

Liceul Teoretic „Emil Racoviță” Vaslui

Jocul Sudoku

Lucrare de atestat profesional la informatică

Candidat
Hulubei Tudor

Îndrumător
prof. Dan Pracsiu

2022

Introducere

Sudoku (din japoneză 数, sū - cifră și 独, doku - unică), este un joc în formă de grilă inventat în 1979 și inspirat de pătratul latin și de problema celor 36 ofițeri a lui Leonhard Euler. Scopul jocului este de a umple această grilă cu cifrele de la 1 la 9 respectând anumite condiții, cu unele cifre fiind de la început dispuse în grilă.

Problema ofițerilor se poate enunța astfel: fie șase regimente diferite, fiecare regiment posedând șase ofițeri de grade diferite. Se cere să se plaseze cei 36 ofițeri într-o grilă de 6 x 6, fiecare ofițer ocupând câte căsuță, în așa fel că fiecare rând și fiecare coloană să conțină toate gradele și toate regimentele. Legătura între Sudoku și problema celor 36 de ofițeri este condiția care împiedică repetiția unui același element în grilă, ajungându-se în final tot la un joc care se folosește de principiul pătratului latin (combinarea a două pătrate latine în cazul pătratului greco-latin, pătrat latin subdivizat în mai multe regiuni în cazul Sudoku).

Grila jocului este un pătrat de nouă pe nouă căsuțe, subdivizat în tot atâtea pătrate identice, numite regiuni (vedeți figura). Regula jocului este simplă: fiecare rând, coloană sau regiune nu trebuie să conțină decât o dată cifrele de la unu la nouă. Formulată altfel, fiecare ansamblu trebuie să conțină cifrele de la unu la nouă o singură dată.

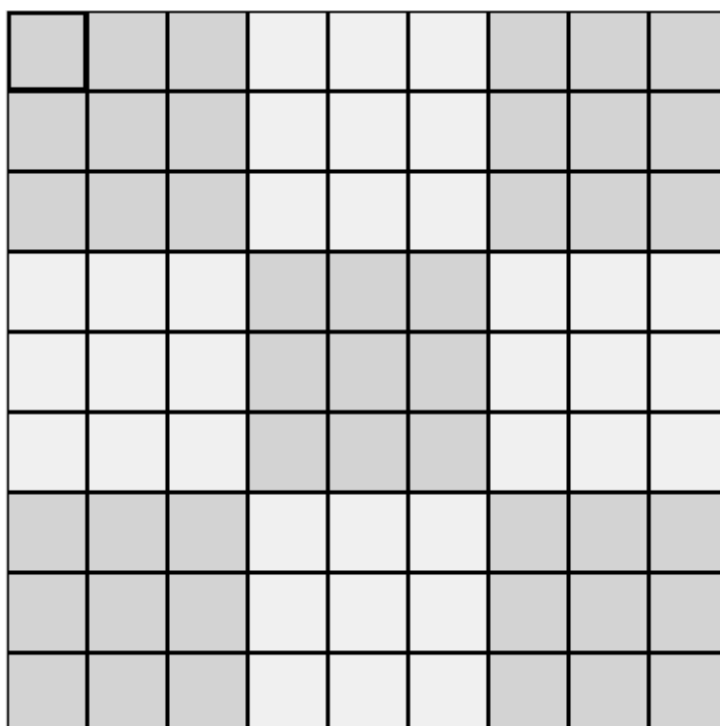
Cifrele nu reprezintă decât o convenție, relațiile aritmetice între ele nefiind de nici un folos. Orice ansamblu de simboluri distincte: litere, forme, culori, pot fi folosite fără a se modifica regulile jocului. *Dell Magazine*, primul care a publicat grile, a folosit cifre în publicațiile sale. Dimpotrivă, *Scramblets*, de la *Penny Press*, și *Sudoku Word*, de *Knight Features Syndicate*, folosesc amândouă litere.

Interesul jocului consistă în simplitatea regulilor sale și în complexitatea soluțiilor sale. Grilele publicate au de obicei un nivel de dificultate indicat, iar editorul are posibilitatea să indice și un timp de rezovare probabil. Cu toate că, în general, grilele ce conțin mai multe cifre completate sunt mai ușoare, inversul nu este în totdeauna adevărat. Dificultatea veritabilă a jocului rămâne totuși în a găsi suita exactă a cifrelor rămase.

Capitolul I – Jocul Sudoku, regulile jocului

Sudoku se joacă într-o grilă care este alcătuită din 9 x 9 căsuțe separate în 9 rânduri (poziționate orizontal) și 9 coloane (poziționate vertical) grupate în pătrate cu dimensiuni 3 x 3.

Fiecare rând, coloană și pătrat (9 căsuțe fiecare) trebuie să fie completat cu numerele de la 1 până la 9 fără a se repeta niciunul dintre numerele în rând, coloană sau pătrat.



La începutul jocului în unele dintre căsuțele ale grilei sunt prezente unele dintre numerele. Scopul este de a se completa toate pătratele goale cu cifrele de la 1 la 9. Cu cât mai mult sunt numerele din grila sudoku atunci când începeți jocul, cu atât mai ușor este puzzle-ul sudoku. Puzzle-urile Sudoku mai dificile au numai câteva numere completate la început, iar cele mai dificile dispun numai de 17 numere.

Este numai o regulă – Fiecare rând, fiecare coloană și fiecare dintre micile pătrate cu mărime de 3 x3 după terminarea completării trebuie să conțină toate cifrele de la 1 la 9. În plus, fiecare dintre cifrele poate exista numai o dată într-un anumit rând, coloană sau pătrat

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 9 | 8 | 2 | | 4 | 3 | | | 7 |
| | 6 | | 5 | 9 | | | | 1 |
| 1 | | | 7 | 2 | | | 9 | 4 |
| | | | | 6 | 5 | 2 | | 9 |
| 5 | | 9 | 8 | 3 | 2 | 7 | | 6 |
| 3 | | 6 | 9 | | 1 | | | 5 |
| | | | | 1 | | | | 3 |
| | 1 | | 3 | | 4 | | 7 | |
| 2 | 3 | 7 | 6 | | 9 | | | |

| | | |
|---|---|---|
| 7 | 1 | 3 |
| 2 | 6 | 5 |
| 3 | 4 | 9 |

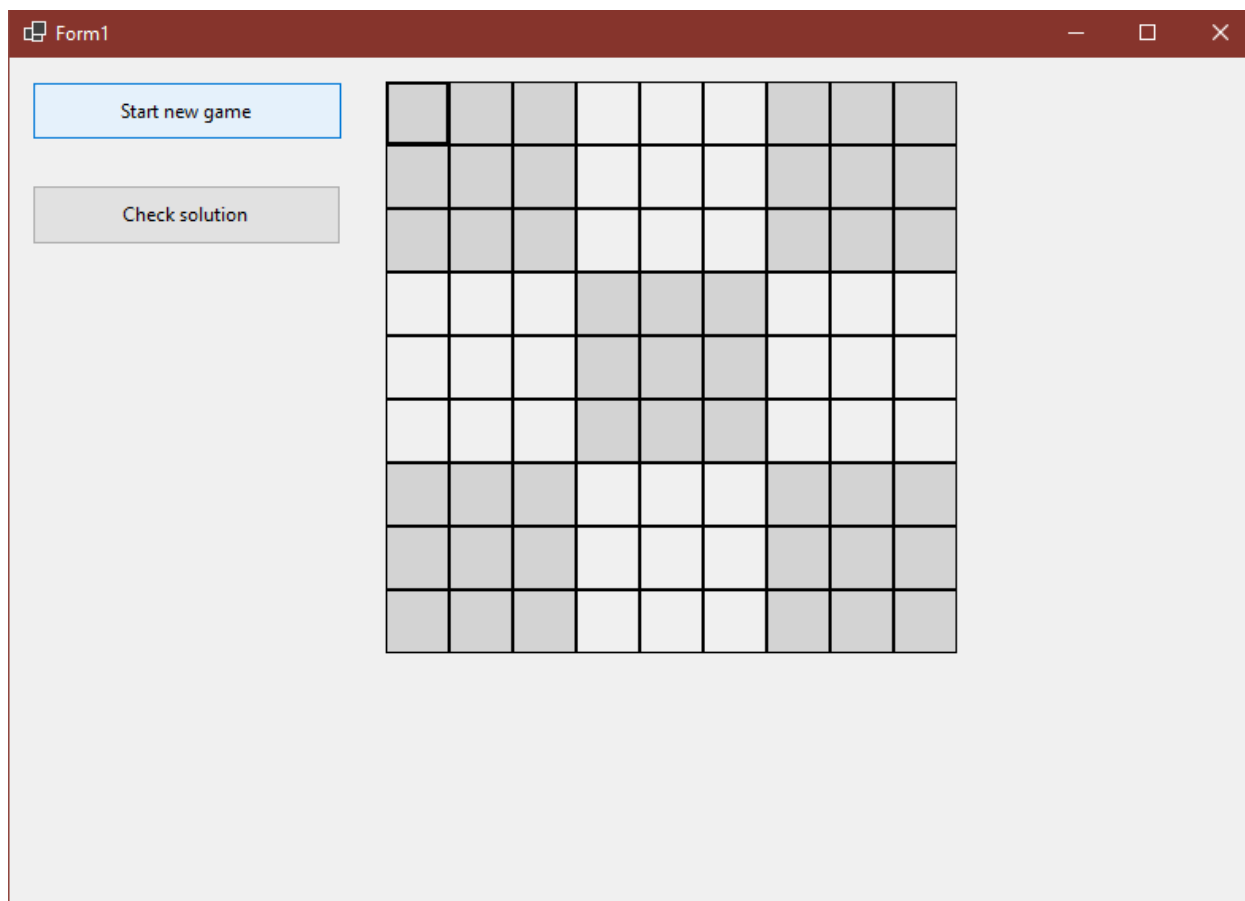
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 2 | 3 | 8 | 9 | 6 | 4 | 2 | 7 |
|---|---|---|---|---|---|---|---|---|

| |
|---|
| 2 |
| 5 |
| 8 |
| 4 |
| 9 |
| 6 |
| 1 |
| 7 |
| 4 |

Exemple de linie ,coloană și pătrat completat incorect.

Capitolul II – Prezentarea aplicației

Proiectul, realizat în C# Visual Studio 2019, conține o pagină care conține două butoane: Start new game și Check solution. Start new game generează un nou joc de sudoku, iar Check solution verifică dacă numerele introduse sunt corecte.



La apăsarea butonului Start new game pentru prima oară se generează o nouă tabla de joc indiferent dacă jocul precedent este terminat sau nu.

Form1

Start new game

Check solution

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3 | 7 | | 9 | | 5 | 1 | | |
| | | 1 | | | | | | |
| 2 | 4 | 5 | | 8 | 1 | | | 9 |
| 1 | 5 | 7 | | 2 | 6 | 4 | 9 | |
| 4 | 3 | 2 | | | | | 6 | |
| | | 6 | 4 | | | | 2 | 1 |
| 8 | | 3 | | | | | 7 | |
| 5 | | | | | | | | 3 |
| 7 | 1 | 9 | | | | | | |

Pentru a se completa tabla de joc se apasă pătratul de completat și se introduce o cifra de la 1 la 9. La apăsarea cifrei 0 se șterge elementul introdus anterior .Nu se pot introduce alte valori înafară de cifre.

Pentru a se verifica soluția se apasă butonul Check solution. Dacă cifrele introduse sunt greșite sau dacă tabla nu este completată integral se va afișa mesajul *Incorrect*. În caz contrar se afișează mesajul *Correct*.

De asemenea, în cazul unei soluții greșite se vor marca cu roșu numerele incorecte.

Următoarele imagini arată cazurile posibile:

The screenshot shows a Windows application window titled 'Form1'. On the left, there are two buttons: 'Start new game' and 'Check solution'. The 'Check solution' button is highlighted with a blue border. To the right of the buttons is a 10x10 grid representing a Sudoku puzzle. The grid contains numbers in some cells, with the number '2' in the 6th row, 4th column highlighted in red. A small dialog box with a red border and a close button is overlaid on the grid, displaying the text 'Incorrect' and an 'OK' button. The grid data is as follows:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | 2 | 4 | | 5 | |
| 8 | 4 | 2 | 7 | 6 | | | | | 1 |
| | | | | | 3 | | 6 | | |
| | 8 | | 3 | | 6 | | | | |
| 2 | 3 | 4 | | 9 | | | | 7 | |
| | 6 | 2 | | 4 | 8 | | | | |
| 7 | 9 | 8 | 1 | | 4 | | | | |
| | | | 8 | 3 | | | | | |
| 4 | 5 | | 6 | | 7 | | | | 8 |

Soluție greșită

Form1

Start new game

Check solution

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3 | 4 | 1 | 5 | 9 | 8 | 6 | 2 | 7 |
| 2 | 7 | 8 | 6 | 3 | 1 | 9 | 5 | 4 |
| 9 | 6 | 5 | 4 | 2 | 7 | 1 | 3 | 8 |
| 5 | 9 | 4 | 3 | 7 | 2 | 8 | 6 | 1 |
| 1 | 3 | 6 | 8 | 4 | 5 | 7 | 9 | 2 |
| 8 | 2 | 7 | 1 | 6 | 9 | 3 | 4 | 5 |
| 6 | 1 | 2 | 9 | 8 | 4 | 5 | 7 | 3 |
| 4 | 8 | 3 | 7 | 5 | 6 | 2 | 1 | 9 |
| 7 | 5 | 9 | 2 | 1 | 3 | 4 | 8 | 6 |

Correct

OK

Soluție corectă

Capitolul III – Prezentarea codului sursă

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace sudoku._3
{
    public partial class Sudoku : Form
    {
        public Sudoku()
        {
            InitializeComponent();
            //creez celulele
            CreateCells();
            //creez jocul
            Start_NewGame();
        }

        class SudokuCell : Button
        {
            public int Value { get; set; }
            public bool used { get; set; }
            public int X { get; set; }
            public int Y { get; set; }
            public void Clear()
            {
                this.Text = string.Empty;
                this.used = false;
            }
        }

        SudokuCell[,] cells = new SudokuCell[9, 9];
        //creez celulele
        void CreateCells()
        {
            for (int i = 0; i < 9; i++)
```

```

        {
            for (int j = 0; j < 9; j++)
            {
                // creez si colorez in functie de
                pozitie celulele
                cells[i, j] = new SudokuCell();
                cells[i, j].Font = new
                Font(SystemFonts.DefaultFont.FontFamily, 20);
                cells[i, j].Size = new Size(40, 40);
                cells[i, j].ForeColor =
                SystemColors.ControlDarkDark;
                cells[i, j].Location = new Point(i *
                40, j * 40);
                if((((i / 3) + (j / 3)) % 2 == 0)
                    cells[i, j].BackColor =
                Color.LightGray;
                cells[i, j].FlatStyle =
                FlatStyle.Flat;
                cells[i, j].FlatAppearance.BorderColor
                = Color.Black;
                cells[i, j].X = i;
                cells[i, j].Y = j;

                // celule apasate
                cells[i, j].KeyPress +=
                Cell_keyPressed;

                Panel1.Controls.Add(cells[i, j]);
            }
        }
    }
    private void Cell_keyPressed(object sender,
    KeyPressEventArgs e)
    {
        var cell = sender as SudokuCell;

        // verific daca e completat
        if (cell.used)
            return;

        int value;
    }

```

```

        // adaug in celula valoarea
        if (int.TryParse(e.KeyChar.ToString(), out
value))
        {
            // resetez daca valoare = 0
            if (value == 0)
                cell.Clear();
            else
                cell.Text = value.ToString();

            cell.ForeColor =
SystemColors.ControlDarkDark;
        }
    }

    void Start_NewGame()
    {
        // elimin valorile din celule
        foreach (var cell in cells)
        {
            cell.Value = 0;
            cell.Clear();
        }
        // apelez recursiv pana gasesc si completez cu
valori valide pentru
        //fiecare pozitie
        // Find(0, -1);
        // Reveal();
    }

    Random random = new Random();

    int Find(int i, int j)
    {
        // incrementez i si j pentru a ma deplasa pe
rand
        //daca j > 8 incrementez i pt a trece pe
urmatorul rand
        j++;
        if(j > 8)
        {
            j = 0;

```

```

        i++;
        // verifica daca se termina linia
        if (i > 8 )
            return 1;
    }
    var value = 0;
    var pos_values = new List<int> { 1, 2, 3, 4,
5, 6, 7, 8, 9 };
    do
    {
        //daca nu mai este nr disponibil
        //ne intoarcem pe pozitie anterioara si
incercam alt nr
        if (pos_values.Count < 1)
        {
            cells[i, j].Value = 0;
            return 0;
        }
        //alegem random un nr din cele ramase
        value = pos_values[random.Next(0,
pos_values.Count)];
        cells[i, j].Value = value;
        //stergem valoarea alesa din lista
        pos_values.Remove(value);
    }
    while (Valid(value, i, j) == 0 || Find(i, j)
== 0);
    // cells[i, j].Text = value.ToString();
    return 1;
}

int Valid(int value, int x, int y)
{
    for(int i = 0; i < 9; i++)
    {
        //verific pe verticala
        if (i != y && cells[x, i].Value == value)
            return 0;
        //verific pe orizontala
        if (i != x && cells[i, y].Value == value)
            return 0;
    }
}

```

```

        //verific patrutul
        for(int i = x - (x % 3); i < x - (x % 3) + 3;
i++)
        {
            for (int j = y - (y % 3); j < y - (y % 3)
+ 3; j++)
            {
                if (i != x && j != y && cells[i,
j].Value == value)
                    return 0;
            }
        }
        return 1;
    }
    //aleg random numerele la care dau reveal
    void Reveal()
    {
        for(int i = 0; i < 50; i++)
        {
            var x = random.Next(9);
            var y = random.Next(9);
            //afisez valoarea
            cells[x, y].Text = cells[x,
y].Value.ToString();
            //schimb culoarea
            cells[x, y].ForeColor = Color.Black;
            cells[x, y].used = true;
        }
    }

    private void Panel1_Paint(object sender,
PaintEventArgs e)
    {
    }

    private void newGameButton_Click(object sender,
EventArgs e)
    {
        Start_NewGame();
        Find(0, -1);
    }

```

```

        Reveal();
    }

    private void checkButton_Click(object sender,
EventArgs e)
    {
        var wrong = new List<SudokuCell>();
        //caut celulele gresite
        foreach (var cell in cells)
            if(cell.Value.ToString() != cell.Text)
                wrong.Add(cell);
        if (wrong.Any())
        {
            //marchez cu rosu nr gresite
            foreach (var k in wrong)
                k.ForeColor = Color.Red;
            MessageBox.Show("Incorrect");
        }
        else
            MessageBox.Show("Correct");
    }
}
}
}

```

Bibliografie

[1] <https://sudokus.ro/reguli/>

[2] <https://ro.wikipedia.org/wiki/Sudoku>

Cuprins

| | |
|--|----|
| Capitolul I – Jocul Sudoku, regulile jocului | 3 |
| Capitolul II – Prezentarea aplicației | 5 |
| Capitolul III – Prezentarea codului sursă..... | 9 |
| Bibliografie..... | 15 |
| Cuprins | 16 |