



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”  
на тему “Побудова найпростіших тривимірних об'єктів за допомогою  
бібліотеки Java3D та їх анімація”

Виконав  
студент III курсу  
групи КП-82

Новохацький Владислав  
*(прізвище, ім'я, по батькові)*

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

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

варіант № 15

## Варіант завдання

### Завдання

За допомогою примітивів *JavaFX* максимально реально зобразити персонажа за варіантом та виконати його 2D анімацію. Для анімації скористатися стандартними засобами бібліотеки *JavaFX*.

Обов'язковою є реалізація таких видів анімації:

- 1) переміщення;
- 2) поворот;
- 3) масштабування.

### Варіант 15:

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.

### 15. Качелі

### Лістинг коду програми

```
package lab4;

import java.awt.Container;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.Timer;
import javax.media.j3d.AmbientLight;
```

```
import javax.media.j3d.Appearance;
import javax.media.j3d.BoundingSphere;
import javax.media.j3d.BranchGroup;
import javax.media.j3d.DirectionalLight;
import javax.media.j3d.Material;
import javax.media.j3d.Texture;
import javax.media.j3d.TextureAttributes;
import javax.media.j3d.Transform3D;
import javax.media.j3d.TransformGroup;
import javax.vecmath.Color3f;
import javax.vecmath.Point3d;
import javax.vecmath.Vector3d;
import javax.vecmath.Vector3f;
import com.sun.j3d.utils.geometry.Box;
import com.sun.j3d.utils.geometry.Cylinder;
import com.sun.j3d.utils.geometry.Primitive;
import com.sun.j3d.utils.image.TextureLoader;
import com.sun.j3d.utils.universe.SimpleUniverse;
```

```
public class Main implements ActionListener {
```

```
    private TransformGroup swing;
    private TransformGroup swingP;
    private Transform3D transform3D = new Transform3D();
    private Transform3D transform3D2 = new Transform3D();
    private Timer timer;
    private float angle = 0;
    private float angleP = 0;
    private float step = 3;
```

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

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

```
    public BranchGroup getSwingGroup() {
        BranchGroup group = new BranchGroup();
        swing = new TransformGroup();
```

```

swingP = new TransformGroup();
swing.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
swingP.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
group.addChild(swing);

swing.addChild(createBox(new float[] {0.02f,0.7f,0.02f}, new float[]
{0.45f,0f,(float) (0.7*Math.sin(20*Math.PI/180))}, -20, new Color3f(1f,.15f,.15f)));
swing.addChild(createBox(new float[] {0.02f,0.7f,0.02f}, new float[]
{0.45f,0f,(float) (-0.7*Math.sin(20*Math.PI/180))}, 20, new Color3f(1f,.15f,.15f)));

swing.addChild(createBox(new float[] {0.02f,0.7f,0.02f}, new float[]
{-0.45f,0f,(float) (0.7*Math.sin(20*Math.PI/180))}, -20, new Color3f(1f,.15f,.15f)));
swing.addChild(createBox(new float[] {0.02f,0.7f,0.02f}, new float[]
{-0.45f,0f,(float) (-0.7*Math.sin(20*Math.PI/180))}, 20, new Color3f(1f,.15f,.15f)));

Cylinder cyl = new Cylinder(0.02f,0.9f, getWAppearance(new
Color3f(1f,.15f,.15f)));
TransformGroup ctg = new TransformGroup();
Transform3D ct = new Transform3D();
Transform3D ct2 = new Transform3D();
Vector3f cv = new Vector3f(.0f,0.65f,.0f);
ct.setTranslation(cv);
ct2.rotZ(90*Math.PI/180);
ct.mul(ct2);
ctg.setTransform(ct);
ctg.addChild(cyl);
swing.addChild(ctg);

swingP.addChild(createBox(new float[] {0.01f,0.5f,0.01f}, new float[]
{-0.25f,0.15f,0f},0, new Color3f(1f,.6f,.0f)));
swingP.addChild(createBox(new float[] {0.01f,0.5f,0.01f}, new float[]
{0.25f,0.15f,0f},0, new Color3f(1f,.6f,.0f)));

swingP.addChild(createBox(new float[] {0.3f,0.015f,0.15f}, new float[]
{0f,-0.35f,0f},0, null));

swing.addChild(swingP);

BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0,
0.0),100.0);
Color3f lightColor = new Color3f(0.9f, 0.9f, 0.6f);
Vector3f lightDirection = new Vector3f(-5.0f, -5.0f, -5.0f);
DirectionalLight light = new DirectionalLight(lightColor,lightDirection);
light.setInfluencingBounds(bounds);
group.addChild(light);
AmbientLight ambientLight = new AmbientLight(new Color3f(.5f, .5f, .5f));

```

```

        ambientLight.setInfluencingBounds(bounds);
        group.addChild(ambientLight);

        return group;
    }

    private TransformGroup createBox(float[] size, float[] vec, float rotation,
    Color3f emissive) {
        Box box = null;
        if (emissive==null) {
            TextureLoader loader = new TextureLoader("./texture.jpg",
"LUMINANCE", new Container());

            Texture texture = loader.getTexture();
            texture.setBoundaryModeS(Texture.WRAP);
            texture.setBoundaryModeT(Texture.WRAP);
            TextureAttributes texAttr = new TextureAttributes();
            texAttr.setTextureMode(TextureAttributes.MODULATE);
            Appearance ap = new Appearance();
            Color3f emis = new Color3f(0.2f, 0.0f, 0.0f);
            Color3f ambient = new Color3f(0.5f, 0.5f, 0.5f);
            Color3f diffuse = new Color3f(0.2f, 0.15f, .15f);
            Color3f specular = new Color3f(.2f, .2f, .2f);
            ap.setMaterial(new Material(ambient, emis, diffuse, specular,
1.0f));

            ap.setTexture(texture);
            ap.setTextureAttributes(texAttr);
            int primflags = Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS;
            box = new Box(size[0], size[1], size[2], primflags, ap);
        } else {
            box = new Box(size[0], size[1], size[2],
getWAppearance(emissive));
        }
        TransformGroup tg = new TransformGroup();
        Transform3D transform = new Transform3D();
        Transform3D transform2 = new Transform3D();
        Vector3f vector = new Vector3f(vec[0], vec[1], vec[2]);
        transform.setTranslation(vector);
        if (rotation!=0) {
            transform2.rotX(rotation*Math.PI/180);
            transform.mul(transform2);
        }
        tg.setTransform(transform);
        tg.addChild(box);
        return tg;
    }
}

```

```

@Override
public void actionPerformed(ActionEvent e) {
    transform3D.rotY(angle);
    swing.setTransform(transform3D);
    angle += 0.05;

    Transform3D t = new Transform3D();
    Transform3D t1 = new Transform3D();
    t1.setTranslation(new Vector3d(0.0, -0.6f, 0.0));
    t.mul(t1,t);
    Transform3D rotate = new Transform3D();
    rotate.rotX(Math.PI/180*angleP);
    t.mul(rotate,t);
    Transform3D t2 = new Transform3D();
    t2.setTranslation(new Vector3d(0.0, 0.6f, 0.0));
    t.mul(t2,t);

    swingP.setTransform(t);

    angleP += step;
    if (Math.abs(angleP)>30) {
        step=-step;
    }
}

private static Appearance getWAppearance(Color3f specular) {
    Appearance ap = new Appearance();
    Color3f emissive = new Color3f(0.0f, 0.0f, 0.0f);
    Color3f ambient = new Color3f(0.2f, 0.2f, 0.2f);
    Color3f diffuse = new Color3f(0.2f, 0.15f, .15f);
    if (specular == null)
        specular = new Color3f(0.0f, 0.8f, 0.0f);
    ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));
    return ap;
}
}

```

## Результат

