

ЛАБОРАТОРНАЯ РАБОТА №1**По дисциплине «Алгоритмические средства компьютерной графики»****ТЕМА. Способы задания двумерных объектов**

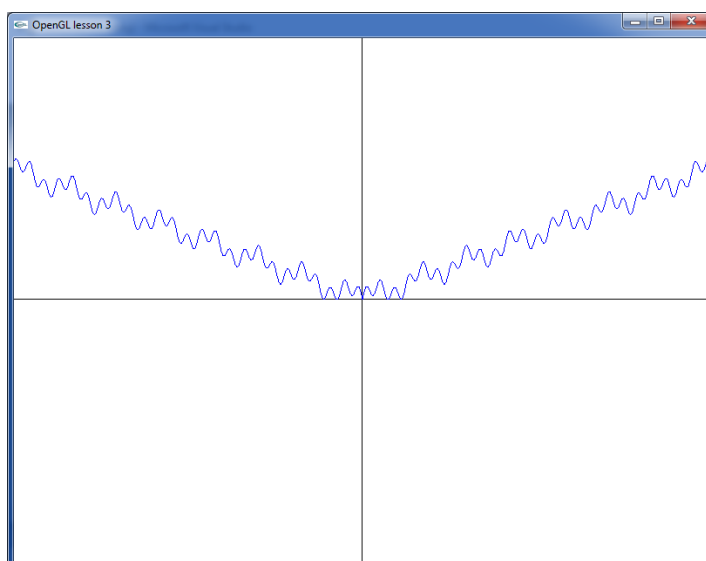
Цель работы: Реализовать различные способы задания объектов и поработать с системой координат в OpenGL с помощью функции gluOrtho2D

Порядок работы:**Задание 1.**

- 1) В среде Visual Studio создайте пустое решение (solution) с именем LR1. В нем создайте консольный проект на языке C++ Win32 Console Application. Назовите его LR1_1.
- 2) В проект добавьте пустой файл с исходным кодом source.cpp
- 3) Напишите программу, которая рисует график функции:

$$y = \left| \frac{1}{4}x + 3 \cos(100x) \sin x \right|$$

для x в интервале от -100 до 100 с шагом 0.5. Добавьте прорисовку осей OX и OY. Задайте размеры мирового окна таким образом, что бы весь график помещался в окно. Например, как показано на рисунке:

**Задание 2.**

- 1) В решении LR1 в среде Visual Studio создайте новый консольный проект на языке C++ Win32 Console Application. Назовите его LR1_2.
- 2) В проект добавьте пустой файл с исходным кодом source.cpp
- 3) Наберите следующий код программы:

```
#include<Windows.h>
#include <GL/glut.h>    /*Для Linux и Windows*/
#include<vector>
#include<fstream>
```

```
using namespace std;

struct Point
{
    int x, y;
};
void reshape(int w, int h);
void display();
void readFromFile();
void lineto(Point p);
void moveto(Point p);
vector<Point> point;
vector<int> code;
Point currentPoint;

int main(int argc, char * argv[])
{
    currentPoint.x = 0; currentPoint.y = 0;
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);

    glutInitWindowSize(800, 600);
    glutInitWindowPosition(100, 150);
    glutCreateWindow("OpenGL lesson 3");
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);

    glutMainLoop();

    return 0;
}

void reshape(int w, int h)
{
    glViewport(0, 0, w, h);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0, 100, 0, 100);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void display()
{
    glClearColor(1, 1, 1, 0);
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0, 0, 0);
    readFromFile();
    glColor3d(1, 0, 0);

    for (int i = 0; i < code.size(); i++)
        if (code[i] == 0)
        {
            moveto(point[i]);
        }
        else
        {

```

```

        lineto(point[i]);
    }
    glFlush();
}

void readFromFile()
{
    fstream f("points.txt", ios::in);
    int pointNumber;
    int x, y; Point p;
    f >> pointNumber;
    for (int i = 0; i < pointNumber; i++)
    {
        f >> p.x >> p.y;
        point.push_back(p);
    }
    int movesNumber, m;
    f >> movesNumber;
    for (int i = 0; i < movesNumber; i++)
    {
        f >> m; code.push_back(m);
    }
    f.close();
}

void moveto(Point p){
    currentPoint.x = p.x; currentPoint.y = p.y;
}

void lineto(Point p){
    glBegin(GL_LINES);
    glVertex2i(currentPoint.x, currentPoint.y);
    glVertex2i(p.x, p.y);
    glEnd();
    currentPoint.x = p.x; currentPoint.y = p.y;
}

```

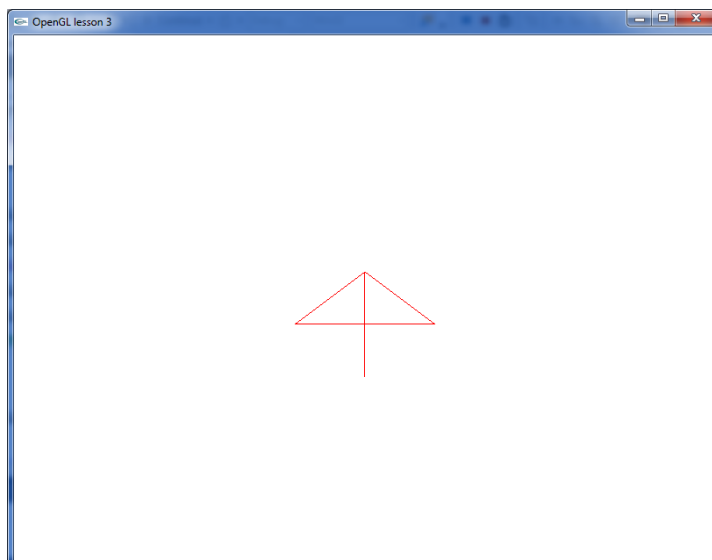
- 4) В папке проекта создайте текстовый файл, назовите его «**points.txt**» и запишите в него координаты объекта следующим образом:

```

5
50 35
50 55
40 45
60 45
50 55
5
0
1
1
1
1
1

```

- 5) Запустите разработанное приложение. Вы должны увидеть следующую картину:



- 6) Поэкспериментируйте с параметрами функции *gluOrtho2D*
- 7) Измените файл с данными таким образом, что бы объект был задан экономным способом. Измените исходный код программы таким образом, что бы он реализовывал экономный способ задания и прорисовки объекта.

Задание 3.

- 1) В решении LR1 в среде Visual Studio создайте новый консольный проект на языке C++ Win32 Console Application. Назовите его LR1_3.
- 2) Напишите программу, которая отображает на экране Ваши инициалы (первые буквы Вашего имени и фамилии). Объекты должны быть заданы экономным способом. Координаты и коды должны храниться в файлах.

Примечание. Координаты букв пригодятся Вам для выполнения следующих практических задания

- 3) Пример прорисовки буквы «П»:

