

```

import com.sun.j3d.utils.geometry.Sphere;
import com.sun.j3d.utils.geometry.Text2D;
import com.sun.j3d.utils.universe.SimpleUniverse;
import com.sun.j3d.utils.universe.Viewer;
import com.sun.j3d.utils.universe.ViewingPlatform;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.GraphicsConfiguration;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import javax.media.j3d.AmbientLight;
import javax.media.j3d.Appearance;
import javax.media.j3d.BoundingSphere;
import javax.media.j3d.BranchGroup;
import javax.media.j3d.Canvas3D;
import javax.media.j3d.ColoringAttributes;
import javax.media.j3d.DirectionallLight;
import javax.media.j3d.Group;
import javax.media.j3d.LineArray;
import javax.media.j3d.Shape3D;
import javax.media.j3d.Transform3D;
import javax.media.j3d.TransformGroup;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.Timer;
import javax.swing.WindowConstants;
import javax.vecmath.Color3f;
import javax.vecmath.Point3d;
import javax.vecmath.Point3f;
import javax.vecmath.Vector3d;
import javax.vecmath.Vector3f;

public class FiboFrame extends JFrame {

    private int currentDotId = 0;
    private List<Dot> dots = new ArrayList<Dot>();
    private BranchGroup root;
    private Map<Integer, ColoredLineArray> lineArrayMap = new HashMap<Integer, ColoredLineArray>();
    int[] lineKeys = { 13, 21 };
    Color[] lineColors = { Color.YELLOW, Color.PINK, Color.GREEN, Color.ORANGE };

    private class ColoredLineArray {

        private int currentLineId;
        private LineArray lineArray;

        public ColoredLineArray(Color color) {
            lineArray = new LineArray(2000, LineArray.COORDINATES);
            lineArray.setCapability(LineArray.ALLOW_COORDINATE_WRITE);
            Appearance aLine = new Appearance();
            ColoringAttributes ca = new ColoringAttributes(new Color3f(color),
ColoringAttributes.SHADE_FLAT);
            aLine.setColoringAttributes(ca);
            Shape3D sLine = new Shape3D(lineArray, aLine);
            BranchGroup bgLine = new BranchGroup();
            bgLine.addChild(sLine);
            root.addChild(bgLine);
        }

        public int getNextLineId() {
            int temp = currentLineId;
            currentLineId += 2;
            return temp;
        }

        public void setCoordinates(int lineId, Point3f p1, Point3f p2) {
            lineArray.setCoordinates(lineId, new Point3f[] { p1, p2 });
        }
    }

    private class Dot extends BranchGroup {

        private int id;
        private float radiusStep = 0.02f;

```

```

private TransformGroup main;
private Transform3D tRadius;
private Map<Integer, Integer> lineIdMap = new HashMap<Integer, Integer>();
private TransformGroup tgHover;
private Text2D text;

public int getId() {
    return id;
}

public Dot(int id) {
    this.id = id;
    main = new TransformGroup();
    main.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
    main.addChild(new Sphere(0.2f));
    addChild(main);
    tRadius = new Transform3D();
    Transform3D tRotation = new Transform3D();
    double angle = (id - 1) * 2.399963;
    tRotation.rotZ(angle);
    // tRotation.rotZ(id * Math.sqrt(2) * 180 / Math.PI);
    tRadius.mul(tRotation);
    for(int lineKey : lineKeys) {
        // if(id == 0 || id % lineKey == 0) {
            tgHover = new TransformGroup();
            tgHover.setCapability(TransformGroup.ALLOW_CHILDREN_WRITE);
            // bgHover = new BranchGroup();
            // bgHover.setCapability(BranchGroup.ALLOW_DETACH);
            // tgHover.addChild(bgHover);
            // text = new Text2D(id + "", new Color3f(Color.WHITE), "Arial", 24, Font.BOLD);
            // text.getAppearance().setCapability(Appearance.ALLOW_TEXTURE_WRITE);
            // bgHover.addChild(text);
            Transform3D tHover = new Transform3D();
            // tHover.rotZ(0 - angle);
            Transform3D tHoverUp = new Transform3D();
            tHoverUp.setTranslation(new Vector3f(-0.1f, -0.1f, 0.5f));
            tHover.mul(tHoverUp);
            Transform3D tHoverSize = new Transform3D();
            tHoverSize.set(2.5);
            tHover.mul(tHoverSize);
            tgHover.setTransform(tHover);
            main.addChild(tgHover);
        // }
    }
    updateRadius();
}

private void updateCoordinates() {
    for(int i = 0; i < lineKeys.length; i++) {
        int lineKey = lineKeys[i];
        int prevId = id - lineKey;
        if(prevId < 0) {
            continue;
        }
        ColoredLineArray cla = lineArrayMap.get(i);
        if(cla == null) {
            cla = new ColoredLineArray(lineColors[i]);
            lineArrayMap.put(i, cla);
        }
        Integer lineId = lineIdMap.get(i);
        if(lineId == null) {
            lineId = cla.getNextLineId();
            lineIdMap.put(i, lineId);
        }
        Dot prevDot = dots.get(prevId);
        // cla.setCoordinates(lineId, prevDot.getPosition(), getPosition());
    }
}

public void updateRadius() {
    // Transform3D size = new Transform3D();
    // size.setScale(1.01);
    // tRadius.mul(size);
    Transform3D move = new Transform3D();
    move.setTranslation(new Vector3f(radiusStep, 0, 0));
    tRadius.mul(move);
    main.setTransform(tRadius);
    updateCoordinates();
    // bgHover.removeAllChildren();
    BranchGroup bgTemp = new BranchGroup();
    bgTemp.setCapability(BranchGroup.ALLOW_DETACH);
}

```

```

        text = new Text2D((currentDotId) + "", new Color3f(Color.WHITE), "Arial", 24, Font.BOLD);
        // bgTemp.addChild(text);
        if(tgHover.numChildren() > 0) {
            // tgHover.setChild(bgTemp, 0);
        } else {
            tgHover.addChild(bgTemp);
        }
    }

    public Point3f getPosition() {
        Point3f p = new Point3f();
        tRadius.transform(p);
        return p;
    }
}

public FiboFrame() {
    setLayout(new BorderLayout());
    GraphicsConfiguration config = SimpleUniverse.getPreferredConfiguration();
    Canvas3D canvas3d = new Canvas3D(config);
    add(canvas3d, BorderLayout.CENTER);
    //=====
    root = new BranchGroup();
    BoundingSphere bounds = new BoundingSphere(new Point3d(), 20f);
    AmbientLight lightA = new AmbientLight();
    lightA.setInfluencingBounds(bounds);
    root.addChild(lightA);
    DirectionalLight lightD1 = new DirectionalLight();
    lightD1.setInfluencingBounds(bounds);
    Vector3f direction1 = new Vector3f(1.0f, -1.0f, -1.0f);
    direction1.normalize();
    lightD1.setDirection(direction1);
    lightD1.setColor(new Color3f(0, 0, 1.0f));
    root.addChild(lightD1);
    root.setCapability(Group.ALLOW_CHILDREN_EXTEND);
    //=====
    Viewer v = new Viewer(canvas3d);
    ViewingPlatform vp = new ViewingPlatform();
    SimpleUniverse u = new SimpleUniverse(vp, v);
    u.addBranchGraph(root);
    vp.setNominalViewingTransform();
    TransformGroup tg = vp.getViewPlatformTransform();
    Transform3D t = new Transform3D();
    tg.getTransform(t);
    Vector3d position = new Vector3d();
    t.get(position);
    Vector3d offset = new Vector3d(0, 0, -20f);
    position.sub(offset);
    t.setTranslation(position);
    tg.setTransform(t);
    //=====
    final Timer timer = new Timer(100, new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            if(currentDotId > 300) {
                return;
            }
            Dot dot = new Dot(currentDotId);
            dots.add(dot);
            root.addChild(dot);
            for(Dot d : dots) {
                d.updateRadius();
            }
            currentDotId++;
        }
    });
    //=====
    JButton button = new JButton("Start");
    button.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            timer.start();
        }
    });
    add(button, BorderLayout.AFTER_LAST_LINE);
    //=====
    setTitle("Java3d Test");
    setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    setSize(500, 500);
    setVisible(true);
    //=====

```

```
}  
  
public static void main(String[] args) {  
    EventQueue.invokeLater(new Runnable() {  
        public void run() {  
            new FiboFrame();  
        }  
    });  
}  
  
}
```