

```
from OpenGL.GL import *
from OpenGL.GLUT import *
from OpenGL.GLU import *

# Global variables to track rotation angles
rotate_x = 0
rotate_y = 0

# Display callback function
def display():
    global rotate_x, rotate_y

    # Clear screen and Z-buffer
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

    # Reset transformations
    glLoadIdentity()

    # Apply rotations
    glRotatef(rotate_x, 1.0, 0.0, 0.0) # Rotate about x-axis
    glRotatef(rotate_y, 0.0, 1.0, 0.0) # Rotate about y-axis

    # Render the cube
    render_Scene()

    # Swap buffers
    glutSwapBuffers()

# Scene render function
def render_Scene():
    # Multi-colored side - FRONT
    glBegin(GL_POLYGON)
    glColor3f(1.0, 0.0, 0.0)
    glVertex3f(0.5, -0.5, -0.5)
    glColor3f(0.0, 1.0, 0.0)
    glVertex3f(0.5, 0.5, -0.5)
    glColor3f(0.0, 0.0, 1.0)
    glVertex3f(-0.5, 0.5, -0.5)
    glColor3f(1.0, 0.0, 1.0)
    glVertex3f(-0.5, -0.5, -0.5)
    glEnd()

    # White side - BACK
    glBegin(GL_POLYGON)
    glColor3f(1.0, 1.0, 1.0)
    glVertex3f(0.5, -0.5, 0.5)
    glVertex3f(0.5, 0.5, 0.5)
    glVertex3f(-0.5, 0.5, 0.5)
    glVertex3f(-0.5, -0.5, 0.5)
    glEnd()

    # Purple side - RIGHT
```

```
glBegin(GL_POLYGON)
glColor3f(1.0, 0.0, 1.0)
glVertex3f(0.5, -0.5, -0.5)
glVertex3f(0.5, 0.5, -0.5)
glVertex3f(0.5, 0.5, 0.5)
glVertex3f(0.5, -0.5, 0.5)
glEnd()

# Green side - LEFT
glBegin(GL_POLYGON)
glColor3f(0.0, 1.0, 0.0)
glVertex3f(-0.5, -0.5, 0.5)
glVertex3f(-0.5, 0.5, 0.5)
glVertex3f(-0.5, 0.5, -0.5)
glVertex3f(-0.5, -0.5, -0.5)
glEnd()

# Blue side - TOP
glBegin(GL_POLYGON)
glColor3f(0.0, 0.0, 1.0)
glVertex3f(0.5, 0.5, 0.5)
glVertex3f(0.5, 0.5, -0.5)
glVertex3f(-0.5, 0.5, -0.5)
glVertex3f(-0.5, 0.5, 0.5)
glEnd()

# Red side - BOTTOM
glBegin(GL_POLYGON)
glColor3f(1.0, 0.0, 0.0)
glVertex3f(0.5, -0.5, -0.5)
glVertex3f(0.5, -0.5, 0.5)
glVertex3f(-0.5, -0.5, 0.5)
glVertex3f(-0.5, -0.5, -0.5)
glEnd()

# Special key callback function
def special_keys(key, x, y):
    global rotate_x, rotate_y

    if key == GLUT_KEY_UP:      # Rotate up
        rotate_x += 5
    elif key == GLUT_KEY_DOWN:  # Rotate down
        rotate_x -= 5
    elif key == GLUT_KEY_LEFT:  # Rotate left
        rotate_y -= 5
    elif key == GLUT_KEY_RIGHT: # Rotate right
        rotate_y += 5

    # Redisplay after rotation
    glutPostRedisplay()

# Reshape callback function
def reshape(width, height):
    # Set a new projection matrix
```

```
glMatrixMode(GL_PROJECTION)
glLoadIdentity()
gluPerspective(40, width / height, 1, 10)
glTranslatef(0.0, 0.0, -5)

# Set the viewport to cover the new window
glViewport(0, 0, width, height)

# Main function
def main():
    # Initialize GLUT
    glutInit()

    # Set display mode
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH)

    # Set the window size
    glutInitWindowSize(500, 500)

    # Create the window
    glutCreateWindow("3D Rotating Cube")

    # Enable depth test
    glEnable(GL_DEPTH_TEST)

    # Set callback functions
    glutDisplayFunc(display)
    glutReshapeFunc(reshape)
    glutSpecialFunc(special_keys)

    # Start the main loop
    glutMainLoop()

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

## 3D Rotating Cube with OpenGL and GLUT

This code demonstrates creating a 3D rotating cube using OpenGL and GLUT libraries. The cube is colored differently on each side and allows for interactive rotation using the arrow keys.

### Functionality

1. **Imports:** Necessary modules from OpenGL libraries are imported for graphics, windowing, and utility functions.
2. **Global Variables:** `rotate_x` and `rotate_y` track the current rotation angles for the cube.
3. **Callback Functions:**
  - `display():`
    - Clears the screen and depth buffer.

- Resets any prior transformations.
- Applies rotations around the X and Y axes based on the global variables.
- Calls `render_Scene` to draw the cube.
- Swaps the display buffers for smooth animation.
- `render_Scene()`:
  - Defines each face (side) of the cube using polygons.
  - Sets a different color for each face.
  - Specifies the vertices for each polygon, creating the cube shape.
- `special_keys(key, x, y)`:
  - Handles special key presses (arrow keys) for rotation:
    - Up arrow: Rotates the cube upwards.
    - Down arrow: Rotates the cube downwards.
    - Left arrow: Rotates the cube left.
    - Right arrow: Rotates the cube right.
  - Updates the global rotation variables based on the key press.
  - Triggers a redraw with `glutPostRedisplay` to reflect the changes.
- `reshape(width, height)`:
  - Defines a new perspective projection matrix based on the window size.
  - Sets the viewport to cover the entire window.

#### 4. Main Function (`main()`):

- Initializes GLUT.
- Sets the display mode with double buffering, RGB color, and depth testing.
- Defines the window size and creates the window with a title.
- Enables depth testing for realistic rendering.
- Registers callback functions for display, window reshape, and special key presses.
- Starts the main GLUT event loop.

## Usage

1. Run the script.
2. Use the arrow keys to rotate the cube in the desired directions.

This program serves as a basic example of creating and manipulating 3D objects in OpenGL. Further exploration could involve implementing lighting, textures, and camera movement for a more immersive experience.