



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-83

Палій Дмитро Володимирович
(прізвище, ім'я, по батькові)

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

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

варіант № 16

Варіант завдання

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.

Варіант: .lwo

В ході виконання даної лабораторної роботи мені не вдалося завантажити модель .lwo, тому була використана модель формату .obj.

Лістинг коду програми

```
package sample;

import javax.vecmath.Color3f;
import com.sun.j3d.utils.universe.*;

import java.awt.Color;
import javax.media.j3d.*;
import javax.media.j3d.Material;
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.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.Map;
import javax.swing.Timer;
import javax.swing.JFrame;

class Main extends JFrame implements ActionListener {
    static final String ballLocation =
"C:\\Users\\dpali\\Documents\\maokr\\lab5\\res\\ball.obj";
    static final String bgLocation =
"C:\\Users\\dpali\\Documents\\maokr\\lab5\\res\\bg.jpg";

    private static final Color3f SPECULAR_LIGHT_COLOR = new
Color3f(Color.white);
    private static final Color3f AMBIENT_LIGHT_COLOR = new
Color3f(Color.white);
    private static final Color3f EMISSIVE_LIGHT_COLOR = new
Color3f(Color.black);

    private Transform3D translateTransform = new Transform3D();
```

```

private SimpleUniverse universe;
private Scene scene;
private BranchGroup root;
private TransformGroup ball = new TransformGroup();
private Canvas3D canvas;
private float xloc=0.1f;
private float yloc=0.5f;
private float zloc=-0.5f;
private float floor=0.05f;
private float yVel = -0.01f;

public static void main(String[] args) {
    Main window = new Main();
    window.setVisible(true);
}

public Main() {
    configureUniverse();
    addLightToUniverse();
    setBackground();
    try {
        loadScene();
    }
    catch (FileNotFoundException exception) {
        System.out.println("file not found" + ballLocation);
    }
    setInitPosition();

    Timer timer = new Timer(20, this);
    timer.start();

    universe.addBranchGraph(root);
}

private void setInitPosition() {
    var initPosition = new Vector3d(xloc, yloc, zloc);
    translateTransform.setTranslation(initPosition);
}

private void configureUniverse(){
    canvas = new Canvas3D(SimpleUniverse.getPreferredConfiguration());
    canvas.setDoubleBufferEnable(true);
    canvas.setFocusable(true);
    add(canvas, BorderLayout.CENTER);

    setTitle("lab 5");
    setSize(960, 540);
    setResizable(true);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setExtendedState(JFrame.MAXIMIZED_BOTH);

    root = new BranchGroup();
    universe = new SimpleUniverse(canvas);
    universe.getViewingPlatform().setNominalViewingTransform();
}

private void addLightToUniverse(){
    Bounds bounds = new BoundingSphere();
    Color3f color = new Color3f(1f, 1f, 1f);
    Vector3f lightDirection = new Vector3f(0f,-1f,-1f);
    DirectionalLight directionalLight = new
DirectionalLight(color,lightDirection);
    directionalLight.setInfluencingBounds(bounds);
    root.addChild(directionalLight);
}

```

```

    }

    private void loadScene() throws FileNotFoundException {
        ObjectFile loader = new ObjectFile(ObjectFile.RESIZE);
        scene = loader.load(new FileReader(ballLocation));

        Map<String, Shape3D> nameMap = scene.getNamedObjects();

        var sceneGroup = scene.getSceneGroup();

        sceneGroup.removeChild(nameMap.get("background"));
        sceneGroup.removeChild(nameMap.get("reflect_panel"));
        sceneGroup.removeChild(nameMap.get("floor"));

        Appearance redAppearance = getAppearance(Color.red);
        Appearance whiteAppearance = getAppearance(Color.white);
        Appearance blackAppearance = getAppearance(Color.black);

        nameMap.get("inner").setAppearance(blackAppearance);
        nameMap.get("top").setAppearance(redAppearance);
        nameMap.get("bottom").setAppearance(whiteAppearance);
        nameMap.get("button").setAppearance(whiteAppearance);
        nameMap.get("buttonrim").setAppearance(whiteAppearance);
        ball.addChild(sceneGroup);
        ball.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        root.addChild(ball);
    }

    private void setBackground() {
        TextureLoader t = new TextureLoader(bgLocation, 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);
    }

    Appearance getAppearance(Color color) {
        Appearance app = new Appearance();
        app.setMaterial(getMaterial(color));
        return app;
    }

    Material getMaterial(Color color) {
        return new Material(AMBIENT_LIGHT_COLOR,
            EMISSIVE_LIGHT_COLOR,
            new Color3f(color),
            SPECULAR_LIGHT_COLOR,
            100F);
    }

    void listSceneNamedObjects(Scene scene) {
        Map<String, Shape3D> nameMap = scene.getNamedObjects();

        for (String name : nameMap.keySet()) {
            System.out.printf("Name: %s\n", name);
        }
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        xloc -= 0.001f;
    }

```

```

        zloc += 0.01f;
        yloc += yVel;
        yVel -= 0.001f;
        if (yloc <= floor) {
            yVel = 0.01f;
            floor -= 0.1;
        }
        if (zloc > 1) {
            floor = 0.05f;
            yVel = -0.01f;
            yloc = 0.5f;
            xloc = 0.1f;
            zloc = -0.5f;
        }

        translateTransform.setTranslation(new Vector3f(xloc,yloc,zloc));

        var rotation = new Transform3D();
        rotation.rotX(0.1);
        translateTransform.mul(rotation);

        ball.setTransform(translateTransform);
    }
}

```

Результат



