

1.Scopul lucrării

Acest proiect constituie jocul Connect 4. Am ales acest proiect deoarece acesta este un joc pe care îl jucam cu prietenii mei, care îți dezvoltă gândirea strategică într-un mod distractiv și recreațional, putând fi jucat de persoane de toate vârstele.

2.Tehnologii utilizate

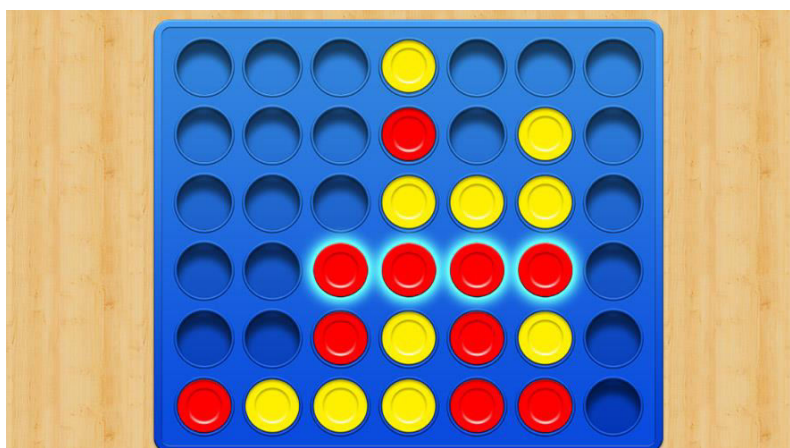
Pentru realizarea acestui proiect am folosit Microsoft Visual Studio 2017 și limbajul de programare C#.

3.Rezumat

Connect 4 este un joc cu 2 jucători. Acesta este format dintr-o tablă de joc suspendată vertical, având 7 coloane și 6 rânduri, 42 de discuri, respectiv, 21 de discuri roșii și 21 de discuri galbene. La început jucătorii își aleg o culoare, roșu sau galben și pe rând dau drumul unui disc de culoarea aleasă, în tabla de joc. Piesele cad în jos, ocupând cel mai mic spațiu disponibil în coloană.



Obiectivul jocului este acela de a fi primul care formează o linie orizontală, verticală sau diagonală de patru dintre propriile discuri, fiind declarat câștigător. În cazul în care nu mai sunt spații libere în tabla de joc, iar niciun jucător nu a reușit să formeze o linie de 4 discuri, atunci se va considera remiză.



4.Algoritm

Pentru a desena tabla de joc am folosit 7 dreptunghiuri de culoare albastră, reprezentând coloanele, apoi am desenat 6 cercuri de culoare albă pe fiecare coloană, tabla de joc reprezentând o matrice cu 6 linii și 7 coloane, având inițial, toate elementele nule. Elementele matricei se modifică în funcție de discurile jucate, valoarea 1 reprezintă discurile jucătorului 1, având discuri roșii, iar valoarea 2 reprezintă discurile jucătorului 2, având discuri galbene.

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    e.Graphics.FillRectangle(Brushes.Blue, 30, 30, 450, 400);
    for(int i=0;i<6;i++)
    {
        for (int j = 0; j < 7; j++)
        {
            if (i == 0)
            {
                this.tabla[j] = new Rectangle(64 * j + 40, 40, 40, 365);
            }
            e.Graphics.FillEllipse(Brushes.White, 64 * j + 40, 64 * i + 40, 40, 40);
        }
    }
}
```

Am folosit un event de tip `MouseClicked` pentru desfășurarea jocului. Am folosit funcția `nrcoloana` care primește ca parametru `"Point mouse"` reprezentând coordonatele mouse-ului. Această funcție returnează -1 dacă mouse-ul se află în afara tablei de joc, sau un număr de la 1 la 7 reprezentând coloana pe care se află. Dacă această funcție returnează un număr pozitiv, se va afla rândul pe care discul se așează primi valoarea corespunzătoare, iar elementul matricei de pe linia și coloana determinată va fi jucătorului care a făcut mutarea respectivă.

```

private int nrcoloana(Point mouse)
{
    for(int i=0;i<this.tabla.Length;i++)
    {
        if(mouse.X>=this.tabla[i].X && mouse.Y>=this.tabla[i].Y)
        {
            if(mouse.X<this.tabla[i].X+this.tabla[i].Width && mouse.Y < this.tabla[i].Y
+ this.tabla[i].Height)
            {
                return i;
            }
        }
    }
    return -1;
}

```

După fiecare mutare se verifica dacă jucătorul care a făcut mutarea respectivă este câștigător, verificându-se toate cazurile posibile de victorie (se verifică dacă există 4 elemente egale, diferite de 0, așezate consecutiv vertical, orizontal sau oblic), în caz afirmativ se afișează ce jucător a câștigat, în caz negativ, jocul continuă cu mutarea celui alt jucător până se ajunge la o victorie sau o remiză.

```

private void Form1_MouseClick(object sender, MouseEventArgs e)
{
    int coloana = this.nrcoloana(e.Location);
    if(coloana!=-1)
    {
        int rand = nrrand(coloana);
        if(rand!=-1)
        {
            this.matrice[rand, coloana] = this.turn;
            if(this.turn==1)
            {

```

```

Graphics g = this.CreateGraphics();

g.FillEllipse(Brushes.Red, 64 * coloana + 40, 64 * rand + 40, 40, 40);
}

if(this.turn == 2)
{
    Graphics g = this.CreateGraphics();

    g.FillEllipse(Brushes.Gold, 64 * coloana + 40, 64 * rand + 40, 40, 40);
}

int winner = castigator(this.turn);

ct++;

if(winner!=-1)
{
    MessageBox.Show("PLAYER " + winner.ToString() + " WINS!");
    Application.Restart();
}

if(ct==42)
{
    MessageBox.Show("IT'S A TIE!");
    Application.Restart();
}

if (this.turn == 1)
{
    Graphics g = this.CreateGraphics();

    g.FillEllipse(Brushes.Gold, 600, 300, 70, 70);

    label1.Visible = false;

    label3.Visible = true;

    this.turn = 2;
}

else
{
    Graphics g = this.CreateGraphics();

```

```

        g.FillEllipse(Brushes.Red, 600, 300, 70, 70);

        label1.Visible = true;

        label3.Visible = false;

        this.turn = 1;
    }
}
}
}

private bool egalitate(int nr1, int nr2, int nr3, int nr4)
{
    if (nr1 == nr2 && nr2 == nr3 && nr3 == nr4 && nr1 != 0)
        return true;
    else
        return false;
}

private int castigator(int player)
{
    //vertical
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<7;j++)
        {
            if (egalitate(matrice[i, j], matrice[i + 1, j], matrice[i + 2, j], matrice[i + 3, j])
== true)
                return player;
        }
    }

    //orizantal
    for(int i=0;i<6;i++)
    {
        for(int j=0;j<4;j++)

```

```

        {
            if (egalitate(matrice[i, j], matrice[i, j+1], matrice[i, j+2], matrice[i, j+3]) ==
true)

                return player;

        }
    }

    //diagonala spre dreapta( / )
    for (int i = 0; i<3; i++)
    {
        for (int j = 3; j <7 ; j++)
        {
            if (egalitate(matrice[i, j], matrice[i+1, j - 1], matrice[i+2, j - 2], matrice[i+3, j
- 3]) == true)
            {
                return player;
            }
        }
    }

    //diagonala spre stanga( \ )
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            if (egalitate(matrice[i, j], matrice[i + 1, j + 1], matrice[i + 2, j + 2], matrice[i
+ 3, j + 3]) == true)
            {
                return player;
            }
        }
    }

    return -1;
}

}

}

```