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 java.util.Random;

import javax.swing.JFrame;

public class Assignment1P1 implements GLEventListener {

public float randNumb() {

int max = 1;

int min = -1;

Random r = new Random();

return r.nextFloat() \* (max - min) + min;

}

public void drawPixel(float x, float y, GLAutoDrawable drawable) {

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

gl.glBegin(GL2.GL\_POINTS);// static field

gl.glVertex3f(x, y, 0);

gl.glEnd();

}

@Override

public void display(GLAutoDrawable drawable) {

for (int i = 0; i < 100; i++) {

float x = randNumb();

float y = randNumb();

System.out.println(i);

drawPixel(x, y, drawable);

}

}

@Override

public void dispose(GLAutoDrawable arg0) {

// method body

}

@Override

public void init(GLAutoDrawable arg0) {

// method body

}

@Override

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

// method body

}

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);

Assignment1P1 l = new Assignment1P1();

glcanvas.addGLEventListener(l);

glcanvas.setSize(400, 400);

// creating frame

final JFrame frame = new JFrame("Showing randomly generated points");

// 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;