**Program to draw a color cube and spin it using openGL transformation matrices.**

**Objective :** In this program the students will learn to draw a color cube and spin it using OpenGL functions.

#include <GL/glut.h>

#include <stdio.h>

#include <stdlib.h>

GLfloat vertices[ ]={ -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 };

GLfloat normals[ ]={ -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 };

GLfloat colors[ ]={ 0.0,0.0,0.0,

1.0,0.0,0.0,

1.0,1.0,0.0,

0.0,1.0,0.0,

0.0,0.0,1.0,

1.0,0.0,1.0,

1.0,1.0,1.0,

0.0,1.0,1.0};

GLubyte cubeIndices[]={0,3,2,1,

2,3,7,6,

0,4,7,3,

1,2,6,5,

4,5,6,7,

0,1,5,4 };

static GLfloat theta[]={0.0,0.0,0.0};

static GLint axis=2;

void display(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);

glLoadIdentity();

glRotatef(theta[0],1.0,0.0,0.0);

glRotatef(theta[1],0.0,1.0,0.0);

glRotatef(theta[2],0.0,0.0,1.0);

glDrawElements(GL\_QUADS,24,GL\_UNSIGNED\_BYTE,cubeIndices);

glFlush();

glutSwapBuffers();

}

void mouse(int btn,int state,int x,int y)

{

if(btn==GLUT\_LEFT\_BUTTON && state==GLUT\_DOWN)axis=0;

if(btn==GLUT\_RIGHT\_BUTTON && state==GLUT\_DOWN) axis=1;

if(btn==GLUT\_MIDDLE\_BUTTON && state==GLUT\_DOWN) axis=2;

}

void spincube()

{

theta[axis]+=2.0;

if(theta[axis]>360.0)

theta[axis]-=360.0;

glutPostRedisplay();

}

void myReshape(int w,int h)

{

glViewport(0,0,w,h);

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);

}

int main(int argc,char \*\*argv)

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_DOUBLE|GLUT\_RGB|GLUT\_DEPTH);

glutInitWindowSize(500,500);

glutCreateWindow("color cube");

glutReshapeFunc(myReshape);

glutDisplayFunc(display);

glutMouseFunc(mouse);

glutIdleFunc(spincube);

glEnable(GL\_DEPTH\_TEST);

glEnableClientState(GL\_COLOR\_ARRAY);

glEnableClientState(GL\_VERTEX\_ARRAY);

glEnableClientState(GL\_NORMAL\_ARRAY);

glVertexPointer(3,GL\_FLOAT,0,vertices);

glColorPointer(3,GL\_FLOAT,0,colors);

glNormalPointer(GL\_FLOAT,0,normals);

glColor3f(1.0,1.0,1.0);

glutMainLoop();

}

**Output:**

