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 cLibrary;

namespace PVM5\_6615

{

public partial class mainForm : Form

{

Random r = new Random();

int numb = 0;

Timer gametime = new Timer();

dataForm df;

player p1, p2;

pForm p1f, p2f;

List<Button> obs;

List<food> foods;

public mainForm()

{

InitializeComponent();

setLayer();

p1f = new pForm();

p2f = new pForm();

obs = new List<Button>();

foods = new List<food>();

this.KeyDown += MainForm\_KeyDown;

gametime.Tick += Gametime\_Tick;

gametime.Interval = 1000;

}

private void setLayer()

{

this.Text = "Player VS Player";

this.IsMdiContainer = true;

this.DoubleBuffered = true;

this.KeyPreview = true;

this.WindowState = FormWindowState.Maximized;

}

public void getData(string n1, string n2, Color c1, Color c2)

{

p1 = new player(n1, c1);

p2 = new player(n2, c2);

}

private void makeP1()

{

p1f.Text = "Name Player: " + p1.Name;

p1f.MdiParent = this;

p1f.updateScore(p1);

p1f.spawnPlayer(p1, p2);

for (int i = 0; i < 10; i++)

{

int rtype = r.Next(1, 4);

int x = r.Next(this.Width);

int y = r.Next(this.Height);

food ftemp = new food(rtype, x + 100, y - 100);

foods.Add(ftemp);

}

p1f.spawnFood(foods);

p1f.createMap(obs);

p1f.Show();

}

private void makeP2()

{

p2f.Text = "Name Player: " + p2.Name;

p2f.MdiParent = this;

p2f.updateScore(p2);

p2f.spawnPlayer(p1, p2);

for (int i = 0; i < 10; i++)

{

int rtype = r.Next(1, 4);

int x = r.Next(this.Width);

int y = r.Next(this.Height);

food ftemp = new food(rtype, x + 100, y - 100);

foods.Add(ftemp);

}

p2f.spawnFood(foods);

p2f.createMap(obs);

p2f.Show();

}

public void startGame()

{

makeP1();

makeP2();

p2f.Left = (p1f.Left + p1f.Width) + 100;

p2f.Top = p1f.Top;

gametime.Start();

df.Close();

}

private void Gametime\_Tick(object sender, EventArgs e)

{

numb++;

if (numb % 10 == 0)

{

p1f.spawnFood(foods);

p2f.spawnFood(foods);

}

if (numb == 120)

{

gametime.Stop();

MessageBox.Show("GAME BERAKHIR!");

}

}

private void MainForm\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.W)

{

Keys temp = Keys.W;

p1f.move(temp);

p2f.move(temp);

}

else if (e.KeyCode == Keys.A)

{

Keys temp = Keys.A;

p1f.move(temp);

p2f.move(temp);

}

else if (e.KeyCode == Keys.S)

{

Keys temp = Keys.S;

p1f.move(temp);

p2f.move(temp);

}

else if (e.KeyCode == Keys.D)

{

Keys temp = Keys.D;

p1f.move(temp);

p2f.move(temp);

}

if (e.KeyCode == Keys.Up)

{

Keys temp = Keys.Up;

p1f.move(temp);

p2f.move(temp);

}

else if (e.KeyCode == Keys.Left)

{

Keys temp = Keys.Left;

p1f.move(temp);

p2f.move(temp);

}

else if (e.KeyCode == Keys.Down)

{

Keys temp = Keys.Down;

p1f.move(temp);

p2f.move(temp);

}

else if (e.KeyCode == Keys.Right)

{

Keys temp = Keys.Right;

p1f.move(temp);

p2f.move(temp);

}

}

private void startToolStripMenuItem\_Click(object sender, EventArgs e)

{

if (df == null) df = new dataForm();

df.MdiParent = this;

df.Show();

}

private void highscoreToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

private void exitToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

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 cLibrary;

namespace PVM5\_6615

{

public partial class dataForm : Form

{

Color c1;

Color c2;

public dataForm()

{

InitializeComponent();

this.Text = "Data Player";

}

private void color1Button\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == DialogResult.OK)

{

if (colorDialog1.Color == Color.Red || colorDialog1.Color == Color.Yellow || colorDialog1.Color == Color.Green)

{

MessageBox.Show("Tidak boleh menggunakan warna tersebut!");

}

else c1 = colorDialog1.Color;

}

}

private void color2Button\_Click(object sender, EventArgs e)

{

if (colorDialog2.ShowDialog() == DialogResult.OK)

{

if (colorDialog2.Color == Color.Red || colorDialog2.Color == Color.Yellow || colorDialog2.Color == Color.Green)

{

MessageBox.Show("Tidak boleh menggunakan warna tersebut!");

}

else c2 = colorDialog2.Color;

}

}

private void playButton\_Click(object sender, EventArgs e)

{

mainForm parent = (mainForm)MdiParent;

string n1 = nameText1.Text;

string n2 = nameText2.Text;

parent.getData(n1, n2, c1, c2);

parent.startGame();

}

}

}

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 cLibrary;

namespace PVM5\_6615

{

public partial class pForm : Form

{

Random r = new Random();

Button player1, player2;

player fp1;

player fp2;

List<Button> foodform = new List<Button>();

public pForm()

{

InitializeComponent();

setLayer();

this.Enabled = false;

}

private void setLayer()

{

this.Width = 450;

this.Height = 450;

this.MinimizeBox = false;

this.MaximizeBox = false;

}

public void spawnPlayer(player p1, player p2)

{

p1.Icon = new Button();

p1.Icon.Size = new Size(30, 30);

p1.Icon.Location = new Point(10, 10);

p1.Icon.BackColor = p1.Color;

fp1 = p1;

player1 = p1.Icon;

this.Controls.Add(player1);

p2.Icon = new Button();

p2.Icon.Size = new Size(30, 30);

p2.Icon.Location = new Point(40, 10);

p2.Icon.BackColor = p2.Color;

fp2 = p2;

player2 = p2.Icon;

this.Controls.Add(player2);

}

public void spawnFood(List<food> foods)

{

foods = new List<food>();

for (int i = 0; i < 10; i++)

{

int rtype = r.Next(1, 4);

int x = r.Next(this.Width);

int y = r.Next(this.Height);

food ftemp = new food(rtype, x + 100, y - 100);

foods.Add(ftemp);

}

for (int i = 0; i < foods.Count; i++)

{

foodform.Add(foods[i].Icon);

}

for (int i = 0; i < foodform.Count; i++)

{

this.Controls.Add(foodform[i]);

}

}

public void createMap(List<Button> obs)

{

obs = new List<Button>();

for (int i = 0; i < 6; i++)

{

for (int j = 0; j < 6; j++)

{

Button btemp = new Button();

btemp.Size = new Size(30, 30);

btemp.Location = new Point(50 + (i \* 60), 40 + (j \* 60));

obs.Add(btemp);

}

}

for (int i = 0; i < obs.Count; i++)

{

this.Controls.Add(obs[i]);

}

}

public void updateScore(player p)

{

scoreLabel.Location = new Point(this.Width - 50, 0);

scoreLabel.Text = p.Score.ToString();

}

public void move(Keys k)

{

if (k == Keys.W)

{

player1.Top -= 30;

eat(player1);

}

else if (k == Keys.A)

{

player1.Left -= 30;

eat(player1);

}

else if (k == Keys.S)

{

player1.Top += 30;

eat(player1);

}

else if (k == Keys.D)

{

player1.Left += 30;

eat(player1);

}

if (k == Keys.Up)

{

player2.Top -= 30;

eat(player2);

}

else if (k == Keys.Left)

{

player2.Left -= 30;

eat(player2);

}

else if (k == Keys.Down)

{

player2.Top += 30;

eat(player2);

}

else if (k == Keys.Right)

{

player2.Left += 30;

eat(player2);

}

}

public void eat(Button player)

{

for (int i = 0; i < foodform.Count; i++)

{

if ((player.Location.X + player.Width) >= (foodform[i].Location.X) &&

player.Location.X <= (foodform[i].Location.X + foodform[i].Width) &&

(player.Location.Y + player.Height) >= foodform[i].Location.Y &&

player.Location.Y <= (foodform[i].Location.Y + foodform[i].Height))

{

this.Controls.Remove(foodform[i]);

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Drawing;

namespace cLibrary

{

public class player

{

private string name;

private int score;

private Button icon;

private Color color;

public player(string name, Color color)

{

this.name = name;

this.score = 0;

this.color = color;

this.icon = new Button();

this.icon.Location = new Point(10, 10);

this.icon.Size = new Size(30, 30);

this.icon.BackColor = this.color;

}

public string Name

{

set { this.name = value; }

get { return this.name; }

}

public int Score

{

set { this.score = value; }

get { return this.score; }

}

public Button Icon

{

set { this.icon = value; }

get { return this.icon; }

}

public Color Color

{

set { this.color = value; }

get { return this.color; }

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Drawing;

using System.Windows.Forms;

namespace cLibrary

{

public class food

{

private string type;

private int val;

private Color color;

private Button icon;

public food(int rtype, int x, int y)

{

if (rtype == 1)

{

this.type = "Red";

this.val = 25;

this.color = Color.Red;

this.icon = new Button();

this.icon.Location = new Point(x, y);

this.icon.Size = new Size(30, 30);

this.icon.BackColor = this.color;

}

else if (rtype == 2)

{

this.type = "Yellow";

this.val = 50;

this.color = Color.Yellow;

this.icon = new Button();

this.icon.Location = new Point(x, y);

this.icon.Size = new Size(30, 30);

this.icon.BackColor = this.color;

}

else if (rtype == 3)

{

this.type = "Green";

this.val = 100;

this.color = Color.Green;

this.icon = new Button();

this.icon.Location = new Point(x, y);

this.icon.Size = new Size(30, 30);

this.icon.BackColor = this.color;

}

}

public string Type

{

set { this.type = value; }

get { return this.type; }

}

public int Val

{

set { this.val = value; }

get { return this.val; }

}

public Color Color

{

set { this.color = value; }

get { return this.color; }

}

public Button Icon

{

set { this.icon = value; }

get { return this.icon; }

}

}

}