Problem 10

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;

namespace Problem10

{

public class CNode

{

public int X, Y;

}

public partial class Form1 : Form

{

List<CNode> LTop = new List<CNode>();

List<CNode> LBottom = new List<CNode>();

int flag = 0;

int pos = 0;

int up = 0; //up not clicked

int top = 0;

int bottom = 0;

int line = 0;

public Form1()

{

this.BackColor = Color.FromArgb(255, 255, 255);

this.MouseDown += MyMouseDown;

this.Paint += Form1\_Paint;

this.KeyDown += Form1\_KeyDown;

}

int posUp=0, posDown=0;

int prevUp=0, prevDown=0;

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

MoveUp(posUp, prevUp);

MoveDown(posDown,prevDown);

switch (e.KeyCode)

{

case Keys.Up:

prevDown = posDown;

posDown++;

if (posDown == LBottom.Count)

{

posDown = 0;

}

break;

case Keys.Down:

prevDown = posDown;

posDown--;

if (posDown < 0)

{

posDown = LBottom.Count - 1;

}

break;

case Keys.Right:

prevUp = posUp;

posUp++;

if (posUp == LTop.Count)

{

posUp = 0;

}

break;

case Keys.Left:

prevUp = posUp;

posUp--;

if (posUp < 0)

{

posUp = LTop.Count - 1;

}

break;

}

}

private void MyMouseDown(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

if (flag == 0)

{

pos = e.Y;

this.Text = pos.ToString();

flag = 1;

line = 1;

}

else

{

if (up == 0 && e.Y < pos)

{

CNode pnn = new CNode();

pnn.X = e.X;

pnn.Y = e.Y;

LTop.Add(pnn);

up = 1; //up clicked

DrawEllipse(e.X,e.Y,up);

}

else if (up == 1 && e.Y > pos)

{

CNode pnn = new CNode();

pnn.X = e.X;

pnn.Y = e.Y;

LBottom.Add(pnn);

up = 0; //remove up click to make it not clicked

DrawEllipse(e.X, e.Y, up);

}

else

{

MessageBox.Show("Error");

}

}

}

else

{

if (top == 0)

{

MessageBox.Show("Top Points: ");

}

while (top < LTop.Count)

{

MessageBox.Show($"{LTop[top].X}, {LTop[top].Y}");

top++;

}

if (bottom == 0)

{

MessageBox.Show("Bottom Points: ");

}

while (bottom < LBottom.Count)

{

MessageBox.Show($"{LBottom[bottom].X}, {LBottom[bottom].Y}");

bottom++;

}

}

}

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

{

if (line == 1)

{

Graphics g = e.Graphics;

Pen p = new Pen(Color.Black, 3); //3 for width

g.DrawLine(p, 0, pos, this.ClientSize.Width, pos);

}

}

void DrawEllipse(int x, int y, int up)

{

Graphics g = this.CreateGraphics();

SolidBrush brush;

if (up == 1)

{

brush = new SolidBrush(Color.RoyalBlue);

}

else

{

brush = new SolidBrush(Color.Maroon);

}

g.FillEllipse(brush, x, y, 10, 10);

}

void MoveUp(int currentPos, int prevPos)

{

Graphics g = this.CreateGraphics();

SolidBrush brushCurrent = new SolidBrush(Color.Gray);

SolidBrush brushPrev = new SolidBrush(Color.RoyalBlue);

g.FillEllipse(brushCurrent, LTop[currentPos].X, LTop[currentPos].Y, 10, 10);

g.FillEllipse(brushPrev, LTop[prevPos].X, LTop[prevPos].Y, 10, 10);

}

void MoveDown(int currentPos, int prevPos)

{

Graphics g = this.CreateGraphics();

SolidBrush brushCurrent = new SolidBrush(Color.Gray);

SolidBrush brushPrev = new SolidBrush(Color.Maroon);

g.FillEllipse(brushCurrent, LBottom[currentPos].X, LBottom[currentPos].Y, 10, 10);

g.FillEllipse(brushPrev, LBottom[prevPos].X, LBottom[prevPos].Y, 10, 10);

}

}

}

Problem 11

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;

namespace Problem11

{

public partial class Form1 : Form

{

bool isDrag = false;

int xOld = 0, yOld = 0, dx = 0, dy = 0;

int draw = 0;

public class CActor

{

public int X, Y;

public int W, H;

public Color cl;

}

List<CActor> Circles = new List<CActor> ();

public Form1()

{

this.MouseDown += Form1\_MouseDown;

this.MouseUp += Form1\_MouseUp;

this.MouseMove += Form1\_MouseMove;

this.Paint += Form1\_Paint;

this.BackColor = Color.LightGray;

this.KeyDown += Form1\_KeyDown;

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

switch(e.KeyCode)

{

case Keys.Space:

draw = 1;

DrawScene();

break;

}

}

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

{

if (draw == 1)

{

DrawScene();

}

}

private void DrawScene()

{

Graphics g = CreateGraphics();

g.Clear(Color.LightGray);

for (int i = 0; i < 4; i++)

{

CActor pnn = new CActor();

pnn.X = 120 + 20 \* i;

pnn.W = 20;

pnn.Y = 200;

pnn.H = 20;

pnn.cl = Color.RoyalBlue;

Circles.Add(pnn);

SolidBrush brush = new SolidBrush(pnn.cl);

g.FillEllipse(brush, pnn.X + dx \* i, pnn.Y + dy, pnn.W, pnn.H);

}

for (int i = 0; i < 4; i++)

{

CActor pnn = new CActor();

pnn.X = 400 + 20 \* i;

pnn.W = 20;

pnn.Y = 200;

pnn.H = 20;

pnn.cl = Color.RoyalBlue;

Circles.Add(pnn);

SolidBrush brush = new SolidBrush(pnn.cl);

g.FillEllipse(brush, pnn.X + dx \* i, pnn.Y + dy, pnn.W, pnn.H);

}

for (int i = 0; i < 8; i++)

{

CActor pnn = new CActor();

pnn.X = 220 + 20 \* i;

pnn.W = 20;

pnn.Y = 110;

pnn.H = 20;

pnn.cl = Color.LightPink;

Circles.Add(pnn);

SolidBrush brush = new SolidBrush(pnn.cl);

g.FillEllipse(brush, pnn.X + dx \* i, pnn.Y + dy, pnn.W, pnn.H);

}

for (int i = 0; i < 8; i++)

{

CActor pnn = new CActor();

pnn.X = 220 + 20 \* i;

pnn.W = 20;

pnn.Y = 280;

pnn.H = 20;

pnn.cl = Color.LightPink;

Circles.Add(pnn);

SolidBrush brush = new SolidBrush(pnn.cl);

g.FillEllipse(brush, pnn.X + dx \* i, pnn.Y + dy, pnn.W, pnn.H);

}

}

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

{

isDrag = false;

}

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

{

if (isDrag == true)

{

dx = e.X - xOld;

dy = e.Y - yOld;

Graphics g = CreateGraphics();

g.Clear(Color.LightGray);

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

{

Circles[i].X += dx;

Circles[i].Y += dy;

SolidBrush brush = new SolidBrush(Circles[i].cl);

g.FillEllipse(brush, Circles[i].X + dx \* i, Circles[i].Y + dy, Circles[i].W, Circles[i].H);

}

xOld = e.X;

yOld = e.Y;

}

}

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

{

if (e.Button == MouseButtons.Left)

{

isDrag = true;

xOld = e.X;

yOld = e.Y;

}

}

}

}

Problem 12

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;

namespace Problem12

{

public partial class Form1 : Form

{

public class CActor

{

public int X, Y;

public int W, H;

public Color cl;

}

List<CActor> Circles = new List<CActor>();

List<CActor> SquaresLeft = new List<CActor>();

List<CActor> SquaresRight = new List<CActor>();

int i = 0;

int ctLastRow = 0;

int row = 0;

int numBlocks1, numBlocks2;

public Form1()

{

InitializeComponent();

this.KeyDown += Form1\_KeyDown;

this.Paint += Form1\_Paint;

this.BackColor = Color.Black;

Random rr = new Random();

numBlocks1 = rr.Next(5, 15);

numBlocks2 = rr.Next(5, 10);

}

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

{

Graphics g = CreateGraphics();

Pen p = new Pen(Color.Orange);

g.DrawLine(p, 20, 30, 300, 30); //up

g.DrawLine(p, 20, 300, 300, 300); //down

g.DrawLine(p, 20, 30, 20, 300); //left

g.DrawLine(p, 300, 30, 300, 300); //right

g.DrawLine(p, 350, 30, 630, 30); //up

g.DrawLine(p, 350, 300, 630, 300); //down

g.DrawLine(p, 350, 30, 350, 300); //left

g.DrawLine(p, 630, 30, 630, 300); //right

int xPos = 20;

int yPos = 30;

for (int j = 0; j < numBlocks1; j++)

{

CActor square = new CActor();

square.X = xPos;

square.Y = yPos;

square.W = 20;

square.H = 20;

square.cl = Color.Orange;

if (xPos + square.W > 265)

{

row++;

xPos = 20;

yPos = 30 + row \* 25;

}

else

{

xPos += 25;

}

SquaresLeft.Add(square);

Pen P = new Pen(square.cl);

g.DrawRectangle(P, square.X, square.Y, square.W, square.H);

SolidBrush brush = new SolidBrush(square.cl);

g.FillRectangle(brush, square.X, square.Y, square.W, square.H);

}

row = 0;

xPos = 350;

yPos = 30;

ctLastRow = 0;

for (int j = 0; j < numBlocks2; j++)

{

CActor square = new CActor();

square.X = xPos;

square.Y = yPos;

square.W = 20;

square.H = 20;

square.cl = Color.Pink;

if (xPos + square.W > 595)

{

row++;

ctLastRow = 0;

xPos = 350;

yPos = 30 + row \* 25;

}

else

{

xPos += 25;

ctLastRow++;

}

SquaresRight.Add(square);

Pen P = new Pen(square.cl);

g.DrawRectangle(P, square.X, square.Y, square.W, square.H);

SolidBrush brush = new SolidBrush(square.cl);

g.FillRectangle(brush, square.X, square.Y, square.W, square.H);

}

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

Graphics g = CreateGraphics();

int startX = 350 + (ctLastRow \* 25);

int startY = 30 + row \* 25;

switch (e.KeyCode)

{

case Keys.Enter:

if (startX + 25 \* i < 600)

{

CActor pnn = new CActor();

pnn.X = startX + 25 \* i;

pnn.W = 20;

pnn.Y = startY;

pnn.H = 20;

pnn.cl = Color.RoyalBlue;

Circles.Add(pnn);

SolidBrush brush = new SolidBrush(pnn.cl);

g.FillEllipse(brush, pnn.X, pnn.Y, pnn.W, pnn.H);

i++;

}

else

{

row++;

ctLastRow = 0;

startX = 350;

startY = 30 + row \* 25;

i = 0;

}

break;

case Keys.Space:

if (SquaresLeft.Count == SquaresRight.Count + Circles.Count)

{

MessageBox.Show("Correct");

}

else

{

MessageBox.Show("Error");

}

break;

}

}

}

}