

**Міністерство освіти і науки України  
Національний технічний університет України «КПІ» імені Ігоря Сікорського  
Кафедра обчислювальної техніки ФІОТ**

**ЗВІТ  
з лабораторної роботи №9  
з навчальної дисципліни «Computer Vision»**

**Тема:**

**ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ НА  
ЦИФРОВИХ ЗОБРАЖЕННЯХ ДЛЯ ЗАДАЧ COMPUTER VISION**

**Виконала:**

Студентка 3 курсу  
Навчальної групи ІС-12  
Мельникова К.О.

**Перевірив:**

Професор кафедри ОТ ФІОТ  
Писарчук О.О.

**Київ 2024**

## Мета роботи:

дослідити методологію і технології створення доповненої реальності.

## Завдання I рівня складності – максимально 7 балів.

З використанням методів бібліотеки OpenGL розробити скрипт, що реалізує реалістичну візуалізацію графічної сцени у композиції та відповідно до взаємовідносин об'єктів сцени:

в графічному вікні розташовано 3D багатокутник в аксонометричній проекції та 3D модель поверхні другого порядку – типи фігур та їх кількість, розмір, взаємне розташування обрати самостійно;

обрані фігури освітлюються точковим джерелом світла, модель світла, метод зафарбовування поверхонь для моделювання світло і тіні - обирається самостійно;

взаємне розташування «акторів» сцени: спостерігач, геометричні фігури, джерело світла – обрати самостійно;

передбачити анімацію сцени, шляхом обертання геометричних фігур відносно нерухомих спостерігача та джерела світла.

Порядок реалізації завдання відобразити у формі структурної схеми етапів конвеєру.

## Програмна реалізація:

```
import sys
import numpy as np
from OpenGL.GL import *
from OpenGL.GLUT import *
from OpenGL.GLU import *

# Ініціалізація параметрів сцени
angle = 0

def init():
    glClearColor(0.0, 0.0, 0.0, 1.0)
    glEnable(GL_DEPTH_TEST)
    glEnable(GL_LIGHTING)
    glEnable(GL_LIGHT0)
    glEnable(GL_COLOR_MATERIAL)
```

```

    glShadeModel(GL_SMOOTH)

def display():
    global angle
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()

    # Налаштування камери
    gluLookAt(5, 5, 5, 0, 0, 0, 0, 1, 0)

    # Визначення джерела світла
    light_pos = [5, 5, 5, 1]
    glLightfv(GL_LIGHT0, GL_POSITION, light_pos)

    # Обертання об'єктів
    glRotatef(angle, 0, 1, 0)

    # Відображення куба
    glPushMatrix()
    glColor3f(1.0, 0.0, 0.0)
    glutSolidCube(2)
    glPopMatrix()

    # Відображення сфери
    glPushMatrix()
    glColor3f(0.0, 0.0, 1.0)
    glTranslatef(3, 0, 0)
    glutSolidSphere(1, 50, 50)
    glPopMatrix()

    glutSwapBuffers()
    angle += 1

def reshape(w, h):
    if h == 0:
        h = 1
    glViewport(0, 0, w, h)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluPerspective(45, w / h, 0.1, 50.0)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()

def main():
    glutInit(sys.argv)

```

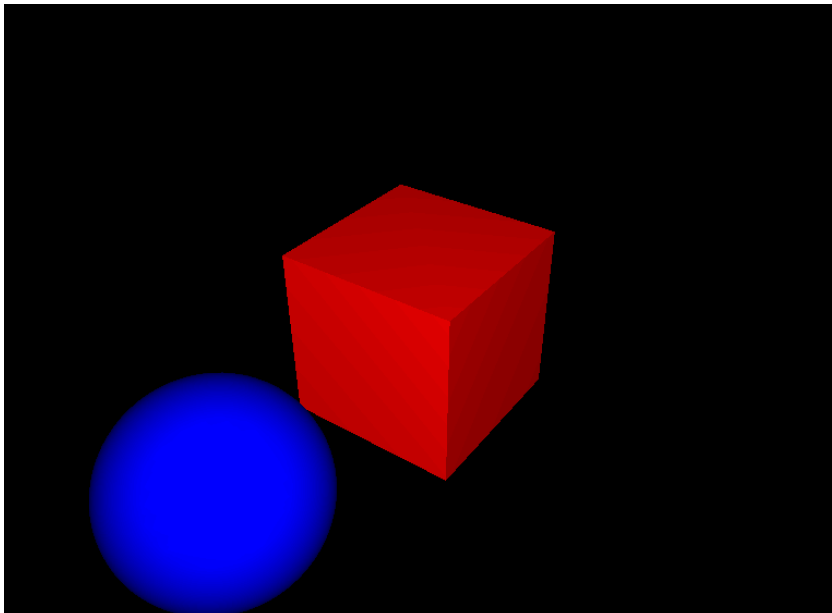
```
glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE | GLUT_DEPTH)
glutInitWindowSize(800, 600)
glutInitWindowPosition(100, 100)
glutCreateWindow(b"OpenGL Scene")
glutDisplayFunc(display)
glutIdleFunc(display)
glutReshapeFunc(reshape)
init()
glutMainLoop()

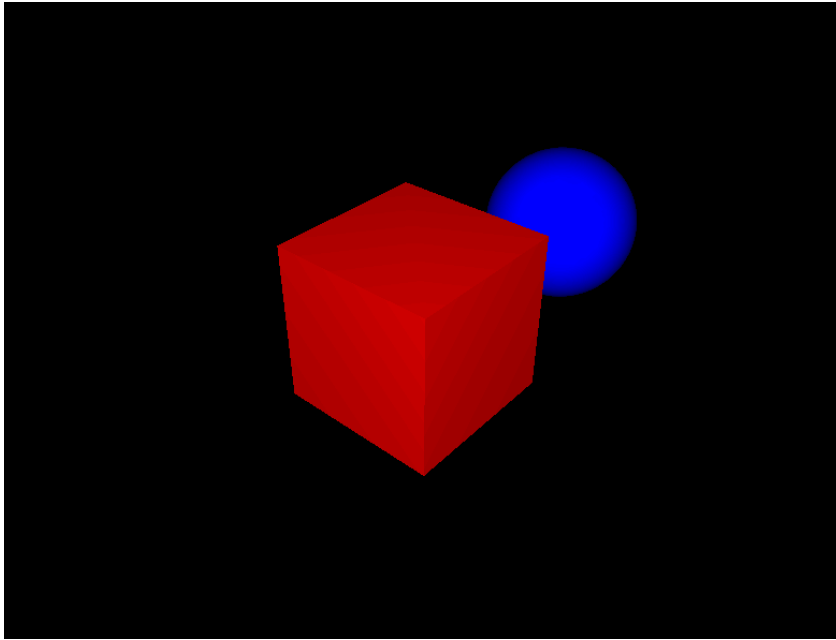
if __name__ == "__main__":
    main()
```

Наведений вище код містить:

- Функцію *init* для ініціалізації параметрів сцени.
- Функцію *display* для відображення об'єктів сцени та обертання.
- Функцію *reshape* для налаштування вікна відображення.
- Головну функцію *main* для запуску програми.

**Результат виконання:**





Можемо спостерігати процес обертання куба та сфери (поверхні другого порядку) навколо осі, що проходить через центр куба.

#### **Висновок:**

У ході виконання даної лабораторної роботи було розроблено програму для реалізації реалістичної візуалізації графічної сцени з використанням бібліотеки OpenGL. Програма дозволяє відображати 3D об'єкти, освітлювати їх точковим джерелом світла та здійснювати анімацію обертання об'єктів. Результати роботи наведеного програмного скрипта в повній мірі задовольняють вимогам поставленого завдання та демонструють основні можливості OpenGL для створення графічних сцен.