

```

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 Demo6
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        /* n - mängulaua mõõde; valge - musta poolt võetud valged nupud;
        * must - valge poolt võetud mustad nupud;
        * lisa - mitmiklöömise lisamine, kui lisa = 1, siis on võimalik
        selle nupuga veel süüa
        * xtest - x koordinaat, mille kaudu saab edasi- ja tagasilöömise
        võimaluse korral kindlaks teha, kumb löömine sooritati
        * värv - mängunupu värv (v - valge, m - must, b - tühi väli); p1 -
        lisamuutuja p.Name meeldejätmiseks; p2 ja p3 - edasi- ja
        * tagasilöömise variantide x, y ja värvi meeldejätmiseks; variant11
        kuni varianti 22 - käigu sooritamisel teiste pakutud
        * variantide eemaldamine
        */
        int n, valge = 0, must = 0, lisa = 0, xtest;
        PictureBox[,] P;
        string värv = "v", p1 = "", p2 = "", p3 = "", variant11 = "",
        variant12 = "", variant21 = "", variant22 = "";

        //Link huvitavatele kombinatsioonidele
        private void linkLabel2_LinkClicked(object sender,
        LinkLabelLinkClickedEventArgs e)
        {
            this.linkLabel1.LinkVisited = true;
            System.Diagnostics.Process.Start("https://www.youtube.com/watch?v
            =titCc4zkhPo");
        }
        // Võimalus mängida internetis kabet
        private void linkLabel3_LinkClicked(object sender,
        LinkLabelLinkClickedEventArgs e)
        {
            this.linkLabel1.LinkVisited = true;
            System.Diagnostics.Process.Start("https://www.vint.ee/et-
            ee/games/#kabe");
        }
        // Kabekoodeks
        private void linkLabel1_LinkClicked(object sender,
        LinkLabelLinkClickedEventArgs e)
        {
            this.linkLabel1.LinkVisited = true;

```

```

        System.Diagnostics.Process.Start("http://kabeliit.ee/kohtunikekog
        u/koodeks/EKL64_2016.pdf");
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        // Mängulaua kujundamine
        n = 8;
        P = new PictureBox[n, n];
        int vasakult = 2, ülevalt = 2, kontroll = 1;
        Color[] colors = new Color[] { Color.Wheat, Color.Sienna };
        for (int i = 0; i < n; i++)
        {
            vasakult = 2;
            if (i % 2 == 0)
            {
                colors[0] = Color.Sienna;
                colors[1] = Color.Wheat;
            }
            else
            {
                colors[0] = Color.Wheat;
                colors[1] = Color.Sienna;
            }
            for (int j = 0; j < n; j++)
            {
                P[i, j] = new PictureBox();
                P[i, j].BackColor = colors[(j % 2 == 0) ? 1 : 0];
                P[i, j].Location = new Point(vasakult, ülevalt);
                P[i, j].Size = new Size(60, 60);
                vasakult += 60;
                /* Niiõelda kolmemõõtmelise massiivi tekitamine, et
                eristada lisaks koordinaatidele ka mängunupu värvi */
                P[i, j].Name = i + " " + j;
                // Mängunuppude lisamine mängulauale
                if (i < 3 && P[i, j].BackColor == Color.Sienna)
                {
                    P[i, j].Image = Properties.Resources.m;
                    P[i, j].Name += " m";
                }
                else if (i > 4 && P[i, j].BackColor == Color.Sienna)
                {
                    P[i, j].Image = Properties.Resources.v;
                    P[i, j].Name += " v";
                }
                P[i, j].SizeMode = PictureBoxSizeMode.CenterImage;
                /* Tuvastada kursori liikumine nupule ja tuua see
                mänguväli teistest esile */
                P[i, j].MouseHover += (sender2, e2) =>
                {
                    PictureBox p = sender2 as PictureBox;
                    if (p.Image != null)
                        p.BackColor = Color.FromArgb(255, 100, 50, 0);
                };
            }
        }
    }

```

```

/* Kursori ära liikumisel nupult muuta mänguväli teistega
sarnaseks */
P[i, j].MouseLeave += (sender2, e2) =>
{
    PictureBox p = sender2 as PictureBox;
    if (p.Image != null)
        p.BackColor = Color.Sienna;
};
/*Kabe kasutajaliidese osa, kui nupuga mänguväljale on
Vajutatud */
P[i, j].Click += (sender3, e3) =>
{
    PictureBox p = sender3 as PictureBox;
    if (p.Image != null)
    {
        /* t = -1, siis arvestatakse valge nupu
        reakoordinaat, t = 1 korral aga musta nupu
        reakoordinaat */
        int t = -1, x, y;
        EemaldabVariandid(); /*Eemaldab käigule pakutavad
        funktsioonid, kui kasutaja on vajutanud mingile
        teisele mängunupule*/
        /* Värv on b, kui sinna väljale, kas liiguti või
        Löödi */
        if (p.Name.Split(' ')[2] == "b")
        {
            //Mitmiklöömise korral
            if (lisa == 1)
            {
                if (värv == "v")
                {
                    värv = "v";
                }
                else if (värv == "m")
                {
                    värv = "m";
                }
            }
            //Tavakäik või ühekordne löömine
            else if (lisa == 0)
            {
                if (värv == "v")
                {
                    värv = "m";
                }
                else if (värv == "m")
                {
                    värv = "v";
                }
            }
        }
        x = Convert.ToInt32(p1.Split(' ')[0]);
        y = Convert.ToInt32(p1.Split(' ')[1]);
        variant11 = "";
        variant12 = "";
        variant21 = "";
        variant22 = "";
    }
}

```

```

//Kas käidav nupp oli must
if(p1.Split(' ')[2] == "m")
{
    //Kas must nupp liikus viimasele reale
    if (Convert.ToInt32(p.Name.Split(' ')[0])
    == 7)
    {
        //tamm
        p.Image = Properties.Resources.mtamm;
        p.Name = p.Name.Replace("b", "m");
        kontroll = 0;
    }
    /* Tavajuhul pärast musta käimist tühjale
    väljale vastavate asenduste tegemine */
    else
    {
        p.Image = Properties.Resources.m;
        p.Name = p.Name.Replace("b", "m");
        kontroll = 0;
    }
}
//Kas käidav nupp oli valge
else if (p1.Split(' ')[2] == "v")
{
    //Kas valge nupp liikus viimasele reale
    if (Convert.ToInt32(p.Name.Split(' ')[0])
    == 0)
    {
        p.Image = Properties.Resources.vtamm;
        p.Name = p.Name.Replace("b", "v");
        kontroll = 0;
    }
    /* Tavajuhul pärast valge käimist tühjale
    väljale vastavate asenduste tegemine */
    else
    {
        p.Image = Properties.Resources.v;
        p.Name = p.Name.Replace("b", "v");
        kontroll = 0;
    }
}
// Väli, kust nupp käidi, on nüüd tühi
P[x, y].Image = null;
/*Kontroll = 0 tähendab, et kuskil on söömine
ja kontroll = 1 puhul saab teha tavakäiku */
if(kontroll == 0)
{
    /* p2 ja p3 edasi- ja tagasilöömise
    variantide koordinaatide ja värvi
    meeldejäätmine, et mitte käidud variandid
    eemalda mängulaualt */
    if ((p2 != "") ^ (p3 != ""))
    {
        if (p2 != "")
        {
            if (p2.Split(' ')[2] == "m")
            {

```

```

        x = Convert.ToInt32
        (p2.Split(' ')[0]);
        y = Convert.ToInt32
        (p2.Split(' ')[1]);
        P[x, y].Image = null;
        must++;
    }
    else if (p2.Split(' ')[2] == "v")
    {
        x = Convert.ToInt32
        (p2.Split(' ')[0]);
        y = Convert.ToInt32
        (p2.Split(' ')[1]);
        P[x, y].Image = null;
        valge++;
    }
}
else if (p3 != "")
{
    {
        if (p3.Split(' ')[2] == "m")
        {
            x = Convert.ToInt32
            (p3.Split(' ')[0]);
            y = Convert.ToInt32
            (p3.Split(' ')[1]);
            P[x, y].Image = null;
            must++;
        }
        else if (p3.Split(' ')[2] ==
        "v")
        {
            x = Convert.ToInt32
            (p3.Split(' ')[0]);
            y = Convert.ToInt32
            (p3.Split(' ')[1]);
            P[x, y].Image = null;
            valge++;
        }
    }
}
label17.Text = valge + "";
label18.Text = must + "";
p2 = "";
p3 = "";
lisa = 0;
}
else if (p2 != "" && p3 != "")
{
    xtest = Convert.ToInt32
    (p.Name.Split(' ')[0]);
    if (p2.Split(' ')[2] == "v" &&
    p3.Split(' ')[2] == "v")
    {

```

```

        if (xtest - x == 2)
        {
            x = Convert.ToInt32
                (p2.Split(' ')[0]);
            y = Convert.ToInt32
                (p2.Split(' ')[1]);
            P[x, y].Image = null;
            valge++;
        }
        else if (x - xtest == 2)
        {
            x = Convert.ToInt32
                (p3.Split(' ')[0]);
            y = Convert.ToInt32
                (p3.Split(' ')[1]);
            P[x, y].Image = null;
            valge++;
        }
    }
    else if (p2.Split(' ')[2] == "m" &&
        p3.Split(' ')[2] == "m")
    {
        if (x - xtest == 2)
        {
            x = Convert.ToInt32
                (p2.Split(' ')[0]);
            y = Convert.ToInt32
                (p2.Split(' ')[1]);
            P[x, y].Image = null;
            must++;
        }
        else if (xtest - x == 2)
        {
            x = Convert.ToInt32
                (p3.Split(' ')[0]);
            y = Convert.ToInt32
                (p3.Split(' ')[1]);
            P[x, y].Image = null;
            must++;
        }
    }

    label17.Text = valge + "";
    label18.Text = must + "";
    p2 = "";
    p3 = "";
    lisa = 0;
}
kontroll = 1;
}

/* Esialgu algab programm siit, vajadusel saab
muuta kumb pool alustab */
else if (p.Name.Split(' ')[2] == värv)
{
    x = Convert.ToInt32(p.Name.Split(' ')[0]);
    y = Convert.ToInt32(p.Name.Split(' ')[1]);
    p1 = p.Name;

```

```

if (p.Name.Split(' ')[2] == "m")
{
    t = 1;
}
try
{
    /* Valge ja musta edasi söömise üks
    Variant */
    if (P[x + t, y + 1].Image != null &&
        P[x + t, y + 1].Name.Split(' ')[2] !=
        p.Name.Split(' ')[2] &&
        P[x + (t * 2), y + 2].Image == null)
    {
        kontroll = 0;
        P[x + (t * 2), y + 2].Image =
        Properties.Resources.b;
        P[x + (t * 2), y + 2].Name =
        (x + (t * 2)) + " " + (y + 2) + " b";
        variant11 = (x + (t * 2)) + " " + (y
        + 2);
        p2 = (x + t) + " " + (y + 1) + " " +
        P[x + t, y + 1].Name.Split(' ')[2];
        if ((P[x + (t * 3), y + 1].Image !=
        null && P[x + (t * 4), y].Image ==
        null) ||
            (P[x + (t * 3), y + 3].Image != null
            && P[x + (t * 4), y + 4].Image ==
            null) ||
            (P[x + t, y + 3].Image != null &&
            P[x, y + 4].Image == null))
        {
            lisa = 1;
        }
        else
        {
            lisa = 0;
        }
        kontroll = 0;
    }
    /* Kui söömist ei ole, siis saab teha
    Tavakäigu */
    else if (kontroll == 1 && P[x + t, y +
    1].Image == null)
    {
        P[x + t, y + 1].Image =
        Properties.Resources.b;
        P[x + t, y + 1].Name = (x + t) + " "
        + (y + 1) + " b";
        variant11 = (x + t) + " " + (y + 1);
    }
    /* Valge ja musta tagasi söömise üks
    Variant */
    if (P[x - t, y + 1].Image != null &&
        P[x - t, y + 1].Name.Split(' ')[2] !=
        p.Name.Split(' ')[2] && P[x - (t * 2), y
        + 2].Image == null)

```

```

{
    kontroll = 0;
    P[x - (t * 2), y + 2].Image =
    Properties.Resources.b;
    P[x - (t * 2), y + 2].Name = (x - (t
    * 2)) + " " + (y + 2) + " b";
    variant12 = (x - (t * 2)) + " " + (y
    + 2);
    p3 = (x - t) + " " + (y + 1) + " " +
    P[x - t, y + 1].Name.Split(' ')[2];
    if ((P[x - (t * 3), y + 1].Image !=
    null && P[x - (t * 4), y].Image ==
    null) ||
    (P[x - (t * 3), y + 3].Image != null
    && P[x - (t * 4), y + 4].Image ==
    null) ||
    (P[x - t, y + 3].Image != null &&
    P[x, y + 4].Image == null))
    {
        lisa = 1;
    }
    else
    {
        lisa = 0;
    }
    kontroll = 0;
}
/* Kui söömist ei ole, siis saab teha
Tavakäigu */
else if (kontroll == 1 && P[x + t, y +
1].Image == null)
{
    P[x + t, y + 1].Image =
    Properties.Resources.b;
    P[x + t, y + 1].Name = (x + t) + " "
    + (y + 1) + " b";
    variant11 = (x + t) + " " + (y + 1);
}
}
catch { }
try
{
    /* Valge ja musta edasi söömise teine
    Variant */
    if (P[x + t, y - 1].Image != null && P[x
    + t, y - 1].Name.Split(' ')[2] !=
    p.Name.Split(' ')[2] && P[x + (t * 2), y
    - 2].Image == null)
    {
        kontroll = 0;
        P[x + (t * 2), y - 2].Image =
        Properties.Resources.b;
        P[x + (t * 2), y - 2].Name = (x + (t
        * 2)) + " " + (y - 2) + " b";
        variant21 = (x + (t * 2)) + " " + (y
        - 2);
        p2 = (x + t) + " " + (y - 1) + " " +

```



```

P[x + t, y - 1].Name.Split(' ')[2];

if ((P[x + (t * 3), y - 1].Image !=
null && P[x + (t * 4), y].Image ==
null) ||
(P[x + (t * 3), y - 3].Image != null
&& P[x + (t * 4), y - 4].Image ==
null) ||
(P[x + t, y - 3].Image != null && P[x,
y - 4].Image == null))
{
    lisa = 1;
}
else
{
    lisa = 0;
}
kontroll = 0;
}
/* Kui söömist ei ole, siis saab teha
Tavakäigu */
else if (kontroll == 1 && P[x + t, y -
1].Image == null)
{
    P[x + t, y - 1].Image =
Properties.Resources.b;
P[x + t, y - 1].Name = (x + t) + " "
+ (y - 1) + " b";
variant21 = (x + t) + " " + (y - 1);
}
/* Valge ja musta tagasi söömise teine
Variant */
if (P[x - t, y - 1].Image != null && P[x
- t, y - 1].Name.Split(' ')[2] !=
p.Name.Split(' ')[2] && P[x - (t * 2), y
- 2].Image == null)
{
    kontroll = 0;
P[x - (t * 2), y - 2].Image =
Properties.Resources.b;
P[x - (t * 2), y - 2].Name = (x - (t
* 2)) + " " + (y - 2) + " b";
variant22 = (x - (t * 2)) + " " + (y
- 2);
p3 = (x - t) + " " + (y - 1) + " " +
P[x - t, y - 1].Name.Split(' ')[2];
if ((P[x - (t * 3), y - 1].Image !=
null && P[x - (t * 4), y].Image ==
null) ||
(P[x - (t * 3), y - 3].Image != null
&& P[x - (t * 4), y - 4].Image ==
null) ||
(P[x, y - 3].Image != null && P[x, y
- 4].Image == null))
{
    lisa = 1;
}
}

```

```

        else
        {
            lisa = 0;
        }
        kontroll = 0;
    }
    /* Kui söömist ei ole, siis saab teha
    Tavakäigu */
    else if (kontroll == 1 && P[x + t, y -
1].Image == null)
    {
        P[x + t, y - 1].Image =
        Properties.Resources.b;
        P[x + t, y - 1].Name = (x + t) + " "
        + (y - 1) + " b";
        variant21 = (x + t) + " " + (y - 1);
    }
    }
    catch { }
    }
    }
    };
    board.Controls.Add(P[i, j]);
}
ülevaalt += 60;
}
}
}

```

/* Eemaldab käigule pakutavad funktsioonid, kui kasutaja on vajutanud mingile teisele mängunupule */

```

public void EemaldabVariandid()
{
    if (variant11 != "")
    {
        int x, y;
        x = Convert.ToInt32(variant11.Split(' ')[0]);
        y = Convert.ToInt32(variant11.Split(' ')[1]);
        P[x, y].Image = null;
    }
    if (variant12 != "")
    {
        int x, y;
        x = Convert.ToInt32(variant12.Split(' ')[0]);
        y = Convert.ToInt32(variant12.Split(' ')[1]);
        P[x, y].Image = null;
    }
    if (variant21 != "")
    {
        int x, y;
        x = Convert.ToInt32(variant21.Split(' ')[0]);
        y = Convert.ToInt32(variant21.Split(' ')[1]);
        P[x, y].Image = null;
    }
    if (variant22 != "")
    {

```

```
        int x, y;  
        x = Convert.ToInt32(variant22.Split(' ')[0]);  
        y = Convert.ToInt32(variant22.Split(' ')[1]);  
        P[x, y].Image = null;  
    }  
}  
}
```