle(STD\_OUTPUT\_HANDLE);

COORD dwPos;

dwPos.X = x;

dwPos.Y = y;

SetConsoleCursorPosition(hCon, dwPos);

}

void OcultarCursor(){

HANDLE hCon;

hCon = GetStdHandle(STD\_OUTPUT\_HANDLE);

CONSOLE\_CURSOR\_INFO cci;

cci.dwSize = 2;

cci.bVisible = FALSE;

SetConsoleCursorInfo(hCon,&cci);

}

void pintar\_limites(){

for(int i=2;i<78;i++){

gotoxy(i,3); printf("%c",205);

gotoxy(i,33); printf("%c",205);

}

for(int i=4;i<33;i++){

gotoxy(2,i); printf("%c",186);

gotoxy(77,i); printf("%c",186);

}

gotoxy(2,3); printf("%c",201);

gotoxy(2,33); printf("%c",200);

gotoxy(77,3); printf("%c",187);

gotoxy(77,33); printf("%c",188);

}

class NAVE{

int x,y;

int corazones;

int vidas;

public:

NAVE(int \_x, int \_y, int \_corazones, int \_vidas);

int X(){ return x;}

int Y(){ return y;}

int VID(){return vidas;}

void COR(){corazones--;}

void pintar();

void borrar();

void mover();

void pintar\_corazones();

void morir();

};

NAVE::NAVE(int \_x, int \_y, int \_corazones, int \_vidas){

x=\_x;

y=\_y;

corazones= \_corazones;

vidas= \_vidas;

}

void NAVE::pintar(){

gotoxy(x,y); printf(" %c",30);

gotoxy(x,y+1); printf(" %c%c%c",40,207,41);

gotoxy(x,y+2); printf("%c%c %c%c",30,190,190,30);

}

void NAVE::borrar(){

gotoxy(x,y); printf(" ");

gotoxy(x,y+1); printf(" ");

gotoxy(x,y+2); printf(" ");

}

void NAVE::mover(){

if(kbhit()){

char tecla = getch();

borrar();

if(tecla==IZQUIERDA && x>3) x--;

if(tecla==DERECHA && x+6 <77) x++;

if(tecla==ARRIBA && y>4) y--;

if(tecla==ABAJO && y+3 <33) y++;

if(tecla == 'e') corazones--;

pintar();

pintar\_corazones();

}

}

void NAVE::pintar\_corazones(){

gotoxy(50,2); printf("VIDAS %d",vidas);

gotoxy(64,2); printf("SALUD");

gotoxy(70,2); printf(" ");

for(int i=0;i<corazones;i++){

gotoxy(70+i,2); printf("%c",3);

}

}

void NAVE::morir(){

if(corazones==0){

borrar();

gotoxy(x,y); printf(" \*\* ");

gotoxy(x,y+1); printf(" \*\*\*\* ");

gotoxy(x,y+2); printf(" \*\* ");

Sleep(250);

borrar();

gotoxy(x,y); printf(" \* \*\* \*");

gotoxy(x,y+1); printf(" \*\*\*\* ");

gotoxy(x,y+2); printf(" \* \*\* \*");

Sleep(250);

borrar();

vidas--;

corazones=3;

pintar\_corazones();

pintar();

}

}

class AST{

int x,y;

public:

AST(int \_x, int \_y):x(\_x),y(\_y){}

void pintar();

void mover();

void choque(class NAVE &N);

int X(){return x;}

int Y(){return y;}

};

void AST::pintar(){

gotoxy(x,y); printf("%c",184);

}

void AST::mover(){

gotoxy(x,y); printf(" ");

y++;

if(y>32){

x= rand()%71+4;

y=4;

}

pintar();

}

void AST::choque(class NAVE &N){

if(x>= N.X() && x<N.X()+6 && y>=N.Y() && y<=N.Y()+2){

N.COR();

N.borrar();

N.pintar();

N.pintar\_corazones();

x=rand()%71+4;

y=4;

}

}

class BALA{

int x,y;

public:

BALA(int \_x, int \_y): x(\_x), y(\_y){}

int X(){return x;}

int Y(){return y;}

void mover();

bool fuera();

};

void BALA::mover(){

gotoxy(x,y); printf(" ");

y--;

gotoxy(x,y); printf("\*");

}

bool BALA::fuera(){

if(y==4) return true;

return false;

}

int main(){

OcultarCursor();

pintar\_limites();

NAVE N(37,30,6,5);

N.pintar();

N.pintar\_corazones();

list <AST\*> A;

list< AST\*>::iterator itA;

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

A.push\_back(new AST(rand()%75+9, rand()%5+4));

}

list<BALA\*> B;

list<BALA\*>::iterator it;

bool game\_over = false;

int puntos=0;

while(!game\_over){

gotoxy(4,2);printf("Score %d",puntos);

if(kbhit()){

char tecla= getch();

if(tecla=='a')

B.push\_back(new BALA(N.X()+2,N.Y()-1));

}

for(it=B.begin(); it!=B.end(); it++){

(\*it)->mover();

if((\*it)->fuera()){

gotoxy((\*it)->X(),(\*it)->Y()); printf(" ");

delete(\*it);

it=B.erase(it);

}

}

for(itA = A.begin(); itA != A.end();itA++){

(\*itA)->mover();

(\*itA)->choque(N);

}

for(itA = A.begin(); itA != A.end();itA++){

for(it=B.begin(); it!=B.end(); it++){

if((\*itA)->X()==(\*it)->X() && ( (\*itA)->Y()+1==(\*it)->Y() || (\*itA)->Y()==(\*it)->Y() ) ){

gotoxy((\*it)->X(),(\*it)->Y());printf(" ");

delete(\*it);

it=B.erase(it);

A.push\_back(new AST(rand()%74+3,4));

gotoxy((\*itA)->X(),(\*itA)->Y());printf(" ");

delete(\*itA);

itA=A.erase(itA);

puntos+=5;

}

}

}

if(N.VID()==0) game\_over = true;

N.morir();

N.mover();

Sleep(30);

}

return 0;

}