

Questão 3)

```
import java.awt.*;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import javax.media.opengl.*;
import java.util.Random;
import com.sun.opengl.util.Animator;

@SuppressWarnings("serial")
public class QuadradosGirantes extends Frame implements GLEventListener {
    static int HEIGHT = 800, WIDTH = 800;
    static GL gl; // interface para o OpenGL
    static GLCanvas canvas; // uma frame desenhável
    static GLCapabilities capabilities;
    Random rand;
    static Animator animator;

    public QuadradosGirantes() {
        capabilities = new GLCapabilities();
        canvas = new GLCanvas();
        canvas.addGLEventListener(this);
        add(canvas, BorderLayout.CENTER);
        gl = canvas.getGL();

        rand = new Random();
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                animator.stop();
                System.exit(0);
            }
        });
    }

    public void init(GLAutoDrawable drawable) {
        animator = new Animator(canvas);
        animator.start(); // thread iniciado
    }

    public void reshape(GLAutoDrawable drawable, int x, int y, int width, int height) {
        WIDTH = width;
        HEIGHT = height;
        gl.glMatrixMode(GL.GL_PROJECTION);
        gl.glLoadIdentity();
        gl.glOrtho(0, width, 0, height, -1.0, 1.0);
        gl.glMatrixMode(GL.GL_MODELVIEW);
        gl.glLoadIdentity();
        gl.glViewport(0, 0, width, height);
    }
}
```

```

void drawSquare() {
    gl.glBegin(GL.GL_POLYGON);
        gl.glVertex3f(100.0f, 0.0f, 0.0f);
        gl.glVertex3f(0.0f, 100.0f, 0.0f);
        gl.glVertex3f(-100.0f, 0.0f, 0.0f);
        gl.glVertex3f(0.0f, -100.0f, 0.0f);
    gl.glEnd();
}

public void display(GLAutoDrawable drawable) {
    gl.glClear (GL.GL_COLOR_BUFFER_BIT|GL.GL_DEPTH_BUFFER_BIT);
    gl.glLoadIdentity();
    gl.glTranslated(100.0f, 150.0f, 0.0f);
    gl.glPushMatrix();
        for (int i = 0; i <=30; ++i)
        {
            gl.glColor3f(rand.nextFloat(), rand.nextFloat(), rand.nextFloat());
            gl.glRotatef(rand.nextFloat()*100.0f, 0.0f, 0.0f, 1.0f);
            gl.glTranslated(rand.nextFloat()*200.0f, rand.nextFloat()*300.0f, 0.0f);
            drawSquare();
        }
    gl.glPopMatrix();
    // torna refrescamento de desenho mais lento
    try {
        Thread.sleep(200);
    } catch (Exception ignore) {
    }
}

public void displayChanged(GLAutoDrawable arg0, boolean arg1, boolean arg2) {
    // TODO
}

public static void main(String[] args) {
    QuadradosGirantes pfolio = new QuadradosGirantes();
    pfolio.setTitle("Pfolio");
    pfolio.setSize(WIDTH, HEIGHT);
    pfolio.setVisible(true);
}
}

```