

JOC 2048

LUCRARE PENTRU ATESTAREA COMPETENȚELOR PROFESIONALE

Profesori coordonatori:

Ivașc Cornelia

Țibu Mirela

Elev: Vîlcu Răzvan-Valeriu

Clasa: a XII-a D

Cuprins

1. Motivația alegerii temei	pag. 3
2. Structura aplicației	pag. 4
3. Detalii tehnice	pag. 5
4. Resurse hard și soft necesare	pag. 8
5. Modalități de utilizare	pag. 10
6. Posibilități de dezvoltare	pag. 11

1. Motivația alegerii temei

Motivația alegerii acestei teme este legată de pasiunea mea pentru jocurile pe calculator, ceea ce a reprezentat pentru mine oportunitatea de a învăța și de a aprofunda cunoștințele pe care le-am dobândit pe parcursul studiilor mele.

2048 este un joc de strategie și a fost creat cu scopul de a oferi utilizatorilor, pe lângă divertisment și o experiență provocatoare.

Dincolo de aspectul pur distractiv, jocul 2048 implică mecanisme și procedee necesare procesului de învățare, ceea ce conduce la stimularea gândirii, coordonarea atenției și dezvoltarea atenției distributive, dezvoltarea memoriei, crearea unor conexiuni și corelații.

2. Structura aplicației

Aplicația a fost realizată în limbajul de programare C#, fiind un proiect de tip *Windows Form Application*.

Când deschidem aplicația, observăm o fereastră de tip pop-up, care conține instrucțiuni de utilizare a jocului (Fig. 1). După ce utilizatorul a citit instrucțiunile și închide fereastra de tip pop-up, pe ecran va apărea jocul (Fig. 2).

Observăm trei componente: tabla de joc, scorul curent și cel mai bun scor.

1. *Tabla de joc* este alcătuită dintr-o matrice 4x4, pe 4 dintre pătrățelele apărând deja niște elemente, puteri ale lui 2, respectiv 2^1 și 2^2 . Pătrățelele au culori diferite, fiecare valoare a puterii lui 2 fiind identificată cu altă culoare. Pătrățelele gri din aplicație reprezintă locuri libere care vor fi ocupate pe parcursul jocului, prin mutările pe care utilizatorul le va face.

2. *Scorul curent* reprezintă scorul acumulat în partida curentă, iar dacă acesta este mai mare decât cel mai bun scor, acesta îi va lua locul, utilizatorul putând observa în timp real modificarea scorului.

3. *Cel mai bun scor* reprezintă scorul cel mai mare pe care un utilizator l-a avut pe parcursul a mai multor jocuri, de la instalarea aplicației.

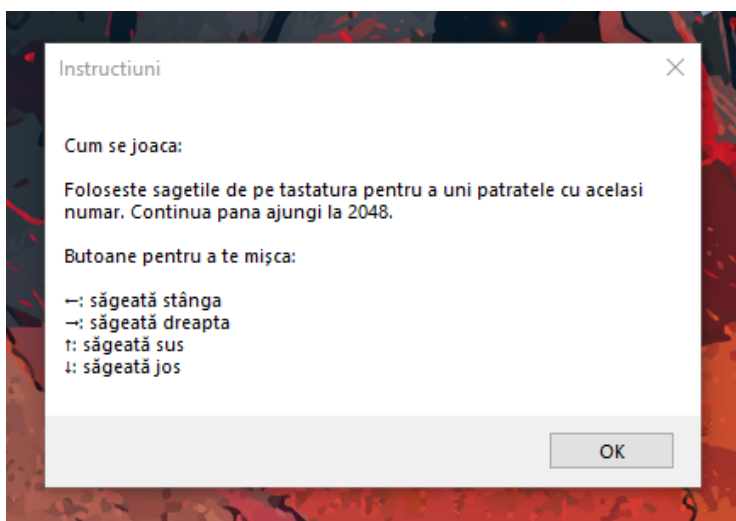


Fig. 1 – Instrucțiuni de utilizare

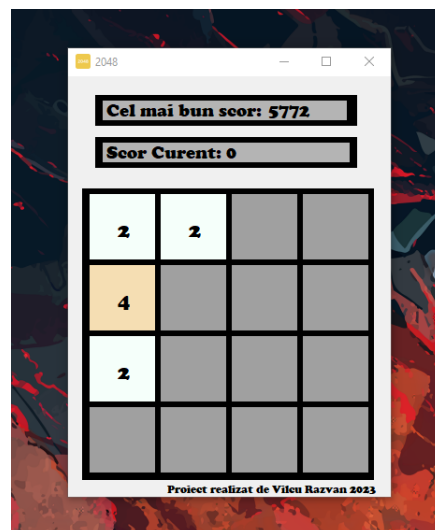


Fig. 2 - Aplicația

3. Detalii tehnice

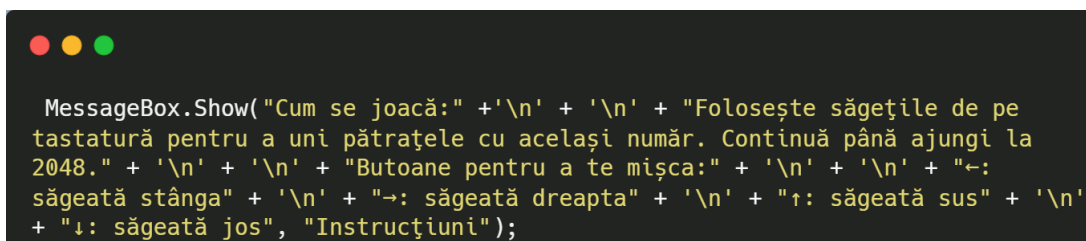
Proiectul a fost construit în două etape, una legată de stilizarea interfeței și cealaltă are legătură cu codul din spatele aplicației.

Prima etapă a fost să construiesc interfața aplicației cu ajutorul instrumentelor de care dispuneam, printre care PictureBox și TextBox, iar pentru a adăuga o iconiță și titlul, atât pentru aplicație, cât și pentru Shortcut, m-am folosit de proprietățile Formului.

A doua etapă a presupus construcția codului, acesta fiind realizat prin îmbinarea mai multor elemente precum:

- *Mesajul ce apare la deschiderea aplicației*

Pentru a crea fereastra de început, unde apar instrucțiunile, am folosit MessageBox. (Fig. 3)



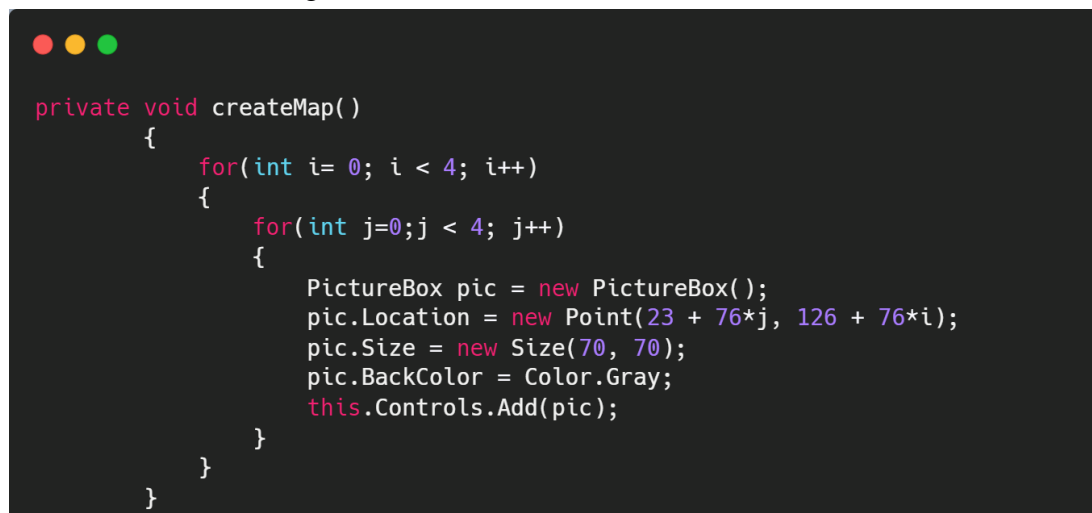
```

MessageBox.Show("Cum se joacă:" + '\n' + '\n' + "Folosește săgețile de pe  
tastatură pentru a uni pătrățele cu același număr. Continuă până ajungi la  
2048." + '\n' + '\n' + "Butoane pentru a te mișca:" + '\n' + '\n' + "←:  
săgeată stânga" + '\n' + "→: săgeată dreapta" + '\n' + "↑: săgeată sus" + '\n'  
+ "↓: săgeată jos", "Instrucțiuni");
    
```

Fig. 3 – Codul pentru a realiza fereastra de tip pop-up.

- *Generarea matricei și pătrățelelor*

Am folosit o funcție pentru a crea o matrice 4x4, care va fi utilizată drept tablă de joc pe care se mișcă pătrățelele. Pentru fiecare loc din matrice, atribuim un pătrățel de culoare gri, semnificând un loc liber (Fig. 4)



```

private void createMap()
{
    for(int i= 0; i < 4; i++)
    {
        for(int j=0;j < 4; j++)
        {
            PictureBox pic = new PictureBox();
            pic.Location = new Point(23 + 76*j, 126 + 76*i);
            pic.Size = new Size(70, 70);
            pic.BackColor = Color.Gray;
            this.Controls.Add(pic);
        }
    }
}
    
```

Fig. 4

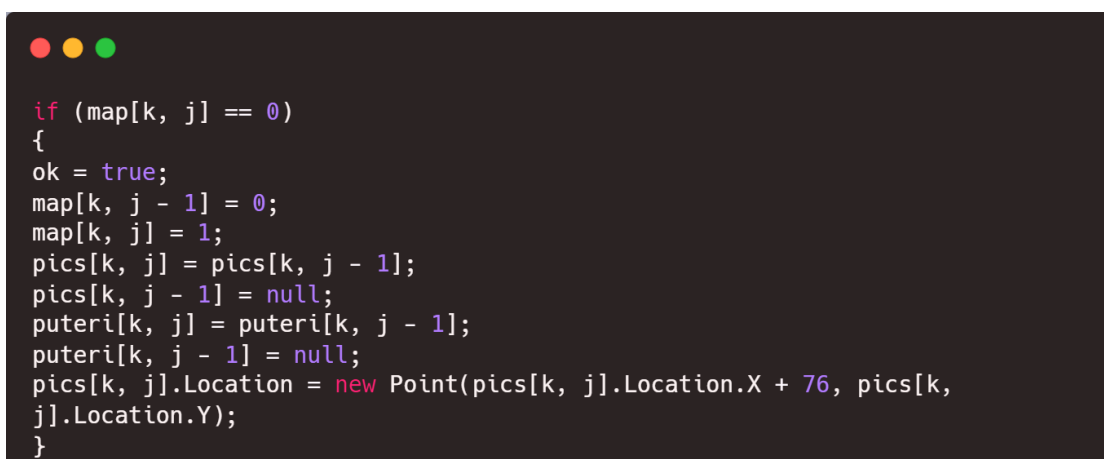
Pentru pătrățele, am realizat două funcții: una ce plasează pe tabla de joc trei pătrățele, cu poziții preconfigurate, iar altă funcție ce generează aleatoriu pe tabla de joc pătrățele ajutătoare, dacă după apăsarea unui buton s-au putut mișca pătrățelele.

- *Funcția de apăsare a unui buton*

În funcție de care buton a fost apăsător de către utilizator, din lista celor menționate în secțiunea instrucțiuni, se vor întâmpla următoarele:

1. Se parcurge matricea în direcția sugerată de butonul apăsător și se verifică dacă există locuri libere.

2. Dacă locurile sunt libere, se mută pătrățelele până la capăt, iar variabila ok se va pune pe valoarea true, semnificând că se pot plasa pătrățele ajutătoare. Pătrățelele cu valori se mută pe pozițiile noi, marcându-se ca fiind ocupate, iar pozițiile de pe care au plecat sunt marcate ca fiind libere. (Fig. 5)



```

if (map[k, j] == 0)
{
    ok = true;
    map[k, j - 1] = 0;
    map[k, j] = 1;
    pics[k, j] = pics[k, j - 1];
    pics[k, j - 1] = null;
    puteri[k, j] = puteri[k, j - 1];
    puteri[k, j - 1] = null;
    pics[k, j].Location = new Point(pics[k, j].Location.X + 76, pics[k, j].Location.Y);
}
    
```

Fig. 5

3. Dacă locurile nu sunt libere, se pot întâmpla două lucruri:

- dacă pătrățelele au aceleași valori, se adună până nu mai sunt două pătrățele de aceeași valoare, una lângă cealaltă, variabila ok punându-se pe valoarea true;

- dacă nu se mișcă, variabila ok va rămâne pe valoarea false, ceea ce înseamnă că nu se vor genera pătrățele ajutătoare. (Fig.6)

- *Variabila Scor și cel mai bun scor.*

Se calculează scorul sesiunii curente, adunând valoarea pătrățelului/pătrățelelor tocmai formate. Dacă scorul este mai mare decât *cel mai bun scor*, se salvează în sistemul calculatorului, ceea ce înseamnă că, dacă după o sesiune de joc, aplicația este închisă și redeschisă, cel mai bun scor va rămâne tot în tabela lui. Se transformă tipul valorii pătrățelelor din una de tip string, în una de tip int, se compară, iar dacă sunt egale, valoarea se adaugă la scor. Dacă scorul este mai mare decât valoarea din sistem, scorul cel mai bun este actualizat. (Fig. 6)

```

int a = int.Parse(puteri[k, j].Text);
int b = int.Parse(puteri[k, j - 1].Text);
if(a == b)
{
    ok = true;
    puteri[k, j].Text = (a+b).ToString();
    scor += (a + b);
    int x = int.Parse(lbl_value.Text);
    if (x <= scor)
    {
        lbl_value.Text = scor.ToString();
        Properties.Settings.Default.h_score = lbl_value.Text;
        Properties.Settings.Default.Save();
    }
    changeColor(a + b, k, j);
    scor_curent.Text = "Scor Curent: " + scor;
    map[k, j - 1] = 0;
    this.Controls.Remove(pics[k, j - 1]);
    this.Controls.Remove(puteri[k, j - 1]);
    pics[k, j - 1] = null;
    puteri[k, j - 1] = null;
}

```

Fig. 6

4. Resurse Hard și Soft necesare

Aplicația a fost realizată în *Microsoft Visual Studio 2022*, în limbajul C#, fiind un proiect de tip *Windows Form Application*. După realizare, în folderul aplicației, găsim atât un setup pentru aplicație (Fig. 7), cât și o versiune publicată a acesteia. (Fig. 8)



Fig. 7 – Setup găsit în: Atestat -> Setup -> 2048_setup -> Release

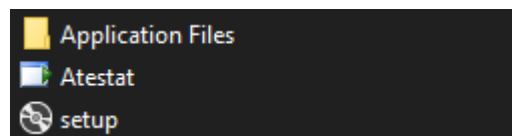


Fig. 8 – Versiunea publicată folosind ClickOne găsită în: Atestat -> Atestat -> 2048 -> bin -> publish

Întreg fișierul, care cuprinde setup, fișierele aplicației și aplicația, are 6.87 MB, din care aplicația are doar 204 KB, dar este nevoie de minim 7.11 MB pentru a avea loc pentru toate elementele esențiale. De asemenea, shortcut-ul are 4.00 KB.

Dacă utilizatorul dorește să folosească setup-ul obișnuit, acesta va fi întâmpinat de fereastra de început a instalării aplicației (Fig. 9). După ce utilizatorul apasă pe butonul Next, acesta va fi redirecționat la o fereastră în care acesta poate selecta unde să instaleze aplicația (Fig. 10). După ce aplicația s-a instalat, acesta o poate deschide din shortcut-ul creat pe desktop.

Pentru a putea instala aplicația și pentru a rula corespunzător, utilizatorul este nevoit să instaleze framework-ul *.NET Desktop Runtime 6.0.16 x86*. Dacă framework-ul nu este instalat deja, în timpul instalării, o fereastră pop-up va apărea, pentru a informa utilizatorul să instaleze framework-ul, să nu mai fie nevoit să facă acest lucru manual.

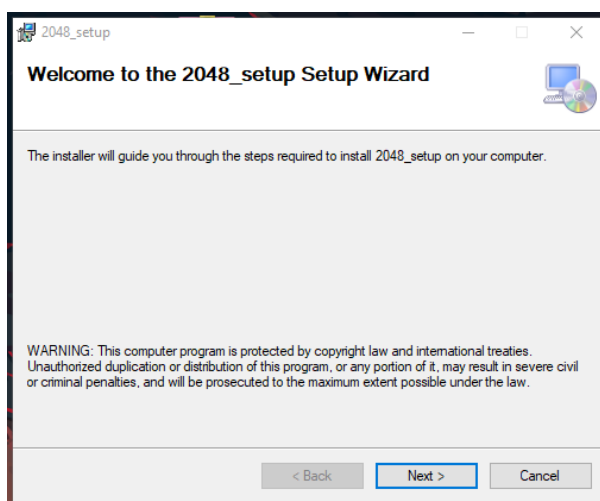


Fig. 9 – Fereastra de început a instalării

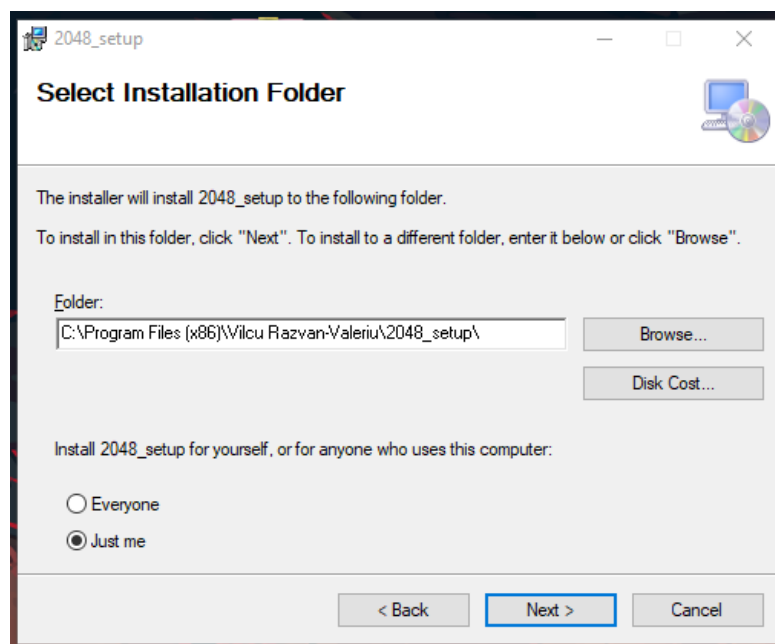


Fig. 10 – Fereastra care te lasă să alegi unde să instalezi aplicația

Aplicația funcționează pe calculatoarele/laptopurile, ce au ca sistem de operare *Microsoft Windows*, dar în caz că aceasta nu se deschide după instalare, aplicația a fost publicată folosind tehnologia *ClickOnce*, permițându-i utilizatorului să acceseze oricând aplicația, cu minimă interacțiune din partea acestuia.

Dacă utilizatorul dorește să șteargă aplicația, acesta o poate face fie acesând setup-ul și selectând opțiunea de a șterge aplicația, fie din Control Panel -> Programs -> Uninstall a program. (Fig. 11)

2048_setup	Vilcu Razvan-Valeriu	5/5/2023	204 KB	1.0.1
------------	----------------------	----------	--------	-------

Fig. 11 – Aplicația, așa cum apare în Control Panel

5. Modalități de utilizare

După ce jocul a fost descărcat, îl putem deschide din shortcut-ul plasat pe desktop. Jocul este unul de strategie, scopul acestuia este de a uni pătrățele cu valori egale până la obținerea pătrățelului cu valoarea 2048 (2^{11}), folosindu-se de proprietatea matematică $2^x + 2^x = 2^{x+1}$.

Pentru a mișca pătrățelele, utilizatorul se folosește de săgețile de pe tastatură, așa cum este menționat și în secțiunea de instrucțiuni. Cu fiecare mișcare, care vizează fie mutarea pătrățelelor, fie însumarea lor, utilizatorul primește un pătrățel ajutător, pe pozițiile din tabla de joc în care nu există alte pătrățele.

În timpul sesiunii, scorul se compară cu cel mai bun scor, iar dacă acesta este mai mare, cel mai bun scor este actualizat constant, salvându-se în sistemul utilizatorului, astfel chiar dacă jocul este închis, cel mai bun scor va rămâne permanent acolo. (Fig. 12)

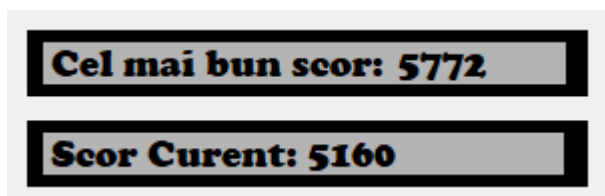


Fig. 12 – Exemplu tabela de scor în timpul unei sesiuni

Jocul se termină când orice buton s-ar apăsa, pătrățelele nu se mai pot însuma și nu mai există spații libere. (Fig. 13)

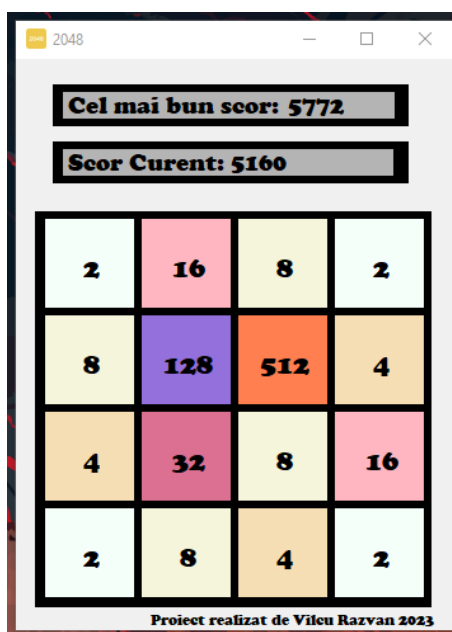


Fig. 13 – Exemplu de joc încheiat, în care utilizatorul a ajuns la 512

6. Posibilități de dezvoltare

Din punctul meu de vedere, aplicația în versiunea finală este completă, fără a mai fi necesar să fie adăugate sau scoase caracteristici. Însă la cererea utilizatorilor, aplicația poate fi dezvoltată astfel:

1. Portarea aplicației pe mai multe sisteme de operare precum iOS, macOS, Linux & Android, deoarece aplicația este suportată numai pe sistemul de operare Windows, pe calculator sau pe laptop.
2. Dezvoltarea unor noi moduri de joc, de astfel și dezvoltarea unor noi dificultăți pentru utilizatorii mai experimentați.