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”

Виконала
студентка 3 курсу
групи КП-83:
Тиченко Анна
Варіант: 20

Київ 2021

Завдання:

Виконати анімацію тривимірної сцени за варіантом.

Код програми

Main.java

```
package com.company;

public class Main {

    public static void main(String[] args) {
        new LadyBug();
    }

}
```

Animation.java

```
package com.company;
import javax.vecmath.*;

import com.sun.j3d.utils.image.TextureLoader;
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.util.Hashtable;
import java.util.Enumeration;

public class LadyBug extends JFrame {
    public Canvas3D myCanvas3D;

    public LadyBug() {
        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);
        setTitle("LadyBug");
        setSize(700,700);
        getContentPane().add("Center", myCanvas3D);
        setVisible(true);
    }

    public void createSceneGraph(SimpleUniverse su){
        ObjectFile f = new ObjectFile(ObjectFile.RESIZE);
        Scene widowScene = null;
        try{
            widowScene = f.load("assets/ladybug.obj");
        }
        catch (Exception e){
            System.out.println("File loading failed:" + e);
        }

        Transform3D scaling = new Transform3D();
        scaling.setScale(1.0/6);
        Transform3D tfRoach = new Transform3D();
        tfRoach.rotX(3*Math.PI/2);
        tfRoach.mul(scaling);
        TransformGroup tgRoach = new TransformGroup(tfRoach);
        TransformGroup sceneGroup = new TransformGroup();


        Hashtable roachNamedObjects = widowScene.getNamedObjects();
        Enumeration enumer = roachNamedObjects.keys();
        String name;
        while (enumer.hasMoreElements()){
            name = (String) enumer.nextElement();
            System.out.println("Name: "+name);
        }


        BoundingSphere bounds = new BoundingSphere(new 
Point3d(120.0,250.0,100.0),Double.MAX_VALUE);
        BranchGroup theScene = new BranchGroup();

        Appearance redCarA = new Appearance();
        setToMyDefaultAppearance(redCarA,new Color3f(0.8f,0.1f,0.0f));
        TransformGroup tgBody = new TransformGroup();
        Shape3D body_widow = (Shape3D) roachNamedObjects.get("ladybug");
        body_widow.setAppearance(redCarA);
        tgBody.addChild(body_widow.cloneTree());

        TransformGroup antenna_1 = new TransformGroup();
        Shape3D antenna1 = (Shape3D) roachNamedObjects.get("antenna");
        antenna_1.addChild(antenna1.cloneTree());
        TransformGroup antenna_2 = new TransformGroup();
        Shape3D antenna2 = (Shape3D) roachNamedObjects.get("antenna2");
        antenna_2.addChild(antenna2.cloneTree());



        // ANIMATION
------------------------------------------------------------------------
-
        int noRotHour = 100;
        int timeRotationHour = 300;

        BoundingSphere bs = new BoundingSphere(new
Point3d(0.0,0.0,0.0),Double.MAX_VALUE);

        // leg1_1 ---------------------------------------
        int timeStart = 0;

        Alpha leg1_1RotAlpha = new
Alpha(noRotHour,Alpha.INCREASING_ENABLE,timeStart,0,timeRotationHour,
                0,0,0,0,0);

        Shape3D leg1_1 = (Shape3D) roachNamedObjects.get("leg1");
        TransformGroup tgLeg1_1 = new TransformGroup();
        tgLeg1_1.addChild(leg1_1.cloneTree());

        Transform3D legRotAxis = new Transform3D();
        legRotAxis.rotZ(Math.PI/2);

        RotationInterpolator leg1_1Rotation = new
RotationInterpolator(leg1_1RotAlpha,tgLeg1_1,legRotAxis,(float)
Math.PI/2,0.0f);
        leg1_1Rotation.setSchedulingBounds(bs);
        tgLeg1_1.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        tgLeg1_1.addChild(leg1_1Rotation);
        //----------------------------------------------------

        // leg2_1 ---------------------------------------
        Alpha leg2_1RotAlpha = new
Alpha(noRotHour,Alpha.INCREASING_ENABLE,100,0,timeRotationHour,
                0,0,0,0,0);

        Shape3D leg2_1 = (Shape3D) roachNamedObjects.get("leg2");
        TransformGroup tgLeg2_1 = new TransformGroup();
        tgLeg2_1.addChild(leg2_1.cloneTree());

        Transform3D leg2RotAxis = new Transform3D();

        RotationInterpolator leg2_1Rotation = new
RotationInterpolator(leg2_1RotAlpha,tgLeg2_1,leg2RotAxis,(float)
Math.PI/8,0.0f);
        leg2_1Rotation.setSchedulingBounds(bs);
        tgLeg2_1.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        tgLeg2_1.addChild(leg2_1Rotation);
        //----------------------------------------------------


        // leg3_1 ---------------------------------------
        Alpha leg4_1RotAlpha = new
Alpha(noRotHour,Alpha.INCREASING_ENABLE,300,0,timeRotationHour,
                0,0,0,0,0);

        Shape3D leg4_1 = (Shape3D) roachNamedObjects.get("leg3");
        TransformGroup tgLeg4_1 = new TransformGroup();
        tgLeg4_1.addChild(leg4_1.cloneTree());

        RotationInterpolator leg4_1Rotation = new
RotationInterpolator(leg4_1RotAlpha,tgLeg4_1,leg2RotAxis,(float)
Math.PI/8,0.0f);
        leg4_1Rotation.setSchedulingBounds(bs);
        tgLeg4_1.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        tgLeg4_1.addChild(leg4_1Rotation);
        //----------------------------------------------------

        // leg1_2 ---------------------------------------

        Alpha leg1_2RotAlpha = new
Alpha(noRotHour,Alpha.INCREASING_ENABLE,200,0,timeRotationHour,
                0,0,0,0,0);

        Shape3D leg1_2 = (Shape3D) roachNamedObjects.get("leg4");
        TransformGroup tgLeg1_2 = new TransformGroup();
        tgLeg1_2.addChild(leg1_2.cloneTree());
        ;

        RotationInterpolator leg1_2Rotation = new
RotationInterpolator(leg1_2RotAlpha,tgLeg1_2,legRotAxis,(float)
Math.PI/2,0.0f);
        leg1_2Rotation.setSchedulingBounds(bs);
        tgLeg1_2.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        tgLeg1_2.addChild(leg1_2Rotation);
        //----------------------------------------------------

        // leg2_2 ---------------------------------------
        Alpha leg2_2RotAlpha = new
Alpha(noRotHour,Alpha.INCREASING_ENABLE,300,0,timeRotationHour,
                0,0,0,0,0);

        Shape3D leg2_2 = (Shape3D) roachNamedObjects.get("leg5");
        TransformGroup tgLeg2_2 = new TransformGroup();
        tgLeg2_2.addChild(leg2_2.cloneTree());

        RotationInterpolator leg2_2Rotation = new
RotationInterpolator(leg2_2RotAlpha,tgLeg2_2,leg2RotAxis,-(float)
Math.PI/8,0.0f);
        leg2_2Rotation.setSchedulingBounds(bs);
        tgLeg2_2.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        tgLeg2_2.addChild(leg2_2Rotation);
        //------------------------------------------------------


        // leg4_2 ---------------------------------------
        Alpha leg4_2RotAlpha = new
Alpha(noRotHour,Alpha.INCREASING_ENABLE,500,0,timeRotationHour,
                0,0,0,0,0);

        Shape3D leg4_2 = (Shape3D) roachNamedObjects.get("leg6");
        TransformGroup tgLeg4_2 = new TransformGroup();
        tgLeg4_2.addChild(leg4_2.cloneTree());

        RotationInterpolator leg4_2Rotation = new
RotationInterpolator(leg4_2RotAlpha,tgLeg4_2,leg2RotAxis,-(float)
Math.PI/8,0.0f);
        leg4_2Rotation.setSchedulingBounds(bs);
        tgLeg4_2.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        tgLeg4_2.addChild(leg4_2Rotation);
        //------------------------------------------------------


        sceneGroup.addChild(tgLeg1_1);
        sceneGroup.addChild(tgLeg2_1);
        sceneGroup.addChild(tgLeg4_1);

        sceneGroup.addChild(tgLeg1_2);
        sceneGroup.addChild(tgLeg2_2);
        sceneGroup.addChild(tgLeg4_2);

        sceneGroup.addChild(antenna_1);
        sceneGroup.addChild(antenna_2);

        sceneGroup.addChild(tgBody.cloneTree());


        // movement widow
--------------------------------------------------------
        Transform3D tCrawl = new Transform3D();
        tCrawl.rotY(-Math.PI/2);

        long crawlTime = 10000;
        Alpha crawlAlpha = new Alpha(1,
                Alpha.INCREASING_ENABLE,
                0,
                0, crawlTime,0,0,0,0,0);

        float crawlDistance = 7.0f; //відстань, на яку просунеться об’єкт
        PositionInterpolator posICrawl = new
PositionInterpolator(crawlAlpha,
                sceneGroup,tCrawl, -5.0f, crawlDistance);
        posICrawl.setSchedulingBounds(bs);
        sceneGroup.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        sceneGroup.addChild(posICrawl);

        tgRoach.addChild(sceneGroup);
        theScene.addChild(tgRoach);


//----------------------------------------------------------------------
--------------------------------

        //створюємо фон
        Background background = new Background(new
TextureLoader("assets/grass.jpg", myCanvas3D).getImage());
        background.setImageScaleMode(Background.SCALE_FIT_MAX);
        background.setApplicationBounds(new BoundingSphere(new
Point3d(),1000));
        background.setCapability(Background.ALLOW_IMAGE_WRITE);
        background.setApplicationBounds(bounds);
        theScene.addChild(background);
        theScene.compile();

        //додаємо сцену до віртуального всесвіту
        su.addBranchGraph(theScene);
    }

    //метод для генерації зовнішньої поверхні
    public static void setToMyDefaultAppearance(Appearance app, Color3f
col){
        app.setMaterial(new Material(col,col,col,col,150.0f));
    }

    //метод для додавання освітлення
    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);
    }

}
```

Результати роботи програми