package demo;

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 java.lang.Math;

import javax.swing.JFrame;

import java.util.Scanner;

class ThirdGLEventListener implements GLEventListener {

/\*\*

\* Interface to the GLU library.

\*/

private GLU glu;

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

static GLCapabilities capabilities = new GLCapabilities(profile);

// The canvas

static GLCanvas glcanvas = new GLCanvas(capabilities);

/\*\*

\* Take care of initialization here.

\*/

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(-250, -150, 250, 150);

gl.glMatrixMode(GL2.GL\_PROJECTION);

gl.glLoadIdentity();

glu.gluOrtho2D(-250.0, 250.0, -150.0, 150.0);

}

/\*\*

\* Take care of drawing here.

\*/

float zone = 0;

public void display(GLAutoDrawable drawable) {

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

// Points for 4

findZone(-25,-6,-25,85);

convertToZone0(gl,-25,-6,-25,85,zone);

findZone(-25,85,-90,30);

convertToZone0(gl,-25,85,-90,30,zone);

findZone(-90,30,-25,30);

convertToZone0(gl,-90,30,-25,30,zone);

// findZone(-80,40,-40,40);

//convertToZone0(gl,-80,40,-40,40,zone);

// Points for 3

findZone(0,80,80,80);

convertToZone0(gl,0,80,80,80,zone);

findZone(80,80,80,40);

convertToZone0(gl,80,80,80,40,zone);

findZone(20,40,80,40);

convertToZone0(gl,20,40,80,40,zone);

findZone(80,40,80,0);

convertToZone0(gl,80,40,80,0,zone);

findZone(0,0,80,0);

convertToZone0(gl,0,0,80,0,zone);

}

public void findZone(float x0, float y0, float x1, float y1) {

float dx = x1-x0;

float dy = y1-y0;

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

if (dx >= 0 && dy >= 0) {

zone = 0;

}

else if (dx < 0 && dy >= 0) {

zone = 3;

}

else if (dx < 0 && dy < 0) {

zone = 4;

}

else if (dx >= 0 && dy < 0) {

zone = 7;

}

} else {

if (dx >= 0 && dy >= 0) {

zone = 1;

}

else if (dx < 0 && dy >= 0) {

zone = 2;

}

else if (dx < 0 && dy < 0) {

zone = 5;

}

else if (dx >= 0 && dy < 0) {

zone = 6;

}

}

}

public void convertToZone0(GL2 gl, float x0, float y0, float x1, float y1, float zone) {

float a, b, c, d;

if(zone==0) {

a = x0;

b = y0;

c = x1;

d = y1;

}

if (zone==1) {

a = y0;

b = x0;

c = y1;

d = x1;

}

else if (zone==2) {

a = y0;

b = -x0;

c = y1;

d = -x1;

}

else if (zone==3) {

a = -x0;

b = y0;

c = -x1;

d = y1;

} else if (zone==4) {

a = -x0;

b = -y0;

c = -x1;

d = -y1;

}

else if (zone == 5) {

a = -y0;

b = -x0;

c = -y1;

d = -x1;

}

else if (zone==6) {

a = -y0;

b = x0;

c = -y1;

d = x1;

}

else {

a = x0;

b = -y0;

c = x1;

d = -y1;

}

mid\_Point\_Line(gl,a,b,c,d);

}

public void mid\_Point\_Line(GL2 gl, float x0, float y0, float x1, float y1) {

float dx = x1 - x0;

float dy = y1 - y0;

float d = 2 \* dy - dx;

float incrNE = 2 \* (dy - dx);

float incrE = 2 \* dy;

float x = x0, y = y0;

while (x < x1) {

originalZone(gl, x, y);

if (d <= 0) {

d = d + incrE;

x = x+1;

}

else {

d = d + incrNE;

x=x+1;

y=y+1;

}

}

}

public void originalZone(GL2 gl, float p, float q) {

float x1, y1;

gl.glPointSize(3.5f);

gl.glColor3d(1, 0, 1);

gl.glBegin(GL2.GL\_POINTS);

if (zone==1) {

x1 = q;

y1 = p;

}

else if (zone==2) {

x1 = -q;

y1 = p;

}

else if (zone==3) {

x1 = -q;

y1 = p;

}

else if (zone==4) {

x1 = -p;

y1 = -q;

}

else if (zone==5) {

x1 = -q;

y1 = -p;

}

else if (zone==6) {

x1 = q;

y1 = -p;

}

else {

x1 = p;

y1 = -q;

}

gl.glVertex2f(x1, y1);

gl.glEnd();

}

public void reshape(GLAutoDrawable drawable, int x, int y, int width,

int height) {

}

public void displayChanged(GLAutoDrawable drawable,

boolean modeChanged, boolean deviceChanged) {

}

public void dispose(GLAutoDrawable arg0)

{

}

}

public class DEMO

{

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);

ThirdGLEventListener b=new ThirdGLEventListener();

glcanvas.addGLEventListener(b);

glcanvas.setSize(400, 400);

//creating frame

final JFrame frame=new JFrame("Basic frame");

//adding canvas to frame

frame.add(glcanvas);

frame.setSize(1200,680);

frame.setVisible(true);

}

}