Московский авиационный институт
(Национальный исследовательский университет)
Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

Лабораторная работа № 2

по курсу «Компьютерная графика»

Тема: Каркасная визуализация выпуклого многогранника. Удаление невидимых линий.

Студент: Сорокин Н.Э.
Группа: М8О-303Б-20
Преподаватель: Филиппов Г.С.
Оценка:

Москва, 2022

1. Постановка задачи

Каркасная визуализация выпуклого многогранника. Удаление невидимых линий.

Вариант 4: Клин

2. Реализация

```
double** Projections::Perspective_Points(double** points, double distance)
{
    double** perspective_points = new double*[6];
    for(int i = 0; i < 6; i++) {
       perspective_points[i] = new double[3];
       perspective_points[i][2] = points[i][2];
       for(int j = 0; j < 2; j++) {
           perspective_points[i][j] = points[i][j] / ((-1 / distance) *
points[i][2] + 1);
       }
    }
    return perspective_points;

}
```

Функция перспективной проекции.

```
int* Operations::roberts_algorithm(double** normals) {
    int* edges_status = new int[9];
    for(int i = 0; i < 9; i++) {
        edges_status[i] = 0;
    }

    double n_0[3] = {0, 0, 100};

    if(dot_product(n_0, normals[0]) > 0) {
        edges_status[3] = edges_status[4] = edges_status[5] = 1;
    }
    if(dot_product(n_0, normals[1]) > 0) {
        edges_status[1] = edges_status[6] = edges_status[7] = 1;
    }
    if(dot_product(n_0, normals[2]) > 0) {
        edges_status[2] = edges_status[5] = edges_status[7] =
edges_status[8] = 1;
    }
    if(dot_product(n_0, normals[3]) > 0) {
        edges_status[0] = edges_status[4] = edges_status[6] =
edges_status[8] = 1;
    }
```

```
    if(dot_product(n_0, normals[4]) > 0) {
        edges_status[0] = edges_status[1] = edges_status[2] =
edges_status[3] = 1;
    }

    return edges_status;
}
```

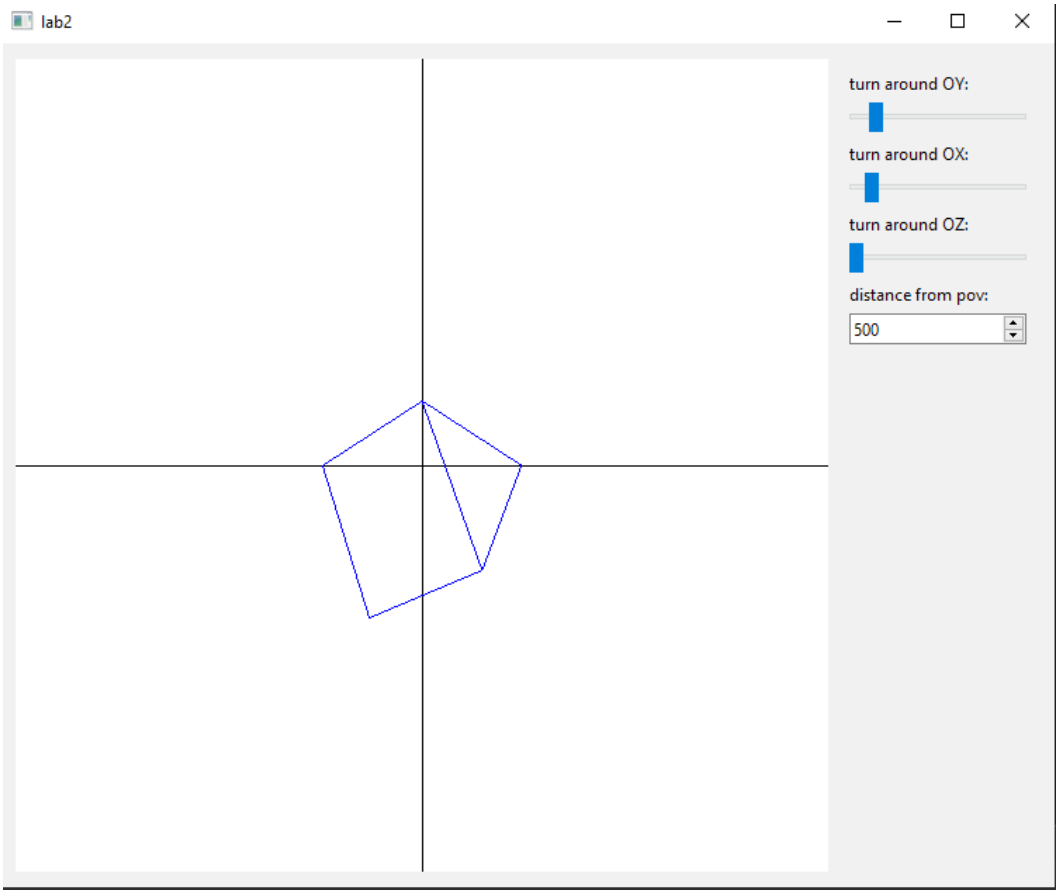
Алгоритм Робертса

```
double** Operations::rotate_OY(double **points, int n, double phi) {
    double** new_points = new double*[n];
    for(int i = 0; i < n; i++) {
        new_points[i] = new double[3];
    }

    for(int i = 0; i < n; i++) {
        new_points[i][0] = points[i][0] * cos(phi) - points[i][2] *
sin(phi);
        new_points[i][1] = points[i][1];
        new_points[i][2] = points[i][0] * sin(phi) + points[i][2] *
cos(phi);
    }

    return new_points;
}
```

3. Вывод программы



4. Вывод

В ходе данной лабораторной работы я освоил основы работы с инструментарием, предоставляемым QT для реализации объёмных фигур и обработки сигналов пользователя с последующей их интерпретацией и реализацией.