



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”
Варіант 7

Виконав
студент 3 курсу
групи КП-81
Каснер Максим

Київ 2021

Завдання на лабораторну роботу

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.

Варіант 7 :

7. Непарний варіант, отже я імпортував модель .obj

Код програми

Main.java

```
import com.sun.j3d.utils.universe.*;

import javax.media.j3d.*;
import javax.vecmath.*;
import javax.media.j3d.Background;

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

import java.awt.*;
import java.io.FileReader;
import java.io.IOException;
import java.util.Hashtable;
import java.util.Map;

import javax.swing.JFrame;

public class Main extends JFrame {
    private static Canvas3D canvas;
    private static SimpleUniverse universe;
    private static BranchGroup root;
```

```

private static TransformGroup ball;
private static TransformGroup stairs;

public Main() throws IOException {
    configureWindow();
    configureCanvas();
    configureUniverse();

    root = new BranchGroup();
    addImageBackground();
    addDirectionalLightToUniverse();
    addAmbientLightToUniverse();
    ChangeViewAngle();

    ball = getBallGroup("box04", "files/ball.obj");
    root.addChild(ball);
    stairs = getStairsGroup("layer0_001", "files/3d-model.obj");
    root.addChild(stairs);

    universe.addBranchGraph(root);
}

private void configureWindow() {
    setTitle("Lab5");
    setExtendedState(JFrame.MAXIMIZED_BOTH);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

private void configureCanvas() {
    canvas = new Canvas3D(SimpleUniverse.getPreferredConfiguration());
    getContentPane().add(canvas);
}

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

private void addImageBackground() {
    TextureLoader t = new TextureLoader("files/flat.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 addDirectionalLightToUniverse() {
        BoundingSphere bounds = new BoundingSphere();
        bounds.setRadius(100);

        DirectionalLight light = new DirectionalLight(new Color3f(1, 1, 1), new Vector3f(-1, -1, -1));
        light.setInfluencingBounds(bounds);

        root.addChild(light);
    }

    private void addAmbientLightToUniverse() {
        AmbientLight light = new AmbientLight(new Color3f(1, 1, 1));
        light.setInfluencingBounds(new BoundingSphere());
        root.addChild(light);
    }

    private void ChangeViewAngle() {
        ViewingPlatform vp = universe.getViewingPlatform();
        TransformGroup vpGroup = vp.getMultiTransformGroup().getTransformGroup(0);
        Transform3D vpTranslation = new Transform3D();
        vpTranslation.setTranslation(new Vector3f(0, 0, 5));
        vpGroup.setTransform(vpTranslation);
    }

    private TransformGroup getStairsGroup(String name, String path) throws IOException {
        Shape3D shape = getModelShape3D(name, path);
        addStairsAppearance(shape);
        Transform3D transform3D = new Transform3D();
        transform3D.setScale(new Vector3d(1.2, 1.4, 1));

        Transform3D rotationY = new Transform3D();
        rotationY.rotY(-Math.PI/2.5);

        transform3D.mul(rotationY);

        TransformGroup group = getModelGroup(shape);
        group.setTransform(transform3D);

        return group;
    }

```

```

private TransformGroup getBallGroup(String name, String path) throws IOException {
    Shape3D shape = getModelShape3D(name, path);
    addBallAppearance(shape);

    Transform3D transform3D = new Transform3D();
    transform3D.setScale(new Vector3d(0.15, 0.15, 0.15));

    Transform3D rotationY = new Transform3D();
    transform3D.mul(rotationY);

    TransformGroup group = getModelGroup(shape);
    group.setTransform(transform3D);

    return group;
}

private TransformGroup getModelGroup(Shape3D shape) {
    TransformGroup group = new TransformGroup();
    group.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
    group.addChild(shape);
    return group;
}

private Shape3D getModelShape3D(String name, String path) throws IOException {
    Scene scene = getSceneFromFile(path);
    Hashtable map = scene.getNamedObjects();
    /*printModelElementsList(map);*/
    Shape3D shape = (Shape3D) map.get(name);
    scene.getSceneGroup().removeChild(shape);
    return shape;
}

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

private void printModelElementsList(Map<String, Shape3D> map) {
    for (String name : map.keySet()) {
        System.out.println("Name: " + name);
    }
}

private void addBallAppearance(Shape3D shape) {
    Appearance appearance = new Appearance();
    Color3f emissive = new Color3f(new Color(45, 13, 14));

```

```

        Color3f ambient = new Color3f(new Color(56, 34, 35));
        Color3f diffuse = new Color3f(new Color(56, 24, 26));
        Color3f specular = new Color3f(new Color(56, 34, 35));
        appearance.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));
        shape.setAppearance(appearance);
    }

    private void addStairsAppearance(Shape3D shape) {
        Appearance appearance = new Appearance();
        Color3f emissive = new Color3f(new Color(89, 79, 53));
        Color3f ambient = new Color3f(new Color(66, 52, 16));
        Color3f diffuse = new Color3f(new Color(66, 52, 16));
        Color3f specular = new Color3f(new Color(0,0, 0));
        appearance.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));
        shape.setAppearance(appearance);
    }

    public static void main(String[] args) {
        try {
            Main window = new Main();
            BallAnimation ballMovement = new BallAnimation(ball, stairs);
            canvas.addKeyListener(ballMovement);
            window.setSize(800,800);
            window.setVisible(true);
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }
}

```

BallAnimation.java

```

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;

import javax.media.j3d.*;
import javax.swing.Timer;
import javax.vecmath.*;

public class BallAnimation implements ActionListener, KeyListener {
    private final TransformGroup ball;
    private final TransformGroup stairs;
    private final Transform3D transform3Dball = new Transform3D();
}

```

```

private float height = 0.0f;
private float sign = 1.0f; // going up or down
private final Timer timer;
private float xloc = -1.1f;
private float zloc = -0.45f;

public BallAnimation(TransformGroup ball, TransformGroup stairs) {
    this.ball = ball;
    this.stairs = stairs;

    Transform3D transform3DStairs = new Transform3D();
    this.stairs.getTransform(transform3DStairs);
    this.ball.getTransform(this.transform3Dball);

    timer = new Timer(50, this);
    timer.start();
}

@Override
public void actionPerformed(ActionEvent e) {
    height += .1 * sign;

    if (Math.abs(height * 2) >= 1)
        sign = -1.0f * sign;

    transform3Dball.setScale(new Vector3d(0.13, 0.13, 0.13));
    transform3Dball.setTranslation(new Vector3f(xloc, 0.5f * height + zloc, 0f));
    ball.setTransform(transform3Dball);
}

public void keyPressed(KeyEvent e) {
    // дія по натисненню на клавішу
    if (e.getKeyChar() == 'd') {
        xloc = xloc + .22f;
        zloc = zloc + .2f;
    }

    if (e.getKeyChar() == 'a') {
        xloc = xloc - .22f;
        zloc = zloc - .2f;
    }
}

@Override
public void keyTyped(KeyEvent e) {
}

@Override

```

```
public void keyReleased(KeyEvent ev) {  
    }  
}
```


Результати роботи програми :



