



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики  
Кафедра програмного забезпечення комп'ютерних систем

**Лабораторна робота № 6**

з дисципліни “Математичні та алгоритмічні основи комп'ютерної графіки”  
на тему “Анімація тривимірних об'єктів”

Виконав  
студент III курсу  
групи КП-82

Новохацький Владислав  
*(прізвище, ім'я, по батькові)*

варіант № 15

Зарахована  
“ \_\_\_\_ ” “ \_\_\_\_ ” 20\_\_ р.  
викладачем

Шкурат Оксаною Сергіївною  
*(прізвище, ім'я, по батькові)*

## Варіант завдання

### Варіант 15:

#### Завдання

Виконати анімацію тривимірної сцени за варіантом.

5. Анімація вертольоту helicopter.obj. У вертольота повинні рухатися обидва гвинти, вертоліт повинен пересуватися по екрану.

#### Лістинг коду програми

```
package lab6;
import javax.vecmath.*;

import com.sun.j3d.utils.universe.*;
import javax.media.j3d.*;
import com.sun.j3d.utils.behaviors.vp.*;
import javax.swing.JFrame;
import com.sun.j3d.loaders.*;
import com.sun.j3d.loaders.objectfile.*;

import java.io.FileReader;
import java.io.IOException;
import java.util.Hashtable;
import java.util.Enumuration;

public class Main extends JFrame{
    public Canvas3D myCanvas3D;

    public Main() throws IOException {
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        myCanvas3D = new Canvas3D(SimpleUniverse.getPreferredConfiguration());
        SimpleUniverse simpUniv = new SimpleUniverse(myCanvas3D);

        simpUniv.getViewingPlatform().setNominalViewingTransform();
```

```

        createSceneGraph(simpUniv);
        addLight(simpUniv);

        OrbitBehavior ob = new OrbitBehavior(myCanvas3D);
        ob.setSchedulingBounds(new BoundingSphere(new
Point3d(0.0,0.0,0.0),Double.MAX_VALUE));
        simpUniv.getViewingPlatform().setViewPlatformBehavior(ob);

        setSize(1280,720);
        getContentPane().add("Center", myCanvas3D);
        setVisible(true);
    }

    public void createSceneGraph(SimpleUniverse su) throws IOException {
        String name;
        BranchGroup helicopterBranchGroup = new BranchGroup();
        Background background = new Background(new
Color3f(45/255f,163/255f,252/255f));

        ObjectFile loader = new ObjectFile(ObjectFile.RESIZE);
        loader.setBasePath("D:\\graphic\\lab6\\");

        Scene heliScene = loader.load(new
FileReader("D:\\graphic\\lab6\\helicopter.obj"));

        Hashtable<?, ?> namedObjects = heliScene.getNamedObjects();
        Enumeration<?> enumer = namedObjects.keys();
        while (enumer.hasMoreElements()){
            name = (String) enumer.nextElement();
            System.out.println("Name: " + name);
        }

        BoundingSphere bounds = new BoundingSphere(new Point3d(0.0,0.0,0.0), 100.0);

        Transform3D heliObj = new Transform3D();
        TransformGroup heliTg = new TransformGroup(heliObj);

        Appearance heliApp = new Appearance();
        setToMyDefaultAppearance(heliApp,new Color3f(0f,23/255f,13/255f));

        Shape3D[] heli = new Shape3D[] { (Shape3D)namedObjects.get("decal"),
(Shape3D)namedObjects.get("alpha"), (Shape3D)namedObjects.get("missile_1"),
            (Shape3D)namedObjects.get("missile_gl"),
(Shape3D)namedObjects.get("main_"), (Shape3D)namedObjects.get("main_body_")};
        for (Shape3D shape:heli){
            shape.setAppearance(heliApp);
            heliTg.addChild(shape.cloneTree());
        }
    }

```

```
}
```

```
Shape3D bigProp = (Shape3D) namedObjects.get("big_propeller");
Shape3D smallProp = (Shape3D) namedObjects.get("small_propeller");
Shape3D glass = (Shape3D) namedObjects.get("glass");

Appearance propApp = new Appearance();
setToMyDefaultAppearance(propApp, new Color3f(0f, 0f, 0f));
bigProp.setAppearance(propApp);
smallProp.setAppearance(propApp);

Appearance glassApp = new Appearance();
setToMyDefaultAppearance(glassApp, new Color3f(163/255f, 194/255f, 180/255f));
glass.setAppearance(glassApp);

TransformGroup bigPropTg = new TransformGroup();
bigPropTg.addChild(bigProp.cloneTree());

Transform3D bigPropRotationAxis = new Transform3D();
bigPropRotationAxis.set(new Vector3d(0, 0, -0.212));
bigPropRotationAxis.setRotation(new AxisAngle4d(0, -0.1, 0, Math.PI/2));

RotationInterpolator bigPropRotation = new RotationInterpolator(
    new Alpha(), bigPropTg,
    bigPropRotationAxis, 0f, (float) Math.PI*20);

bigPropRotation.setSchedulingBounds(bounds);
bigPropTg.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
bigPropTg.addChild(bigPropRotation);

heliTg.addChild(bigPropTg);

TransformGroup smallPropTg = new TransformGroup();
smallPropTg.addChild(smallProp.cloneTree());

Transform3D smallPropRotationAxis = new Transform3D();
smallPropRotationAxis.set(new Vector3d(0, 0.0585, 0.845));
smallPropRotationAxis.setRotation(new AxisAngle4d(0, 0, -0.1, Math.PI/2));

RotationInterpolator smallPropRotation = new RotationInterpolator(
    new Alpha(), smallPropTg,
    smallPropRotationAxis, 0f, (float) Math.PI*20);

smallPropRotation.setSchedulingBounds(bounds);
```

```

smallPropTg.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
smallPropTg.addChild(smallPropRotation);

heliTg.addChild(smallPropTg);

heliTg.addChild(glass.cloneTree());

Transform3D transform3D = new Transform3D();
transform3D.setTranslation(new Vector3f(5f,0f,0f));
TransformGroup transformGroup = new TransformGroup();
transformGroup.addChild(heliTg);
transformGroup.setTransform(transform3D);

TransformGroup bformGroup = new TransformGroup();
bformGroup.setCapability(
    TransformGroup.ALLOW_TRANSFORM_WRITE);

Transform3D rot = new Transform3D();
rot.rotZ(-Math.PI/160);

RotationInterpolator interpolator =
    new RotationInterpolator(new Alpha(-1,5000),bformGroup,rot,0.0f,
(float)Math.PI*2);

interpolator.setSchedulingBounds(bounds);

bformGroup.addChild(interpolator);
bformGroup.addChild(transformGroup);

helicopterBranchGroup.addChild(bformGroup);

background.setApplicationBounds(bounds);
helicopterBranchGroup.addChild(background);

helicopterBranchGroup.compile();
su.addBranchGraph(helicopterBranchGroup);
}

public void addLight(SimpleUniverse su){
    BranchGroup bgLight = new BranchGroup();
    BoundingSphere bounds = new BoundingSphere(new Point3d(0.0,0.0,0.0), 100.0);
    Color3f lightColour1 = new Color3f(1.0f,1.0f,1.0f);
    Vector3f lightDir1 = new Vector3f(-1.0f,0.0f,-0.5f);

```

```
        DirectionalLight light1 = new DirectionalLight(lightColour1, lightDir1);
        light1.setInfluencingBounds(bounds);
        bgLight.addChild(light1);

        su.addBranchGraph(bgLight);
    }

    public static void setToMyDefaultAppearance(Appearance app, Color3f col) {
        app.setMaterial(new Material(col,col,col,col,150.0f));
    }

    public static void main(String[] args) throws IOException {
        Main main = new Main();
    }
}
```

## Результат

