Учреждение образования

«Гомельский Государственный технический университет имени П.О.Сухого»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

Отчет по лабораторной работе №1

Геометрические преобразования двумерных объектов (перенос, масштабирование, поворот)

По дисциплине «Алгоритмические основы современной компьютерной графики»

Выполнил студент гр.ИТЗ-21

Касаткин Д.О.

Проверил преподаватель

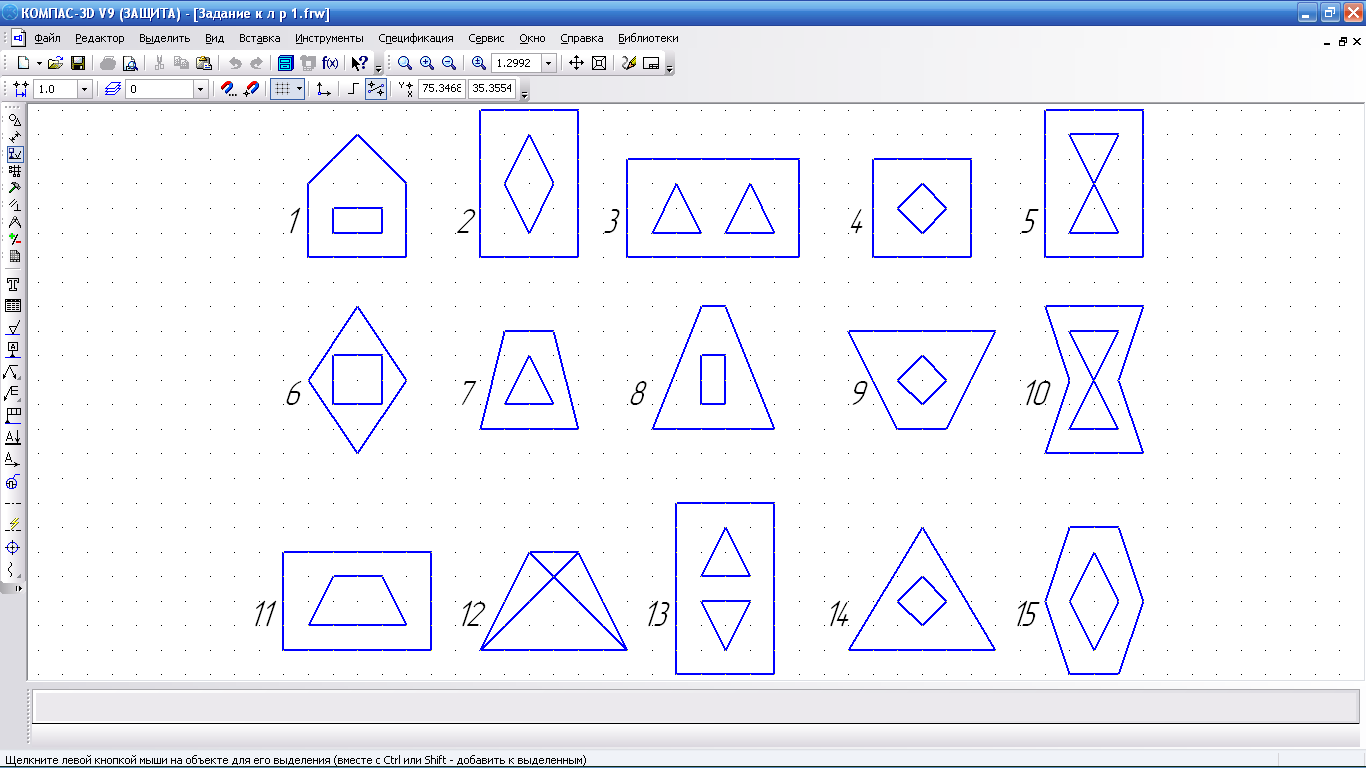
Комракова М.В.

Гомель 2019

Лабораторная работа №1

Геометрические преобразования двумерных объектов (перенос, масштабирование, поворот)

Задание на лабораторную работу.



Для выбранного варианта:

* В пакете Mathcad создал матрицу ключевых точек для каждой фигуры, образующей замкнутый контур.
* Провел перенос фигуры на расстояние вдоль горизонтали и вертикали.
* Провел неоднородное и однородное масштабирование заданной фигуры.
* Провести поворот заданной фигуры.
* По координатам ключевых точек построил изображение заданной фигуры, используя 2D график.

Листинг программы:

using SharpGL;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace lab1\_new

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

create\_osxy();

drow\_figures();

}

List<Osxy> osxy\_rictangle = new List<Osxy>();

List<Osxy> left\_triangle = new List<Osxy>();

List<Osxy> right\_triangle = new List<Osxy>();

private void create\_osxy()

{

osxy\_rictangle.Add(new Osxy(- 7f, 3.5f));

osxy\_rictangle.Add(new Osxy(7f, 3.5f));

osxy\_rictangle.Add(new Osxy(7f, -3f));

osxy\_rictangle.Add(new Osxy(-7f, -3f));

left\_triangle.Add(new Osxy(3.5f, 3f));

left\_triangle.Add(new Osxy(6f, -2f));

left\_triangle.Add(new Osxy(1f, -2f));

right\_triangle.Add(new Osxy(-3.5f, 3f));

right\_triangle.Add(new Osxy(-6f, -2f));

right\_triangle.Add(new Osxy(-1f, -2f));

}

private void Button1\_Click(object sender, EventArgs e)

{

foreach (Osxy osxy in osxy\_rictangle)

{

osxy.x -= 0.1f;

}

foreach (Osxy osxy in left\_triangle)

{

osxy.x -= 0.1f;

}

foreach (Osxy osxy in right\_triangle)

{

osxy.x -= 0.1f;

}

drow\_figures();

}

private void Button2\_Click(object sender, EventArgs e)

{

foreach (Osxy osxy in osxy\_rictangle)

{

osxy.x += 0.1f;

}

foreach (Osxy osxy in left\_triangle)

{

osxy.x += 0.1f;

}

foreach (Osxy osxy in right\_triangle)

{

osxy.x += 0.1f;

}

drow\_figures();

}

private void multyplie\_matrix(float[][] A,int countrows)

{

foreach (Osxy osxy in osxy\_rictangle)

{

}

}

Single angle = 0;

private void drow\_figures()

{

OpenGL gl = this.openGLControl1.OpenGL;

// Очистка экрана и буфера глубин

gl.Clear(OpenGL.GL\_COLOR\_BUFFER\_BIT | OpenGL.GL\_DEPTH\_BUFFER\_BIT);

// Сбрасываем модельно-видовую матрицу

gl.LoadIdentity();

gl.Rotate(angle, 0f, 0f, 1f);

gl.Translate(0.0f, 0.0f, -20.0f);

gl.Begin(OpenGL.GL\_LINE\_LOOP);

gl.Color(1f, 0f, 0f);

//рисуем левый прямоугольник

foreach (Osxy figures in osxy\_rictangle)

{

gl.Vertex(figures.x, figures.y);

}

gl.End();

gl.Begin(OpenGL.GL\_LINE\_LOOP);

// Указываем цвет вершин

gl.Color(1f, 0f, 0f);

//рисуем левый треугольник

foreach (Osxy figures in left\_triangle)

{

gl.Vertex(figures.x, figures.y);

}

gl.End();

gl.Begin(OpenGL.GL\_LINE\_LOOP);

// Указываем цвет вершин

gl.Color(1f, 0f, 0f);

//рисуем правый треугольник

foreach (Osxy figures in right\_triangle)

{

gl.Vertex(figures.x, figures.y);

}

// Завершаем работу

gl.End();

}

private void Button3\_Click(object sender, EventArgs e)

{

foreach (Osxy osxy in osxy\_rictangle)

{

osxy.y -= 0.1f;

}

foreach (Osxy osxy in left\_triangle)

{

osxy.y -= 0.1f;

}

foreach (Osxy osxy in right\_triangle)

{

osxy.y -= 0.1f;

}

drow\_figures();

}

private void Button4\_Click(object sender, EventArgs e)

{

foreach (Osxy osxy in osxy\_rictangle)

{

osxy.y += 0.1f;

}

foreach (Osxy osxy in left\_triangle)

{

osxy.y += 0.1f;

}

foreach (Osxy osxy in right\_triangle)

{

osxy.y += 0.1f;

}

drow\_figures();

}

private void Button5\_Click(object sender, EventArgs e)

{

foreach (Osxy osxy in osxy\_rictangle)

{

osxy.x \*= 1.1f;

osxy.y \*= 1.1f;

}

foreach (Osxy osxy in left\_triangle)

{

osxy.x \*= 1.1f;

osxy.y \*= 1.1f;

}

foreach (Osxy osxy in right\_triangle)

{

osxy.x \*= 1.1f;

osxy.y \*= 1.1f;

}

drow\_figures();

}

private void Button6\_Click(object sender, EventArgs e)

{

foreach (Osxy osxy in osxy\_rictangle)

{

osxy.x /= 1.1f;

osxy.y /= 1.1f;

}

foreach (Osxy osxy in left\_triangle)

{

osxy.x /= 1.1f;

osxy.y /= 1.1f;

}

foreach (Osxy osxy in right\_triangle)

{

osxy.x /= 1.1f;

osxy.y /= 1.1f;

}

drow\_figures();

}

private void Button7\_Click(object sender, EventArgs e)

{

angle+=10;

drow\_figures();

//double x = x \* Math.Cos(-30 \* Math.PI / 180) - y \* Math.Sin(30 \* Math.PI / 180);

// double y = x \* Math.Sin(-30 \* Math.PI / 180) + y \* Math.Cos(30 \* Math.PI / 180);

}

private void Button8\_Click(object sender, EventArgs e)

{

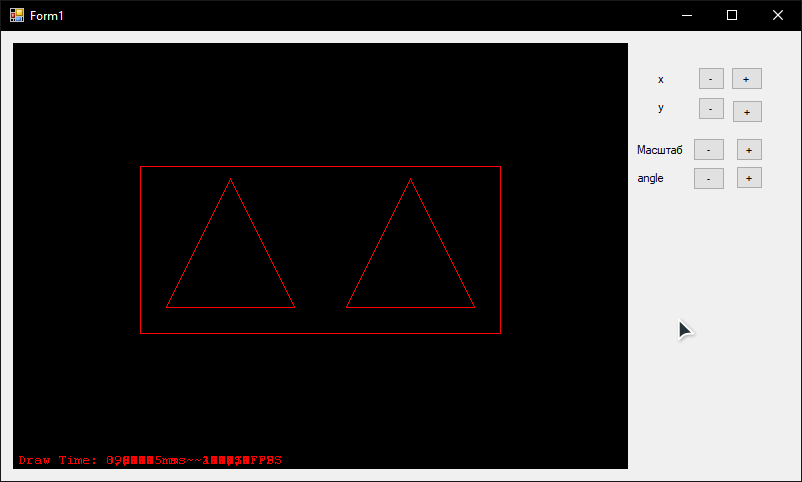
angle-=10;

drow\_figures();

}

}

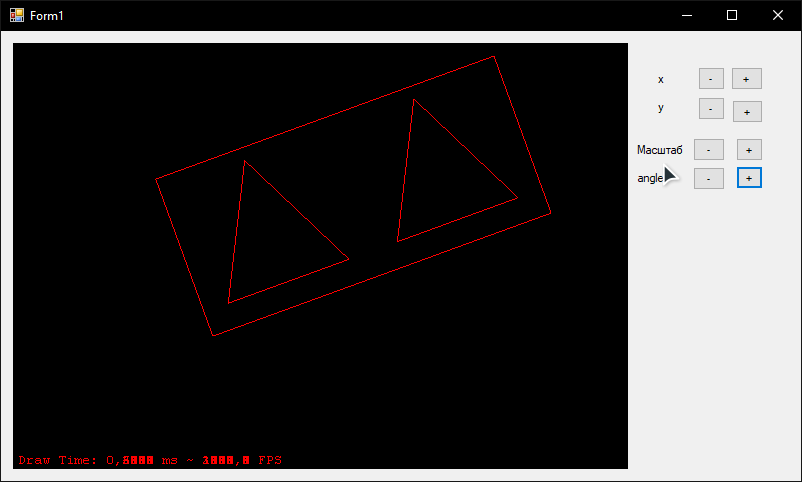
}



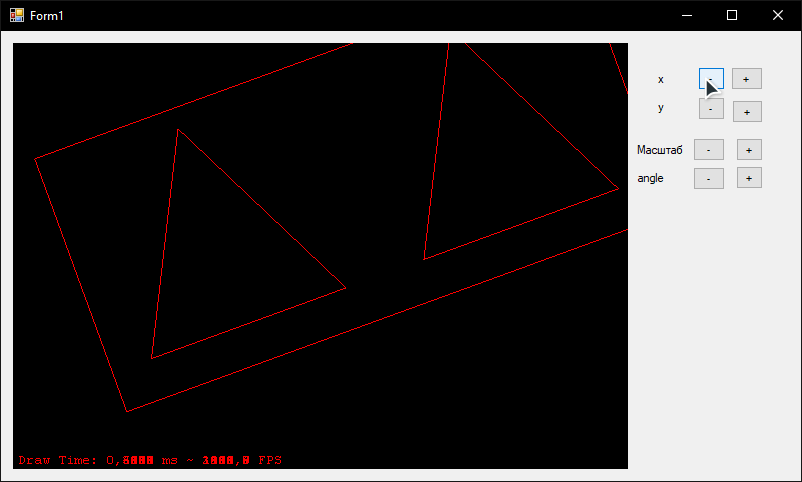
Скриншот 1 : запуск программы.



Скриншот 2 : перемещение по осям.



Скриншот 3: поворот фигуры.



Скриншот 4 : увеличение масштаба.

**Вывод:** в ходе выполнения работы провели геометрические преобразования двумерных объектов: масштабирование, перемещение, поворот.