

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: Ф. А. Иванов
Преподаватель: Г. С. Филиппов
Группа: М8О-308Б-19
Дата: 21.12.2021
Оценка:
Подпись:

Москва, 2021

Лабораторная работа №4-5

Тема: : 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.

Вариант 3: Шар.

1 Описание

Шаг 1.

АППРОКСИМАЦИЯ, АППРОКСИМИРОВАНИЕ [лат. *approximare* приближаться] - математическое приближенное выражение каких-либо величин (или геометрических объектов) через другие, более простые величины.

Передо мной стоит задача - изучить и освоить работу библиотеки *OpenGL*. Для построения сферы и ее аппроксимации буду использовать инструмент *OpenGL.GLUT*, у него есть встроенная функция построения сферы и задания ее точности аппроксимации. Так же библиотека позволяет реализовать освещение объекта. Источник света расположу в правом верхнем углу позади смотрящего на объект. Интенсивность будет регулировать пользователь.

Шаг 2.

В качестве визуального взаимодействия с пользователем использую библиотеку *PySimpleGUI*.

2 Исходный код

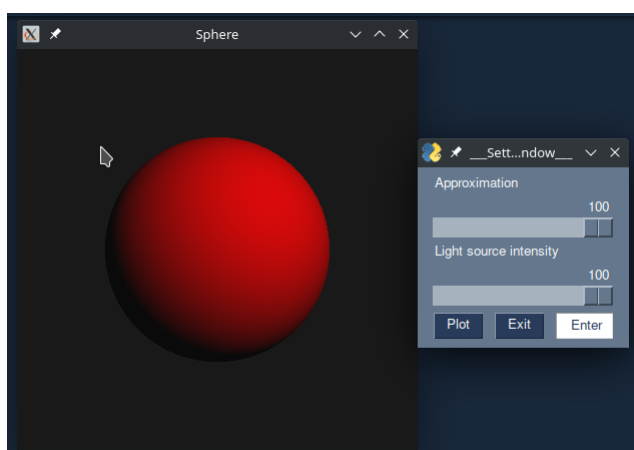
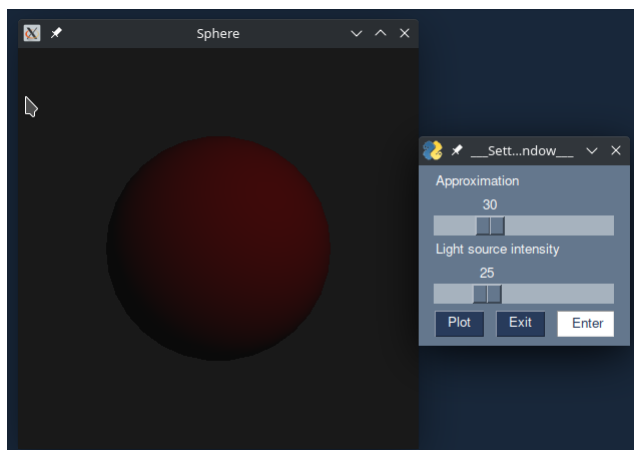
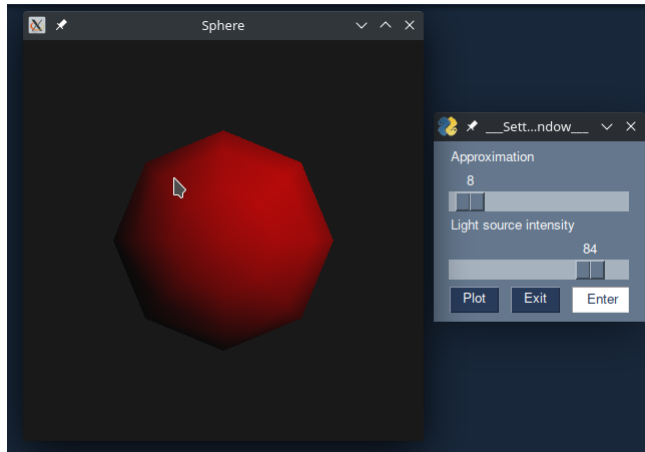
```
1 from OpenGL.GLUT import *
2 from OpenGL.GLU import *
3 from OpenGL.GL import *
4 import sys
5 import PySimpleGUI as sg
6
7 name = 'Sphere'
8 a = 50
9 intensive = 0.5
10
11 def draw():
12     glutInit(sys.argv)
13     #GLUT_DOUBLE -
14     #GLUT_RGB - rgb
15     #GLUT_DEPTH -
16     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH)
17     glutInitWindowSize(400,400)
18     glutCreateWindow(name)
19
20     #
21     glClearColor(.1,.1,.1,1.)
22     #
23     glShadeModel(GL_SMOOTH)
24
25     #
26     glEnable(GL_CULL_FACE)
27     glEnable(GL_DEPTH_TEST)
28     glEnable(GL_LIGHTING)
29     lightZeroPosition = [10.,10.,10.,1.] #
30     lightZeroColor = [intensive,1.0,1.0,1.0] #
31     #
32     glLightfv(GL_LIGHT0, GL_POSITION, lightZeroPosition)
33     glLightfv(GL_LIGHT0, GL_DIFFUSE, lightZeroColor)
34     glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, 0.1)
35     glLightf(GL_LIGHT0, GL_LINEAR_ATTENUATION, 0.05)
36
37     #
38     glEnable(GL_LIGHT0)
39
40     #
41     glutDisplayFunc(display)
42     #
43     glMatrixMode(GL_PROJECTION)
44     gluPerspective(40.,1.,1.,40.)
45     #
46     glMatrixMode(GL_MODELVIEW) #
47     gluLookAt(0,0,10,
```

```

48         0,0,0,
49         0,1,0)
50     glPushMatrix()
51     glutMainLoop()
52
53
54 def display():
55     glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
56     glPushMatrix()
57     color = [1.0,0.,0.,1.]
58     glMaterialfv(GL_FRONT, GL_DIFFUSE, color)
59     #  (,      0Z      0Z)
60     glutSolidSphere(2,a,a)
61     glPopMatrix()
62     glutSwapBuffers()
63
64 layout = [
65     [sg.Text('Approximation')],
66     [sg.Slider(orientation='horizontal', key='slider1', range=(4,100))],
67     [sg.Text('Light source intensity')],
68     [sg.Slider(orientation='horizontal', key='slider2', range=(0,100))],
69     [sg.Button('Plot'), sg.Button('Exit'), sg.Button('Enter')]]
70
71 window = sg.Window('___Setting window___', layout)
72
73 while True:
74     event, values = window.read()
75     if event in (sg.WIN_CLOSED, 'Exit'):
76         break
77     elif event == 'Plot':
78         draw()
79     elif event == 'Enter':
80         a = int(values['slider1'])
81         intensive = int(values['slider2'])/100
82         draw()
83 window.close()

```

3 Пример работы программы



4 Выводы

Выполнив 4-ую лабораторную работу по курсу Компьютерной графики, а закрепил такое свойство, как аппроксимация фигур и смог реализовать данный принцип в своем задании. Познакомился с инструментами библиотеки *OpenGL*. С помощью библиотеки смог построить 3D-сферу. Смог реализовать простейшую модель освещения. *OpenGL* - удобная библиотека для построения 3D-графики.

Список литературы

- [1] *Документация по библиотеке PySimpleGUI*
URL: <https://pysimplegui.readthedocs.io/en/latest/> (дата обращения: 19.12.2021).
- [2] *22 материала для изучения OpenGL*
URL: <https://proglib.io/p/learn-opengl> (дата обращения: 19.12.2021).
- [3] *Руководство по Numpy*
URL: <https://numpy.org/doc/stable/reference/> (дата обращения: 19.12.2021).