







## Cube Rotation

```

#include<stdlib.h>
#include<stdio.h>
#include<GL/glut.h>
#include<math.h>
float v[8][3] = { {-1,-1,1},{-1,1,1},{1,1,1},{1,-1,1},{-1,-1,-1},{-1,1,-1},{1,1,-1},{1,-1,-1} };
float p[8][3] = {
{0,0,1},{0,1,1},{1,1,1},{1,0,1},{0,0,0},{0,1,0},{1,1,0},{1,0,0}};
float theta[3] = { 0,0,0 };
int flag = 2;

void myinit()
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-2, 2, -2, 2, -2, 2);
    glMatrixMode(GL_MODELVIEW);
}
void idlefunc()
{
    theta[flag]++;
    if (theta[flag] > 360)theta[flag] = 0;
    glutPostRedisplay();
}
void mousefunc(int button, int status, int x, int y)
{
    if (button == GLUT_LEFT_BUTTON && status == GLUT_DOWN)
        flag = 2;
    if (button == GLUT_MIDDLE_BUTTON && status == GLUT_DOWN)
        flag = 1;
    if (button == GLUT_RIGHT_BUTTON && status == GLUT_DOWN)
        flag = 0;
}
void drawpoly(int a, int b, int c, int d)
{
    glBegin(GL_POLYGON);
    glColor3fv(p[a]);
    glVertex3fv(v[a]);
    glColor3fv(p[b]);
    glVertex3fv(v[b]);
    glColor3fv(p[c]);
    glVertex3fv(v[c]);
    glColor3fv(p[d]);
    glVertex3fv(v[d]);
    glEnd();
}
void colorcube()

```



