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 PVM4\_6615

{

public partial class Form1 : Form

{

Random r = new Random();

int xmouse= 0, ymouse= 0;

int score = 0;

Rectangle scoreBox;

Timer tuniversal = new Timer();

Timer tfood = new Timer();

snake player;

List<snake> enemies;

List<food> foods;

bool move = false;

bool gameover = false;

public Form1()

{

InitializeComponent();

setLayer();

scoreBox = new Rectangle(this.Width - 100, 0, 150, 60);

player = new snake();

enemies = new List<snake>();

foods = new List<food>();

this.MouseMove += Form1\_MouseMove;

this.Paint += Form1\_Paint;

tuniversal.Tick += Tuniversal\_Tick;

spawnFood();

spawnSnakes();

tuniversal.Start();

}

private void setLayer()

{

this.DoubleBuffered = true;

this.MaximizeBox = false;

this.WindowState = FormWindowState.Maximized;

this.BackgroundImage = Image.FromFile("bg.jpg");

}

private void tableScore(Graphics g)

{

Font f = new Font(FontFamily.GenericMonospace, 40, FontStyle.Regular);

Brush b = new SolidBrush(Color.IndianRed);

Point p = new Point(this.Width - 200, 0);

g.DrawString(score.ToString(), f, b, p);

}

private void spawnFood()

{

tfood.Interval = 1500;

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

{

food ftemp = new food(this.Width, this.Height);

foods.Add(ftemp);

this.Refresh();

};

tfood.Start();

}

private void spawnSnakes()

{

for (int i = 0; i < 5; i++)

{

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

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

snake etemp = new snake(x, y);

enemies.Add(etemp);

this.Refresh();

}

}

private void movePlayer()

{

if (move == true)

{

for (int i = player.Body.Count - 1; i > 0; i--)

{

player.Body[i] = player.Body[i - 1];

}

move = false;

}

player.Body[0] = new Rectangle(xmouse, ymouse, 30, 30);

player.eat(foods, ref score);

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

{

player.die(enemies[i], ref gameover);

}

if (gameover == true)

{

tuniversal.Stop();

tfood.Stop();

MessageBox.Show("Game Over!" + "\n" + "Score: " + score);

}

}

private void moveSnakes()

{

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

{

int rmove = r.Next(1, 5);

enemies[i].moveSnakes(rmove, this.Width - 100, this.Height - 100);

enemies[i].snakesEat(foods, rmove);

if (enemies[i].collideWithPlayer(player) == true)

{

if (gameover == false)

{

score += 100;

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

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

enemies.RemoveAt(i);

enemies.Add(new snake(x, y));

}

}

}

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

{

for (int j = 0; j < enemies.Count; j++)

{

if (i != j)

{

if (enemies[i].canibal(enemies[j]) == true)

{

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

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

enemies.RemoveAt(i);

enemies.Add(new snake(x, y));

}

}

}

}

}

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

{

tuniversal.Interval = 100;

movePlayer();

moveSnakes();

this.Refresh();

}

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

{

move = true;

xmouse = e.X;

ymouse = e.Y;

}

private void Form1\_Paint(object sender, PaintEventArgs e)

{

Graphics g = e.Graphics;

tableScore(g);

player.draw(g);

if (foods.Count > 0)

{

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

{

foods[i].draw(g);

}

}

if (enemies.Count > 0)

{

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

{

enemies[i].drawSnakes(g);

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Drawing;

namespace cLibrary

{

public class snake

{

Random r = new Random();

private int tailsize;

private int x, y;

private Brush brush;

private Color color;

private List<Rectangle> body = new List<Rectangle>();

public snake()

{

this.tailsize = 1;

for (int i = 0; i < this.tailsize; i++)

{

Rectangle btemp = new Rectangle(this.x + (i \* 30), this.y, 30, 30);

this.body.Add(btemp);

}

}

public snake(int x, int y)

{

this.tailsize = r.Next(3, 6);

this.x = x;

this.y = y;

for (int i = 0; i < this.tailsize; i++)

{

Rectangle btemp = new Rectangle(this.x + (i \* 30), this.y, 30, 30);

this.body.Add(btemp);

}

}

public void draw(Graphics g)

{

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

{

if (i == 0)

{

this.color = Color.Red;

this.brush = new SolidBrush(this.color);

}

else

{

this.color = Color.Green;

this.brush = new SolidBrush(this.color);

}

g.FillEllipse(this.brush, this.body[i]);

}

}

public void drawSnakes(Graphics g)

{

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

{

if (i == 0)

{

this.color = Color.Red;

this.brush = new SolidBrush(this.color);

}

else

{

this.color = Color.Blue;

this.brush = new SolidBrush(this.color);

}

g.FillEllipse(this.brush, this.body[i]);

}

}

public void moveSnakes(int rmove, int w, int h)

{

for (int i = this.body.Count - 1; i > 0; i--)

{

this.body[i] = this.body[i - 1];

this.x = this.body[i].X;

this.y = this.body[i].Y;

}

Rectangle head = this.body[0];

if (rmove == 1) //kanan

{

this.body[0] = new Rectangle(head.X + 30, head.Y, 30, 30);

if (perfectMove(this.body[0]) == true) this.body[0] = new Rectangle(head.X - 30, head.Y, 30, 30);

}

else if (rmove == 2) //kiri

{

this.body[0] = new Rectangle(head.X - 30, head.Y, 30, 30);

if (perfectMove(this.body[0]) == true) this.body[0] = new Rectangle(head.X + 30, head.Y, 30, 30);

}

else if (rmove == 3) //atas

{

this.body[0] = new Rectangle(head.X, head.Y - 30, 30, 30);

if (perfectMove(this.body[0]) == true) this.body[0] = new Rectangle(head.X, head.Y + 30, 30, 30);

}

else if (rmove == 4) //bawah

{

this.body[0] = new Rectangle(head.X, head.Y + 30, 30, 30);

if (perfectMove(this.body[0]) == true) this.body[0] = new Rectangle(head.X, head.Y - 30, 30, 30);

}

if (this.body[0].X > w) this.body[0] = new Rectangle(head.X - 30, head.Y, 30, 30);

else if (this.body[0].X < 0) this.body[0] = new Rectangle(head.X + 30, head.Y, 30, 30);

if (this.body[0].Y > h) this.body[0] = new Rectangle(head.X, head.Y - 30, 30, 30);

else if (this.body[0].Y < 0) this.body[0] = new Rectangle(head.X, head.Y + 30, 30, 30);

}

private bool perfectMove(Rectangle head)

{

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

{

if (i != 0) if (head.IntersectsWith(this.body[i])) return true;

}

return false;

}

public void snakesEat(List<food> f, int direct)

{

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

{

if (this.body[0].IntersectsWith(f[i].Shape))

{

if (direct == 1) this.body.Add(new Rectangle(this.x + (this.body.Count \* 30), this.y, 30, 30));

else if (direct == 2) this.body.Add(new Rectangle(this.x - (this.body.Count \* 30), this.y, 30, 30));

else if (direct == 3) this.body.Add(new Rectangle(this.x, this.y - (this.body.Count \* 30), 30, 30));

else if (direct == 4) this.body.Add(new Rectangle(this.x, this.y + (this.body.Count \* 30), 30, 30));

f.RemoveAt(i);

}

}

}

public bool canibal(snake s)

{

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

{

if (this.body[0].IntersectsWith(s.body[i])) return true;

}

return false;

}

public bool collideWithPlayer(snake p)

{

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

{

for (int j = 0; j < p.body.Count; j++)

{

if (this.body[i].IntersectsWith(p.body[j])) return true;

}

}

return false;

}

public void eat(List<food> f, ref int score)

{

if (f.Count > 0)

{

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

{

if (body[0].IntersectsWith(f[i].Shape))

{

if (f[i].Type == "small") score += 20;

else if (f[i].Type == "medium") score += 40;

else if (f[i].Type == "big") score += 60;

f.RemoveAt(i);

Rectangle btemp = new Rectangle(this.x + (this.tailsize \* 30), this.y, 30, 30);

this.body.Add(btemp);

}

}

}

}

public void die(snake s, ref bool ya)

{

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

{

for (int j = 0; j < this.body.Count; j++)

{

if (this.body[j].IntersectsWith(s.body[i]))

{

ya = true;

}

}

}

}

public int Tailsize

{

set { this.tailsize = value; }

get { return this.tailsize; }

}

public Brush Brush

{

set { this.brush = value; }

get { return this.brush; }

}

public List<Rectangle> Body

{

set { this.body = value; }

get { return this.body; }

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Drawing;

namespace cLibrary

{

public class food

{

Random r = new Random();

private int x, y;

private string type;

private Brush brush;

private Color color;

private Rectangle shape;

public food(int w, int h)

{

this.x = r.Next(w);

this.y = r.Next(h);

this.color = Color.FromArgb(r.Next(256), r.Next(256), r.Next(256));

this.brush = new SolidBrush(this.color);

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

if (rfood == 1)

{

this.type = "small";

this.shape = new Rectangle(this.x - 100, this.y - 100, 20, 20);

}

else if (rfood == 2)

{

this.type = "medium";

this.shape = new Rectangle(this.x, this.y, 30, 30);

}

else if (rfood == 3)

{

this.type = "big";

this.shape = new Rectangle(this.x, this.y, 40, 40);

}

}

public void draw(Graphics g)

{

g.FillEllipse(this.brush, this.shape);

}

public String Type

{

set { this.type = value; }

get { return this.type; }

}

public Rectangle Shape

{

set { this.shape = value; }

get { return this.shape; }

}

}

}