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;

public class NewClass implements GLEventListener {

private GLU glu;

@Override

public void display(GLAutoDrawable drawable) {

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

DrawMPL(gl, -45, 0, 0, 45);

DrawMPL(gl, 45, 0, 0, 45);

DrawMPL(gl, -45, 0, 45, 0);

DrawMPL(gl, -30, -45,-30, 0);

DrawMPL(gl, 30, -45,30, 0);

DrawMPL(gl, -30, -45,30, -45);

DrawMPL(gl, -10, -45,-10, -15);

DrawMPL(gl, 10, -45,10, -15);

DrawMPL(gl, -10, -15,10, -15);

//

//

gl.glEnd();

}

@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, -100,100, 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

}

static int dx, dy, b, zone;

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

gl.glPointSize(3.0f);

gl.glColor3d(1, 0, 0);

gl.glBegin(GL2.GL\_POINTS);

findZone(x1, y1, x2, y2);

System.out.println(zone);

int new\_x1, new\_y1, new\_x2, new\_y2;

new\_x1 = convertX(x1, y1);

new\_y1 = convertY(x1, y1);

new\_x2 = convertX(x2, y2);

new\_y2 = convertY(x2, y2);

System.out.println(new\_x2 + " " + new\_y2);

x1 = new\_x1;

x2 = new\_x2;

y1 = new\_y1;

y2 = new\_y2;

int dx = x2 - x1;

int dy = y2 - y1;

int dinit = (2 \* dy) - dx;

int dE = 2 \* dy;

int dNE = 2 \* (dy - dx);

int y = y1;

int x = x1;

while (x <= x2) {

gl.glVertex3f(convertX\_to\_m(x, y), convertY\_to\_m(x, y), 0);

x++;

if (dinit > 0) {

y++;

dinit = dinit + dNE;

} else {

dinit = dinit + dE;

}

}

}

static int findZone(int x1, int y1, int x2, int y2) {

zone = 0;

dx = x2 - x1;

dy = y2 - y1;

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;

}

return zone;

}

static int convertX(int x, int y) {

int x\_main;

if (zone == 1) {

x\_main = y;

} else if (zone == 2) {

x\_main = y;

} else if (zone == 3) {

x\_main = -x;

} else if (zone == 4) {

x\_main = -x;

} else if (zone == 5) {

x\_main = -y;

} else if (zone == 6) {

x\_main = -y;

} else if (zone == 7) {

x\_main = x;

} else {

x\_main = x;

}

return x\_main;

}

static int convertX\_to\_m(int x, int y) {

if (zone == 1) {

x = y;

} else if (zone == 2) {

x = -y;

} else if (zone == 3) {

x = -x;

} else if (zone == 4) {

x = -x;

} else if (zone == 5) {

x = -y;

} else if (zone == 6) {

x = y;

} else if (zone == 7) {

x = x;

} else {

x = x;

}

return x;

}

static int convertY(int x, int y) {

int y\_main = y;

if (zone == 1) {

y\_main = x;

} else if (zone == 2) {

y\_main = -x;

} else if (zone == 3) {

y\_main = y;

} else if (zone == 4) {

y\_main = -y;

} else if (zone == 5) {

y\_main = -x;

} else if (zone == 6) {

y\_main = x;

} else if (zone == 7) {

y\_main = -y;

}

return y\_main;

}

static int convertY\_to\_m(int x, int y) {

if (zone == 1) {

y = x;

} else if (zone == 2) {

y = x;

} else if (zone == 3) {

y = y;

} else if (zone == 4) {

y = -y;

} else if (zone == 5) {

y = -x;

} else if (zone == 6) {

y = -x;

} else if (zone == 7) {

y = -y;

} else {

y = y;

}

return y;

}

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);

NewClass l = new NewClass();

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;