



PROIECT

PENTRU OBTINEREA ATESTĂRII COMPETENȚELOR PROFESIONALE

FILIERA TEORETICĂ / PROFIL REAL
MATEMATICĂ-INFORMATICĂ, intensiv informatică

TEMA:
APLICAȚII ALE TABLOURILOR BIDIMENSIONALE
JOCURI DIVERSE

ÎNDRUMĂTOR:
Prof. Alexandri Maria-Lia

ABSOLVENT:
VRÎNCEANU ALIN-VLĂDUȚ

Cuprins

1. INTRODUCERE.....	3
2. CONSIDERAȚII TEORETICE	4
2.1.Introducere în C++	4
2.2 Istoricul C++	4
2.3 Despre Codeblocks	5
3. CERINȚELE ȘI SPECIFICAȚIILE APLICAȚIEI.....	7
Minime.....	7
Recomandate.....	7
4. MINI-GAME <i>Jailbreaker</i>	8
Primul nivel.....	9
Al doilea nivel	11
Al treilea nivel	12
Al patrulea nivel	13
Game Over.....	14
Finalul jocului	15
5. CONCLUZII	16
BIBLIOGRAFIE.....	17
ANEXE.....	18

1. INTRODUCERE

Aplicații ale tablourilor bidimensionale – jocuri diverse : avioane, sudoku, labirint, cinci (patru) în linie, perechi, tetris s.a. Jocul pe care l-am ales reprezintă o aplicație grafică a matricilor și se numește ***Jailbreaker***

Am utilizat modulul grafic și limbajul de programare C++ în crearea acestui mini-game bazat pe ieșirea dintr-un labirint.

Încă de când am văzut prima dată matricile la ora de informatică m-am „îndrăgostit” de ele, fapt ce m-a determinat să aleg această temă care cuprinde o vastă galerie de idei ce pot fi puse în practică. Punând laolaltă cunoștințele mele despre modulul grafic și matrici, am reușit să creez un joc ce poate reprezenta cu siguranță tema aleasă.

5	31	17	43	29
39	15	41	27	3
13	49	25	1	37
47	23	9	35	11
21	7	33	19	45

Matrice pătratică cu 5 elemente

3	1	2	-3
-1	0	1	-2
0	0	4	5
0	0	3	0

Matrice pătratică cu 4 elemente

Matricile prezentate mai sus sunt MATRICI PĂTRATICE (numărul de linii este egal cu numărul de coloane), iar în C++ are anumite proprietăți ajutătoare.

Matricea se declară folosind comanda: tip [număr max. de linii] [număr max. de coloane];. Exemplu: `int x[10][5]`. Am declarat o matrice de tip `int` cu maxim 10 linii și 5 coloane. Ca la vectori, matricea declarată este goală. Trebuie să îi atribuim elemente fiecărui spațiu liber. Atribuirea se face aproximativ ca la vectori, doar că aici vom avea două repetiții, deoarece fiecărei linii îi corespunde n-1 coloane, etc, și vom avea nevoie de o variabilă *i*, ce va reprezenta linia, și una *j*, ce va reprezenta coloana. Vom mai folosi încă 2 variabile, *n* și *m*, care vor reprezenta numărul actual de linii, respectiv coloane

2. CONSIDERAȚII TEORETICE

2.1.Introducere în C++

C++ (pronunțat ca în engleză /'si: plʌs plʌs/) este un limbaj de programare general, compilat. Este un limbaj multi-paradigmă, cu verificarea statică a tipului variabilelor ce suportă programare procedurală, abstractizare a datelor, programare orientată pe obiecte. În anii 1990, C++ a devenit unul dintre cele mai populare limbaje de programare comerciale, rămânând astfel până azi.

Bjarne Stroustrup de la Bell Labs a dezvoltat C++ (inițial denumit C cu clase) în anii 1980, ca o serie de îmbunătățiri ale limbajului C. Acestea au început cu adăugarea noțiunii de clase, apoi de funcții virtuale, suprascrierea operatorilor, moștenire multiplă, șabloane(engleză template) și excepții. Limbajul de programare C++ a fost standardizat în 1998 ca și ISO 14882:1998, versiunea curentă fiind din 2003, ISO 14882:2003. Următoarea versiune standard, cunoscută informal ca C++0x, este în lucru.

2.2 Istoricul C++

Stroustrup a început să lucreze la C cu clase în 1979. Ideea creării unui nou limbaj a venit din experiența de programare pentru pregătirea tezei sale de doctorat. Stroustrup a descoperit că Simula avea facilități foarte utile pentru proiecte mari, însă era prea lent, în timp ce BCPL era rapid, însă nu era de nivel înalt și era nepotrivit pentru proiecte mari. Când a început să lucreze pentru Bell Labs, avea sarcina de a analiza nucleul UNIX referitor la calcul distribuit. Amintindu-și de experiența sa din perioada lucrării de doctorat, Stroustrup a început să îmbunătățească C cu facilități asemănătoare Simula. C a fost ales deoarece era rapid și portabil. La început facilitățile adăugate C-ului au fost clase, clase derivate, verificare a tipului, inline și argumente cu valori implicite.

În timp ce Stroustrup a proiectat C cu clase (mai apoi C++), a scris de asemenea și Cfront, un compilator care genera cod sursă C din cod C cu clase. Prima lansare comercială a fost în 1985.

În 1982, numele limbajului a fost schimbat de la C cu clase la C++. Au fost adăugate noi facilități, inclusiv funcții virtuale, supraîncărcarea operatorilor și a

funcțiilor, referințe, constante, alocare dinamică, un control al tipului mai puternic și noua variantă de comentariu pe un singur rând (liniile care încep cu caracterele '//').

În 1985 a fost lansată prima ediție a cărții "The C++ Programming Language" (Limbajul de programare C++), oferind informații importante despre limbaj, care încă nu era un standard oficial. În 1989 a fost lansată versiunea 2.0 a C++. Au apărut acum moștenirea multiplă, clase abstracte, funcții statice, funcții constante și membri protected. În 1990 o altă carte a fost lansată, oferind suport pentru standarde viitoare. Ultimele adăugări includeau template-uri, excepții, spații de nume (namespace-uri) și tipul boolean.

O dată cu evoluția limbajului C++, a evoluat și o bibliotecă standard. Prima adăugire a fost biblioteca de intrări/ieșiri (I/O stream), care oferea facilități pentru a înlocui funcțiile tradiționale C cum ar fi printf și scanf. Mai târziu, printre cele mai semnificative adăugări la biblioteca standard a fost STL (Standard Template Library) (Biblioteca de formate standard).

După ani de lucru, un comitet ANSI-ISO a standardizat C++ în 1998 (ISO/IEC 14882:1998).

2.3 Despre Codeblocks

Code::Blocks este un IDE pentru limbajele de programare C++, C și Fortran ce a fost lansat în versiune stabilă în 2008.

Acesta este gratuit, open-source și cross-platform (compatibil cu Windows, Mac și Linux), având proprietatea de a suporta folosirea de multiple compilatoare, inclusiv pe cel specific IDE-ului Microsoft Visual Studio C++, ceea ce face ca dezvoltatorul să aibă un control mai mare asupra software-ului creat și asupra portabilității codului sursă de pe alte IDE-uri.

Programul furnizează toate elementele necesare unui IDE performant:

- **Editor de cod** – permite evidențierea codului în funcție de sintaxă, sugerarea de cod la tastare, posibilitatea de reformatare textuală a codului, personalizarea interfeței grafice, a fontului și a mărimi de font, organizarea facilă a fișierelor deschise – sub formă de file ș.a.

- **Depanator (debugger)** – facilitează depistarea rapidă a erorilor de programare, prin inserția de *breakpoints*, vizualizarea funcțiilor locale și a argumentelor acestora, vizualizarea regiștrilor CPU, definirea de *watches* ș.a.
- **Designer GUI** – permite proiectarea interfețelor grafice într-un mod vizual, de tipul WYSIWYG (What You See Is What You Get). Designerul se numește *wxSmith* și este derivat din librăria *wxWidgets*, librărie ce permite crearea de interfețe grafice cross-platform.
- **Posibilitatea migrării proiectelor de pe alte IDE-uri**, de exemplu de pe DEV-C++ și Visual Studio.
- **Posibilitatea extinderii funcționalității** prin intermediul plug-ins-urilor; în plus, plug-ins-urile pot fi create direct din cadrul programului.
- **Existența unei comunități numeroase de dezvoltatori** care conlucrează la îmbunătățirea programului sau/și discută dificultățile întâlnite în cadrul acestuia. Mai mult decât atât, IDE-ul este open-source, deci oricine poate crea versiuni noi, particularizate după nevoile proprii.
- **Documentație completă** cu privire la folosirea și configurarea IDE-ului (în limba engleză).

3. CERINȚELE ȘI SPECIFICAȚIILE APLICAȚIEI

Cerințele de sistem ale aplicației sunt orientative și nu reprezintă un criteriu pentru care această poate funcționa sau nu pe dispozitivul dumneavoastră.

Minime

- **Sistem de Operare:** Windows 7/8/10.
- **Procesor (CPU):** cu două sau mai multe nuclee fizice.
- **Memorie (RAM):** 1 GB.
- **Placă Video:** NVIDIA GeForce 8600, ATI Radeon HD 4550.
- **Hard Disk:** cu 7200RPM și ~ 50MB spațiu liber.

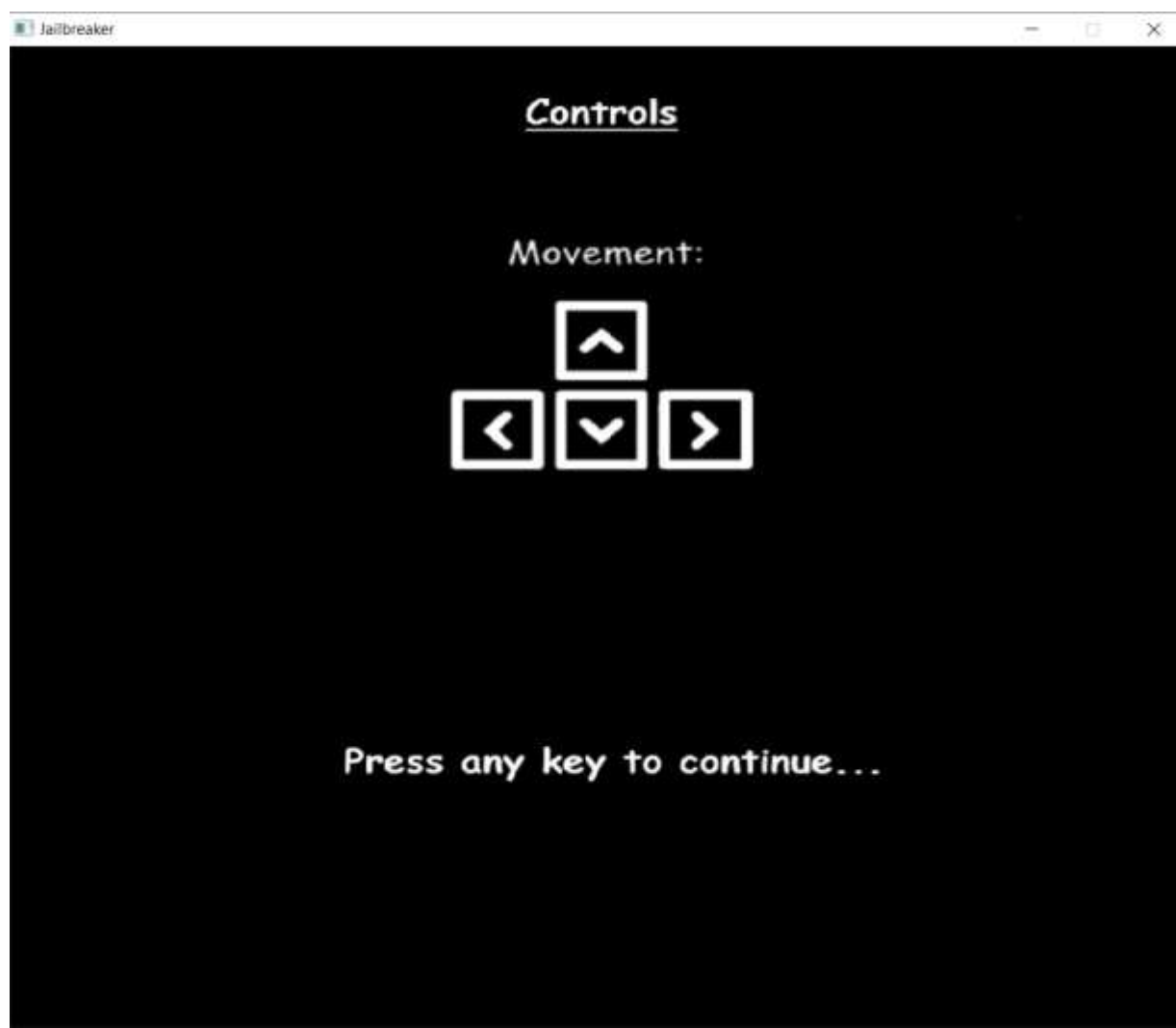
Recomandate

- **Sistem de Operare:** Windows 7/8/10 - 64 bit.
- **Procesor (CPU):** Intel Core i5.
- **Memorie (RAM):** 4 GB (or more).
- **Placă Video:** GeForce GTX660 (2 GB) / Radeon HD 7850 (2 GB).
- **Solid State Drive:** ~ 50MB spațiu liber.

Este necesar ca modulul grafic (graphics.h) să fie instalat în IDE-ul de pe dispozitivul dumneavoastră.

4. MINI-GAME *Jailbreaker*

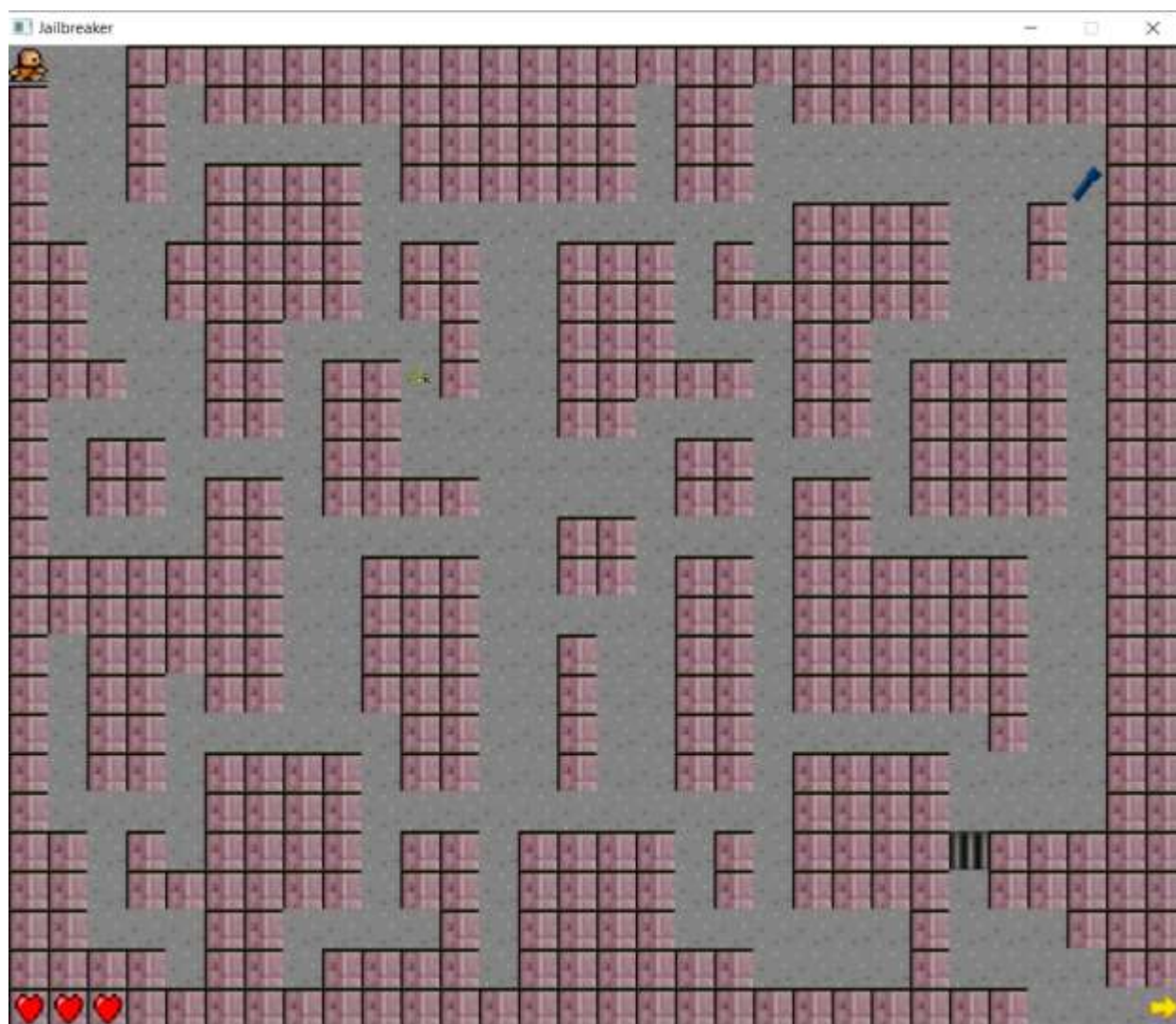
Scopul principal al aplicației este de a conduce propriul caracter la ieșirea din labirint (care se termină la nivelul 4) fără a-și pierde toate cele 3 vieți. Fiecare nivel prezintă noi provocări și pentru a avansa, caracterul trebuie să ajungă la săgeata galbenă din colțul din dreapta-jos al fiecărei hărți. Nivelurile ascund câte o cheie care îi permit caracterului să avanseze mai departe în labirint. Jocul se deschide cu o imagine ce prezintă controalele.



Prezentarea controalelor

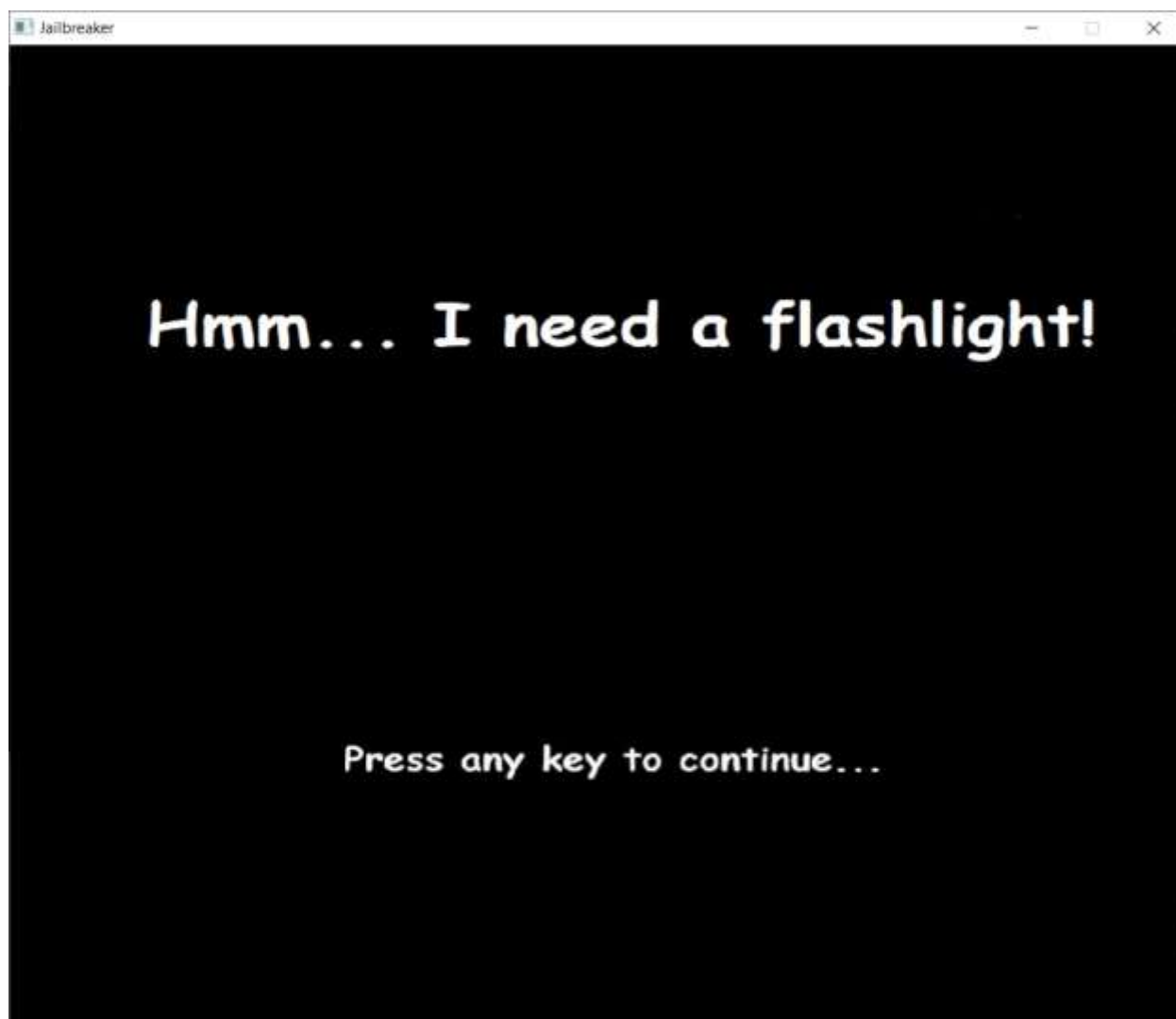
Primul nivel

Caracterul trebuie să ia cheia și lanterna pentru a putea deschide poarta și a avansa în nivel. Lanterna va fi necesară la proba următoare.



Harta primului nivel

În cazul în care caracterul nu ia lanternă, va fi întâmpinat de un mesaj, întrucât camera nivelului 2 nu este luminată. Jucătorul va fi automat întors la nivelul precedent.

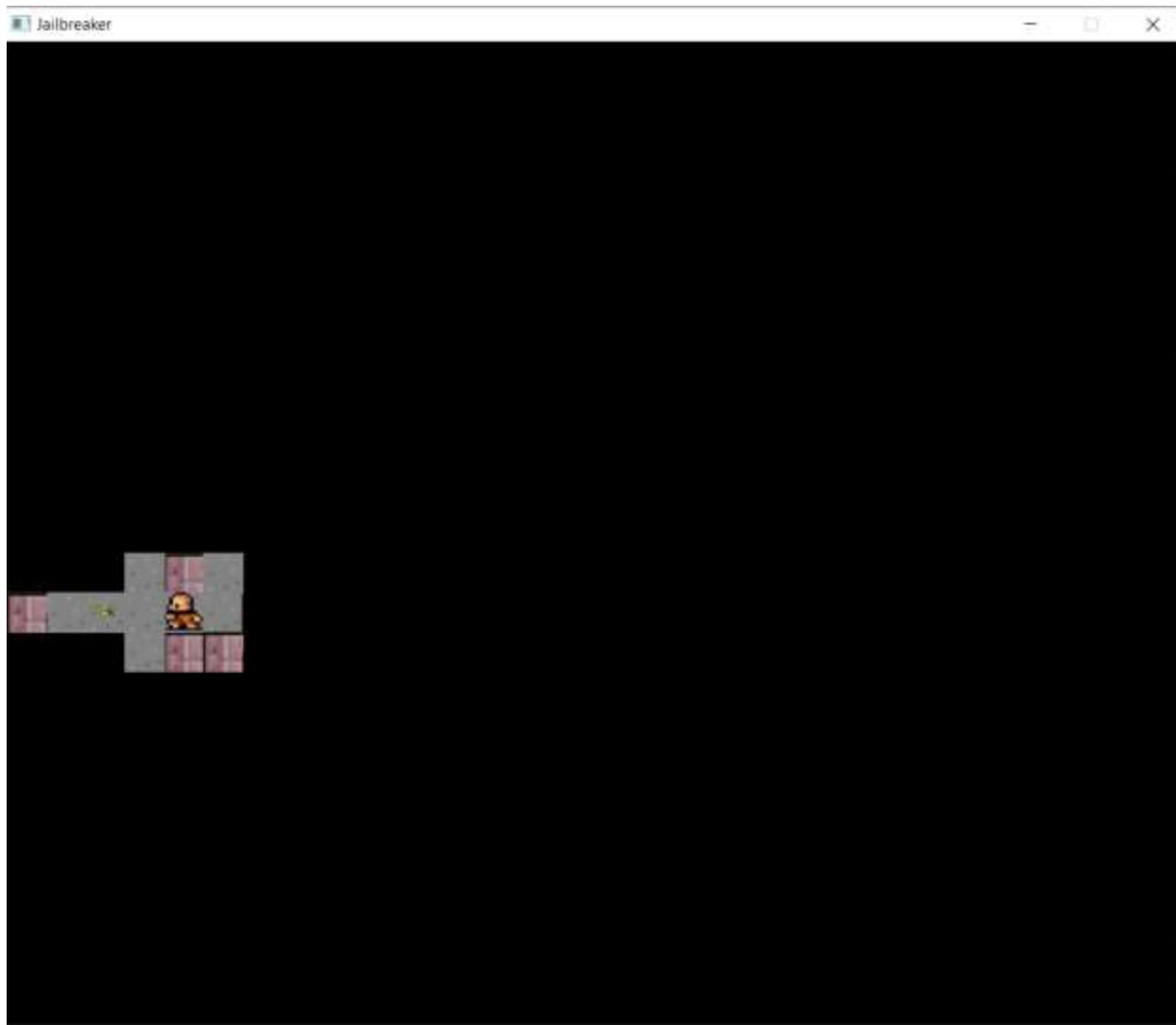


Mesajul care apare în cazul în care lanterna este omisă

Al doilea nivel

Nivelul 2 prezintă un grad mai mare de dificultate, întrucât harta nu este luminată. Lanterna aprinde doar pătrățelele ce înconjoară caracterul și încă 6 în direcția în care acesta se îndreaptă.

Atenție! Dacă în față jucătorului este un perete, lumina poate fi mai mică de 6 pătrățele.



Harta nivelului 2

Al treilea nivel

Pe parcurs, jocul devine din ce în ce mai greu. Astfel, cel de-al treilea nivel implică 5 polițiști care vor îngreuna drumul jucătorului până la cheie, respectiv, până la poartă. Odată ce polițiștii te-au văzut pe aceeași linie/coloană cu ei, aceștia te vor urmări. Dacă jucătorul este prins, acesta va pierde o viață, iar nivelul se va reseta.

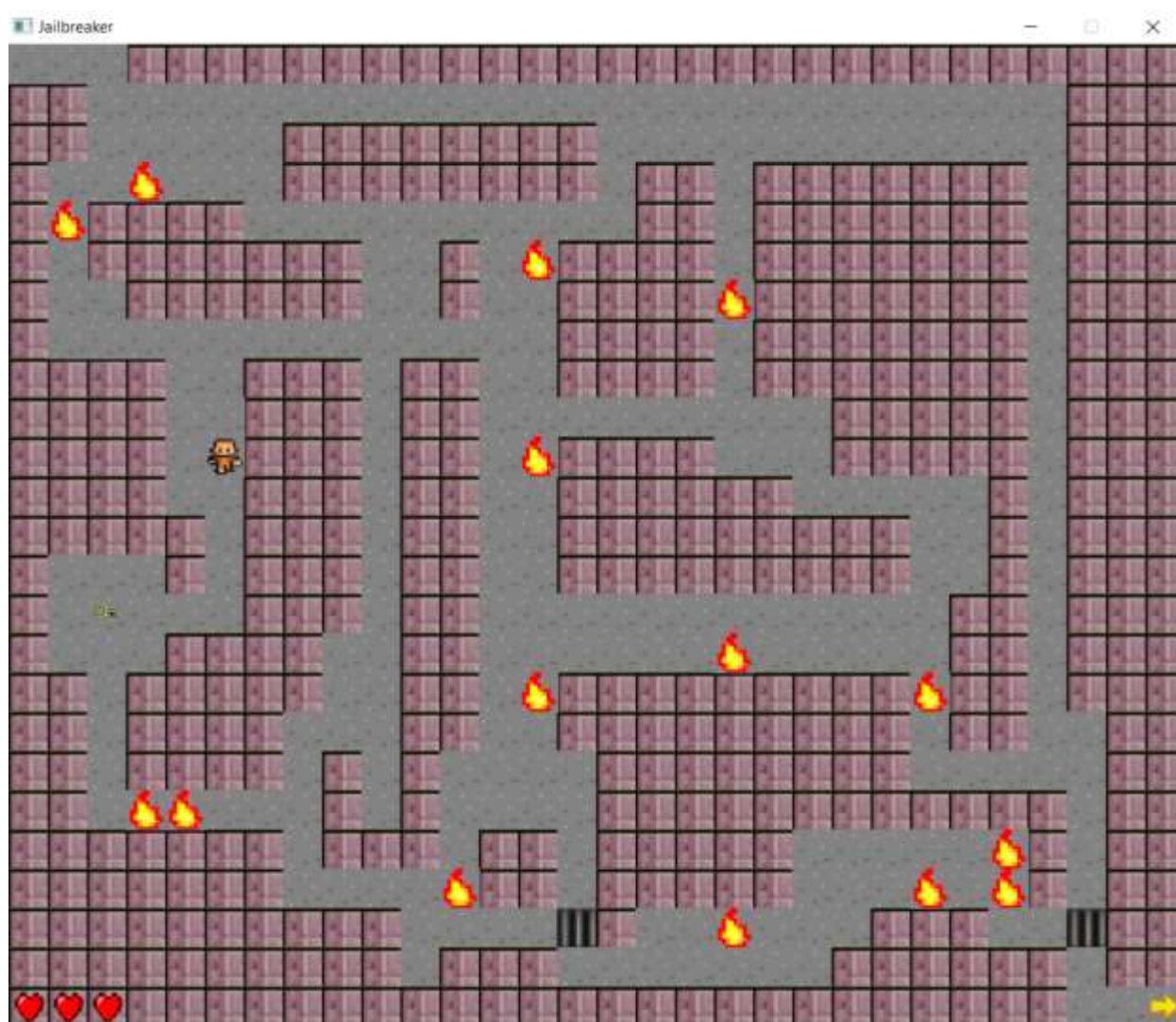


Harta nivelului 3

Al patrulea nivel

Ultimul nivel este caracterizat de harta care începe să fie cuprinsă de flăcări. Caracterul trebuie să ajungă cât mai repede la cheie și să scape cât timp toate ieșirile nu sunt blocate.

Atenție! Jucătorul poate trece prin foc, dar va pierde o viață pentru fiecare flacără.



Harta nivelului 4

Game Over

În cazul în care jucătorul rămâne fără vieți, va fi întâmpinat de un mesaj „Game Over”, fiind nevoit să reia jocul de la început.



Mesajul de „Game Over”

Finalul jocului

Odată ce jucătorul a trecut prin toate nivelurile cu succes, acesta va fi întâmpinat de un mesaj, după care jocul se va închide.



Mesajul de la finalul jocului

5. CONCLUZII

Aplicația se află încă în stadiul de dezvoltare. Pe viitor, se urmărește îmbunătățirea performanței jocului pentru calculatoarele cu o putere de procesare mai slabă.

De asemenea, vor fi adăugate mai multe niveluri ce prezintă provocări cu un nivel de dificultate mai ridicat. Totodată, și îmbunătățirea graficilor, pentru o experiență mai plăcută a jucătorilor, se află în lista de update-uri.

Mai multe instrumente de care să se folosească jucătorul (ex: pistol, găleată cu apă, rangă etc.) vor fi adăugate. Urmează să se implementeze și un sistem de scor bazat pe bănuți, ce vor fi ascunși în labirint. Se vor adăuga mai multe costumații pentru caracter și mai multe NPC-uri (non-player character, ex: alte personaje ce vor să evadeze).

Pentru jucătorii ce adoră provocările, se va adăuga un nivel asemănător celui de-al treilea, în care polițiștii vor ști unde este caracterul și îl vor urmări constant (folosind algoritmul lui Lee).

BIBLIOGRAFIE

- www.google.com
- www.infoscience.3x.ro
- www.microsoft.com
- Wikipedia.ro
- Ctrl-d.ro/tips

ANEXE

```
#include <iostream>
#include <fstream>
#include <windows.h>
#include <graphics.h>
#include <mmsystem.h>
using namespace std;

int a[100][100], i, ii, j, jj, m, n, level=1, t, schimbare,
revers[5], revers3[5], viata=3, ok_lanterna,
prima_data_lanterna, track, ok_OY=1, ok_OX=1, track3, hit;
int x, y, x1, y1, x2[5], y2[5], x3[5], y3[5], x_lvl2,
y_lvl2; ///coordonatele caracterelor
char ch; ///directia de deplasare
ifstream f("lvl1.in");
ifstream g("lvl2.in");
ifstream h("lvl3.in");
ifstream ff("lvl4.in");
int i1, j1, i2[5], j2[5], i3[5], j3[5];
int ok=0, game_over=1, prima_data_cheie=0, prima_data_poarta=0;
void init()
{
    t=0;
    i=1;
    j=1;
    hit=0;
    schimbare=0;
    ok=0;
    prima_data_lanterna=0;
    prima_data_cheie=0;
    prima_data_poarta=0;
}

void citire1()
{
    f>>m>>n;
    for(i=1; i<=m; i++)
        for(j=1; j<=n; j++)
            f>>a[i][j];
    f>>i1>>j1;
}

void citire2()
{
    g>>m>>n;
```

```

        for(i=1; i<=m; i++)
            for(j=1; j<=n; j++)
                g>>a[i][j];
        g>>i1>>j1;
    }
void citire3()
{
    h>>m>>n;
    for(i=1; i<=m; i++)
        for(j=1; j<=n; j++)
            h>>a[i][j];
    h>>i1>>j1;
    i2[1]=12;
    j2[1]=8;
    i2[2]=12;
    j2[2]=24;
    ///
    i3[1]=17;
    j3[1]=7;
    i3[2]=8;
    j3[2]=15;
    i3[3]=20;
    j3[3]=15;
}

```

```

void citire4()
{
    ff>>m>>n;
    for(i=1; i<=m; i++)
        for(j=1; j<=n; j++)
            ff>>a[i][j];
    ff>>i1>>j1;
}
int W,H,L=30;
void fereastr()
{
    W=n*L;
    H=m*L;
    int top=20, left=200;        ///coordonatele coltului din
    stanga sus pt fereastr
    initwindow(W,H,"Jailbreaker",left,top);
    PlaySound("1.wav", NULL, SND_FILENAME|SND_LOOP|SND_ASYNC);
    readimagefile("controls.jpg",0,0,W,H);
}

```

```

        getch();
    }
    void harta()
    {
        if(level!=2)
            for(i=1; i<=m; i++)
                for(j=1; j<=n; j++)
                {
                    x=(j-1)*L;
                    y=(i-1)*L;
                    if(a[i][j]==0)
                        readimagefile("0.jpg",x,y,x+L,y+L);
                    if(a[i][j]==1)
                        readimagefile("1.jpg",x,y,x+L,y+L);
                    if(a[i][j]==2)
                        readimagefile("2.jpg",x,y,x+L,y+L);
                    if(a[i][j]==3)
                        readimagefile("3.jpg",x,y,x+L,y+L);
                    if(a[i][j]==4)
                        readimagefile("flashlight.jpg",x,y,x+L,y+L);
                    if(i==25&&j==30)
                        readimagefile("next.jpg",x,y,x+L,y+L);

                    if(i==25&&j==1)
                        for(ii=1; ii<=viata; ii++)
                        {
                            x=(j-1)*L;
                            y=(i-1)*L;
                            readimagefile("viata.jpg",x,y,x+L,y+L);
                            if(ii<viata)
                                j++;
                        }
                }
        x=(j1-1)*L;
        y=(i1-1)*L;
        readimagefile("D.jpg",x,y,x+L,y+L);
    }

    void tranzitie()
    {
        cleardevice();
        if(ok_lanterna==1)
            level++;
    }
    void conditii_cheie_poarta_lanterna()
    {
        if(a[i1][j1]==2 && prima_data_cheie==0)

```

```

{
    ok=1;
    a[i1][j1]=1;
    prima_data_cheie=1;
    PlaySound("2.wav", NULL, SND_FILENAME|SND_ASYNC);
    delay(1000);
    PlaySound("1.wav", NULL,
SND_FILENAME|SND_LOOP|SND_ASYNC);
}

```

```

if(a[i1][j1]==4 && prima_data_lanterna==0)
{
    ok_lanterna=1;
    a[i1][j1]=1;
    prima_data_lanterna=1;
    PlaySound("2.wav", NULL, SND_FILENAME|SND_ASYNC);
    delay(1000);
    PlaySound("1.wav", NULL,
SND_FILENAME|SND_LOOP|SND_ASYNC);
}
if(schimbare==1 && ok==1 && prima_data_poarta==0)
{
    prima_data_poarta=1;
    schimbare=0;
    PlaySound("2.wav", NULL, SND_FILENAME|SND_ASYNC);
    delay(1000);
    PlaySound("1.wav", NULL,
SND_FILENAME|SND_LOOP|SND_ASYNC);
}
}
void conditii_deschidere_poarta()
{
    if(a[i1+1][j1]==3)
    {
        a[i1+1][j1]=1;
        x1=(j1-1)*L;
        y1=(i1)*L;
        delay(5);
        readimagefile("1.jpg",x1,y1,x1+L,y1+L);
    }
}

```

```

    if(a[i1-1][j1]==3)
    {
        a[i1-1][j1]=1;
        x1=(j1-1)*L;
        y1=(i1-2)*L;
        delay(5);
        readimagefile("1.jpg",x1,y1,x1+L,y1+L);
    }

    if(a[i1][j1-1]==3)
    {
        a[i1][j1-1]=1;
        x1=(j1-2)*L;
        y1=(i1-1)*L;
        delay(5);
        readimagefile("1.jpg",x1,y1,x1+L,y1+L);
    }
    if(a[i1][j1+1]==3)
    {
        a[i1][j1+1]=1;
        x1=(j1)*L;
        y1=(i1-1)*L;
        delay(5);
        readimagefile("1.jpg",x1,y1,x1+L,y1+L);
    }
    schimbare=1;
}
void miscare_politist_OX()
{
    for(int nr_pol=1; nr_pol<=2; nr_pol++)
    {
        x2[nr_pol]=(j2[nr_pol]-1)*L;
        y2[nr_pol]=(i2[nr_pol]-1)*L;
        if(j1==j2[nr_pol])
            track=1;
        if(j1!=j2[nr_pol])
            track=0;
        if(a[i2[nr_pol]][j2[nr_pol]-1]==0||a[i2[nr_pol]][j2[nr_pol]-1]==3)
        {
            revers[nr_pol]=1;

            readimagefile("policeRIGHT.jpg",x2[nr_pol],y2[nr_pol],x2[nr_pol]
+L,y2[nr_pol]+L);

```

```

    }
    else if(a[i2[nr_pol]][j2[nr_pol]-
1] !=0 && revers[nr_pol]==0 && track==0)

readimagefile("policeLEFT.jpg",x2[nr_pol],y2[nr_pol],x2[nr_pol]+
L,y2[nr_pol]+L);

if(a[i2[nr_pol]][j2[nr_pol]+1]==0 || a[i2[nr_pol]][j2[nr_pol]+1]==
3)
    {
        revers[nr_pol]=0;

readimagefile("policeLEFT.jpg",x2[nr_pol],y2[nr_pol],x2[nr_pol]+
L,y2[nr_pol]+L);
    }
    else
if(a[i2[nr_pol]][j2[nr_pol]+1] !=0 && revers[nr_pol]==1 && track==0)

readimagefile("policeRIGHT.jpg",x2[nr_pol],y2[nr_pol],x2[nr_pol]
+L,y2[nr_pol]+L);
    t++;
    if(t%10==0)
    {
        if(track==1)
        {
            if(i1<i2[nr_pol])
            {
                for(i=i1+1; i<i2[nr_pol]; i++)
                if(a[i][j1]==0)
                {
                    track=0;
                    break;
                }
            }
            if(track==1)
            {
                i2[nr_pol]--;

readimagefile("1.jpg",x2[nr_pol],y2[nr_pol],x2[nr_pol]+L,y2[nr_p
ol]+L);

                y2[nr_pol]=(i2[nr_pol]-1)*L;

readimagefile("policeUP.jpg",x2[nr_pol],y2[nr_pol],x2[nr_pol]+L,
y2[nr_pol]+L);
            }
        }
    }
}

```

```

if(i1>i2[nr_pol])
{
    for(i=i2[nr_pol]+1; i<i1; i++)
        if(a[i][j1]==0)
        {
            track=0;
            break;
        }
    if(track==1)
    {
        i2[nr_pol]++;

readimagefile("1.jpg",x2[nr_pol],y2[nr_pol],x2[nr_pol]+L,y2[nr_pol]+L);

        y2[nr_pol]=(i2[nr_pol]-1)*L;

readimagefile("policeDOWN.jpg",x2[nr_pol],y2[nr_pol],x2[nr_pol]+L,y2[nr_pol]+L);
    }
}
if(track==0)
{
    if(revers[nr_pol]==0)
    {
        j2[nr_pol]--;

readimagefile("1.jpg",x2[nr_pol],y2[nr_pol],x2[nr_pol]+L,y2[nr_pol]+L);

    }
    if(revers[nr_pol]==1)
    {
        j2[nr_pol]++;

readimagefile("1.jpg",x2[nr_pol],y2[nr_pol],x2[nr_pol]+L,y2[nr_pol]+L);

    }
}
if(t==11)
    t=0;

```



```

if (i1==i2[nr_pol]&&j1==j2[nr_pol])
{
    viata--;
    PlaySound("3.wav", NULL, SND_FILENAME|SND_ASYNC);
    delay(1000);
    PlaySound("1.wav", NULL, SND_FILENAME|SND_ASYNC);
    y=(25-1)*L;
    x=(viata)*L;
    readimagefile("0.jpg",x,y,x+L,y+L);
    i1=1;
    j1=1;
    x=(j1-1)*L;
    y=(i1-1)*L;
    readimagefile("D.jpg",x,y,x+L,y+L);
}
}
}
void miscare_politist_OY()
{
    for(int nr3_pol=1; nr3_pol<=3; nr3_pol++)
    {
        x3[nr3_pol]=(j3[nr3_pol]-1)*L;
        y3[nr3_pol]=(i3[nr3_pol]-1)*L;
        if(i1==i3[nr3_pol])
            track3=1;
        if(i1!=i3[nr3_pol])
            track3=0;
        if(a[i3[nr3_pol]-1][j3[nr3_pol]]==0||a[i3[nr3_pol]-
1][j3[nr3_pol]]==3)
        {
            revers3[nr3_pol]=1;

            readimagefile("policeDOWN.jpg",x3[nr3_pol],y3[nr3_pol],x3[nr3_po
1]+L,y3[nr3_pol]+L);
        }
        else if(a[i3[nr3_pol]-
1][j3[nr3_pol]]!=0&&revers3[nr3_pol]==0&&track3==0)

            readimagefile("policeUP.jpg",x3[nr3_pol],y3[nr3_pol],x3[nr3_pol]
+L,y3[nr3_pol]+L);

        if(a[i3[nr3_pol]+1][j3[nr3_pol]]==0||a[i3[nr3_pol]+1][j3[nr3_pol]
]==3)
        {
            revers3[nr3_pol]=0;

```

```

readimagefile("policeUP.jpg",x3[nr3_pol],y3[nr3_pol],x3[nr3_pol]
+L,y3[nr3_pol]+L);
    }
    else
if(a[i3[nr3_pol]+1][j3[nr3_pol]]!=0&&revers3[nr3_pol]==1&&track3
==0)

readimagefile("policeDOWN.jpg",x3[nr3_pol],y3[nr3_pol],x3[nr3_po
l]+L,y3[nr3_pol]+L);
    if(t%15==0)
    {
        if(track3==1)
        {
            if(j1<j3[nr3_pol])
            {
                for(j=j1+1; j<j3[nr3_pol]; j++)
                    if(a[i1][j]==0)
                    {
                        track3=0;
                        break;
                    }
                if(track3==1)
                {
                    j3[nr3_pol]--;

readimagefile("1.jpg",x3[nr3_pol],y3[nr3_pol],x3[nr3_pol]+L,y3[n
r3_pol]+L);

                    x3[nr3_pol]=(j3[nr3_pol]-1)*L;

readimagefile("policeLEFT.jpg",x3[nr3_pol],y3[nr3_pol],x3[nr3_po
l]+L,y3[nr3_pol]+L);
                }
            }
            if(j1>j3[nr3_pol])
            {
                for(j=j3[nr3_pol]+1; j<j1; j++)

                    if(a[i1][j]==0)
                    {
                        track3=0;
                        break;
                    }
                if(track3==1)
                {
                    j3[nr3_pol]++;

```

```

readimagefile("1.jpg",x3[nr3_pol],y3[nr3_pol],x3[nr3_pol]+L,y3[nr3_pol]+L);
                                x3[nr3_pol]=(j3[nr3_pol]-1)*L;

readimagefile("policeRIGHT.jpg",x3[nr3_pol],y3[nr3_pol],x3[nr3_pol]+L,y3[nr3_pol]+L);
                                }
                                }
                                }
                                if(track3==0)
                                {
                                    if(revers3[nr3_pol]==0)
                                    {
                                        i3[nr3_pol]--;

readimagefile("1.jpg",x3[nr3_pol],y3[nr3_pol],x3[nr3_pol]+L,y3[nr3_pol]+L);
                                }
                                if(revers3[nr3_pol]==1)
                                {
                                    i3[nr3_pol]++;

readimagefile("1.jpg",x3[nr3_pol],y3[nr3_pol],x3[nr3_pol]+L,y3[nr3_pol]+L);
                                }
                                }
                                }

                                }

if(i1==i3[nr3_pol]&&j1==j3[nr3_pol])
{
    PlaySound("3.wav", NULL, SND_FILENAME|SND_ASYNC);
    delay(1000);
    PlaySound("1.wav", NULL, SND_FILENAME|SND_ASYNC);
    viata--;
    y=(25-1)*L;
    x=(viata)*L;
    readimagefile("0.jpg",x,y,x+L,y+L);
    i1=1;
    j1=1;
    x=(j1-1)*L;

```

```

        y=(i1-1)*L;
        readimagefile("D.jpg",x,y,x+L,y+L);
    }
}
void miscare_carac()
{
    if(kbhit())
    {
        ch=getch();
        x=(j1-1)*L;
        y=(i1-1)*L;
        if(ch==KEY_RIGHT && (a[i1][j1+1]==1 || a[i1][j1+1]==2 ||
a[i1][j1+1]==4 || a[i1][j1+1]==5))
        {
            readimagefile("1.jpg",x,y,x+L,y+L);
            j1++;
            x=(j1-1)*L;
            y=(i1-1)*L;
            readimagefile("D.jpg",x,y,x+L,y+L);
            if(level==2)
                for(j=j1+1; a[i1][j-1]!=0 && j<=j1+6; j++)
                {
                    x_lvl2=(j-1)*L;
                    y_lvl2=(i1-1)*L;
                    if(a[i1][j]==0)

readimagefile("0.jpg",x_lvl2,y_lvl2,x_lvl2+L,y_lvl2+L);

                    if(a[i1][j]==1)

readimagefile("1.jpg",x_lvl2,y_lvl2,x_lvl2+L,y_lvl2+L);
                    if(a[i1][j]==2)

readimagefile("2.jpg",x_lvl2,y_lvl2,x_lvl2+L,y_lvl2+L);
                    if(a[i1][j]==3)

readimagefile("3.jpg",x_lvl2,y_lvl2,x_lvl2+L,y_lvl2+L);
                    readimagefile("tranzitie.jpg",0,0,L*(j1-
2),L*30);
                }
        }
        if(ch==KEY_LEFT && (a[i1][j1-1]==1 || a[i1][j1-1]==2 ||
a[i1][j1-1]==4 || a[i1][j1-1]==5))
        {
            readimagefile("1.jpg",x,y,x+L,y+L);
            j1--;

```

```

x=(j1-1)*L;
y=(i1-1)*L;
readimagefile("A.jpg",x,y,x+L,y+L);
if(level==2)
    for(j=j1-1; a[i1][j+1]!=0 && j>=j1-6; j--)
    {
        x_lv12=(j-1)*L;
        y_lv12=(i1-1)*L;
        if(a[i1][j]==0)

readimagefile("0.jpg",x_lv12,y_lv12,x_lv12+L,y_lv12+L);
        if(a[i1][j]==1)

readimagefile("1.jpg",x_lv12,y_lv12,x_lv12+L,y_lv12+L);
        if(a[i1][j]==2)

readimagefile("2.jpg",x_lv12,y_lv12,x_lv12+L,y_lv12+L);
        if(a[i1][j]==3)

readimagefile("3.jpg",x_lv12,y_lv12,x_lv12+L,y_lv12+L);

readimagefile("tranzitie.jpg",L*(j1+1),0,L*30,L*30);
    }
}

if(ch==KEY_UP && (a[i1-1][j1]==1 || a[i1-1][j1]==2 || a[i1-1][j1]==4 || a[i1-1][j1]==5))
{
    readimagefile("1.jpg",x,y,x+L,y+L);
    i1--;
    x=(j1-1)*L;
    y=(i1-1)*L;
    readimagefile("W.jpg",x,y,x+L,y+L);
    if(level==2)
        for(i=i1-1; a[i+1][j1]!=0 && i>=i1-6; i--)
        {
            x_lv12=(j1-1)*L;
            y_lv12=(i-1)*L;
            if(a[i][j1]==0)

readimagefile("0.jpg",x_lv12,y_lv12,x_lv12+L,y_lv12+L);
            if(a[i][j1]==1)

```

```

readimagefile("1.jpg",x_lvl2,y_lvl2,x_lvl2+L,y_lvl2+L);
    if(a[i][j1]==2)

readimagefile("2.jpg",x_lvl2,y_lvl2,x_lvl2+L,y_lvl2+L);
    if(a[i][j1]==3)

readimagefile("3.jpg",x_lvl2,y_lvl2,x_lvl2+L,y_lvl2+L);

readimagefile("tranzitie.jpg",0,L*(i1+1),L*30,L*30);
    }
    }
    if(ch==KEY_DOWN && (a[i1+1][j1]==1 || a[i1+1][j1]==2 ||
a[i1+1][j1]==4 || a[i1+1][j1]==5))
    {
        readimagefile("1.jpg",x,y,x+L,y+L);
        i1++;
        x=(j1-1)*L;
        y=(i1-1)*L;
        readimagefile("S.jpg",x,y,x+L,y+L);
        if(level==2)
            for(i=i1+1; a[i-1][j1]!=0 && i<=i1+6; i++)
            {
                x_lvl2=(j1-1)*L;
                y_lvl2=(i-1)*L;
                if(a[i][j1]==0)

readimagefile("0.jpg",x_lvl2,y_lvl2,x_lvl2+L,y_lvl2+L);

                if(a[i][j1]==1)

readimagefile("1.jpg",x_lvl2,y_lvl2,x_lvl2+L,y_lvl2+L);
                    if(a[i][j1]==2)

readimagefile("2.jpg",x_lvl2,y_lvl2,x_lvl2+L,y_lvl2+L);
                    if(a[i][j1]==3)

readimagefile("3.jpg",x_lvl2,y_lvl2,x_lvl2+L,y_lvl2+L);

readimagefile("tranzitie.jpg",0,0,L*30,L*(i1-2));
            }
    }
    if(level==2)
    {
        for(i=i1-1; i<=i1+1; i++)
            for(j=j1-1; j<=j1+1; j++)
                if(i1!=i&&j1!=j)

```

```

        {
            x=(j-1)*L;
            y=(i-1)*L;
            if(a[i][j]==0)
                readimagefile("0.jpg",x,y,x+L,y+L);
            if(a[i][j]==1)
                readimagefile("1.jpg",x,y,x+L,y+L);
            if(a[i][j]==2)
                readimagefile("2.jpg",x,y,x+L,y+L);
            if(a[i][j]==3)
                readimagefile("3.jpg",x,y,x+L,y+L);
            if(i==25&&j==30)

readimagefile("next.jpg",x,y,x+L,y+L);
        }
    }
}
void foc()
{
    if(kbhit()||hit==1)
    {
        hit=1;
        i=rand()%25+1;
        j=rand()%29+1;
        t++;

if(t%2000000==0)
        if(i!=i1&&j!=j1)
        {
            if(a[i][j]==1)
            {
                a[i][j]=5;
                x1=(j-1)*L;
                y1=(i-1)*L;
                readimagefile("lit.jpg",x1,y1,x1+L,y1+L);
            }
            if(a[i1][j1]==5)
            {
                PlaySound("3.wav", NULL,
SND_FILENAME|SND_ASYNC);
                delay(1000);
                PlaySound("1.wav", NULL,
SND_FILENAME|SND_ASYNC);
                viata--;
                y=(25-1)*L;
                x=(viata)*L;

```

```

        readimagefile("0.jpg",x,y,x+L,y+L);
        a[i1][j1]=1;
    }

    }
    if(t==2000001)
        t=0;
}
}
int main()
{
    citire1();

    ///-----

    fereastra();

    ///-----

    harta();

    ///-----
    ch=getch();
    while(game_over==1)
    {
        miscare_carac();
        if(ok==1 && (a[i1+1][j1]==3 || a[i1-1][j1]==3 ||
a[i1][j1-1]==3 || a[i1][j1+1]==3))
            conditii_deschidere_poarta();
        conditii_cheie_poarta_lanterna();
        if(i1==25 && j1==30 && ok_lanterna==1)
        {
            tranzitie();
            if(level==2)
                citire2();
            if(level==3)
                citire3();
            if(level==4)
                citire4();
            if(level!=5)
            {
                init();
                harta();
            }
        }
        else if(i1==25 && j1==30 && ok_lanterna==0)
        {

```



```

        readimagefile("findflash.jpg",0,0,W,H);
        delay(2000);
        getch();
        tranzitie();
        i1=25;
        j1=28;
        harta();
    }
    if(level==3)
    {
        miscare_politist_OX();
        miscare_politist_OY();
    }
    if(level==4)
    {
        foc();
    }
    if(viata==0)
    {
        game_over=0;
        readimagefile("gameover.jpg",0,0,W,H);
        delay(2000);
        getch();
    }
    if(level==5)
    {
        game_over=0;
        readimagefile("congrats.jpg",0,0,W,H);
        delay(2000);
        getch();
    }
}

if (HWND hWnd = FindWindow(NULL,"Jailbreaker"))
    SendMessage(hWnd, WM_CLOSE, 0, 0);

return 0;
}

```