

**Московский авиационный институт
(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»
Кафедра: 806 «Вычислительная математика и программирование»

**Лабораторная работа № 3
по курсу «Компьютерная графика»**

Студент:	Обыденкова Ю. Ю.
Группа:	М8О-308Б-18
Вариант:	2
Преподаватель:	Филиппов Г.С.
Оценка:	
Дата:	

Москва, 2020

Основы построения фотореалистичных изображений

Постановка задачи

Используя результаты ЛР №2, аппроксимировать заданное тело выпуклым многогранником. Точность аппроксимации задается пользователем. Обеспечить возможность вращения и масштабирования многогранника и удаление невидимых линий и поверхностей. Реализовать простую модель закраски для случая одного источника света. Параметры освещения и отражающие свойства материала задаются пользователем в диалоговом режиме.

Вариант задания: 2. Прямой эллиптический цилиндр

Общие сведения о программе

Язык программирования: Python

Библиотеки: numpy, vpython.

vpy.quad - боковые грани задаём прямоугольниками,

vpy.triangle - нижнюю и верхнюю грани задаём треугольниками (т.к. нельзя задавать полигонами)

scene.lights = [] - обнуляем стандартное освещение, после чего задаем лампе свои параметры.

scene.width и scene.height - задаём размеры нашей картинки

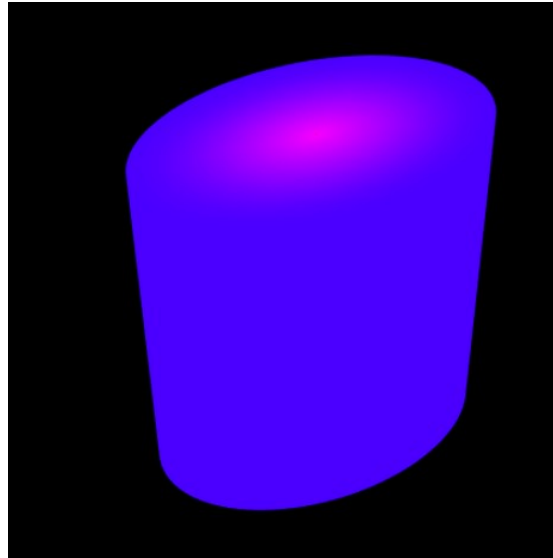
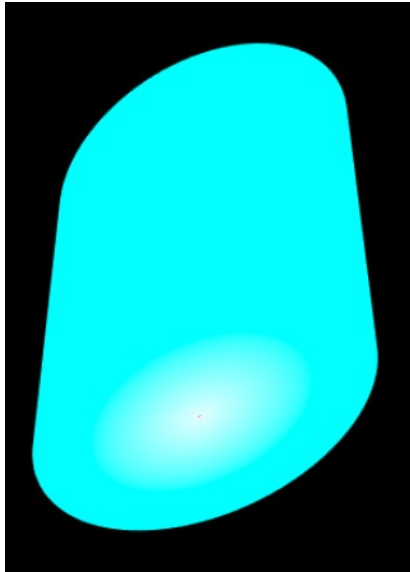
Руководство по использованию программы

~:\$ python3 "3.py"

Введите точность аппроксимации: 100

Введите цвет: 0 0 255

Пользователю предлагается задать параметры: количества вершин и цвета материала. Вращать фигуру можно, зажав правую кнопку мыши.



Код программы

```
import numpy as np
import vpython as vpy
n = int(input("Введите точность аппроксимации: "))
c1, c2, c3 = (float(x) for x in input("Введите цвет: ").split())
color = vpy.vec(c1, c2, c3)
h = 3.0
a = 1.5
b = 1.0
t = np.linspace(0, 2 * np.pi, n + 1)
x = a * np.cos(t)
y = b * np.sin(t)
# Задаем координаты вершин
# 2 * n вершин на контуры эллипсов и две центральные
vertexes = np.empty(shape=(2 * n + 2), dtype=object)
vertexes[2 * n + 1] = vpy.vertex(pos=vpy.vec(0, h, 0), color=vpy.color.red)
vertexes[2 * n] = vpy.vertex(pos=vpy.vec(0, 0, 0), color=vpy.color.red)
for i in range(n):
    vertexes[i] = vpy.vertex(pos=vpy.vec(x[i], 0, y[i]), color=color)
    vertexes[i + n] = vpy.vertex(pos=vpy.vec(x[i], h, y[i]), color=color)
```

```

# Рисуем основания
# С помощью треугольников
for i in range(n):
    vpy.triangle(
        v0=vertexes[i],
        v2=vertexes[(i + 1) % n],
        v1=vertexes[2 * n]
    )
    vpy.triangle(
        v0=vertexes[i + n],
        v2=vertexes[(i + 1) % n + n],
        v1=vertexes[2 * n + 1]
    )
# Рисуем боковые грани
# С помощью 4-х угольников
for i in range(n):
    vpy.quad(
        v3=vertexes[i],
        v2=vertexes[(i + 1) % n],
        v1=vertexes[(i + 1) % n + n],
        v0=vertexes[i + n]
    )
# vpy.scene.lights = []
# lamp = vpy.distant_light(direction=vpy.vec(0, 0, 10), color=vpy.vec(0.3, 0.0,
0.2))
# vpy.scene.ambient = vpy.vec(0.1, 0.1, 0.1)
vpy.scene.width = 2400
vpy.scene.height = 1200

```

Вывод

Благодаря проделанной работе, я освоила новую для себя библиотеку vpython.