



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 № 5

з дисципліни “Математичні та алгоритмічні основи комп’ютерної графіки”

Виконав
студент III курсу
групи КП-81

Дикий Ілля
(прізвище, ім’я, по батькові)

Зарахована
“ ____ ” “ ____ ” 20 ____ р.
викладачем

Шкурат Оксаною Сергіївною
(прізвище, ім’я, по батькові)

варіант № 3

Завдання

Імпортувати моделі тривимірних об'єктів форматів, що визначені варіантом. Створити реалістичну анімацію об'єкту. Додати до сцени фон, інші об'єкти для надання сцені реалістичного вигляду. Для цього використати текстури, матеріали, імпортувати додаткові об'єкти з відкритих бібліотек, за бажанням створити прості об'єкти у графічному редакторі. Студенти, які мають непарний номер варіанту у списку групи імпортують моделі формату .obj, парний варіант – .lwo.

Лістинг коду програми

Ladybird.java

```
package com;

import com.sun.j3d.loaders.Scene;
import com.sun.j3d.loaders.lw3d.Lw3dLoader;
import com.sun.j3d.loaders.objectfile.ObjectFile;
import com.sun.j3d.utils.image.TextureLoader;
import com.sun.j3d.utils.universe.SimpleUniverse;
import com.sun.j3d.utils.universe.ViewingPlatform;

import javax.media.j3d.*;
import javax.swing.*;
import javax.vecmath.*;
import java.awt.*;
import java.io.FileReader;
import java.io.IOException;
import java.util.Map;

public class Ladybird extends JFrame {
    static SimpleUniverse universe;
    static Scene scene;
    static Map<String, Shape3D> nameMap;
    static BranchGroup root;
    static Canvas3D canvas;

    static TransformGroup wholeLadybird;
    static Transform3D transform3D;

    public Ladybird() throws IOException{
        configureWindow();
    }
}
```

```

        configureCanvas();
        configureUniverse();
        addModelToUniverse();
        setLadybirdElementsList();
        addAppearance();
        addImageBackground();
        addLightToUniverse();
        addOtherLight();
        ChangeViewAngle();
        root.compile();
        universe.addBranchGraph(root);
    }

    private void configureWindow() {
        setTitle("Ladybird Animation Example");
        setSize(760,640);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    private void configureCanvas(){
        canvas=new Canvas3D(SimpleUniverse.getPreferredConfiguration());
        canvas.setDoubleBufferEnable(true);
        getContentPane().add(canvas, BorderLayout.CENTER);
    }

    private void configureUniverse(){
        root= new BranchGroup();
        universe= new SimpleUniverse(canvas);
        universe.getViewingPlatform().setNominalViewingTransform();
    }

    private void addModelToUniverse() throws IOException{
        scene =
getSceneFromFile("/home/stilpert/Education/MAOKG/lab5/data/ladybird.obj");
        root=scene.getSceneGroup();
    }

    private void addLightToUniverse(){
        Bounds bounds = new BoundingSphere();
        Color3f color = new Color3f(65/255f, 30/255f, 25/255f);
        Vector3f lightdirection = new Vector3f(-1f,-1f,-1f);
    }

```

```

        DirectionalLight dirlight = new
DirectionalLight(color,lightdirection);
        dirlight.setInfluencingBounds(bounds);
        root.addChild(dirlight);
    }

    private void printModelElementsList(Map<String,Shape3D> nameMap) {
        for (String name : nameMap.keySet()) {
            System.out.printf("Name: %s\n", name);}
    }

    private void setLadybirdElementsList() {
        nameMap=scene.getNamedObjects();
        printModelElementsList(nameMap);

        wholeLadybird = new TransformGroup();
        transform3D = new Transform3D();
        Vector3f translate = new Vector3f();
        translate.set( .0f, -1.0f, .0f);
        transform3D.setScale(new Vector3d(0.6,0.6,0.6));
        Transform3D rotateTransformY = new Transform3D();
        rotateTransformY.rotY(-Math.PI/2);
        transform3D.mul(rotateTransformY);
        transform3D.setTranslation(translate);
        wholeLadybird.setTransform(transform3D);
        root.removeChild(nameMap.get("ladybird"));
        wholeLadybird.addChild(nameMap.get("ladybird"));
        wholeLadybird.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        root.addChild(wholeLadybird);
    }

    Texture getTexture(String path) {
        TextureLoader textureLoader = new
TextureLoader(path,"LUMINANCE",canvas);
        Texture texture = textureLoader.getTexture();
        texture.setBoundaryModeS(Texture.WRAP);
        texture.setBoundaryModeT(Texture.WRAP);
        texture.setBoundaryColor( new Color4f( 0.0f, 1.0f, 0.0f, 0.0f ) );
        return texture;
    }

```

```

Material getMaterial() {
    Material material = new Material();
    material.setAmbientColor ( new Color3f( 0.33f, 0.26f, 0.23f ) );
    material.setDiffuseColor ( new Color3f( 0.50f, 0.11f, 0.00f ) );
    material.setSpecularColor( new Color3f( 0.95f, 0.73f, 0.00f ) );
    material.setShininess( 0.3f );
    material.setLightingEnable(true);
    return material;
}

private void addAppearance(){
    Appearance ladybirdAppearance = new Appearance();

ladybirdAppearance.setTexture(getTexture("/home/stilpert/Education/MAOKG/lab5
/data/wings.jpg"));
    TextureAttributes texAttr = new TextureAttributes();
    texAttr.setTextureMode(TextureAttributes.MODULATE);
    ladybirdAppearance.setTextureAttributes(texAttr);
    ladybirdAppearance.setMaterial(getMaterial());
    Shape3D ladybird = nameMap.get("ladybird");
    ladybird.setAppearance(ladybirdAppearance);
}

private void addColorBackground(){
    Background background = new Background(new Color3f(Color.CYAN));
    BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0,
0.0),100.0);
    background.setApplicationBounds(bounds);
    root.addChild(background);
}

private void addImageBackground(){
    TextureLoader t = new
TextureLoader("/home/stilpert/Education/MAOKG/lab5/data/green.jpg", canvas);
    Background background = new Background(t.getImage());
    background.setImageScaleMode(Background.SCALE_FIT_ALL);
    BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0,
0.0),100.0);
    background.setApplicationBounds(bounds);
    root.addChild(background);
}

```

```

private void ChangeViewAngle(){
    ViewingPlatform vp = universe.getViewingPlatform();
    TransformGroup vpGroup =
vp.getMultiTransformGroup().getTransformGroup(0);
    Transform3D vpTranslation = new Transform3D();
    Vector3f translationVector = new Vector3f(0.0F, -1.2F, 6F);
    vpTranslation.setTranslation(translationVector);
    vpGroup.setTransform(vpTranslation);
}

private void addOtherLight(){
    Color3f directionalLightColor = new Color3f(Color.BLACK);
    Color3f ambientLightColor = new Color3f(Color.WHITE);
    Vector3f lightDirection = new Vector3f(-1F, -1F, -1F);

    AmbientLight ambientLight = new AmbientLight(ambientLightColor);
    DirectionalLight directionalLight = new
DirectionalLight(directionalLightColor, lightDirection);

    Bounds influenceRegion = new BoundingSphere();

    ambientLight.setInfluencingBounds(influenceRegion);
    directionalLight.setInfluencingBounds(influenceRegion);
    root.addChild(ambientLight);
    root.addChild(directionalLight);
}

public static Scene getSceneFromFile(String location) throws IOException {
    ObjectFile file = new ObjectFile(ObjectFile.RESIZE);
    file.setFlags (ObjectFile.RESIZE | ObjectFile.TRIANGULATE |
ObjectFile.STRIPIFY);
    return file.load(new FileReader(location));
}

//Not always works
public static Scene getSceneFromLwoFile(String location) throws
IOException {
    Lw3dLoader loader = new Lw3dLoader();
    return loader.load(new FileReader(location));
}

```

```

public static void main(String[]args){
    try {
        Ladybird window = new Ladybird();
        AnimationLadybird ladybirdMovement = new
AnimationLadybird(wholeLadybird, transform3D, window);
        window.addKeyListener(ladybirdMovement);
        window.setVisible(true);
    }
    catch (IOException ex) {
        System.out.println(ex.getMessage());
    }
}
}

```

AnimationLadybird.java

```

package com;

import javax.media.j3d.Transform3D;
import javax.media.j3d.TransformGroup;
import javax.swing.*;
import javax.vecmath.Vector3d;
import javax.vecmath.Vector3f;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;

public class AnimationLadybird implements ActionListener, KeyListener {
    private Button go;
    private TransformGroup wholeLadybird;
    private Transform3D translateTransform;
    private Transform3D rotateTransformX;
    private Transform3D rotateTransformY;
    private Transform3D rotateTransformZ;

    private JFrame mainFrame;

```

```

private float sign=1.0f;
private float zoom=0.6f;
private float xloc=0.3f;
private float yloc=-1.0f;
private float zloc=0.0f;
private int moveType=1;
private Timer timer;

public AnimationLadybird(TransformGroup wholeLadybird, Transform3D trans,
JFrame frame){
    go = new Button("Go");
    this.wholeLadybird=wholeLadybird;
    this.translateTransform=trans;
    this.mainFrame=frame;

    rotateTransformX= new Transform3D();
    rotateTransformY= new Transform3D();
    rotateTransformZ= new Transform3D();

    Ladybird.canvas.addKeyListener(this);
    timer = new Timer(100, this);

    Panel p =new Panel();
    p.add(go);
    mainFrame.add("North",p);
    go.addActionListener(this);
    go.addKeyListener(this);
}

@Override
public void actionPerformed(ActionEvent e) {
    // start timer when button is pressed
    if (e.getSource()==go){
        if (!timer.isRunning()) {
            timer.start();
            go.setLabel("Stop");
        }
        else {
            timer.stop();
            go.setLabel("Go");
        }
    }
}

```



```

    }
    else {
        Move(moveType);
        translateTransform.setScale(new Vector3d(zoom, zoom, zoom));
        translateTransform.setTranslation(new Vector3f(xloc, yloc, zloc));
        wholeLadybird.setTransform(translateTransform);
    }
}

private void Move(int mType){
    if(mType==1){ //fly forward and back
        xloc += 0.1 * sign;
        if (Math.abs(xloc *2) >= 3 ) {
            sign = -1.0f * sign;
            rotateTransformY.rotY(Math.PI);
            translateTransform.mul(rotateTransformY);
        }
    }
    if(mType==2){
        yloc += 0.1 * sign;
        if ( yloc >= 1 || yloc <= -2) {
            sign = -1.0f * sign;
            rotateTransformY.rotY(Math.PI);
            translateTransform.mul(rotateTransformY);
        }
    }
}

@Override
public void keyTyped(KeyEvent e) {
    //Invoked when a key has been typed.
}

@Override
public void keyPressed(KeyEvent e) {
    if (e.getKeyChar()=='l') {
        if(moveType==1){
            rotateTransformX.rotX(sign * Math.PI/2);
            rotateTransformZ.rotZ((sign * -Math.PI/2));
            translateTransform.mul(rotateTransformX);
            translateTransform.mul(rotateTransformZ);
        }
    }
}

```

```

        moveType=2;
    }
    else if(moveType==2){
        rotateTransformY.rotY(sign * -Math.PI/2);
        rotateTransformZ.rotZ((sign * Math.PI/2));
        translateTransform.mul(rotateTransformY);
        translateTransform.mul(rotateTransformZ);
        moveType=1;
    }
}

@Override
public void keyReleased(KeyEvent e) {
    // Invoked when a key has been released.
}

}

```

Результати

St...



