

Московский авиационный институт  
(национальный исследовательский университет)

Факультет информационных технологий и прикладной  
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №6 по курсу «Компьютерная графика»

Студент: Ф. А. Иванов  
Преподаватель: Г. С. Филиппов  
Группа: М8О-308Б-19  
Дата: 24.12.2021  
Оценка:  
Подпись:

Москва, 2021

## Лабораторная работа №6

**Тема:** : Создание шейдерных анимационных эффектов в OpenGL 2.1.

**Задача:** Для поверхности, созданной в л.р. No5, обеспечить выполнение следующего шейдерного эффекта:

**Вариант 3:** Зеркальное освещение от источника света в заданной позиции.

# 1 Описание

## Шаг 1.

Ход решения такой же, как и в лабораторной работе №5. Добавляются только источники освещения. Для того чтоб получить блик, нужно установить 3 источника света с цветом красный, зеленый и синий. Для создания источников света и установки нужных параметров использую функции *glLightfv* и *glLightf*. Для установки положения источка используется параметр *GL\_POSITION* в функции *glLightfv*.

## 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 move = 0
10 intensive = 0.5
11 X = 30
12 Y = 30
13 Z = 30
14
15 def key_event(key):
16     if key == 27:
17         exit(0)
18
19 def draw():
20     glutInit(sys.argv)
21     #GLUT_DOUBLE -
22     #GLUT_RGB - rgb
23     #GLUT_DEPTH -
24     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH)
25     glutInitWindowSize(400,400)
26     glutCreateWindow(name)
27
28     #
29     glClearColor(.1,.1,.1,1.)
30     #
31     glShadeModel(GL_SMOOTH)
32
33     #
34     glEnable(GL_CULL_FACE)
35     glEnable(GL_DEPTH_TEST)
36     glEnable(GL_LIGHTING)
37     lightZeroPosition = [X,Y,Z,1.] #
38     lightZeroColor1 = [intensive,1.0,1.0,1.0] #
39     lightZeroColor2 = [intensive,1.0,1.0,1.0]
40     lightZeroColor3 = [intensive,1.0,1.0,1.0]
41     #
42     glLightfv(GL_LIGHT0, GL_POSITION, lightZeroPosition)
43     glLightfv(GL_LIGHT0, GL_SPECULAR, lightZeroColor1)
44     glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, 0.1)
45     glLightf(GL_LIGHT0, GL_LINEAR_ATTENUATION, 0.05)
46
47     glLightfv(GL_LIGHT1, GL_POSITION, lightZeroPosition)
```

```

48     glLightfv(GL_LIGHT1, GL_DIFFUSE, lightZeroColor2)
49     glLightf(GL_LIGHT1, GL_CONSTANT_ATTENUATION, 0.1)
50     glLightf(GL_LIGHT1, GL_LINEAR_ATTENUATION, 0.05)
51
52     glLightfv(GL_LIGHT2, GL_POSITION, lightZeroPosition)
53     glLightfv(GL_LIGHT2, GL_DIFFUSE, lightZeroColor3)
54     glLightf(GL_LIGHT2, GL_CONSTANT_ATTENUATION, 0.1)
55     glLightf(GL_LIGHT2, GL_LINEAR_ATTENUATION, 0.05)
56
57     #
58     glEnable(GL_LIGHT0)
59     glEnable(GL_LIGHT1)
60     glEnable(GL_LIGHT2)
61
62     #
63     glutDisplayFunc(display)
64     # ,
65     glMatrixMode(GL_PROJECTION)
66     gluPerspective(40.,1.,1.,40.)
67     # , ,
68     glMatrixMode(GL_MODELVIEW) #
69     gluLookAt(0,0,10,
70               0,0,0,
71               0,1,0)
72     glPushMatrix()
73     glutMainLoop()
74
75
76 def display():
77     glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
78     glPushMatrix()
79     #color = [0., 0., 0., 1.]
80     color = [50.0]
81     glMaterialfv(GL_FRONT, GL_SHININESS, color)
82     # (, 0Z 0Z)
83     #glutWireSphere(2,a,a)
84     glutSolidSphere(2,a,a)
85     glPopMatrix()
86     glutSwapBuffers()
87
88 layout = [
89     [sg.Text('Approximation')],
90     [sg.Slider(orientation='horizontal', key='slider1', range=(4,100))],
91     [sg.Text('Light source intensity')],
92     [sg.Slider(orientation='horizontal', key='slider2', range=(0,100))],
93     [sg.Text('Camera X:')],
94     [sg.Slider(orientation='horizontal', key='slider3', range=(-100,100))],
95     [sg.Text('Camera Y:')],
96     [sg.Slider(orientation='horizontal', key='slider4', range=(-100,100))],

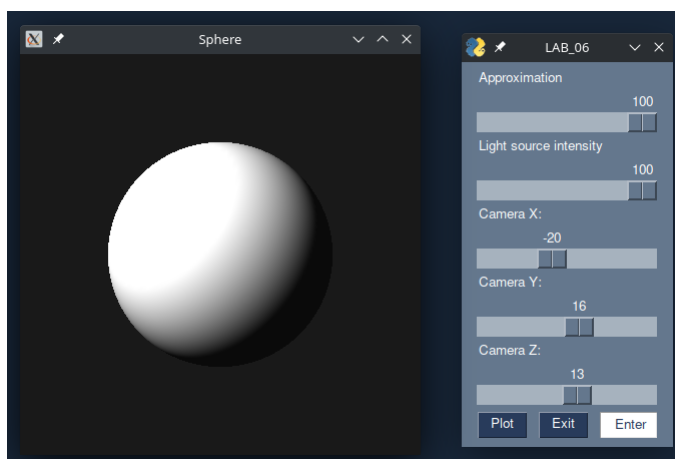
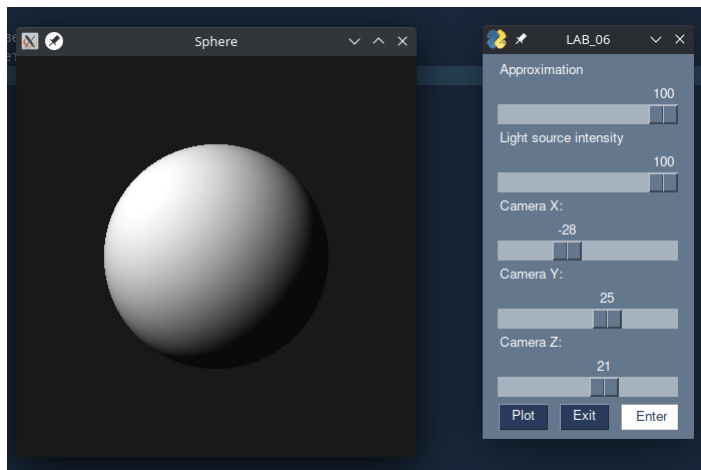
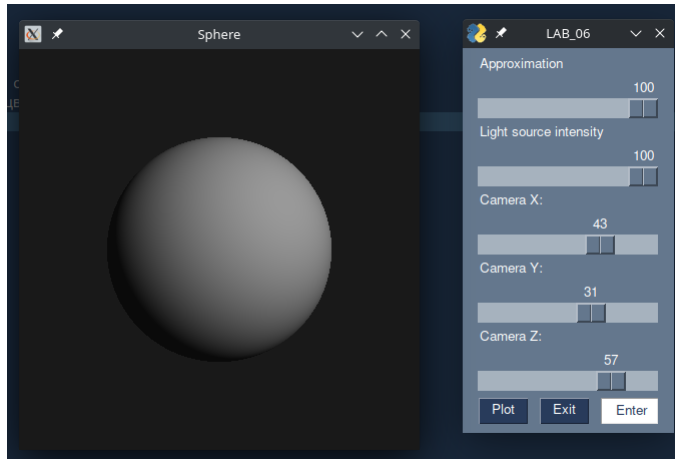
```

```

97     [sg.Text('Camera Z:'),
98     [sg.Slider(orientation='horizontal', key='slider5', range=(-100,100))],
99     [sg.Button('Plot'), sg.Button('Exit'), sg.Button('Enter')]]
100
101 window = sg.Window('LAB_06', layout)
102
103 while True:
104     event, values = window.read()
105     if event in (sg.WIN_CLOSED, 'Exit'):
106         break
107     elif event == 'Plot':
108         draw()
109     elif event == 'Enter':
110         a = int(values['slider1'])
111         intensive = int(values['slider2'])/100
112         X = int(values['slider3'])
113         Y = int(values['slider4'])
114         Z = int(values['slider5'])
115         draw()
116 window.close()

```

### 3 Пример работы программы



## 4 Выводы

Выполнив 5-ую лабораторную работу по курсу Компьютерной графики, а закрепил некоторые инструменты библиотеки *OpenGL*. С помощью библиотеки смог построить 3D-сферу. Смог реализовать зеркальную модель освещения. Научился управлять положением источника цвета.



## 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).