```
/*
 * Shuttle.java
 *
 * More user-defined geometry
 *  (with GeometryInfo)
 * Make a star trek shuttle craft.
 *
 * written by mike slattery - oct 2003
 */

import java.applet.Applet;
import java.awt.*;
import com.sun.j3d.utils.applet.MainFrame;
import com.sun.j3d.utils.geometry.*;
import com.sun.j3d.utils.universe.*;
import javax.media.j3d.*;
import javax.vecmath.*;
import com.sun.j3d.utils.behaviors.keyboard.*;

public class Shuttle extends Applet {

     SimpleUniverse simpleU=null;

     public BranchGroup createSceneGraph() {
                // Create the root of the branch graph
                BranchGroup objRoot = new BranchGroup();

                // Ambient light is just around (sort of like a cloudy day)
                float dimmer = 0.3f;
                AmbientLight lightA = new AmbientLight(new Color3f(dimmer, dimmer, dim
mer));
                BoundingSphere bs = new BoundingSphere(new Point3d(0.0,0.0,0.0),1000.0
);
                lightA.setInfluencingBounds(bs);
                objRoot.addChild(lightA);

        // Directional light comes from somewhere, so only lights up
        // one side of things.
        DirectionalLight lightD = new DirectionalLight();
        lightD.setInfluencingBounds(bs);
        Vector3f direction = new Vector3f(-1.0f, -1.0f, -0.5f);
        direction.normalize();
        lightD.setDirection(direction);
        lightD.setColor(new Color3f(0.9f, 0.5f, 0.9f));
        objRoot.addChild(lightD);



        // Define shuttle-craft
        double[] sh_coords = {
          // top
          -1.0,0.3,-0.5,
          -1.0,0.3,0.5,
          1.0,0.3,0.5,
          1.2,0.3,0.0,
          1.0,0.3,-0.5,
          // starboard side
          -1.0,0.3,0.5,
          -1.0,-0.3,0.5,
          1.5,-0.3,0.5,
          1.0,0.3,0.5,
          // bottom
          -1.0,-0.3,0.5,
          -1.0,-0.3,-0.5,
          1.5,-0.3,-0.5,
          1.5,-0.3,0.5,
          // port side
            -1.0,0.3,-0.5,
                   1.0,0.3,-0.5,
            1.5,-0.3,-0.5,
          -1.0,-0.3,-0.5,
          /*  poorly folded front
          // front
          1.0,0.3,-0.5,
            1.2,0.3,0.0,
          1.0,0.3,0.5,
          1.5,-0.3,0.5,
          1.5,-0.3,-0.5,
          */
          // front (as triangles
            1.0,0.3,-0.5,
            1.2,0.3,0.0,
            1.5,-0.3,-0.5,
            1.2,0.3,0.0,
            1.5,-0.3,0.5,
             1.5,-0.3,-0.5,
            1.2,0.3,0.0,
            1.0,0.3,0.5,
            1.5,-0.3,0.5,
          // back
          -1.0,-0.3,-0.5,
          -1.0,-0.3,0.5,
          -1.0,0.3,0.5,
          -1.0,0.3,-0.5,
          // starboard wing (top)
          -0.8,0.0,0.5,
          -0.8,0.0,1.1,
          -0.5,0.0,1.1,
          0.0,0.0,0.5,
          // starboard wing (bottom)
           -0.8,0.0,0.5,
            0.0,0.0,0.5,
                        -0.5,0.0,1.1,
                        -0.8,0.0,1.1,
                  // port wing (top)
                     -0.8,0.0,-0.5,
                              0.0,0.0,-0.5,
                                          -0.5,0.0,-1.1,
                        -0.8,0.0,-1.1,
          // port wing (bottom)
                   -0.8,0.0,-0.5,
                   -0.8,0.0,-1.1,
                   -0.5,0.0,-1.1,
          0.0,0.0,-0.5,
        };

        int[] strip_cnt = {5, 4, 4, 4, 3, 3, 3, 4, 4, 4, 4, 4};

        GeometryInfo gi = new GeometryInfo(GeometryInfo.POLYGON_ARRAY);
        gi.setCoordinates(sh_coords);
        gi.setStripCounts(strip_cnt);
        // Need normals to apply lighting
        NormalGenerator ng = new NormalGenerator();
        ng.generateNormals(gi);
```

```
        //
        Shape3D shuttle = new Shape3D();
        shuttle.setGeometry(gi.getGeometryArray());
        //
        Color3f black = new Color3f(0.0f,0.0f,0.0f);
        Color3f white = new Color3f(1.0f,1.0f,1.0f);
         Appearance appB = new Appearance();
                                Color3f colorB = new Color3f(0.02f, 0.7f, 0.02f);
        appB.setMaterial(new Material(colorB, black, colorB, white, 80.0f));
        /*
        PolygonAttributes pA = new PolygonAttributes();
        pA.setCullFace(PolygonAttributes.CULL_NONE);
        appB.setPolygonAttributes(pA);

        */
        /*
          PolygonAttributes polyAttr = new PolygonAttributes();
                        polyAttr.setPolygonMode(PolygonAttributes.POLYGON_LINE);
                appB.setPolygonAttributes(polyAttr);
                */
        shuttle.setAppearance(appB);
        //
        objRoot.addChild(shuttle);


                return objRoot;
    } // end of CreateSceneGraph method of Shuttle

    public Shuttle() {
        setLayout(new BorderLayout());
        Canvas3D canvas3D = new Canvas3D(SimpleUniverse.getPreferredConfiguration());
        add("Center", canvas3D);

        // SimpleUniverse is a Convenience Utility class
        simpleU = new SimpleUniverse(canvas3D);

        BranchGroup scene = createSceneGraph();

                //...and, set up a KeyNavigator to let us move about
                //
                TransformGroup vpTrans = simpleU.getViewingPlatform().getViewPlatformT
ransform();

                KeyNavigatorBehavior keyNav = new KeyNavigatorBehavior(vpTrans);
                keyNav.setSchedulingBounds(new BoundingSphere(new Point3d(), 1000.0));
        scene.addChild(keyNav);

                scene.compile();

        simpleU.addBranchGraph(scene);
    }

    //  The following allows this to be run as an application
    //  as well as an applet

    public static void main(String[] args) {
        Frame frame = new MainFrame(new Shuttle(), 256, 256);
    } // end of main (method of Shuttle)

} // end of class Shuttle
```