



Avioane



LUCRARE PENTRU OBTINEREA ATESTATULUI
PROFESIONAL LA INFORMATICĂ

Coordonator,
prof. **Kalmar Violeta**

Elev: **Boloș Mihai-Răzvan**
Clasa a XII-a D

Baia Mare

2023

Colegiul Național “Gheorghe Șincai” Baia Mare

Avioane

LUCRARE PENTRU OBTINEREA ATESTATULUI
PROFESIONAL LA INFORMATICĂ

Coordonator,
prof. **Kalmar Violeta**

Elev **Boloș Mihai-Răzvan**
Clasa a XII-a D

Baia Mare

2023



Cuprins

1. Introducere.....	2
1.1. Descrierea programului.....	2
2. Ghid de utilizare.....	3
3. Probleme și soluții în implementare	5
3.1. Idei de dezvoltare.....	6
4. Bibliografie	7
5. Cod sursă.....	8



1. Introducere

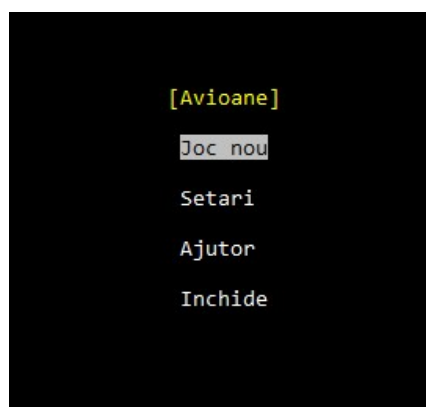
Ideea de la care am pornit pentru realizarea acestui proiect mi-a venit de la jocul de avioane pe hârtia de pătrățele pe care l-am îndrăgit atât de mult în copilărie. Pentru realizarea aplicației *Avioane* am utilizat limbajul de programare C++, deoarece este un limbaj de programare eficient, practic și actual.

1.1. Descrierea programului

Avioane este o aplicație realizată cu ajutorul programului Code::Blocks, unde aceasta funcționează în consola de comandă a acestui program. Inițial, programul se deschide fiind necesară alegerea dimensiunii ferestrei de joc, ulterior aceasta poate fi schimbată din interiorul aplicației propriu-zise.

Programul constă în patru meniuri selectabile care realizează diferite operațiuni pentru o experiență de joc cât mai simplificată și atrăgătoare. Prima opțiune a programului este inactivă și conține numele aplicației, iar sub aceasta urmează celelalte opțiuni, selectabile.

Dacă o opțiune selectată este una marginală (prima sau ultima), la apăsarea tastei pentru a trece la opțiunea precedentă, respectiv la opțiunea următoare, programul va selecta opțiunea aflată în capătul opus al listei de opțiuni.

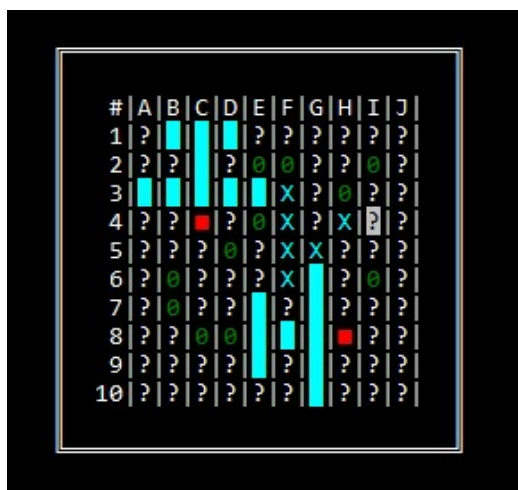




2. Ghid de utilizare

Deplasarea prin aplicație se realizează cu ajutorul săgeților tastaturii, iar alegerea unei opțiuni selectate se realizează prin apăsarea tastei *Enter*. Ieșirea din orice meniu se realizează prin alegerea opțiunii *Ieșire* sau cu ajutorul tastei *Esc*. La apăsarea tastei *A* sau a tastei *H* (*ajutor*, respectiv *help*), se va deschide meniul *Ajutor* care conține informații despre funcționalitatea programului. Ieșirea din aplicație se poate realiza și cu ajutorul tastei *End*, care va închide fereastra de joc.

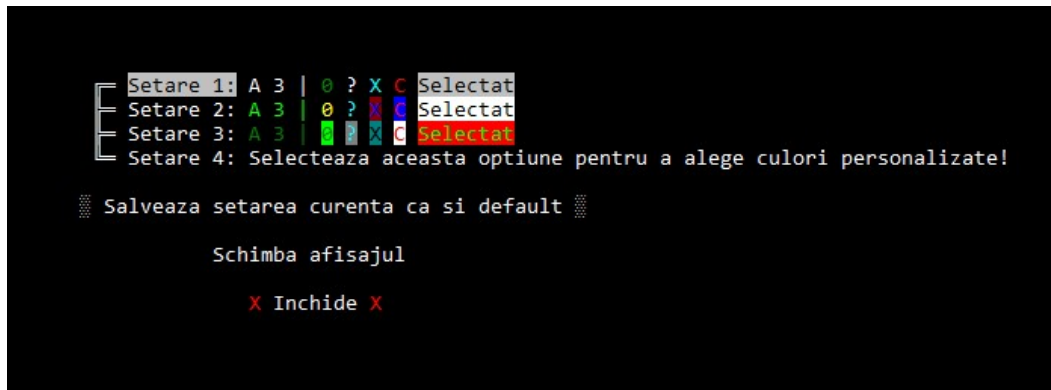
Meniul ***Joc nou*** este principalul meniu al acestei aplicații, acesta permite introducerea numărului de avioane cu care se va juca. Numărul introdus trebuie să fie cuprins între 1 și 4, în caz contrar, aplicația va afișa un mesaj reprezentativ. După introducerea numărului de avioane, programul afișează pe ecran un pătrat de 10x10, reprezentând suprafața standard de joc, plină de ?, deoarece pozițiile respective nu au fost descoperite. Odată cu alegerea unei opțiuni selectate, ? se va transforma în 0, X, sau C. Fiecare dintre aceste caractere are o semnificație: 0, în cazul în care poziția respectivă nu este o parte de avion, X în cazul în care poziția este o parte de avion, iar C dacă poziția este capul avionului. În acest ultim caz, avionul se va colora specific, pentru a ști că acel avion a fost descoperit, iar acesta să nu fie confundat cu alt avion în procesul său de descoperire. După descoperirea tuturor avioanelor, se va afișa numărul de încercări folosite pentru câștigarea jocului. Deplasarea prin pătrat va fi în continuare posibilă, iar ieșirea din meniu se va realiza cu ajutorul tastei *Esc*.



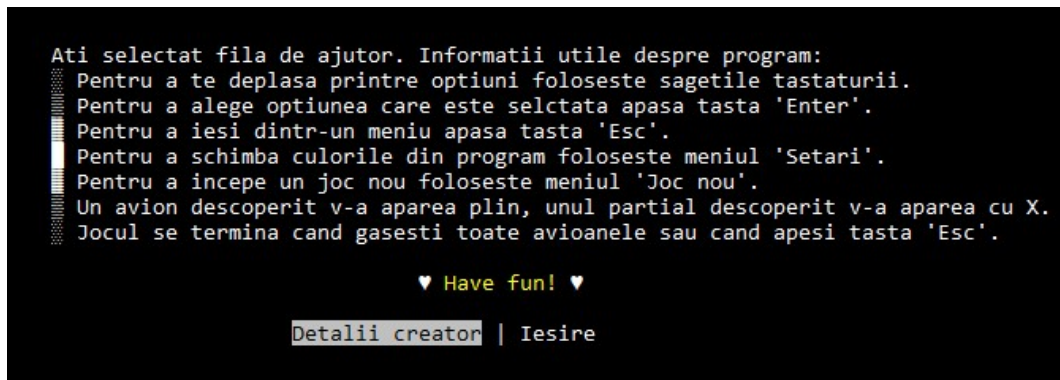
Meniul ***Setari*** permite personalizarea culorilor din întregul program. Meniul are trei setări predefinite, iar cea de-a patra opțiune permite selectarea unor culori personalizate pentru toate tipurile de caractere folosite. Sub cea de-a patra opțiune, se găsește opțiunea care permite memorarea culorilor selectate ca fiind implicite, acest lucru înseamnă că la redeschiderea aplicației, culorile vor rămâne identice cu cele de dinaintea închiderii aplicației. Între această opțiune și opțiunea de închidere a meniului, se găsește opțiunea care permite schimbarea afișajului, acest lucru schimbă dimensiunea ferestrei de joc. Cele două dimensiuni disponibile sunt: cea normală care are o dimensiune minimă obligatorie,



de 1280x800 pixeli, iar cealaltă dimensiune este limitată de dimensiunea ecranului calculatorului, deoarece fereastra de joc va acoperi întreaga suprafață disponibilă, aceasta fiind în modul *fullscreen*. Ultima opțiune de pe acest meniu este opțiunea de închidere a meniului **Setari**, așadar prin această opțiune se revine la meniul inițial al programului.



Meniul **Ajutor** afișează câteva instrucțiuni utile despre program, cum ar fi modul de joc și deplasarea prin program. Sub aceste informații se găsesc două opțiuni selectabile, una dintre ele este cea de închidere a meniului cu informații, iar cealaltă reprezintă fila cu detalii despre creatorul aplicației. În interiorul meniului cu detalii despre creator, pe lângă opțiunea de ieșire, se găsește și opțiunea de deschidere a documentației, acest lucru permite vizualizarea acestor detalii complexe despre program, fiind deschise din interiorul aplicației. Odată cu deschiderea documentației, fereastra de joc se va închide. Acest meniu poate fi deschis din orice alt meniu al programului, cu ajutorul tastelor *A* sau *H*.



Meniul **Inhide** este aflat ultimul în meniul principal, însă acesta poate fi acționat cu ajutorul tastei *Esc*. La alegerea acestei opțiuni, este necesară confirmarea deciziei de a închide programul, dacă răspunsul este afirmativ, programul va afișa iconița reprezentativă a programului, iar după puțin timp aplicația se va închide. Pentru o ieșire rapidă din program, se poate apăsa oricând tasta *End* a tastaturii, astfel programul se va opri forțat.



3. Probleme și soluții în implementare

Câteva probleme care au apărut în dezvoltarea acestei aplicații, dar și soluțiile folosite pentru realizarea cu succes a acestui proiect:

- Avioanele sunt plasate aleatoriu în matricea folosită în joc, așadar nu există probleme la unul sau la două avioane, însă în cazul cu trei sau patru avioane există posibilitatea să nu încapă toate avioanele, astfel programul să rămână blocat. Pentru rezolvarea acestei probleme, programul tratează particular cazurile cu trei sau patru avioane. În cazul cu 3 avioane, programul plasează un avion în unul dintre cele patru colțuri ale matricei, într-o orientare aleatorie, ulterior cele două avioane rămase încap în matrice cu siguranță. În cazul cu patru avioane, programul pune câte un avion în fiecare colț într-o orientare aleatorie.
- Avioanele trebuie să fie generate în memorie, iar afișarea lor să se realizeze în funcție de partea avionului descoperită. Descoperirea avioanelor se realizează prin conectarea a două matrice cu dimensiune identică, una dintre ele fiind cea care conține avioanele propriu-zise, iar una care este folosită pentru interacțiunea matricei cu utilizatorul. La apăsarea tastei *Enter*, programul verifică poziția din matricea cu avioane și modifică în matricea plină cu semne de întrebare, în funcție de ceea ce este în matrice cu *0*, *X* sau *C*. O altă problemă a fost numărarea încercărilor, deoarece, odată cu terminarea jocului sau la încercarea unei poziții deja descoperite, programul nu trebuie să numere acele încercări.
- Afișarea textului este centrată, acest lucru este realizat în mod diferit pentru fiecare tip de afișaj, așadar la lansarea în execuție a programului, acesta calculează spațiul disponibil în consola de joc și pentru fiecare linie de text, programul mută cursorul la mijlocul ecranului, minus jumătatea lungimii textului de afișat. Acest lucru se realizează pentru fiecare meniu care se afișează pe parcursul jocului.



3.1. Idei de dezvoltare

Jocul de avioane poate fi dezvoltat, astfel încât acesta să fie mai atractiv pentru utilizatori. Câteva idei în acest sens:

- Folosirea unor librării speciale, cum ar fi SDL2, pentru a crea o fereastră dedicată, adăugând posibilitatea de a juca cu mouse-ul și folosind culori, meniuri și imagini interactive.
- Folosirea unei baze de date pentru a salva un joc și posibilitatea de continuare a acestuia și de pe alt dispozitiv.
- Opțiunea de joc online, astfel fiind posibilă adăugarea de avioane de către utilizator și descoperirea avioanelor adăugate de alt utilizator.
- Folosirea unui program care descoperă avioanele puse de utilizator care dă un scor în funcție de complexitatea pozițiilor avioanelor pentru a afla noi strategii de joc.



4. Bibliografie

<https://www.w3schools.com/cpp/default.asp>

<https://cplusplus.com/forum/beginner/1481/>

<https://stackoverflow.com/questions/4053837/colorizing-text-in-the-console-with-c>

<https://www.asciitable.com>

<https://ascii-generator.site>



5. Cod sursă

```
#include <iostream>
#include <fstream>
#include <conio.h>
#include <cstdlib>
#include <cstring>
#include <string>
#include <stdlib>
#include <ctime>
#include <windows.h>
using namespace std;

const unsigned short no=4,no_start=3,ns=7,nf=2;
char
m[no][15],ng[15][15],g[15][15],fname[25],settings[ns][50],inf
o[nf][25];
int oi,oj,set_opt,opt;
int screen_c, screen_r;
const short window_width = 1280, window_height = 800;
char str[100];
bool fscreen = false;

struct{
    int setsel,setX,setC,set0,setaux,setshrug;
} setare[ns+1];
struct{
    int dir;
    struct{
        int ci,cj;
    } directie[4];
} chenar[4];
void atrib()
{
    strcpy(m[0],"Joc nou");
    strcpy(m[1],"Setari");
    strcpy(m[2],"Ajutor");
    strcpy(m[3],"Inchide");

    ng[0][0]=' ',ng[0][1]='#';
    ng[1][0]=' ',ng[1][1]='1';
    ng[2][0]=' ',ng[2][1]='2';
    ng[3][0]=' ',ng[3][1]='3';
    ng[4][0]=' ',ng[4][1]='4';
    ng[5][0]=' ',ng[5][1]='5';
    ng[6][0]=' ',ng[6][1]='6';
    ng[7][0]=' ',ng[7][1]='7';
    ng[8][0]=' ',ng[8][1]='8';
    ng[9][0]=' ',ng[9][1]='9';
    ng[10][0]='1',ng[10][1]='0';

    ng[0][2]='A',ng[0][3]='B',ng[0][4]='C',ng[0][5]='D',ng[0][6]='
E';

    ng[0][7]='F',ng[0][8]='G',ng[0][9]='H',ng[0][10]='I',ng[0][11]
='J';

    for(int i=1; i<=11; i++)
        for(int j=2; j<=11; j++)
            ng[i][j]='?';

    for(int i=0; i<=11; i++)
        for(int j=0; j<=11; j++)
            g[i][j]=ng[i][j];

    strcpy(settings[0],"Setare 1:");
    strcpy(settings[1],"Setare 2:");
    strcpy(settings[2],"Setare 3:");
    strcpy(settings[3],"Setare 4:");
    strcpy(settings[4],"Salveaza setarea curenta ca si default");
    strcpy(settings[5],"Schimba afisajul");
    strcpy(settings[6],"Inchide");

    setare[0].setsel=112;
    setare[0].setX=11;
    setare[0].setC=12;
    setare[0].set0=2;
    setare[0].setaux=15;
    setare[0].setshrug=15;

    setare[1].setsel=240;
    setare[1].setX=73;
    setare[1].setC=156;
    setare[1].set0=14;
    setare[1].setaux=10;
    setare[1].setshrug=11;

    setare[2].setsel=202;
    setare[2].setX=48;
    setare[2].setC=252;
    setare[2].set0=162;
    setare[2].setaux=2;
    setare[2].setshrug=139;

    strcpy(info[0],"Detalii creator");
    strcpy(info[1],"Iesire");

    /* directie:
    0 - asus
    1 - astg
    2 - ajos
    3 - adrp */
    //////////////////////////////////////////////////
    chenar[0].directie[0].ci = 4;
    chenar[0].directie[0].cj = 4;

    chenar[0].directie[1].ci = 3;
    chenar[0].directie[1].cj = 5;

    chenar[0].directie[2].ci = 1;
    chenar[0].directie[2].cj = 4;

    chenar[0].directie[3].ci = 3;
    chenar[0].directie[3].cj = 2;
    //////////////////////////////////////////////////
    chenar[1].directie[0].ci = 4;
    chenar[1].directie[0].cj = 9;

    chenar[1].directie[1].ci = 3;
    chenar[1].directie[1].cj = 11;

    chenar[1].directie[2].ci = 1;
    chenar[1].directie[2].cj = 9;

    chenar[1].directie[3].ci = 3;
    chenar[1].directie[3].cj = 8;
    //////////////////////////////////////////////////
    chenar[2].directie[0].ci = 10;
```



```

    chenar[2].directie[0].cj = 4;

    chenar[2].directie[1].ci = 8;
    chenar[2].directie[1].cj = 5;

    chenar[2].directie[2].ci = 7;
    chenar[2].directie[2].cj = 4;

    chenar[2].directie[3].ci = 8;
    chenar[2].directie[3].cj = 2;
    //////////////////////////////////////
    chenar[3].directie[0].ci = 10;
    chenar[3].directie[0].cj = 9;

    chenar[3].directie[1].ci = 8;
    chenar[3].directie[1].cj = 11;

    chenar[3].directie[2].ci = 7;
    chenar[3].directie[2].cj = 9;

    chenar[3].directie[3].ci = 8;
    chenar[3].directie[3].cj = 8;
    //////////////////////////////////////
}
void clf()
{
    ofstream file("default.bls");
    file<<"";
    file.close();
}
void ShowConsoleCursor(bool showFlag)
{
    HANDLE out =
    GetStdHandle(STD_OUTPUT_HANDLE);

    CONSOLE_CURSOR_INFO  cursorInfo;

    GetConsoleCursorInfo(out, &cursorInfo);
    cursorInfo.bVisible = showFlag; // set the cursor visibility
    SetConsoleCursorInfo(out, &cursorInfo);
}
void fullscreen(){
    RECT desktop;
    GetWindowRect(GetDesktopWindow(), &desktop);
    int horizontal = desktop.right;
    int vertical = desktop.bottom;

    MoveWindow(GetConsoleWindow(),desktop.left,desktop.top,
    horizontal,vertical,false);

    ShowWindow(GetConsoleWindow(),
    SW_SHOWMAXIMIZED);

    CONSOLE_SCREEN_BUFFER_INFO scrBufferInfo;

    GetConsoleScreenBufferInfo(GetStdHandle(STD_OUTPUT_
    HANDLE), &scrBufferInfo);
    short winWidth = scrBufferInfo.srWindow.Right -
    scrBufferInfo.srWindow.Left + 1;
    short winHeight = scrBufferInfo.srWindow.Bottom -
    scrBufferInfo.srWindow.Top + 1;

    COORD newSize;
    newSize.X = winWidth;
    newSize.Y = winHeight;

    SetConsoleScreenBufferSize(GetStdHandle(STD_OUTPUT_
    HANDLE), newSize);

    CONSOLE_SCREEN_BUFFER_INFO csbi;

    GetConsoleScreenBufferInfo(GetStdHandle(STD_OUTPUT_
    HANDLE), &csbi);
    screen_c = csbi.srWindow.Right - csbi.srWindow.Left + 1;
    screen_r = csbi.srWindow.Bottom - csbi.srWindow.Top +
    1;

    ShowConsoleCursor(false);
}
void tc(int c)
{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HA
    NDLE),c);
}
void gotoxy(int x, int y){
    COORD c;
    c.X = x;
    c.Y = y;

    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HA
    NDLE), c);
}
void afis(int o)
{

```

```

SetConsoleScreenBufferSize(GetStdHandle(STD_OUTPUT_
HANDLE), newSize);

    CONSOLE_SCREEN_BUFFER_INFO csbi;

    GetConsoleScreenBufferInfo(GetStdHandle(STD_OUTPUT_
    HANDLE), &csbi);
    screen_c = csbi.srWindow.Right - csbi.srWindow.Left + 1;
    screen_r = csbi.srWindow.Bottom - csbi.srWindow.Top +
    1;

    ShowConsoleCursor(false);
}
void not_fullscreen(){
    RECT desktop;
    GetWindowRect(GetDesktopWindow(), &desktop);
    int horizontal = desktop.right;
    int vertical = desktop.bottom;

    MoveWindow(GetConsoleWindow(),(horizontal-
    window_width)/2,(vertical-
    window_height)/2,window_width,window_height,false);

    CONSOLE_SCREEN_BUFFER_INFO scrBufferInfo;

    GetConsoleScreenBufferInfo(GetStdHandle(STD_OUTPUT_
    HANDLE), &scrBufferInfo);
    short winWidth = scrBufferInfo.srWindow.Right -
    scrBufferInfo.srWindow.Left + 1;
    short winHeight = scrBufferInfo.srWindow.Bottom -
    scrBufferInfo.srWindow.Top + 1;

    COORD newSize;
    newSize.X = winWidth;
    newSize.Y = winHeight;

    SetConsoleScreenBufferSize(GetStdHandle(STD_OUTPUT_
    HANDLE), newSize);

    CONSOLE_SCREEN_BUFFER_INFO csbi;

    GetConsoleScreenBufferInfo(GetStdHandle(STD_OUTPUT_
    HANDLE), &csbi);
    screen_c = csbi.srWindow.Right - csbi.srWindow.Left + 1;
    screen_r = csbi.srWindow.Bottom - csbi.srWindow.Top +
    1;

    ShowConsoleCursor(false);
}
void tc(int c)
{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HA
    NDLE),c);
}
void gotoxy(int x, int y){
    COORD c;
    c.X = x;
    c.Y = y;

    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HA
    NDLE), c);
}
void afis(int o)
{

```

[illegible]



```

        a=(a==1)?2:1;
        if((int)c1==79) exit(0);
    }
}
void optiune_informatii(int k)
{
    int ok=1,of=0;
    char c1,c2;
    int r,c;

    while(ok)
    {
        r = screen_r/2 - 6;
        c = screen_c/2 - 35;
        tc(15),system("CLS");
        gotoxy(c,r),cout<<"Ati selectat fila de ajutor. Informatii
        utile despre program:";
        gotoxy(c,++r),cout<<char(176)<<" Pentru a te deplasa
        printre optiuni foloseste sagetile tastaturii.";
        gotoxy(c,++r),cout<<char(177)<<" Pentru a alege
        optiunea care este selctata apasa tasta 'Enter'.";
        gotoxy(c,++r),cout<<char(178)<<" Pentru a iesi dintr-un
        meniu apasa tasta 'Esc'.";
        gotoxy(c,++r),cout<<char(219)<<" Pentru a schimba
        culorile din program foloseste meniul 'Setari'.";
        gotoxy(c,++r),cout<<char(178)<<" Pentru a incepe un
        joc nou foloseste meniul 'Joc nou.'.";
        gotoxy(c,++r),cout<<char(177)<<" Un avion descoperit
        v-a aparea plin, unul partial descoperit v-a aparea cu X.";
        gotoxy(c,++r),cout<<char(176)<<" Jocul se termina cand
        gasesti toate avioanele sau cand apesi tasta 'Esc'.";

        gotoxy(screen_c/2-6,r+=2),cout<<char(3),tc(14),cout<<"
        Have fun!",tc(15),cout<<'<<<char(3);
        gotoxy(screen_c/2-16,r+=2);
        if(of==0)
            tc(setare[set_opt].setsel),cout<<info[0],tc(15),cout<<" |
        "<<info[1];
        else
            tc(15),cout<<info[0]<<" |
        ",tc(setare[set_opt].setsel),cout<<info[1],tc(15);
        c1=getch();
        if((int)c1==13 && of==0)
            dezv();
        if((int)c1==13 && of==nf-1 || (int)c1==27)
        {
            tc(15);
            system("CLS");
            ok=0;
        }
        if((int)c1==32)
        {
            c2=getch();
            switch((int)c2)
            {
                case 75:
                    if(of==0)
                        of=1;
                    else
                        of=0;
                    break;
                case 77:
                    if(of==1)
                        of=0;
                    else
                        of=1;
            }
        }
    }
}

```

```

        break;
        case 79:
            exit(0);
    }
}
    if(k==0)
        afis(2);
}
void afis2()
{
    tc(15);
    system("CLS");
}
void afis_njoc(int oi, int oj)
{
    tc(15);
    system("CLS");
    int r=screen_r/2-7,c=screen_c/2-14;
    gotoxy(c,r);
    cout<<char(201);
    for(int i=1; i<=27; i++)
        cout<<char(205);
    cout<<char(187);
    gotoxy(c,++r);
    cout<<char(186)<<" "<<char(186);
    gotoxy(c,++r);

    for(int i=0; i<11; i++)
    {
        cout<<char(186)<<'<<'<<';

        for(int j=0; j<=11; j++)
        {
            if(i==oi && j==oj)
                tc(setare[set_opt].setsel);
            else
            {
                if(g[i][j]=='X' || g[i][j]==char(219))
                    tc(setare[set_opt].setX);
                else if((g[i][j]=='C' || g[i][j]==char(254)) && i!=0)
                    tc(setare[set_opt].setC);
                else if(g[i][j]=='0' && j!=1)
                    tc(setare[set_opt].set0);
                else
                    tc(setare[set_opt].setshrug);
            }
        }
        if(j==0)
        {
            tc(setare[set_opt].setaux);
            cout<<g[i][j];
        }
        else if(j==1)
        {
            tc(setare[set_opt].setaux);
            cout<<g[i][j]<<"<<";
        }
        else if(i==0)
        {
            tc(setare[set_opt].setaux);
            cout<<g[i][j]<<"<<";
        }
        else
        {
            cout<<g[i][j];
            tc(setare[set_opt].setaux);
        }
    }
}

```



```

        cout<<"|";
    }
}

tc(15);
cout<<' '<<' '<<char(186);
gotoxy(c,++r);
}
cout<<char(186)<<"          "<<char(186);
gotoxy(c,++r);
cout<<char(200);
for(int i=1; i<=27; i++)
    cout<<char(205);
cout<<char(188);
}
void desen()
{
    int c=screen_c/2-45,r=screen_r/2-21;

    gotoxy(c,++r);cout<<"
:#+%@@@%#=. ";
    gotoxy(c,++r);cout<<"
:#@@@@@@@@@@@@@#. ";
    gotoxy(c,++r);cout<<"
:#@@@@@@@@@@@@@@@@@@";
    gotoxy(c,++r);cout<<"    --.
:#@@@@@@@@@@@@@@@@@@@@@%";
    gotoxy(c,++r);cout<<"    -%@@@@@%*=.
.*@@@@@@@@@@@@@@@@@@@@@@";
    gotoxy(c,++r);cout<<"    -
%@@@@@@@@@@@@@@@@@@@@@%*+.
.*@@@@@@@@@@@@@@@@@@@@@@@@@@";
    gotoxy(c,++r);cout<<"    -
%@@@@@@@@@@@@@@@@@@@@@@@@@%*+.
+@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@";
    gotoxy(c,++r);cout<<"    -
%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@"%*+.;
.+@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@";
    gotoxy(c,++r);cout<<"
.*@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@#+-;
+@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@%=";
    gotoxy(c,++r);cout<<"
:#+%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@%#+-.
=@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@%=";
    gotoxy(c,++r);cout<<"
:#+#+@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@%#+-
:=%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@%";
    gotoxy(c,++r);cout<<"    -
*@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@#";
    gotoxy(c,++r);cout<<"    -
*@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@#";
    gotoxy(c,++r);cout<<"
:#+%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@#";
    gotoxy(c,++r);cout<<"
.=#+#+@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

```

```

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@*.
";
    gotoxy(c,++r);cout<<"
.=*@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@*.
";
    gotoxy(c,++r);cout<<"
:*@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@*.
";
    gotoxy(c,++r);cout<<"
:+%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@#";
    gotoxy(c,++r);cout<<"
:=#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@-";
    gotoxy(c,++r);cout<<"    .-
#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@";
    gotoxy(c,++r);cout<<"
+@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@#";
    gotoxy(c,++r);cout<<"
=@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@=";
    gotoxy(c,++r);cout<<"
=@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@";
    gotoxy(c,++r);cout<<"
=%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@*";
    gotoxy(c,++r);cout<<"    -
%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@-";
    gotoxy(c,++r);cout<<"
:%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@#-
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.
";
    gotoxy(c,++r);cout<<"
:#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@*
#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@#";
    gotoxy(c,++r);cout<<"
:#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@*.
*@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@-";
    gotoxy(c,++r);cout<<"    ..
.*@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@.
=@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@%";
    gotoxy(c,++r);cout<<"    :#@@@@@@@@%##*+=-
*@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@+.
:@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@+";
    gotoxy(c,++r);cout<<"
:#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@+
.%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@:";
    gotoxy(c,++r);cout<<"
:#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@+
@@@@@@@@@@@@=";
    gotoxy(c,++r);cout<<"
+@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@";
    gotoxy(c,++r);cout<<":#@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@%
=@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@*";
    gotoxy(c,++r);cout<<"
:=#+#+@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@=";
    gotoxy(c,++r);cout<<"
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@:";

```



```

gotoxy(c,r++);cout<<"
.:=#@@@@@@@@@@@@@@@@@@@@@@@@@@@@-
.%@@@@@@@@@@@@@@@@@@@@@@@@@%    ";
gotoxy(c,r++);cout<<"
.:=%@@@@@@@@@@@@@@@@@@@@@@@@@
#@@@@@@@@@@@@@@@@@@@@@@@@@*    ";
gotoxy(c,r++);cout<<"
#@@@@@@@@@@@@@@@@@@@@-
+@@@@@@@@@@@@@@@@@@@@@@@@-    ";
gotoxy(c,r++);cout<<"
.@@@@@@@@@@@@@@@@@@*    -
@@@@@@@@@@@@@@@@@@@@@%    ";
gotoxy(c,r++);cout<<"
@@@@@@@@@@@@@@@@@@@@
:%@@@@@@@@@@@@@@@@@@@@@+    ";
gotoxy(c,r++);cout<<"
*@@@@@@@@@@@@@@@@-
#@@@@@@@@@@@@@@@@@=    ";
gotoxy(c,r++);cout<<"
%@@@@@@@@@@@@@@@@*
+@@@@@@@@@@@@@@@@+    ";
gotoxy(c,r++);cout<<"    :@@@@@@@@@@@@@=
-@@@@@@@@@+    ";
gotoxy(c,r++);cout<<"    +@@@@@@@@@%=
:@@@@@+    ";
gotoxy(c,r++);cout<<"    #@@@=
.#+    ";
gotoxy(c,r++);cout<<"    .%=
";
cout<<char(7);
}
void spatiu(int a)
{
    tc(15);
    cout<<' ';
    tc(a);
}
void save_default()
{
    clf();
    ofstream file("default.bls");
    file<<str<<"\n";
    file<<set_opt;
    if(set_opt == 3)
        file<<"\n"<<setare[3].setsel<<"\n"<<setare[3].setX<<"\n"<<setar
e[3].setC<<"\n"<<setare[3].set0<<"\n"<<setare[3].setaux<<"\n"<
<setare[3].setshrug;
    file.close();
}
void culori()
{
    int da=1;
    char c1;
    afis2();
    int x,y;
    while(da)
    {
        x=screen_c/2-75;
        y=screen_r/2-19;
        gotoxy(x,y);
        tc(15);
        for(int i=1,k=15; i<=255; i++)
        {
            if(i==k+1)
                k=k+15,tc(15),gotoxy(x,++y);
            if(i>=100)

```

```

        cout<<i<<' ';
        else if(i<=99 && i>=10)
            cout<<' '<<i<<' ';
        else
            cout<<' '<<' '<<i<<' ';
        tc(i);
        cout<<"Color";
        tc(15);
        cout<<' ';
    }
    da=0;
}

ShowConsoleCursor(true);
int r=screen_r/2+3,c=screen_c/2-28;
gotoxy(c,r);
cout<<"Introdu id-ul culorii pentru fiecare din elementele
urmatoare:";
gotoxy(c,++r);
gotoxy(c,++r),cout<<char(201)<<char(205)<<" Pentru
elementele selectate: ";
cin>>setare[3].setsel;
gotoxy(c,++r),cout<<char(204)<<char(205)<<" Pentru o
bucata de avion: ";
cin>>setare[3].setX;
gotoxy(c,++r),cout<<char(204)<<char(205)<<" Pentru
capul avionului: ";
cin>>setare[3].setC;
gotoxy(c,++r),cout<<char(204)<<char(205)<<" Pentru o
lovitura ratata: ";
cin>>setare[3].set0;
gotoxy(c,++r),cout<<char(204)<<char(205)<<" Pentru
auxiliare (numere, litere si |): ";
cin>>setare[3].setaux;
gotoxy(c,++r),cout<<char(200)<<char(205)<<" Pentru
pozitiile necunoscute (?): ";
cin>>setare[3].setshrug;
set_opt=3;
ShowConsoleCursor(false);
}
void optiune_setari(int a)
{
    opt = a;
    int ok=1,r,c;
    char c1,c2;
    tc(15);
    system("CLS");
    while(ok)
    {
        c=screen_c/2-17;
        r=screen_r/2-4;
        gotoxy(c,r);
        if((int)c1==13)
        {
            if(!strcmp(settings[opt],"Setare 1:"))
                set_opt=0;
            if(!strcmp(settings[opt],"Setare 2:"))
                set_opt=1;
            if(!strcmp(settings[opt],"Setare 3:"))
                set_opt=2;
            if(!strcmp(settings[opt],"Setare 4:"))
                set_opt=3;
        }
        afis2();
        gotoxy(c,++r);

```



```

for(int i=0; i<ns-4; i++)
{
    if(i!=0 && i!=ns-3)
        cout<<char(204)<<char(205)<<' ';
    else if(i==0)
        cout<<char(201)<<char(205)<<' ';
    else
        cout<<char(200)<<char(205)<<' ';

    if(i==opt)
    {
        tc(setare[set_opt].setsel);
        cout<<settings[i];
    }
    else
        tc(15),cout<<settings[i];
    tc(15),cout<<' ';
    tc(setare[i].setaux);

    cout<<'A',spatiu(setare[i].setaux),cout<<'3',spatiu(setare[i].set
aux),cout<<'|';
    tc(15),cout<<' ';
    tc(setare[i].set0);
    cout<<'0';
    tc(15),cout<<' ';
    tc(setare[i].setshrug);
    cout<<'?';
    tc(15),cout<<' ';
    tc(setare[i].setX);
    cout<<'X';
    tc(15),cout<<' ';
    tc(setare[i].setC);
    cout<<'C';
    tc(15),cout<<' ';
    tc(setare[i].setsel);
    cout<<"Selectat";
    tc(15),cout<<' ';
    gotoxy(c,++r);
}
if(opt==3)
    cout<<char(200)<<char(205)<<'
',tc(setare[set_opt].setsel),cout<<settings[3];
else
    cout<<char(200)<<char(205)<<' '<<settings[3];
tc(15);
cout<<" Selecteaza aceasta optiune pentru a alege culori
personalizate!";
r+=2;
gotoxy(screen_c/2-19,r);
if(opt==4)
    cout<<' '<<char(176)<<'
',tc(setare[set_opt].setsel),cout<<settings[4],tc(15),cout<<'
'<<char(176);
else
    cout<<' '<<char(176)<<' '<<settings[4]<<'
'<<char(176);
tc(15);
r+=2;
gotoxy(screen_c/2-8,r);
if(opt==5)
    cout<<'
',tc(setare[set_opt].setsel),cout<<settings[5],tc(15);
else
    cout<<' ',tc(15),cout<<settings[5];
tc(15);
r+=2;

```

```

gotoxy(screen_c/2-5,r);
if(opt==6)
    cout<<' ',tc(12),cout<<'X',tc(15),cout<<'
',tc(setare[set_opt].setsel),cout<<settings[6],tc(15),cout<<'
',tc(12),cout<<'X';
else
    cout<<' ',tc(12),cout<<'X',tc(15),cout<<'
'<<settings[6],tc(15),cout<<' ',tc(12),cout<<'X';
tc(15);
cout<<' ';
c1=getch();
if((int)c1=='a' || (int)c1=='h' || (int)c1=='A' || (int)c1=='H')
    optiune_informatii(1);
if((int)c1==27)
{
    system("CLS");
    ok=0;
}
if((int)c1==13)
{
    if(opt==ns-1)
    {
        system("CLS");
        ok=0;
    }
    if(opt==3)
        culori();
    if(opt==4)
        save_default();
    if(opt==5){
        if(fscreen == false) fscreen=true,fullscreen();
        else fscreen=false,not_fullscreen();
    }
}
else if((int)c1==32)
{
    c2=getch();
    switch((int)c2)
    {
        case 72:
            if(opt==0)
                opt=ns-1;
            else
                opt--;
            break;
        case 80:
            if(opt==ns-1)
                opt=0;
            else
                opt++;
            break;
        case 79:
            exit(0);
    }
}
}
afis(1);
}
void astg(int i,int j)
{
    ng[i][j]='C';
    ng[i][j-1]='X',ng[i+1][j-1]='X',ng[i+2][j-1]='X',ng[i-1][j-
1]='X',ng[i-2][j-1]='X';
    ng[i][j-2]='X';
    ng[i][j-3]='X',ng[i+1][j-3]='X',ng[i-1][j-3]='X';
}

```




```

void adrp(int i,int j)
{
    ng[i][j]='C';
    ng[i][j+1]='X',ng[i+1][j+1]='X',ng[i+2][j+1]='X',ng[i-1][j+1]='X',ng[i-2][j+1]='X';
    ng[i][j+2]='X';
    ng[i][j+3]='X',ng[i+1][j+3]='X',ng[i-1][j+3]='X';
}
void asus(int i,int j)
{
    ng[i][j]='C';
    ng[i-1][j]='X',ng[i-1][j-1]='X',ng[i-1][j-2]='X',ng[i-1][j+1]='X',ng[i-1][j+2]='X';
    ng[i-2][j]='X';
    ng[i-3][j]='X',ng[i-3][j-1]='X',ng[i-3][j+1]='X';
}
void ajos(int i,int j)
{
    ng[i][j]='C';
    ng[i+1][j]='X',ng[i+1][j+1]='X',ng[i+1][j+2]='X',ng[i+1][j-1]='X',ng[i+1][j-2]='X';
    ng[i+2][j]='X';
    ng[i+3][j]='X',ng[i+3][j+1]='X',ng[i+3][j-1]='X';
}
bool vstg(int i, int j)
{
    if(ng[i][j]=='?' && ng[i][j-1]=='?' && ng[i+1][j-1]=='?' && ng[i+2][j-1]=='?' && ng[i-1][j-1]=='?' && ng[i-2][j-1]=='?' && ng[i][j-2]=='?' && ng[i][j-3]=='?' && ng[i-1][j-3]=='?' && ng[i+1][j-3]=='?')
        return true;
    else
        return false;
}
bool vdrp(int i, int j)
{
    if(ng[i][j]=='?' && ng[i][j+1]=='?' && ng[i+1][j+1]=='?' && ng[i+2][j+1]=='?' && ng[i-1][j+1]=='?' && ng[i-2][j+1]=='?' && ng[i][j+2]=='?' && ng[i][j+3]=='?' && ng[i-1][j+3]=='?' && ng[i+1][j+3]=='?')
        return true;
    else
        return false;
}
bool vsus(int i, int j)
{
    if(ng[i][j]=='?' && ng[i-1][j]=='?' && ng[i-1][j+1]=='?' && ng[i-1][j+2]=='?' && ng[i-1][j-1]=='?' && ng[i-1][j-2]=='?' && ng[i-2][j]=='?' && ng[i-3][j]=='?' && ng[i-3][j+1]=='?' && ng[i-3][j-1]=='?')
        return true;
    else
        return false;
}
bool vjos(int i, int j)
{
    if(ng[i][j]=='?' && ng[i+1][j]=='?' && ng[i+1][j+1]=='?' && ng[i+1][j+2]=='?' && ng[i+1][j-1]=='?' && ng[i+1][j-2]=='?' && ng[i+2][j]=='?' && ng[i+3][j]=='?' && ng[i+3][j+1]=='?' && ng[i+3][j-1]=='?')
        return true;
    else
        return false;
}
bool stg(int i, int j)
{

```

```

    if(ng[i][j]=='C' && ng[i][j-1]=='X' && ng[i+1][j-1]=='X' && ng[i+2][j-1]=='X' && ng[i-1][j-1]=='X' && ng[i-2][j-1]=='X' && ng[i][j-2]=='X' && ng[i][j-3]=='X' && ng[i-1][j-3]=='X')
        return true;
    else
        return false;
}
bool drp(int i, int j)
{
    if(ng[i][j]=='C' && ng[i][j+1]=='X' && ng[i+1][j+1]=='X' && ng[i+2][j+1]=='X' && ng[i-1][j+1]=='X' && ng[i-2][j+1]=='X' && ng[i][j+2]=='X' && ng[i][j+3]=='X' && ng[i-1][j+3]=='X' && ng[i+1][j+3]=='X')
        return true;
    else
        return false;
}
bool sus(int i, int j)
{
    if(ng[i][j]=='C' && ng[i-1][j]=='X' && ng[i-1][j+1]=='X' && ng[i-1][j+2]=='X' && ng[i-1][j-1]=='X' && ng[i-1][j-2]=='X' && ng[i-2][j]=='X' && ng[i-3][j]=='X' && ng[i-3][j+1]=='X' && ng[i-3][j-1]=='X')
        return true;
    else
        return false;
}
bool jos(int i, int j)
{
    if(ng[i][j]=='C' && ng[i+1][j]=='X' && ng[i+1][j+1]=='X' && ng[i+1][j+2]=='X' && ng[i+1][j-1]=='X' && ng[i+1][j-2]=='X' && ng[i+2][j]=='X' && ng[i+3][j]=='X' && ng[i+3][j+1]=='X' && ng[i+3][j-1]=='X')
        return true;
    else
        return false;
}
void avion(int p)
{
    int ok=0,d;
    int i,j;
    if(p==4 || p==3){
        srand((unsigned) time(NULL));
        for(int i=0; i<=4; i++) chenar[i].dir = (rand() % 4);
    }
    if(p==4){
        for(int k = 0; k<=3; k++){
            switch(chenar[k].dir){
                case 0:
                    asus(chenar[k].directie[0].ci,chenar[k].directie[0].cj);
                    break;
                case 1:
                    astg(chenar[k].directie[1].ci,chenar[k].directie[1].cj);
                    break;
                case 2:
                    ajos(chenar[k].directie[2].ci,chenar[k].directie[2].cj);
                    break;
                case 3:
                    adrp(chenar[k].directie[3].ci,chenar[k].directie[3].cj);
                    break;
            }
        }
    }
}

```



```

    }
}
else{
    if(p==3){
        srand((unsigned) time(NULL));
        int k = (rand() % 4);
        switch(chenar[k].dir){
            case 0:
                asus(chenar[k].directie[0].ci,chenar[k].directie[0].cj);
                break;
            case 1:
                astg(chenar[k].directie[1].ci,chenar[k].directie[1].cj);
                break;
            case 2:
                ajos(chenar[k].directie[2].ci,chenar[k].directie[2].cj);
                break;
            case 3:
                adrp(chenar[k].directie[3].ci,chenar[k].directie[3].cj);
                break;
        }
        p--;
    }
    while(ok<p){
        srand((unsigned)time(0));
        i=rand()%10+1;
        srand((unsigned)time(0));
        j=rand()%10+2;
        srand((unsigned)time(0));
        d=rand()%4;
        if(d==0)
            if(i+3<=10 && j-2>=2 && j+2<=11)
                if(vjos(i,j)==true)
                    ajos(i,j),ok++;
        if(d==1)
            if(i-3>=1 && j-2>=2 && j+2<=11)
                if(vsus(i,j)==true)
                    asus(i,j),ok++;
        if(d==2)
            if(j+3<=11 && i+2<=10 && i-2>=1)
                if(vdrp(i,j)==true)
                    adrp(i,j),ok++;
        if(d==3)
            if(j-3>=2 && i+2<=10 && i-2>=1)
                if(vstg(i,j)==true)
                    astg(i,j),ok++;
    }
}
}
void easter_egg(int q){
    tc(15);
    system("CLS");
    if(q==1)
        gotoxy(screen_c/2-5,screen_r/2),cout<<"Nice try
",tc(12),cout<<char(3),tc(15),cout<<' ';
    else
        gotoxy(screen_c/2-26,screen_r/2),cout<<"Introdu un
numar valid de avioane (minim 1, maxim 4)";
}
void game()
{
    int l=0,a;
    tc(15);

```

```

    system("CLS");
    atrib();
    gotoxy(screen_c/2-14,screen_r/2-1);
    ShowConsoleCursor(true);
    cout<<"Cu cate avioane vrei sa joci?">gotoxy(screen_c/2-
8,screen_r/2),cout<<"Numar avioane: ";
    cin>>a;
    ShowConsoleCursor(false);
    char c1,c2;
    int oi=1,oj=2,k=1,x=0,cnt=0;
    if(a>=1 && a<=4){
        avion(a);
        do
        {
            k=1;
            afis_njoc(oi,oj);
            if((int)c1==13 && x!=a)
            {
                if(ng[oi][oj]=='?' && g[oi][oj]!='0')
                    g[oi][oj]='0',cnt++;
                if(ng[oi][oj]=='X' && g[oi][oj]!='X' &&
g[oi][oj]!=char(219))
                    g[oi][oj]='X',cnt++;
                if(ng[oi][oj]=='C' && g[oi][oj]!='C' &&
g[oi][oj]!=char(254))
                {
                    g[oi][oj]=254;
                    if(drp(oi,oj))
                    {
                        g[oi][oj+1]=219,g[oi+1][oj+1]=219,g[oi+2][oj+1]=219,g[oi-
1][oj+1]=219,g[oi-2][oj+1]=219;
                        g[oi][oj+2]=219;
                        g[oi][oj+3]=219,g[oi+1][oj+3]=219,g[oi-
1][oj+3]=219;
                    }
                    else if(stg(oi,oj))
                    {
                        g[oi][oj-1]=219,g[oi+1][oj-1]=219,g[oi+2][oj-
1]=219,g[oi-1][oj-1]=219,g[oi-2][oj-1]=219;
                        g[oi][oj-2]=219;
                        g[oi][oj-3]=219,g[oi+1][oj-3]=219,g[oi-1][oj-
3]=219;
                    }
                    else if(sus(oi,oj))
                    {
                        g[oi-1][oj]=219,g[oi-1][oj-1]=219,g[oi-1][oj-
2]=219,g[oi-1][oj+1]=219,g[oi-1][oj+2]=219;
                        g[oi-2][oj]=219;
                        g[oi-3][oj]=219,g[oi-3][oj-1]=219,g[oi-
3][oj+1]=219;
                    }
                    else if(jos(oi,oj))
                    {
                        g[oi+1][oj]=219,g[oi+1][oj+1]=219,g[oi+1][oj+2]=219,g[oi+1
][oj-1]=219,g[oi+1][oj-2]=219;
                        g[oi+2][oj]=219;
                        g[oi+3][oj]=219,g[oi+3][oj+1]=219,g[oi+3][oj-
1]=219;
                    }
                    x++;
                    cnt++;
                }
            }
            afis_njoc(oi,oj);

```



```

if(x==a)
{
    gotoxy(screen_c/2-22,screen_r/2+10);
    cout<<"Felicitari! Ati castigat! Ati ghicit din ";
    tc(23);
    cout<<cnt;
    tc(15);
    cout<<" incercari!";
    gotoxy(screen_c/2-13,screen_r/2+11);
    cout<<"(pentru a iesi apasa 'Esc')";
}
////////////////////////////////////
// afis solutie
/*gotoxy(0,0);
for(int i=0; i<11; i++)
{
    for(int j=0; j<=11; j++)
        cout<<ng[i][j];
    cout<<endl;
}*/
////////////////////////////////////
c1=getch();
if((int)c1=='a' || (int)c1=='h' || (int)c1=='A' ||
(int)c1=='H')
    optiune_informatii(1);
if((int)c1==27)
{
    system("CLS");
    int o=0;
    afis(o);
    k=0;
}
else if((int)c1==32)
{
    c2=getch();
    switch((int)c2)
    {
        case 75:
            if(oj==2)
                oj=11;
            else
                oj--;
            break;
        case 77:
            if(oj==11)
                oj=2;
            else
                oj++;
            break;
        case 72:
            if(oi==1)
                oi=10;
            else
                oi--;
            break;
        case 80:
            if(oi==10)
                oi=1;
            else
                oi++;
            break;
        case 79:
            exit(0);
    }
}

```

```

}
while(k);
}
else{
    if(a==INT_MAX)
        easter_egg(1);
    else
        easter_egg(0);
    Sleep(4000);
    afis(0);
}
}
bool inchidere()
{
    int a=1;
    while(true)
    {
        char c;
        tc(15);
        system("CLS");
        gotoxy(screen_c/2-10,screen_r/2-1);
        cout<<"Sigur vrei sa iesi?";
        gotoxy(screen_c/2-3,screen_r/2);
        if(a==1)
        {
            tc(setare[set_opt].setsel);
            cout<<"NU";
            tc(15);
            cout<<" | DA";
        }
        else if(a==2)
        {
            tc(15);
            cout<<"NU | ";
            tc(setare[set_opt].setsel);
            cout<<"DA";
        }
        tc(15),cout<<' ';
        c=getch();
        if((int)c=='a' || (int)c=='h' || (int)c=='A' || (int)c=='H')
            optiune_informatii(1);
        if((int)c==27)
            return true;
        if((int)c==13)
        {
            if(a==1)
                return false;
            else
                return true;
        }
        else if((int)c!='a' && (int)c!='h' && (int)c!='A' &&
(int)c!='H')
        {
            char c2=getch();
            if((int)c2 == 75 || (int)c2 == 77){
                if(a==1)
                    a=2;
                else
                    a=1;
            }
            if((int)c2==79) exit(0);
        }
    }
}
void joc()
{

```



```

int o=0,j=1;
char c1,c2;
do
{
    afis(o);
    if((int)c1==13)
    {
        if(!strcmp(m[o],"Joc nou"))
            game();
        if(!strcmp(m[o],"Ajutor"))
            optiune_informatii(0);
        if(!strcmp(m[o],"Setari"))
            optiune_setari(0);
    }
    c1=getch();
    if((int)c1=='a' || (int)c1=='h' || (int)c1=='A' || (int)c1=='H')
        optiune_informatii(1);
    if((int)c1==27)
        if(inchidere())
        {
            tc(15);
            system("CLS");
            desen();
            Sleep(1800);
            j=0;
        }
    if((int)c1==13)
    {
        if(o==no-1)
        {
            system("CLS");
            if(inchidere())
            {
                tc(15);
                system("CLS");
                desen();
                Sleep(1800);
                j=0;
            }
        }
    }
    else if((int)c1==32)
    {
        c2=getch();
        switch((int)c2)
        {
            case 72:
                if(o==0)
                    o=no-1;
                else
                    o--;
                break;
            case 80:
                if(o==no-1)
                    o=0;
                else
                    o++;
                break;
            case 79:
                exit(0);
        }
    }
}
while(j);

```

```

}
void inchideree() {
    tc(15);
    system("CLS");
    desen();
    Sleep(1800);
    exit(0);
}
void afiss(int o){
    int c = screen_c/2, r = screen_r/2-4;
    char o1[11]="Joc normal",o2[15]="Joc
fullscreen",o3[8]="Inchide";

    gotoxy(c-4,r);
    tc(14),cout<<"[Avioane]";
    r+=2;

    gotoxy(c-5,r);
    tc(15);
    if(o == 0) tc(setare[set_opt].setsel),cout<<o1;
    else cout<<o1;

    r+=2;
    gotoxy(c-7,r);
    tc(15);
    if(o == 1) tc(setare[set_opt].setsel),cout<<o2;
    else cout<<o2;

    r+=2;
    gotoxy(c-4,r);
    tc(15);
    if(o == 2) tc(setare[set_opt].setsel),cout<<o3;
    else cout<<o3;

    tc(15),cout<<' ';
}
int start(){
    char c1,c2;
    int o=0;
    while(1){
        afiss(o);
        c1 = getch();
        if((int)c1=='a' || (int)c1=='h' || (int)c1=='A' || (int)c1=='H')
            optiune_informatii(1);
        if((int)c1 == 27) inchideree();
        else if((int)c1 == 13){
            if(o == 0){
                fscreen = false;
                return 0;
            }
            else if(o == 1){
                fscreen = true;
                return 0;
            }
            else inchideree();
        }
        if((int)c1 == -32){
            c2 = getch();
            switch((int)c2){
                case 72:
                    if(o == 0) o = no_start-1;
                    else o--;
                    break;
                case 80:
                    if(o == no_start-1) o = 0;
                    else o++;

```



```
        break;
    case 79:
        exit(0);
    }
}

}

int main()
{
    ifstream f("default.bls");
    f.get(str,100);
    f>>set_opt;
    if(set_opt == 3)
f>>setare[3].setsetl>>setare[3].setX>>setare[3].setC>>setare[3
].set0>>setare[3].setaux>>setare[3].setshrug;
    f.close();

    atrib();

    not_fullscreen();
    start();

    if(fscreen == true) fullscreen();

    joc();
}
```