



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 “Побудова найпростіших тривимірних об’єктів за допомогою
бібліотеки Java3D та їх анімація”

Виконав студент III курсу групи КП-82

Анікєєв Ігор Анатолійович
Зараховано: Шкурат Оксаною Сергіївною

варіант № 1

Київ 2021

Варіант завдання

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.

Варіант: 1 літак

Лістинг коду програми

```
package sample; import

com.sun.j3d.utils.universe.SimpleUniverse;

import javax.media.j3d.BranchGroup;
import javax.media.j3d.Canvas3D;
import javax.swing.*; import
java.awt.event.ActionEvent; import
java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;

public class Main extends JFrame implements ActionListener, KeyListener
    { Plane plane; float angleX = 0; float angleY = 0;

    public Main() { super("Lab4"); plane = new Plane(); Canvas3D canvas3D = new
    Canvas3D(SimpleUniverse.getPreferredConfiguration());

    add(canvas3D);
    canvas3D.addKeyListener(this);

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

    BranchGroup scene = plane.createSceneGraph();
    SimpleUniverse u = new SimpleUniverse(canvas3D);
    u.getViewingPlatform().setNominalViewingTransform();
    u.addBranchGraph(scene);

    setSize(800, 800);
    setLocationRelativeTo(null);
    setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    setVisible(true); } public static void main(String[]
    args) { new Main(); }

    @Override
    public void keyPressed(KeyEvent keyEvent) { }

    @Override
    public void keyTyped(KeyEvent keyEvent) { }

    @Override
    public void keyReleased(KeyEvent e) { }

    @Override
    public void actionPerformed(ActionEvent e) {

    if(keyEvent.getKeyCode() == KeyEvent.VK_S) {
        angleX -= 0.02;
        plane.rotate(angleX, angleY);
    }

    if(keyEvent.getKeyCode() == KeyEvent.VK_W) {
        angleX += 0.02;
        plane.rotate(angleX, angleY);
    }

    if(keyEvent.getKeyCode() == KeyEvent.VK_A) {
        angleY -= 0.02;
        plane.rotate(angleX, angleY);
    }

    if(keyEvent.getKeyCode() == KeyEvent.VK_D) {
        angleY += 0.02;
        plane.rotate(angleX, angleY);
    }
    }
```

```

    }
}

package sample;

import com.sun.j3d.utils.geometry.Box;
import com.sun.j3d.utils.geometry.Cone;
import com.sun.j3d.utils.geometry.Cylinder;
import
com.sun.j3d.utils.geometry.Primitive;

import javax.media.j3d.*;
import javax.vecmath.Color3f;
import javax.vecmath.Point3d;
import javax.vecmath.Point3f;
import javax.vecmath.Vector3f;

public class Plane { private static int primflags =
    Primitive.GENERATE_NORMALS +
    Primitive.GENERATE_TEXTURE_COORDS; private TransformGroup
    objectTransformGroup; private Transform3D
    planeTransform3D = new Transform3D();

    public BranchGroup createSceneGraph() {
        BranchGroup objRoot = new BranchGroup();

        objectTransformGroup = new TransformGroup();
        objectTransformGroup.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        makeObj();
        objRoot.addChild(objectTransformGroup);

        //налаштуємо освітлення
        BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0),
            100.0);
        Color3f light1Color = new Color3f(0.6f, 0.6f, 0.8f);
        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(1f, 1f, 1f);
        AmbientLight ambientLightNode = new AmbientLight(ambientColor);
        ambientLightNode.setInfluencingBounds(bounds);
        objRoot.addChild(ambientLightNode);

        return objRoot;
    }

    public void makeObj() {
        Appearance ap = new Appearance();
        Color3f emissive = new Color3f(0.0f, 0.0f, 0.0f);
        Color3f ambient = new Color3f(0.5f, 0.5f, 0.5f);
        Color3f diffuse = new Color3f(0.1f, 0.1f, 0.1f);
        Color3f specular = new Color3f(0.3f, 0.3f, 0.3f);
        ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));

        {
            Transform3D transform3D = new Transform3D();
            TransformGroup transformGroup = new TransformGroup();
            transform3D.rotZ(Math.PI/2);
            transform3D.setTranslation(new Vector3f(-0.3f, 0.0f,
                0.0f)); transformGroup.setTransform(transform3D); Cone nose
                = new Cone(0.09f, 0.19f, primflags, ap);

            transformGroup.addChild(nose);
            objectTransformGroup.addChild(transformGroup);
        }
    }
}

```

```

{
    Transform3D transform3D = new Transform3D();
    TransformGroup transformGroup = new TransformGroup();
    transform3D.rotZ(-Math.PI/2);
    transform3D.setTranslation(new Vector3f(0.1f, 0.0f,
    0.0f)); transformGroup.setTransform(transform3D); Cone
    body = new Cone(0.09f, 0.62f, primflags, ap);

    transformGroup.addChild(body);
    objectTransformGroup.addChild(transformGroup);
}

{
    Transform3D transform3D = new Transform3D();
    TransformGroup transformGroup = new TransformGroup();
    transform3D.rotZ(-Math.PI/3);
    transform3D.setTranslation(new Vector3f(0.1f, 0.15f, 0.0f));
    transformGroup.setTransform(transform3D);
    Box wing1 = new Box(0.1f, 0.3f, 0.006f, primflags, ap);
    transformGroup.addChild(wing1);
    objectTransformGroup.addChild(transformGroup);
}

{
    Transform3D transform3D = new Transform3D();
    TransformGroup transformGroup = new TransformGroup();
    transform3D.rotZ(Math.PI/3);
    transform3D.setTranslation(new Vector3f(0.1f, -0.15f, 0.0f));
    transformGroup.setTransform(transform3D);
    Box wing2 = new Box(0.1f, 0.3f, 0.006f, primflags, ap);
    transformGroup.addChild(wing2);
    objectTransformGroup.addChild(transformGroup);
}

{
    Transform3D transform3D = new Transform3D();
    TransformGroup transformGroup = new TransformGroup();
    transform3D.rotZ(Math.PI/4);
    transform3D.setTranslation(new Vector3f(0.2f, 0.0f, 0.0f));
    transformGroup.setTransform(transform3D);
    Box tail = new Box(0.15f, 0.15f, 0.006f, primflags, ap);
    transformGroup.addChild(tail);
    objectTransformGroup.addChild(transformGroup);
}

{
    Transform3D transform3D = new Transform3D();
    TransformGroup transformGroup = new TransformGroup();
    transform3D.rotZ(Math.PI/2);
    transform3D.setTranslation(new Vector3f(0.06f, 0.130f, 0.0f));
    transformGroup.setTransform(transform3D);
    Cylinder turbine1 = new Cylinder(0.030f, 0.5f, primflags, ap);
    transformGroup.addChild(turbine1);
    objectTransformGroup.addChild(transformGroup);
}

{
    Transform3D transform3D = new Transform3D();
    TransformGroup transformGroup = new TransformGroup();
    transform3D.rotZ(Math.PI/2);
    transform3D.setTranslation(new Vector3f(0.06f, -0.130f, 0.0f));
    transformGroup.setTransform(transform3D);
    Cylinder turbine2 = new Cylinder(0.030f, 0.5f, primflags, ap);
    transformGroup.addChild(turbine2);
    objectTransformGroup.addChild(transformGroup);
}

{
    Transform3D transform3D = new Transform3D();

```

```

        TransformGroup transformGroup = new TransformGroup();
        transform3D.rotZ(Math.PI/2);
        transform3D.setTranslation(new Vector3f(-0.2f, -0.130f, 0.0f));
        transformGroup.setTransform(transform3D);
        Cone cone1 = new Cone(0.025f, 0.04f, primflags, ap);
        transformGroup.addChild(cone1);
        objectTransformGroup.addChild(transformGroup);
    }

    {
        Transform3D transform3D = new Transform3D();
        TransformGroup transformGroup = new TransformGroup();
        transform3D.rotZ(Math.PI/2);
        transform3D.setTranslation(new Vector3f(-0.2f, 0.130f, 0.0f));
        transformGroup.setTransform(transform3D);
        Cone cone2 = new Cone(0.025f, 0.04f, primflags, ap);
        transformGroup.addChild(cone2);
        objectTransformGroup.addChild(transformGroup);
    }

    {
        Transform3D transform3D = new Transform3D();
        TransformGroup transformGroup = new TransformGroup();
        transform3D.rotY(Math.PI/4);
        transform3D.setTranslation(new Vector3f(0.3f, 0.130f, 0.05f));
        transformGroup.setTransform(transform3D);
        Box part1 = new Box(0.03f, 0.006f, 0.06f, primflags, ap);
        transformGroup.addChild(part1);
        objectTransformGroup.addChild(transformGroup);
    }

    {
        Transform3D transform3D = new Transform3D();
        TransformGroup transformGroup = new TransformGroup();
        transform3D.rotY(Math.PI/4);
        transform3D.setTranslation(new Vector3f(0.3f, -0.130f, 0.05f));
        transformGroup.setTransform(transform3D);
        Box part2 = new Box(0.03f, 0.006f, 0.06f, primflags, ap);
        transformGroup.addChild(part2);
        objectTransformGroup.addChild(transformGroup);
    }
}

public void rotate(float angleX, float angleY) {
    Transform3D rotX = new Transform3D();
    Transform3D rotY = new Transform3D();

    rotX.rotX(angleX);
    rotY.rotZ(angleY);
    rotX.mul(rotY);
    objectTransformGroup.setTransform(rotX);
}
}

```

Результат

