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)

Трёхмtрное графическое программирование с применением OpenGL

Отчет о выполнении практической 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 трёхмерных объектов при помощи 2D примитивов библиотеки OpenGL.

Задача - Построить трехмерную фигуру с применением простейших примитивов OpenGL. Вращать фигуру по таймеру и с помощью клавиш управления курсором. Организовать работу спримененим z-буфера и проективных преобразований glFrustum.

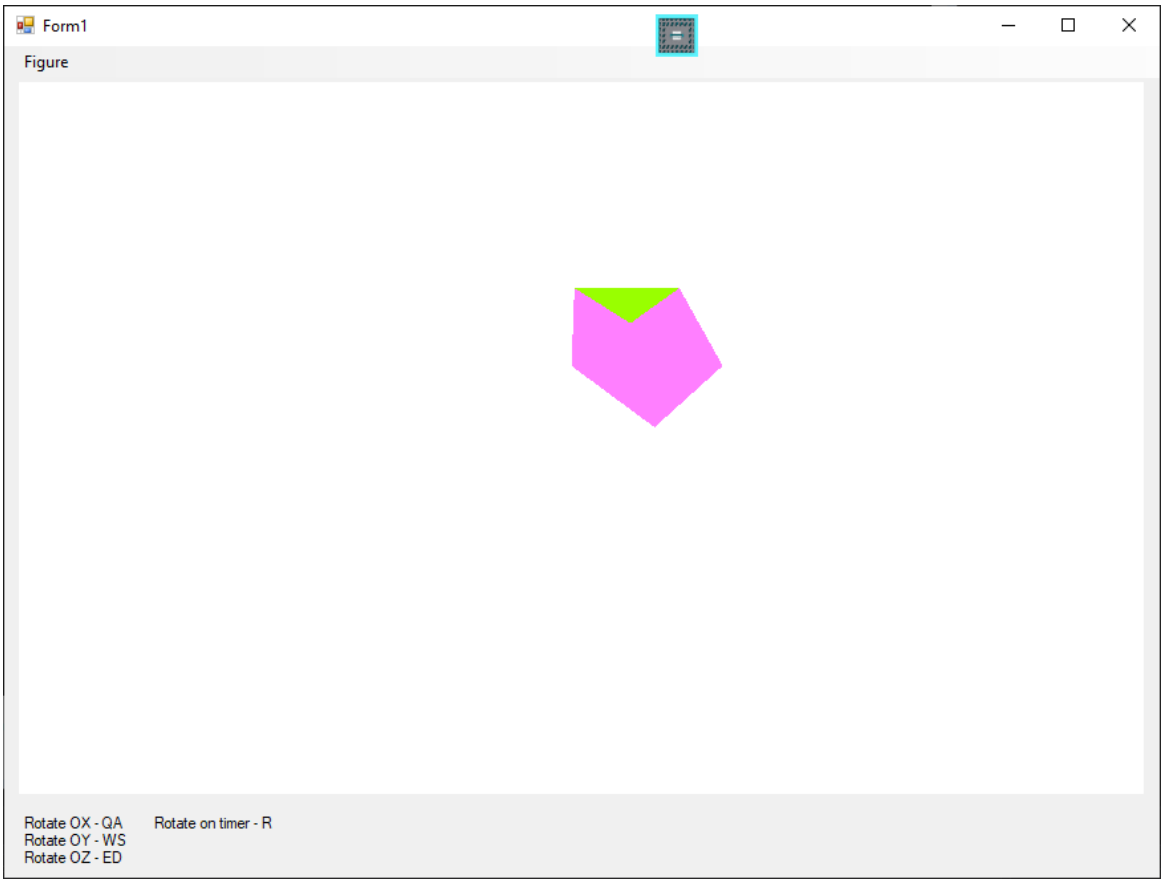
Описание используемой среды программирования

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;
- оснащенность 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.

Ход работы



2

Рисунок 1 – Интерфейс программы

Для начала работы требуется задать проекцию и активировать z-буфер.

```
private void Form1_Load(object sender, EventArgs e)
{
    Gl.glEnable(Gl.GL_DEPTH_TEST);
    Gl.glMatrixMode(Gl.GL_PROJECTION);
    Gl.glLoadIdentity();
    Gl.glFrustum(-1f, 1f, -1f, 1f, 3f, 10f);
    Gl.glMatrixMode(Gl.GL_MODELVIEW);
    Gl.glLoadIdentity();
    Gl.glTranslatef(0f, 0f, -8.0f);
    Gl.glRotatef(30, 1, 0, 0);
}
```

Рисунок 2.2 – Задание проекции и активация z-буфера

Следующий этап – построить фигуру. В данном случае усечённая пирамида построена с помощью GL_TRIANGLES – верх и низ, и GL_QUAD_STRIP - бока.

```
Ссылок: 8
public void DrawFigure()
{
    Gl.glClearColor(1f, 1f, 1f, 1);
    Gl.glClear(Gl.GL_COLOR_BUFFER_BIT);
    Gl.glClear(Gl.GL_DEPTH_BUFFER_BIT);

    Gl.glPushMatrix();

    Gl.glRotatef(xAngle, 1, 0, 0);
    Gl.glRotatef(yAngle, 0, 1, 0);
    Gl.glRotatef(zAngle, 0, 0, 1);

    //крышка
    Gl.glColor3f(0.6f, 1, 0);
    Gl.glBegin(Gl.GL_TRIANGLES);
        for (int i = 0; i < 3; i++)
        {
            Gl.glVertex3f(figure[i, 0], figure[i, 1], figure[i, 2]);
        }
    Gl.glEnd();

    //дно
    Gl.glColor3f(0.6f, 0, 0.6f);
    Gl.glBegin(Gl.GL_TRIANGLES);
        for (int i = 3; i < 6; i++)
        {
            Gl.glVertex3f(figure[i, 0], figure[i, 1], figure[i, 2]);
        }
    Gl.glEnd();

    //бока
    Gl.glColor3f(1, 0.5f, 1);
    Gl.glBegin(Gl.GL_QUAD_STRIP);
        for (int i = 0; i < 3; i++)
        {
            Gl.glVertex3f(figure[i, 0], figure[i, 1], figure[i, 2]);
            Gl.glVertex3f(figure[i+3, 0], figure[i+3, 1], figure[i+3, 2]);

        }
        Gl.glVertex3f(figure[0, 0], figure[0, 1], figure[0, 2]);
        Gl.glVertex3f(figure[3, 0], figure[3, 1], figure[3, 2]);
    Gl.glEnd();

    Gl.glPopMatrix();

    canvas.Invalidate();
}
```

Рисунок 2.3 – Отрисовка фигуры

При нажатии на кнопку Figure –Reset происходит обнуление всех углов поворота и отрисовка фигуры в изначальном виде:

С помощью клавиш QAWSEDR можно управлять вращением фигуры.

QA, WS, ED – задают изменение угла поворота по осям OX, OY и OZ соответственно, а клавиша R активирует вращение по таймеру.

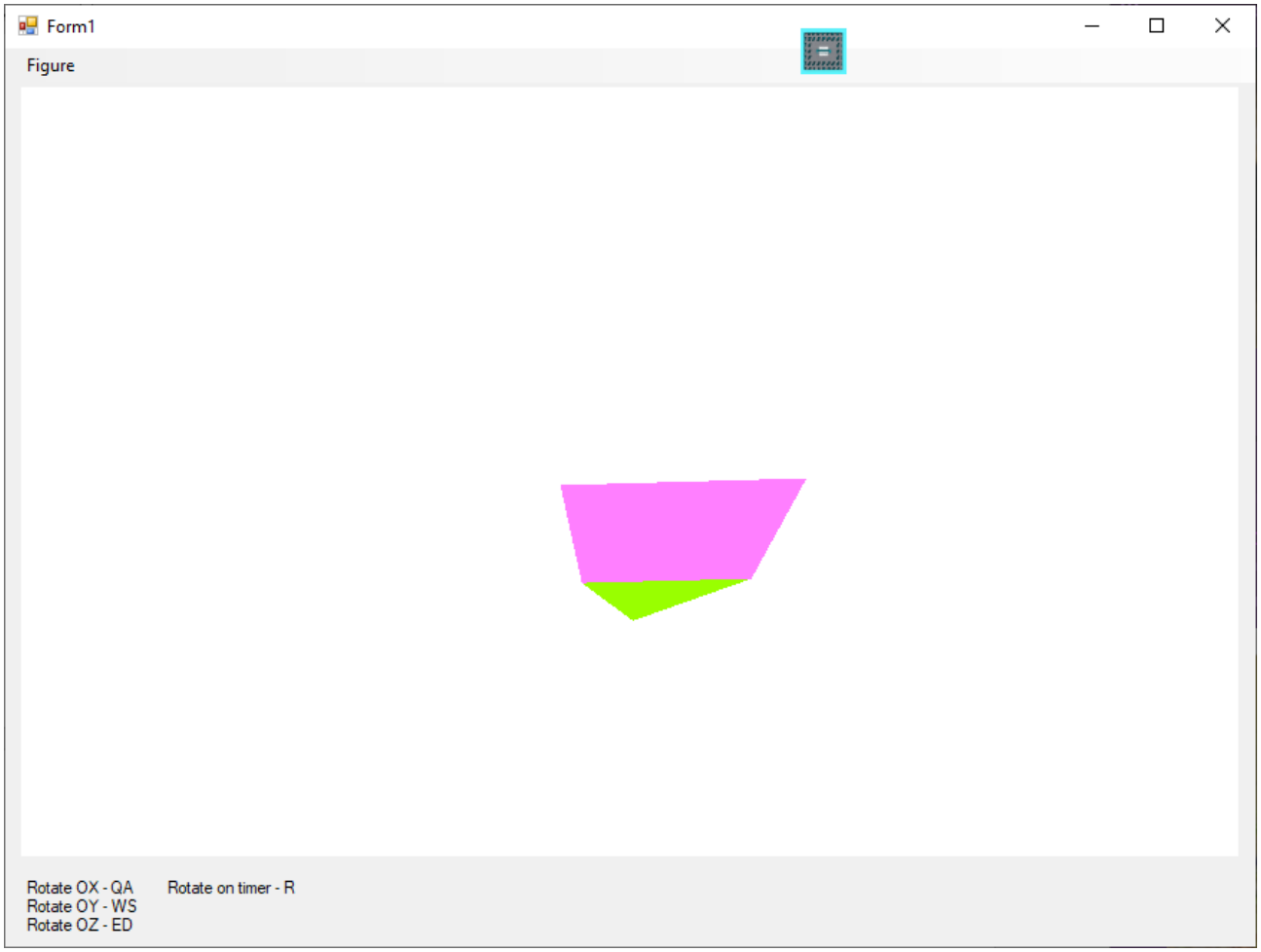


Рисунок 2.4 - Вращение фигуры

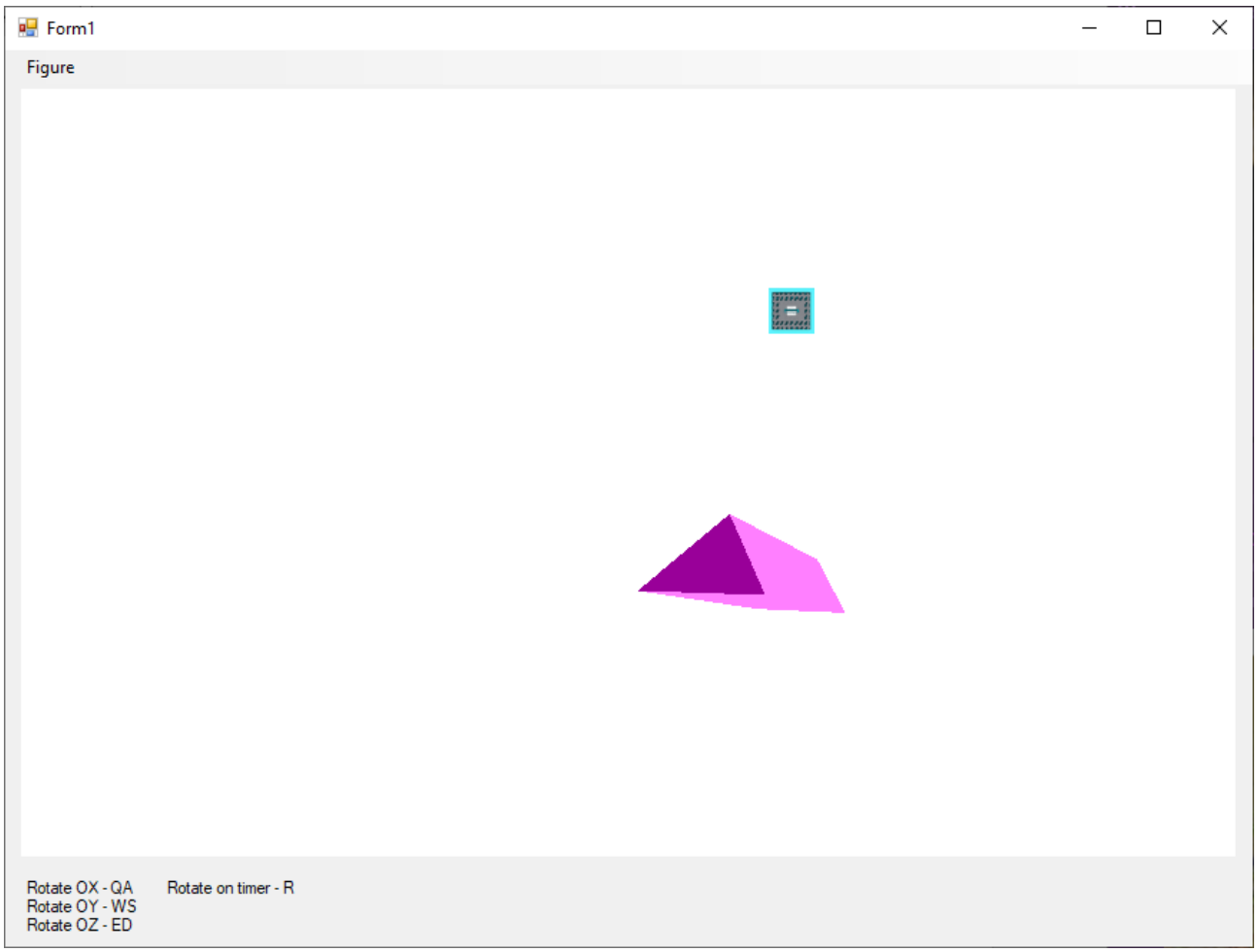


Рисунок 2.5 – Вращение фигуры

3 3 Ответы на вопросы

1. A, C, E, F
2. B, D
3. B, D, F
4. B, D
5. A
6. A
7. A
8. C
9. B
10. C
11. A
12. A

13.B

14.B

15.A

16.B

17.A

18.B

19.C

20.B