



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
Матіюк Дарина Андріївна

Зарахована
“ ____ ” “ _____ ” 20__ р.
Шкурат Оксаною Сергіївною

варіант №12

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: мухомор.

Лістинг коду програми

Main.java

```

public class Main extends Applet implements ActionListener {
    private final TransformGroup tg = new TransformGroup();
    private final Transform3D t3D = new Transform3D();
    private final Timer timer = new Timer(50, this);
    private float angle = 0;
    private double scale = 0;
    private boolean rotateY = true;
    private boolean isDecreasing = false;
    public static void main(String[] args) {
        var obj = new Main();
        MainFrame mf = new MainFrame(obj, 600, 600);
        mf.run();
    }
    private Main() {
        setLayout(new BorderLayout());
        GraphicsConfiguration config =
SimpleUniverse.getPreferredConfiguration();
        Canvas3D c = new Canvas3D(config);
        add("Center", c);
        SimpleUniverse universe = new SimpleUniverse(c);
        timer.start();
        universe.getViewingPlatform().setNominalViewingTransform();
        universe.addBranchGraph(createSceneGraph());
    }
    private BranchGroup createSceneGraph() {
        BranchGroup root = new BranchGroup();
        tg.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        root.addChild(tg);
        buildFungus();
        //light
        BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0,
0.0),100);
        Color sunLightColor = new Color(242, 255, 0);
        DirectionalLight lightDirect = new DirectionalLight(new
Color3f(sunLightColor), new Vector3f(0, 0, 0));
        lightDirect.setInfluencingBounds(bounds);
        root.addChild(lightDirect);
        AmbientLight ambientLightNode = new AmbientLight(new Color3f(new
Color(100, 255, 255)));
        ambientLightNode.setInfluencingBounds(bounds);
        root.addChild(ambientLightNode);
        //
        return root;
    }
    private void buildFungus() {
        //leg
        TransformGroup fungusLegGroup = new TransformGroup();
        Transform3D transformLeg = new Transform3D();
        Cylinder fungusLeg = Fungus.getLeg(0.1f, 0.4f);
        Vector3f vectorLeg = new Vector3f(.0f, 0.1f, .0f);
        transformLeg.setTranslation(vectorLeg);
        fungusLegGroup.setTransform(transformLeg);
        fungusLegGroup.addChild(fungusLeg);
        tg.addChild(fungusLegGroup);
        //middle leg
    }
}

```

```

TransformGroup fungusMiddleLegGroup = new TransformGroup();
Transform3D transformMiddleLeg = new Transform3D();
Cylinder fungusMiddleLeg = Fungus.getMiddleLeg(0.2f, 0.1f);
Vector3f vectorMiddleLeg = new Vector3f(.0f, 0.3f, .0f);
transformMiddleLeg.setTranslation(vectorMiddleLeg);
fungusMiddleLegGroup.setTransform(transformMiddleLeg);
fungusMiddleLegGroup.addChild(fungusMiddleLeg);
tg.addChild(fungusMiddleLegGroup);
//head
TransformGroup fungusHeadGroup = new TransformGroup();
Transform3D transformHead = new Transform3D();
Cone fungusHead = Fungus.getHead(0.4f, 0.2f);
Vector3f vectorHead = new Vector3f(.0f, 0.4f, .0f);
transformHead.setTranslation(vectorHead);
fungusHeadGroup.setTransform(transformHead);
fungusHeadGroup.addChild(fungusHead);
tg.addChild(fungusHeadGroup);
//circle1
TransformGroup fungusCircleGroup1 = new TransformGroup();
Transform3D transformCircle1 = new Transform3D();
Sphere fungusCircle1 = Fungus.getCircle(0.05f);
Vector3f vectorCircle1 = new Vector3f(0.1f, 0.45f, .0f);
transformCircle1.setTranslation(vectorCircle1);
fungusCircleGroup1.setTransform(transformCircle1);
fungusCircleGroup1.addChild(fungusCircle1);
tg.addChild(fungusCircleGroup1);
//circle2
TransformGroup fungusCircleGroup2 = new TransformGroup();
Transform3D transformCircle2 = new Transform3D();
Sphere fungusCircle2 = Fungus.getCircle(0.05f);
Vector3f vectorCircle2 = new Vector3f(-0.1f, 0.45f, .0f);
transformCircle2.setTranslation(vectorCircle2);
fungusCircleGroup2.setTransform(transformCircle2);
fungusCircleGroup2.addChild(fungusCircle2);
tg.addChild(fungusCircleGroup2);
//circle3
TransformGroup fungusCircleGroup3 = new TransformGroup();
Transform3D transformCircle3 = new Transform3D();
Sphere fungusCircle3 = Fungus.getCircle(0.05f);
Vector3f vectorCircle3 = new Vector3f(0.3f, 0.35f, .0f);
transformCircle3.setTranslation(vectorCircle3);
fungusCircleGroup3.setTransform(transformCircle3);
fungusCircleGroup3.addChild(fungusCircle3);
tg.addChild(fungusCircleGroup3);
//circle4
TransformGroup fungusCircleGroup4 = new TransformGroup();
Transform3D transformCircle4 = new Transform3D();
Sphere fungusCircle4 = Fungus.getCircle(0.05f);
Vector3f vectorCircle4 = new Vector3f(-0.3f, 0.35f, .0f);
transformCircle4.setTranslation(vectorCircle4);
fungusCircleGroup4.setTransform(transformCircle4);
fungusCircleGroup4.addChild(fungusCircle4);
tg.addChild(fungusCircleGroup4);
}
@Override
public void actionPerformed(ActionEvent e) {
    t3D.rotY(angle);
    angle += 0.05;
    if (angle >= 25) {
        rotateY = !rotateY;
    }
}

```

```
        angle = 0;  
    }  
    tg.setTransform(t3D);  
}  
}
```

Результат

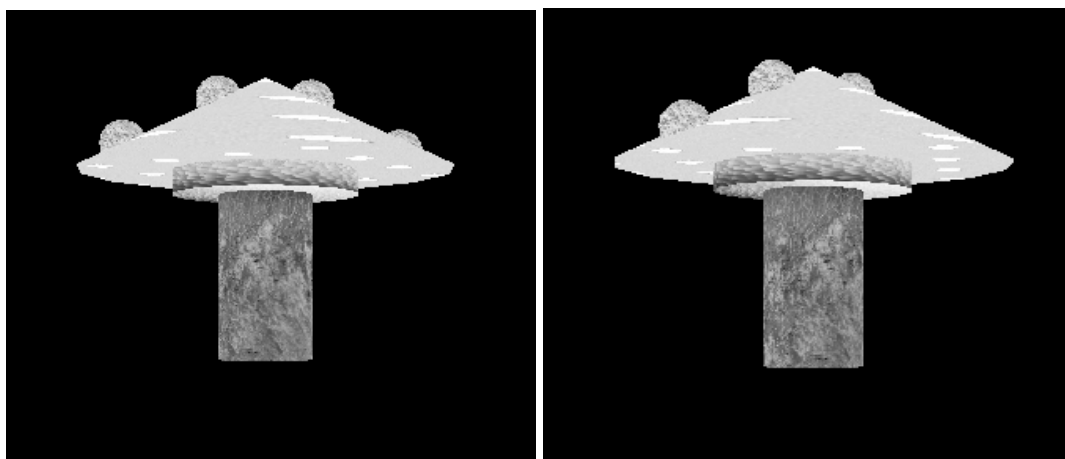


Рис. 1-2. Результати роботи програми.