



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”
тема: “Імпорт тривімирних моделей у середовище програмування java3D,
обробка та маніпуляція цих зображень”

Виконав
студент III курсу
групи КП-83
Ландо Максим Юрійович
варіант № 9

Зарахована
“ ____ ” “ ____ ” 20__ р.
викладачем
Шкурат Оксаною Сергіївною

Київ 2021

Мета:

1. Здобути навички імпорту моделей, побудованих у тривимірних редакторах, (об'єктів форматів .obj, .lwo, .3ds) до бібліотеки java3D.
2. Навчитися анімувати імпортовані об'єкти.

Задання 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: Ракета

Лістинг коду програми

Rocket.java

```
package app;

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

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

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.Map;

import javax.swing.JFrame;

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

    private static TransformGroup rocket;

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

        root = new BranchGroup();

        addImageBackground();

        addDirectionalLightToUniverse();
        addAmbientLightToUniverse();

        ChangeViewAngle();

        rocket = getRocketGroup();
        root.addChild(rocket);

        root.compile();
        universe.addBranchGraph(root);
    }

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

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

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

    private void addImageBackground() {
        TextureLoader t = new TextureLoader("data/space.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(new Point3d(0.0, 0.0, 0.0), 1000000.0);
        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, 6));
        vpGroup.setTransform(vpTranslation);
    }

    private TransformGroup getRocketGroup() throws IOException {
        Shape3D shape = getModelShape3D("gas_tank", "data/rocket/gas-tank.obj");
        System.out.println(shape);
        Transform3D transform3D = new Transform3D();
        transform3D.setScale(new Vector3d(0.3, 0.3, 0.3));
        Transform3D transform3D2 = new Transform3D();
        TransformGroup group = getModelGroup(shape);
        group.removeAllChildren();
        TransformGroup tg = new TransformGroup();
        tg.setTransform(transform3D2);
        group.setTransform(transform3D);
        group.addChild(tg);
        tg.addChild(shape);

        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);
        Map<String, Shape3D> map = scene.getNamedObjects();
        Shape3D shape = map.get(name);
        scene.getSceneGroup().removeChild(shape);
        return shape;
    }

    private Scene getSceneFromFile(String path) throws IOException {
        ObjectFile file = new ObjectFile(ObjectFile.RESIZE);

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

    public static void main(String[] args) {
        try {
            Rocket window = new Rocket();
            RocketAnimation rocketMovement = new RocketAnimation(rocket);
            canvas.addKeyListener(rocketMovement);
            window.setVisible(true);
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }
}

```

RocketAnimation.java

```
package app;
```

```

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 RocketAnimation implements ActionListener, KeyListener {
    private TransformGroup rocket;
    private Transform3D transform3D = new Transform3D();

    private float x = 0;
    private float y = 0;

    private boolean w = false;
    private boolean s = false;
    private boolean a = false;
    private boolean d = false;

    public RocketAnimation(TransformGroup rocket) {
        this.rocket = rocket;
        this.rocket.setTransform(this.transform3D);

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

    private void Move() {
        if (w) {
            y += 0.02f;
        }
        if (s) {
            y -= 0.02f;
        }
        if (a) {
            x -= 0.02f;
        }
        if (d) {
            x += 0.02f;
        }
        transform3D.setTranslation(new Vector3f(x, y, 0));

        rocket.setTransform(transform3D);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        Move();
    }

    @Override
    public void keyPressed(KeyEvent ev) {
        switch (ev.getKeyChar()) {
            case 'w':
                w = true;
                break;
            case 's':
                s = true;
                break;
            case 'a':
                a = true;
                break;
            case 'd':
                d = true;
                break;
        }
    }

    @Override
    public void keyTyped(KeyEvent e) {
    }
}

```

```
@Override
public void keyReleased(KeyEvent ev) {
    switch (ev.getKeyChar()) {
        case 'w':
            w = false;
            break;
        case 's':
            s = false;
            break;
        case 'a':
            a = false;
            break;
        case 'd':
            d = false;
            break;
    }
}
```

Результат

