

Московский авиационный институт  
(национальный исследовательский университет)

Факультет информационных технологий и прикладной  
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №4-5 по курсу «Компьютерная графика»

Студент: П. А. Мохляков  
Преподаватель: Г. С. Филиппов  
Группа: М8О-308Б-19  
Дата:  
Оценка:  
Подпись:

Москва, 2021

# Лабораторная работа №4-5

## Ознакомление с технологией OpenGL.

**Задача:** Создать графическое приложение с использованием OpenGL. Используя результаты Л.Р.№3, изобразить заданное тело (то же, что и в л.р. №3) с использованием средств OpenGL 2.1. Использовать буфер вершин. Точность аппроксимации тела задается пользователем. Обеспечить возможность вращения и масштабирования многогранника и удаление невидимых линий и поверхностей. Реализовать простую модель освещения на GLSL. Параметры освещения и отражающие свойства материала задаются пользователем в диалоговом режиме.

### 1 Исходный код

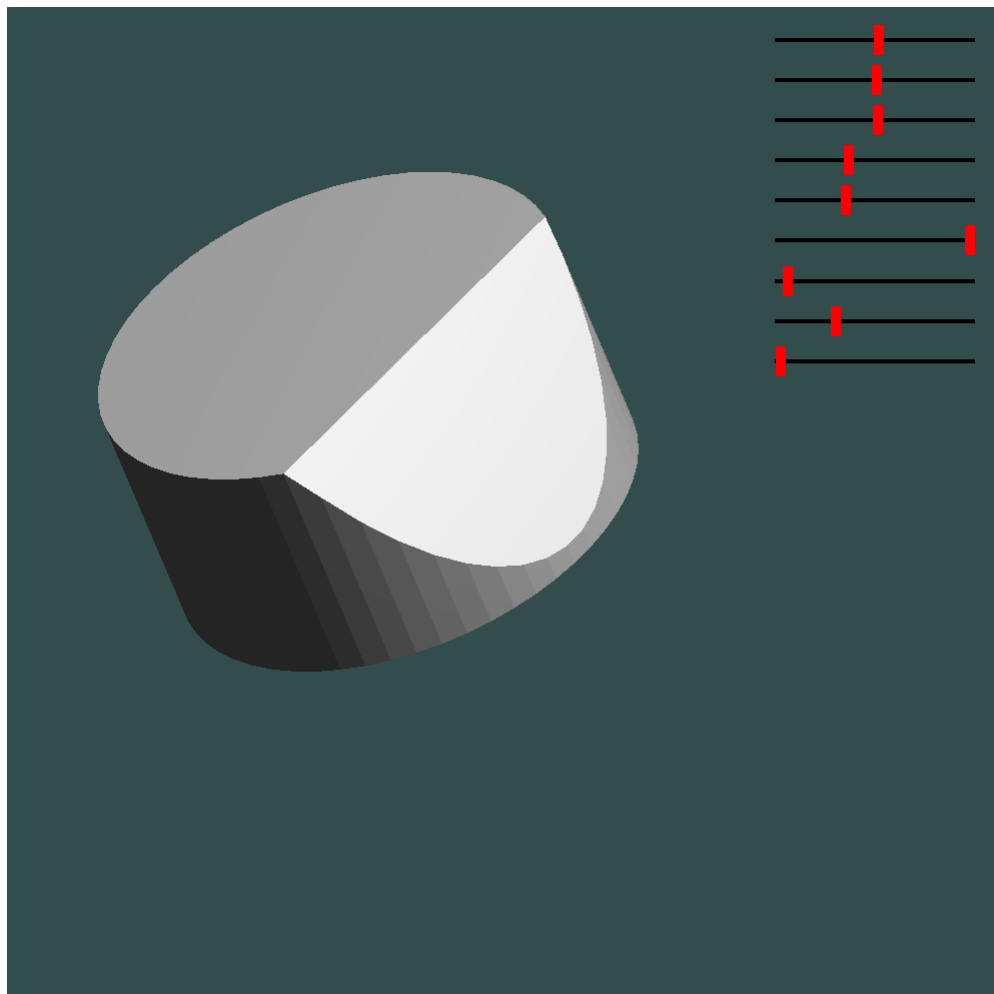
```
1 void get_object(std::vector<std::vector<float>> &points, std::vector<std::vector<int>>
  &faces, float a, float b, float h, int n, float A, float B, float D){
2   points.clear();
3   faces.clear();
4   float m = 0;
5   float t = 0;
6   for(int i = 0; i < n; ++i){
7       float x = a*b*glm::cos(t)/glm::sqrt(glm::pow(b,2)*glm::pow(cos(t),2) + glm::pow
        (a,2)*glm::pow(glm::sin(t),2));
8       float y = a*b*glm::sin(t)/glm::sqrt(glm::pow(b,2)*glm::pow(cos(t),2) + glm::pow
        (a,2)*glm::pow(glm::sin(t),2));
9       std::vector<float> point = {x,0.0f,y};
10      points.push_back(point);
11      if(A*x + B*y > m){
12          m = A*x + B*y;
13      }
14      t += 2*3.1415926f/ n;
15  }
16  D += m;
17  std::vector<int> top;
18  std::vector<int> cut;
19  std::vector<int> back(n);
20  for(int i = 0; i < n; ++i){
21      float x = a*b*glm::cos(t)/glm::sqrt(glm::pow(b,2)*glm::pow(cos(t),2) + glm::pow
        (a,2)*glm::pow(glm::sin(t),2));
22      float y = a*b*glm::sin(t)/glm::sqrt(glm::pow(b,2)*glm::pow(cos(t),2) + glm::pow
        (a,2)*glm::pow(glm::sin(t),2));
23      if(h <= D - A*x - B*y){
24          std::vector<float> point = {x,h,y};
25          points.push_back(point);
26          top.push_back(i + n);
```

```

27         }else{
28             std::vector<float> point = {x,D - A*x - B*y,y};
29             points.push_back(point);
30             cut.push_back(i + n);
31         }
32         t += 2*3.1415926f/ n;
33     }
34     for(int i = 0; i < n; ++i){
35         back[i] = n - i - 1;
36     }
37     faces.push_back(back);
38     for(int i = 0; i < n; ++i){
39         std::vector<int> face = {i,(i+1)%n,(i+1)%n + n,i + n};
40         faces.push_back(face);
41     }
42     if(top.size() >= 3){
43         faces.push_back(top);
44     }
45     if(cut.size() >= 3){
46         int s = cut.size();
47         for(int i = 0; i < s; ++i){
48             int j = (i + 1) % cut.size();
49             if(glm::abs(cut[j] - cut[i]) > 1 && glm::abs(cut[j] - cut[i]) < n){
50                 int first = (cut[i] + 1) % n + n;
51                 int second = (cut[j] - 1) % n + n;
52                 cut.insert(cut.begin() + i+1,first);
53                 cut.insert(cut.begin() + i+2,second);
54             }
55         }
56         faces.push_back(cut);
57     }
58 }
59 }

```

## 2 Скриншоты программы



## 3 Выводы

Выполнив данную лабораторную работу я изучил построение фигур с помощью инструментов OpenGL Core mode. Так же я познакомился с базовым GLSL и реализовал на нем базовое освещение