

**Московский Авиационный Институт
(национальный исследовательский университет)
Факультет прикладной математики**

**Компьютерная графика
Отчет по лабораторной работе №4-5**

Бондарева Елена
Группа М8О-305Б-21

Преподаватель: Симкин О.В.

Москва, 2023

1. **Тема:** ознакомление с технологией OpenGL.
2. **Цель работы:** создать графическое приложение с использованием OpenGL. изобразить заданное тело (прямой усеченный круговой конус) 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
3. **Задание** (*вариант №11*) прямой усеченный круговой конус
4. **Идея, метод, алгоритм** решение задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Программа:

```
import pygame
from pygame.locals import*
from OpenGL.GL import*
from OpenGL.GLU import*
import math

def draw_frustum(bottom_radius, top_radius, height, sides):
    angle_step=360.0/sides

    glBegin(GL_TRIANGLE_STRIP)

    for i in range(sides + 1):
        angle=i*angle_step

        x1=bottom_radius*math.cos(math.radians(angle))
        y1=bottom_radius*math.sin(math.radians(angle))

        x2=top_radius*math.cos(math.radians(angle))
        y2=top_radius*math.sin(math.radians(angle))

        glVertex3f(x1, y1, -height / 2.0)
        glVertex3f(x2, y2, height / 2.0)

    glEnd()

    glBegin(GL_TRIANGLE_FAN)
    glVertex3f(0, 0, -height / 2.0)

    for i in range(sides + 1):
        angle = i * angle_step

        x = bottom_radius*math.cos(math.radians(angle))
        y = bottom_radius*math.sin(math.radians(angle))

        glVertex3f(x, y, -height / 2.0)
```

```

glEnd()

glBegin(GL_TRIANGLE_FAN)
glVertex3f(0, 0, height / 2.0)

for i in range(sides+1):
    angle=i*angle_step
    x=top_radius*math.cos(math.radians(angle))
    y=top_radius*math.sin(math.radians(angle))
    glVertex3f(x, y, height / 2.0)
glEnd()

def main():
    pygame.init()
    display=(1280,720)
    pygame.display.set_mode(display, DOUBLEBUF|OPENGL)

    gluPerspective(60, (display[0]/display[1]), 0.1, 50.0)
    glTranslatef(0.0, 0.0, -5)
    glEnable(GL_LIGHTING)
    glEnable(GL_LIGHT0)

    glEnable(GL_LIGHTING)
    glEnable(GL_DEPTH_TEST)

    glLightfv(GL_LIGHT0, GL_POSITION, [0,1,0,3])
    glLightfv(GL_LIGHT0, GL_DIFFUSE, [1.0,1.0,1.0,1.0])
    glEnable(GL_LIGHT0)

    glMaterial(GL_FRONT, GL_DIFFUSE, [1.0, 0.0, 0.0, 1.0])

    while True:
        for event in pygame.event.get():
            if event.type==pygame.QUIT:
                pygame.quit()
                quit()
        glRotatef(1, 3, 1, 1)
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

        def draw_scaled_frustum(scale_x, scale_y, scale_z):
            glPushMatrix()
            glScalef(scale_x, scale_y, scale_z)
            draw_frustum(1, 0.5, 2, 32)
            glPopMatrix()
            pygame.display.flip()
            pygame.time.wait(10)

        if __name__ == "__main__":
            main()

```

