

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО  
Факультет прикладної математики  
Кафедра програмного забезпечення комп'ютерних систем

**ЗВІТ**  
з лабораторної роботи № 4  
Варіант 8

Виконав:  
студент 3-го курсу,  
групи КП-82,  
спеціальності 121 –  
Інженерія  
програмного  
забезпечення  
Клапатюк Олександр  
Петрович

Київ – 2020

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.

## SnowMan.java

```
public class SnowMan implements ActionListener {
    private TransformGroup carTransformGroup = new TransformGroup();
    private Transform3D carTransform3D = new Transform3D();
    private Timer timer;
    private float angle= 0;

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

    private SnowMan() {
        timer = new Timer(50, this);
        timer.start();
        // створення групи об'єктів branch Group
        BranchGroup scene = createSceneGraph();
        SimpleUniverse u = new SimpleUniverse();
        //встановлення точки перегляду
        u.getViewingPlatform().setNominalViewingTransform();
        u.addBranchGraph(scene);
    }

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

        carTransformGroup = new TransformGroup();
        carTransformGroup.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        buildCastleSkeleton();
        objRoot.addChild(carTransformGroup);
        // встановлення кольору заднього фону
        Background background = new Background(new Color3f(0f, 1.0f, 1.0f));
        // вказуємо сферу внутрішній простір, якої буде освітлено
        BoundingSphere sphere = new BoundingSphere(new Point3d(0,0,0), 100000);
        background.setApplicationBounds(sphere);
        objRoot.addChild(background);
        // вказуємо сферу внутрішній простір, якої буде освітлено
        BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0),100.0);

        Color3f light1Color = new Color3f(1.0f, 0.5f, 0.4f);
        //визначаємо напрям поширення світла
        Vector3f light1Direction = new Vector3f(.8f, .8f, .0f);
        // власне об'єкт освітлення
        DirectionalLight light1 = new DirectionalLight(light1Color, light1Direction);

        // встановлення того, яка частина сфери має бути освітлена
        light1.setInfluencingBounds(bounds);
        objRoot.addChild(light1);

        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 buildCastleSkeleton() {
        Sphere body1 = SnowManBody.getPart(.3f);
        Transform3D body1T = new Transform3D();
```

```

body1T.setTranslation(new Vector3f(.0f, -.4f, .0f));
TransformGroup body1TG = new TransformGroup();
body1TG.setTransform(body1T);
body1TG.addChild(body1);
carTransformGroup.addChild(body1TG);

```

```

Sphere body2 = SnowManBody.getPart(.25f);
Transform3D body2T = new Transform3D();
body2T.setTranslation(new Vector3f(.0f, .0f, .0f));
TransformGroup body2TG = new TransformGroup();
body2TG.setTransform(body2T);
body2TG.addChild(body2);
carTransformGroup.addChild(body2TG);

```

```

Sphere body3 = SnowManBody.getPart(.2f);
Transform3D body3T = new Transform3D();
body3T.setTranslation(new Vector3f(.0f, .35f, .0f));
TransformGroup body3TG = new TransformGroup();
body3TG.setTransform(body3T);
body3TG.addChild(body3);
carTransformGroup.addChild(body3TG);

```

```

Cylinder body4 = SnowManBody.getHat(.2f, .05f);
Transform3D body4T = new Transform3D();
body4T.setTranslation(new Vector3f(.0f, .5f, .0f));
TransformGroup body4TG = new TransformGroup();
body4TG.setTransform(body4T);
body4TG.addChild(body4);
carTransformGroup.addChild(body4TG);

```

```

Cylinder body5 = SnowManBody.getHat(.1f, .2f);
Transform3D body5T = new Transform3D();
body5T.setTranslation(new Vector3f(.0f, .60f, .0f));
TransformGroup body5TG = new TransformGroup();
body5TG.setTransform(body5T);
body5TG.addChild(body5);
carTransformGroup.addChild(body5TG);

```

```

Cone body6 = SnowManBody.getCarrot(0.05f, 0.2f);
Transform3D body6T = new Transform3D();
body6T.rotX(Math.PI/2);
body6T.setTranslation(new
Vector3f(.0f, .3f, .25f));
TransformGroup body6TG = new TransformGroup();
body6TG.setTransform(body6T);
body6TG.addChild(body6);
carTransformGroup.addChild(body6TG);

```

```

createButton(.0f, -.1f, .21f);
createButton(.0f, .0f, .23f);
createButton(.0f, .1f, .21f);

```

```

createButton(-.05f, .4f, .16f);
createButton(.05f, .4f, .16f);

```

```

}

```

```

private void createButton(float x, float y, float z) {
    TransformGroup tg = new TransformGroup();
    Transform3D transform = new Transform3D();
    transform.rotX(Math.PI/2);
    Cylinder button = SnowManBody.getButton(.02f);
    Vector3f vector = new Vector3f(x, y, z);
    transform.setTranslation(vector);
    tg.setTransform(transform);
    tg.addChild(button);
    carTransformGroup.addChild(tg);
}

```

```

@Override
public void actionPerformed(ActionEvent e) {
    carTransform3D.rotY(angle);
    carTransformGroup.setTransform(carTransform3D);
    angle += 0.05;
}

```

# SnowManBody.java

```
public class SnowManBody {
    public static Sphere getPart(float radius) {
        int primflags = Primitive.GENERATE_NORMALS + Primitive.GENERATE_TEXTURE_COORDS;
        return new Sphere(radius, primflags, getPartAppearance());
    }

    public static Cylinder getButton(float radius) {
        int primflags = Primitive.GENERATE_NORMALS + Primitive.GENERATE_TEXTURE_COORDS;
        return new Cylinder(radius, 0.05f, primflags, getButtonAppearance());
    }

    public static Cone getCarrot(float radius, float height) {
        int primflags = Primitive.GENERATE_NORMALS + Primitive.GENERATE_TEXTURE_COORDS;
        return new Cone(radius, height, primflags, getCarrotAppearance());
    }

    public static Cylinder getHat(float radius, float height) {
        int primflags = Primitive.GENERATE_NORMALS + Primitive.GENERATE_TEXTURE_COORDS;
        return new Cylinder(radius, height, primflags, getHatAppearance());
    }

    private static Appearance getPartAppearance() {
        TextureLoader loader = new TextureLoader("source_folder\\snow.jpg", "LUMINANCE", new
            Container());

        Texture texture = loader.getTexture();
        texture.setBoundaryModeS(Texture.WRAP);
        texture.setBoundaryModeT(Texture.WRAP);
        texture.setBoundaryColor(new Color4f(0.0f, 1.0f, 1.0f, 0.0f));

        TextureAttributes texAttr = new TextureAttributes();
        texAttr.setTextureMode(TextureAttributes.MODULATE);

        Appearance ap = new Appearance();
        ap.setTexture(texture);
        ap.setTextureAttributes(texAttr);

        Color3f emissive = new Color3f(new Color(0,0, 0));
        Color3f ambient = new Color3f(new Color(200,200, 200));
        Color3f diffuse = new Color3f();
        Color3f specular = new Color3f(new Color(0,0, 0));
        ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));
        return ap;
    }

    private static Appearance getCarrotAppearance() {
        TextureLoader loader = new TextureLoader("source_folder\\carrot.jpg", "LUMINANCE", new
            Container());

        Texture texture = loader.getTexture();
        texture.setBoundaryModeS(Texture.WRAP);
        texture.setBoundaryModeT(Texture.WRAP);
        texture.setBoundaryColor(new Color4f(0.0f, 1.0f, 1.0f, 0.0f));

        TextureAttributes texAttr = new TextureAttributes();
        texAttr.setTextureMode(TextureAttributes.MODULATE);

        Appearance ap = new Appearance();
        ap.setTexture(texture);
        ap.setTextureAttributes(texAttr);

        Color3f emissive = new Color3f(new Color(0,0, 0));
        Color3f ambient = new Color3f(new Color(100,50, 50));
        Color3f diffuse = new Color3f();
        Color3f specular = new Color3f(new Color(0,0, 0));
        ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));
        return ap;
    }

    private static Appearance getHatAppearance() {
        TextureLoader loader = new TextureLoader("source_folder\\hat.jpg", "LUMINANCE", new
            Container());

        Texture texture = loader.getTexture();
        texture.setBoundaryModeS(Texture.WRAP);
        texture.setBoundaryModeT(Texture.WRAP);
        texture.setBoundaryColor(new Color4f(0.0f, 1.0f, 1.0f, 0.0f));

        TextureAttributes texAttr = new TextureAttributes();
```

```

texAttr.setTextureMode(TextureAttributes.MODULATE);

Appearance ap = new Appearance();
ap.setTexture(texture);
ap.setTextureAttributes(texAttr);

Color3f emissive = new Color3f(new Color(0,0, 0));
Color3f ambient = new Color3f(new Color(200,100, 100));
Color3f diffuse = new Color3f(Color.red);
Color3f specular = new Color3f(new Color(0,0, 0));
ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));
return ap;
}

private static Appearance getButtonAppearance() {
    TextureLoader loader = new TextureLoader("source_folder\\sharingan.jpg", "LUMINANCE", new
        Container());

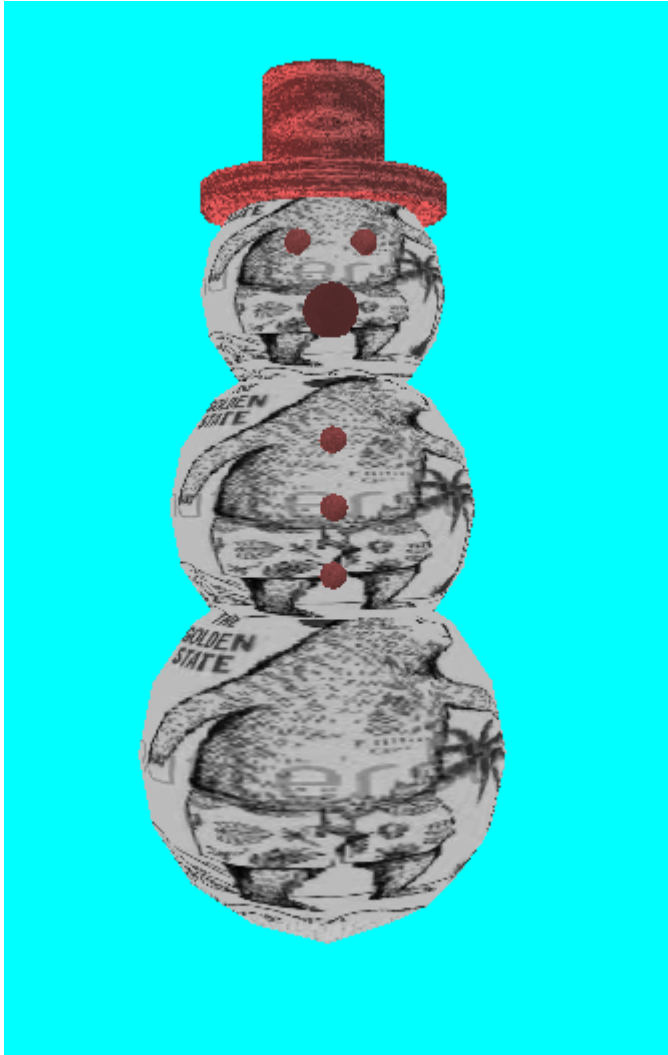
    Texture texture = loader.getTexture();
    texture.setBoundaryModeS(Texture.WRAP);
    texture.setBoundaryModeT(Texture.WRAP);
    texture.setBoundaryColor(new Color4f(0.0f, 1.0f, 1.0f, 0.0f));

    TextureAttributes texAttr = new TextureAttributes();
    texAttr.setTextureMode(TextureAttributes.MODULATE);
    Appearance ap = new Appearance();
    ap.setTexture(texture);
    ap.setTextureAttributes(texAttr);
    Color3f emissive = new Color3f(new Color(0,0, 0));
    Color3f ambient = new Color3f(new Color(200,100, 100));
    Color3f diffuse = new Color3f(Color.red);
    Color3f specular = new Color3f(new Color(0,0, 0));
    ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));
    return ap;
}
}

```

## Приклад роботи





Висновок:

- 1) вивчив стандартні засоби Java3D для візуалізації зображення;
- 2) вивчив засоби анімації примітивів та складених об'єктів в Java3D.