package Lab3;

import javax.swing.JFrame;

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;

class FirstGLEventListener implements GLEventListener {

/\*\*

\* Interface to the GLU library.

\*/

private GLU glu;

/\*\*

\* 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, 1.0f);

gl.glViewport(-10, -10, 10, 10);

gl.glMatrixMode(GL2.GL\_PROJECTION);

gl.glLoadIdentity();

glu.gluOrtho2D(-10.0, 10.0, -10.0, 10.0);

}

/\*\*

\* Take care of drawing here.

\*/

public void display(GLAutoDrawable drawable) {

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

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

// for drawing individual pixels

gl.glPointSize(5); //increase pixel size

gl.glBegin(GL2.GL\_POINTS);

gl.glColor3f(1, 0.5f, 0); //color orange

//printing number 33; make sure values of x and y are within viewport range in init()

MidpointLine(gl, -7, 5, -2, 5);

MidpointLine(gl, -2, 0, -2, 5);

MidpointLine(gl, -7, 0, -2, 0);

MidpointLine(gl, -2, -5, -2, 0);

MidpointLine(gl, -7, -5, -2, -5);

gl.glColor3f(1,1,0); //color yellow

MidpointLine(gl, 2, 5, 7, 5);

MidpointLine(gl, 7, 0, 7, 5);

MidpointLine(gl, 2, 0, 7, 0);

MidpointLine(gl, 7, -5, 7, 0);

MidpointLine(gl, 2, -5, 7, -5);

gl.glEnd();

}

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

float dx = x2 - x1;

float dy = y2 - y1;

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

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

Midpoint(gl, x1, y1, x2, y2, 0); //zone 0

}

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

Midpoint(gl, -x1, y1, -x2, y2, 3); //zone 3

}

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

Midpoint(gl, -x1, -y1, -x2, -y2, 4); //zone 4

}

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

Midpoint(gl, x1, -y1, x2, -y2, 7); //zone 7

}

}

else {

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

Midpoint(gl, y1, x1, y2, x2, 1); //zone 1

}

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

Midpoint(gl, y1, -x1, y2, -x2, 2); //zone 2

}

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

Midpoint(gl, -y1, -x1, -y2, -x2, 5); //zone 5

}

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

Midpoint(gl, -y1, x1, -y2, x2, 6); //zone 6

}

}

}

public static void Midpoint(GL2 gl, float x1, float y1, float x2, float y2, int zone) {

float dx, dy, incrE, incrNE, d, x, y;

if(x2 < x1) {

float temp = x1;

x1 = x2;

x2 = temp;

temp = y1;

y1 = y2;

y2 = temp;

}

dx = x2 - x1;

dy = y2 - y1;

d = (2 \* dy) - dx;

incrE = 2 \* dy;

incrNE = 2 \* (dy - dx);

x = x1;

y = y1;

WritePixel(gl, x, y, zone);

while(x < x2) {

if(d <= 0) {

// choose E

d = d + incrE;

x = (float) (x + 0.001);

}

else {

//choose NE

d = d + incrNE;

x = (float) (x + 0.001);

y = (float) (y + 0.001);

}

WritePixel(gl, x, y, zone);

}

}

public static void WritePixel(GL2 gl, float x, float y, int zone) {

if(zone == 0) {

gl.glVertex2f(x, y);

}

else if(zone == 1) {

gl.glVertex2f(y, x);

}

else if(zone == 2) {

gl.glVertex2f(-y, x);

}

else if(zone == 3) {

gl.glVertex2f(-x, y);

}

else if(zone == 4) {

gl.glVertex2f(-x, -y);

}

else if(zone == 5) {

gl.glVertex2f(-y, -x);

}

else if(zone == 6) {

gl.glVertex2f(y, -x);

}

else {

gl.glVertex2f(x, -y);

}

}

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 Rafsan\_18301033\_Assign03 {

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);

FirstGLEventListener b = new FirstGLEventListener();

glcanvas.addGLEventListener(b);

glcanvas.setSize(400, 400);

// creating frame

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

// adding canvas to frame

frame.add(glcanvas);

frame.setSize(640, 480);

frame.setVisible(true);

}

}