

1. Задание

Создать приложение, выводящее объемный объект средствами Open GL.
Среда программирования – Visual C++ или QT Creator с MinGW и Open GL.

2. Текст программы

```
#include <glut.h>

GLfloat rx = 0;           // Угол поворота сцены вокруг оси X
GLfloat ry = 0;           // Угол поворота сцены вокруг оси Y

int mx, my;               // Координаты мыши
bool ldown = false;       // Нажата левая клавиша мыши?
bool rdown = false;       // Нажата правая клавиша мыши?

GLfloat vertices[] =      // массив вершин
{
    0.5f, 0.5f, 0.5f,      // Верх право квадрата (верх)
    -0.5f, 0.5f, 0.5f,    // Верх лево
    -0.5f, 0.5f, -0.5f,   // Низ лево
    0.5f, 0.5f, -0.5f,    // Низ право

    0.5f, -0.5f, -0.5f,    // Верх право квадрата (низ)
    -0.5f, -0.5f, -0.5f,   // Верх лево
    -0.5f, -0.5f, 0.5f,    // Низ лево
    0.5f, -0.5f, 0.5f,     // Низ право

    0.5, 0.5, -0.5,        // Право верх квадрата (Перед)
    -0.5, 0.5, -0.5,       // Лево верх
    -0.5, -0.5, -0.5,      // Лево низ
    0.5, -0.5, -0.5,       // Право низ

    -0.5f, 0.5f, 0.5f,     // Верх право квадрата (Зад)
    0.5f, 0.5f, 0.5f,      // Верх лево
    0.5f, -0.5f, 0.5f,     // Низ лево
    -0.5f, -0.5f, 0.5f,    // Низ право

    -0.5f, 0.5f, -0.5f,    // Верх право квадрата (слева)
    -0.5f, 0.5f, 0.5f,     // Верх лево
    -0.5f, -0.5f, 0.5f,    // Низ лево
    -0.5f, -0.5f, -0.5f,   // Низ право

    0.5f, 0.5f, 0.5f,      // Верх право квадрата (право)
    0.5f, 0.5f, -0.5f,     // Верх лево
    0.5f, -0.5f, -0.5f,    // Низ лево
    0.5f, -0.5f, 0.5f,     // Низ право
};

void Display()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glEnable(GL_DEPTH_TEST);
    //включение теста глубины

    glEnableClientState(GL_VERTEX_ARRAY);
    // Включаем режим вершинных массивов.
    glColor4f(0.1,0.5,0.3,1.0);
    glVertexPointer(3, GL_FLOAT, 3*sizeof(float), vertices);
}
```

```

    glDrawArrays(GL_QUADS, 0, 24);

    glRotatef(rx, 1, 0, 0);
    glRotatef(ry, 0, 1, 0);

    glutSwapBuffers();
    // меняем буферы т.к GLUT_DOUBLE
    glFlush();
}

void mouseButton(int button, int state, int x, int y) // функция обработки
событий нажатия клавиши
{
    if (button == GLUT_LEFT_BUTTON) //Левая кнопка
    {
        switch (state)
        {
            case GLUT_DOWN: //Если нажата
                ldown = true; //установить флаг
                mx = x; //Запомнить
                my = y;
                break;
            case GLUT_UP: // Если отжата
                ldown = false; // установить флаг
                break;
        }
    }
}

void mouseMove(int x, int y) // функция для считывания
{
    if (ldown) // Левая кнопка нажата
    {
        rx += 0.001 * (y - my); //Изменение угла поворота
        ry += 0.001 * (x - mx);
        mx = x;
        my = y;
        glutPostRedisplay(); //Перерисовать экран
    }
}

void Reshape(int w, int h) // функция для пропорционального изменения
приметива
{
    if (h == 0)
        // предотвращение деления на 0
    {
        h = 1;
    }
    glViewport(0, 0, w, h);
    // задаём область вывода изображения
    glMatrixMode(GL_PROJECTION);
    // устанавливаем матрицу проекций
    glLoadIdentity();
    // делаем её единичной
    glOrtho(20.0, 20.0, 20.0, 20.0, 50.0, -50.0); //
    установка двумерной ортографической системы координат
    glMatrixMode(GL_MODELVIEW);
    // режим видовой матрицы
    glLoadIdentity();
    // делаем её единичной
}

```

```
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA);
    // включаем двойную буферизацию
    glutInitWindowSize(800, 600);
    // задаём размер окна
    glutInitWindowPosition((glutGet(GLUT_SCREEN_WIDTH) - 800) / 2,
(glutGet(GLUT_SCREEN_HEIGHT) - 600) / 2); // начальные координаты окна

    //(отсчитываются с левого верхнего угла)
    glutCreateWindow("OpenGL lesson 1");
    // называем окно

    glutReshapeFunc(Reshape);
    // вызов функции масштабирования
    glutDisplayFunc(Display);
    // вызов функции рисования

    glutMouseFunc(mouseButton);
    glutMotionFunc(mouseMove);

    glutMainLoop();
    // запуск всего инициализированного (аналогия
main)

    return 0;
}
```

4. Приложение

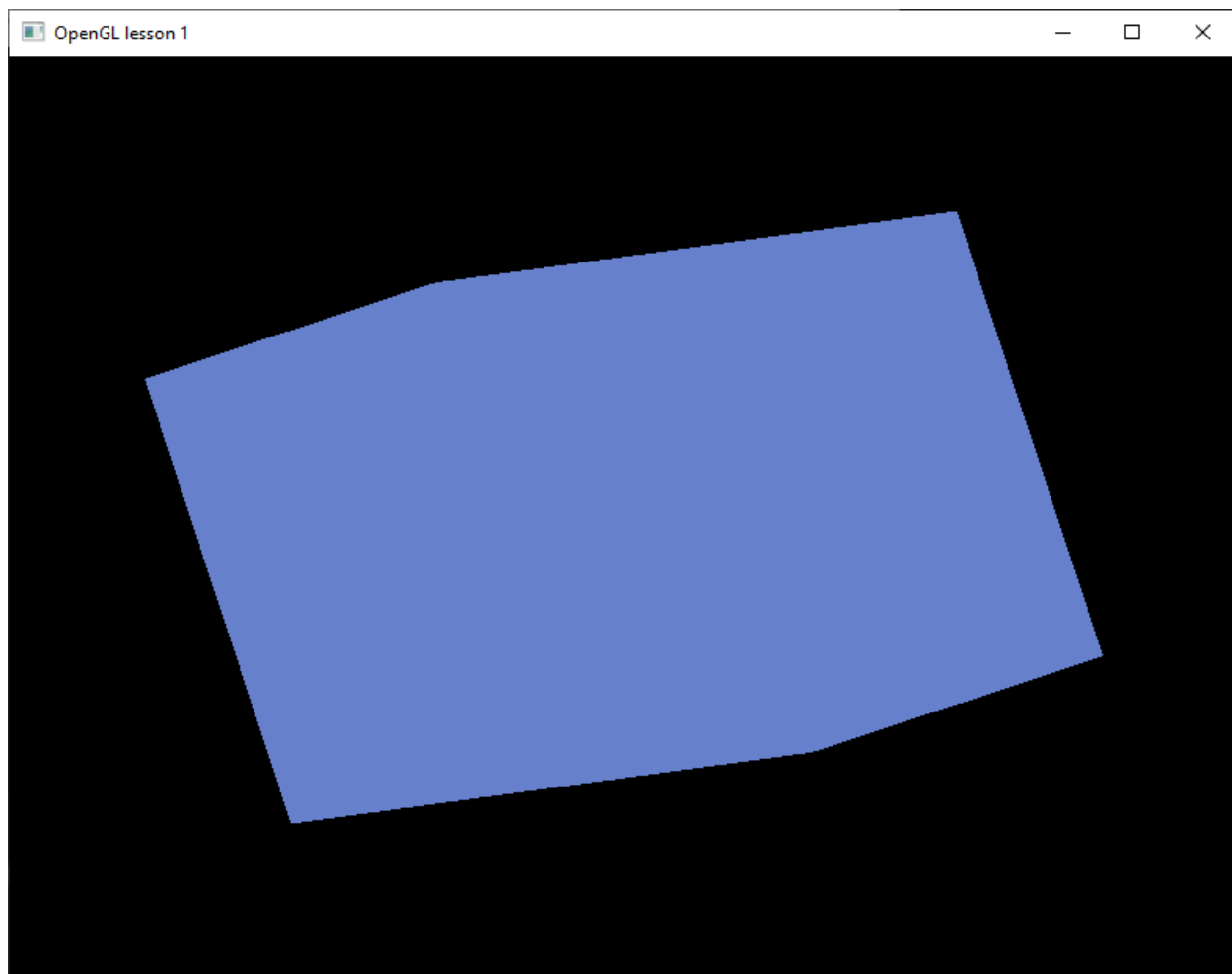


Рисунок 1 – Пример работы программы