import com.jogamp.opengl.GL2;

import com.jogamp.opengl.GLAutoDrawable;

import com.jogamp.opengl.GLCapabilities;

import com.jogamp.opengl.GLEventListener;

import com.jogamp.opengl.GLProfile;

import com.jogamp.opengl.awt.GLCanvas;

import com.jogamp.opengl.glu.GLU;

import javax.swing.JFrame;

public class Assignment3 implements GLEventListener{

private GLU glu;

@Override

public void display(GLAutoDrawable drawable) {

final GL2 gl = drawable.getGL().getGL2();

//drawing 3

MidpointLine(gl, -20,90,30,90);

MidpointLine(gl, 30,90,30,40);

MidpointLine(gl, -20,40,30,40);

MidpointLine(gl, 30,40,30,-10);

MidpointLine(gl, -20,-10,30,-10);

//drawing 2

MidpointLine(gl, 40,90,90,90);

MidpointLine(gl, 90,90,90,40);

MidpointLine(gl, 40,40,90,40);

MidpointLine(gl, 40,40,40,-10);

MidpointLine(gl, 40,-10,90,-10);

}

@Override

public void dispose(GLAutoDrawable arg0) {

//method body

}

@Override

public void init(GLAutoDrawable gld) {

GL2 gl = gld.getGL().getGL2();

glu = new GLU();

gl.glClearColor(0.0f, 0.0f, 0.0f, 0.0f);

gl.glViewport(-100, -50, 50, 100);

gl.glMatrixMode(GL2.GL\_PROJECTION);

gl.glLoadIdentity();

glu.gluOrtho2D(-100.0, 100.0, -100.0, 100.0);

}

@Override

public void reshape(GLAutoDrawable arg0, int arg1, int arg2, int arg3, int arg4) {

// method body

}

public void MidpointLine(GL2 gl, int x1, int y1, int x2, int y2) {

gl.glPointSize(3.0f);

gl.glColor3d(0, 1, 1);

//write your own code

int zone = findZone(x1, y1, x2, y2);

int inZone0[] = new int[4];

inZone0 = convertToZone0(x1, y1, x2, y2, zone);

x1 = inZone0[0];

y1 = inZone0[1];

x2 = inZone0[2];

y2 = inZone0[3];

int d = 2 \* (y2 - y1) - (x2 - x1);

int ne = 2 \* (y2 - y1) - 2 \* (x2 - x1);

int e = 2 \* (y2 - y1);

int dNew;

if (d > 0) {

dNew = d + ne;

} else {

dNew = d + e;

}

int inOriginalZone[] = new int[2];

inOriginalZone = convertToOriginalZone(x1, y1, zone);

gl.glBegin(GL2.GL\_POINTS);// static field

gl.glVertex3f(inOriginalZone[0], inOriginalZone[1], 0);

gl.glEnd();

while (x1 != x2 || y1 != y2) {

x1++;

if(d>0)

y1 += 1;

d = dNew;

if (d > 0) {

dNew = d + ne;

} else {

dNew = d + e;

}

inOriginalZone = convertToOriginalZone(x1, y1, zone);

gl.glBegin(GL2.GL\_POINTS);// static field

gl.glVertex3f(inOriginalZone[0], inOriginalZone[1], 0);

gl.glEnd();

}

}

static int findZone(int x1, int y1, int x2, int y2) {

int dx = x2 - x1;

int dy = y2 - y1;

if (dx < 0) {

if (dy < 0) {

if (Math.abs(dx) > Math.abs(dy)) {

return 4;

} else {

return 5;

}

} else {

if (Math.abs(dx) > Math.abs(dy)) {

return 3;

} else {

return 2;

}

}

} else {

if (dy < 0) {

if (Math.abs(dx) > Math.abs(dy)) {

return 7;

} else {

return 6;

}

} else {

if (Math.abs(dx) > Math.abs(dy)) {

return 0;

} else {

return 1;

}

}

}

}

static int[] convertToZone0(int x1, int y1, int x2, int y2, int zone) {

int a[] = new int[4];

if (zone == 0) {

a[0] = x1;

a[1] = y1;

a[2] = x2;

a[3] = y2;

} else if (zone == 1) {

a[0] = y1;

a[1] = x1;

a[2] = y2;

a[3] = x2;

} else if (zone == 2) {

a[0] = y1;

a[1] = -x1;

a[2] = y2;

a[3] = -x2;

} else if (zone == 3) {

a[0] = -x1;

a[1] = y1;

a[2] = -x2;

a[3] = y2;

} else if (zone == 4) {

a[0] = -x1;

a[1] = -y1;

a[2] = -x2;

a[3] = -y2;

} else if (zone == 5) {

a[0] = -y1;

a[1] = -x1;

a[2] = -y2;

a[3] = -x2;

} else if (zone == 6) {

a[0] = -y1;

a[1] = x1;

a[2] = -y2;

a[3] = x2;

} else if (zone == 7) {

a[0] = x1;

a[1] = -y1;

a[2] = x2;

a[3] = -y2;

}

return a;

}

static int[] convertToOriginalZone(int x, int y, int zone) {

int a[] = new int[2];

if (zone == 0) {

a[0] = x;

a[1] = y;

} else if (zone == 1) {

a[0] = y;

a[1] = x;

} else if (zone == 2) {

a[0] = -y;

a[1] = x;

} else if (zone == 3) {

a[0] = -x;

a[1] = y;

} else if (zone == 4) {

a[0] = -x;

a[1] = -y;

} else if (zone == 5) {

a[0] = -y;

a[1] = -x;

} else if (zone == 6) {

a[0] = y;

a[1] = -x;

} else if (zone == 7) {

a[0] = x;

a[1] = -y;

}

return a;

}

public static void main(String[] args) {

//getting the capabilities object of GL2 profile

final GLProfile profile = GLProfile.get(GLProfile.GL2);

GLCapabilities capabilities = new GLCapabilities(profile);

// The canvas

final GLCanvas glcanvas = new GLCanvas(capabilities);

Assignment3 l = new Assignment3();

glcanvas.addGLEventListener(l);

glcanvas.setSize(400, 400);

//creating frame

final JFrame frame = new JFrame ("Last two digits of my ID");

//adding canvas to frame

frame.getContentPane().add(glcanvas);

frame.setSize(frame.getContentPane().getPreferredSize());

frame.setVisible(true);

}//end of main

}//end of classimport javax.media.opengl.GL2;