

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ КИЇВСЬКИЙ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ  
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики  
Інженерія програмного забезпечення

**ЛАБОРАТОРНА РОБОТА №4**  
з дисципліни *“Математичні та алгоритмічні основи  
комп’ютерної графіки”*

**Виконав:**

студент 3-го курсу, групи КП-83,  
спеціальності 121 – Інженерія  
програмного забезпечення *Медведєв  
Ілля Анатолійович*

**Зарахована:**

*Шкурат О. С.*

Київ – 2021

**Мета:**

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

**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 взаємодії з об'єктом за допомогою миші та клавіатури.*

**Варіант 13: Стілець з кубом**

## Текст коду програм

### Main.java

```
import com.sun.j3d.utils.universe.SimpleUniverse;

import javax.media.j3d.*;
import javax.swing.*;
import javax.vecmath.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class App implements ActionListener {
    private final float higherEyeLimit = 4.0f;
    private final float lowerEyeLimit = 5.0f;
    private final float furthestEyeLimit = 18.0f;
    private final float closestEyeLimit = 25.0f;

    private TransformGroup treeTransformGroup;
    private TransformGroup viewingTransformGroup;
    private Transform3D treeTransform3D = new Transform3D();
    private Transform3D viewingTransform = new Transform3D();
    private float angle = 0;
    private float eyeHeight;
    private float eyeDistance;
    private boolean descending = true;
    private boolean approaching = true;

    public static void main(String[] args) {
        App book = new App();
    }

    private App() {
        Timer timer = new Timer(50, this);

        SimpleUniverse universe = new SimpleUniverse();

        viewingTransformGroup = universe.getViewingPlatform().getViewPlatformTransform();
        universe.addBranchGraph(createSceneGraph());

        eyeHeight = higherEyeLimit;
        eyeDistance = furthestEyeLimit;
        timer.start();
    }

    private Appearance getPagesMaterials() {
        Appearance ap = new Appearance();

        Color3f emissive = new Color3f(new Color(255, 250, 200));
        Color3f ambient = new Color3f(new Color(255, 250, 200));
        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 Appearance getCoverMaterials() {
        Appearance ap = new Appearance();
```

```

        Color3f emissive = new Color3f(new Color(70, 0, 0));
        Color3f ambient = new Color3f(new Color(50, 0, 0));
        Color3f diffuse = new Color3f(new Color(50, 0, 0));
        Color3f specular = new Color3f(new Color(50, 0, 0));

        ap.setMaterial(new Material(ambient, emissive, diffuse, specular,
        10.0f)); return ap;
    }

    private Appearance getStripeMaterials() {
        Appearance ap = new Appearance();

        Color3f emissive = new Color3f(new Color(0, 30, 100));
        Color3f ambient = new Color3f(new Color(0, 30, 100));
        Color3f diffuse = new Color3f(new Color(0, 30, 100));
        Color3f specular = new Color3f(new Color(0, 30, 100));
        ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));

        return ap;
    }

    private Appearance getWritingMaterials() {
        Appearance ap = new Appearance();

        Color3f emissive = new Color3f(Color.black);
        Color3f ambient = new Color3f(Color.black);
        Color3f diffuse = new Color3f(new Color(0, 60,
        0)); Color3f specular = new Color3f(Color.black);
        ap.setMaterial(new Material(ambient, emissive, diffuse, specular,
        5.0f)); return ap;
    }

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

        treeTransformGroup = new TransformGroup();
        treeTransformGroup.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        buildBook();

        objRoot.addChild(treeTransformGroup);
        createBackground(objRoot);
        BoundingSphere bounds = createBoundingSphere(objRoot);
        createAmbientLight(objRoot, bounds);

        return objRoot;
    }

    private void createAmbientLight(BranchGroup objRoot, BoundingSphere bounds)
    { Color3f ambientColor = new Color3f(1.0f, 1.0f, 1.0f);
      AmbientLight ambientLightNode = new AmbientLight(ambientColor);
      ambientLightNode.setInfluencingBounds(bounds);
      objRoot.addChild(ambientLightNode);
    }

    private BoundingSphere createBoundingSphere(BranchGroup objRoot) {
        BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0),
        100000.0); Color3f light1Color = new Color3f(1.0f, 1, 1);
        Vector3f light1Direction = new Vector3f(4.0f, -7.0f, -12.0f);
    }

```

```

        DirectionalLight light1 = new DirectionalLight(light1Color, light1Direction);
        light1.setInfluencingBounds(bounds);
        objRoot.addChild(light1);
        return bounds;
    }

    private void createBackground(BranchGroup objRoot) {
        Background background = new Background(new Color3f(0.9f, 0.9f, 0.9f)); // white
color
        BoundingSphere sphere = new BoundingSphere(new Point3d(0, 0, 0), 1000000000);
        background.setApplicationBounds(sphere); objRoot.addChild(background);
    }

    private TransformGroup buildTG() {
        return buildTG(new Vector3f(), new Transform3D());
    }

    private TransformGroup buildTG(Vector3f translation)
    { return buildTG(translation, new Transform3D());
    }

    private TransformGroup buildTG(Vector3f translation, Transform3D rotation) {
        Transform3D transform = new Transform3D();
        TransformGroup transformG = new TransformGroup();
        transform.mul(rotation, transform);
        transform.setTranslation(translation);
        transformG.setTransform(transform); return
        transformG;
    }

    private void buildBook() {
        TransformGroup pagesTG = buildPages();
        buildCover(pagesTG);
        buildCover2(pagesTG);
        buildCover3(pagesTG);
        buildWritings(pagesTG);
        buildStripe(pagesTG);
        treeTransformGroup.addChild(pagesTG);
    }

    private void buildStripe(TransformGroup pagesTG) {
        com.sun.j3d.utils.geometry.Box stripe = new com.sun.j3d.utils.geometry.Box(1f,
0.2f, 8.5f, getStripeMaterials());

        TransformGroup stripeTG = buildTG(new Vector3f(3, 0, -3));
        stripeTG.addChild(stripe);
        pagesTG.addChild(stripeTG);
    }

    private void buildWritings(TransformGroup pagesTG) {
        com.sun.j3d.utils.geometry.Box writing1 = new com.sun.j3d.utils.geometry.Box(4f,
2f, 1f, getWritingMaterials());
        TransformGroup writing1TG = buildTG(new Vector3f(0, 0.4f, 3));
        writing1TG.addChild(writing1);
        pagesTG.addChild(writing1TG);

        com.sun.j3d.utils.geometry.Box writing2 = new
com.sun.j3d.utils.geometry.Box(4f, 1f, 4f, getWritingMaterials());
        TransformGroup writing2TG = buildTG(new Vector3f(2.55f, 0,
0)); writing2TG.addChild(writing2);
    }

```

```

        pagesTG.addChild(writing2TG);
    }

    private void buildCover3(TransformGroup pagesTG) {
        com.sun.j3d.utils.geometry.Box cover3 = new com.sun.j3d.utils.geometry.Box(0.5f,
2.4f, 8.5f, getCoverMaterials());
        TransformGroup cover3TG = buildTG(new Vector3f(6, 0, 0));
        cover3TG.addChild(cover3);
        pagesTG.addChild(cover3TG);
    }

    private void buildCover2(TransformGroup pagesTG) {
        com.sun.j3d.utils.geometry.Box cover2 = new com.sun.j3d.utils.geometry.Box(6.5f,
0.2f, 8.5f, getCoverMaterials());
        TransformGroup cover2TG = buildTG(new Vector3f(0, -5.5f, 0), new
Transform3D(new Quat4d(0, 0, 1, -0.3), new Vector3d(1, 1, 1 ), 1));
        cover2TG.addChild(cover2);
        pagesTG.addChild(cover2TG);
    }

    private void buildCover(TransformGroup pagesTG) {
        com.sun.j3d.utils.geometry.Box cover = new com.sun.j3d.utils.geometry.Box(6.5f,
0.2f, 8.5f, getCoverMaterials());
        TransformGroup coverTG = buildTG(new Vector3f(0, 2, 0));

        coverTG.addChild(cover);
        pagesTG.addChild(coverTG);
    }

    private TransformGroup buildPages() {
        com.sun.j3d.utils.geometry.Box pages = new com.sun.j3d.utils.geometry.Box(6,
2, 8, getPagesMaterials());
        TransformGroup pagesTG = buildTG();
        pagesTG.addChild(pages);
        return pagesTG;
    }

    // ActionListener interface
    @Override
    public void actionPerformed(ActionEvent e)
    {
        float delta = 0.02f;

        // rotation of the castle
        treeTransform3D.rotZ(angle);
        treeTransformGroup.setTransform(treeTransform3D);

        angle += delta;

        // movement of the camera up and down
        if (eyeHeight > higherEyeLimit)
            descending = true;
        else if (eyeHeight < lowerEyeLimit)
            descending = false;

        if (descending)
            eyeHeight -= delta;
        else
            eyeHeight += delta;

        // change camera distance to the scene
        if (eyeDistance > furthestEyeLimit)
            approaching = true;
        else if (eyeDistance < closestEyeLimit)

```

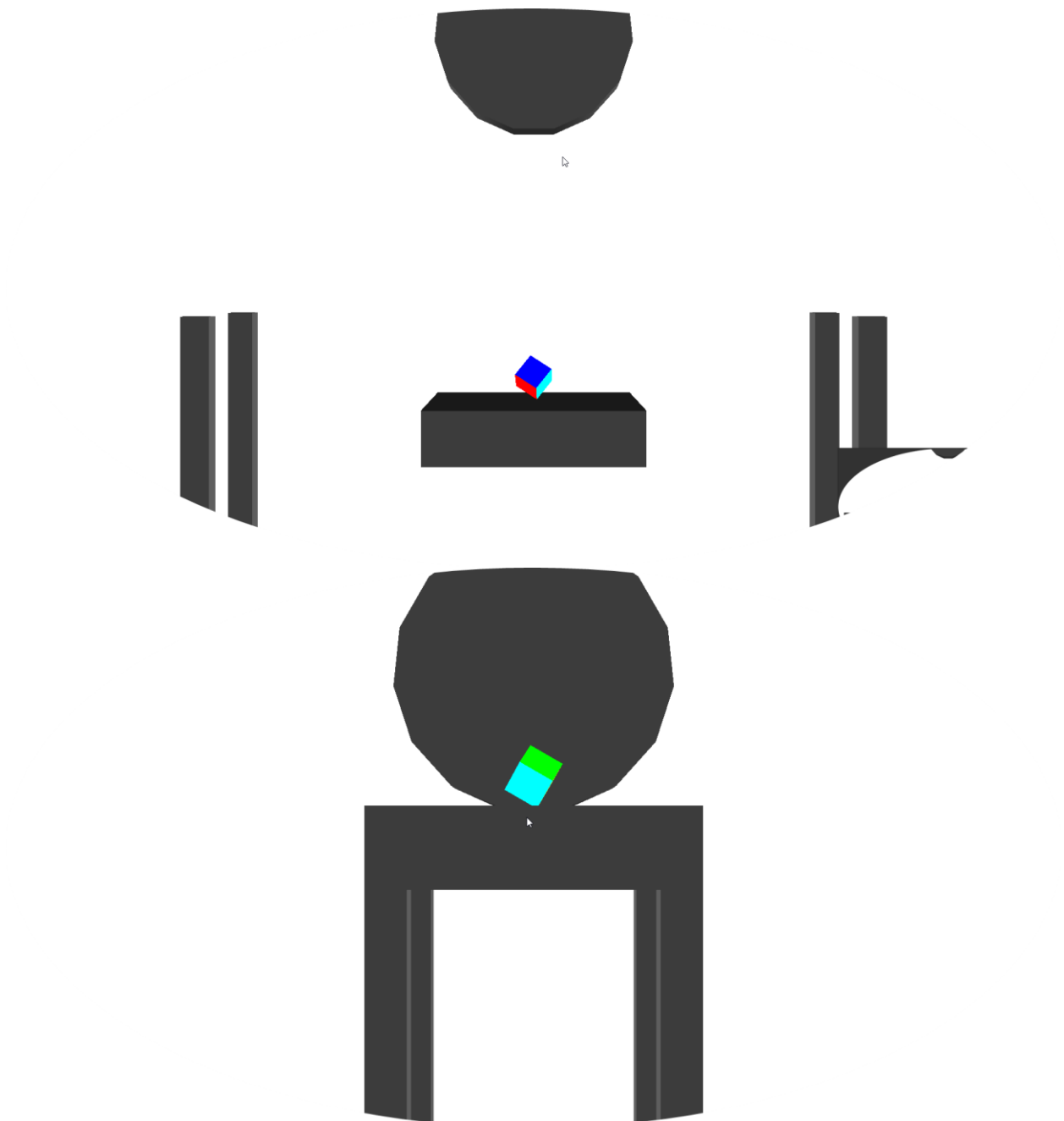
```
        approaching = false;

    if (approaching)
        eyeDistance -= delta;
    else
        eyeDistance += delta;

    Point3d eye = new Point3d(eyeDistance, eyeDistance, eyeHeight); // eye
    Point3d center = new Point3d(.0f, .0f, 0.1f); // target
    Vector3d up = new Vector3d(.0f, .0f, 1.0f);

    viewingTransform.lookAt(eye, center, up);
    viewingTransform.invert();
    viewingTransformGroup.setTransform(viewingTransform);
}
}
```

## Результати роботи програми





## **Висновки**

Виконавши дану лабораторну роботу, я вивчив стандарти засоби Java3D для візуалізації зображення, вивчив засоби анімації примітивів та складених об'єктів в Java3D.