Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Московский государственный технический университет имени Н.Э.Баумана (национальный исследовательский университет)»

Джабаров Р.А. ИУ9-32Б

**Компьютерная графика.**

**Лабораторная работа №2**

Москва

2023

**Код:**

import math

from OpenGL.GL import \*

from OpenGL.GLU import \*

from OpenGL.GLUT import \*

import numpy as np

object\_matrix = np.identity(4)

last\_mouse\_x = None

last\_mouse\_y = None

def handle\_mouse\_down(button, state, x, y):

    global mouse\_down, last\_mouse\_x, last\_mouse\_y

    if button == GLUT\_LEFT\_BUTTON and state == GLUT\_DOWN:

        last\_mouse\_x = x

        last\_mouse\_y = y

def handle\_mouse\_move(x, y):

    global object\_matrix, last\_mouse\_x, last\_mouse\_y

    delta\_x = x - last\_mouse\_x

    delta\_y = y - last\_mouse\_y

    rotation\_y = np.identity(4)

    rotation\_x = np.identity(4)

    rotation\_y[0, 0] = np.cos(delta\_x / 100.0)

    rotation\_y[2, 0] = -np.sin(delta\_x / 100.0)

    rotation\_y[0, 2] = np.sin(delta\_x / 100.0)

    rotation\_y[2, 2] = np.cos(delta\_x / 100.0)

    rotation\_x[1, 1] = np.cos(delta\_y / 100.0)

    rotation\_x[2, 1] = np.sin(delta\_y / 100.0)

    rotation\_x[1, 2] = -np.sin(delta\_y / 100.0)

    rotation\_x[2, 2] = np.cos(delta\_y / 100.0)

    object\_matrix = np.dot(rotation\_x, object\_matrix)

    object\_matrix = np.dot(rotation\_y, object\_matrix)

    last\_mouse\_x = x

    last\_mouse\_y = y

scale = 1.0

def mouse\_wheel\_callback(button, direction, x, y):

    global scale

    if direction > 0:

        scale += 0.1

    else:

        scale -= 0.1

    glutPostRedisplay()

def draw\_scene():

    global object\_matrix, scale

    glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT)

    glMatrixMode(GL\_MODELVIEW)

    glLoadIdentity()

    glPushMatrix()

    glMultMatrixf(object\_matrix)

    glScalef(scale\*0.25, scale\*0.25, scale\*0.25)

    glBegin(GL\_QUADS)

    glColor3f(0.0, 0.0, 1.0);

    glVertex3f(1, -1, 1)

    glVertex3f(1, 1, 1)

    glVertex3f(-1, 1, 1)

    glVertex3f(-1, -1, 1)

    glColor3f(0.0, 1.0, 0.0)

    glVertex3f(-1, -1, 1)

    glVertex3f(-1, 1, 1)

    glVertex3f(-1, 1, -1)

    glVertex3f(-1, -1, -1)

    glColor3f(0.0, 1.0, 1.0)

    glVertex3f(-1, -1, -1)

    glVertex3f(-1, 1, -1)

    glVertex3f(1, 1, -1)

    glVertex3f(1, -1, -1)

    glColor3f(1.0, 0.0, 0.0)

    glVertex3f(1, -1, -1)

    glVertex3f(1, 1, -1)

    glVertex3f(1, 1, 1)

    glVertex3f(1, -1, 1)

    glColor3f(1.0, 0.0, 1.0)

    glVertex3f(1, 1, 1)

    glVertex3f(1, 1, -1)

    glVertex3f(-1, 1, -1)

    glVertex3f(-1, 1, 1)

    glColor3f(1.0, 1.0, 0.0)

    glVertex3f(1, -1, 1)

    glVertex3f(-1, -1, 1)

    glVertex3f(-1, -1, -1)

    glVertex3f(1, -1, -1)

    glEnd()

    glPopMatrix()

    glutSwapBuffers()

# глобальная переменная-флаг

wireframe\_mode = False

def handle\_key\_press(key, x, y):

    global wireframe\_mode

    if key == b'q':

        wireframe\_mode = not wireframe\_mode

        if wireframe\_mode:

            glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE)

            glDisable(GL\_CULL\_FACE)

        else:

            glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL)

            glEnable(GL\_CULL\_FACE)

    fl = 0

    if key == b'e':

        glMatrixMode(GL\_PROJECTION)

        glPushMatrix()

        glLoadIdentity()

        matrix = [1, 0, 0, 0.1, 0, 1, 0, 0.1, 0, 0, 0, -0.1, 0, 0, 0, 1]

        glPopMatrix()

        glMultMatrixf(matrix)

    glutPostRedisplay()

def reshape(width, height):

   aspect = float(width) / float(height)

   glViewport(0, 0, width, height)

   glMatrixMode(GL\_PROJECTION)

   glLoadIdentity()

   if aspect > 1:

      glOrtho(-aspect, aspect, -1.0, 1.0, -1.0, 1.0)

   else:

      glOrtho(-1.0, 1.0, -1.0/aspect, 1.0/aspect, -1.0, 1.0)

   glMatrixMode(GL\_MODELVIEW)

   glLoadIdentity()

glutInit()

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB | GLUT\_DEPTH)

glutInitWindowSize(600, 600)

glutCreateWindow(b"PyOpenGL Example")

glutMouseWheelFunc(mouse\_wheel\_callback)

glutReshapeFunc(lambda w, h: glViewport(0, 0, w, h))

glutMouseFunc(handle\_mouse\_down)

glutMotionFunc(handle\_mouse\_move)

glutKeyboardFunc(handle\_key\_press)

glutReshapeFunc(reshape)

glEnable(GL\_CULL\_FACE)

glutDisplayFunc(draw\_scene)

glutMainLoop()

**Задача:**

Данный код отрисовывает куб, который может быть вращен мышью, масштабирован колесом мыши и переключаться в режим "только контуры" клавишей "q". Также существует возможность включения трехточечной перспективы с помощью клавиши "e".

**Основная теория:**

Для отрисовки куба используется библиотека OpenGL. С помощью glMatrixMode задается матрица текущей проекции (в данном случае это модельная матрица), которая будет использована для последующих преобразований (вращение, масштабирование, перемещение).

**Практическая реализация:**

Переменная object\_matrix инициализируется единичной матрицей. Эта переменная используется для вращения и масштабирования объекта. Обработчик событий handle\_mouse\_down запоминает координаты мыши при нажатии левой кнопки. Обработчик handle\_mouse\_move определяет величину смещения мыши, вычисляет две матрицы вращения для изменения угла вокруг осей x и y, перемножает их и умножает на object\_matrix. Колесо мыши обрабатывается в функции mouse\_wheel\_callback, которая изменяет переменную scale для масштабирования куба. Функция draw\_scene отрисовывает сам куб и вызывается при каждой перерисовке окна. Обработчик handle\_key\_press переключает режим отображения куба (обычный или только контуры) и перемещает куб вдоль осей.

**Заключение:**

Данный код представляет собой простой пример использования OpenGL для отрисовки объектов с возможностью вращения и масштабирования с помощью мыши и перемещения вдоль осей с помощью клавиши.