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 MdAbdulKahharSiddikiShan\_18301221\_Assignment03 implements GLEventListener{

private GLU glu;

@Override

public void display(GLAutoDrawable drawable) {

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

midpointLine(gl,-70.0,45.0,20.0,45.0);

midpointLine(gl,20.0,0.0,20.0,45.0);

midpointLine(gl,-70.0,0.0,20.0,0.0);

midpointLine(gl,-70.0,-45.0,-70.0,0.0);

midpointLine(gl,-70.0,-45.0,20.0,-45.0);

midpointLine(gl,50.0,-45.0,50.0,45.0);

}

@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, double x1, double y1, double x2, double y2) {

gl.glPointSize(3.0f);

gl.glColor3d(1, 0, 0);

//write your own code

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

// System.out.println("Zone: "+zone);

double [] array = convertToZone0(x1, y1, zone);

x1 = array[0];

y1 = array[1];

array = convertToZone0(x2, y2, zone);

x2 = array[0];

y2 = array[1];

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

dx = x2 - x1;

dy = y2 - y1;

d = 2 \* dy - dx;

incrE = 2 \* dy;

incrNE = 2 \* (dy - dx);

x = x1;

y = y1;

gl.glBegin(GL2.GL\_POINTS);

while (x <x2) {

if (d <= 0) {// to choose E

d = d + incrE;

x = x + 1.0;

}

else {// to choose NE

gl.glVertex2d(x, y);

d = d + incrNE;

x = x + 1.0;

y = y + 1.0;

}

double [] originalZonePoints = convertToOriginalZone(x, y, zone);

double xOriginal = originalZonePoints[0];

double yOriginal = originalZonePoints[1];

gl.glVertex2d(xOriginal, yOriginal);

}// end of while loop

gl.glEnd();

}

public int findZone(double x1, double y1, double x2, double y2) {

double dx = x2 - x1;

double dy = y2 - y1;

int zone;

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

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

zone = 6;

}

return zone;

}

public double [] convertToZone0(double x1, double y1, int zone) {

double [] result = new double[2];

if(zone==0) {

result[0] = x1;

result[1] = y1;

}

else if(zone==1) {

result[0] = y1;

result[1] = x1;

}

else if(zone==2) {

result[0] = y1;

result[1] = -x1;

}

else if(zone==3) {

result[0] = -x1;

result[1] = y1;

}

else if(zone==4) {

result[0] = -x1;

result[1] = -y1;

}

else if(zone==5) {

result[0] = -y1;

result[1] = -x1;

}

else if(zone==6) {

result[0] = -y1;

result[1] = x1;

}

else {

result[0] = x1;

result[1] = -y1;

}

return result;

}//end of convertToZone0

public double [] convertToOriginalZone(double x1, double y1, int zone){

double [] result = new double[2];

if(zone==0) {

result[0] = x1;

result[1] = y1;

}

else if(zone==1) {

result[0] = y1;

result[1] = x1;

}

else if(zone==2) {

result[0] = -y1;

result[1] = x1;

}

else if(zone==3) {

result[0] = -x1;

result[1] = y1;

}

else if(zone==4) {

result[0] = -x1;

result[1] = -y1;

}

else if(zone==5) {

result[0] = -y1;

result[1] = -x1;

}

else if(zone==6) {

result[0] = y1;

result[1] = -x1;

}

else {

result[0] = x1;

result[1] = -y1;

}

//System.out.println("Original zone: "+result[0]+" "+result[1]);

return result;

}// end of convertToOriginalZone

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);

MdAbdulKahharSiddikiShan\_18301221\_Assignment03 l = new MdAbdulKahharSiddikiShan\_18301221\_Assignment03();

glcanvas.addGLEventListener(l);

glcanvas.setSize(400, 400);

//creating frame

final JFrame frame = new JFrame ("straight Line");

//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;