



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(ла)  
студент(ка) III курсу  
групи КП - 81  
Івахненко Маргарита Василівна  
*(прізвище, ім'я, по батькові)*

Київ 2021

## **Постановка 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.

## Тексти коду програми

### Animation.java

```
package com.sample;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.media.j3d.*;
import javax.swing.JFrame;
import javax.swing.Timer;
import javax.vecmath.*;

public class Animation implements ActionListener {
    private Button go;
    private TransformGroup wholeBall;
    private Transform3D translateTransform;

    private JFrame mainFrame;

    private float xloc=0.3f;
    private float yloc=0.3f;
    private float zloc=0.0f;
    private int moveType = 0;
    private Timer timer;
    private boolean turn = true;
    private float angle = 0;
    private float dx = 0;
    private float dy = 0;

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

        timer = new Timer(50, this);

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

    private void initialPlaneState(){
        xloc = -2.3f;
        yloc = .7f;
        dx = dy = 0;
    }
}
```

```

        angle = (float) ( 5 * Math.PI/ 8);
        turn = true;
        moveType = 0;
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == go){
            if (!timer.isRunning()) {
                initialPlaneState();
                timer.start();
                go.setLabel("Stop");
            }
            else {
                timer.stop();
                go.setLabel("Go");
            }
        } else {
            move();
            translateTransform.setTranslation(new Vector3f(xloc,
yloc, zloc));
            wholeBall.setTransform(translateTransform);
        }
    }

    private void move(){
        xloc += 0.08;
        if (moveType == 0 && xloc > -1.6)
            moveType++;
        if (moveType == 1) {
            angle += 0.6f;
            yloc -= 0.024 - 0.027 * Math.sin(angle);
        }
        if (xloc > 2)
            initialPlaneState();
    }
}

```

## Main.java

```
package com.sample;

import com.sun.j3d.loaders.Scene;
import com.sun.j3d.loaders.objectfile.ObjectFile;
import com.sun.j3d.utils.image.TextureLoader;
import com.sun.j3d.utils.universe.*;

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

public class Main extends JFrame {
    static SimpleUniverse universe;
    static Scene staircase1, staircase2, ball;
    static BranchGroup root;
    static Canvas3D canvas;

    static TransformGroup wholeBall;
    static Transform3D transform3D;

    public Main() throws IOException {
        configureWindow();
        configureCanvas();
        configureUniverse();
        addStairsToUniverse();
        addBallToUniverse();
        addAppearanceToObjects();
        addColorBackground();
        addLightToUniverse();
        ChangeViewAngle();
        root.compile();
        universe.addBranchGraph(root);
    }

    private void configureWindow() {
        setTitle("lab5");
        setSize(760,680);
        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 addStairsToUniverse() throws IOException {
        staircase1 = getSceneFromFile("source/staircase1.obj");
        staircase2 = getSceneFromFile("source/staircase1.obj");
        BranchGroup group = new BranchGroup();
        TransformGroup tg = new TransformGroup();
        Transform3D transform3D = new Transform3D();
        transform3D.setTranslation(new Vector3f(-1.2f, .4f, 0));
        tg.setTransform(transform3D);
        tg.addChild(staircase1.getSceneGroup());
        group.addChild(tg);
        group.addChild(staircase2.getSceneGroup());
        root = group;
    }

    public void addBallToUniverse() throws IOException {
        ball = getSceneFromFile("source/pokeball.obj");
        wholeBall = new TransformGroup();

        wholeBall.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        transform3D = new Transform3D();
        transform3D.setTranslation(new Vector3f(-2.3f, .7f, 0));
        transform3D.setScale(0.1);
        wholeBall.setTransform(transform3D);
        wholeBall.addChild(ball.getSceneGroup());
        root.addChild(wholeBall);
    }

    Texture getTexture(String path) {
        TextureLoader textureLoader = new
TextureLoader(path, "RGB", 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(.15f, .15f, .15f));
        material.setDiffuseColor (new Color3f(1f, 1f, 1f));
        material.setSpecularColor(new Color3f(0f, 0f, 0f));
    }

```

```

        material.setEmissiveColor(new Color3f(0f, 0f, 0));
        material.setShininess(0.3f);
        return material;
    }

    private void addAppearance(Shape3D object, String texturePath){
        Appearance objectAppearance = new Appearance();
        objectAppearance.setTexture(getTexture(texturePath));
        TextureAttributes texAttr = new TextureAttributes();
        texAttr.setTextureMode(TextureAttributes.MODULATE);

texAttr.setPerspectiveCorrectionMode(TextureAttributes.NICEST);
        objectAppearance.setTextureAttributes(texAttr);
        objectAppearance.setMaterial(getMaterial());
        object.setAppearance(objectAppearance);
    }

    private void addAppearanceToObjects() {
        addAppearance((Shape3D)
staircase1.getNamedObjects().get("stairsteps_layer1"),
"source/texture.jpg");
        addAppearance((Shape3D)
staircase2.getNamedObjects().get("stairsteps_layer1"),
"source/texture.jpg");
        addAppearance((Shape3D)
ball.getNamedObjects().get("mesh_soccer_ball"), "source/ball.png");
    }

    private void addColorBackground(){
        Background background = new Background(new
Color3f(Color.GREEN));
        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(-.6F, .8F, 3.8F);
        vpTranslation.setTranslation(translationVector);
        vpGroup.setTransform(vpTranslation);
    }

    private void addLightToUniverse() {
        Color3f directionalLightColor = new Color3f(Color.WHITE);
        Color3f ambientLightColor = new Color3f(Color.WHITE);
    }

```

```

        Vector3f lightDirection = new Vector3f(-3F, -3F, -3F);

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

        Bounds influenceRegion = new BoundingSphere();
        Transform3D transform3D = new Transform3D();
        transform3D.setScale(8);
        influenceRegion.transform(transform3D);

        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);
        return file.load(new FileReader(location));
    }

    public static void main(String[]args){
        try {
            Main window = new Main();
            new Animation(wholeBall, transform3D, window);
            window.setVisible(true);
        }
        catch (IOException ex) {
            System.out.println(ex.getMessage());
        }
    }
}

```

## Результат роботи програми



