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();

Midpointcirlce(gl,50,"center");

Midpointcirlce(gl,25,"up");

Midpointcirlce(gl,25,"right");

Midpointcirlce(gl,25,"left");

Midpointcirlce(gl,25,"down");

}

public void Midpointcirlce(GL2 gl,int radius, String g){

int x, y, d;

d=1-radius;

x=0;

y=radius;

eightway(gl,x,y,g);

while(x<y){

if(d<0){

d=d+2\*x+3;

x=x+1;

}

else{

d=d+2\*x-2\*y+5;

x=x+1;

y=y-1;

}

eightway(gl,x,y,g);

}

}

public void eightway(GL2 gl,int x,int y,String g){

gl.glPointSize(3.5f);

gl.glColor3d(1, 0, 1);

gl.glBegin(GL2.GL\_POINTS);

int nx;

int ny;

if(g=="center") {

gl.glColor3d(1, 0, 1);

//for zone 1

nx=x;

ny=y;

gl.glVertex2f(nx, ny);

//for zone 0

nx=y;

ny=x;

gl.glVertex2f(nx, ny);

//for zone 2

nx=-x;

ny=y;

gl.glVertex2f(nx, ny);

//for zone 3

nx=-y;

ny=x;

gl.glVertex2f(nx, ny);

//for zone 4

nx=-y;

ny=-x;

gl.glVertex2f(nx, ny);

//for zone 5

nx=-x;

ny=-y;

gl.glVertex2f(nx, ny);

//for zone 6

nx=x;

ny=-y;

gl.glVertex2f(nx, ny);

//for zone 7

nx=y;

ny=-x;

gl.glVertex2f(nx, ny);

}

if(g=="up"){

gl.glColor3d(1, 1, 0);

//for zone 1

nx=x;

ny=y+25;

gl.glVertex2f(nx, ny);

//for zone 0

nx=y;

ny=x+25;

gl.glVertex2f(nx, ny);

//for zone 2

nx=-x;

ny=y+25;

gl.glVertex2f(nx, ny);

//for zone 3

nx=-y;

ny=x+25;

gl.glVertex2f(nx, ny);

//for zone 4

nx=-y;

ny=-x+25;

gl.glVertex2f(nx, ny);

//for zone 5

nx=-x;

ny=-y+25;

gl.glVertex2f(nx, ny);

//for zone 6

nx=x;

ny=-y+25;

gl.glVertex2f(nx, ny);

//for zone 7

nx=y;

ny=-x+25;

gl.glVertex2f(nx, ny);

}

if(g=="right"){

//for zone 1

nx=x+25;

ny=y;

gl.glVertex2f(nx, ny);

//for zone 0

nx=y+25;

ny=x;

gl.glVertex2f(nx, ny);

//for zone 2

nx=-x+25;

ny=y;

gl.glVertex2f(nx, ny);

//for zone 3

nx=-y+25;

ny=x;

gl.glVertex2f(nx, ny);

//for zone 4

nx=-y+25;

ny=-x;

gl.glVertex2f(nx, ny);

//for zone 5

nx=-x+25;

ny=-y;

gl.glVertex2f(nx, ny);

//for zone 6

nx=x+25;

ny=-y;

gl.glVertex2f(nx, ny);

//for zone 7

nx=y+25;

ny=-x;

gl.glVertex2f(nx, ny);

}

if(g=="left"){

//for zone 1

nx=x-25;

ny=y;

gl.glVertex2f(nx, ny);

//for zone 0

nx=y-25;

ny=x;

gl.glVertex2f(nx, ny);

//for zone 2

nx=-x-25;

ny=y;

gl.glVertex2f(nx, ny);

//for zone 3

nx=-y-25;

ny=x;

gl.glVertex2f(nx, ny);

//for zone 4

nx=-y-25;

ny=-x;

gl.glVertex2f(nx, ny);

//for zone 5

nx=-x-25;

ny=-y;

gl.glVertex2f(nx, ny);

//for zone 6

nx=x-25;

ny=-y;

gl.glVertex2f(nx, ny);

//for zone 7

nx=y-25;

ny=-x;

gl.glVertex2f(nx, ny);

}

if(g=="down"){

gl.glColor3d(1, 1, 1);

//for zone 1

nx=x;

ny=y-25;

gl.glVertex2f(nx, ny);

//for zone 0

nx=y;

ny=x-25;

gl.glVertex2f(nx, ny);

//for zone 2

nx=-x;

ny=y-25;

gl.glVertex2f(nx, ny);

//for zone 3

nx=-y;

ny=x-25;

gl.glVertex2f(nx, ny);

//for zone 4

nx=-y;

ny=-x-25;

gl.glVertex2f(nx, ny);

//for zone 5

nx=-x;

ny=-y-25;

gl.glVertex2f(nx, ny);

//for zone 6

nx=x;

ny=-y-25;

gl.glVertex2f(nx, ny);

//for zone 7

nx=y;

ny=-x-25;

gl.glVertex2f(nx, ny);

}

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);

}

}