KELOMPOK PENGACARA JAYA

8 January 2023

ALP ALGORITHM & PROGRAMMING LANGUAGE SEMESTER 1

SNAKE GAME

Anggota: Nathan,Juan,Benito,
Malvin, Gio, Megan

Let's get started

AGENDA

Introduction

Demo Snake Game

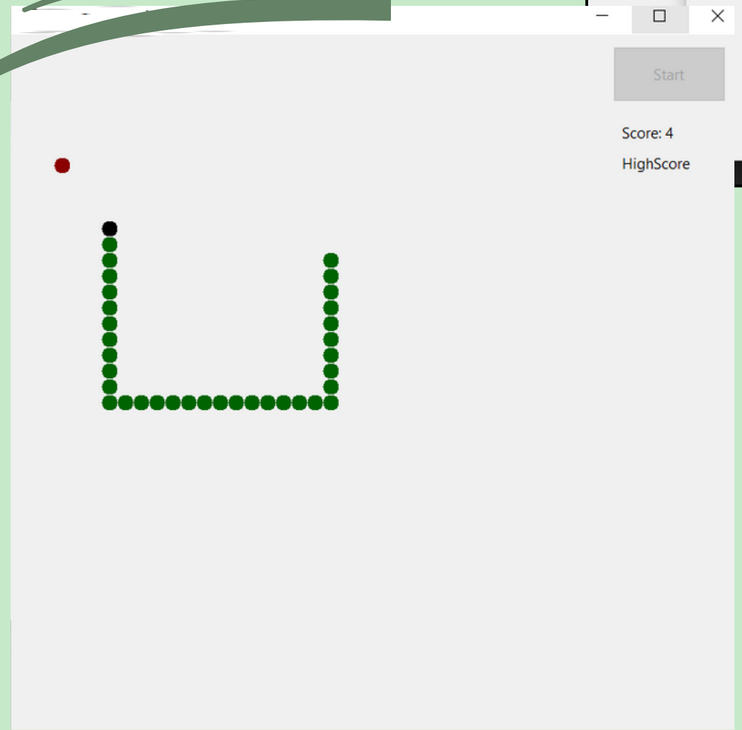
Toolbox

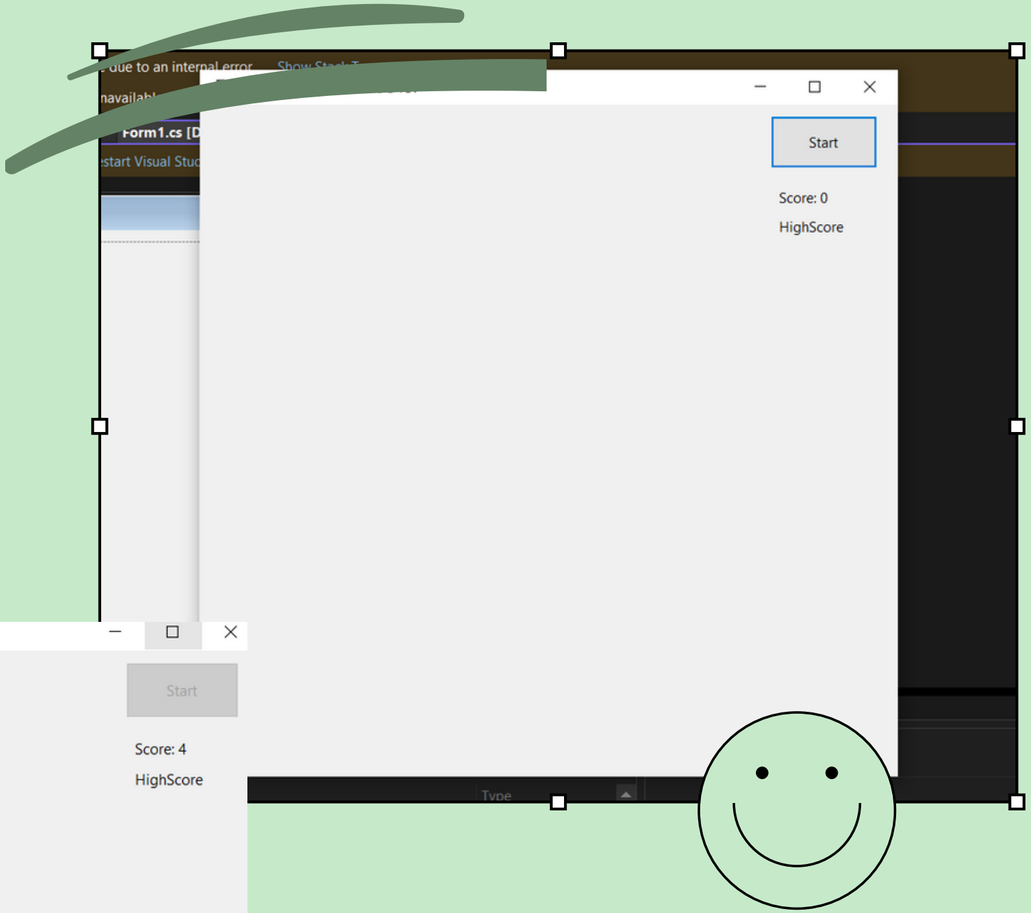
Events

INTRODUCTION

SNAKE GAME adalah permainan mengontrol gerakan ular untuk mendapatkan makanan yang tersebar di labirin.

SNAKE GAME yang kita buat menggunakan WINDOWS FORM.

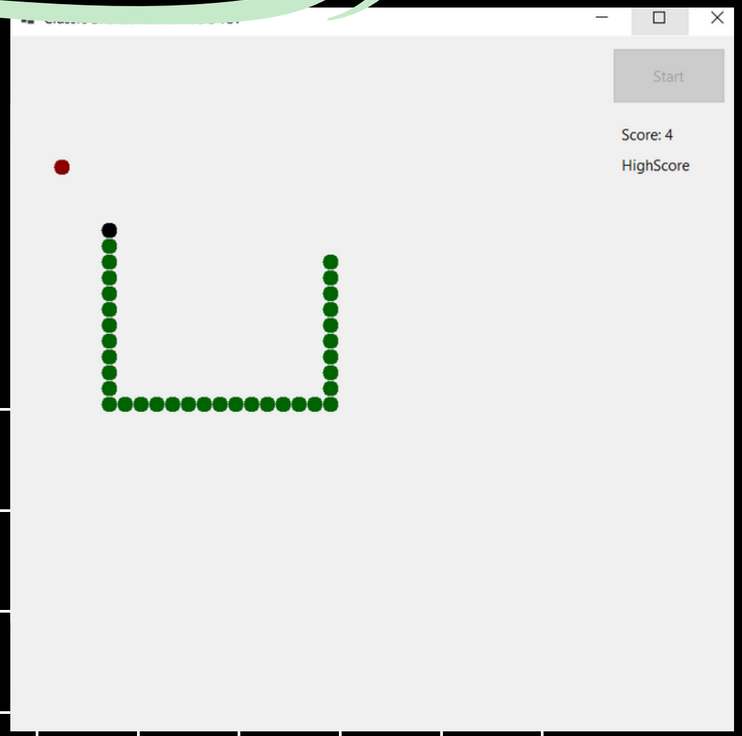


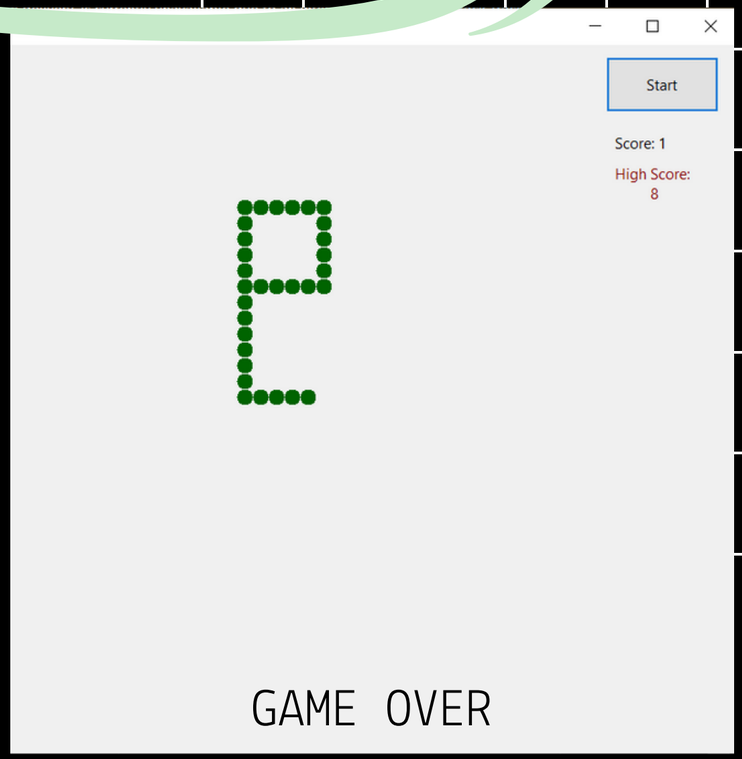


Back to Agenda Page

DEMO SNAKE GAME

Back to Agenda Page





TOOLBOX

a.) Button > startButton
b.) PictureBox > picCanvas
c.) TextBox > txtScore & txtHighScore
d.) Timer > GameTimer

EVENTS

a.) Form(Form1) > KeyDown: KeyIsDown, KeyUp: KeyIsUp

b.) PictureBox(picCanvas) > Paint: UpdatePictureBoxGraphics

c.) Button(StartButton) > Click: StartGame

d.) Timer(GameTimer) > GameTimerEvent

CIRCLE CLASS

```
e                                   SnakeGame.Settings                    Width
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SnakeGame
{
    21 references
    class Settings
    {
        6 references
        public static int Width { get; set; }
        6 references
        public static int Height { get; set; }

        public static string directions;

        1 reference
        public Settings()
        {
            Width = 16;
            Height = 16;
            directions = "left";
        }
    }
}
```

SETTINGS CLASS

```
e                    SnakeGame.Settings                    Width
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SnakeGame
{
    21 references
    class Settings
    {
        6 references
        public static int Width { get; set; }
        6 references
        public static int Height { get; set; }

        public static string directions;

        1 reference
        public Settings()
        {
            Width = 16;
            Height = 16;
            directions = "left";
        }
    }
}
```

VARIABLES

```
snake_game                    snake_game.Form1                    Snake
using System.Drawing.Imaging;
using SnakeGame;
using static System.Formats.Asn1.AsnWriter;
using static System.Windows.Forms.VisualStyles.VisualStyleElement.Rebar;

namespace snake_game
{
    3 references
    public partial class Form1 : Form
    {
        private List<Circle> Snake = new List<Circle>();
        private Circle food = new Circle();

        int maxWidth;
        int maxHeight;

        int score;
        int highScore;

        int timeSinceStart;

        bool gameInProgress = false;

        Random rand = new Random();

        bool goLeft, goRight, goDown, goUp;

        1 reference
        public Form1()
```

KEYISDOWN

```
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;
    }
}
```

KEYISUP

```
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;
    }
}
```

START GAME

```
private void StartGame(object sender, EventArgs e)
{
    RestartGame();
}
```

RESTART GAME

```
private void RestartGame()
{
    maxWidth = picCanvas.Width / Settings.Width - 1;
    maxHeight = picCanvas.Height / Settings.Height - 1;

    Snake.Clear();

    startButton.Enabled = false;
    score = 0;
    txtScore.Text = "Score: " + score;

    Circle head = new Circle { X = 10, Y = 5 };
    Snake.Add(head); // adding the head part of the snake to the list

    for (int i = 0; i < 30; i++)
    {
        Circle body = new Circle();
        Snake.Add(body);
    }

    food = new Circle { X = rand.Next(2, maxWidth), Y = rand.Next(2, maxHeight) };
    gameInProgress = true;
    timeSinceStart = 0;
    gameTimer.Start();

}
```

UPDATEPICTUREBOXGRAPHIC

```
private void UpdatePictureBoxGraphics(object sender, PaintEventArgs e)
{
    Graphics canvas = e.Graphics;

    Brush snakeColour;

    for (int i = 0; i < Snake.Count && i <= timeSinceStart; i++)
    {
        if (i == 0)
        {
            snakeColour = Brushes.Black;
        }
        else
        {
            snakeColour = Brushes.DarkGreen;
        }

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

    if (gameInProgress == true)
    {
        canvas.FillEllipse(Brushes.DarkRed, new Rectangle
        (
        food.X * Settings.Width,
        food.Y * Settings.Height,
        Settings.Width, Settings.Height
        ));
    }
    timeSinceStart++;
}
```

GAME TIMER EVENT

1

```
private void GameTimerEvent(object sender, EventArgs e)
{
    // setting the directions

    if (goLeft)
    {
        Settings.directions = "left";
    }
    if (goRight)
    {
        Settings.directions = "right";
    }
    if (goDown)
    {
        Settings.directions = "down";
    }
    if (goUp)
    {
        Settings.directions = "up";
    }
    // end of directions
```

2

```
    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;
            }


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

3

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

            for (int j = 1; j < Snake.Count; j++)
            {

                if (Snake[i].X == Snake[j].X && Snake[i].Y == Snake[j].Y)
                {
                    GameOver();
                }

            }


        }
        else
        {
            Snake[i].X = Snake[i - 1].X;
            Snake[i].Y = Snake[i - 1].Y;
        }
    }
    picCanvas.Invalidate();
}
```

EATFOOD

```
private void EatFood()
{
    score += 1;

    txtScore.Text = "Score: " + score;

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

    Snake.Add(body);

    food = new Circle { X = rand.Next(2, maxWidth), Y = rand.Next(2, maxHeight) };
}
```

GAMEOVER

```
private void GameOver()
{
    gameTimer.Stop();
    startButton.Enabled = true;

    if (score > highScore)
    {
        highScore = score;
        txtHighScore.Text = "High Score: " + Environment.NewLine + highScore;
        txtHighScore.ForeColor = Color.Maroon;
        txtHighScore.TextAlign = ContentAlignment.MiddleCenter;
    }
    gameInProgress = false;
}
```

THANK YOU :)