Universitatea Tehnică a Moldovei
Facultatea Calculatoare Informatică şi Microelectronică

Raport

la disciplina:
MIDPS

Lucrarea de laborator Nr.6

Tema: *Lucru in echipa. Aplicatie complexa*

A efectuat:	st.gr.Ti-141 Coban Tudor
A verificat	lect. univ. Cojanu Irina

Chişinău 2016

Scopul lucrării: Crearea unei aplicatii complexe in echipa.

Obiectivele lucrării:

- Crearea unei aplicatii complexe in echipa.
- Divizarea sarcinilor pe membrii echipei

IDE – Visual Studio
Limbaj de dezvoltare – C#

Sarcina personala:

- Crearea repozitoriului , fisierului .gitignore , incarcarea schemei , crearea branchului propriu.
- Crearea structurei aplicatiei
- Dezvoltarea Modulelor Menu si Game
- Merge la branchul personal(dev1) in master

Entry-point
Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace _1010alfa
{
    static class Program
    {

        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Menu());
        }
    }

}
```

Modulul Menu

Menu.cs

```
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;
using System.Resources;
using System.Runtime.InteropServices;

namespace _1010alfa
{
    public partial class Menu : Form
    {
        public Menu()
        {
            InitializeComponent();
            SetCursor();
        }

        Game f = new Game();

        private void Menu_Load(object sender, EventArgs e)
        {
            panel1.Top = panel1.Height * -1;
            System.IO.StreamWriter sr1 = new System.IO.StreamWriter(System.IO.File.Open("name.txt",
System.IO.FileMode.Create, System.IO.FileAccess.Write));
            {
                sr1.WriteLine(textBox1.Text);
                sr1.Close();
            }

            System.IO.StreamReader sr = new System.IO.StreamReader(System.IO.File.Open("score.txt",
System.IO.FileMode.Open, System.IO.FileAccess.Read));
            label1.Text = sr.ReadLine();
            sr.Close();

            listBox1.Items.Clear();
            System.IO.StreamReader sr3 = new System.IO.StreamReader(System.IO.File.Open("champions.txt",
System.IO.FileMode.Open));
            {
                string s = sr3.ReadLine();
                while (s != null)
                {
                    listBox1.Items.Add(s);
                    s = sr3.ReadLine();
                }
                sr3.Close();
            }
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {
            this.Hide();
            f.Show();
        }

        private void pictureBox4_Click(object sender, EventArgs e)
        {
            if (f.voice == true)
            {
                f.voice = false;
                pictureBox4.BackgroundImage = Image.FromFile(Application.StartupPath + 
@"\\images\\voise_off.png");
            }
            else
            {
                f.voice = true;
                pictureBox4.BackgroundImage = Image.FromFile(Application.StartupPath + 
@"\\images\\voise_on.png");
            }
        }

        private void pictureBox5_Click(object sender, EventArgs e)
        {
            if (f.day_night == true)
            {
                f.day_night = false;
                this.BackColor = Color.White;
                label1.ForeColor = Color.SeaGreen;
            }
            else
            {
                label1.ForeColor = Color.White;
                f.day_night = true;
                this.BackColor = Color.Gray;
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            timer1.Start();
        }

        bool a = true;

        private void timer1_Tick(object sender, EventArgs e)
        {
            panel1.Top = panel1.Top + 20;
            if (panel1.Top >= 100 && a == true)
            {
                timer1.Stop();
                a = false;
            }
            if (panel1.Top >= this.Height)
            {
                timer1.Stop();
                a = true;
                panel1.Top = panel1.Height * -1;
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            timer1.Start();
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            System.IO.StreamWriter sr1 = new System.IO.StreamWriter(System.IO.File.Open("name.txt", 
System.IO.FileMode.Create, System.IO.FileAccess.Write));
            {
                sr1.WriteLine(textBox1.Text);
                sr1.Close();
            }
        }

        [DllImport("User32.dll")]
        private static extern IntPtr LoadCursorFromFile(String str);
        IntPtr hCursor;

        public void SetCursor()
        {
            hCursor = LoadCursorFromFile(Application.StartupPath + "\\amaya-arrow.cur");
            this.Cursor = new Cursor(hCursor);
            pictureBox3.Cursor = new Cursor(hCursor);
            label1.Cursor = new Cursor(hCursor);
            button2.Cursor = new Cursor(hCursor);
            button1.Cursor = new Cursor(hCursor);
            pictureBox1.Cursor = new Cursor(hCursor);
            pictureBox4.Cursor = new Cursor(hCursor);
            pictureBox5.Cursor = new Cursor(hCursor);
        }



        private void panel1_Paint(object sender, PaintEventArgs e)
        {

        }
    }
}
```

Modulul Game

Game.cs

```
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;
using System.Resources;
using System.Runtime.InteropServices;

namespace _1010alfa
{
    public partial class Game : Form
    {
        public Boolean voice;
        public Boolean day_night;


        PictureBox[,] picturePanel1 = new PictureBox[5, 5],
                      picturePanel2 = new PictureBox[5, 5],
                      picturePanel3 = new PictureBox[5, 5];


        int[,] table = new int[10, 10];


        const int a = 10;
        PictureBox[,] myBoxes = new PictureBox[a, a];
        PictureBox[,] myBoxes1 = new PictureBox[a, a];

        piece k = new piece(),
                  k1 = new piece(),
                  k2 = new piece();

        String index;

        int Score = 0, HighScore;

        Boolean mouse_down = false, isin = false;
        PictureBox name;

        String index_piece;
        int v = 0;

        Sound sound = new Sound();

        int complexity = 6;

        public Game()
        {
            InitializeComponent();
            SetCursor();
            k.get_piece();
            make_piece(k.color, k.form, picturePanel1, 20, 500, "1");
            k1.get_piece();
            make_piece(k1.color, k1.form, picturePanel2, 194, 500, "2");
            k2.get_piece();
            make_piece(k2.color, k2.form, picturePanel3, 380, 500, "3");
            k.complexity = complexity;
            k1.complexity = complexity;
            k2.complexity = complexity;
        }

        String name_player;

        private void Form1_Load(object sender, EventArgs e)
        {
            complexity = 6;

            System.IO.StreamReader sr3 = new System.IO.StreamReader(System.IO.File.Open("name.txt", 
System.IO.FileMode.Open));
            {
                name_player = sr3.ReadLine();
                sr3.Close();
            }

            if (day_night == true)
            {
                this.BackColor = Color.Gray;
                label1.ForeColor = Color.White;
                label2.ForeColor = Color.White;
            }
            else
            {
                this.BackColor = Color.White;
                label1.ForeColor = Color.SeaGreen;
                label2.ForeColor = Color.SeaGreen;
            }



            System.IO.StreamReader sr = new System.IO.StreamReader(System.IO.File.Open("score.txt", 
System.IO.FileMode.Open));
            {
                HighScore = Convert.ToInt16(sr.ReadLine());
                label1.Text = HighScore.ToString();
                sr.Close();
            }
            for (int i = 0; i < 10; i++)
            {
                for (int j = 0; j < 10; j++)
                {
                    table[i, j] = 0;
                }
            }


            int x = 75, y = 100;
            for (int i = 0; i < a; i++)
            {
                for (int j = 0; j < a; j++)
                {
                    myBoxes1[i, j] = new PictureBox();
                    myBoxes1[i, j].Size = new Size(38, 38);
                    myBoxes1[i, j].Name = i.ToString() + j.ToString();
                    myBoxes1[i, j].BackColor = Color.Transparent;
                    myBoxes1[i, j].BackgroundImage = Image.FromFile(Application.StartupPath +
@"\images\0.png");
                    myBoxes1[i, j].BackgroundImageLayout = ImageLayout.Center;
                    myBoxes1[i, j].Location = new Point(x, y);

                    myBoxes1[i, j].Visible = true;

                    myBoxes[i, j] = new PictureBox();
                    myBoxes[i, j].Size = new Size(38, 38);
                    myBoxes[i, j].Name = i.ToString() + j.ToString();
                    myBoxes[i, j].BackColor = Color.Transparent;
                    myBoxes[i, j].BackgroundImage = Image.FromFile(Application.StartupPath +
@"\images\0.png");
                    myBoxes[i, j].BackgroundImageLayout = ImageLayout.Center;
                    myBoxes[i, j].Location = new Point(x, y);
                    x = x + 38;
                    myBoxes[i, j].Visible = true;

                    this.Controls.Add(myBoxes[i, j]);
                    this.Controls.Add(myBoxes1[i, j]);
                }
                x = 75;
                y = y + 38;
            }
            this.Width = myBoxes[0, 9].Left + myBoxes[0, 9].Width + 75;
            label2.Left = pictureBox3.Left - 20 - label2.Width;
            panel1.Left = this.Width / 2 - panel1.Width / 2;
            panel1.Top = panel1.Height * -1;
        }

        private void make_piece(Bitmap btm, string[] uzor, PictureBox[,] picturePanel, int x0, int y,
String Name)
        {
            int x = x0;
            for (int i = 0; i < 5; i++)
            {
                for (int j = 0; j < 5; j++)
                {
                    picturePanel[i, j] = new PictureBox();

                    picturePanel[i, j].Size = new Size(22, 22);
                    picturePanel[i, j].Name = Name + i.ToString() + j.ToString();
                    picturePanel[i, j].BackgroundImageLayout = ImageLayout.Stretch;
                    picturePanel[i, j].Location = new Point(x, y);
                    picturePanel[i, j].Visible = true;
                    picturePanel[i, j].Cursor = new Cursor(hCursor); ;
                    picturePanel[i, j].BackColor = Color.Transparent;

                    this.Controls.Add(picturePanel[i, j]);

                    if (uzor[i][j] == '1')
                    {
                        picturePanel[i, j].BackgroundImage = btm;
                    }
                    else
                    {
                        picturePanel[i, j].BackgroundImage = null;
                        picturePanel[i, j].Visible = false;
                    }

                    picturePanel[i, j].MouseMove += new MouseEventHandler(this.mouse_move);
                    picturePanel[i, j].MouseDown += new MouseEventHandler(this.mouse_dw);
                    picturePanel[i, j].MouseUp += new MouseEventHandler(this.mouse_up);

                    x = x + 22 + 2;
                }
                x = x0;
                y = y + 22 + 2;
            }
        }

        private void mouse_move(object sender, MouseEventArgs e)
        {
            name = (PictureBox)sender;

            switch (name.Name[0])
            {
                case '1':
                    name = (PictureBox)picturePanel1[0, 0];
                    break;
                case '2':
                    name = (PictureBox)picturePanel2[0, 0];
                    break;
                case '3':
                    name = (PictureBox)picturePanel3[0, 0];
                    break;
            }

            if (mouse_down == true)
            {
                name.Location = new Point(e.X + name.Left - 20, e.Y + name.Top - 20);
                recalc(35, 3);
            }
        }

        private void mouse_dw(object sender, MouseEventArgs e)
        {
            PictureBox name = (PictureBox)sender;

            index_piece = name.Name;
            mouse_down = true;
            recalc(35, 3);
        }

        private void mouse_up(object sender, MouseEventArgs e)
        {
            mouse_down = false;


            if (MousePosition.X > 20 + this.Left && MousePosition.X < 20 + this.Left + 527
                && MousePosition.Y > 100 + this.Top && MousePosition.Y < 100 + this.Top + 527)
            {
```

```
                isin = true;
                PictureBox name = (PictureBox)sender;
                Boolean h = false;
                switch (name.Name[0])
                {
                    case '1':
                        h = check_piece(k);
                        break;
                    case '2':
                        h = check_piece(k1);
                        break;
                    case '3':
                        h = check_piece(k2);
                        break;
                }
                if (h == true)
                {
                    if (voice == true)
                    {
                        sound.play("1.wav");
                    }
                    switch (name.Name[0])
                    {
                        case '1':
                            piece_location(k);
                            break;
                        case '2':
                            piece_location(k1);
                            break;
                        case '3':
                            piece_location(k2);
                            break;
                    }
                }
                else
                {
                    isin = false;
                }
            }
            else
            {
                isin = false;
            }

            if (isin == false)
            {

                switch (name.Name[0])
                {
                    case '1':
                        name.Location = new Point(20, 500);
                        break;
                    case '2':
                        name.Location = new Point(194, 500);
                        break;
                    case '3':
                        name.Location = new Point(380, 500);
                        break;
                }
                recalc(22, 2);
            }
        }

        private void piece_location(piece piece)
        {
            for (int i = 0; i < 5; i++)
            {
                for (int j = 0; j < 5; j++)
                {
                    switch (name.Name[0])
                    {
                        case '1':
                            picturePanel1[i, j].Visible = false;
                            k.visible = false;
                            break;
                        case '2':
                            picturePanel2[i, j].Visible = false;
                            k1.visible = false;
                            break;
                        case '3':
                            picturePanel3[i, j].Visible = false;
                            k2.visible = false;
                            break;
                    }
                }
            }

            for (int i = index[0] - index_piece[1]; i < index[0] - index_piece[1] + 5; i++)
            {
                for (int j = index[1] - index_piece[2]; j < index[1] - index_piece[2] + 5; j++)
                {
                    if (piece.form[i - (index[0] - index_piece[1])][j - (index[1] - index_piece[2])] ==
'1')
                    {
                        table[i, j] = 1;
                        myBoxes[i, j].Visible = true;
                        myBoxes[i, j].BackgroundImage = piece.color;
                        Score++;
                    }
                }
            }
            label2.Text = Score.ToString();
            if (Score > HighScore)
            {
                HighScore = Score;
                label1.Text = HighScore.ToString();

                System.IO.StreamWriter sr = new System.IO.StreamWriter(System.IO.File.Open("score.txt",
System.IO.FileMode.Create, System.IO.FileAccess.Write));
                {
                    sr.WriteLine(HighScore.ToString());
                    sr.Close();
                }
                label2.Text = Score.ToString();
                label2.Left = pictureBox3.Left - 20 - label2.Width;
            }
            v++;
            if (v == 3)
            {
                v = 0;
                next_pieces();
            }

            delete_row();

            if (game_over() == true)
            {
                if (voice == true) sound.play("2.wav");


                int f = -1, k = 0;
                string line;
                string[] ss = new string[1];

                System.IO.StreamReader sr2 = new
System.IO.StreamReader(System.IO.File.Open("champions.txt", System.IO.FileMode.Open));
                {
                    line = sr2.ReadLine();
                    while (line != null)
                    {
                        ss[k] = line;
                        Array.Resize(ref ss, ss.Length + 1);
                        if (line.Split(' ')[0] == name_player) f = k;
                        k++;
                        line = sr2.ReadLine();
                    }
                    sr2.Close();
                }

                System.IO.StreamWriter sr1 = new
System.IO.StreamWriter(System.IO.File.Open("champions.txt", System.IO.FileMode.Create,
System.IO.FileAccess.Write));
                {
                    if (f != -1)
                    {
                        for (int i = 0; i < k; i++)
                        {
                            if (i == f && Score > Convert.ToInt16(ss[i].Split(' ')[1]))
                            {
                                sr1.WriteLine(ss[i].Split(' ')[0] + " " + Score);
                            }
                            else
                            {
                                sr1.WriteLine(ss[i]);
                            }
                        }
                    }

                    else
                    {
                        for (int i = 0; i < k; i++)
                        {
                            sr1.WriteLine(ss[i]);
                        }
                        sr1.WriteLine(name_player + " " + Score.ToString());
                    }
                    sr1.Close();
                }

                MessageBox.Show("Game over!");
                v = 0;
                complexity = 6;
                Score = 0;
                label2.Text = "0";

                next_pieces();

                for (int i = 0; i < a; i++)
                {
                    for (int j = 0; j < a; j++)
                    {
                        table[i, j] = 0;
                        myBoxes[i, j].BackgroundImage = Image.FromFile(Application.StartupPath +
@"\images\0.png");
                    }
                }
            }
        }
```

```
    private Boolean game_over()
    {
        for (int i = 0; i < 10; i++)
        {
          for (int j = 0; j < 10; j++)
          {
            if (k.visible == true && game_over_f(k, i, j) == true)
            {
              return false;
            }
            else if (k1.visible == true && game_over_f(k1, i, j) == true)
            {
              return false;
            }
            //if (game_over_f(k2, (i.ToString() + j.ToString())) == true && k2.visible ==
true)
            else if (k2.visible == true && game_over_f(k2, i, j) == true)
            {
              return false;
            }
          }
        }
        return true;
    }

    private Boolean game_over_f(piece piece, int index1, int index2)
    {

      try
      {

        for (int r = 0; r < 5; r++)
        {
          for (int h = 0; h < 5; h++)
          {
            if (piece.form[r][h] == '1' && table[index1 + r - 2, index2 + h - 2] == 1)
            {
              return false;
            }
          }
        }
      }
      catch
      {
        return false;
      }
      return true;
    }

    private void delete_row()
    {
      int count = 0, count1 = 0;

      for (int i = 0; i < 10; i++)
      {
        for (int j = 0; j < 10; j++)
        {
          if (table[i, j] == 1)
          {
            count++;
            if (count == 10)
            {
              c = i;
              ss = "row";
              Score = Score + 10;
```

```
                            complexity = complexity + 2;
                            label2.Text = Score.ToString();
                            label2.Left = pictureBox3.Left - 20 - label2.Width;
                            timer1.Start();
                        }
                    }

                    if (table[j, i] == 1)
                    {
                        count1++;
                        if (count1 == 10)
                        {
                            c = i;
                            ss = "column";
                            Score = Score + 10;
                            complexity = complexity + 2;
                            label2.Text = Score.ToString();
                            label2.Left = pictureBox3.Left - 20 - label2.Width;
                            timer1.Start();
                        }
                    }
                }
                count = 0;
                count1 = 0;
            }
            if (Score > HighScore)
            {
                HighScore = Score;
                label1.Text = HighScore.ToString();

                System.IO.StreamWriter sr = new System.IO.StreamWriter(System.IO.File.Open("score.txt",
System.IO.FileMode.Create, System.IO.FileAccess.Write));
                {
                    sr.WriteLine(HighScore.ToString());
                    sr.Close();
                }

                label2.Text = Score.ToString();
                label2.Left = pictureBox3.Left - 20 - label2.Width;
            }
        }

        private void del()
        {
            if (ss == "row")
            {
                for (int i = 0; i < 10; i++)
                {
                    myBoxes[c, i].BackgroundImage = Image.FromFile(Application.StartupPath +
@"\images\0.png");
                    myBoxes[c, i].BackgroundImageLayout = ImageLayout.Center;

                }
            }
            else
            {
                for (int i = 0; i < 10; i++)
                {
                    myBoxes[i, c].BackgroundImage = Image.FromFile(Application.StartupPath +
@"\images\0.png");
                    myBoxes[i, c].BackgroundImageLayout = ImageLayout.Center;

                }
            }
            int x = 75, y = 100;
            for (int i = 0; i < a; i++)
            {
                for (int j = 0; j < a; j++)
                {
                    myBoxes[i, j].Size = new Size(38, 38);
                    myBoxes[i, j].BackColor = Color.Transparent;
                    myBoxes[i, j].Location = new Point(x, y);
                    x = x + 38;
                    myBoxes[i, j].Visible = true;
                }
                x = 75;
                y = y + 38;
            }
            delete_row();
        }

        int c;
        String ss;

        private void timer1_Tick(object sender, EventArgs e)
        {
            if (ss == "row")
            {
                for (int i = 0; i < 10; i++)
                {
                    myBoxes[c, i].BackgroundImageLayout = ImageLayout.Stretch;
                    table[c, i] = 0;
                    myBoxes[c, i].Size = new Size(myBoxes[c, i].Width - 2, myBoxes[c, i].Height - 2);
                    myBoxes[c, i].Location = new Point(myBoxes[c, i].Location.X + 1, myBoxes[c,
i].Location.Y + 1);
                }
                if (myBoxes[c, 9].Width == 0)
                {
                    timer1.Stop();
                    del();
                }
            }
            else
            {
                for (int i = 0; i < 10; i++)
                {
                    myBoxes[i, c].BackgroundImageLayout = ImageLayout.Stretch;
                    table[i, c] = 0;
                    myBoxes[i, c].Size = new Size(myBoxes[i, c].Width - 2, myBoxes[i, c].Height - 2);
                    myBoxes[i, c].Location = new Point(myBoxes[i, c].Location.X + 1, myBoxes[i,
c].Location.Y + 1);
                }
                if (myBoxes[9, c].Width == 0)
                {
                    timer1.Stop();
                    del();
                }
            }
        }

        private Boolean check_piece(piece piece)
        {
            index = fit_index(MousePosition.X, MousePosition.Y);

            try
            {
                for (int i = index[0] - index_piece[1]; i < index[0] - index_piece[1] + 5; i++)
                {
                    for (int j = index[1] - index_piece[2]; j < index[1] - index_piece[2] + 5; j++)
                    {
                        if (piece.form[i - (index[0] - index_piece[1])][j - (index[1] - index_piece[2])]
== '1')
```

```
                        {
                            if (table[i, j] != 0)
                            {
                                return false;
                            }
                        }
                    }
                }
            }
            catch
            {
                return false;
            }
            return true;
        }

        private String fit_index(int x, int y)
        {

            for (int i = 0; i < 10; i++)
            {
                for (int j = 0; j < 10; j++)
                {
                    if (x > myBoxes[i, j].Location.X + this.Left && x < myBoxes[i, j].Location.X + 
this.Left + myBoxes[i, j].Width
                        && y > myBoxes[i, j].Location.Y + this.Top  && y < myBoxes[i, j].Location.Y + 
this.Top + myBoxes[i, j].Height )
                    {

                        return i.ToString() + j.ToString();
                    }
                }
            }
            return "";
        }

        private void recalc(int size, int otstup)
        {
            Point loc;
            loc = name.Location;
            for (int i = 0; i < 5; i++)
            {
                for (int j = 0; j < 5; j++)
                {
                    switch (name.Name[0])
                    {
                        case '1':
                            picturePanel1[i, j].Size = new Size(size, size);
                            picturePanel1[i, j].Location = loc;
                            break;
                        case '2':
                           picturePanel2[i, j].Size = new Size(size, size);
                           picturePanel2[i, j].Location = loc;
                            break;
                        case '3':
                            picturePanel3[i, j].Size = new Size(size, size);
                            picturePanel3[i, j].Location = loc;
                            break;
                    }
                    loc.X = loc.X + size + otstup;
                }
                loc.X = name.Location.X;
                loc.Y = loc.Y + size + otstup;
            }
        }
        private void next_pieces()
        {
            for (int i = 0; i < 5; i++)
            {
                for (int j = 0; j < 5; j++)
                {
                    picturePanel1[i, j].Visible = false;
                    picturePanel2[i, j].Visible = false;
                    picturePanel3[i, j].Visible = false;
                }
            }

            k.get_piece();
            make_piece(k.color, k.form, picturePanel1, 20, 500, "1");
            k1.get_piece();
            make_piece(k1.color, k1.form, picturePanel2, 194, 500, "2");
            k2.get_piece();
            make_piece(k2.color, k2.form, picturePanel3, 380, 500, "3");
            k.complexity = complexity;
            k1.complexity = complexity;
            k2.complexity = complexity;

            for (int i = 0; i < a; i++)
            {
                for (int j = 0; j < a; j++)
                {

                    this.Controls.Add(myBoxes[i, j]);
                    this.Controls.Add(myBoxes1[i, j]);
                }
            }
        }

        private void Form1_Paint(object sender, PaintEventArgs e)
        {
            Graphics g = e.Graphics;
            Pen pen = new Pen(Color.SeaGreen, 10);
            g.DrawRectangle(pen, 0, 0, this.Width, this.Height);
        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {

            int f = -1, k = 0;
            string line;
            string[] ss = new string[1];

            System.IO.StreamReader sr2 = new System.IO.StreamReader(System.IO.File.Open("champions.txt",
System.IO.FileMode.Open));
            {
                line = sr2.ReadLine();
                while (line != null)
                {
                    ss[k] = line;
                    Array.Resize(ref ss, ss.Length + 1);
                    if (line.Split(' ')[0] == name_player) f = k;
                    k++;
                    line = sr2.ReadLine();
                }
                sr2.Close();
            }

            System.IO.StreamWriter sr1 = new System.IO.StreamWriter(System.IO.File.Open("champions.txt",
System.IO.FileMode.Create, System.IO.FileAccess.Write));
            {
                if (f != -1)
                {
                    for (int i = 0; i < k; i++)
                    {
                        if (i == f && Score > Convert.ToInt16(ss[i].Split(' ')[1]))
                        {
                            sr1.WriteLine(ss[i].Split(' ')[0] + " " + Score);
                        }
                        else
                        {
                            sr1.WriteLine(ss[i]);
                        }
                    }
                }

                else
                {
                    for (int i = 0; i < k; i++)
                    {
                        sr1.WriteLine(ss[i]);
                    }
                    sr1.WriteLine(name_player + " " + Score.ToString());
                }
                sr1.Close();
            }

            Form f1 = new Menu();
            this.Hide();
            f1.Show();
        }

        private void pictureBox2_Click(object sender, EventArgs e)
        {
            v = 0;
            complexity = 6;
            Score = 0;
            label2.Text = Score.ToString();
            label2.Left = pictureBox3.Left - 20 - label2.Width;
            for (int i = 0; i < a; i++)
            {
                for (int j = 0; j < a; j++)
                {
                    table[i, j] = 0;
                    myBoxes[i, j].BackgroundImage = Image.FromFile(Application.StartupPath +
@"\images\0.png");
                }
            }
            next_pieces();
        }

        int go = 1;

        private void pictureBox5_Click(object sender, EventArgs e)
        {

            timer2.Start();
        }

        private void timer2_Tick(object sender, EventArgs e)
        {
            if (panel1.Top >= 240 && go == 1)
            {
                timer2.Stop();
                go = 2;
            }
            if (panel1.Top >= this.Height && go == 2)
            {
                timer2.Stop();
                panel1.Top = panel1.Height * -1 - 30;
                go = 1;
            }


            panel1.Top = panel1.Top + 30;
        }

        private void pictureBox4_Click(object sender, EventArgs e)
        {
            timer2.Start();
        }

        [DllImport("User32.dll")]
        private static extern IntPtr LoadCursorFromFile(String str);
        IntPtr hCursor;

        public void SetCursor()
        {
            hCursor = LoadCursorFromFile(Application.StartupPath + "\\amaya-arrow.cur");
            this.Cursor = new Cursor(hCursor);
            pictureBox3.Cursor = new Cursor(hCursor);
            pictureBox5.Cursor = new Cursor(hCursor);
            pictureBox1.Cursor = new Cursor(hCursor);
            pictureBox2.Cursor = new Cursor(hCursor);
            pictureBox4.Cursor = new Cursor(hCursor);
            hCursor = LoadCursorFromFile(Application.StartupPath + "\\help.cur");
        }
    }
}
```

Fisierele adaugatoare

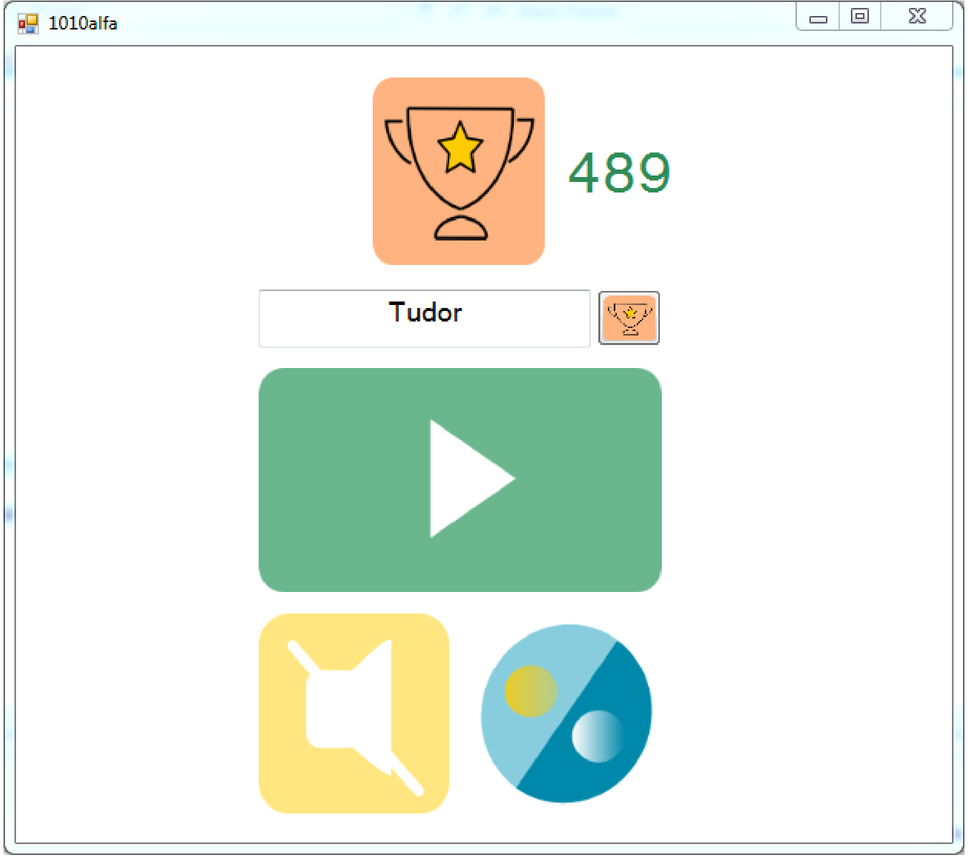
Piece.cs

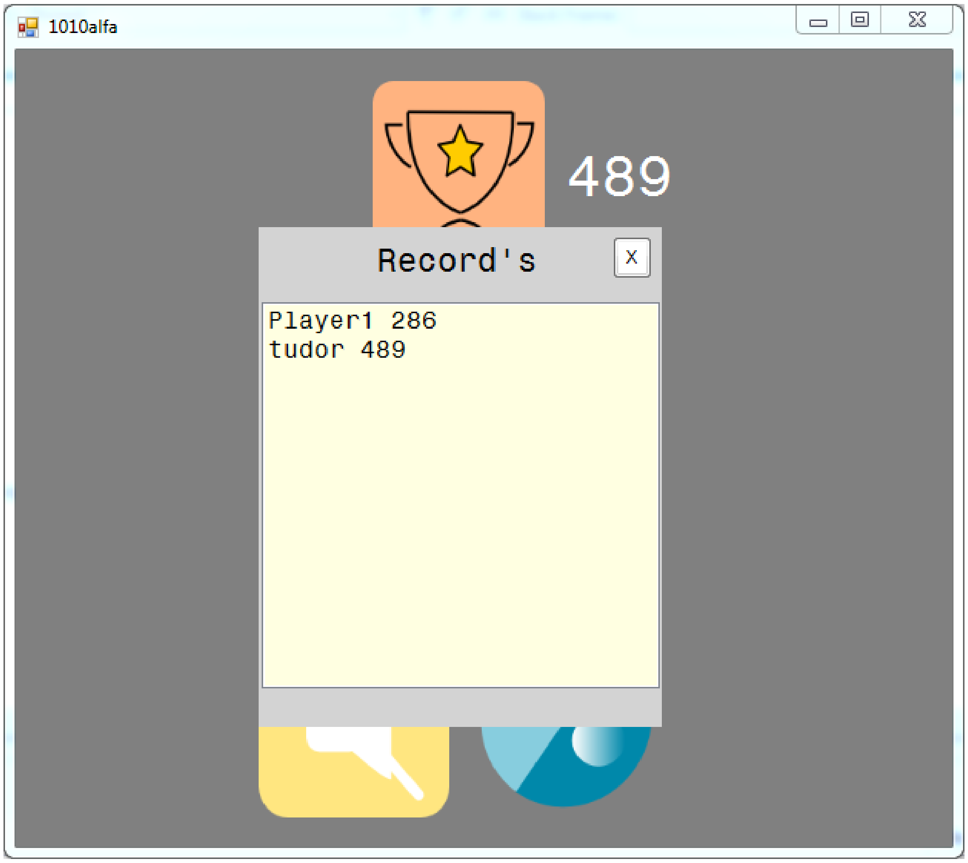
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Drawing;
using System.Windows.Forms;

namespace _1010alfa
{
    class piece
    {

        public string[] form;
        public Bitmap color;
        public Boolean visible;
        public int complexity = 6;

        string[] form1 = new string[5]
                                   {"00000",
                                    "00000",
                                    "00100",
                                    "00000",
                                    "00000"};

        string[] form2 = new string[5]
                                   {"00000",
                                    "00000",
                                    "00110",
                                    "00000",
                                    "00000"};

        string[] form3 = new string[5]
                                   {"00000",
                                    "00100",
                                    "00100",
                                    "00000",
                                    "00000"};

        string[] form4 = new string[5]
                                   {"00000",
                                    "00100",
                                    "00100",
                                    "00100",
                                    "00000"};
        string[] form5 = new string[5]
                                   {"00000",
                                    "00000",
                                    "01110",
                                    "00000",
                                    "00000"};

        string[] form6 = new string[5]
                                   {"00100",
                                    "00100",
                                    "00100",
                                    "00100",
                                    "00000"};
        string[] form7 = new string[5]
                                   {"00000",
                                    "00000",
                                    "11110",
                                    "00000",
                                    "00000"};

        string[] form8 = new string[5]
                                   {"00100",
                                    "00100",
                                    "00100",
                                    "00100",
                                    "00100"};
        string[] form9 = new string[5]
                                   {"00000",
                                    "00000",
                                    "11111",
                                    "00000",
                                    "00000"};

        string[] form10 = new string[5]
                                   {"00000",
                                    "01000",
                                    "01100",
                                    "00000",
                                    "00000"};
        string[] form11 = new string[5]
                                   {"00000",
                                    "01100",
                                    "00100",
                                    "00000",
                                    "00000"};

        string[] form12 = new string[5]
                                   {"00000",
                                    "01100",
                                    "01100",
                                    "00000",
                                    "00000"};

        string[] form13 = new string[5]
                                   {"01000",
                                    "01000",
                                    "01110",
                                    "00000",
                                    "00000"};
        string[] form14 = new string[5]
                                   {"00000",
                                    "00000",
                                    "01110",
                                    "01000",
                                    "01000"};
        string[] form15 = new string[5]
                                   {"00000",
                                    "00000",
                                    "01110",
                                    "00010",
                                    "00010"};
        string[] form16 = new string[5]
                                   {"00000",
                                    "00010",
                                    "00010",
                                    "01110",
                                    "00000"};


        string[] form17 = new string[5]
                                   {"00000",
                                    "00100",
                                    "01110",
                                    "00000",
                                    "00000"};
        string[] form18 = new string[5]
                                   {"00000",
                                    "00000",
                                    "01110",
                                    "00100",
                                    "00000"};
        string[] form19 = new string[5]
                                   {"00000",
                                    "00100",
                                    "01110",
                                    "00100",
                                    "00000"};

        string[] form20 = new string[5]
                                   {"00000",
                                    "01110",
                                    "01110",
                                    "01110",
                                    "00000"};

        string[] form21 = new string[5]
                                   {"00000",
                                    "01000",
                                    "00100",
                                    "00010",
                                    "00000"};
        string[] form22 = new string[5]
                                   {"00000",
                                    "00010",
                                    "00100",
                                    "01000",
                                    "00000"};
        public void get_piece()
        {
            visible = true;
            int a;
            Random rand = new Random();
            if (complexity >= 22)
            {
                a = rand.Next(1, 22);
            }
            else
            {
                a = rand.Next(1, complexity);
            }

            switch (a)
            {
                case 1:
                    form = form1;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\1.png"));
                    break;
                case 2:
                    form = form2;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\2.png"));
                    break;
                case 3:
                    form = form3;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\2.png"));
                    break;
                case 4:
                    form = form4;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\3.png"));
                    break;
                case 5:
                    form = form5;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\3.png"));
                    break;
                case 6:
                    form = form6;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\4.png"));
                    break;
                case 7:
                    form = form7;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\4.png"));
                    break;
                case 8:
                    form = form8;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\5.png"));
                    break;
                case 9:
                    form = form9;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\5.png"));
                    break;
                case 10:
                    form = form10;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\3angel.png"));
                    break;
                case 11:
                    form = form10;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\3angel.png"));
                    break;
                case 12:
                    form = form11;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\4rec.png"));
                    break;
                case 13:
                    form = form12;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\5angel.png"));
                    break;
                case 14:
                    form = form13;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\5angel.png"));
                    break;
                case 15:
                    form = form14;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\5angel.png"));
                    break;
                case 16:
                    form = form15;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\5angel.png"));
                    break;
                case 17:
                    form = form16;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\d1.png"));
                    break;
                case 18:
                    form = form17;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\d1.png"));
                    break;
                case 19:
                    form = form18;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\d1.png"));
                    break;
                case 20:
                    form = form19;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\4rec.png"));
                    break;
                case 21:
                    form = form20;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\d2.png"));
                    break;
                case 22:
                    form = form21;
                    color = new Bitmap(Image.FromFile(Application.StartupPath + @"\images\d2.png"));
                    break;
            }
        }
    }
}
```

Sound.cs

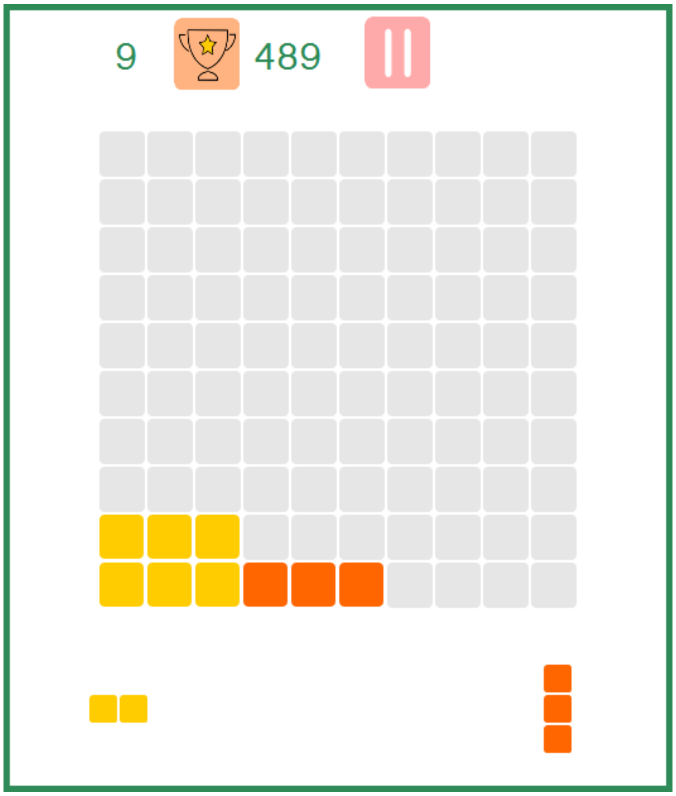
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Media;
using System.Windows.Forms;

namespace _1010alfa
{
    class Sound
    {
        public void play(String name)
        {
            MediaPlayer player = new MediaPlayer();

            player.Open(new Uri(Application.StartupPath + "\\Sound\\" +  name, UriKind.Absolute));
            player.Play();

        }
    }
}
```

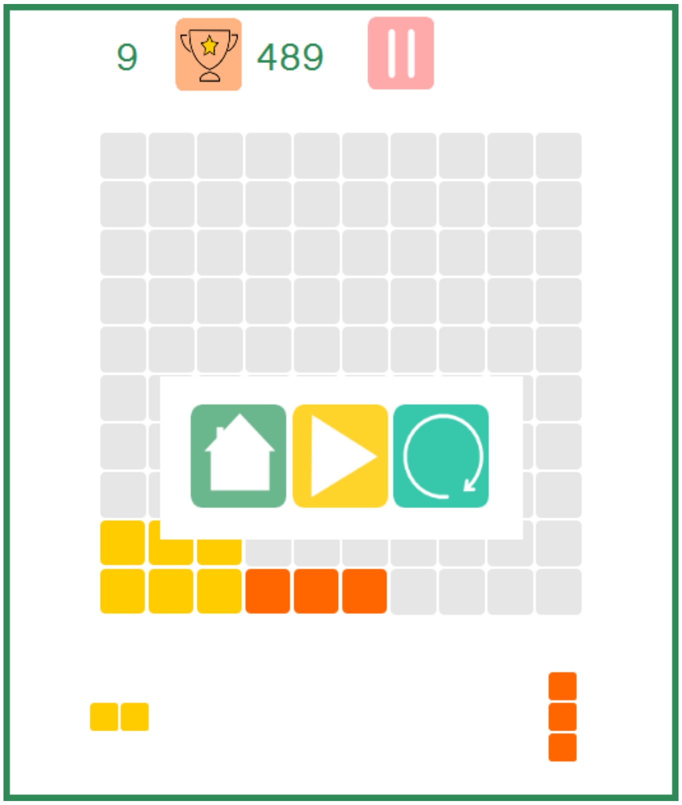
Rularea aplicatiei





```
Player1 286
tudor 489
```





Concluzii

In urma efectuarii lucrarii de laborator Nr.6 la MIDPS am obtinut capacitati practice de creare a aplicatiilor grafice in mediul de dezvoltare Visual Studio , lucrului in echipa asupra proiectului , si dezvoltarea lui pe Github . In urma acestei lucrari de laborator am obtinut cunostinte practice asupra crearii unei aplicatii grafice in Visual Studio pentru sistemul Windows , si am deprins lucrul cu elemente grafice si implementarea evenimentelor.