Министерство образования Республики Беларусь

Министерство образования и науки Российской Федерации

МУВПО “Белорусско-Российский университет”

Кафедра “ПОИТ”

Отчет по

Лабораторной работе №7

Аффинные преобразования

Выполнил: ст. гр. АСОИ-181

Самусев С.В.

Преподаватель

Шилов А.В.

Могилёв 2020

Цель работы: изучение алгоритмов параллельного сдвига, растяжения – сжатия и поворота графического объекта на плоскости и в пространстве.

Практическое задание:

Разработать программу для перемещения, сжатия, поворота объекта №1.

using System;

using System.Drawing;

using System.Windows.Forms;

using System.Drawing.Drawing2D;

namespace z1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

this.BackColor = Color.White;

this.KeyPreview = true;

panel1.Paint += new PaintEventHandler(panel1Paint);

}

private void panel1Paint(object sender, PaintEventArgs e)

{

Graphics g = e.Graphics;

DrawAxis(g);

ApplyTransformation(g);

}

private void ApplyTransformation(Graphics g)

{

Matrix m = new Matrix();

m.Translate(panel1.Width / 2, panel1.Height / 2);

if (rbTranslation.Checked)

{

int dx = Convert.ToInt16(tbTranslationX.Text);

int dy = -Convert.ToInt16(tbTranslationY.Text);

m.Translate(dx, dy);

}

else if (rbScale.Checked)

{

float sx = Convert.ToSingle(tbScaleX.Text);

float sy = Convert.ToSingle(tbScaleY.Text);

m.Scale(sx, sy);

}

else if (rbRotation.Checked)

{

float angle = Convert.ToSingle(tbRotaionAngle.Text);

float x = Convert.ToSingle(tbRotateAtX.Text);

float y = -Convert.ToSingle(tbRotateAtY.Text);

g.FillEllipse(Brushes.Black, x - 4, y - 4, 8, 8);

m.RotateAt(angle, new PointF(x, y));

}

else if (rbShear.Checked)

{

float alpha = Convert.ToSingle(tbShearX.Text);

float beta = Convert.ToSingle(tbShearY.Text);

m.Shear(alpha, beta);

}

g.Transform = m;

DrawFigure(g, Color.Black);

}

private void DrawFigure(Graphics g, Color color)

{

Pen p = new Pen(Color.Navy, 3);

int x1 = -100, y1 = 0, x2 = 0, y2 =-100;

g.DrawLine(p, x1, y1, x2, y2);

x1 = 0; y1 = -100; x2 = 100; y2 = 0;

g.DrawLine(p, x1, y1, x2, y2);

x1 = -100; y1 = 0; x2 = -50;y2 = 0;

g.DrawLine(p, x1, y1, x2, y2);

x1 = -50; y1 = 0; x2 = -50; y2 = 100;

g.DrawLine(p, x1, y1, x2, y2);

x1 = -50; y1 = 100; x2 = 50; y2 = 100;

g.DrawLine(p, x1, y1, x2, y2);

x1 = 50; y1 = 100; x2 = 50; y2 = 0;

g.DrawLine(p, x1, y1, x2, y2);

x1 = 50; y1 = 0; x2 = 100; y2 = 0;

g.DrawLine(p, x1, y1, x2, y2);

}

private void DrawAxis(Graphics g)

{

Matrix m = new Matrix();

m.Translate(panel1.Width / 2, panel1.Height / 2);

g.Transform = m;

g.DrawLine(Pens.Blue, -panel1.Width / 2, 0,panel1.Width / 2, 0);

g.DrawLine(Pens.Blue, 0, -panel1.Height / 2,0, panel1.Height / 2);

g.DrawString("X", this.Font, Brushes.Blue, panel1.Width / 2 - 20, -20);

g.DrawString("Y", this.Font, Brushes.Blue, 5, -panel1.Height / 2 + 5);

int tick = 40;

StringFormat sf = new StringFormat();

sf.Alignment = StringAlignment.Far;

for (int i = -200; i <= 200; i += tick)

{

g.DrawLine(Pens.Blue, i, -3, i, 3);

g.DrawLine(Pens.Blue, -3, i, 3, i);

SizeF sizeXTick = g.MeasureString(i.ToString(),

this.Font);

if (i != 0)

{

g.DrawString(i.ToString(), this.Font, Brushes.Blue,

i + sizeXTick.Width / 2, 4f, sf);

g.DrawString((-i).ToString(), this.Font, Brushes.Blue,

-3f, i - sizeXTick.Height / 2, sf);

}

else

{

g.DrawString("0", this.Font, Brushes.Blue,

new PointF(i - sizeXTick.Width / 3, 4f), sf);

}

}

}

private void button1\_Click\_1(object sender, EventArgs e)

{

panel1.Invalidate();

}

private void button2\_Click(object sender, EventArgs e)

{

tbTranslationX.Text = "0";

tbTranslationY.Text = "0";

tbScaleX.Text = "1";

tbScaleY.Text = "1";

tbRotaionAngle.Text = "0";

tbRotateAtX.Text = "0";

tbRotateAtY.Text = "0";

tbShearX.Text = "0";

tbShearY.Text = "0";

panelbm.Invalidate();

}

private void button1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.F3 && e.Modifiers == Keys.Alt)

{

button1.PerformClick();

}

}

private void button2\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.F2 && e.Modifiers == Keys.Alt)

{

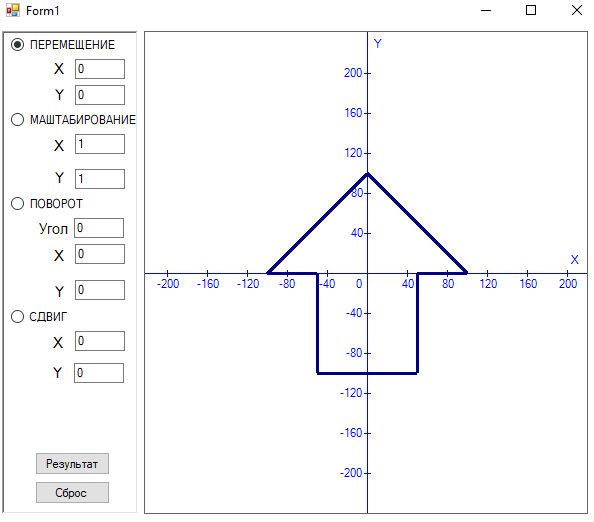
button2.PerformClick();

}

}

}

}



Блок-схема

