**Министерство образования и науки Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа информационных технологий и робототехники

Отделение Информационных технологий

Профиль Геоинформационные системы

**ОТЧЕТ  
по лабораторной работе №4**

Базовые алгоритмы 3D-геометрии

по дисциплине Компьютерная графика

Выполнил студент группы 8И5Б \_\_\_\_\_\_\_\_\_\_ Смирнов П.О.

\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ 2018 г.

Отчет принят:

Принял: \_\_\_\_\_\_\_\_\_\_\_ Шумихин В.С.

\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ 2018 г.

Томск 2018 г.

# Цель

Изучить способы выполнения базовых преобразований и построения проекций.

# Задание №13

## Даны следующие координаты:

### (0;0;0) (0;11;0);

### (8;0;0) (0;11;0);

### (8;0;5) (0;11;0);

### (0;0;5) (0;11;0);

### Написать программу, которая по заданным координатам строит ортографическую проекцию.

### Реализовать в данной программе вращение по разным осям, масштабирование и перенос.

### Все действия производить по нажатию горячих клавиш.

### Предусмотреть возможность возврата к первоначальной проекции.

**Ход работы**

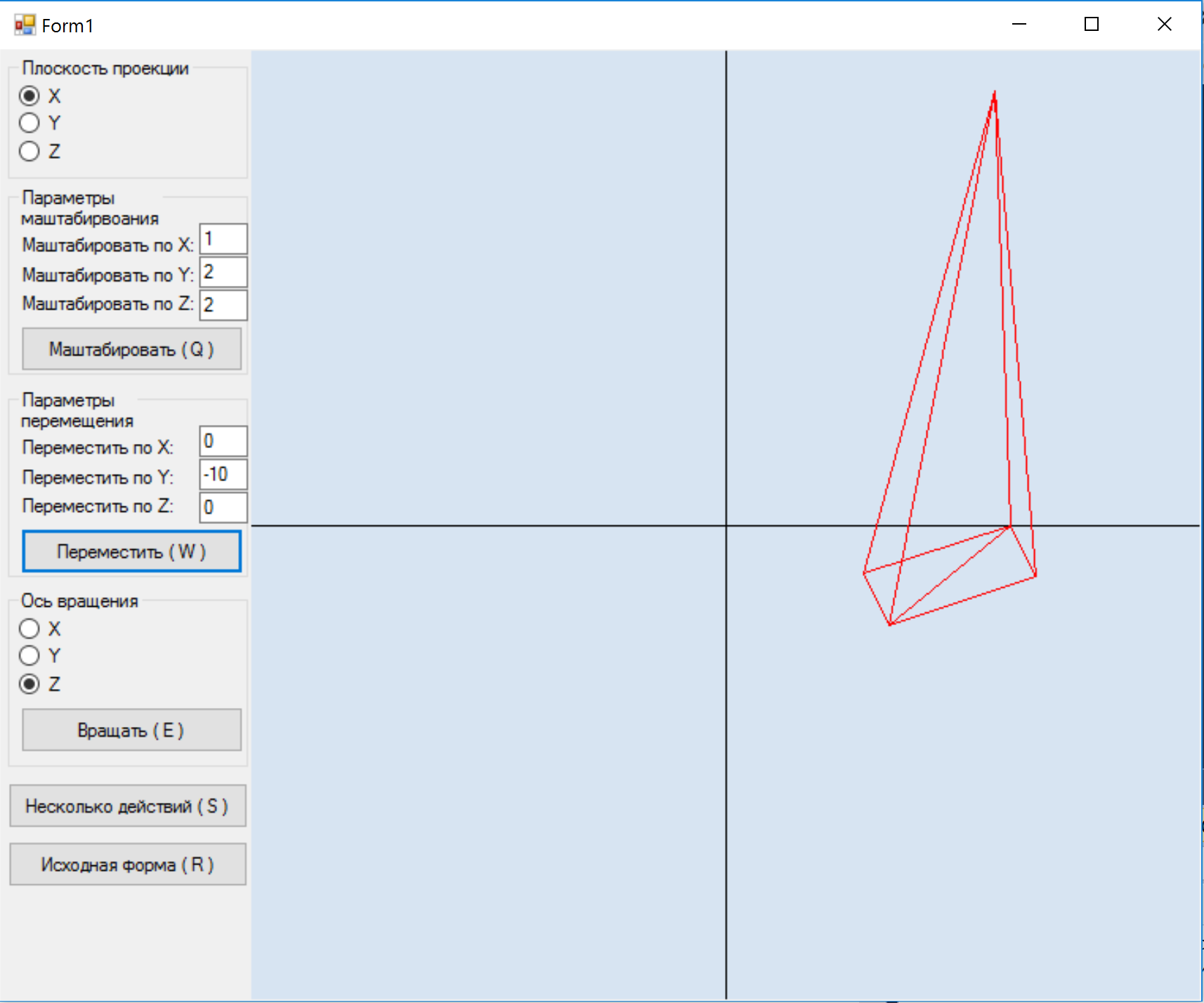


Рисунок 1 – Область работы

**Вывод**

В ходе проделанной лабораторной работы были изучены способы выполнения преобразований трехмерных фигур. Для реализации поворота, перемещения и масштабирования фигур были использованы соответствующие матрицы преобразований. Для всех операций использовались однородные координаты точек. При таком подходе необходимо нормализовать координаты после каждого преобразования.

Отображение трехмерных фигур на плоскости невозможно без проецирования. В данной лабораторной работе использовалась ортографическая проекция многомерной фигуры на плоскость XOY.

Для удобства работы пользователя с программой было реализовано управление с использованием «горячих клавиш».

**Код программы**

namespace TryGr4

{

public partial class Form1 : Form

{

Workspace myWorkspace = new Workspace();

Matrix array4x4 = new Matrix();

public Graphics gr;

public Form1()

{

InitializeComponent();

this.KeyPreview = true;

radioButtonViewX.Checked = true;

myWorkspace.InitForm(this);

}

public Point MapToScreen(double x, double y)

{

x = pictureBox1.Width \* ((x + 50.0) / 100.0);

y = pictureBox1.Height \* (1 - (y + 46.0) / 92.0);

Point SreenCoord = new Point((int)x, (int)y);

return SreenCoord;

}

public List<TextBox> GetTextBoxes()

{

List<TextBox> listOfTextBoxes = new List<TextBox>();

listOfTextBoxes.Add(textBoxScaleX);

listOfTextBoxes.Add(textBoxScaleY);

listOfTextBoxes.Add(textBoxScaleZ);

listOfTextBoxes.Add(textBoxMoveX);

listOfTextBoxes.Add(textBoxMoveY);

listOfTextBoxes.Add(textBoxMoveZ);

return listOfTextBoxes;

}

public List<RadioButton> GetRadioButtons()

{

List<RadioButton> listOfCheckBoxes = new List<RadioButton>();

listOfCheckBoxes.Add(radioButtonViewX);

listOfCheckBoxes.Add(radioButtonViewY);

listOfCheckBoxes.Add(radioButtonViewZ);

listOfCheckBoxes.Add(radioButtonRotateX);

listOfCheckBoxes.Add(radioButtonRotateY);

listOfCheckBoxes.Add(radioButtonRotateZ);

return listOfCheckBoxes;

}

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

{

myWorkspace.scale = true;

pictureBox1.Invalidate();

}

private void pictureBox1\_Paint(object sender, PaintEventArgs e)

{

gr = e.Graphics;

Pen OurPen = new Pen(Color.Black, 1);

Point pointStart1 = MapToScreen(0,-300);

Point pointStart2 = MapToScreen(0, 300);

Point pointEnd1 = MapToScreen(-300, 0);

Point pointEnd2 = MapToScreen(300, 0);

e.Graphics.DrawLine(OurPen, pointStart1.X, pointStart1.Y, pointStart2.X,pointStart2.Y);

e.Graphics.DrawLine(OurPen, pointEnd1.X, pointEnd1.Y, pointEnd2.X, pointEnd2.Y);

InitializeMatrix();

if (myWorkspace.begin)

{

myWorkspace.InitMatrix(array4x4);

}

myWorkspace.Draw();

}

public void InitializeMatrix()

{

array4x4.AddNode(new GeoPoint(0, 0, 0, 1));

array4x4.AddNode(new GeoPoint(0, 11, 0, 1));

array4x4.AddNode(new GeoPoint(0, 0, 5, 1));

array4x4.AddNode(new GeoPoint(0, 0, 0, 1));

array4x4.AddNode(new GeoPoint(8, 0, 0, 1));

array4x4.AddNode(new GeoPoint(0, 11, 0, 1));

array4x4.AddNode(new GeoPoint(8, 0, 5, 1));

array4x4.AddNode(new GeoPoint(0, 0, 5, 1));

array4x4.AddNode(new GeoPoint(0, 0, 0, 1));

array4x4.AddNode(new GeoPoint(8, 0, 0, 1));

array4x4.AddNode(new GeoPoint(8, 0, 5, 1));

array4x4.AddNode(new GeoPoint(0, 0, 5, 1));

}

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

{

myWorkspace.ToComeBack();

myWorkspace.begin = true;

pictureBox1.Invalidate();

}

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

{

myWorkspace.move = true;

pictureBox1.Invalidate();

}

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

{

myWorkspace.rotate = true;

pictureBox1.Invalidate();

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyData == Keys.Q)

{

myWorkspace.scale = true;

pictureBox1.Invalidate();

}

if (e.KeyData == Keys.W)

{

myWorkspace.move = true;

pictureBox1.Invalidate();

}

if (e.KeyData == Keys.E)

{

myWorkspace.rotate = true;

pictureBox1.Invalidate();

}

if (e.KeyData == Keys.R)

{

myWorkspace.ToComeBack();

myWorkspace.begin = true;

pictureBox1.Invalidate();

}

if (e.KeyData == Keys.S)

{

myWorkspace.scale = true;

myWorkspace.move = true;

myWorkspace.rotate = true;

pictureBox1.Invalidate();

}

}

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

{

myWorkspace.scale = true;

myWorkspace.move = true;

myWorkspace.rotate = true;

pictureBox1.Invalidate();

}

}

public class Workspace

{

public bool begin = true;

public bool move = false, rotate = false, scale = false;

Matrix Nodes = new Matrix();

Matrix StartMatrix;

public Matrix SourceNodes = new Matrix();

Form1 form;

public Matrix SaveStartMatrix()

{

StartMatrix = Nodes.MakeCopy();

return StartMatrix;

}

public void CheckRadioButtons()

{

List<RadioButton> localListOfCheckBoxes = form.GetRadioButtons();

if (localListOfCheckBoxes[0].Checked)

{ m = 0; k = 1; l = 2; }

if (localListOfCheckBoxes[1].Checked)

{ m = 1; k = 0; l = 2; }

if (localListOfCheckBoxes[2].Checked)

{ m = 2; k = 0; l = 1; }

}

public void InitForm(Form1 inputForm)

{

form = inputForm;

}

public void InitMatrix(Matrix inputMatrix)

{

Nodes = inputMatrix;

}

public void Draw()

{

if (!begin)

{

if (scale)

{

Nodes = ToScale().MakeCopy();

scale = false;

}

if (move)

{

Nodes = Move().MakeCopy();

move = false;

}

if(rotate)

{

Nodes = Rotate().MakeCopy();

rotate = false;

}

}

else

{

Nodes = SaveStartMatrix();

begin = false;

}

CheckRadioButtons();

for (int i = 0; i < Nodes.matrix.Count; i++)

{

if (i < Nodes.matrix.Count / 2)

{

Point start1 = form.MapToScreen(Nodes.matrix[i].coord[k], Nodes.matrix[i].coord[l]);

Point start2 = form.MapToScreen(Nodes.matrix[i + Nodes.matrix.Count / 2].coord[k], Nodes.matrix[i + Nodes.matrix.Count / 2].coord[l]);

form.gr.DrawLine(new Pen(Color.Red), form.MapToScreen(Nodes.matrix[i].coord[k], Nodes.matrix[i].coord[l]), form.MapToScreen(Nodes.matrix[i + Nodes.matrix.Count / 2].coord[k], Nodes.matrix[i + Nodes.matrix.Count / 2].coord[l]));

if (i != Nodes.matrix.Count / 2 - 1)

form.gr.DrawLine(new Pen(Color.Red),form.MapToScreen(Nodes.matrix[i].coord[k], Nodes.matrix[i].coord[l]),form.MapToScreen(Nodes.matrix[i + 1].coord[k], Nodes.matrix[i + 1].coord[l]));

else

form.gr.DrawLine(new Pen(Color.Red),form.MapToScreen(Nodes.matrix[i].coord[k], Nodes.matrix[i].coord[l]),form.MapToScreen(Nodes.matrix[0].coord[k], Nodes.matrix[0].coord[l]));

}

else

{

if (i != Nodes.matrix.Count - 1)

form.gr.DrawLine(new Pen(Color.Red),form.MapToScreen(Nodes.matrix[i].coord[k], Nodes.matrix[i].coord[l]),form.MapToScreen(Nodes.matrix[i + 1].coord[k], Nodes.matrix[i + 1].coord[l]));

else

form.gr.DrawLine(new Pen(Color.Red),form.MapToScreen(Nodes.matrix[i].coord[k], Nodes.matrix[i].coord[l]),form.MapToScreen(Nodes.matrix[Nodes.matrix.Count / 2].coord[k], Nodes.matrix[Nodes.matrix.Count / 2].coord[l]));

}

}

}

public Matrix Rotate(double inputAngle = 45)

{

List<RadioButton> localListOfCheckBoxes = form.GetRadioButtons();

inputAngle = inputAngle \* 3.14 / 180.0;

Matrix TurnMatrix = new Matrix();

if (localListOfCheckBoxes[3].Checked)

{

TurnMatrix.AddNode(new GeoPoint(1, 0, 0, 0));

TurnMatrix.AddNode(new GeoPoint(0, Math.Cos(inputAngle), -Math.Sin(inputAngle), 0));

TurnMatrix.AddNode(new GeoPoint(0, Math.Sin(inputAngle), Math.Cos(inputAngle), 0));

TurnMatrix.AddNode(new GeoPoint(0, 0, 0, 1));

}

if (localListOfCheckBoxes[4].Checked)

{

TurnMatrix.AddNode(new GeoPoint(Math.Cos(inputAngle), 0, -Math.Sin(inputAngle), 0));

TurnMatrix.AddNode(new GeoPoint(0, 1, 0, 0));

TurnMatrix.AddNode(new GeoPoint(Math.Sin(inputAngle), 0, Math.Cos(inputAngle), 0));

TurnMatrix.AddNode(new GeoPoint(0, 0, 0, 1));

}

if (localListOfCheckBoxes[5].Checked)

{

TurnMatrix.AddNode(new GeoPoint(Math.Cos(inputAngle), -Math.Sin(inputAngle), 0, 0));

TurnMatrix.AddNode(new GeoPoint(Math.Sin(inputAngle), Math.Cos(inputAngle), 0, 0));

TurnMatrix.AddNode(new GeoPoint(0, 0, 1, 0));

TurnMatrix.AddNode(new GeoPoint(0, 0, 0, 1));

}

Matrix ResultMatrix = Nodes \* TurnMatrix;

return ResultMatrix;

}

public Matrix Move()

{

Matrix MoveMatrix = new Matrix();

List<TextBox> checkTextBoxes = form.GetTextBoxes();

for (int i = 3; i < checkTextBoxes.Count; i++)

{

int res;

bool isInt = Int32.TryParse(checkTextBoxes[i].Text, out res);

if (!isInt)

{

MessageBox.Show("Введите в поля число");

return Nodes;

}

}

int dx\_mov = Convert.ToInt32(checkTextBoxes[3].Text);

int dy\_mov = Convert.ToInt32(checkTextBoxes[4].Text);

int dz\_mov = Convert.ToInt32(checkTextBoxes[5].Text);

MoveMatrix.AddNode(new GeoPoint(1, 0, 0, 0));

MoveMatrix.AddNode(new GeoPoint(0, 1, 0, 0));

MoveMatrix.AddNode(new GeoPoint(0, 0, 1, 0));

MoveMatrix.AddNode(new GeoPoint(dx\_mov, dy\_mov, dz\_mov, 1));

Matrix ResultMatrix = Nodes \* MoveMatrix;

return ResultMatrix;

}

public void ToComeBack()

{

Nodes = SourceNodes;

}

private Matrix ToScale()

{

Matrix ScaleMatrix = new Matrix();

List<TextBox> checkTextBoxes = form.GetTextBoxes();

for (int i = 0; i < checkTextBoxes.Count - 3; i++)

{

int res;

bool isInt = Int32.TryParse(checkTextBoxes[i].Text, out res);

if(!isInt)

{

MessageBox.Show("Введите в поля число");

return Nodes;

}

}

ScaleMatrix.AddNode(new GeoPoint((double)Convert.ToInt32(checkTextBoxes[0].Text), 0, 0, 0));

ScaleMatrix.AddNode(new GeoPoint(0, (double)Convert.ToInt32(checkTextBoxes[1].Text), 0, 0));

ScaleMatrix.AddNode(new GeoPoint(0, 0, (double)Convert.ToInt32(checkTextBoxes[2].Text), 0));

ScaleMatrix.AddNode(new GeoPoint(0, 0, 0, 1));

Matrix ResultMatrix = Nodes \* ScaleMatrix;

return ResultMatrix;

}

}