#### Министерство науки и высшего образования 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"

Выполнил:

ст. гр. ПРИ-117

Емельянов Д.В.

Принял:

Жигалов И.Е.

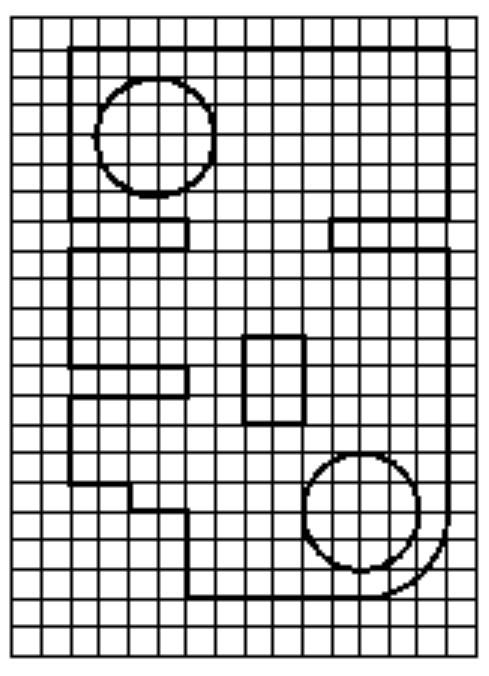
Владимир, 2020 г.

**Цель работы**

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

**Выполнение работы**

**Вариант №5**

****

Измененный метод AnT\_KeyDown под свои константы:

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

{

// Z и X отвечают за масштабирование

if (e.KeyCode == Keys.Z)

{

float coef;

if (comboBox1.SelectedIndex == 0) // X

{

coef = 0.5f;

}

else // Y,Z

{

coef = 1/2.5f;

}

// вызов функции, в которой мы реализуем масштабирование -

// передаем коэфициент масштабирования и выбранную ось в окне программы

CreateZoom(1 + coef, comboBox1.SelectedIndex);

}

if (e.KeyCode == Keys.X)

{

float coef;

if (comboBox1.SelectedIndex == 0) // X

{

coef = 0.5f;

}

else // Y,Z

{

coef = 1/2.5f;

}

// вызов функции, в которой мы реализуем масштабирование -

// передаем коэфициент масштабирования и выбранную ось в окне программы

CreateZoom(1 - coef, comboBox1.SelectedIndex);

}

// W и S отвечают за перенос

if (e.KeyCode == Keys.W)

{

float coef;

if (comboBox1.SelectedIndex == 0) // X

{

coef = 150f;

}

else if (comboBox1.SelectedIndex == 1) // Y

{

coef = 0f;

}

else //Z

{

coef = 0f;

}

// вызов функции, в которой мы реализуем перенос -

// передаем коэфициент переноса и выбранную ось в окне программы

CreateTranslate(coef, comboBox1.SelectedIndex);

}

if (e.KeyCode == Keys.S)

{

float coef;

if (comboBox1.SelectedIndex == 0) // X

{

coef = 150f;

}

else // Y,Z

{

coef = 0f;

}

// вызов функции, в которой мы реализуем перенос -

// передаем коэфициент переноса и выбранную ось в окне программы

CreateTranslate(-coef, comboBox1.SelectedIndex);

}

// A и D отвечают за поворот

if (e.KeyCode == Keys.A)

{

// вызов функции, в которой мы реализуем поворот -

// передаем угол поворота относительно оси Z

CreateRotate((float)Math.PI, comboBox1.SelectedIndex);

}

if (e.KeyCode == Keys.D)

{

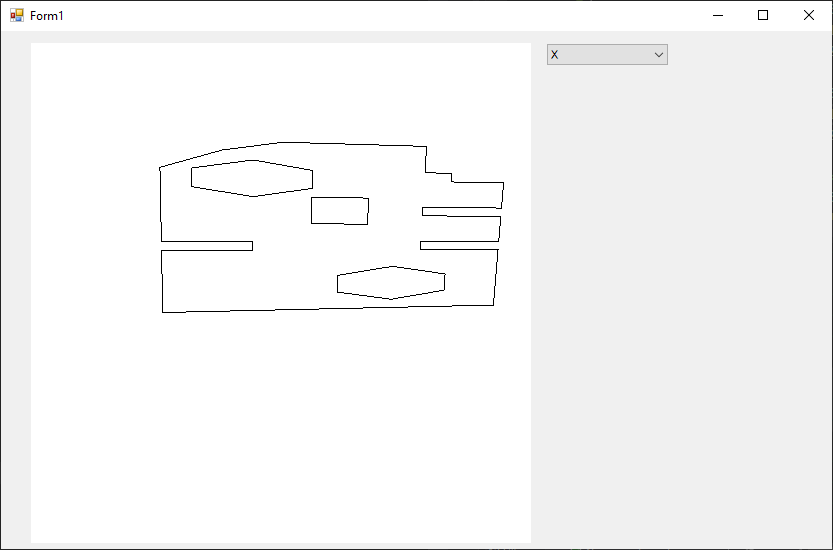
// вызов функции, в которой мы реализуем поворот -

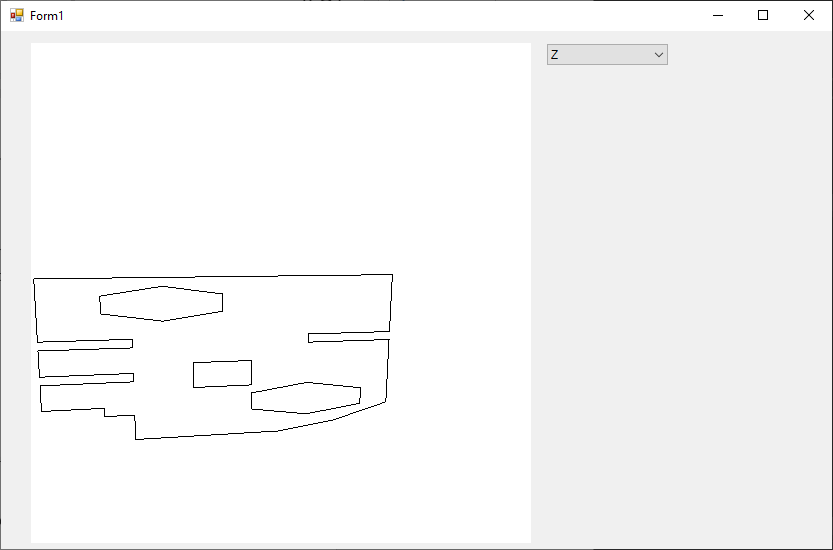
// передаем угол поворота относительно оси Z

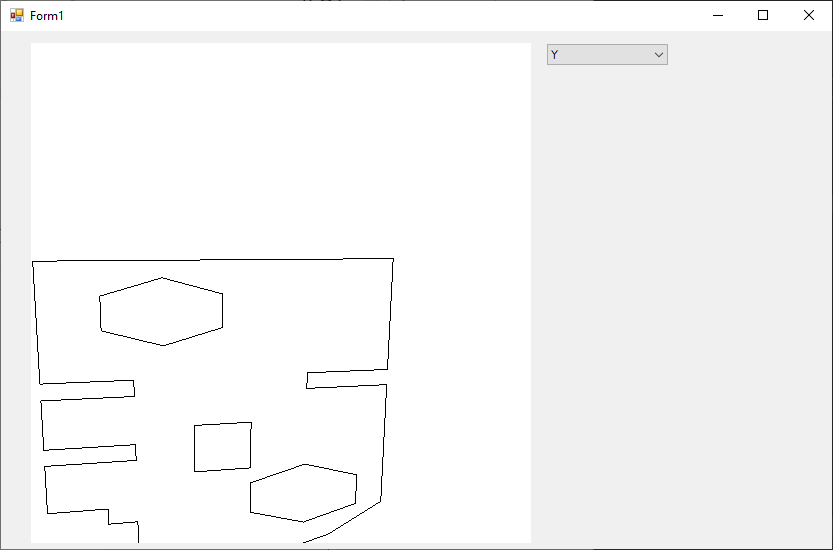
CreateRotate(-(float)Math.PI, comboBox1.SelectedIndex);

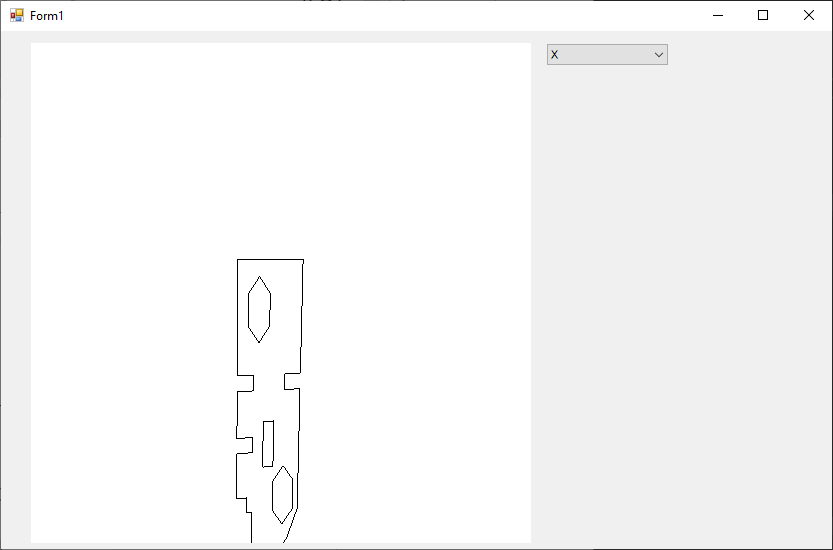
}

}









Измененный код программы:

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 < 25; i++)

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

Gl.glEnd();

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

for (int i = 25; i < 29; i++)

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

Gl.glEnd();

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

for (int i = 29; 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 = 38; i < 47; i++)

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

Gl.glEnd();

Gl.glPopMatrix();

Gl.glPopMatrix();

Gl.glFlush();

AnT.Invalidate();

}

// реализация поворота

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);

}

// функция умножения вектора координат точек объекта на матрицу преобразования

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 Form1\_Load(object sender, EventArgs e)

{

// инициализация OpenGL, много раз комментированная ранее

// инициализация бибилиотеки glut

Glut.glutInit();

// инициализация режима экрана

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

// установка цвета очистки экрана (RGBA)

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

// установка порта вывода

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

// активация проекционной матрицы

Gl.glMatrixMode(Gl.GL\_PROJECTION);

// очистка матрицы

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);

//Правый угол фигуры

GeomObject[0, 0] = 0.8f;

GeomObject[0, 1] = -0.2f;

GeomObject[0, 2] = 0.0f;

GeomObject[1, 0] = 0.8f;

GeomObject[1, 1] = 0.5f;

GeomObject[1, 2] = 0.0f;

//Прорез(лесенка справа)

GeomObject[2, 0] = 0.5f;

GeomObject[2, 1] = 0.5f;

GeomObject[2, 2] = 0.0f;

GeomObject[3, 0] = 0.5f;

GeomObject[3, 1] = 0.6f;

GeomObject[3, 2] = 0.0f;

GeomObject[4, 0] = 0.8f;

GeomObject[4, 1] = 0.6f;

GeomObject[4, 2] = 0.0f;

GeomObject[5, 0] = 0.8f;

GeomObject[5, 1] = 0.9f;

GeomObject[5, 2] = 0.0f;

GeomObject[6, 0] = 0.5f;

GeomObject[6, 1] = 0.9f;

GeomObject[6, 2] = 0.0f;

GeomObject[7, 0] = 0.5f;

GeomObject[7, 1] = 1.0f;

GeomObject[7, 2] = 0.0f;

GeomObject[8, 0] = 0.8f;

GeomObject[8, 1] = 1.0f;

GeomObject[8, 2] = 0.0f;

GeomObject[9, 0] = 0.8f;

GeomObject[9, 1] = 1.3f;

GeomObject[9, 2] = 0.0f;

GeomObject[10, 0] = 0.6f;

GeomObject[10, 1] = 1.3f;

GeomObject[10, 2] = 0.0f;

GeomObject[11, 0] = 0.6f;

GeomObject[11, 1] = 1.4f;

GeomObject[11, 2] = 0.0f;

GeomObject[12, 0] = 0.5f;

GeomObject[12, 1] = 1.4f;

GeomObject[12, 2] = 0.0f;

GeomObject[13, 0] = 0.5f;

GeomObject[13, 1] = 1.7f;

GeomObject[13, 2] = 0.0f;

//Скругление верхнего угла

GeomObject[14, 0] = 0.0f;

GeomObject[14, 1] = 1.7f;

GeomObject[14, 2] = 0.0f;

GeomObject[15, 0] = -0.2f;

GeomObject[15, 1] = 1.6f;

GeomObject[15, 2] = 0.0f;

GeomObject[16, 0] = -0.3f;

GeomObject[16, 1] = 1.5f;

GeomObject[16, 2] = 0.0f;

GeomObject[17, 0] = -0.4f;

GeomObject[17, 1] = 1.4f;

GeomObject[17, 2] = 0.0f;

GeomObject[18, 0] = -0.4f;

GeomObject[18, 1] = 0.6f;

GeomObject[18, 2] = 0.0f;

GeomObject[19, 0] = -0.4f;

GeomObject[19, 1] = 0.6f;

GeomObject[19, 2] = 0.0f;

GeomObject[20, 0] = -0.1f;

GeomObject[20, 1] = 0.6f;

GeomObject[20, 2] = 0.0f;

GeomObject[21, 0] = -0.1f;

GeomObject[21, 1] = 0.5f;

GeomObject[21, 2] = 0.0f;

GeomObject[22, 0] = -0.4f;

GeomObject[22, 1] = 0.5f;

GeomObject[22, 2] = 0.0f;

GeomObject[23, 0] = -0.4f;

GeomObject[23, 1] = -0.2f;

GeomObject[23, 2] = 0.0f;

GeomObject[24, 0] = 0.8f;

GeomObject[24, 1] = -0.2f;

GeomObject[24, 2] = 0.0f;

//Прямоугольник внутри фигуры

GeomObject[25, 0] = 0.1f;

GeomObject[25, 1] = 1.1f;

GeomObject[25, 2] = 0.0f;

GeomObject[26, 0] = 0.3f;

GeomObject[26, 1] = 1.1f;

GeomObject[26, 2] = 0.0f;

GeomObject[27, 0] = 0.3f;

GeomObject[27, 1] = 0.8f;

GeomObject[27, 2] = 0.0f;

GeomObject[28, 0] = 0.1f;

GeomObject[28, 1] = 0.8f;

GeomObject[28, 2] = 0.0f;

//Окружность левая

GeomObject[29, 0] = -0.1f;

GeomObject[29, 1] = 1.5f;

GeomObject[29, 2] = 0.0f;

GeomObject[30, 0] = 0.1f;

GeomObject[30, 1] = 1.4f;

GeomObject[30, 2] = 0.0f;

GeomObject[31, 0] = 0.1f;

GeomObject[31, 1] = 1.3f;

GeomObject[31, 2] = 0.0f;

GeomObject[32, 0] = 0.1f;

GeomObject[32, 1] = 1.2f;

GeomObject[32, 2] = 0.0f;

GeomObject[33, 0] = -0.1f;

GeomObject[33, 1] = 1.1f;

GeomObject[33, 2] = 0.0f;

GeomObject[34, 0] = -0.3f;

GeomObject[34, 1] = 1.2f;

GeomObject[34, 2] = 0.0f;

GeomObject[35, 0] = -0.3f;

GeomObject[35, 1] = 1.3f;

GeomObject[35, 2] = 0.0f;

GeomObject[36, 0] = -0.3f;

GeomObject[36, 1] = 1.4f;

GeomObject[36, 2] = 0.0f;

GeomObject[37, 0] = -0.1f;

GeomObject[37, 1] = 1.5f;

GeomObject[37, 2] = 0.0f;

//Окружность правая

GeomObject[38, 0] = 0.4f;

GeomObject[38, 1] = 0.3f;

GeomObject[38, 2] = 0.0f;

GeomObject[39, 0] = 0.6f;

GeomObject[39, 1] = 0.2f;

GeomObject[39, 2] = 0.0f;

GeomObject[40, 0] = 0.6f;

GeomObject[40, 1] = 0.1f;

GeomObject[40, 2] = 0.0f;

GeomObject[41, 0] = 0.6f;

GeomObject[41, 1] = 0.0f;

GeomObject[41, 2] = 0.0f;

GeomObject[42, 0] = 0.4f;

GeomObject[42, 1] = -0.1f;

GeomObject[42, 2] = 0.0f;

GeomObject[43, 0] = 0.2f;

GeomObject[43, 1] = 0.0f;

GeomObject[43, 2] = 0.0f;

GeomObject[44, 0] = 0.2f;

GeomObject[44, 1] = 0.1f;

GeomObject[44, 2] = 0.0f;

GeomObject[45, 0] = 0.2f;

GeomObject[45, 1] = 0.2f;

GeomObject[45, 2] = 0.0f;

GeomObject[46, 0] = 0.4f;

GeomObject[46, 1] = 0.3f;

GeomObject[46, 2] = 0.0f;

count\_elements = GeomObject.GetLength(0); //количество строк

comboBox1.SelectedIndex = 0;

RenderTimer.Start();

}

**Выводы**

 Изучил методы геометрических преобразований графических объектов, приобрел навыки использования средств геометрических преобразований при составлении графических программ.