#include <GL/glut.h>

#include <math.h>

#define BASE\_HEIGHT 4.0/2

#define BASE\_RADIUS 1.0/2

#define HEAD\_HEIGHT 1.25/2

#define HEAD\_RADIUS 0.75/2

#define NECK\_HEIGHT 0.5/2

#define EYE\_LEVEL 0.75/2

#define NOSE\_LENGTH 0.5/2

#define LOWER\_ARM\_HEIGHT 2.0/2

#define LOWER\_ARM\_WIDTH 0.5/2

#define UPPER\_ARM\_HEIGHT 1.25/2

#define UPPER\_ARM\_WIDTH 0.5/2

#define ARM\_TRANSLATION 0.22/2

#define alpha 0.0

#define pi 3.14159265

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

static GLint axis = 0;

GLUquadricObj\* p;

GLfloat x = 0.0;

GLfloat y = 0.0;

GLfloat xpos = 0.0;

GLfloat ypos = 0.0;

GLfloat zpos = 0.0;

GLfloat ambient[3];

GLfloat diffuse[3];

GLfloat specular[3];

GLfloat shiness[] = { 50.0f };

float width = 500;

float height = 500;

void base(void);

void head(void);

void neck(void);

void upper\_rarm(void);

void upper\_larm(void);

void lower\_rarm(void);

void lower\_larm(void);

void init(void);

void display(void);

void reshape(int width, int height);

void keyboard(unsigned char, int, int);

void processSpecialKeys(int, int, int);

void jump(void);

void lsphere(void);

void init1(void);

void base(void) {

double angle, angleInc;

int i;

angle = pi / 180;

angleInc = angle;

glPushMatrix();

ambient[0] = 2.0; ambient[1] = 0.0; ambient[2] = 0.0;

diffuse[0] = 1.0; diffuse[1] = 0.0; diffuse[2] = 0.0;

specular[0] = 0.7; specular[1] = 0.6; specular[2] = 0.5;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, shiness);

glRotatef(-90.0, 1.0, 0.0, 0.0);//

gluQuadricDrawStyle(p, GLU\_LINE);

gluCylinder(p, BASE\_RADIUS, BASE\_RADIUS, BASE\_HEIGHT, 20, 20);

glPopMatrix();

glPushMatrix();

gluQuadricDrawStyle(p, GLU\_LINE);// BODY STRUCTURE

glTranslatef(0.0, BASE\_HEIGHT, 0.0);

glRotatef(-90.0, 1.0, 0.0, 0.0);

gluDisk(p, 0.0, BASE\_RADIUS, 20, 20);

glTranslatef(0.0, 0.0, -BASE\_HEIGHT); //HERE BRAIN OPEN

gluDisk(p, 0.0, BASE\_RADIUS, 20, 20);

glPopMatrix();

}

void neck(void) {

glPushMatrix();

ambient[0] = 1.0; ambient[1] = 1.0; ambient[2] = 0.0;

diffuse[0] = 1.0; diffuse[1] = 1.0; diffuse[2] = 0.0;

specular[0] = 0.7; specular[1] = 0.6; specular[2] = 0.5;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, shiness);

glTranslatef(0.0, BASE\_HEIGHT, 0.0);

glRotatef(-90.0, 1.0, 0.0, 0.0);

gluQuadricDrawStyle(p, GLU\_FILL);

gluCylinder(p, HEAD\_RADIUS / 2, HEAD\_RADIUS / 2, HEAD\_HEIGHT, 8, 6);

glPopMatrix();

}

void head(void) {

glPushMatrix();

ambient[0] = 1.0; ambient[1] = 0.0; ambient[2] = 1.0;

diffuse[0] = 1.0; diffuse[1] = 0.0; diffuse[2] = 1.0;

specular[0] = 0.7; specular[1] = 0.6; specular[2] = 0.5;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, shiness);

glRotatef(-90.0, 1.0, 0.0, 0.0);

gluQuadricDrawStyle(p, GLU\_FILL);

gluCylinder(p, HEAD\_RADIUS, HEAD\_RADIUS, HEAD\_HEIGHT, 20, 20);

glPushMatrix();

gluDisk(p, 0.0, HEAD\_RADIUS, 20, 20);

glTranslatef(0.0, 0.0, HEAD\_HEIGHT);

gluDisk(p, 0.0, HEAD\_RADIUS, 20, 20);

glPopMatrix();

glPushMatrix();

glTranslatef(0.25, -HEAD\_RADIUS + 0.12, EYE\_LEVEL);

ambient[0] = 1.0; ambient[1] = 1.0; ambient[2] = 1.0;

diffuse[0] = 1.0; diffuse[1] = 1.0; diffuse[2] = 1.0;

specular[0] = 0.5; specular[1] = 0.5; specular[2] = 0.5;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, shiness);

gluQuadricDrawStyle(p, GLU\_FILL);

gluSphere(p, 0.125, 6, 6);

glPopMatrix();

glPushMatrix();

glTranslatef(-0.25, -HEAD\_RADIUS + 0.12, EYE\_LEVEL);

ambient[0] = 1.0; ambient[1] = 1.0; ambient[2] = 1.0;

diffuse[0] = 1.0; diffuse[1] = 1.0; diffuse[2] = 1.0;

specular[0] = 0.5; specular[1] = 0.5; specular[2] = 0.5;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, shiness);

gluQuadricDrawStyle(p, GLU\_FILL);

gluSphere(p, 0.125, 6, 6);

glPopMatrix();

glPushMatrix();

ambient[0] = 1.0; ambient[1] = 0.5; ambient[2] = 0.0;

diffuse[0] = 1.0; diffuse[1] = 0.5; diffuse[2] = 0.0;

specular[0] = 0.5; specular[1] = 0.5; specular[2] = 0.5;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, shiness);

glTranslatef(0.0, -HEAD\_RADIUS, NOSE\_LENGTH);

glRotatef(90.0, 1.0, 0.0, 0.0);

gluQuadricDrawStyle(p, GLU\_FILL);

gluCylinder(p, 0.125, 0, NOSE\_LENGTH, 8, 6);

glPopMatrix();

glPopMatrix();

}

void lower\_rarm(void) {

glPushMatrix();

ambient[0] = 0.0; ambient[1] = 1.0; ambient[2] = 0.0;

diffuse[0] = 0.0; diffuse[1] = 1.0; diffuse[2] = 0.0;

specular[0] = 0.7; specular[1] = 0.6; specular[2] = 0.5;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, shiness);

glTranslatef(0.0, 0.5 \* LOWER\_ARM\_HEIGHT, ARM\_TRANSLATION);

glScalef(LOWER\_ARM\_WIDTH, LOWER\_ARM\_HEIGHT, LOWER\_ARM\_WIDTH);

glutSolidCube(1.0);

glPopMatrix();

}

void lower\_larm(void) {

glPushMatrix();

ambient[0] = 0.0; ambient[1] = 1.0; ambient[2] = 0.0;

diffuse[0] = 0.0; diffuse[1] = 1.0; diffuse[2] = 0.0;

specular[0] = 0.7; specular[1] = 0.6; specular[2] = 0.5;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, shiness);

glTranslatef(0.0, 0.5 \* LOWER\_ARM\_HEIGHT, -ARM\_TRANSLATION);

glScalef(LOWER\_ARM\_WIDTH, LOWER\_ARM\_HEIGHT, LOWER\_ARM\_WIDTH);

glutSolidCube(1.0);

glPopMatrix();

}

void upper\_rarm(void) {

glPushMatrix();

ambient[0] = 0.0; ambient[1] = 0.0; ambient[2] = 1.0;

diffuse[0] = 0.0; diffuse[1] = 0.0; diffuse[2] = 1.0;

specular[0] = 0.7; specular[1] = 0.6; specular[2] = 0.5;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, shiness);

glTranslatef(0.0, 0.5 \* UPPER\_ARM\_HEIGHT, ARM\_TRANSLATION);

glScalef(UPPER\_ARM\_WIDTH, UPPER\_ARM\_HEIGHT, UPPER\_ARM\_WIDTH);

glutSolidCube(1.0);

glPopMatrix();

}

void upper\_larm(void) {

glPushMatrix();

ambient[0] = 0.0; ambient[1] = 0.0; ambient[2] = 1.0;

diffuse[0] = 0.0; diffuse[1] = 0.0; diffuse[2] = 1.0;

specular[0] = 0.7; specular[1] = 0.6; specular[2] = 0.5;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, shiness);

glTranslatef(0.0, 0.5 \* UPPER\_ARM\_HEIGHT, -ARM\_TRANSLATION);

glScalef(UPPER\_ARM\_WIDTH, UPPER\_ARM\_HEIGHT, UPPER\_ARM\_WIDTH);

glutSolidCube(1.0);

glPopMatrix();

}

void init1(void) {

GLfloat mat\_specular[] = { 1.0, 1.0, 1.0, 1.0 };

GLfloat mat\_shininess[] = { 50.0 };

GLfloat light\_position[] = { 1.0, 1.0, 1.0, 1.0 };

GLfloat white\_light[] = { 0.3, 0.8, 0.8, 1.0 }; // brightness for the above circle

GLfloat lmodel\_ambient[] = { 1.0, 1.0, 0.0, 1.0 };

glClearColor(0.0, 0.0, 0.0, 0.0);

glShadeModel(GL\_SMOOTH);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, mat\_specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, mat\_shininess);

glLightfv(GL\_LIGHT0, GL\_POSITION, light\_position);

glLightfv(GL\_LIGHT0, GL\_DIFFUSE, white\_light);

glLightfv(GL\_LIGHT0, GL\_SPECULAR, white\_light);

glLightModelfv(GL\_LIGHT\_MODEL\_AMBIENT, lmodel\_ambient);

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

glEnable(GL\_DEPTH\_TEST);

}

void lsphere(void) {

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

glutSolidSphere(1.0, 20, 16);

glFlush();

}

void display(void) {

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

glLoadIdentity();

gluLookAt(0.3, 5.3, 12.0, 0.0, 1.25, 0.0, 0.0, 1.0, 0.0);

glPushMatrix();

ambient[0] = 1.0; ambient[1] = 0.3; ambient[2] = 0.3;

diffuse[0] = 1.0; diffuse[1] = 1.0; diffuse[2] = 1.0;

specular[0] = 0.7; specular[1] = 0.6; specular[2] = 0.5;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, shiness);

glTranslatef(0.0, 5.0, 0.0);

lsphere();

glPopMatrix();

glPushMatrix();

ambient[0] = 0.3; ambient[1] = 0.3; ambient[2] = 0.3;

diffuse[0] = 0.0; diffuse[1] = 0.0; diffuse[2] = 1.0;

specular[0] = 0.7; specular[1] = 0.6; specular[2] = 0.5;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, shiness);

glTranslatef(3.0, 0.5, 1.0);

glutSolidCube(1.0);

glPopMatrix();

glPushMatrix();

ambient[0] = 0.3; ambient[1] = 0.3; ambient[2] = 0.3;

diffuse[0] = 1.0; diffuse[1] = 0.0; diffuse[2] = 1.0;

specular[0] = 0.7; specular[1] = 0.6; specular[2] = 0.5;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, shiness);

glTranslatef(-4.0, 0.5, 3.0);

glutSolidCone(0.5, 1.5, 20, 15);

glPopMatrix();

glPushMatrix();

ambient[0] = 0.3; ambient[1] = 0.3; ambient[2] = 0.3;

diffuse[0] = 1.0; diffuse[1] = 0.0; diffuse[2] = 0.0;

specular[0] = 0.7; specular[1] = 0.6; specular[2] = 0.5;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, shiness);

glTranslatef(-3.0, 0.5, -3.0);

glutSolidTeapot(1.0);

glPopMatrix();

glPushMatrix();

ambient[0] = 0.5; ambient[1] = 0.5; ambient[2] = 0.5;

diffuse[0] = 1.0; diffuse[1] = 1.0; diffuse[2] = 1.0;

specular[0] = 0.7; specular[1] = 0.6; specular[2] = 0.5;

glMaterialfv(GL\_FRONT, GL\_AMBIENT, ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, shiness);

glBegin(GL\_POLYGON);

glVertex3f(5.0, 0.0, 5.0);

glVertex3f(5.0, 0.0, -5.0);

glVertex3f(-5.0, 0.0, -5.0);

glVertex3f(-5.0, 0.0, 5.0);

glVertex3f(5.0, 0.0, 5.0);

glEnd();

glPopMatrix();

glTranslatef(xpos, ypos, zpos);

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

base();

neck();

glPushMatrix();

glTranslatef(0.0, BASE\_HEIGHT + HEAD\_HEIGHT / 2, 0.0);

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

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

head();

glPopMatrix();

glPushMatrix();

glTranslatef(BASE\_RADIUS, BASE\_HEIGHT - BASE\_RADIUS / 2, 0.0);

glRotatef(180.0, 0.0, 0.0, 1.0);

glRotatef(270.0, 0.0, 1.0, 0.0);

glRotatef(theta[4], 0.0, 0.0, 1.0);

lower\_rarm();

glTranslatef(0.0, LOWER\_ARM\_HEIGHT, 0.0);

glRotatef(0.0, 0.0, 0.0, 180.0);

glRotatef(theta[6], 0.0, 0.0, 1.0);

upper\_rarm();

glPopMatrix();

glPushMatrix();

glTranslatef(-BASE\_RADIUS, BASE\_HEIGHT - BASE\_RADIUS / 2, 0.0);

glRotatef(180.0, 0.0, 0.0, 1.0);

glRotatef(270.0, 0.0, 1.0, 0.0);

glRotatef(theta[3], 0.0, 0.0, 1.0);

lower\_larm();

glTranslatef(0.0, LOWER\_ARM\_HEIGHT, 0.0);

glRotatef(0.0, 0.0, 0.0, 180.0);

glRotatef(theta[5], 0.0, 0.0, 1.0);

upper\_larm();

glPopMatrix();

glFlush();

glutSwapBuffers();

}

void keyboard(unsigned char key, int x, int y) {

switch (key) {

case 'a': theta[1] += 5.0;

if (theta[1] > 90.0)

theta[1] = 90.0;

break;

case 'z': theta[1] -= 5.0;

if (theta[1] <= -90.0)

theta[1] = -90.0;

break;

case 'o': theta[2] += 5.0;

if (theta[2] >= 45.0)

theta[2] = 45.0;

break;

case 'p': theta[2] -= 5.0;

if (theta[2] < -45.0)

theta[2] = -45.0;

break;

case 'k': theta[3] -= 5.0; break;

case 'l': theta[3] += 5.0; break;

case 'h': theta[4] -= 5.0; break;

case 'j': theta[4] += 5.0; break;

case 'n': theta[5] -= 5.0; break;

case 'm': theta[5] += 5.0; break;

case 'v': theta[6] -= 5.0; break;

case 'b': theta[6] += 5.0; break;

case 'e': theta[0] = theta[1] = theta[2] = theta[3] = theta[4] = theta[5] = theta[6] = xpos = ypos = zpos = 0.0;

break;

case 'r': theta[0] = theta[1] = theta[2] = theta[3] = theta[4] = theta[5] = theta[6] = 0.0;

break;

case 'q': exit(0); break;

}

glutPostRedisplay();

}

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

switch (key) {

case GLUT\_KEY\_UP:

xpos -= cos(90 \* pi / 180 + theta[0] \* pi / 180);

zpos += sin(90 \* pi / 180 + theta[0] \* pi / 180);

if (xpos > 5)

xpos = 5;

if (zpos > 5)

zpos = 5;

if (xpos < -5)

xpos = -5;

if (zpos < -5)

zpos = -5;

break;

case GLUT\_KEY\_DOWN:

xpos += cos(90 \* pi / 180 + theta[0] \* pi / 180);

zpos -= sin(90 \* pi / 180 + theta[0] \* pi / 180);

if (xpos > 5)

xpos = 5;

if (zpos > 5)

zpos = 5;

if (xpos < -5)

xpos = -5;

if (zpos < -5)

zpos = -5;

break;

case GLUT\_KEY\_LEFT: theta[0] -= 5.0; break;

case GLUT\_KEY\_RIGHT: theta[0] += 5.0; break;

case GLUT\_KEY\_PAGE\_UP: ypos += 1.0; break;

case GLUT\_KEY\_PAGE\_DOWN: ypos -= 1.0; break;

}

glutPostRedisplay();

}

void reshape(int w, int h) {

glViewport(0, 0, w, h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(100.0, (GLfloat)w / (GLfloat)h, 0.5, 100.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

void init(void) {

GLfloat lightIntensity[] = { 0.1f, 0.1f, 0.7f, 1.0f };

GLfloat light\_position[] = { 2.0f, 6.0f, 3.0f, 0.0f };

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

glShadeModel(GL\_SMOOTH);

glEnable(GL\_DEPTH\_TEST);

glEnable(GL\_NORMALIZE);

glLightfv(GL\_LIGHT0, GL\_POSITION, light\_position);

glLightfv(GL\_LIGHT0, GL\_DIFFUSE, lightIntensity);

glClearColor(0.0, 0.0, 0.0, 0.0);

glColor3f(1.0, 0.0, 0.0);

p = gluNewQuadric();

}

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

glutInit(&argc, argv);

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

glutInitWindowSize(width, height);

glutCreateWindow("Robot");

init();

//p = gluNewQuadric();

init1();

glutReshapeFunc(reshape);

glutDisplayFunc(display);

glutKeyboardFunc(keyboard);

glutSpecialFunc(processSpecialKeys);

glutMainLoop();

return 0;

}