Министерство образования и науки Российской

**Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем

и программной инженерии

**Лабораторная работа № 9**

**по дисциплине**

**«Программирование компьютерной графики»**

**По теме: «Геометрическое преобразование в 2D»**

Выполнила:

ст. гр. ПРИ-118

Левченко В.П

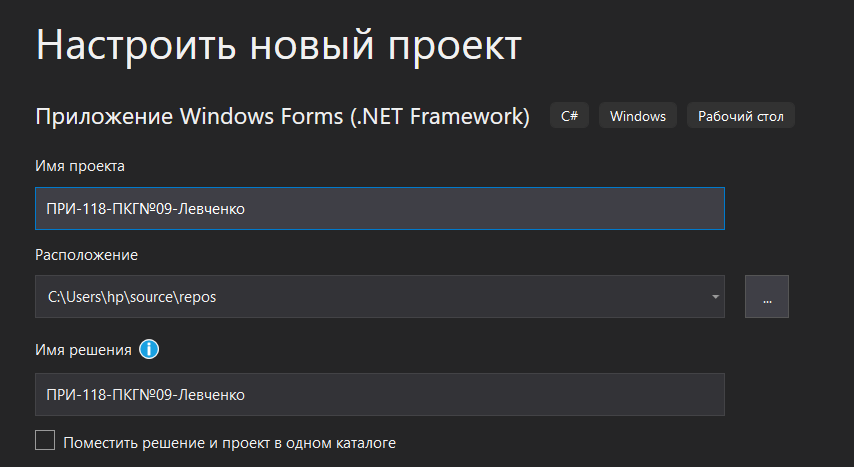
Принял:

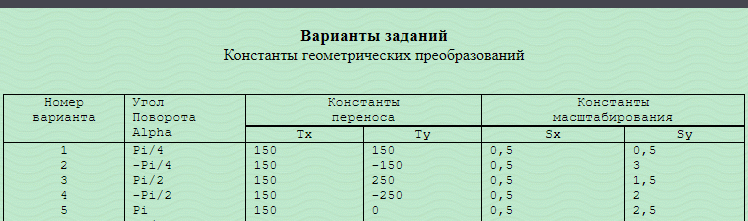
Жигалов И.Е

Владимир, 2021

**Цель работы**

Изучение методов геометрических преобразований графических объектов, приобретение навыков использования средств геометрических преобразований при составлении графических программ.

  
Рисунок 1. Создание проекта

  
Рисунок 2. Индивидуальное задание

using System;

using System.Windows.Forms;

using Tao.FreeGlut;

using Tao.OpenGl;

namespace ПРИ\_118\_ПКГ\_09\_Левченко

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

AnT.InitializeContexts();

}

private float[,] GeomObject = new float[37, 3];

private int count\_elements = 0;

//начальный угол поворота, константы переноса и масштабирования

private float angle\_of\_rotation = (float)3.14;

private float Tx = 1.50f;

private float Ty = 0f;

private float Sx = 0.5f;

private float Sy = 2.5f;

private void Form1\_Load(object sender, EventArgs e)

{

Glut.glutInit();

Glut.glutInitDisplayMode(Glut.GLUT\_RGB | Glut.GLUT\_DOUBLE | Glut.GLUT\_DEPTH);

Gl.glClearColor(255, 255, 255, 1);

Gl.glViewport(0, 0, AnT.Width, AnT.Height);

Gl.glMatrixMode(Gl.GL\_PROJECTION);

Gl.glLoadIdentity();

Glu.gluPerspective(45, (float)AnT.Width / (float)AnT.Height, 0.1, 200);

Gl.glMatrixMode(Gl.GL\_MODELVIEW);

Gl.glLoadIdentity();

Gl.glEnable(Gl.GL\_DEPTH\_TEST);

GeomObject[2, 0] = 0.7f;//

GeomObject[2, 1] = 0.0f;

GeomObject[2, 2] = 0.0f;

GeomObject[3, 0] = 0.9f;//

GeomObject[3, 1] = 0.2f;

GeomObject[3, 2] = 0.0f;

GeomObject[4, 0] = 0.9f;//

GeomObject[4, 1] = 1.2f;

GeomObject[4, 2] = 0.0f;

GeomObject[5, 0] = 0.5f;

GeomObject[5, 1] = 1.2f;

GeomObject[5, 2] = 0.0f;

GeomObject[6, 0] = 0.5f;

GeomObject[6, 1] = 1.3f;

GeomObject[6, 2] = 0.0f;

GeomObject[7, 0] = 0.9f;

GeomObject[7, 1] = 1.3f;

GeomObject[7, 2] = 0.0f;

GeomObject[8, 0] = 0.9f;

GeomObject[8, 1] = 1.8f;

GeomObject[8, 2] = 0.0f;

GeomObject[9, 0] = -0.4f;

GeomObject[9, 1] = 1.8f;

GeomObject[9, 2] = 0.0f;

GeomObject[10, 0] = -0.4f;

GeomObject[10, 1] = 1.3f;

GeomObject[10, 2] = 0.0f;

GeomObject[11, 0] = 0.0f;

GeomObject[11, 1] = 1.3f;

GeomObject[11, 2] = 0.0f;

GeomObject[12, 0] = 0.0f;

GeomObject[12, 1] = 1.2f;

GeomObject[12, 2] = 0.0f;

GeomObject[13, 0] = -0.4f;

GeomObject[13, 1] = 1.2f;

GeomObject[13, 2] = 0.0f;

GeomObject[14, 0] = -0.4f;

GeomObject[14, 1] = 0.8f;

GeomObject[14, 2] = 0.0f;

GeomObject[15, 0] = 0.0f;

GeomObject[15, 1] = 0.8f;

GeomObject[15, 2] = 0.0f;

GeomObject[16, 0] = 0.0f;

GeomObject[16, 1] = 0.7f;

GeomObject[16, 2] = 0.0f;

GeomObject[17, 0] = -0.4f;

GeomObject[17, 1] = 0.7f;

GeomObject[17, 2] = 0.0f;

GeomObject[18, 0] = -0.4f;

GeomObject[18, 1] = 0.4f;

GeomObject[18, 2] = 0.0f;

GeomObject[19, 0] = -0.2f;

GeomObject[19, 1] = 0.4f;

GeomObject[19, 2] = 0.0f;

GeomObject[20, 0] = -0.2f;

GeomObject[20, 1] = 0.3f;

GeomObject[20, 2] = 0.0f;

GeomObject[21, 0] = 0.0f;

GeomObject[21, 1] = 0.3f;

GeomObject[21, 2] = 0.0f;

//квадраты

GeomObject[23, 0] = -0.3f;//

GeomObject[23, 1] = 1.7f;

GeomObject[23, 2] = 0.0f;

GeomObject[24, 0] = 0.1f;//

GeomObject[24, 1] = 1.7f;

GeomObject[24, 2] = 0.0f;

GeomObject[25, 0] = 0.1f;//

GeomObject[25, 1] = 1.4f;

GeomObject[25, 2] = 0.0f;

GeomObject[26, 0] = -0.3f;

GeomObject[26, 1] = 1.4f;

GeomObject[26, 2] = 0.0f;

GeomObject[28, 0] = 0.3f;

GeomObject[28, 1] = 0.2f;

GeomObject[28, 2] = 0.0f;

GeomObject[29, 0] = 0.7f;

GeomObject[29, 1] = 0.2f;

GeomObject[29, 2] = 0.0f;

GeomObject[30, 0] = 0.7f;

GeomObject[30, 1] = 0.6f;

GeomObject[30, 2] = 0.0f;

GeomObject[31, 0] = 0.3f;

GeomObject[31, 1] = 0.6f;

GeomObject[31, 2] = 0.0f;

GeomObject[33, 0] = 0.2f;

GeomObject[33, 1] = 0.8f;

GeomObject[33, 2] = 0.0f;

GeomObject[34, 0] = 0.4f;

GeomObject[34, 1] = 0.8f;

GeomObject[34, 2] = 0.0f;

GeomObject[35, 0] = 0.4f;

GeomObject[35, 1] = 1.1f;

GeomObject[35, 2] = 0.0f;

GeomObject[36, 0] = 0.2f;

GeomObject[36, 1] = 1.1f;

GeomObject[36, 2] = 0.0f;

count\_elements = 36;

comboBox1.SelectedIndex = 0;

RenderTimer.Start();

}

private void Draw()

{

Gl.glClear(Gl.GL\_COLOR\_BUFFER\_BIT | Gl.GL\_DEPTH\_BUFFER\_BIT);

Gl.glClearColor(255, 255, 255, 1);

Gl.glLoadIdentity();

Gl.glColor3f(0, 0, 0);

Gl.glPushMatrix();

Gl.glTranslated(0, 0, -7);

Gl.glRotated(15, 1, 1, 0);

Gl.glPushMatrix();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

for (int i = 0; i < 22; i++)

Gl.glVertex3d(GeomObject[i, 0], GeomObject[i, 1], GeomObject[i, 2]);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

for (int i = 23; i < 27; i++)

Gl.glVertex3d(GeomObject[i, 0], GeomObject[i, 1], GeomObject[i, 2] + 0.3);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

for (int i = 28; i < 32; i++)

Gl.glVertex3d(GeomObject[i, 0], GeomObject[i, 1], GeomObject[i, 2] + 0.3);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

for (int i = 33; i < 37; i++)

Gl.glVertex3d(GeomObject[i, 0], GeomObject[i, 1], GeomObject[i, 2] + 0.3);

Gl.glEnd();

Gl.glPopMatrix();

Gl.glPopMatrix();

Gl.glFlush();

AnT.Invalidate();

}

private void CreateZoom(float coef, int os)

{

float[,] Zoom3D = new float[3, 3];

Zoom3D[0, 0] = 1;

Zoom3D[1, 0] = 0;

Zoom3D[2, 0] = 0;

Zoom3D[0, 1] = 0;

Zoom3D[1, 1] = 1;

Zoom3D[2, 1] = 0;

Zoom3D[0, 2] = 0;

Zoom3D[1, 2] = 0;

Zoom3D[2, 2] = 1;

Zoom3D[os, os] = coef;

multiply(GeomObject, Zoom3D);

}

private void CreateTranslate(float translate, int os)

{

float[,] Tran2D = new float[3, 3];

Tran2D[0, 0] = 1;

Tran2D[1, 0] = 0;

Tran2D[2, 0] = 0;

Tran2D[0, 1] = 0;

Tran2D[1, 1] = 1;

Tran2D[2, 1] = 0;

Tran2D[0, 2] = 0;

Tran2D[1, 2] = 0;

Tran2D[2, 2] = 1;

Tran2D[2, os] = translate;

multiply(GeomObject, Tran2D);

}

private void CreateRotate(float angle, int os)

{

float[,] Rotate3D = new float[3, 3];

switch (os)

{

case 0:

{

Rotate3D[0, 0] = 1;

Rotate3D[1, 0] = 0;

Rotate3D[2, 0] = 0;

Rotate3D[0, 1] = 0;

Rotate3D[1, 1] = (float)Math.Cos(angle);

Rotate3D[2, 1] = (float)-Math.Sin(angle);

Rotate3D[0, 2] = 0;

Rotate3D[1, 2] = (float)Math.Sin(angle);

Rotate3D[2, 2] = (float)Math.Cos(angle);

break;

}

case 1:

{

Rotate3D[0, 0] = (float)Math.Cos(angle);

Rotate3D[1, 0] = 0;

Rotate3D[2, 0] = (float)Math.Sin(angle);

Rotate3D[0, 1] = 0;

Rotate3D[1, 1] = 1;

Rotate3D[2, 1] = 0;

Rotate3D[0, 2] = (float)-Math.Sin(angle);

Rotate3D[1, 2] = 0;

Rotate3D[2, 2] = (float)Math.Cos(angle);

break;

}

case 2:

{

Rotate3D[0, 0] = (float)Math.Cos(angle);

Rotate3D[1, 0] = (float)-Math.Sin(angle);

Rotate3D[2, 0] = 0;

Rotate3D[0, 1] = (float)Math.Sin(angle);

Rotate3D[1, 1] = (float)Math.Cos(angle);

Rotate3D[2, 1] = 0;

Rotate3D[0, 2] = 0;

Rotate3D[1, 2] = 0;

Rotate3D[2, 2] = 1;

break;

}

}

multiply(GeomObject, Rotate3D);

}

private void multiply(float[,] obj, float[,] matrix)

{

float res\_1, res\_2, res\_3;

for (int ax = 0; ax < count\_elements; ax++)

{

res\_1 = (obj[ax, 0] \* matrix[0, 0] + obj[ax, 1] \* matrix[0, 1] + obj[ax, 2] \* matrix[0, 2]);

res\_2 = (obj[ax, 0] \* matrix[1, 0] + obj[ax, 1] \* matrix[1, 1] + obj[ax, 2] \* matrix[1, 2]);

res\_3 = (obj[ax, 0] \* matrix[2, 0] + obj[ax, 1] \* matrix[2, 1] + obj[ax, 2] \* matrix[2, 2]);

obj[ax, 0] = res\_1;

obj[ax, 1] = res\_2;

obj[ax, 2] = res\_3;

}

}

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

{

Draw();

}

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

{

if (e.KeyCode == Keys.Z)

{

CreateZoom(Sx, comboBox1.SelectedIndex);

}

if (e.KeyCode == Keys.X)

{

CreateZoom(Sy, comboBox1.SelectedIndex);

}

if (e.KeyCode == Keys.W)

{

CreateTranslate(Tx, comboBox1.SelectedIndex);

}

if (e.KeyCode == Keys.S)

{

CreateTranslate(Ty, comboBox1.SelectedIndex);

}

if (e.KeyCode == Keys.A)

{

CreateRotate(angle\_of\_rotation, comboBox1.SelectedIndex);

}

if (e.KeyCode == Keys.D)

{

CreateRotate(angle\_of\_rotation, comboBox1.SelectedIndex);

}

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

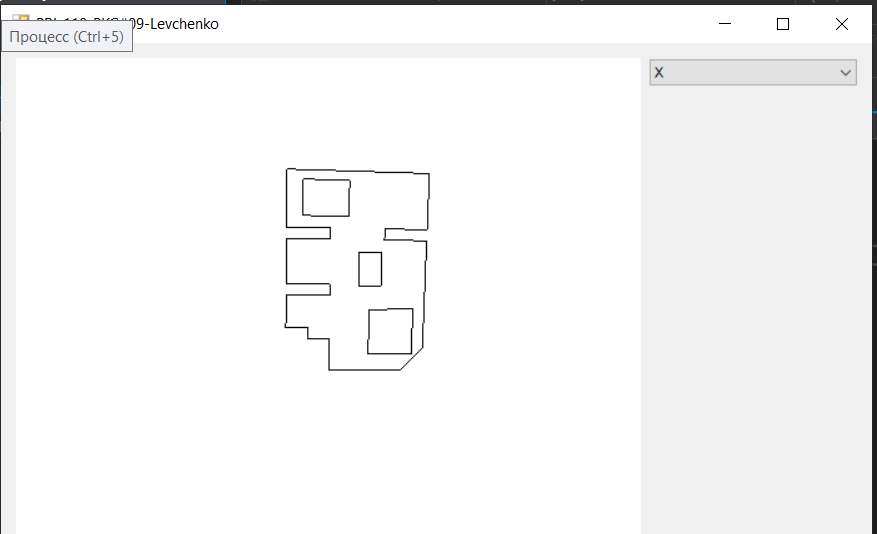
{

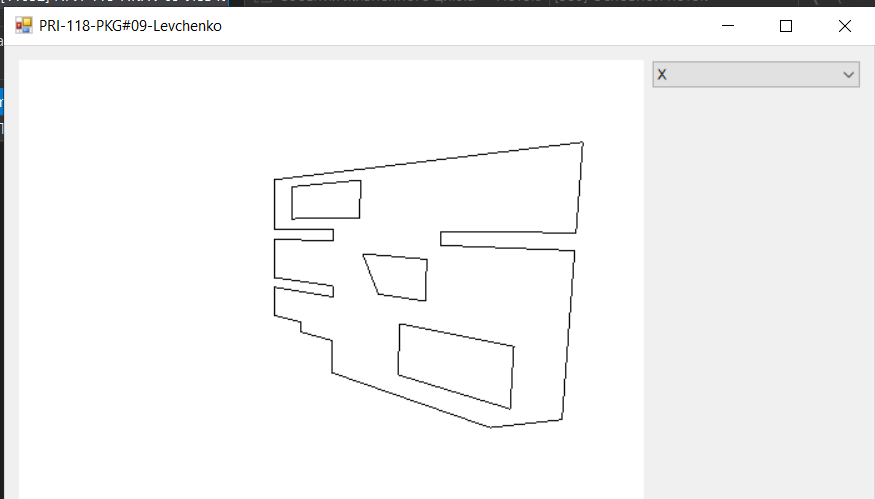
AnT.Focus();

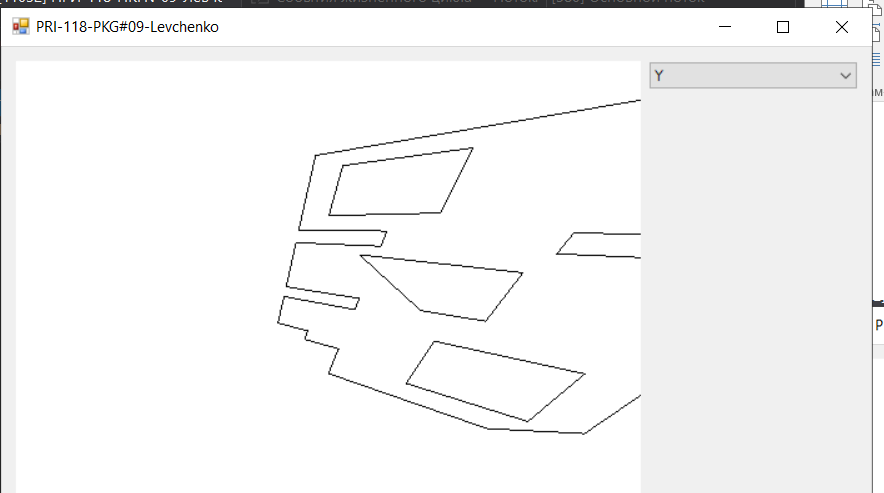
}

}

}

  
Рисунок 2. Результат запуска

  
Рисунок 3. Поворот по x

  
Рисунок 4. Поворот по y

**Вывод**

Изучены методы геометрических преобразований графических объектов, приобретены навыки использования средств геометрических преобразований при составлении графических программ.