

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра автоматизированной обработки информации (АОИ)

Графическое двумерное программирование с применением OpenGL

Отчет о выполнении лабораторной работы
по дисциплине «Компьютерная графика»

Студент гр. 429-3

_____ Бабец А. А.

«__» _____ 20__ г.

Принял:

канд. техн. наук, доцент каф.АОИ

_____ Т.О. Перемитина

«__» _____ 2021 г.

Томск 20__

Введение

Цель практической работы – получение навыков моделирования 2D сцены с применением OpenGL.

Задача - построить сцену, включающую две 2D фигуры (согласно варианту задания). С использованием событий KeyDown и/или MouseMove организовать возможность выполнения перечисленных преобразований:

- вращение фигуры А вокруг фигуры Б по и против часовой стрелке;
- поворот, масштабирование и сдвиг только фигуры А;
- поворот, масштабирование и сдвиг только фигуры Б;
- организовать вывод фигуры А в цикле.

.

Описание используемой среды программирования

Рассмотрим особенности среды программирования Microsoft Visual Studio:

- подсветка синтаксиса и простое автозавершение кода;
- анализ кода при загрузке и непосредственно при вводе;
- понятный и удобный интерфейс программы позволяет легко и быстро привыкнуть к работе и повышает её продуктивность;
- оснащённость инструментами для сборки, средой выполнения тестов, инструментами покрытия и встроенным терминальным окном.

Вышеперечисленные особенности послужили тому, что была выбрана среда разработки Microsoft Visual Studio для написания кода.

Описание метода решения

Для проведения преобразований фигур был создан метод, проводящий изменения по заданным параметрам и рисующий фигуры. Параметры можно изменять с помощью нажатий клавиш.

```
private void draw_figure()
{
    Gl.glViewport(0, 0, canvas.Width, canvas.Height);
    Gl.glClearColor(1f, 1f, 1f, 1);

    //фигура A
    Gl.glPushMatrix();
    Gl.glColor4f(1f, 0f, 1f, 0);
    Gl.glLineWidth(2);

    Gl.glTranslatef(-0.6f, 0f, 0);
    Gl.glTranslatef(xA, yA, 0);
    Gl.glRotatef(aA, 0, 0, 1);

    Gl.glTranslatef((B[1, 0] + B[8, 0]) / 2 + 0.6f, (B[0, 1] + B[3, 1]) / 2, 0);
    Gl.glRotatef(aaA, 0, 0, 1);
    Gl.glTranslatef(-(B[1, 0] + B[8, 0]) / 2 + 0.6f, -(B[0, 1] + B[3, 1]) / 2, 0);

    Gl.glScalef(sA, sA, 0);

    Gl.glBegin(Gl.GL_LINE_LOOP);
    Gl.glVertex2f(A[0, 0], A[0, 1]);
    Gl.glVertex2f(A[1, 0], A[1, 1]);
    Gl.glVertex2f(A[2, 0], A[2, 1]);
    Gl.glVertex2f(A[3, 0], A[3, 1]);
    Gl.glVertex2f(A[4, 0], A[4, 1]);
    Gl.glVertex2f(A[5, 0], A[5, 1]);
    Gl.glVertex2f(A[6, 0], A[6, 1]);
    Gl.glVertex2f(A[7, 0], A[7, 1]);
    Gl.glEnd();
}
```

Рисунок 1 –начало метода изменения и отрисовки фигуры.

```

Gl.glPointSize(6f);
Gl.glEnable(Gl.GL_POINT_SMOOTH);
Gl.glBegin(Gl.GL_POINTS);
    Gl.glVertex2f(A[0, 0], A[0, 1]);
    Gl.glVertex2f(A[1, 0], A[1, 1]);
    Gl.glVertex2f(A[2, 0], A[2, 1]);
    Gl.glVertex2f(A[3, 0], A[3, 1]);
    Gl.glVertex2f(A[4, 0], A[4, 1]);
    Gl.glVertex2f(A[5, 0], A[5, 1]);
    Gl.glVertex2f(A[6, 0], A[6, 1]);
    Gl.glVertex2f(A[7, 0], A[7, 1]);
Gl.glEnd();

Gl.glColor3f(0, 1, 0);
Gl.glBegin(Gl.GL_LINE_LOOP);
    Gl.glVertex2f(A[0, 0], A[0, 1]);
    Gl.glVertex2f(A[2, 0], A[2, 1]);
    Gl.glVertex2f(A[4, 0], A[4, 1]);
    Gl.glVertex2f(A[6, 0], A[6, 1]);
Gl.glEnd();

Gl.glColor3f(0, 0, 0);
Gl.glLineWidth(1);
Gl.glBegin(Gl.GL_LINES);
    Gl.glVertex2f(A[0, 0], A[0, 1]);
    Gl.glVertex2f(A[4, 0], A[4, 1]);
    Gl.glVertex2f(A[2, 0], A[2, 1]);
    Gl.glVertex2f(A[6, 0], A[6, 1]);
Gl.glEnd();

Gl.glColor3f(0, 0, 0);
Gl.glBegin(Gl.GL_POINTS);
    Gl.glVertex2f((A[3, 0] + A[7, 0]) / 2, (A[0, 1] + A[4, 1]) / 2);
Gl.glEnd();

Gl.glPopMatrix();

//фигура B
Gl.glLineWidth(2);
Gl.glPushMatrix();

Gl.glTranslatef(xB, yB, 0);
Gl.glRotatef(aB, 0, 0, 1);
Gl.glScalef(sB, sB, 0);

```

Рисунок 2 – продолжение метода изменения и отрисовки фигуры.

```

Gl.glBegin(Gl.GL_LINE_LOOP);
    Gl.glVertex2f(B[1, 0], B[1, 1]);
    Gl.glVertex2f(B[2, 0], B[2, 1]);
    Gl.glVertex2f(B[4, 0], B[4, 1]);
    Gl.glVertex2f(B[8, 0], B[8, 1]);
Gl.glEnd();

Gl.glBegin(Gl.GL_LINE_LOOP);
    Gl.glVertex2f(B[0, 0], B[0, 1]);
    Gl.glVertex2f(B[3, 0], B[3, 1]);
    Gl.glVertex2f(B[5, 0], B[5, 1]);
    Gl.glVertex2f(B[6, 0], B[6, 1]);
    Gl.glVertex2f(B[7, 0], B[7, 1]);
Gl.glEnd();

Gl.glColor3f(0, 0, 0);
Gl.glBegin(Gl.GL_POINTS);
    Gl.glVertex2f(B[0, 0], B[0, 1]);
    Gl.glVertex2f(B[3, 0], B[3, 1]);
    Gl.glVertex2f(B[4, 0], B[4, 1]);
    Gl.glVertex2f(B[8, 0], B[8, 1]);
Gl.glEnd();

Gl.glPopMatrix();

canvas.Invalidate();
}

```

Рисунок 3 – конец метода изменения и отрисовки фигуры.

При запуске программы и нажатии на кнопку «Reset» в разделе «Figure» в верхнем меню происходит отрисовка фигур в начальном положении.

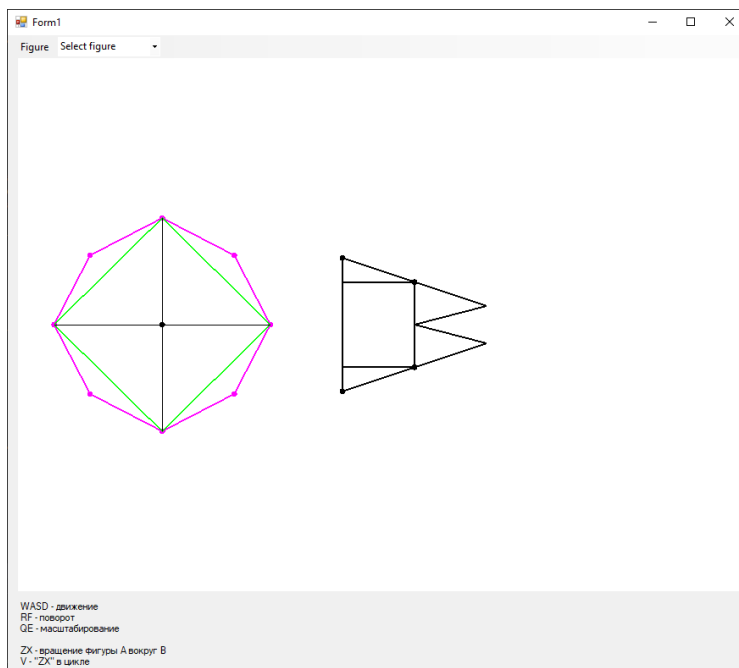


Рисунок 4 – начальное положение фигур.

Далее в верхнем меню нужно выбрать фигуру, используя объект «Select Figure».

Далее можно управлять выбранной фигурой используя клавиши.

Клавиши WASD отвечают за перемещение фигуры

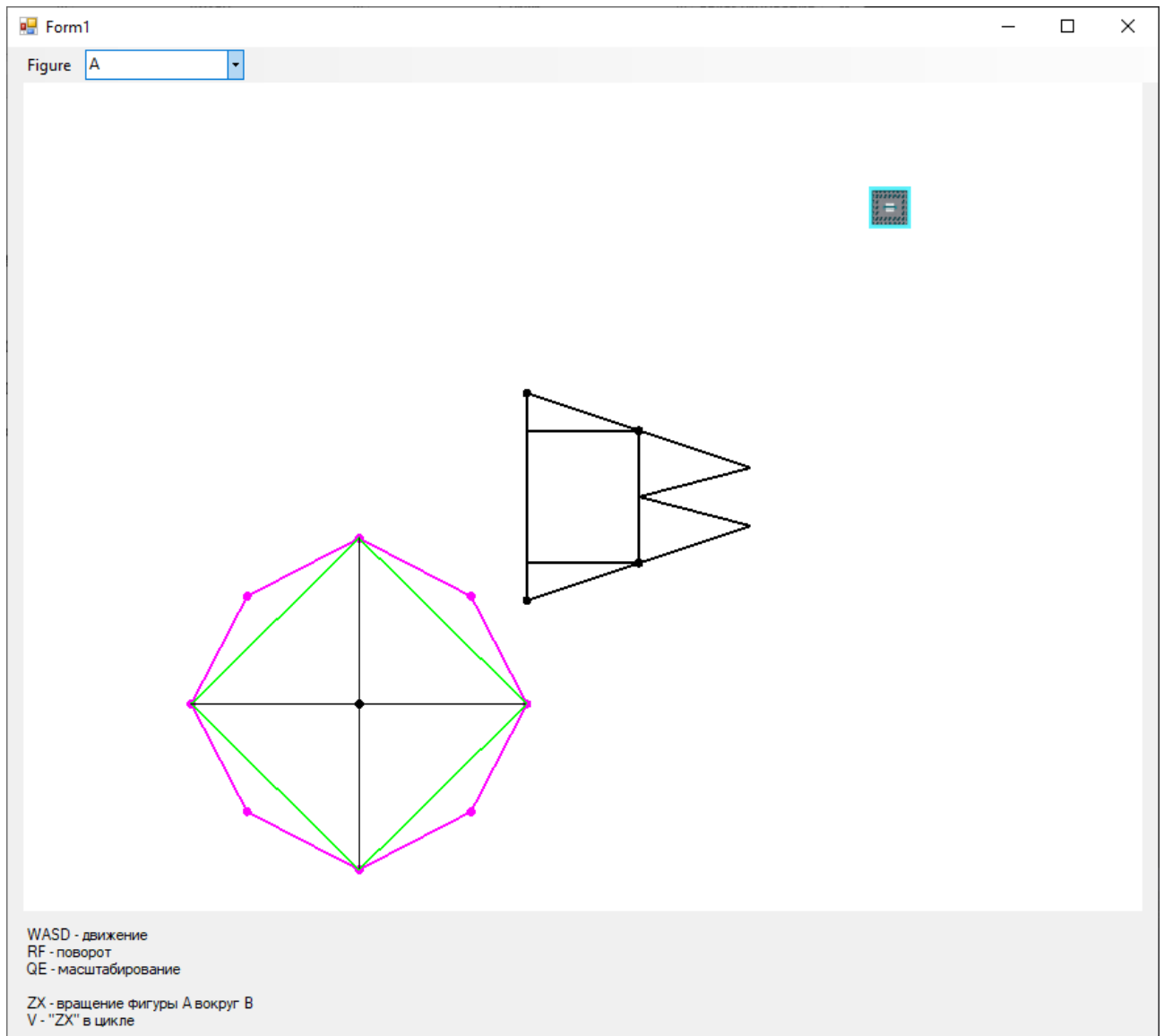


Рисунок 5 – перемещение фигуры A

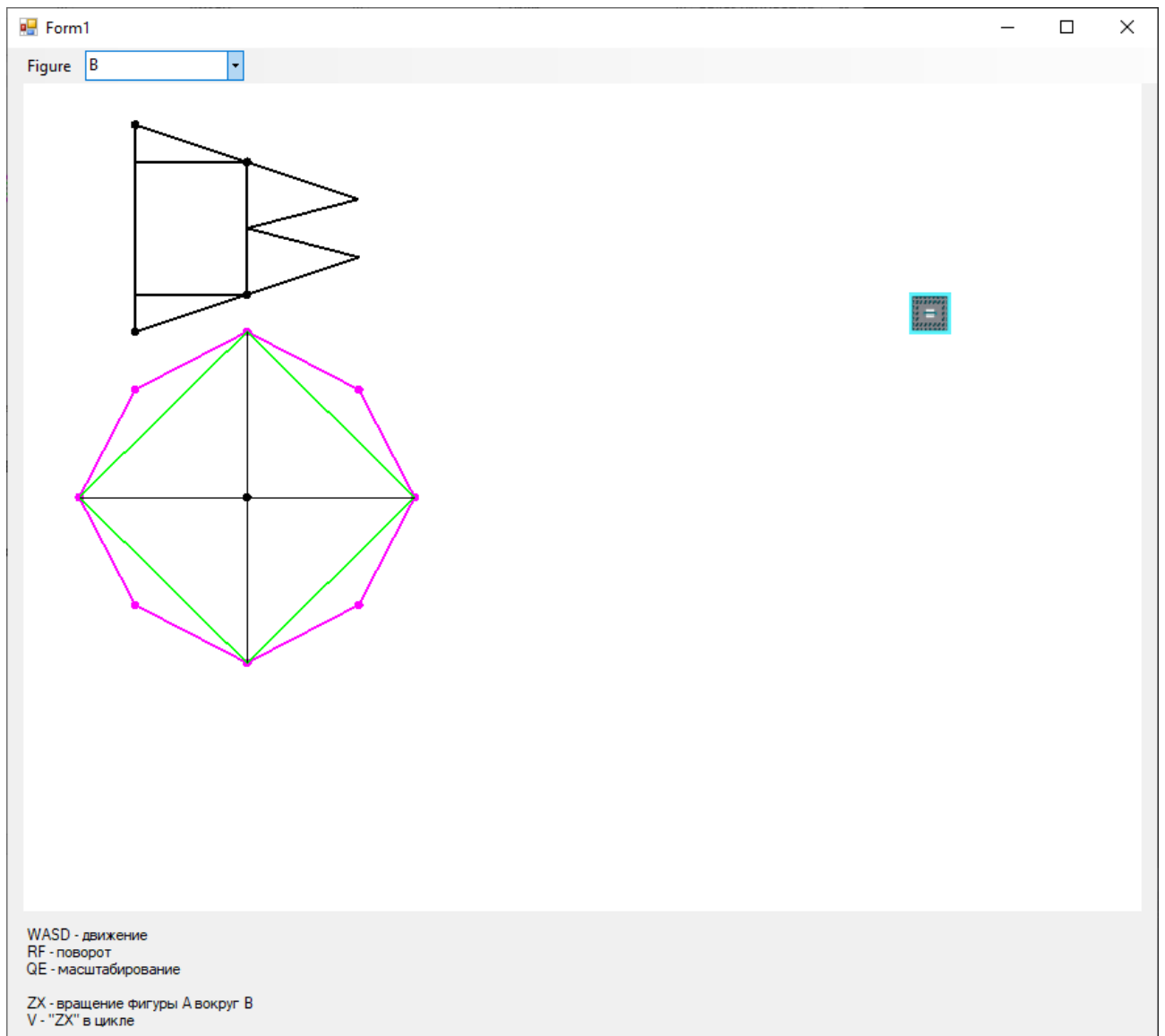


Рисунок 5 – перемещение фигуры В.

Клавиши RF отвечают за поворот фигуры против и по часовой стрелке.

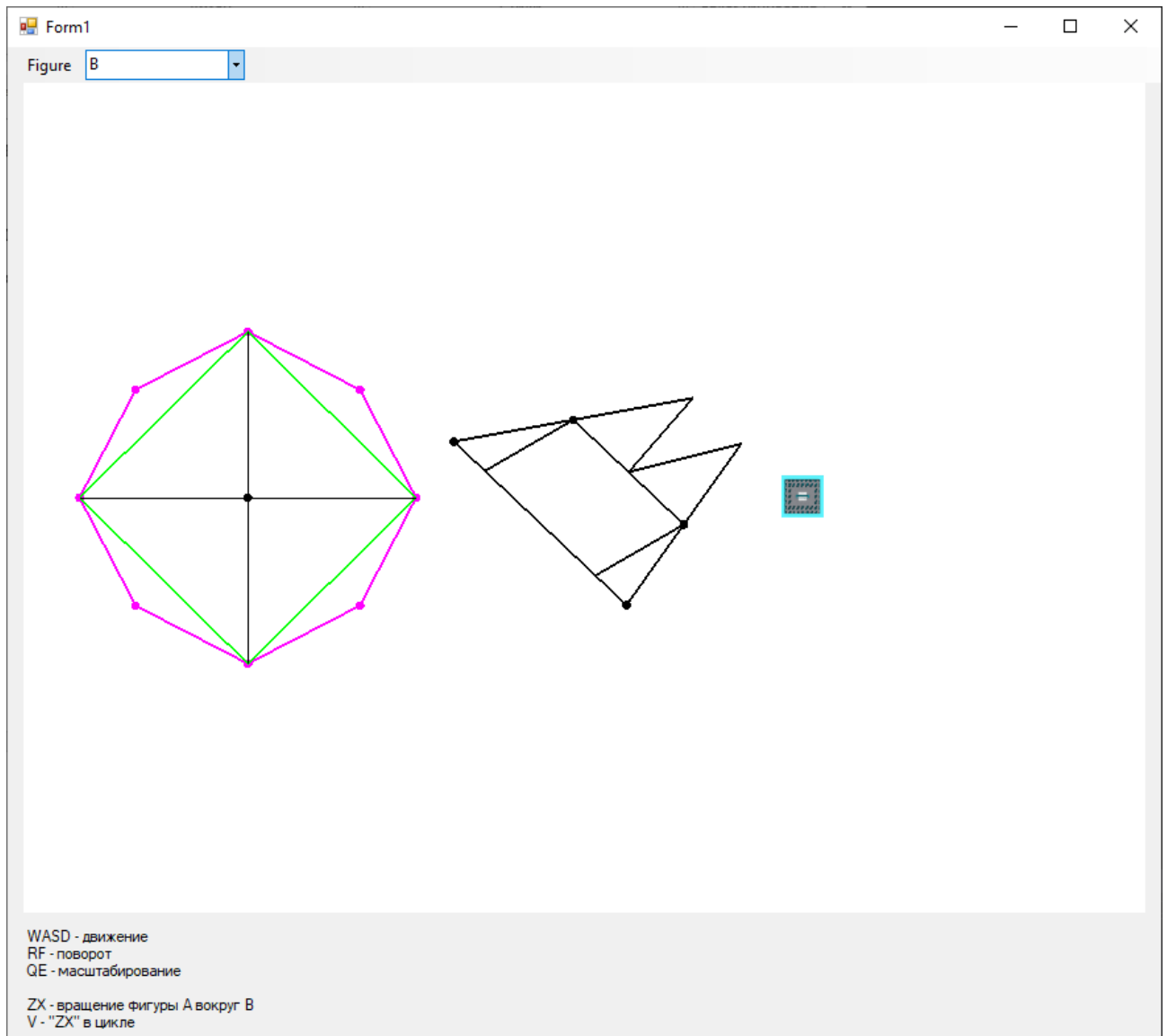


Рисунок 6 – поворот против часовой стрелки.

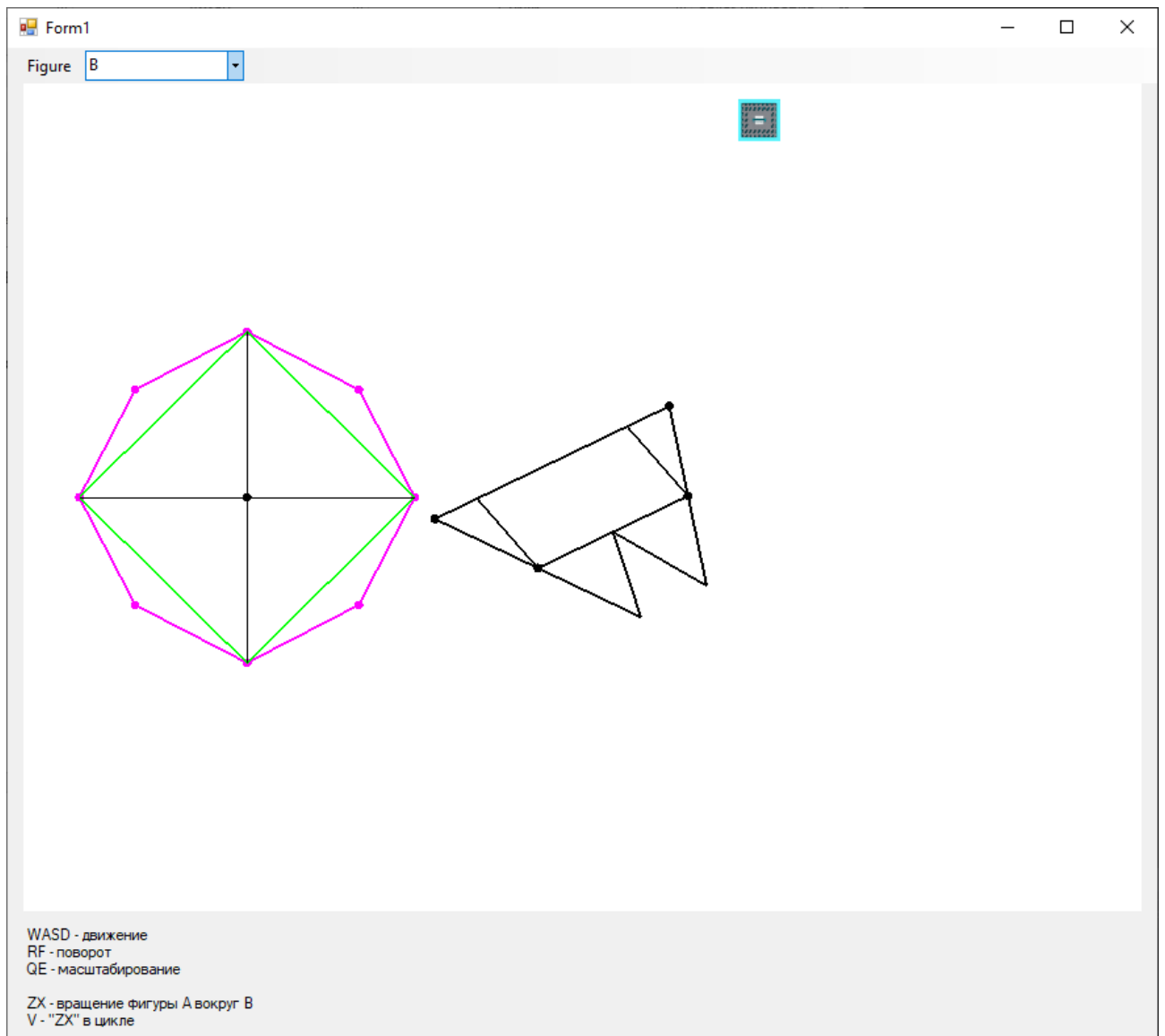


Рисунок 7 – поворот по часовой стрелке.

Клавиши QE отвечают за масштабирование, увеличение и уменьшение соответственно.

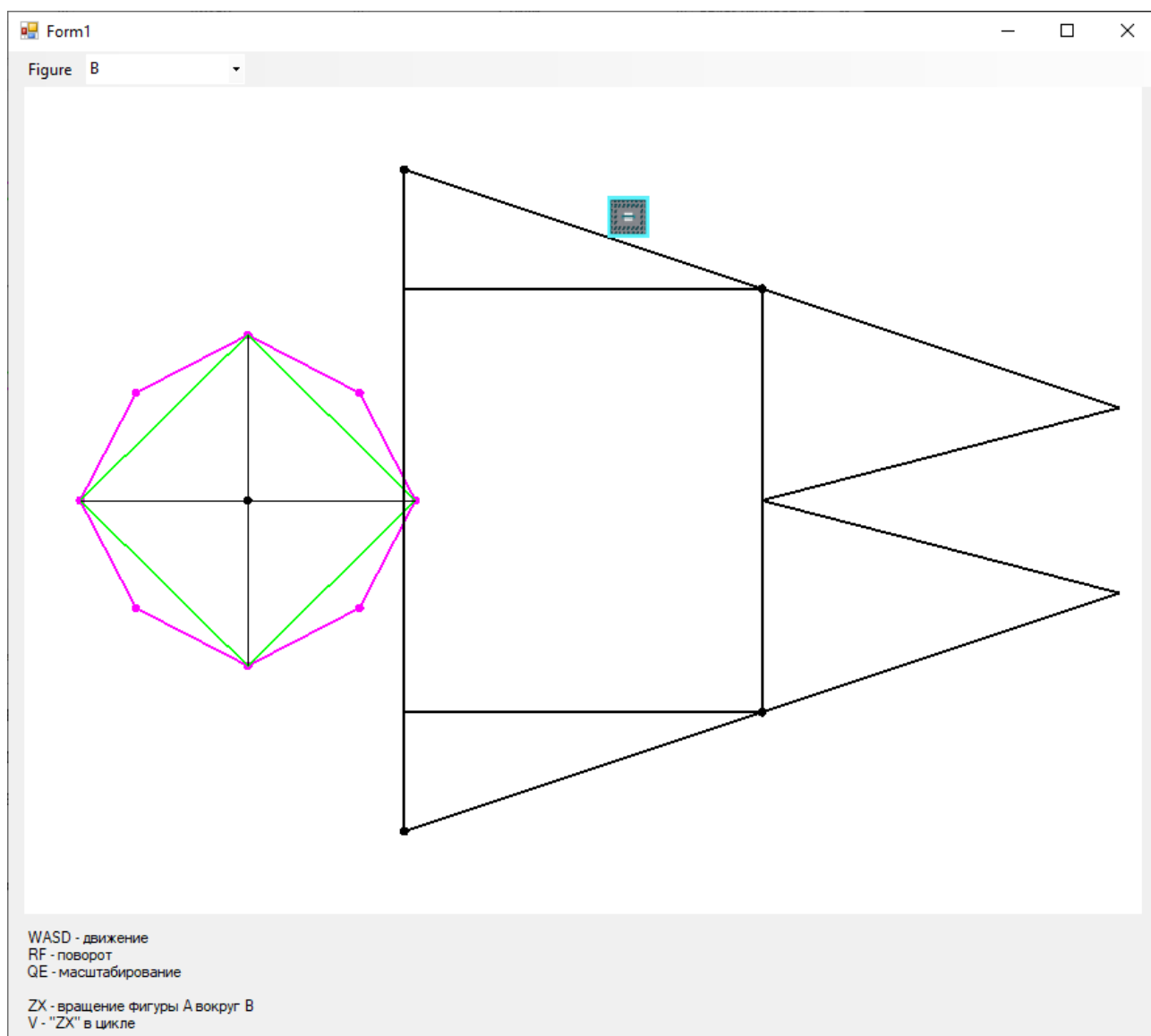


Рисунок 8 – увеличение фигуры

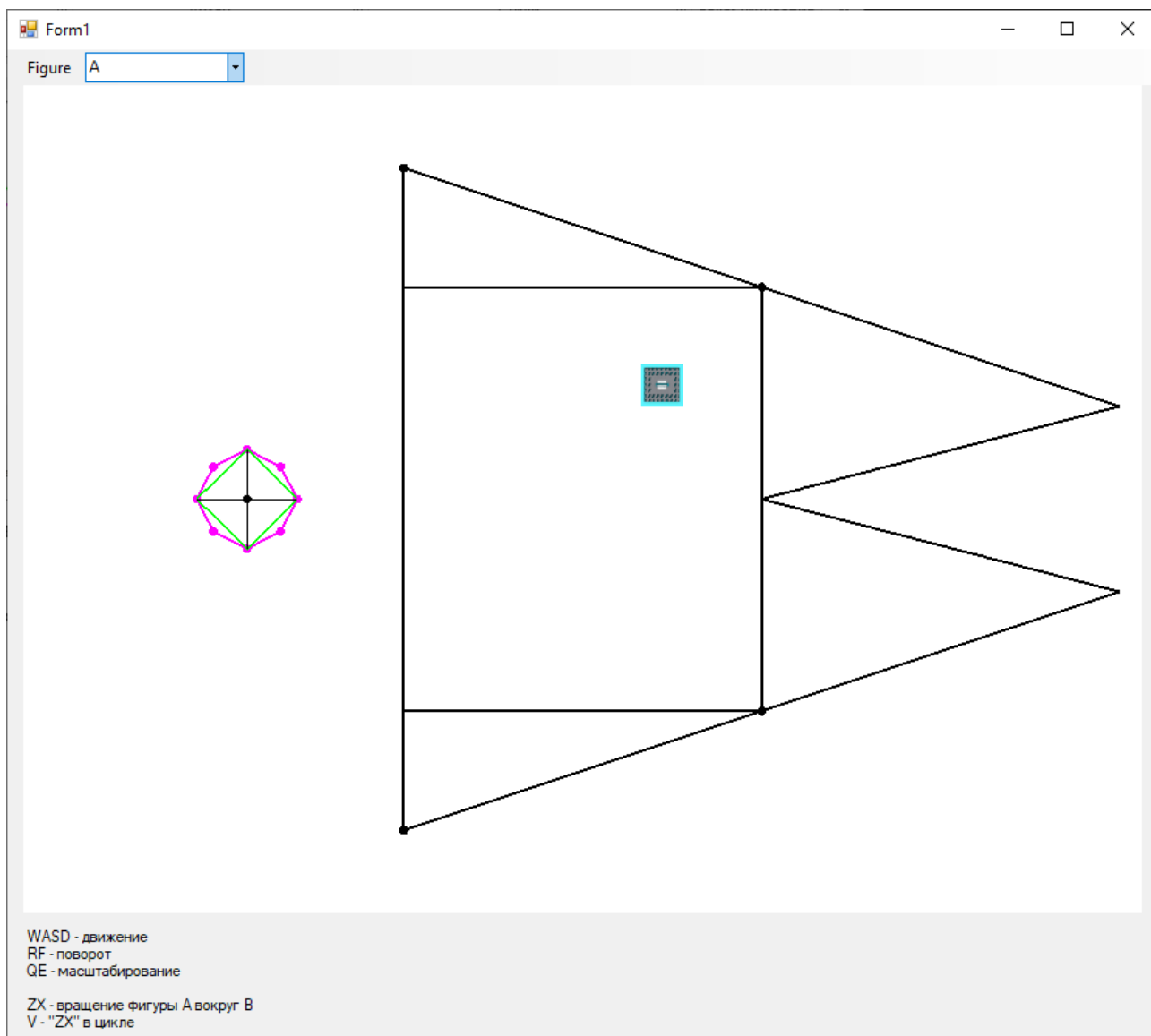


Рисунок 9 – уменьшение фигуры.

Клавиши ZX отвечают за вращение фигуры A вокруг фигуры B.

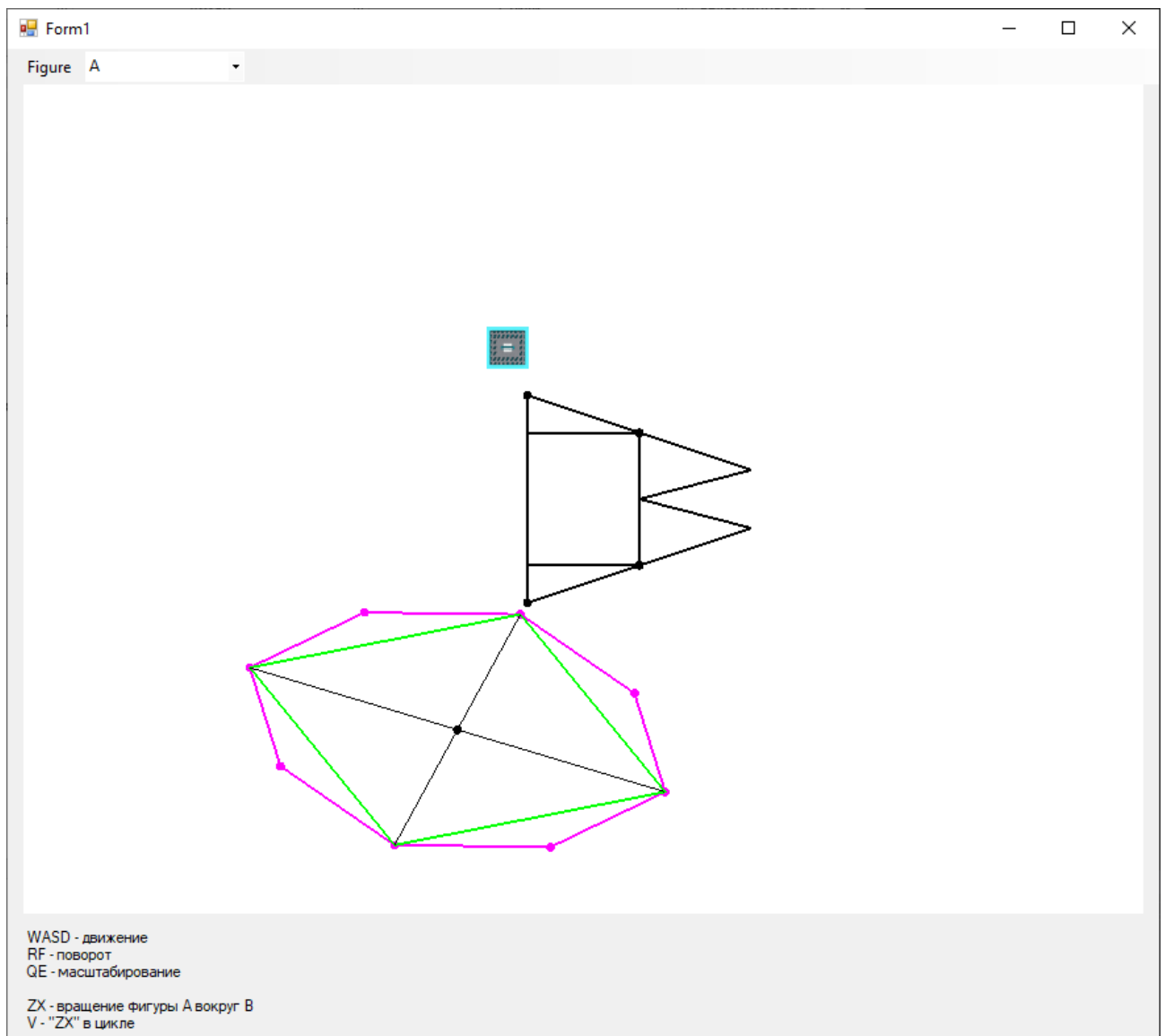


Рисунок 10 – вращение фигуры A вокруг фигуры B.

При нажатии на клавишу V происходит циклическая отрисовка фигуры A с поворотом вокруг фигуры B.

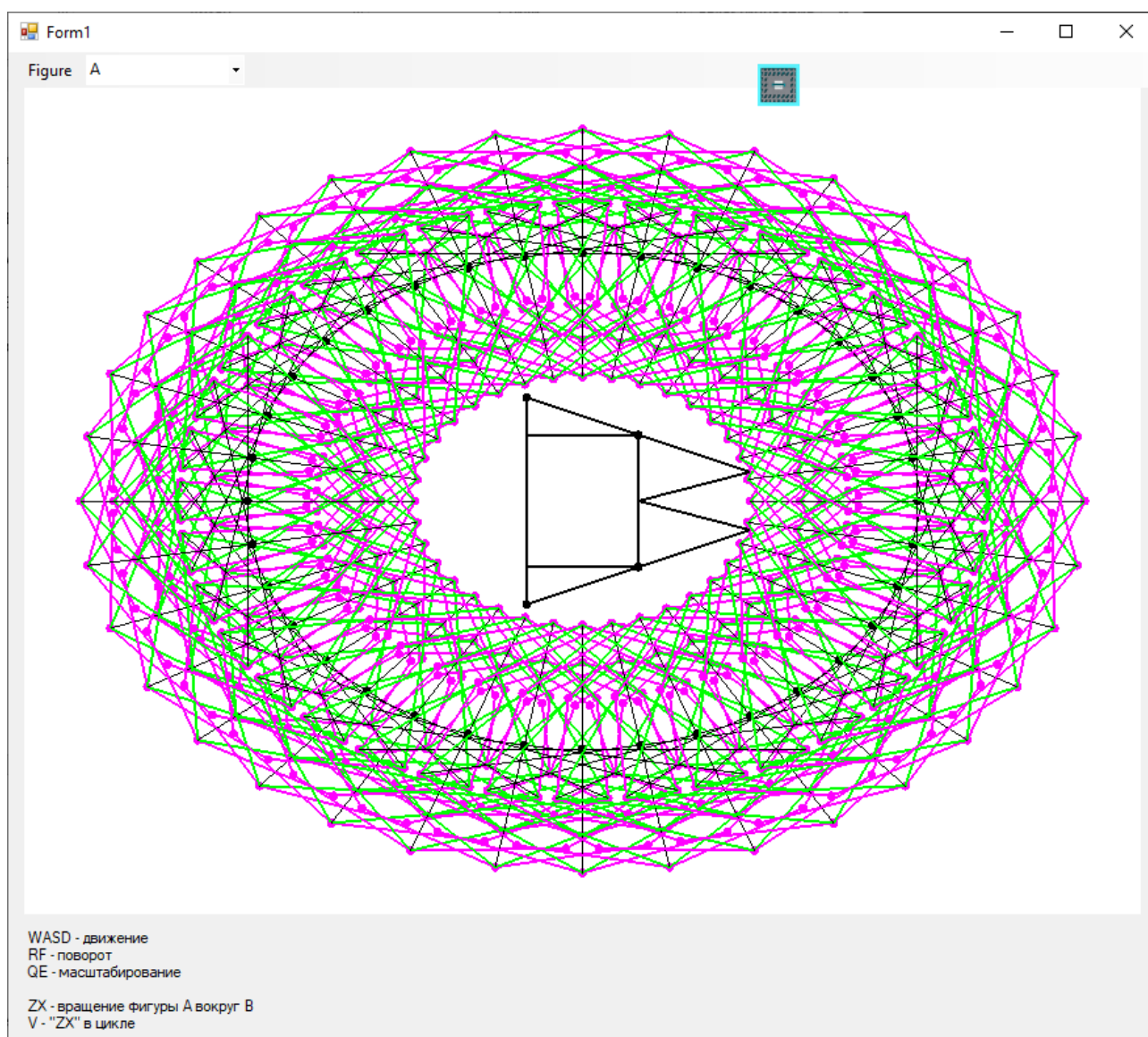


Рисунок 11 - циклическая отрисовка фигуры A с поворотом вокруг фигуры B.

Заключение

В ходе практической работы были получены навыки моделирования 2D сцены с применением OpenGL.