Министерство образования и науки Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение науки и 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»**

**Тема: «Геометрические преобразования в 2D»**

Выполнил:

ст. гр. ПРИ-117

Хлызова В.Г.

Принял:

Жигалов И.Е.

Владимир, 2020

ЦЕЛЬ РАБОТЫ

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

ЗАДАНИЕ

1. Ознакомиться по методическим указаниям и литературе с теоретическим материалом.

2. Выполнить действия, приведенные в разделе 9.2. При разработке программы имя проекта, создаваемого в MS Visual Studio, должно содержать фамилию студента и группу (например, Ivanov\_Ivan\_ISG\_105\_lab\_1).

3. При разработке приложения из раздела 9.2 двухмерный объект, начальный угол поворота, константы переноса и масштабирования использовать в соответствии с вашим вариантом.

4. Изменить программу из раздела 9.2, введя в нее возможность двухмерных геометрических преобразований не только в плоскости X0Y, но и в плоскостях X0Z и Y0Z для точек, описываемых пространственными координатами P=(x,y,z). У пользователя должна быть возможность выбора третьей оси для переноса и масштабирования - Z. Плоский поворот точек объекта при этом должен выполняться вокруг каждой их трех осей по выбору - X, Y, Z

ХОД РАБОТЫ

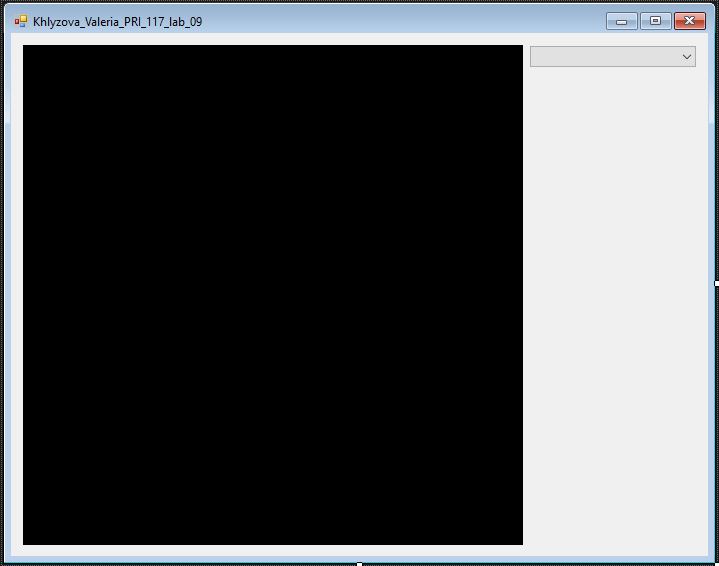


Рисунок 1. Конструктор формы.

Обработчик событий формы

using System;

using System.Windows.Forms;

using Tao.FreeGlut;

using Tao.OpenGl;

namespace Khlyzova\_Valeria\_PRI\_117\_lab\_09

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

AnT.InitializeContexts();

}

private float[,] GeomObject = new float[50, 3];

private int count\_elements = 0;

//начальный угол поворота, константы переноса и масштабирования

private float angle\_of\_rotation = (float)3.14 / 4;

private float Tx = 2.50f;

private float Ty = -0.50f;

private float Sx = 1.5f;

private float Sy = 0.5f;

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);

// координаты фигуры из варианта (15 вариант)

GeomObject[0, 0] = 0.1f;

GeomObject[0, 1] = 0.0f;

GeomObject[0, 2] = 0.0f;

GeomObject[1, 0] = 0.3f;

GeomObject[1, 1] = 0.0f;

GeomObject[1, 2] = 0.0f;

GeomObject[2, 0] = 0.3f;

GeomObject[2, 1] = 0.2f;

GeomObject[2, 2] = 0.0f;

GeomObject[3, 0] = 0.1f;

GeomObject[3, 1] = 0.2f;

GeomObject[3, 2] = 0.0f;

GeomObject[4, 0] = -0.1f;

GeomObject[4, 1] = 0.4f;

GeomObject[4, 2] = 0.0f;

GeomObject[5, 0] = 0.0f;

GeomObject[5, 1] = 0.6f;

GeomObject[5, 2] = 0.0f;

GeomObject[6, 0] = 0.4f;

GeomObject[6, 1] = 0.6f;

GeomObject[6, 2] = 0.0f;

GeomObject[7, 0] = 0.5f;

GeomObject[7, 1] = 0.4f;

GeomObject[7, 2] = 0.0f;

GeomObject[8, 0] = 0.5f;

GeomObject[8, 1] = -0.2f;

GeomObject[8, 2] = 0.0f;

GeomObject[9, 0] = 0.5f;

GeomObject[9, 1] = 0.2f;

GeomObject[9, 2] = 0.0f;

GeomObject[10, 0] = 0.6f;

GeomObject[10, 1] = 0.2f;

GeomObject[10, 2] = 0.0f;

GeomObject[11, 0] = 0.6f;

GeomObject[11, 1] = -0.2f;

GeomObject[11, 2] = 0.0f;

GeomObject[12, 0] = 0.8f;

GeomObject[12, 1] = -0.2f;

GeomObject[12, 2] = 0.0f;

GeomObject[13, 0] = 0.8f;

GeomObject[13, 1] = 0.9f;

GeomObject[13, 2] = 0.0f;

GeomObject[14, 0] = 0.6f;

GeomObject[14, 1] = 0.9f;

GeomObject[14, 2] = 0.0f;

GeomObject[15, 0] = 0.6f;

GeomObject[15, 1] = 1.1f;

GeomObject[15, 2] = 0.0f;

GeomObject[16, 0] = 0.9f;

GeomObject[16, 1] = 1.1f;

GeomObject[16, 2] = 0.0f;

GeomObject[17, 0] = 0.9f;

GeomObject[17, 1] = 1.5f;

GeomObject[17, 2] = 0.0f;

GeomObject[18, 0] = 0.6f;

GeomObject[18, 1] = 1.8f;

GeomObject[18, 2] = 0.0f;

GeomObject[19, 0] = -0.2f;

GeomObject[19, 1] = 1.8f;

GeomObject[19, 2] = 0.0f;

GeomObject[20, 0] = -0.5f;

GeomObject[20, 1] = 1.5f;

GeomObject[20, 2] = 0.0f;

GeomObject[21, 0] = -0.5f;

GeomObject[21, 1] = 1.1f;

GeomObject[21, 2] = 0.0f;

GeomObject[22, 0] = -0.2f;

GeomObject[22, 1] = 1.1f;

GeomObject[22, 2] = 0.0f;

GeomObject[23, 0] = -0.2f;

GeomObject[23, 1] = 0.9f;

GeomObject[23, 2] = 0.0f;

GeomObject[24, 0] = -0.4f;

GeomObject[24, 1] = 0.9f;

GeomObject[24, 2] = 0.0f;

GeomObject[25, 0] = -0.4f;

GeomObject[25, 1] = -0.2f;

GeomObject[25, 2] = 0.0f;

GeomObject[26, 0] = -0.2f;

GeomObject[26, 1] = -0.2f;

GeomObject[26, 2] = 0.0f;

GeomObject[27, 0] = -0.2f;

GeomObject[27, 1] = 0.2f;

GeomObject[27, 2] = 0.0f;

GeomObject[28, 0] = -0.1f;

GeomObject[28, 1] = 0.2f;

GeomObject[28, 2] = 0.0f;

GeomObject[29, 0] = -0.1f;

GeomObject[29, 1] = -0.2f;

GeomObject[29, 2] = 0.0f;

GeomObject[30, 0] = 0.5f;

GeomObject[30, 1] = 1.7f;

GeomObject[30, 2] = 0.0f;

GeomObject[31, 0] = 0.7f;

GeomObject[31, 1] = 1.5f;

GeomObject[31, 2] = 0.0f;

GeomObject[32, 0] = 0.5f;

GeomObject[32, 1] = 1.3f;

GeomObject[32, 2] = 0.0f;

GeomObject[33, 0] = 0.3f;

GeomObject[33, 1] = 1.5f;

GeomObject[33, 2] = 0.0f;

GeomObject[34, 0] = -0.1f;

GeomObject[34, 1] = 1.7f;

GeomObject[34, 2] = 0.0f;

GeomObject[35, 0] = 0.1f;

GeomObject[35, 1] = 1.5f;

GeomObject[35, 2] = 0.0f;

GeomObject[36, 0] = -0.1f;

GeomObject[36, 1] = 1.3f;

GeomObject[36, 2] = 0.0f;

GeomObject[37, 0] = -0.3f;

GeomObject[37, 1] = 1.5f;

GeomObject[37, 2] = 0.0f;

count\_elements = 38;

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();

// отрисовываем фигуру

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

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

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

Gl.glEnd();

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

for (int i = 0; i < 4; 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 = 4; i < 8; i++)

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

Gl.glEnd();

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

for (int i = 4; i < 8; 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 = 8; i < 30; i++)

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

Gl.glEnd();

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

for (int i = 8; i < 30; 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 = 30; i < 34; i++)

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

Gl.glEnd();

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

for (int i = 30; i < 34; 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 = 34; i < 38; i++)

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

Gl.glEnd();

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

for (int i = 34; i < 38; 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: // вокруг оси Х

{

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: // вокруг оси Y

{

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: // вокруг оси Z

{

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;

// добавляем точки для оси Z

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)

{

// используем константы геометрических преобразований из варианта

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();

}

}

}

Пример работы программы

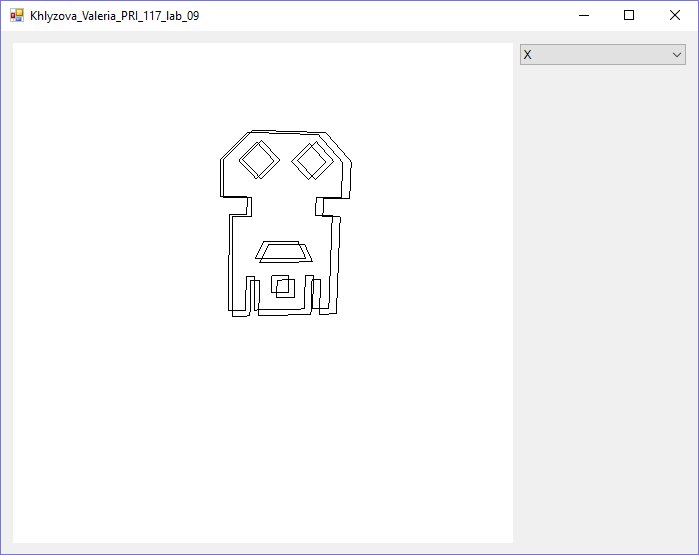


Рисунок 2. Исходный вид фигуры.

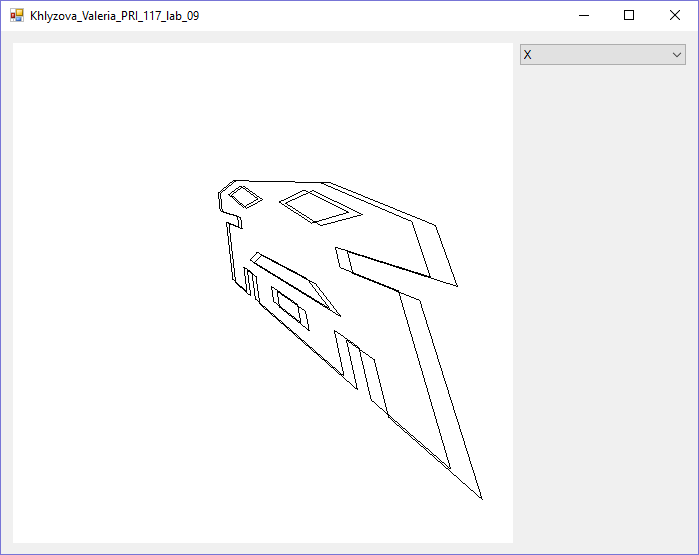


Рисунок 3. Применение нескольких трансформаций по оси X.

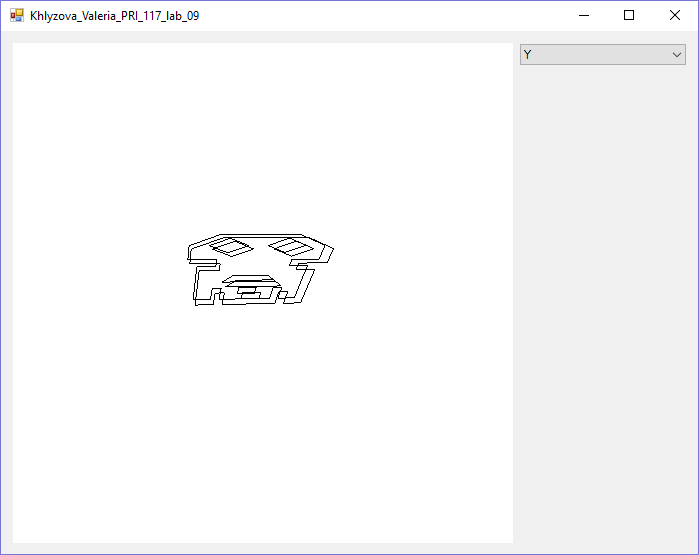


Рисунок 4. Применение нескольких трансформаций по оси Y.

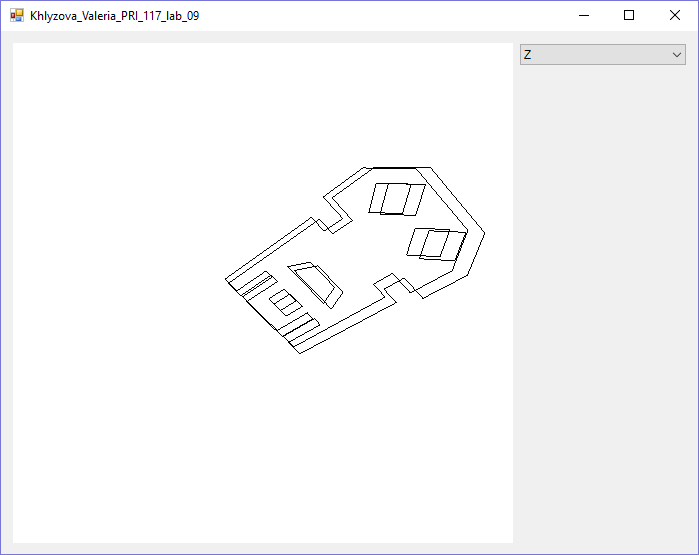


Рисунок 5. Применение нескольких трансформаций по оси Z.

ВЫВОД

В ходе выполнения лабораторной работы были изучены методы геометрических преобразований графических объектов, приобретены навыки использования средств геометрических преобразований при составлении графических программ.