Draw a color cube and spin it using OpenGL transformation matrices.

```
#include <GL/glut.h>
// Vertex coordinates of the cube
GLfloat vertices[][3] = {
   {-1.0, -1.0, -1.0},
   {1.0, -1.0, -1.0},
   {1.0, 1.0, -1.0},
   {-1.0, 1.0, -1.0},
   {-1.0, -1.0, 1.0},
   {1.0, -1.0, 1.0},
   {1.0, 1.0, 1.0},
   {-1.0, 1.0, 1.0}
};

// Colors for each vertex
GLfloat colors[][3] = {
   {0.0, 0.0, 0.0}, // Black
   {1.0, 0.0, 0.0}, // Red
   {1.0, 1.0, 0.0}, // Yellow
   {0.0, 1.0, 0.0}, // Green
   {0.0, 0.0, 1.0}, // Blue
   {1.0, 0.0, 1.0}, // Magenta
   {1.0, 1.0, 1.0}, // White
   {0.0, 1.0, 1.0}  // Cyan
};

// Function to draw polygons (faces of the cube)
void polygon(int a, int b, int c, int d) {
   glBegin(GL_POLYGON);
   glColor3fv(colors[a]); glVertex3fv(vertices[a]);
   glColor3fv(colors[b]); glVertex3fv(vertices[b]);
   glColor3fv(colors[c]); glVertex3fv(vertices[c]);
   glColor3fv(colors[d]); glVertex3fv(vertices[d]);
   glEnd();
}

// Function to draw the entire cube
void colorcube() {
   polygon(0, 3, 2, 1); // Front face
   polygon(2, 3, 7, 6); // Top face
   polygon(0, 4, 7, 3); // Left face
   polygon(1, 2, 6, 5); // Right face
   polygon(4, 5, 6, 7); // Back face
   polygon(0, 1, 5, 4); // Bottom face
}

// Angles of rotation for x, y, z axes
static GLfloat theta[] = {0.0, 0.0, 0.0};
static GLint axis = 2; // Default rotation axis is z-axis
```

```
// Display function for rendering the cube
void display() {
   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
   glLoadIdentity();
   glRotatef(theta[0], 1.0, 0.0, 0.0); // Rotate around x-axis
   glRotatef(theta[1], 0.0, 1.0, 0.0); // Rotate around y-axis
   glRotatef(theta[2], 0.0, 0.0, 1.0); // Rotate around z-axis
   colorcube(); // Draw the cube
   glutSwapBuffers();
}

// Function to update the rotation angle for the cube
void spinCube() {
   theta[axis] += 0.5; // Increment rotation angle
   if (theta[axis] > 360.0) theta[axis] -= 360.0; // Ensure angle stays within 0-
360
   glutPostRedisplay(); // Request a redraw
}

// Function to handle mouse input for rotating along different axes
void mouse(int btn, int state, int x, int y) {
   if (state == GLUT_DOWN) {
      if (btn == GLUT_LEFT_BUTTON) axis = 0;  // Rotate along x-axis
      if (btn == GLUT_MIDDLE_BUTTON) axis = 1; // Rotate along y-axis
      if (btn == GLUT_RIGHT_BUTTON) axis = 2;  // Rotate along z-axis
   }
}
```

```
// Reshape function to handle window resizing
void myReshape(int w, int h) {
   glViewport(0, 0, w, h); // Set the viewport to cover the new window size
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   if (w <= h)
      glOrtho(-2.0, 2.0, -2.0 * (GLfloat)h / (GLfloat)w, 2.0 * (GLfloat)h /
(GLfloat)w, -10.0, 10.0);
   else
      glOrtho(-2.0 * (GLfloat)w / (GLfloat)h, 2.0 * (GLfloat)w / (GLfloat)h, -2.0,
2.0, -10.0, 10.0);
   glMatrixMode(GL_MODELVIEW);
}

// Main function
int main(int argc, char** argv) {
   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
   glutInitWindowSize(500, 500);
   glutCreateWindow("Color Cube Viewer");
   glutReshapeFunc(myReshape);  // Register reshape function
   glutDisplayFunc(display);    // Register display function
   glutMouseFunc(mouse);        // Register mouse input function
   glutIdleFunc(spinCube);      // Register idle function (for continuous
spinning)
   glEnable(GL_DEPTH_TEST);     // Enable depth testing
```

```
    glutMainLoop();              // Enter the GLUT event loop
    return 0;
}
```