

**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 №4
з дисципліни “Математичні та алгоритмічні основи комп'ютерної графіки”

Виконала
студентка 3 курсу
групи КП-83:
Тиченко Анна
Варіант: 20

Київ 2021

Завдання:

За допомогою засобів, що надає бібліотека Java3D, побудувати тривимірний об'єкт. Для цього скористатися основними примітивами, що буде доцільно використовувати згідно варіанту: сфера, конус, паралелепіпед, циліндр. Об'єкт має складатися з 5-15 примітивів. Задати матеріал кожного примітиву, в разі необхідності накласти текстуру. В сцені має бути мінімум одне джерело освітлення. Виконати анімацію сцени таким чином, щоб можна було розглянути об'єкт з усіх сторін. За бажанням можна виконати інтерактивні взаємодію з об'єктом за допомогою миші та клавіатури.

Код програми

Main.java

```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import com.sun.j3d.utils.geometry.*;
import com.sun.j3d.utils.image.TextureLoader;
import com.sun.j3d.utils.universe.SimpleUniverse;
import javax.media.j3d.*; import javax.swing.Timer; import
javax.vecmath.*;

public class Main implements ActionListener {
    private TransformGroup planeTransformGroup;
    private Transform3D planeTransform3D = new Transform3D();
    private Timer timer;
    private float angle = 0;

    public static void main(String[] args) {
        new Main();
    }

    public Main() {
        timer = new Timer(50, this);
        timer.start();
        BranchGroup scene = createSceneGraph();
        SimpleUniverse u = new SimpleUniverse();
        u.getViewingPlatform().setNominalViewingTransform();
        u.addBranchGraph(scene);
    }

    public BranchGroup createSceneGraph() {
```

```

        // створюємо групу об'єктів
        BranchGroup objRoot = new BranchGroup();

        planeTransformGroup = new TransformGroup();

planeTransformGroup.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        buildPlane();
        TextureLoader loader = new TextureLoader("D:/sky.jpg",
"LUMINANCE", new Container());
        var texture = loader.getImage();
        Background background = new Background(texture);
        background.setImageScaleMode(Background.SCALE_FIT_MAX);
        background.setCapability(Background.ALLOW_IMAGE_WRITE);
        objRoot.addChild(planeTransformGroup);
        objRoot.addChild(background);

        // налаштовуємо освітлення
        BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0,
0.0),100.0);
        Color3f light1Color = new Color3f(1.0f, 0.5f, 0.4f);
        Vector3f light1Direction = new Vector3f(4.0f, -7.0f, -12.0f);
        DirectionalLight light1 = new DirectionalLight(light1Color,
light1Direction);
        light1.setInfluencingBounds(bounds);
        objRoot.addChild(light1);

        // встановлюємо навколишнє освітлення
        Color3f ambientColor = new Color3f(1.0f, 1.0f, 1.0f);
        AmbientLight ambientLightNode = new AmbientLight(ambientColor);
        ambientLightNode.setInfluencingBounds(bounds);
        objRoot.addChild(ambientLightNode);
        return objRoot;
    }

    private void buildPlane() {    // створюємо гілки ялинки
        TransformGroup tgTop = new TransformGroup();
        Transform3D transformTop = new Transform3D();
        Cone coneTop = PlaneBody.getCone(0.8f, 0.1f);
        Vector3f vectorTop = new Vector3f(.0f, 0.2f, .0f);
        transformTop.setTranslation(vectorTop);
        transformTop.rotX(Math.PI/2);
        tgTop.setTransform(transformTop);
        tgTop.addChild(coneTop);
        planeTransformGroup.addChild(tgTop);
        createHead(0.1f, .0f, .0f, -0.4f, "", new
Color3f(Color.DARK_GRAY));
        createTail(.0f, .0f, 0.3f, "", new Color3f(Color.DARK_GRAY));
        createWings(.0f, .04f, -0.2f, "", new
Color3f(Color.DARK_GRAY));
    }

    private void createHead(float radius, float x, float y, float z,

```

```

String picture, Color3f emissive) {
    TransformGroup tg = new TransformGroup();
    Transform3D transform = new Transform3D();
    Sphere cone = PlaneDetails.getSphere(radius, picture, emissive);
    Vector3f vector = new Vector3f(x, y, z);
    transform.setTranslation(vector);
    tg.setTransform(transform);
    tg.addChild(cone);
    planeTransformGroup.addChild(tg);
}

private void createTail(float x, float y, float z, String picture,
Color3f emissive) {
    TransformGroup tg = new TransformGroup();
    Transform3D transform = new Transform3D();
    Box box = PlaneDetails.getBox(0.2f,0.01f,0.1f, picture,
emissive);
    Vector3f vector = new Vector3f(x, y, z);
    transform.setTranslation(vector);
    tg.setTransform(transform);
    tg.addChild(box);
    planeTransformGroup.addChild(tg);
}

private void createWings(float x, float y, float z, String picture,
Color3f emissive) {
    TransformGroup tg = new TransformGroup();
    Transform3D transform = new Transform3D();
    Box box = PlaneDetails.getBox(0.7f,0.01f,0.15f, picture,
emissive);
    Vector3f vector = new Vector3f(x, y, z);
    transform.rotX(Math.PI/21);
    transform.setTranslation(vector);
    tg.setTransform(transform);
    tg.addChild(box);
    planeTransformGroup.addChild(tg);
}

@Override public void actionPerformed(ActionEvent e) {
    planeTransform3D.rotY(angle);
    planeTransformGroup.setTransform(planeTransform3D);
    angle += 0.05;
}
}

```

Результати роботи програми

