



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”

Виконав  
студент III курсу  
групи КП-81

Дикий Ілля  
*(прізвище, ім’я, по батькові)*

Зарахована  
“ \_\_\_\_ ” “ \_\_\_\_ ” 20 \_\_\_\_ р.  
викладачем

Шкурат Оксаною Сергіївною  
*(прізвище, ім’я, по батькові)*

варіант № 3

## **Завдання**

**Тема:** «Побудова найпростіших тривимірних об'єктів за допомогою бібліотеки Java3D та їх анімація»

**Мета:**

1. вивчення стандартних засобів Java3D для візуалізації зображення;
2. вивчення засобів анімації примітивів та складених об'єктів в Java3D.

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 взаємодії з об'єктом за допомогою миші та клавіатури.

## Лістинг коду програми

```
import java.awt.Container;
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 PingPongTable implements ActionListener {

    private TransformGroup tableTransformGroup;
    private Transform3D tableTransform3D = new Transform3D();
    private Timer timer;
    float angle = 0;

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

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

        BranchGroup scene = createSceneGraph();
        SimpleUniverse u = new SimpleUniverse();
        TransformGroup cameraTG =
u.getViewingPlatform().getViewPlatformTransform();
        Vector3f translate = new Vector3f();
        Transform3D T3D = new Transform3D();
        translate.set( 0.0f, 0.5f, 4.0f);
        T3D.setTranslation(translate);
        cameraTG.setTransform(T3D);

        u.addBranchGraph(scene);
    }

    public BranchGroup createSceneGraph() {
        // створюємо групу об'єктів
```

```

BranchGroup objRoot = new BranchGroup();
// створюємо об'єкт, що будемо додавати до групи
tableTransformGroup = new TransformGroup();

tableTransformGroup.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
buildTable();
objRoot.addChild(tableTransformGroup);

// 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(3.0f, -3.0f, 3.0f);
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 buildTable() {

    moveBox(0.6f, .0f, 0.36f, getLeg());
    moveBox(-0.6f, .0f, 0.36f, getLeg());
    moveBox(0.6f, .0f, -0.36f, getLeg());
    moveBox(-0.6f, .0f, -0.36f, getLeg());

    int primflags = Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS;

    Appearance apDesk = new Appearance();
    Color3f emissive = new Color3f(0.0f, 0.05f, 0.0f);
    Color3f ambient = new Color3f(0.2f, 0.3f, 0.15f);
    Color3f diffuse = new Color3f(0.2f, 0.15f, .15f);
    Color3f specular = new Color3f(0.0f, 0.1f, 0.0f);

```

```

        apDesk.setMaterial(new Material(ambient, emissive, diffuse, specular,
1.0f));
        moveBox(.0f, 0.38f, .0f, new Box(1.37f, 0.01f, 0.76f, primflags,
apDesk));

//          // завантажуюмо текстуру
        TextureLoader loader = new
TextureLoader("/home/stilpert/Education/MAOKG/lab4/src/net.jpg", "LUMINANCE",
new Container());
        Texture texture = loader.getTexture();
        // задаємо властивості границі текстури
        texture.setBoundaryModeS(Texture.WRAP);
        texture.setBoundaryModeT(Texture.WRAP);
        texture.setBoundaryColor(new Color4f(.0f, .0f, .0f, 0.0f));
        // встановлюємо атрибути текстури
        TextureAttributes texAttr = new TextureAttributes();
        texAttr.setTextureMode(TextureAttributes.MODULATE);
        // створюємо новий вигляд
        Appearance apNet = new Appearance();
        apNet.setTexture(texture);
        apNet.setTextureAttributes(texAttr);
        Color3f emissive1 = new Color3f(0.2f, 0.2f, 0.2f);
        Color3f ambient1 = new Color3f(0.3f, 0.3f, 0.3f);
        Color3f diffuse1 = new Color3f(0.2f, 0.15f, .15f);
        Color3f specular1 = new Color3f(0.5f, 0.5f, 0.5f);
        apNet.setMaterial(new Material(ambient1, emissive1, diffuse1,
specular1, 1.0f));
        moveBox(.0f, 0.5f, .0f, new Box(0.003f, 0.076f, 0.91f, primflags,
apNet));

    }

private void moveBox(float x, float y, float z, Box box){
    TransformGroup trBox = new TransformGroup();
    Transform3D transformBox = new Transform3D();
    Vector3f vectorBox = new Vector3f(x, y, z);
    transformBox.setTranslation(vectorBox);
    trBox.setTransform(transformBox);
    trBox.addChild(box);
    tableTransformGroup.addChild(trBox);
}

```

```

private Box getLeg(){
    int primflags = Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS;
    Appearance ap = new Appearance();
    Color3f emissive = new Color3f(0.01f, 0.01f, 0.01f);
    Color3f ambient = new Color3f(0.2f, 0.2f, 0.2f);
    Color3f diffuse = new Color3f(0.14f, 0.15f, .15f);
    Color3f specular = new Color3f(0.1f, 0.1f, 0.1f);
    ap.setMaterial(new Material(ambient, emissive, diffuse, specular,
1.0f));
    return new Box(0.03f, 0.38f, 0.03f,  primflags, ap);
}

@Override
public void actionPerformed(ActionEvent e) {
    tableTransform3D.rotY(angle);
    tableTransformGroup.setTransform(tableTransform3D);
    angle += 0.02;
}
}

```

## Результати

