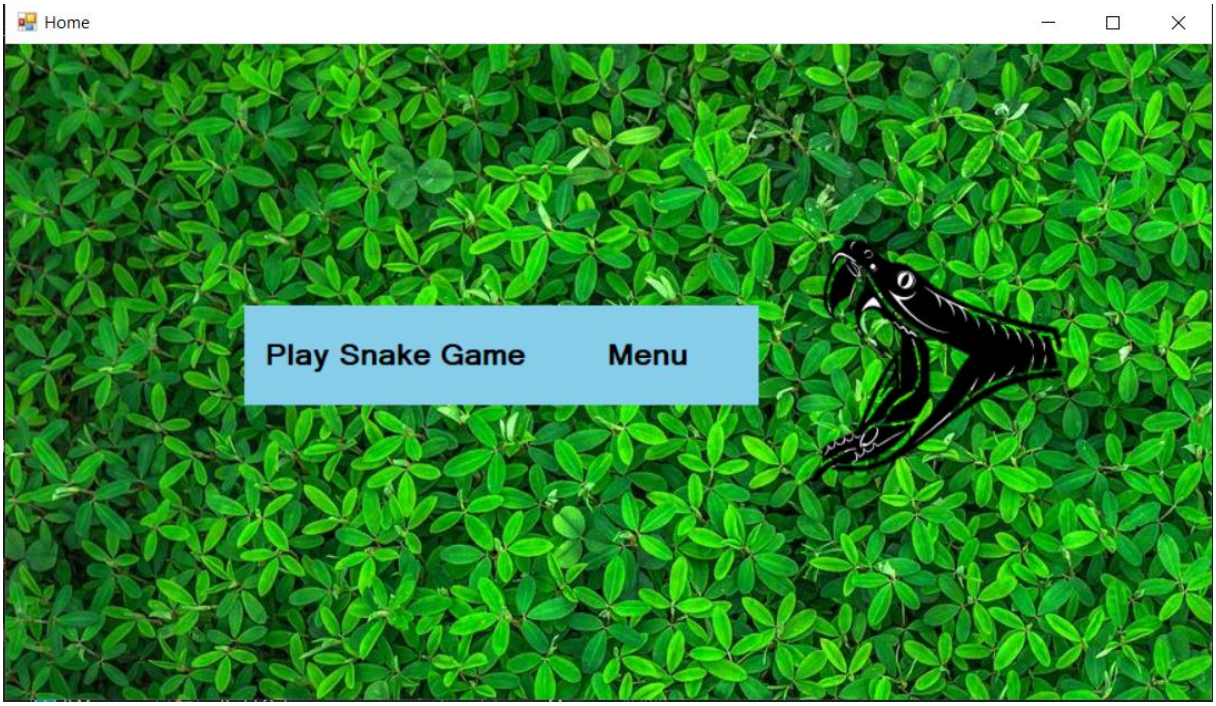
Igra Kača



Igralec uporablja tipke  za igranje igre kača. Na igralnem oknu se naključno prikazuje hrana. Ko kača hrano poje igralec dobi točko, kača pa se podaljša za dodaten člen. Igro spremljajo zvočni efekti in otežujejo naključno prikazane ovire. Prav tako, lahko igralec obarva kači trup v svoji najljubši barvi.

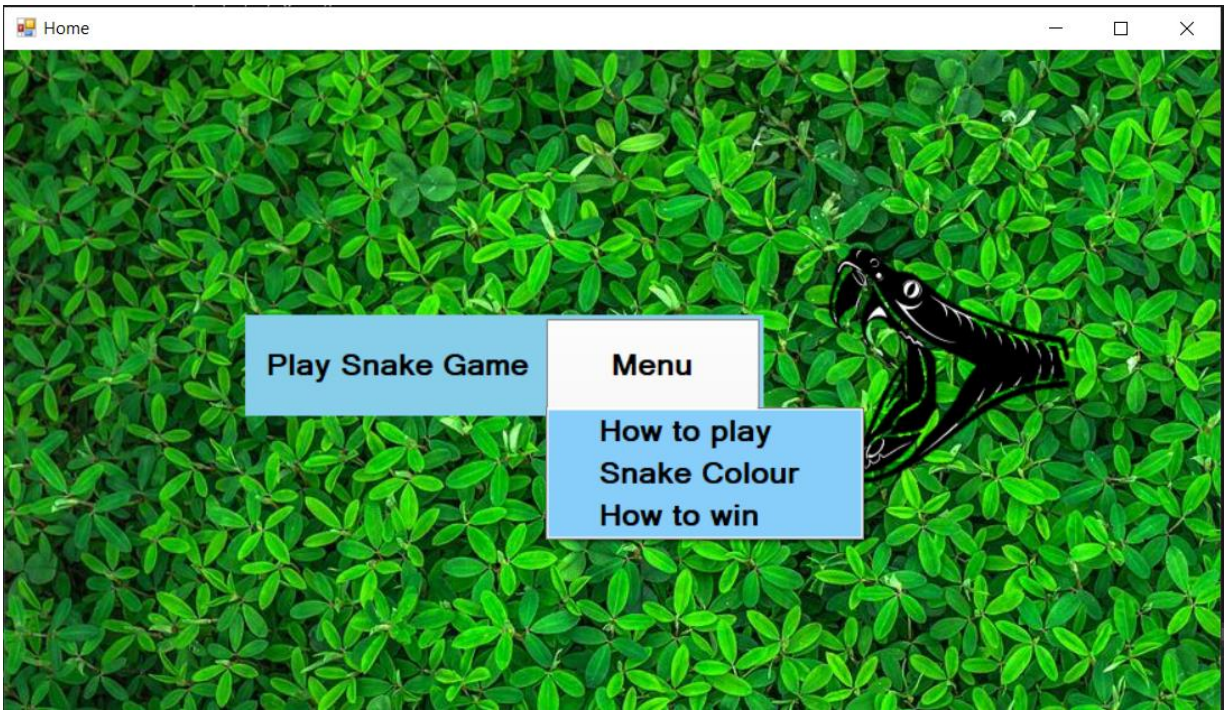
Igralec ustavi igro s klikom na gumb enter, nato se prikažeta gumba 'Start' in 'Go home'. S klikom na nadaljuje, s klikom na gumb 'Go home' pa odpre začetno okno.

OSNOVNO OKNO

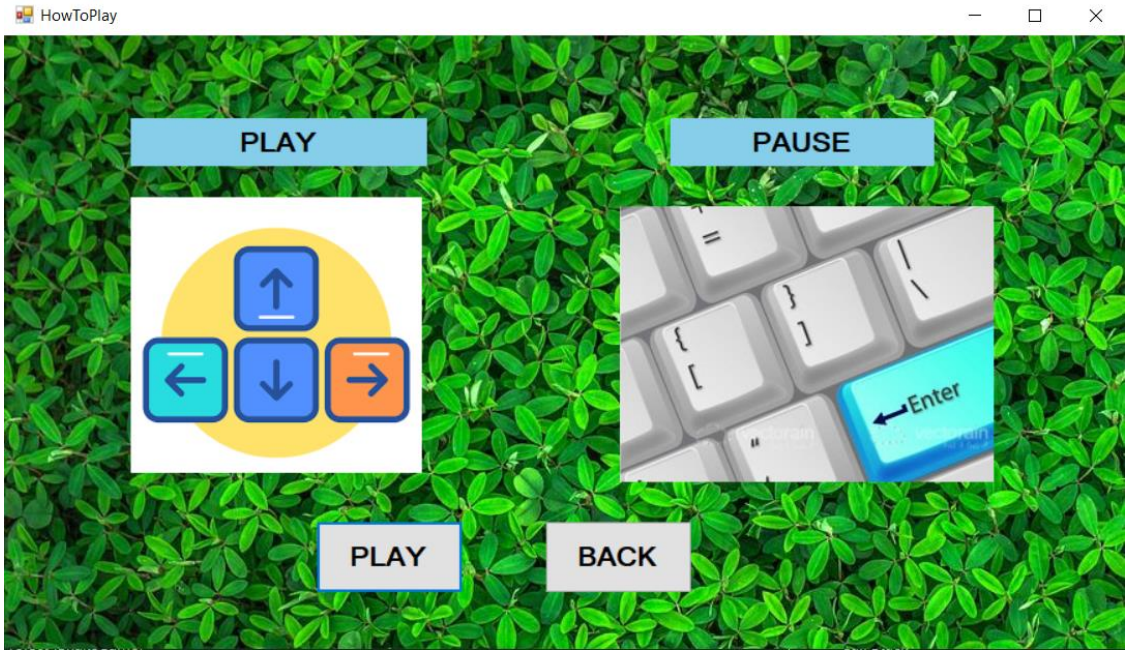
Pri zagonu igre se odpre osnovno okno, kjer nas pričakata dva glavna gumb.

S klikom na 'Play Snake Game' uporabnik začne igrati igro. V tem primeru je barva kače avtomatsko izbrana. Poleg takojšne igre pa uporabnik lahko pridobi še nekaj dodatnih informacij o samem poteku igre. S klikom na 'How to play' ali na gumb 'How to win' se odpre okno, kjer so razložena navodila za igro in opis poti do zmage.

S klikom na gumb 'Snake Colour' pa uporabniku ponudimo možnost obarvati kačo v svoji najljubši barvi.

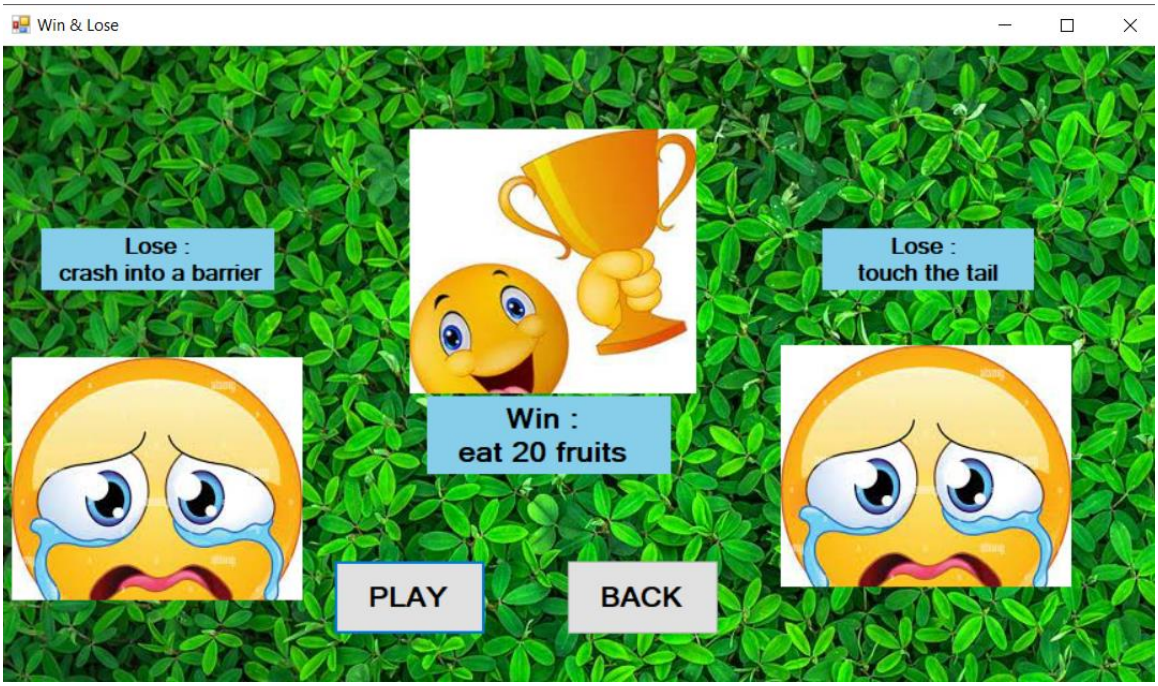


OKNO ' How to play '



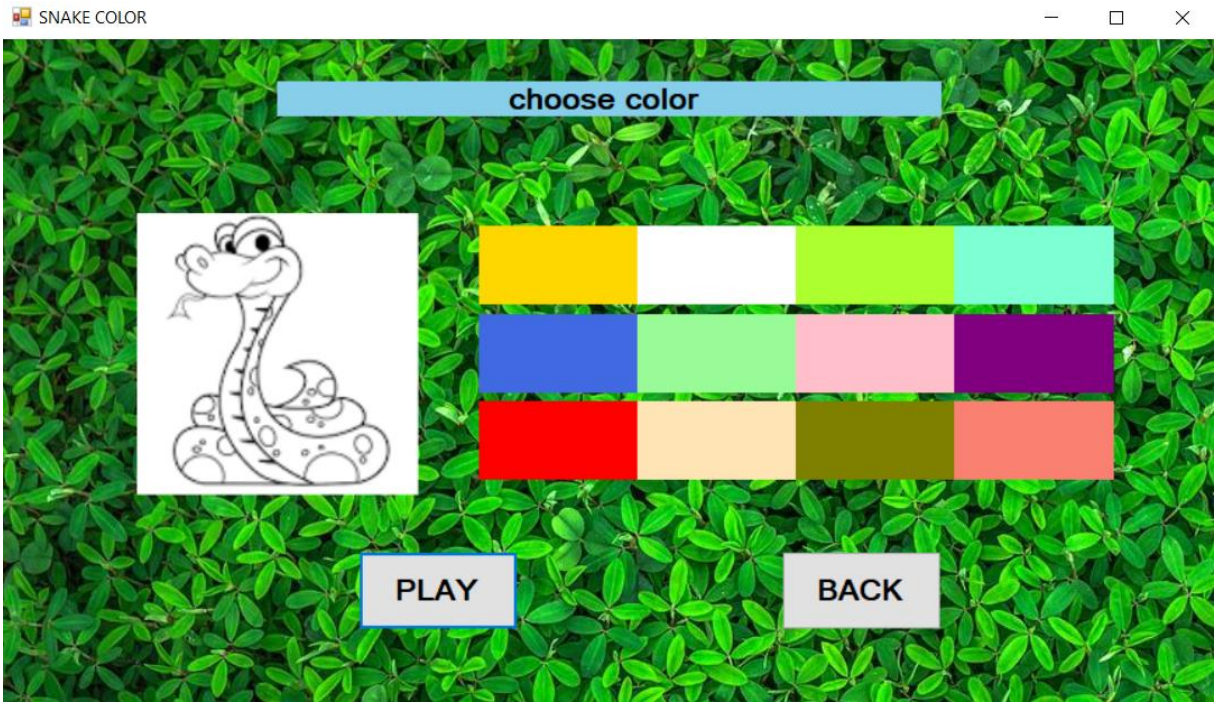
V oknu 'How to play' so predstavljena navodila za igro. Torej, kača se premika levo, desno, gor in dol s puščicami. Igro pa zaustavimo s klikom na Enter. Po prebranih navodilih, uporabnik začne z igro ali pa se vrne v začetno okno.

OKNO 'How to win'

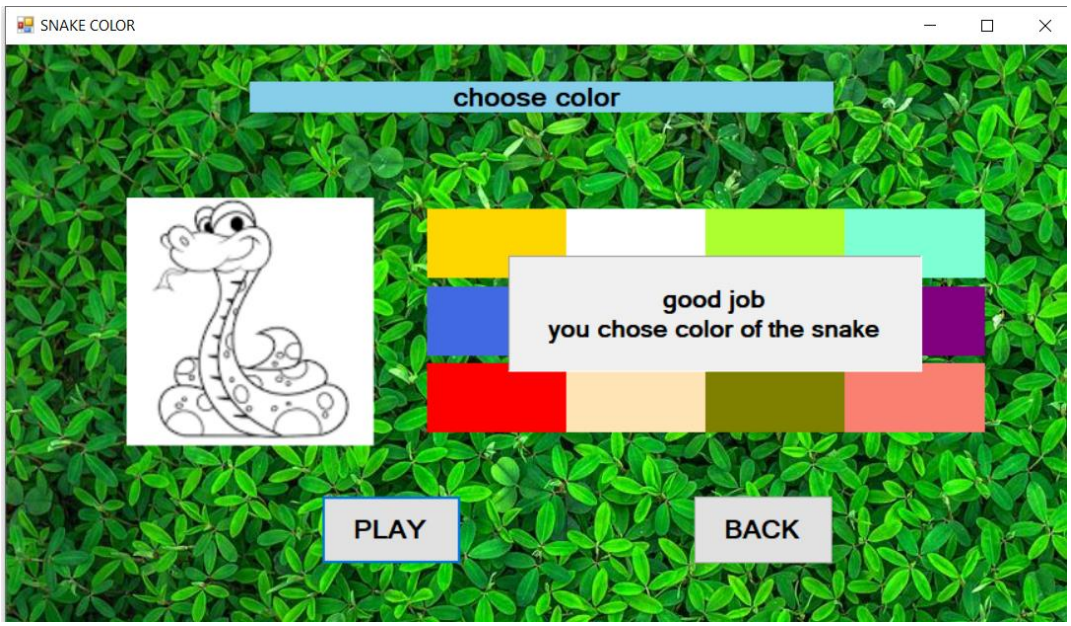


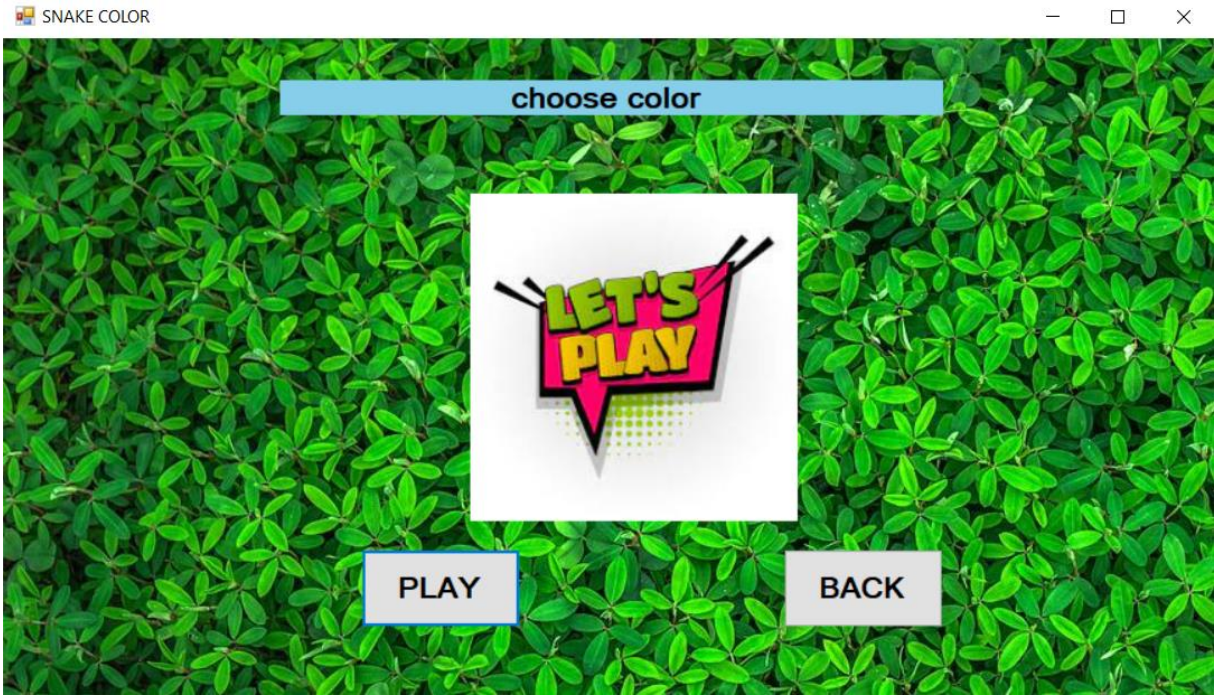
V oknu 'How to win' je predstavljena pot do zmage. Igralec igro kača zmaga pri zbranih 20 sadežev. Izgubi pa v primeru dotika z repom ali nesrečnem dotiku ovire. Ko uporabnik pridobi željene informacije lahko nadaljuje z igro ali pa se vrne na začetno okno.

OKNO ' Snake Color '



Uporabnik v oknu ' Snake Color 'ima možnost izbrati barvo kače. Željeno barvo izbere s klikom na eno izmed barv. V primeru, da je uspešno izbral barvo bo prejel povratno informacijo. Po izbrani barvi prične z igro s klikom na gumb Play ali pa se vrne na glavno okno.




SNAKE COLOR
choose color
LET'S PLAY
PLAY
BACK

PREMIKANJE KAČE

```
bool goLeft, goRight, goDown, goUp;

public Form1()
{
    InitializeComponent();
    new Settings();

}

private void KeyIsDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Left && Settings.directions != "right")
    {
        goLeft = true;
    }
    if (e.KeyCode == Keys.Right && Settings.directions != "left")
    {
        goRight = true;
    }
    if (e.KeyCode == Keys.Up && Settings.directions != "down")
    {
        goUp = true;
    }
    if (e.KeyCode == Keys.Down && Settings.directions != "up")
    {
        goDown = true;
    }
}
```

```
private void KeyIsUp(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Left)
    {
        goLeft = false;
    }
    if (e.KeyCode == Keys.Right)
    {
        goRight = false;
    }
    if (e.KeyCode == Keys.Up)
    {
        goUp = false;
    }
    if (e.KeyCode == Keys.Down)
    {
        goDown = false;
    }
}
```

Prijavimo se na dogodke, kjer poslušamo kdaj se bo zgodil klik na določen gumb tipkovnici.

Definiramo katere tipke na tipkovnici bomo uporabili.

Ker bomo s kliki na določene gumbe premikali kačo nas zanimajo dogodki ko se zgodi klik. Prav tako, nas zanima ko klika na gumb ni, saj ne želimo, da se kača premika brez našega nadzora.

Želja je premikati kačo glede na klike na tipkovnico.

Poslušamo, kdaj se zgodi dogodek in če je klik na levo puščico definiramo funkcijo, ki bo povedala kaj se more takrat zgoditi, zato je nastavljena na true. Ko uporabnik spusti gumb pa nastavimo funkcijo na false, saj želimo funkcijo zaustaviti.

```
private void GameTimerEvent(object sender, EventArgs e)
{
    if (goLeft)
    {
        Settings.directions = "left";
    }
    if (goRight)
    {
        Settings.directions = "right";
    }
    if (goDown)
    {
        Settings.directions = "down";
    }
    if (goUp)
    {
        Settings.directions = "up";
    }
```

Sedaj, ko imamo definirane dogodke, kateri se izvršijo po klikih na tipkovnici pa moramo povedati kaj naj se zgodi.

Ko se je zgodil klik na levo puščico, smo ustvarili dogodek 'goLeft', katerega smo nastavili smer premikanja levo.

```
for (int i = Snake.Count - 1; i >= 0; i--)
{
    if (i == 0)
    {

        switch (Settings.directions)
        {
            case "left":
                Snake[i].X--;
                break;
            case "right":
                Snake[i].X++;
                break;
            case "down":
                Snake[i].Y++;
                break;
            case "up":
                Snake[i].Y--;
                break;
        }

        if (Snake[i].X < 0)
        {
            Snake[i].X = maxWidth;
        }
        if (Snake[i].X > maxWidth)
        {
            Snake[i].X = 0;
        }
        if (Snake[i].Y < 0)
        {
            Snake[i].Y = maxHeight;
        }
        if (Snake[i].Y > maxHeight)
        {
            Snake[i].Y = 0;
        }
```

Kača se premika čez celo okno. Pri stiku z robom se prikaže na drugi strani.

Dogodki so ustvarjeni na en klik na tipkovnici, torej na en premik. V primeru smeri > levo < se kača premakne za eno enoto v levo in nato konča s premikanjem.

IZGLED KAČE , HRANE IN OVIRE

```
private void UpdatePictureBoxGraphics(object sender, PaintEventArgs e)
{
    string userColor = PassingValue;
    // Določimo pot do slike, kjer je shranjena slika za kačino glavo.
    string path = @"C:\Users\Ldon Kranjec\Desktop\HanaKranjec_P3_SN_popravki\Hana_Kranjec_P3_proj
    Image snakeHead = Image.FromFile(path);
    Graphics canvas = e.Graphics;
    Brush snakeColour;
    for (int i = 0; i < Snake.Count; i++)
    {
        if (i == 0) // prvi el kače je glava, katera je drugačna od trupa
        {
            canvas.DrawImage(snakeHead, new Rectangle(
            Snake[i].X * Settings.Width,
            Snake[i].Y * Settings.Height,
            Settings.Width + 10, Settings.Height + 10));
        }
```

Kača je sestavlja iz slike in krožcev. Slika predstavlja glavo, prav tako je tudi večje velikosti kot trup, krožci pa predstavljajo trup.

Izbrano sliko sem naložila v Resource file.

S funkcijo preverimo vse člene kače, če smo pri členu z indeksom 0, torej smo pri začetem členu oz. glavi, zato mu dodamo sliko.

```
else
{
    if (userColor == "0")
    {
        // trup so krožci avtomatsko zelene barve
        snakeColour = System.Drawing.Brushes.Chartreuse;
        canvas.FillEllipse(snakeColour, new Rectangle(
        Snake[i].X * Settings.Width,
        Snake[i].Y * Settings.Height,
        Settings.Width, Settings.Height
        ));
    }
```

Trup je obarvan z avtomatsko zeleno ali pa barvo, katero je izbral uporabnik.

Določitev avtomatskega barvanja ali določene barve nas pove vrednost v spremenljivki 'useColor'. Če je vrednost v spremenljivki 0, uporabimo avtomatsko barvanje.

```
else
{
    // trup so krožci izbrane barve
    List<String> colors1 = new List<string> { "Brushes.Gold", "Brushes.Yellow", "Brushes.Olive", "Brushes.Salmon",
                                              "Brushes.Red", "Brushes.GreenYellow", "Brushes.Moccasin", "Brushes.Pal
    List<Brush> colors2 = new List<Brush> { Brushes.Gold, Brushes.Yellow, Brushes.Olive, Brushes.Salmon, Brushes.Pi
                                            Brushes.Red, Brushes.GreenYellow, Brushes.Moccasin, Brushes.PaleGreen,
    int j = 0;
    foreach (string col in colors1)
    {
        string[] cut = col.Split('.');
        if (cut[1] == userColor)
        {
            Brush brushSnake = colors2[j];

            canvas.FillEllipse(brushSnake, new Rectangle(
            Snake[i].X * Settings.Width,
            Snake[i].Y * Settings.Height,
            Settings.Width, Settings.Height
            ));

        }
```

V primeru, da je v spremenljivki ' useColor' zapisana barva uporabimo sledeči postopek, kjer sestavimo dve tabeli, kateri opisujeta barve. V prvi imamo barve v drugi pa nize z barvami.

Z zanko preverimo ali vsebuje kateri od nizov ime barve iz spremenljivke 'useColor'. Če vsebuje potem uporabim to barvo iz tabele z zapisi barv.

```
int num = rand.Next(0, 5);
Brush[] foodi = new Brush[] { Brushes.AliceBlue, Brushes.Aqu
Brushes.Yellow, Brushes.Blue, Brushes.Pink, Brushes.Purple }
Brush f = foodi[num];
canvas.FillEllipse(f, new Rectangle(
food.X * Settings.Width,
        food.Y * Settings.Height,
Settings.Width, Settings.Height
    ));
```

Hrana se prikazuje na naključno izbranih točkah znotraj igralnega okna. Prav tako je naključno izbrana barva za efekt utripanja.

```
// izgled ovire
System.Drawing.Brush[] colors = new Brush[] { Brushes.Alice
Brushes.Yellow, Brushes.Blue, Brushes.Pink, Brushes.White }
Brush br = colors[num];
Pen myPen = new Pen(br, 3);

Rectangle myRectangle = new Rectangle(barrier.X * Settings.
barrier.Y * Settings.Width, Settings.Width, Settings.Height
canvas.DrawRectangle(myPen, myRectangle);
```

Poleg hrane se prikazuje tudi ovira. Prav tako na naključni lokaciji znotraj in utripa.

```
private void timerBarriera_Tick(object sender, EventArgs e)
{
    timerBarriera.Interval = 1000;
    barrier = new Moving { X = rand.Next(0, maxWidth), Y = rand.N
}
```

Ovire prikazujemo na naključno izbranih lokacijah znotraj igralnega panela, tako da uporabimo časovnik. S pomočjo časovnika lahko na določen izbran prikazujemo oviro.

```
private List<Moving> Snake = new List<Moving>();
private Moving food = new Moving();
private Moving barrier = new Moving();
```

Za tekoče premikanje kače in naključno prikazovanje hrane in ovire definiramo premikanje, saj se bodo tako krožci iz kačinega trupa prikazovali povezano. Hrana in ovira pa tekoče prikazovala na naključnih lokacijah znotraj igralnega panela.

Prenos barve iz okna 'Snake Color' v igro

' Snake Color ' okno :

```
public void playButten_Click(object sender, EventArgs e)
{

    Form1 playGame = new Form1(colorSnake);
    playGame.PassingValue = colorSnake;
    playGame.ShowDialog();

    this.Close();
}
```

Igralno okno :

```
public Form1(string passingValue)
{
    InitializeComponent();
    new Settings();
    string userColor = passingValue;
}

public string PassingValue {get; set; }
```

```
private void UpdatePictureBoxGraphics(object sender, PaintEventArgs e)
{
    string userColor = PassingValue;
```

Vrednost, katero želimo prenesti med okni je shranjena v spremenljivki ' colorSnake ' .

Pri definiranju dogodka po kliku na gumb 'Play' ustvarimo novo okno, katero je v našem primeru igralno okno.

S klikom na gumb 'Play' bomo prenesli vrenost v drugo okno.

Pri ustvarjanju novega okna zapišemo še parameter katerega želimo prenesti pri prehodu na igralno okno.

Ustvarimo povezavo.

Pri definiranju igralne okna sprejmemo preneseno vrednost.

Določimo lasnost.

Nato uporabimo prenešeno vrednost.

Zvočni efekti

```
private void GameOver()

    // Konec igre, zaigra zvok za poražence in izpišejo se točke.
    gameTimer.Stop();
    startButton.Visible = true;
    startButton.Enabled = true;
    goHomeButton.Visible = true;
    goHomeButton.Enabled = true;
    System.IO.Stream str1 = Properties.Resources.sound_gameOver;
    SoundPlayer sound_gameOver = new SoundPlayer(str1);
    sound_gameOver.Play();
    lbl_gameOver.Visible = true;
    txtScore.Text = $"High score is {score}.";
}
```

Ob vsakem premiku kače dodamo zvočni efekt, prav tako ob stiku kače s hrano in koncem igre.

Zvočni efekte lahko dodajamo tako, da najprej dodamo imenski prostor

> System.Media;< ,

nato shranimo zvok v Resources.

Določimo povezavo do imenika z zvokom in predvajamo.

Gumb Start

```
private void RestartGame()
{
    // Začela se je igra, kačo postavimo na začetno mesto.
    // Točke imajo na začetku vrednost 0.
    maxWidth = picCanvas.Width / Settings.Width - 1;
    maxHeight = picCanvas.Height / Settings.Height - 1;

    Snake.Clear();
    startButton.Visible = false;
    startButton.Enabled = false;
    goHomeButton.Visible = false;

    score = 0;
    txtScore.Text = "Score: " + score;

    Moving head = new Moving { X = 10, Y = 5 };
    Snake.Add(head);

    for (int i = 0; i < 10; i++)
    {
        Moving body = new Moving();
        Snake.Add(body);
    }
    // Hrana se bo prikazovala z naključnimi kordinatami znotraj igralnega okna.
    food = new Moving { X = rand.Next(2, maxWidth - 5), Y = rand.Next(2, maxHeight - 5) };
    barrier = new Moving { X = rand.Next(2, maxWidth - 5), Y = rand.Next(2, maxHeight - 5) };
    gameTimer.Start();
    lbl_gameOver.Visible = false;
}
```

Ko se odpre igralno okno igro zaženemo s klikom na Start. Pripravimo spremenljivko za shranjevanje igralčevih točk, ugasnemo gumb Start, saj ga med igranjem igre ne potrebujemo. Prikažemo kačo, dodamo hrano in oviro na naključno mesto na igralnem oknu.

Prav tako ugasnemo in skrijemo gumb za povratek na začetno okno. Po kliku na Enter se prikažeta gumba 'Start' in 'Go home' za nadaljevanje igre oziroma povratek na začetno okno.

Kača poje hrano, točka

```
    if (Snake[i].X == food.X && Snake[i].Y == food.Y)
    {
        EatFood();
    }
```

```
private void EatFood()
    {
    // Kača se je prekrila s hrano, zato uporabnik dobi dodatno točko.
    // Kača se podaljša in hrana dobi nove naključne kordinate kamor se naključno prikaže.
        score += 1;
        txtScore.Text = "Score: " + score;

        Moving body = new Moving
        {
            X = Snake[Snake.Count - 1].X,
            Y = Snake[Snake.Count - 1].Y
        };

        Snake.Add(body);

        food = new Moving { X = rand.Next(2, maxWidth - 5), Y = rand.Next(2, maxHeight - 5) };
        barrier = new Moving { X = rand.Next(2, maxWidth - 5), Y = rand.Next(2, maxHeight - 5) };
        // Ob prekriti hrani s kačo se izvede zmagovalni zvok.
        System.IO.Stream str1 = Properties.Resources.sound_eatFood;
        SoundPlayer sound_eatFood = new SoundPlayer(str1);
        sound_eatFood.Play();
}
```

Kača poje hrano na način prekrivanja, ko se koordinate kače in hrane prekrijeta pravimo, da je kača hrano pojedla.

Igralec pri tem dobi dodatno točko. To seveda izpišemo. Kača se podaljša za eno enoto. Hrana pa se prikaže na novi naključni lokaciji.

Zamrznitev igre

```
private void PauseGame()
{
    gameTimer.Enabled = false;
    startButton.Visible = true;
    startButton.Enabled = true;

}
```

Tako, kot smo se prijavili na dogodke za premike kače se prijavimo tudi za poslušanje, kdaj se bo zgodil dogodek klik na gumb 'Enter'. S klikom na ta gumb zamrznemo igro.

Po zamrznitvi igre, ponovno aktiviramo gumb Start, da bomo lahko nadaljevali igro.

KONEC IGRE

```
private void GameOver()

    // Konec igre, zaigra zvok za poražence in izpišejo se točke.
    gameTimer.Stop();
    startButton.Visible = true;
    startButton.Enabled = true;
    goHomeButton.Visible = true;
    goHomeButton.Enabled = true;
    System.IO.Stream str1 = Properties.Resources.sound_gameOver;
    SoundPlayer sound_gameOver = new SoundPlayer(str1);
    sound_gameOver.Play();
    lbl_gameOver.Visible = true;
    txtScore.Text = $"High score is {score}.";
}
```

Igralec je zmagovalec, če je zbral 20 točk.

Na vsakem koraku, ko kača poje hrano se pokliče funkcija >WinWin<, ki preveri ali je kača že pojedla 20 enot hrane. V primeru, da je pojedla se izvede zmagovalni konec.

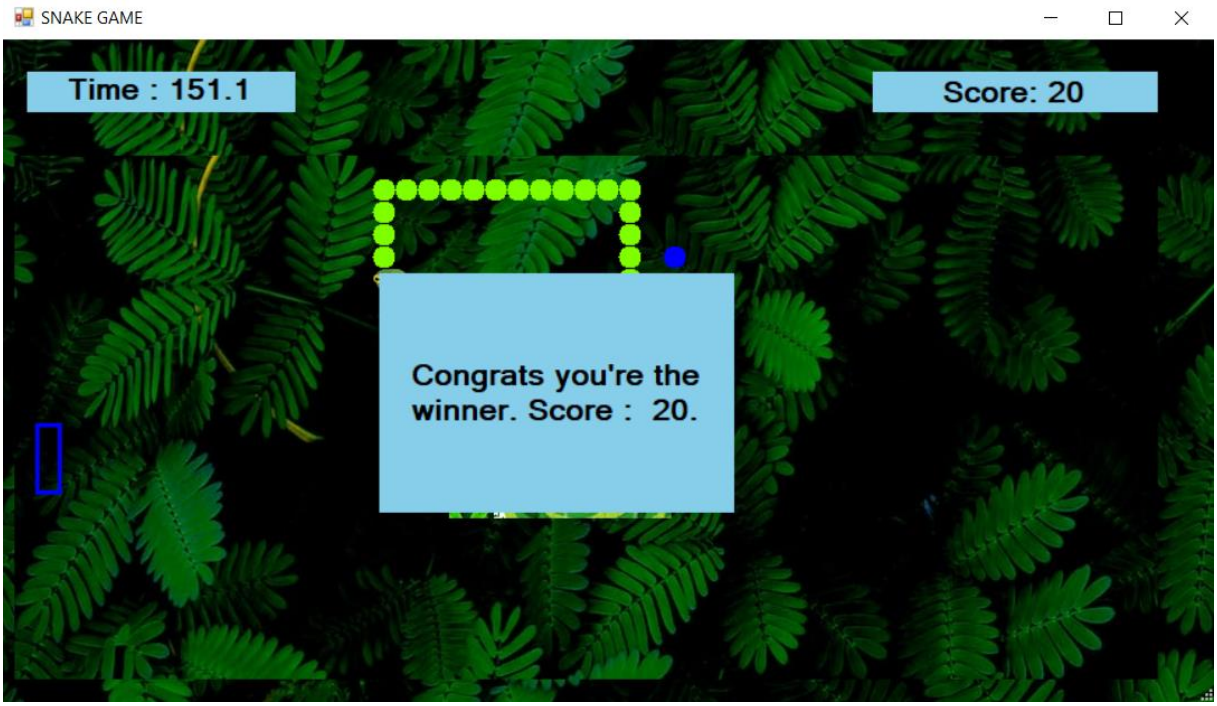
ZMAGA

```
private void WinWin()
{
    gameTimer.Stop();
    pictureBox_win.Visible = true;
    txtScore.Text = $"Congrats you're the winner. Score :  {score}.";
    startButton.Enabled = true;
    System.IO.Stream str1 = Properties.Resources.sound_eatFood;
    SoundPlayer sound_win= new SoundPlayer(str1);
    sound_win.Play();
}
```

```
if (Snake[i].X == food.X && Snake[i].Y == food.Y)
{
    EatFood();
    if (score == 20)
    {
        WinWin();
    }
}
```

Definiramo metodo ' WinWin' za preverjanje ali je igralec zmagal.

Na vsakem koraku, ko se kača prekrije s hrano preverimo ali je točk 20. V tem primeru pokličemo metodo WinWin, katera je različica metode GameOver.



Prehod med okni

```
private void playSnakeGameToolStripMenuItem1_Click(object sender, EventArgs e)
{
    string colorSnake = "0";
    Form1 playGame = new Form1(colorSnake);
    playGame.PassingValue = colorSnake;
    playGame.ShowDialog();
    this.Hide();
}

private void playSnakeGameToolStripMenuItem_Click(object sender, EventArgs e)
{
    HowToPlay howToPlay = new HowToPlay();
    howToPlay.Show();
    this.Hide();


}
```

Pri prehodu med okni prenesemo vrednost parametra. Ker imamo definirano, da okno igra odpiramo s parametrom. Moram tukaj definirati nek niz, ki ga bomo prenesli, ker samo iz okna 'Snake color' prenesemo barvo kače.

Definiramo novo okno in mu ukažemo naj

Viri

- Opis izdelave Igra kača **https://www.mooict.com/c-tutorial-create-a-classic-snake-game-in-visual-studio/** (dostopno, 10.09.2022)
- Razred Brush https://docs.microsoft.com/en-us/dotnet/api/system.drawing.solidbrush?view=dotnet-plat-ext-6.0 (dostopno, 10.09.2022)
- Video posnetek Prenos parametrov in odpiranje oken v c# https://www.youtube.com/watch?v=l1E2oFjnqRo (dostopno, 10.09.2022)
- Uporaba virov https://stackoverflow.com/questions/13592150/load-image-from-resources (dostopno, 10.09.2022)