Experiment No: 8

Experiment Name:

Rotation of a 3D Cube using OpenGL

Objective:

The objective of this experiment is to understand and implement **3D rotation transformation** by drawing and rotating a **three-dimensional cube** using **OpenGL**.

Theory:

A 3D cube is a solid object having six square faces. In computer graphics, 3D objects are represented using **x, y, and z coordinates**.

Rotation in 3D graphics changes the orientation of an object about one or more axes:

- X-axis
- Y-axis
- Z-axis

Software Requirements:

- Code::Blocks / C++
- OpenGL
- GLUT Library

Procedure / Description of Code:

1. Initialize GLUT and create an OpenGL window.
2. Enable **depth testing** to handle 3D visibility.
3. Set up **perspective projection** using `gluPerspective()`.
4. Translate the cube inside the viewing volume.

5. Draw all six faces of the cube using `GL_QUADS`.
6. Assign different colors to each face for clarity.
7. Apply rotation using `glRotatef()` about X, Y, and Z axes.
8. Use a **timer function** to update the rotation angle continuously.
9. Display the rotating cube on the screen.

SOURCE CODE:

```
#include <windows.h>

#include <GL/glut.h>


float angle = 0.0f;

void display()

{

   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

   glLoadIdentity();

   glTranslatef(0.0f, 0.0f, -6.0f);

   // Rotate cube

   glRotatef(angle, 1.0f, 1.0f, 1.0f);

   // Draw cube

   glBegin(GL_QUADS);

   // Front face

   glColor3f(1, 0, 0);

   glVertex3f(-1, -1,  1);
```

```
glVertex3f( 1, -1,  1);
glVertex3f( 1,  1,  1);
glVertex3f(-1,  1,  1);
// Back face
glColor3f(0, 1, 0);
glVertex3f(-1, -1, -1);
glVertex3f(-1,  1, -1);
glVertex3f( 1,  1, -1);
glVertex3f( 1, -1, -1);
// Top face
glColor3f(0, 0, 1);
glVertex3f(-1,  1, -1);
glVertex3f(-1,  1,  1);
glVertex3f( 1,  1,  1);
glVertex3f( 1,  1, -1);
// Bottom face
glColor3f(1, 1, 0);
glVertex3f(-1, -1, -1);
glVertex3f( 1, -1, -1);
glVertex3f( 1, -1,  1);
```

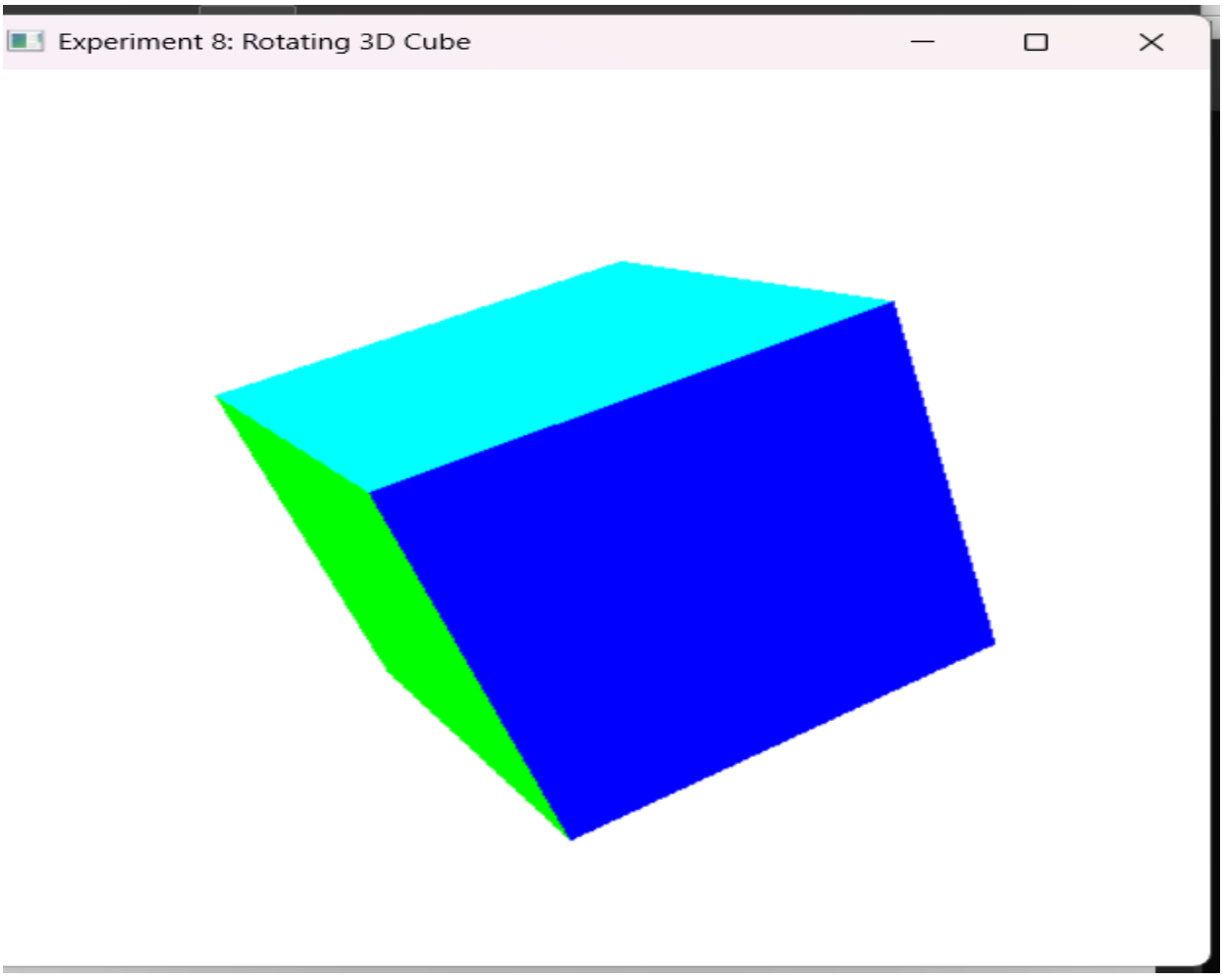
```
    glVertex3f(-1, -1,  1);
    // Right face
    glColor3f(0, 1, 1);
    glVertex3f( 1, -1, -1);
    glVertex3f( 1,  1, -1);
    glVertex3f( 1,  1,  1);
    glVertex3f( 1, -1,  1);
    // Left face
    glColor3f(1, 0, 1);
    glVertex3f(-1, -1, -1);
    glVertex3f(-1, -1,  1);
    glVertex3f(-1,  1,  1);
    glVertex3f(-1,  1, -1);
    glEnd();
    glutSwapBuffers();
}
void timer(int)
{
    angle += 1.0f;
    if (angle > 360)
```

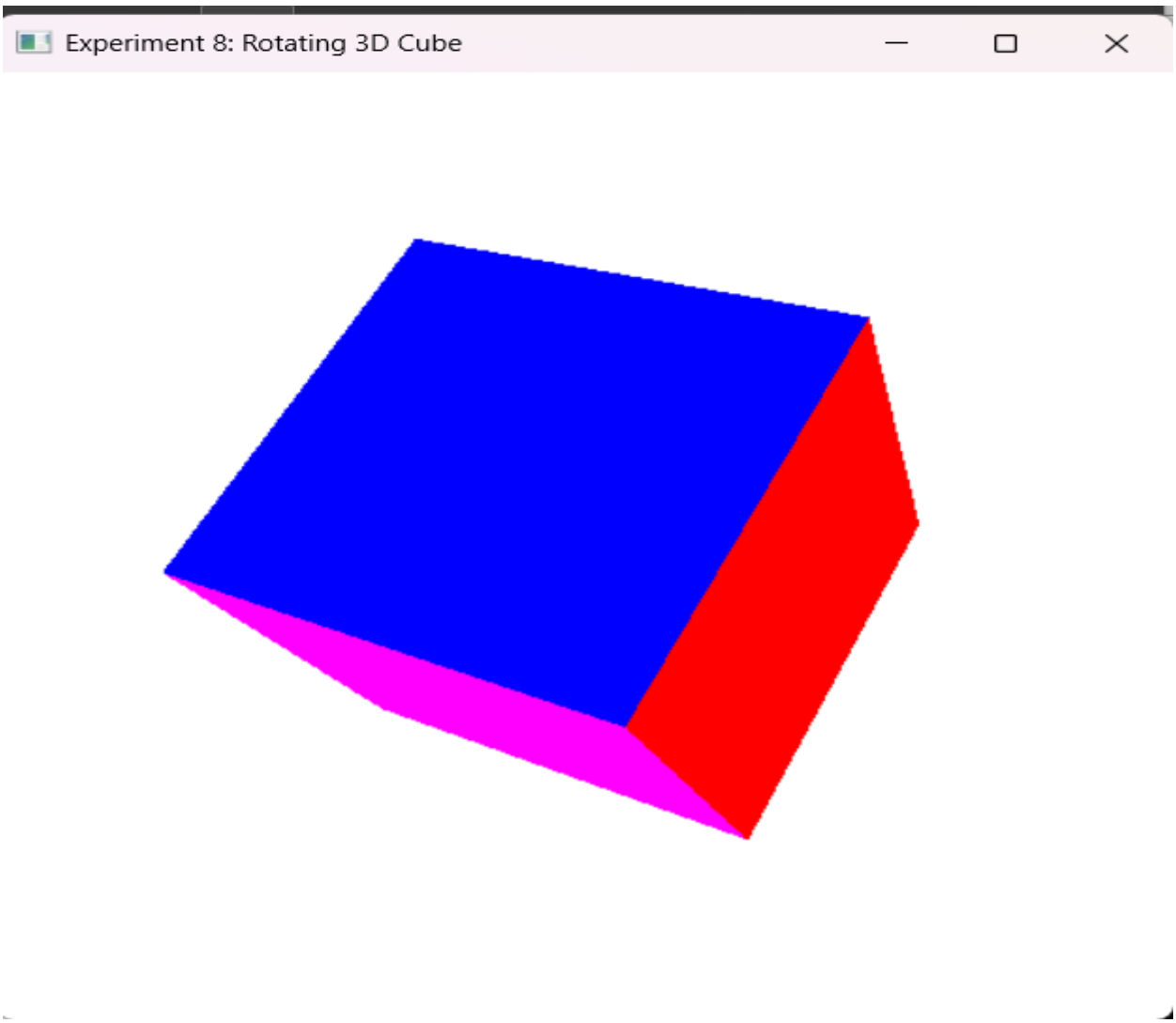
```
        angle -= 360;


    glutPostRedisplay();
    glutTimerFunc(16, timer, 0);
}
void init()
{
    glClearColor(1, 1, 1, 1);
    glEnable(GL_DEPTH_TEST);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(45, 1, 1, 100);
    glMatrixMode(GL_MODELVIEW);
}
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB |
GLUT_DEPTH);
    glutInitWindowSize(500, 500);
```

```
    glutCreateWindow("Experiment 8: Rotating 3D Cube");

    init();
    glutDisplayFunc(display);
    glutTimerFunc(0, timer, 0);
    glutMainLoop();
    return 0;
}
```





Conclusion / Discussion:

This experiment demonstrates the concept of 3D rotation transformation using OpenGL. By applying rotation along multiple axes and enabling depth testing, realistic 3D motion is achieved. This experiment helps in understanding the fundamentals of 3D graphics and object transformations used in games, simulations, and animation systems.