#include <stdlib.h>

#include <GL/glut.h>

#include <iostream>

GLsizei W = 1000, H = 1000;

GLfloat Rx = 0.0, Ry = 0.0;

void Key(int key, int x, int y);

void Cube(GLfloat x, GLfloat y, GLfloat z, GLfloat l);

void Cube(GLfloat x, GLfloat y, GLfloat z, GLfloat l) {

GLfloat h = l \* 0.5;

glRotatef(Rx, 1, 0, 0);

glRotatef(Ry, 0, 1, 0);

GLfloat V[] =

{

//FRONT

x - h,y + h,z + h,

x + h,y + h,z + h,

x + h,y - h,z + h,

x - h,y - h,z + h,

//BACK

x - h,y + h,z - h,

x + h,y + h,z - h,

x + h,y - h,z - h,

x - h,y - h,z - h,

//LEFT

x - h,y + h,z + h,

x - h,y + h,z - h,

x - h,y - h,z - h,

x - h,y - h,z + h,

//RIGHT

x + h,y + h,z + h,

x + h,y + h,z - h,

x + h,y - h,z - h,

x + h,y - h,z + h,

//TOP

x - h,y + h,z + h,

x - h,y + h,z - h,

x + h,y + h,z - h,

x + h,y + h,z + h,

//BOTTOM

x - h,y - h,z + h,

x - h,y - h,z - h,

x + h,y - h,z - h,

x + h,y - h,z + h,

};

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);

glEnableClientState(GL\_VERTEX\_ARRAY);

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

glDrawArrays(GL\_QUADS, 0, 24);

glDisableClientState(GL\_VERTEX\_ARRAY);

}

void Key(int key, int x, int y) {

if (key == GLUT\_KEY\_RIGHT) {

Ry += 4;

}

else if (key == GLUT\_KEY\_LEFT) {

Ry -= 4;

}

else if (key == GLUT\_KEY\_UP) {

Rx += 4;

}

else if (key == GLUT\_KEY\_DOWN) {

Rx -= 4;

}

glutPostRedisplay();

}

void start() {

glViewport(0.0, 0.0, W, H);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0 ,W, 0, H, 0, 1000000000000000000);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

void display() {

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

glPushMatrix();

Cube((W / 2), (H / 2), -500, 200);

glPopMatrix();

glutSwapBuffers();

}

int main(int argc, char\* argv[]) {

glutInit(&argc, argv);

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

glutInitWindowSize(W, H);

glutCreateWindow("Cubo");

start();

glutDisplayFunc(display);

glutSpecialFunc(Key);

glutMainLoop();

return 0;

}