Lab4Cam.c

```
#include
#include
#include

GLfloat Cx=0,Cy=0,Cz=3;

void MyInit()
{
glClearColor(0,0,0,1);
glEnable(GL_DEPTH_TEST);

glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glFrustum(-1,1,-1,1,2,10);
glMatrixMode(GL_MODELVIEW);
}

void Square(GLfloat A[],GLfloat B[],GLfloat C[],GLfloat D[])
{
glBegin(GL_POLYGON);
glVertex3fv(A);
glVertex3fv(B);
glVertex3fv(C);
glVertex3fv(D);
glEnd();
}

void Cube(GLfloat V0[],GLfloat V1[],GLfloat V2[],GLfloat V3[],GLfloat
V4[],GLfloat V5[],GLfloat V6[],GLfloat V7[])
{
glColor3f(1,0,0);
Square(V0,V1,V2,V3);
glColor3f(0,1,0);
Square(V4,V5,V6,V7);
glColor3f(0,0,1);
Square(V0,V4,V7,V3);
glColor3f(1,1,0);
Square(V1,V5,V6,V2);
glColor3f(1,0,1);
Square(V3,V2,V6,V7);
glColor3f(0,1,1);
Square(V0,V1,V5,V4);
}

void Draw()
{
GLfloat V[8][3] = {
{-0.5, 0.5, 0.5},
{ 0.5, 0.5, 0.5},
{ 0.5,-0.5, 0.5},
{-0.5,-0.5, 0.5},
{-0.5, 0.5,-0.5},
{ 0.5, 0.5,-0.5},
{ 0.5,-0.5,-0.5},
{-0.5,-0.5,-0.5}
};
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

glLoadIdentity();
gluLookAt(Cx,Cy,Cz,0,0,0,0,1,0);

Cube(V[0],V[1],V[2],V[3],V[4],V[5],V[6],V[7]);

glutSwapBuffers();
}

void Key(unsigned char ch,int x,int y)
{
switch(ch)
{
case 'x' : Cx = Cx - 0.5; break;
case 'X' : Cx = Cx + 0.5; break;

case 'y' : Cy = Cy - 0.5; break;
case 'Y' : Cy = Cy + 0.5; break;

case 'z' : Cz = Cz - 0.5; break;
case 'Z' : Cz = Cz + 0.5; break;
}
glutPostRedisplay();
}

int main(int argC,char *argV[])
{
glutInit(&argC,argV);
glutInitWindowSize(600,600);
glutInitWindowPosition(100,150);
glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH);
glutCreateWindow("Color Cube with Camera");
MyInit();
glutDisplayFunc(Draw);
glutKeyboardFunc(Key);
glutMainLoop();
return 0;
}
```