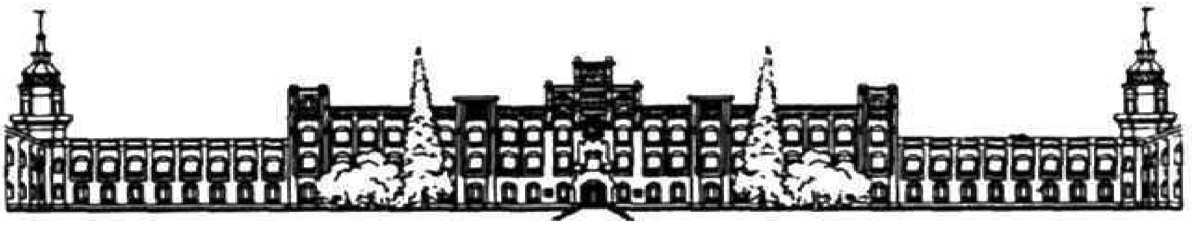

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота № 4

з дисципліни “Математичні та алгоритмічні основи комп’ютерної графіки”
тема “Побудова найпростіших тривимірних об'єктів за допомогою бібліотеки Java3D та їх анімація”

Виконав(ла)
студент(ка) III курсу
групи КП-83
Симонюк Володимир Павлович
варіант №18

Зарахована
“____” “____________” 20___ р.
Шкурат Оксаною Сергіївною

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:

18. Людина-робот

Лістинг коду програми

Main.java

```
public class Main extends JFrame implements KeyListener {
    private double movingDelta = 0.2d;
    private double angleDelta = Math.PI / 50;

    private SimpleUniverse universe;
    private Point3d watcher = new Point3d(3d, 0d, 0d);
    private Vector3d xAxis = new Vector3d(-1d, 0d, 0d);
    private Vector3d zAxis = new Vector3d(0d, 0d, 1d);

    public Main() {
        super("Lab 4");
        setLayout(new BorderLayout());
        setSize(1200, 800);
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        Canvas3D canvas = new Canvas3D(SimpleUniverse.getPreferredConfiguration());
        canvas.addKeyListener(this);

        universe = new SimpleUniverse(canvas);
        universe.addBranchGraph(createSceneGraph());
        updateViewPosition();

        add(BorderLayout.CENTER, canvas);

        setVisible(true);
    }

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

        group.addChild(new Robot(1));

        Color3f lightColor = new Color3f(1, 1, 1);
        BoundingSphere lightArea = new BoundingSphere(new Point3d(0, 0, 0), 100);

        Vector3f lightDirection1 = new Vector3f(-1, 1, -1);
        DirectionalLight light1 = new DirectionalLight(lightColor, lightDirection1);
        light1.setInfluencingBounds(lightArea);
        group.addChild(light1);

        Vector3f lightDirection2 = new Vector3f(-1, 0, 0);
        DirectionalLight light2 = new DirectionalLight(lightColor, lightDirection2);
        light2.setInfluencingBounds(lightArea);
        group.addChild(light2);

        return group;
    }

    private void updateViewPosition() {
        Transform3D viewingTransform = new Transform3D();

        Point3d lookAtPosition = new Point3d(
                watcher.x + xAxis.x,
                watcher.y + xAxis.y,
                watcher.z + xAxis.z
        );

        viewingTransform.lookAt(watcher, lookAtPosition, zAxis);
        viewingTransform.invert();


universe.getViewingPlatform().getViewPlatformTransform().setTransform(viewingTransfor
m);
    }

    private void updateEyePosition(int keyCode) {
        switch (keyCode) {
            case KeyEvent.VK_W: {
                rotateViewY(-angleDelta);
                break;
            }
            case KeyEvent.VK_A: {
                rotateViewX(-angleDelta);
                break;
            }
            case KeyEvent.VK_S: {
                rotateViewY(angleDelta);
                break;
            }
            case KeyEvent.VK_D: {
                rotateViewX(angleDelta);
                break;
            }

            case KeyEvent.VK_Z: {
                moveViewBySightDirection(movingDelta);
                break;
            }
            case KeyEvent.VK_X: {
                moveViewBySightDirection(-movingDelta);
                break;
            }

            case KeyEvent.VK_LEFT: {
                moveWatcherAroundScene(-movingDelta);
                rotateViewX(-angleDelta);
                break;
            }
            case KeyEvent.VK_RIGHT: {
                moveWatcherAroundScene(movingDelta);
                rotateViewX(angleDelta);
                break;
            }
        }
```

```java
        updateViewPosition();
    }

    public void moveWatcherAroundScene(double delta) {
        Vector3d n = new Vector3d();
        n.cross(xAxis, zAxis);
        n.normalize();
        n.scale(delta);
        watcher.add(n);
    }

    public void rotateViewX(double angle) {
        xAxis = VecmathHelper.rotateVector(xAxis, zAxis, angle);
    }

    public void rotateViewY(double angle) {
        Vector3d horizon = new Vector3d();
        horizon.cross(xAxis, zAxis);
        xAxis = VecmathHelper.rotateVector(xAxis, horizon, angle);
    }

    public void moveViewBySightDirection(double delta) {
        Vector3d deltaVector = new Vector3d(xAxis);
        deltaVector.scale(delta);
        watcher.add(deltaVector);
    }

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

    @Override
    public void keyTyped(KeyEvent e) { }

    @Override
    public void keyPressed(KeyEvent e) {
        updateEyePosition(e.getKeyCode());
    }

    @Override
    public void keyReleased(KeyEvent e) { }
}
```

Robot.java

```java
public class Robot extends TransformGroup {
    public Robot(double size) {
        double delta = size / 10;
        TransformGroup group = new TransformGroup();

        Head head = new Head(delta * 3);
        Transform3D headTransform = new Transform3D();
        headTransform.setTranslation(new Vector3d(0, 0, 11 * delta / 4));
        head.setTransform(headTransform);
        group.addChild(head);

        Arm rightArm = new Arm(4 * delta);
        Transform3D rightArmTransform = new Transform3D();
        rightArmTransform.rotZ(-Math.PI / 2);
        rightArmTransform.setTranslation(new Vector3d(0, -2.5 * delta, delta));
        rightArm.setTransform(rightArmTransform);
        group.addChild(rightArm);
        Arm leftArm = new Arm(4 * delta);
        Transform3D leftArmTransform = new Transform3D();
        leftArmTransform.rotZ(Math.PI / 2);
        leftArmTransform.setTranslation(new Vector3d(0, 2.5 * delta, delta));
        leftArm.setTransform(leftArmTransform);
        group.addChild(leftArm);

        Leg rightLeg = new Leg(4 * delta);
        Transform3D rightLegTransform = new Transform3D();
        rightLegTransform.setTranslation(new Vector3d(0, -3 * delta / 4, -3 *
delta));
        rightLeg.setTransform(rightLegTransform);
        group.addChild(rightLeg);


        Leg leftLeg = new Leg(4 * delta);
        Transform3D leftLegTransform = new Transform3D();
        leftLegTransform.setTranslation(new Vector3d(0, 3 * delta / 4, -3 * delta));
        leftLeg.setTransform(leftLegTransform);
        group.addChild(leftLeg);

        Body body = new Body(3 * delta);
        Transform3D bodyTransform = new Transform3D();
        bodyTransform.setTranslation(new Vector3d(0, 0, delta / 2));
        body.setTransform(bodyTransform);
        group.addChild(body);

        addChild(group);
    }
}
```

Arm.java

```
public class Arm extends TransformGroup {
    public Arm(double size) {
        float delta = (float)(size / 8);

        TransformGroup group = new TransformGroup();

        TransformGroup armGroup = new TransformGroup();
        Cylinder arm = new Cylinder((float)(delta * 0.4), delta * 5);
        arm.setAppearance(getAppearance());
        armGroup.addChild(arm);
        Transform3D armTransform  = new Transform3D();
        armTransform.rotZ(Math.PI / 2);
        armGroup.setTransform(armTransform);
        group.addChild(armGroup);



        TransformGroup handGroup = new TransformGroup();
        Shape3D hand = new Frustum()
                .setHeight(3 * delta / 4)
                .setInnerRadius(3 * delta / 8)
                .setOuterRadius(9 * delta / 16)
                .compile(getAppearance());
        handGroup.addChild(hand);
        Transform3D handTransform = new Transform3D();
        handTransform.rotY(Math.PI / 2);
        handTransform.setTranslation(new Vector3f(23 * (float) delta / 8, 0, 0));
        handGroup.setTransform(handTransform);
        group.addChild(handGroup);

        addChild(group);
    }
    public Appearance getAppearance() {
        Appearance appearance = new Appearance();
        appearance.setMaterial(
                new Material(
                        new Color3f(0.4453f, 0.4453f, 0.4453f),
                        new Color3f(0f, 0f, 0f),
                        new Color3f(0.4453f, 0.4453f, 0.4453f),
                        new Color3f(1f, 1f, 1f),
                        70f
                )
        );
        return appearance;
    }
}
```

Body.java

```
public class Body extends TransformGroup {
    public Body(double size) {
        TransformGroup group = new TransformGroup();

        TransformGroup upGroup = new TransformGroup();
        Shape3D up = new Frustum()
                .setHeight(size / 6)
                .setInnerRadius(size / 4)
                .setOuterRadius(size / 2)
                .compile(getAppearance());
        upGroup.addChild(up);
        Transform3D upTransform = new Transform3D();
        upTransform.rotX(Math.PI);
        upTransform.setTranslation(new Vector3f(0, 0, 5 * (float) size / 12));
        upGroup.setTransform(upTransform);
        group.addChild(upGroup);

        TransformGroup downGroup = new TransformGroup();
        Shape3D down = new Frustum()
                .setHeight(5 * size / 6)
                .setInnerRadius(size / 3)
                .setOuterRadius(size / 2)
                .compile(getAppearance());
        downGroup.addChild(down);
        Transform3D downTransform = new Transform3D();
        downTransform.setTranslation(new Vector3f(0, 0, (float) -size / 12));
        downGroup.setTransform(downTransform);
        group.addChild(downGroup);

        addChild(group);
    }


    public Appearance getAppearance() {
        Appearance appearance = new Appearance();
        appearance.setMaterial(
                new Material(
                        new Color3f(0.4453f, 0.4453f, 0.4453f),
                        new Color3f(0f, 0f, 0f),
                        new Color3f(0.4453f, 0.4453f, 0.4453f),
                        new Color3f(1f, 1f, 1f),
                        70f
                )
        );
        return appearance;
    }
}
```

Eye.java

```
public class Eye extends TransformGroup {
    public Eye(double size) {
        TransformGroup group = new TransformGroup();

        Sphere eye = new Sphere((float) size / 2);
        eye.setAppearance(getEyeAppearance());
        group.addChild(eye);

        TransformGroup pupilGroup = new TransformGroup();
        Transform3D pupilTransform = new Transform3D();
        pupilTransform.setTranslation(new Vector3f((float) size / 3, 0f, 0f));
        pupilGroup.setTransform(pupilTransform);
        pupilGroup.addChild(new Cube().setSize((float) size /
6).compile(getPupilAppearance()));
        group.addChild(pupilGroup);

        addChild(group);
    }

    public Appearance getEyeAppearance() {
        Appearance appearance = new Appearance();
        appearance.setMaterial(
                new Material(
                        new Color3f(1f, 1f, 0.7969f),
                        new Color3f(0f, 0f, 0f),
                        new Color3f(1f, 1f, 0.7969f),
                        new Color3f(1f, 1f, 1f),
                        70f
                )
        );
        return appearance;
    }

    public Appearance getPupilAppearance() {
        Appearance appearance = new Appearance();
        appearance.setMaterial(
                new Material(
                        new Color3f(0f, 0f, 0f),
                        new Color3f(0f, 0f, 0f),
                        new Color3f(0f, 0f, 0f),
                        new Color3f(0f, 0f, 0f),
                        70f
                )
        );
        return appearance;
    }
}
```

Head.java

```
public class Head extends TransformGroup {

    public Head(double size) {
        TransformGroup group = new TransformGroup();

        Shape3D cylinder = new Frustum()
                .setHeight(size / 2)
                .setInnerRadius(size / 4)
                .setOuterRadius(size / 4)
                .compile(getAppearance());
        group.addChild(cylinder);

        TransformGroup halfSphere1Group = new TransformGroup();
        Shape3D halfSphere1 = new HalfSphere()
                .setRadius(size / 4)
                .compile(getAppearance());
        halfSphere1Group.addChild(halfSphere1);
        Transform3D halfSphere1Transform = new Transform3D();
        halfSphere1Transform.setTranslation(new Vector3d(0, 0, size / 4));
        halfSphere1Group.setTransform(halfSphere1Transform);
        group.addChild(halfSphere1Group);

        Eye leftEye = new Eye(3 * (float) size / 16);
        Transform3D leftEyeTransform = new Transform3D();
        leftEyeTransform.setTranslation(new Vector3f(
                9 * (float) size / 50,
                (float) size / 8,
                5 * (float) size / 32
        ));
        leftEye.setTransform(leftEyeTransform);
        group.addChild(leftEye);

        Eye rightEye = new Eye(3 * (float) size / 16);
        Transform3D rightEyeTransform = new Transform3D();
        rightEyeTransform.setTranslation(new Vector3f(
                9 * (float) size / 50,
                (float) -size / 8,
                5 * (float) size / 32
        ));
        rightEye.setTransform(rightEyeTransform);
        group.addChild(rightEye);

        // -------------------MOUTH-------------------------
        float offset = (float)(size / 4);

        TransformGroup topMouthGroup = new TransformGroup();
        Shape3D topMouth = new Cube()
                .setSize(size / 4)
                .compile(getAppearance());
        topMouthGroup.addChild(topMouth);
        Transform3D topMouthTransform = new Transform3D();
        topMouthTransform.setTranslation(new Vector3f( 3 * (float) size / 16, 0,
(float) size / 4 - offset));
        topMouthTransform.setScale(new Vector3d(0.75, 1, 0.02));
        topMouthGroup.setTransform(topMouthTransform);
        group.addChild(topMouthGroup);

        TransformGroup bottomMouthGroup = new TransformGroup();
        Shape3D bottomMouth = new Cube()
                .setSize(size / 4)
                .compile(getAppearance());
        bottomMouthGroup.addChild(bottomMouth);
        Transform3D bottomMouthTransform = new Transform3D();
        bottomMouthTransform.setTranslation(new Vector3f( 3 * (float) size / 16, 0,
(float) size / 16 - offset));
        bottomMouthTransform.setScale(new Vector3d(0.75, 1, 0.02));
        bottomMouthGroup.setTransform(bottomMouthTransform);
        group.addChild(bottomMouthGroup);

        TransformGroup leftMouthGroup = new TransformGroup();
        Shape3D leftMouth = new Cube()
                .setSize(size / 4)
                .compile(getAppearance());
        leftMouthGroup.addChild(leftMouth);
        Transform3D leftMouthTransform = new Transform3D();
        leftMouthTransform.setTranslation(new Vector3f(
                3 * (float) size / 16,
                (float) size / 4 - 0.02f * (float) size / 4,
                5 * (float) size / 32 - offset));
        leftMouthTransform.setScale(new Vector3d(0.75, 0.02, 0.39));
        leftMouthGroup.setTransform(leftMouthTransform);
        group.addChild(leftMouthGroup);

        TransformGroup rightMouthGroup = new TransformGroup();
        Shape3D rightMouth = new Cube()
                .setSize(size / 4)
                .compile(getAppearance());
        rightMouthGroup.addChild(rightMouth);
        Transform3D rightMouthTransform = new Transform3D();
        rightMouthTransform.setTranslation(new Vector3f(
                3 * (float) size / 16,
                (float) -size / 4 + 0.02f * (float) size / 4,
                5 * (float) size / 32 - offset));
        rightMouthTransform.setScale(new Vector3d(0.75, 0.02, 0.39));
        rightMouthGroup.setTransform(rightMouthTransform);
        group.addChild(rightMouthGroup);

        TransformGroup backMouthGroup = new TransformGroup();
        Shape3D backMouth = new Cube()
                .setSize(size / 4)
                .compile(leftEye.getPupilAppearance());
        backMouthGroup.addChild(backMouth);
        Transform3D backMouthTransform = new Transform3D();
        backMouthTransform.setTranslation(new Vector3f(
                (float) size / 4,
                0,
                5 * (float) size / 32 - offset));
        backMouthTransform.setScale(new Vector3d(0.02, 0.98, 0.39));
        backMouthGroup.setTransform(backMouthTransform);
        group.addChild(backMouthGroup);
        // -------------------MOUTH-------------------------

        addChild(group);
    }

    public Appearance getAppearance() {
        Appearance appearance = new Appearance();
        appearance.setMaterial(
                new Material(
                        new Color3f(0.4453f, 0.4453f, 0.4453f),
                        new Color3f(0f, 0f, 0f),
                        new Color3f(0.4453f, 0.4453f, 0.4453f),
                        new Color3f(1f, 1f, 1f),
                        70f
                )
        );
        return appearance;
    }
}
```

Leg.java

```
public class Leg extends TransformGroup {
    public Leg(double size) {
        float delta = (float)(size / 8);

        TransformGroup group = new TransformGroup();

        TransformGroup legGroup = new TransformGroup();
        Cylinder leg = new Cylinder((float)(delta * 0.4), (float)(delta * 8));
        leg.setAppearance(getAppearance());
        legGroup.addChild(leg);
        Transform3D legTransform  = new Transform3D();
        legTransform.rotX(Math.PI / 2);
        legGroup.setTransform(legTransform);
        group.addChild(legGroup);
        TransformGroup footGroup = new TransformGroup();
        Shape3D foot = new HalfSphere()
                .setRadius(delta)
                .compile(getAppearance());
        footGroup.addChild(foot);
        Transform3D footTransform = new Transform3D();
        footTransform.setTranslation(new Vector3d(0, 0, -delta * 4));
        footGroup.setTransform(footTransform);
        group.addChild(footGroup);

        addChild(group);
    }

    public Appearance getAppearance() {
        Appearance appearance = new Appearance();
        appearance.setMaterial(
                new Material(
                        new Color3f(0.4453f, 0.4453f, 0.4453f),
                        new Color3f(0f, 0f, 0f),
                        new Color3f(0.4453f, 0.4453f, 0.4453f),
                        new Color3f(1f, 1f, 1f),
                        70f
                )
        );
        return appearance;
    }
}
```

Результат

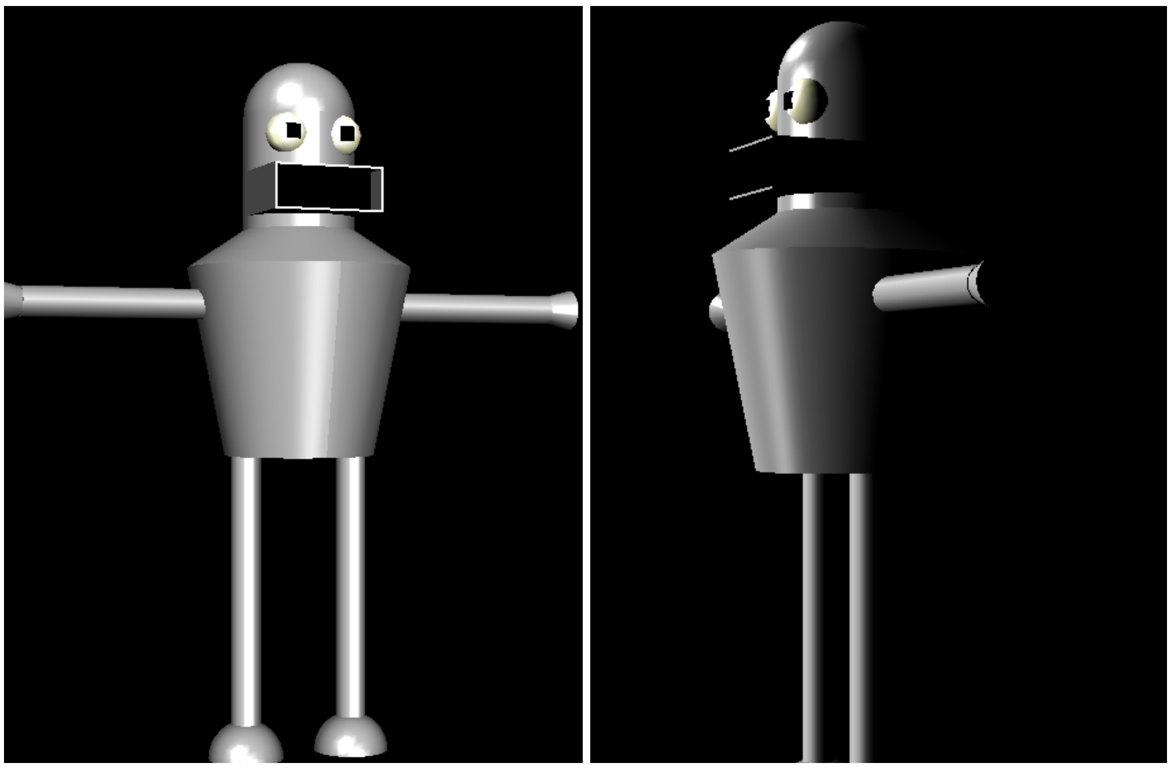

Рис. 1-2. Результати роботи програми