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;

//Md. Shamiul Islam

//ID: 17301108

//section

public class CSE423Lab2Task01 implements GLEventListener {

private GLU glu;

public void display(GLAutoDrawable drawable) {

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

gl.glClear(GL2.GL\_COLOR\_BUFFER\_BIT);

//my ID is 17301108, i have to print 08

//for 0

drawMidpointLine(gl, -70, 70, -20, 70); //top

drawMidpointLine(gl, -70, 0, -70, 35); //lower left

drawMidpointLine(gl, -70, 35, -70, 70); // upper left

drawMidpointLine(gl, -20, 0, -20, 35); // lower right

drawMidpointLine(gl, -20, 35, -20, 70); // upper right

drawMidpointLine(gl, -70, 0 , -20, 0); //bottom

//for 8

drawMidpointLine(gl, 0, 70, 50, 70); //top

drawMidpointLine(gl, 0, 0, 0, 35); //lower left

drawMidpointLine(gl, 0, 35, 0, 70); //upper left

drawMidpointLine(gl, 50, 0, 50, 35); //lower right

drawMidpointLine(gl, 50, 35, 50, 70); //upper right

drawMidpointLine(gl, 0, 35, 50, 35); //middle

drawMidpointLine(gl, 0, 0, 50, 0); //bottom

}

public void dispose(GLAutoDrawable arg0) {

//method body

}

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);

}

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

// method body

}

public int findZone(int dx, int dy) {

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

dx = Math.abs(dx);

dy = Math.abs(dy);

if (dx >= dy) {

return 0;

} else {

return 1;

}

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

dx = Math.abs(dx);

dy = Math.abs(dy);

if (dx >= dy) {

return 3;

} else {

return 2;

}

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

dx = Math.abs(dx);

dy = Math.abs(dy);

if (dx >= dy) {

return 4;

} else {

return 5;

}

} else {

dx = Math.abs(dx);

dy = Math.abs(dy);

if (dx >= dy) {

return 7;

} else {

return 6;

}

}

}

public int convertToZoneZeroX(int x, int y, int zone) {

switch (zone) {

case 1:

return y;

case 2:

return y;

case 3:

return -x;

case 4:

return -x;

case 5:

return -y;

case 6:

return -y;

default:

return x;

}

}

public int convertToZoneZeroY(int x, int y, int zone) {

switch (zone) {

case 1:

return x;

case 2:

return -x;

case 4:

return -y;

case 5:

return -x;

case 6:

return x;

case 7:

return -y;

default:

return y;

}

}

public void OriginalZone(GL2 gl, int x, int y, int zone){

switch (zone) {

case 1:

gl.glVertex2d(y,x);

return;

case 2:

gl.glVertex2d(-y,x);

return;

case 3:

gl.glVertex2d(-x,y);

return;

case 4:

gl.glVertex2d(-x,-y);

return;

case 5:

gl.glVertex2d(-y,-x);

return;

case 6:

gl.glVertex2d(y,-x);

return;

case 7:

gl.glVertex2d(x,-y);

return;

default:

gl.glVertex2d(x,y);

}

}

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

gl.glPointSize(5.0f);

gl.glColor3d(1, 0, 0);

gl.glBegin(GL2.GL\_POINTS);

int dx = (x2 - x1);

int dy = (y2 - y1);

int zone = findZone(dx, dy);

int ddx, ddy, xx1, xx2, yy1, yy2, x, y;

xx1 = convertToZoneZeroX(x1, y1, zone);

yy1 = convertToZoneZeroY(x1, y1, zone);

xx2 = convertToZoneZeroX(x2, y2, zone);

yy2 = convertToZoneZeroY(x2, y2, zone);

ddx = (xx2 - xx1);

ddy = (yy2 - yy1);

int D = (2 \* ddy) - ddx;

int NE = 2 \* (ddy - ddx);

int E = 2 \* ddy;

x = xx1;

y = yy1;

while (x <= xx2) {

OriginalZone(gl,x,y,zone);

x++;

if (D > 0) {

y++;

D = D + NE;

} else {

D = D + E;

}

}

gl.glEnd();

}

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);

CSE423Lab2Task01 t = new CSE423Lab2Task01();

glcanvas.addGLEventListener(t);

glcanvas.setSize(500, 500);

//creating frame

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

//adding canvas to frame

frame.getContentPane().add(glcanvas);

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

frame.setVisible(true);

}

}