МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Лабораторная работа №7

по дисциплине «Визуальное программирование»

Выполнил студент Пшеничный Д.О.

Факультет, группа ФКФН, ПО(аб)-81

Проверила Резак Е.В.

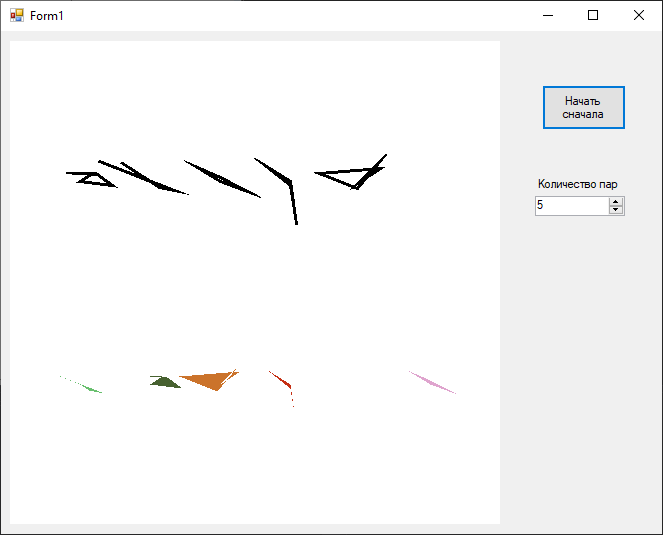
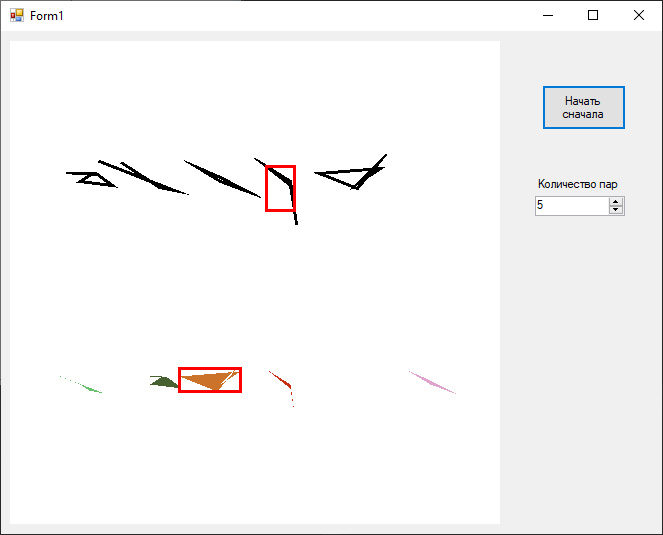
Хабаровск – 2021г.

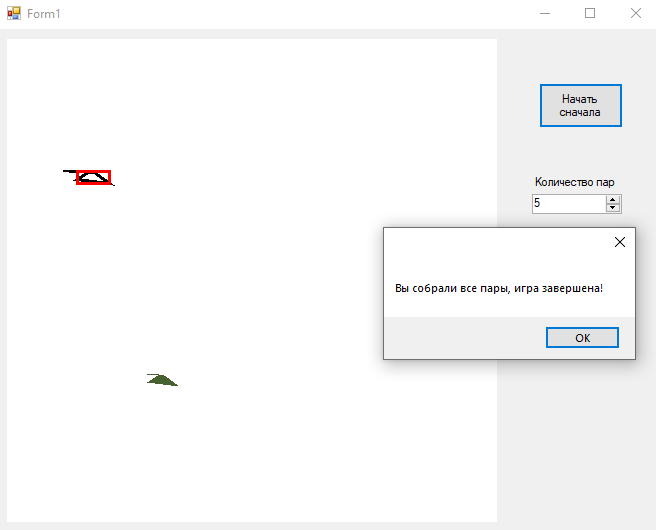
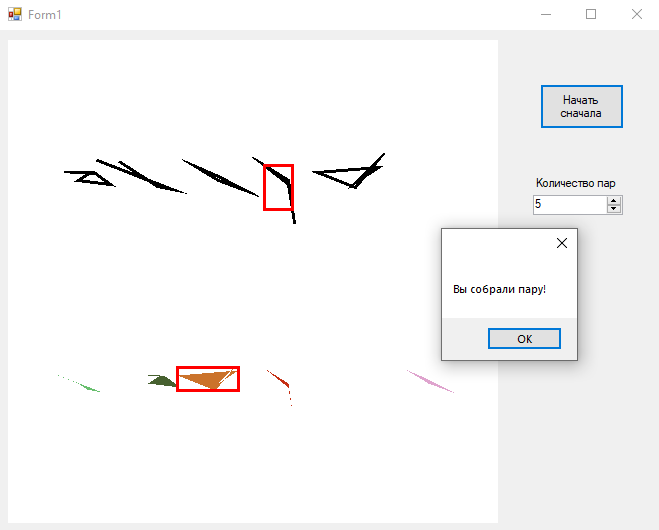
#### Вариант 17

Написать игру "Расставь фигуры".

Правила игры: необходимо заполнить пустые ячейки, расположенные на

игровом поле, подходящими геометрическими фигурами



**Исходный код программы**

public partial class Form1 : Form

{

Graphics graphics;

Pen blackPen;

Pen redPen;

Random random;

int selectedHole = -1;

int selectedFigure = -1;

Rectangle holeRectangle;

Rectangle figureRectangle;

List<Hole> holeList = new List<Hole>();

List<Figure> figureList = new List<Figure>();

public Form1()

{

InitializeComponent();

graphics = panel1.CreateGraphics();

blackPen = new Pen(Color.Black, 3f);

redPen = new Pen(Color.Red, 3f);

random = new Random();

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

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

{

holeList.Clear();

figureList.Clear();

selectedFigure = -1;

selectedHole = -1;

int spaceValue = 70;

for(int i = 0; i < numericUpDown1.Value; i++)

{

Point[] poly = GeneratePoly(random.Next(4, 6));

holeList.Add(new Hole(

new Point(spaceValue + spaceValue \* i, 2 \* spaceValue),

poly,

Color.Black,

i));

figureList.Add(new Figure(

new Point(spaceValue + spaceValue \* i, panel1.Height - 2 \* spaceValue),

poly,

Color.FromArgb(random.Next(256), random.Next(256), random.Next(256)),

i));

}

List<int> places = new List<int>();

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

{

bool ch = true;

int place = 0;

while (ch)

{

ch = false;

place = random.Next(0, 6);

foreach (var k in places)

if (k == place)

{

ch = true;

break;

}

}

places.Add(place);

figureList[j].SetPos(spaceValue + spaceValue \* place, panel1.Height - 2 \* spaceValue);

}

DrawAll();

}

private void DrawAll()

{

graphics.Clear(Color.White);

foreach (var i in holeList)

graphics.DrawPolygon(blackPen, i.Points);

foreach (var i in figureList)

graphics.FillPolygon(new SolidBrush(i.FillColor), i.FixedPoly());

if(selectedHole != -1)

graphics.DrawRectangle(redPen, holeRectangle);

if (selectedFigure != -1)

graphics.DrawRectangle(redPen, figureRectangle);

}

private Point[] GeneratePoly(int pointsNum)

{

Point[] points = new Point[pointsNum + 1];

points[0] = new Point(0, 0);

for(int i = 1; i < pointsNum; i++)

{

int radius = random.Next(15, 31);

int angle = random.Next(30, 35);

points[i] = new Point((int)((radius + points[i - 1].X) \* Math.Cos(-angle)), (int)((radius + points[i - 1].Y) \* Math.Sin(-angle)));

}

points[pointsNum] = points[0];

return points;

}

private void panel1\_MouseClick(object sender, MouseEventArgs e)

{

foreach(var i in holeList)

{

if (e.X >= i.GetLeftTop().X && e.X <= i.GetRightBottom().X && e.Y >= i.GetLeftTop().Y && e.Y <= i.GetRightBottom().Y)

{

holeRectangle = new Rectangle(i.GetLeftTop(), new Size(i.GetRightBottom().X - i.GetLeftTop().X,

i.GetRightBottom().Y - i.GetLeftTop().Y));

selectedHole = i.PairInd;

}

}

foreach (var i in figureList)

{

if (e.X >= i.GetLeftTop().X && e.X <= i.GetRightBottom().X && e.Y >= i.GetLeftTop().Y && e.Y <= i.GetRightBottom().Y)

{

figureRectangle = new Rectangle(i.GetLeftTop(), new Size(i.GetRightBottom().X - i.GetLeftTop().X,

i.GetRightBottom().Y - i.GetLeftTop().Y));

selectedFigure = i.PairInd;

}

}

if (selectedFigure == selectedHole && selectedHole != -1)

Match(selectedFigure);

DrawAll();

}

private void Match(int pairInd)

{

foreach(var i in holeList)

{

if (i.PairInd == pairInd)

{

holeList.Remove(i);

break;

}

}

foreach (var i in figureList)

{

if (i.PairInd == pairInd)

{

figureList.Remove(i);

break;

}

}

selectedFigure = -1;

selectedHole = -1;

MessageBox.Show("Вы собрали пару!");

if (holeList.Count == 0)

MessageBox.Show("Вы собрали все пары, игра завершена!");

}

private void label2\_Click(object sender, EventArgs e)

{

}

}