Exp.No: 06 Pavipraja.P

Date: 18-09-25 231701037

**Create and Render 3D Objects with Basic Shading**

**Aim:**

To write a program to create and render 3D objects (cube, sphere) using polygons and apply basic color and shading techniques.

**Procedure:**

1. Start the program.
2. Define vertices and faces of 3D objects (cube, sphere).
3. For each face, compute:

* Surface normal (to determine brightness for shading).
* Color intensity using Lambert’s cosine law I = I0 ​ ⋅ cosθ, where θ is angle between face normal and light direction.

1. Apply colors to each face based on computed intensity (flat shading).
2. Display the 3D objects using matplotlib with appropriate color for each face.
3. Stop.

**Code:**

import numpy as np

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d.art3d import Poly3DCollection

cube\_vertices = np.array([

    [0, 0, 0], [1, 0, 0], [1, 1, 0], [0, 1, 0],

    [0, 0, 1], [1, 0, 1], [1, 1, 1], [0, 1, 1]

])

cube\_faces = [[0,1,2,3], [4,5,6,7], [0,1,5,4],

              [2,3,7,6], [0,3,7,4], [1,2,6,5]]

light\_dir = np.array([0, 0, 1])  # light coming from +Z

def compute\_face\_color(vertices, face):

    v0, v1, v2 = vertices[face[0]], vertices[face[1]], vertices[face[2]]

    normal = np.cross(v1 - v0, v2 - v0)

    normal = normal / np.linalg.norm(normal)

    intensity = max(0, np.dot(normal, light\_dir))  # Lambert shading

    return (intensity, intensity, intensity)  # grayscale color

fig = plt.figure()

ax = fig.add\_subplot(111, projection='3d')

for face in cube\_faces:

    face\_vertices = [cube\_vertices[i] for i in face]

    color = compute\_face\_color(cube\_vertices, face)

    ax.add\_collection3d(Poly3DCollection([face\_vertices],

                                         facecolors=[color], edgecolors='k'))

u, v = np.mgrid[0:2\*np.pi:30j, 0:np.pi:15j]

x = np.cos(u) \* np.sin(v) + 2  # shift sphere in +x

y = np.sin(u) \* np.sin(v)

z = np.cos(v)

ax.plot\_surface(x, y, z, color='skyblue', edgecolor='k', alpha=0.8)

ax.set\_xlabel('X')

ax.set\_ylabel('Y')

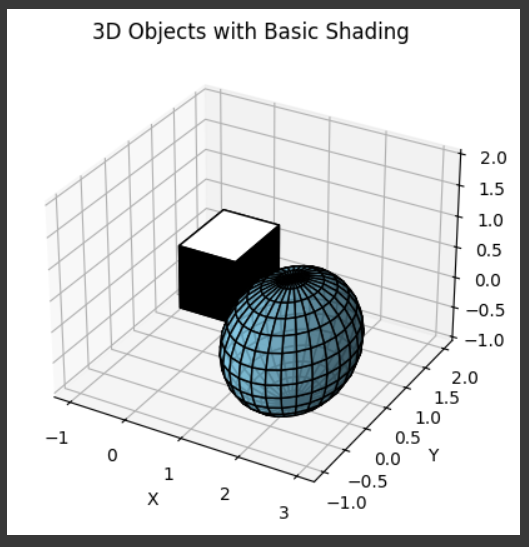
ax.set\_zlabel('Z')

ax.set\_title("3D Objects with Basic Shading")

ax.auto\_scale\_xyz([-1, 3], [-1, 2], [-1, 2])

plt.show()

**Output:**

****

**Result:**

The program successfully renders 3D objects (cube and sphere) using polygons.Basic color and shading are applied, producing a realistic 3D effect.