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 PVM3\_6615

{

public partial class Form1 : Form

{

Random r = new Random();

Timer small;

Timer medium;

Timer big;

Timer heart;

Timer danger;

Timer whale;

int score = 0, hp= 3;

fish player;

List<PictureBox> foodPb;

List<fish> dbfood;

List<int> HighScore;

public Form1()

{

InitializeComponent();

foodPb = new List<PictureBox>();

dbfood = new List<fish>();

HighScore = new List<int>();

this.Width = 1000;

this.Height = 600;

menuPanel.Visible = false;

scoreList.Visible = false;

small = new Timer();

medium = new Timer();

big = new Timer();

heart = new Timer();

danger = new Timer();

whale = new Timer();

startUp();

resources();

spawnSmall();

spawnMedium();

spawnBig();

spawnDanger();

spawnWhale();

spawnHeart();

}

private void startUp()

{

playerPb.Size = new Size(75, 50);

playerPb.Location = new Point(500, 300);

playerPb.SizeMode = PictureBoxSizeMode.StretchImage;

player = new fish(playerPb.Location.X, playerPb.Location.Y);

}

private void resources()

{

score = player.Score;

hpLabel.Text = hp.ToString();

scoreLabel.Text = score.ToString();

}

private void endMenu()

{

menuPanel.Size = new Size(500, 500);

menuPanel.Location = new Point(500, 100);

menuPanel.Visible = true;

small.Stop();

medium.Stop();

big.Stop();

heart.Stop();

danger.Stop();

whale.Stop();

gerakPaus.Stop();

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

{

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

}

foodPb.Clear();

dbfood.Clear();

}

private void spawnSmall()

{

small.Interval = 5000;

small.Tick += (s, e) =>

{

int move = r.Next(1, 3);

fish ftemp = new fish("small");

PictureBox fishPb = new PictureBox();

if (move == 1) fishPb.Location = new Point(0, r.Next(0, this.Height - 100));

else fishPb.Location = new Point(this.Width, r.Next(0, this.Height - 100));

fishPb.Size = new Size(50, 50);

fishPb.Image = Image.FromFile("images/small.jpg");

fishPb.SizeMode = PictureBoxSizeMode.StretchImage;

fishMove(fishPb, move);

foodPb.Add(fishPb);

dbfood.Add(ftemp);

this.Controls.Add(fishPb);

};

small.Start();

}

private void spawnMedium()

{

medium.Interval = 10000;

medium.Tick += (s, e) =>

{

int move = r.Next(1, 3);

fish ftemp = new fish("medium");

PictureBox fishPb = new PictureBox();

if (move == 1) fishPb.Location = new Point(0, r.Next(0, this.Height - 100));

else fishPb.Location = new Point(this.Width, r.Next(0, this.Height - 100));

fishPb.Size = new Size(100, 80);

fishPb.Image = Image.FromFile("images/medium.jpg");

fishPb.SizeMode = PictureBoxSizeMode.StretchImage;

fishMove(fishPb, move);

foodPb.Add(fishPb);

dbfood.Add(ftemp);

this.Controls.Add(fishPb);

};

medium.Start();

}

private void spawnBig()

{

big.Interval = 15000;

big.Tick += (s, e) =>

{

int move = r.Next(1, 3);

fish ftemp = new fish("big");

PictureBox fishPb = new PictureBox();

if (move == 1) fishPb.Location = new Point(0, r.Next(0, this.Height - 100));

else fishPb.Location = new Point(this.Width, r.Next(0, this.Height - 100));

fishPb.Size = new Size(160, 100);

fishPb.Image = Image.FromFile("images/big.jpg");

fishPb.SizeMode = PictureBoxSizeMode.StretchImage;

fishMove(fishPb, move);

foodPb.Add(fishPb);

dbfood.Add(ftemp);

this.Controls.Add(fishPb);

};

big.Start();

}

private void spawnDanger()

{

danger.Interval = 15000;

danger.Tick += (s, e) =>

{

int x = this.Width;

int y = r.Next(0, this.Height - 100);

PictureBox dangerPb = new PictureBox();

dangerPb.Location = new Point(x, y);

dangerPb.Size = new Size(100, 100);

dangerPb.Image = Image.FromFile("images/danger.png");

dangerPb.SizeMode = PictureBoxSizeMode.StretchImage;

whaleMove(dangerPb);

this.Controls.Add(dangerPb);

};

danger.Start();

}

private void spawnWhale()

{

whale.Interval = 20000;

whale.Tick += (s, e) =>

{

int x = this.Width;

int y = r.Next(0, this.Height - 100);

fish ftemp = new fish("whale");

PictureBox fishPb = new PictureBox();

fishPb.Location = new Point(x, y);

fishPb.Size = new Size(200, 150);

fishPb.Image = Image.FromFile("images/whale.jpg");

fishPb.SizeMode = PictureBoxSizeMode.StretchImage;

whaleMove(fishPb);

foodPb.Add(fishPb);

dbfood.Add(ftemp);

this.Controls.Add(fishPb);

};

whale.Start();

}

private void fishMove(PictureBox fish, int move)

{

gerakIkan.Interval = 80;

gerakIkan.Tick += (s, e) =>

{

if (move == 1) fish.Left += 10;

else fish.Left -= 10;

};

gerakIkan.Start();

}

private void whaleMove(PictureBox fish)

{

gerakPaus.Interval = 20;

gerakPaus.Tick += (s, e) =>

{

fish.Left -= 10;

};

gerakPaus.Start();

}

private void spawnHeart()

{

heart.Interval = 60000;

heart.Tick += (s, e) =>

{

int x = r.Next(0, this.Width - 100);

int y = r.Next(0, this.Height - 100);

fish ftemp = new fish("powerup");

PictureBox fishPb = new PictureBox();

fishPb.Location = new Point(x, y);

fishPb.Size = new Size(50, 50);

fishPb.Image = Image.FromFile("images/heart.png");

fishPb.SizeMode = PictureBoxSizeMode.StretchImage;

foodPb.Add(fishPb);

dbfood.Add(ftemp);

this.Controls.Add(fishPb);

};

heart.Start();

}

private void eat()

{

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

{

if ((playerPb.Location.X + playerPb.Width) >= (foodPb[i].Location.X) &&

playerPb.Location.X <= (foodPb[i].Location.X + foodPb[i].Width) &&

(playerPb.Location.Y + playerPb.Height) >= foodPb[i].Location.Y &&

playerPb.Location.Y <= (foodPb[i].Location.Y + foodPb[i].Height))

{

if (player.Type == "small")

{

if (dbfood[i].Type == "small" && dbfood[i] != null)

{

player.Ctrsmall += 1;

player.Score += 20;

if (player.Ctrsmall == 10)

{

player.Ctrsmall = 0;

player.Type = "medium";

playerPb.Size = new Size(100, 80);

}

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

foodPb.RemoveAt(i);

dbfood.RemoveAt(i);

}

else if (dbfood[i].Type == "medium" && dbfood[i] != null)

{

playerPb.Left = 0;

hp -= 1;

if (hp <= 0)

{

hp = 0;

endMenu();

}

}

else if (dbfood[i].Type == "big" && dbfood[i] != null)

{

playerPb.Left = 0;

hp -= 1;

if (hp <= 0)

{

hp = 0;

endMenu();

}

}

else if (dbfood[i].Type == "whale" && dbfood[i] != null)

{

playerPb.Left = 0;

hp -= 1;

if (hp <= 0)

{

hp = 0;

endMenu();

}

}

else

{

hp += 1;

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

foodPb.RemoveAt(i);

dbfood.RemoveAt(i);

}

}

else if (player.Type == "medium")

{

if (dbfood[i].Type == "small" && dbfood[i] != null)

{

player.Ctrsmall += 1;

player.Score += 20;

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

foodPb.RemoveAt(i);

dbfood.RemoveAt(i);

}

else if (dbfood[i].Type == "medium" && dbfood[i] != null)

{

player.Ctrmedium += 1;

player.Score += 50;

foodPb.RemoveAt(i);

dbfood.RemoveAt(i);

}

else if (dbfood[i].Type == "big" && dbfood[i] != null)

{

playerPb.Left = 0;

hp -= 1;

if (hp <= 0)

{

hp = 0;

endMenu();

}

}

else if (dbfood[i].Type == "whale" && dbfood[i] != null)

{

playerPb.Left = 0;

hp -= 1;

if (hp <= 0)

{

hp = 0;

endMenu();

}

}

else

{

hp += 1;

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

foodPb.RemoveAt(i);

dbfood.RemoveAt(i);

}

if (player.Ctrsmall >= 5 && player.Ctrmedium >= 10)

{

player.Type = "big";

playerPb.Size = new Size(160, 100);

player.Ctrsmall = 0;

player.Ctrmedium = 0;

}

}

else if (player.Type == "big")

{

if (dbfood[i].Type == "small" && dbfood[i] != null) player.Score += 20;

else if (dbfood[i].Type == "medium" && dbfood[i] != null) player.Score += 50;

else if (dbfood[i].Type == "big" && dbfood[i] != null) player.Score += 100;

else if (dbfood[i].Type == "whale" && dbfood[i] != null)

{

playerPb.Left = 0;

hp -= 1;

if (hp <= 0)

{

hp = 0;

endMenu();

}

}

else

{

hp += 1;

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

foodPb.RemoveAt(i);

dbfood.RemoveAt(i);

}

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

foodPb.RemoveAt(i);

dbfood.RemoveAt(i);

}

resources();

HighScore.Add(score);

}

}

}

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

{

menuPanel.Visible = false;

hp = 3;

score = 0;

small = new Timer();

medium = new Timer();

big = new Timer();

heart = new Timer();

foodPb = new List<PictureBox>();

dbfood = new List<fish>();

HighScore = new List<int>();

this.Width = 1000;

this.Height = 600;

small = new Timer();

medium = new Timer();

big = new Timer();

danger = new Timer();

whale = new Timer();

gerakPaus = new Timer();

startUp();

resources();

spawnSmall();

spawnMedium();

spawnBig();

spawnDanger();

spawnWhale();

spawnHeart();

}

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

{

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

{

HighScore.Sort();

scoreList.Items.Add((i + 1) + ". " + HighScore[i]);

}

scoreList.Visible = true;

}

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

{

this.Close();

}

private void Form1\_MouseMove(object sender, MouseEventArgs e)

{

playerPb.Location = new Point(e.X, e.Y);

player.X = playerPb.Location.X;

player.Y = playerPb.Location.Y;

eat();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace cLibrary

{

public class fish

{

Random r = new Random();

private string type;

private int x, y;

private int score;

private int ctrsmall;

private int ctrmedium;

public fish(int x, int y)

{

this.type = "small";

this.score = 0;

this.x = x;

this.y = y;

this.ctrsmall = 0;

this.ctrmedium = 0;

}

public fish(string type)

{

if (type == "small")

{

int x = r.Next(1, 800);

int y = r.Next(1, 400);

this.type = type;

this.x = x;

this.y = y;

}

else if (type == "medium")

{

int x = r.Next(1, 800);

int y = r.Next(1, 400);

this.type = type;

this.x = x;

this.y = y;

}

else if (type == "big")

{

int x = r.Next(1, 800);

int y = r.Next(1, 400);

this.type = type;

this.x = x;

this.y = y;

}

else if (type == "whale")

{

int x = r.Next(1, 800);

int y = r.Next(1, 400);

this.type = type;

this.x = x;

this.y = y;

}

else if (type == "powerup")

{

int x = r.Next(1, 800);

int y = r.Next(1, 400);

this.type = type;

this.x = x;

this.y = y;

}

}

public string Type

{

set { this.type = value; }

get { return this.type; }

}

public int Score

{

set { this.score = value; }

get { return this.score; }

}

public int X

{

set { this.x = value; }

get { return this.x; }

}

public int Y

{

set { this.y = value; }

get { return this.y; }

}

public int Ctrsmall

{

set { this.ctrsmall = value; }

get { return this.ctrsmall; }

}

public int Ctrmedium

{

set { this.ctrmedium = value; }

get { return this.ctrmedium; }

}

}

}