#include <stdio.h>

#include <stdarg.h>

#include <math.h>

#include <GL/glut.h>

int W = 800, H = 800;

double Ry = 0;

double Rx = 0;

void display();

void specialKeys();

void display() {

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

glLoadIdentity();

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

glRotatef(Rx, 1.0, 0.0, 0.0);

glRotatef(Ry, 0.0, 1.0, 0.0);

//FRONT

glBegin(GL\_POLYGON);

glColor3f(1.0, 0.5, 0.0);

glVertex3f(-0.3, -0.3, -0.3);

glVertex3f(-0.3, 0.3, -0.3);

glVertex3f(0.3, 0.3, -0.3);

glVertex3f(0.3, -0.3, -0.3);

glEnd();

//BACK

glBegin(GL\_POLYGON);

glColor3f(1.0, 1.0, 0.0);

glVertex3f(0.3, -0.3, 0.3);

glVertex3f(0.3, 0.3, 0.3);

glVertex3f(-0.3, 0.3, 0.3);

glVertex3f(-0.3, -0.3, 0.3);

glEnd();

//RIGHT

glBegin(GL\_POLYGON);

glColor3f(1.0, 0.0, 1.0);

glVertex3f(0.3, -0.3, -0.3);

glVertex3f(0.3, 0.3, -0.3);

glVertex3f(0.3, 0.3, 0.3);

glVertex3f(0.3, -0.3, 0.3);

glEnd();

//LEFT

glBegin(GL\_POLYGON);

glColor3f(0.0, 1.0, 0.0);

glVertex3f(-0.3, -0.3, 0.3);

glVertex3f(-0.3, 0.3, 0.3);

glVertex3f(-0.3, 0.3, -0.3);

glVertex3f(-0.3, -0.3, -0.3);

glEnd();

//UP

glBegin(GL\_POLYGON);

glColor3f(0.0, 0.0, 1.0);

glVertex3f(0.3, 0.3, 0.3);

glVertex3f(0.3, 0.3, -0.3);

glVertex3f(-0.3, 0.3, -0.3);

glVertex3f(-0.3, 0.3, 0.3);

glEnd();

//DOWN

glBegin(GL\_POLYGON);

glColor3f(1.0, 0.0, 0.0);

glVertex3f(0.3, -0.3, -0.3);

glVertex3f(0.3, -0.3, 0.3);

glVertex3f(-0.3, -0.3, 0.3);

glVertex3f(-0.3, -0.3, -0.3);

glEnd();

glFlush();

glutSwapBuffers();

}

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

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

Ry += 5;

}

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

Ry-= 5;

}

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

Rx += 5;

}

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

Rx -= 5;

}

glutPostRedisplay();

}

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

glutInit(&argc, argv);

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

glutInitWindowSize(800, 800);

glutCreateWindow("Cubo");

glEnable(GL\_DEPTH\_TEST);

glutDisplayFunc(display);

glutSpecialFunc(specialKeys);

glutMainLoop();

return 0;

}