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

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота № 4

з дисципліни “ МАОКГ”

Виконала
студентка III курсу
групи КП-83
Снітко Маріанна Дмитрівна
(*прізвище, ім’я, по батькові*)
Варіант № 19

Зарахована
“____” “_____” 2021 р.
викладачем
Шкурат О. С.
(*прізвище, ім’я, по батькові*)

Київ 2021

Варіант завдання №19

Персональний комп'ютер

Лістинг коду програми

PC.java

```
package application;

import com.sun.j3d.utils.geometry.*;
import com.sun.j3d.utils.geometry.Box;
import com.sun.j3d.utils.universe.SimpleUniverse;
import javax.media.j3d.*;
import javax.swing.*;
import javax.vecmath.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class PC 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 PC();

    }

    private PC() {
        timer = new Timer(50, this);
        timer.start();
        BranchGroup scene = createSceneGraph();
        SimpleUniverse u = new SimpleUniverse();
        u.getViewingPlatform().setNominalViewingTransform();
        u.addBranchGraph(scene);

    }

    private BranchGroup createSceneGraph() {
        BranchGroup objRoot = new BranchGroup();
        carTransformGroup = new TransformGroup();
        carTransformGroup.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        MakePC();

        objRoot.addChild(carTransformGroup);

        Background background = new Background(new Color3f(1.0f, 1.0f, 1.0f));
        BoundingSphere sphere = new BoundingSphere(new Point3d(0,0,0), 100000);
        background.setApplicationBounds(sphere);
        objRoot.addChild(background);

        BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0,
0.0),100.0);
        Color3f light1Color = new Color3f(1.0f, 0.5f, 0.4f);
        Vector3f light1Direction = new Vector3f(.8f, .8f, .0f);
        DirectionalLight light1 = new DirectionalLight(light1Color,
light1Direction);
        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 MakePC() {
        Box leg1 = PCBuilder.getPart(.02f,.3f,.3f);
        Transform3D body1T = new Transform3D();
        body1T.setTranslation(new Vector3f(.4f, .0f, .0f));
        TransformGroup body1TG = new TransformGroup();
        body1TG.setTransform(body1T);
        body1TG.addChild(leg1);
        carTransformGroup.addChild(body1TG);


        Box leg2 = PCBuilder.getPart(.02f,.3f,.3f);
        Transform3D body2T = new Transform3D();
        body2T.setTranslation(new Vector3f(-.4f, .0f, .0f));
        TransformGroup body2TG = new TransformGroup();
        body2TG.setTransform(body2T);
        body2TG.addChild(leg2);
        carTransformGroup.addChild(body2TG);


        Box table = PCBuilder.getPart(.6f,.01f,.34f);
        Transform3D body3T = new Transform3D();
        body3T.setTranslation(new Vector3f(.0f, .3f, .0f));
        TransformGroup body3TG = new TransformGroup();
        body3TG.setTransform(body3T);
        body3TG.addChild(table);
        carTransformGroup.addChild(body3TG);


        Box PC_Box = PCBuilder.getPCBox(.08f,.2f,.28f);
        Transform3D body4T = new Transform3D();
        body4T.setTranslation(new Vector3f(.2f, -.1f, .025f));
        TransformGroup body4TG = new TransformGroup();
        body4TG.setTransform(body4T);
        body4TG.addChild(PC_Box);
        carTransformGroup.addChild(body4TG);

        Box PC_Box_back = PCBuilder.getPCBack(.078f,.168f,.005f);
        Transform3D body5T = new Transform3D();
        body5T.setTranslation(new Vector3f(.2f, -.07f, -.26f));
        TransformGroup body5TG = new TransformGroup();
        body5TG.setTransform(body5T);
        body5TG.addChild(PC_Box_back);
        carTransformGroup.addChild(body5TG);

        Cylinder PC_Monitor_bottom = PCBuilder.getMonitor(.1f,.03f);
        Transform3D body6T = new Transform3D();
        body6T.setTranslation(new Vector3f(.0f, .32f, .0f));
        TransformGroup body6TG = new TransformGroup();
        body6TG.setTransform(body6T);
        body6TG.addChild(PC_Monitor_bottom);
        carTransformGroup.addChild(body6TG);

        Cylinder PC_Monitor_leg = PCBuilder.getMonitor(.013f,.08f);
        Transform3D body7T = new Transform3D();
        body7T.setTranslation(new Vector3f(.0f, .37f, .0f));
        TransformGroup body7TG = new TransformGroup();
        body7TG.setTransform(body7T);
        body7TG.addChild(PC_Monitor_leg);
```

```
carTransformGroup.addChild(body7TG);

Box PC_Monitor = PCBuilder.getMonitor(.3f,.18f,0.015f);
Transform3D body8T = new Transform3D();
body8T.setTranslation(new Vector3f(.0f, .59f, .0f));
TransformGroup body8TG = new TransformGroup();
body8TG.setTransform(body8T);
body8TG.addChild(PC_Monitor);
carTransformGroup.addChild(body8TG);

Box PC_Screen = PCBuilder.getScreen(.29f,.165f,0.001f);
Transform3D body9T = new Transform3D();
body9T.setTranslation(new Vector3f(.0f, .59f, .015f));
TransformGroup body9TG = new TransformGroup();
body9TG.setTransform(body9T);
body9TG.addChild(PC_Screen);
carTransformGroup.addChild(body9TG);

Cylinder PC_Push = PCBuilder.getPush(.05f,0.001f);
Transform3D body10T = new Transform3D();
body10T.rotX(Math.PI/2);
body10T.setTranslation(new Vector3f(.2f, -.19f, .305f));
TransformGroup body10TG = new TransformGroup();
body10TG.setTransform(body10T);
body10TG.addChild(PC_Push);
carTransformGroup.addChild(body10TG);

Box PC_CD1 = PCBuilder.getCD(.06f,.008f,0.001f);
Transform3D body11T = new Transform3D();
body11T.setTranslation(new Vector3f(.2f, -0.f, .305f));
TransformGroup body11TG = new TransformGroup();
body11TG.setTransform(body11T);
body11TG.addChild(PC_CD1);
carTransformGroup.addChild(body11TG);


Box PC_CD2 = PCBuilder.getCD(.06f,.008f,0.001f);
Transform3D body12T = new Transform3D();
body12T.setTranslation(new Vector3f(.2f, 0.02f, .305f));
TransformGroup body12TG = new TransformGroup();
body12TG.setTransform(body12T);
body12TG.addChild(PC_CD2);
carTransformGroup.addChild(body12TG);

Box PC_CD3 = PCBuilder.getCD(.06f,.008f,0.001f);
Transform3D body13T = new Transform3D();
body13T.setTranslation(new Vector3f(.2f, 0.04f, .305f));
TransformGroup body13TG = new TransformGroup();
body13TG.setTransform(body13T);
body13TG.addChild(PC_CD3);
carTransformGroup.addChild(body13TG);


Box PC_USB1 = PCBuilder.getButton(.013f,.005f,0.001f);
Transform3D body14T = new Transform3D();
body14T.setTranslation(new Vector3f(.22f, -.02f, .305f));
TransformGroup body14TG = new TransformGroup();
body14TG.setTransform(body14T);
body14TG.addChild(PC_USB1);
carTransformGroup.addChild(body14TG);

Box PC_USB2 = PCBuilder.getButton(.013f,.005f,0.001f);
Transform3D body15T = new Transform3D();
body15T.setTranslation(new Vector3f(.18f, -.02f, .305f));
TransformGroup body15TG = new TransformGroup();
body15TG.setTransform(body15T);
body15TG.addChild(PC_USB2);
```

```
        carTransformGroup.addChild(body15TG);

    }

    @Override
    public void actionPerformed(ActionEvent e) {
        carTransform3D.rotY(angle);
        carTransform3D.setScale(0.6);
        carTransformGroup.setTransform(carTransform3D);
        angle += 0.03;
    }
}
```

PCBuilder.java

```
package application;

import com.sun.j3d.utils.geometry.*;
import com.sun.j3d.utils.image.TextureLoader;

import javax.media.j3d.Appearance;
import javax.media.j3d.Material;
import javax.media.j3d.Texture;
import javax.media.j3d.TextureAttributes;
import javax.vecmath.Color3f;
import javax.vecmath.Color4f;
import java.awt.*;

public class PCBuilder {
        public static Box getPart(float x, float y, float z) {
            int primitive_flags = Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS;
            return new Box(x, y, z, primitive_flags, getTableAppearance());
        }
        public static Box getPCBox(float x, float y, float z) {
                int primitive_flags = Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS;
                return new Box(x, y, z, primitive_flags,
get_PC_box_Appearance());
        }
        public static Box getPCBack(float x, float y, float z) {
                int primitive_flags = Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS;
                return new Box(x, y, z, primitive_flags,
get_PC_box_back_Appearance());
        }

        public static Box getMonitor(float x, float y, float z) {
                int primitive_flags = Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS;
                return new Box(x, y, z, primitive_flags,
get_Monitor_Appearance());
        }

        public static Cylinder getMonitor(float radius, float height) {
                int primitive_flags = Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS;
                return new Cylinder(radius, height, primitive_flags,
get_Monitor_Appearance());
        }
```

```
        public static Cylinder getPush(float radius, float height) {
                int primitive_flags = Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS;
                return new Cylinder(radius, height, primitive_flags,
get_Push_Appearance());
        }
        public static Box getScreen(float x, float y, float z) {
                int primitive_flags = Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS;
                return new Box(x,y,z, primitive_flags, get_Screen_Appearance());
        }
        public static Box getButton(float x, float y, float z) {
                int primitive_flags = Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS;
                return new Box(x,y,z, primitive_flags, get_Button_Appearance());
        }

        public static Cylinder getButton(float radius, float height) {
                int primitive_flags = Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS;
                return new Cylinder(radius,height, primitive_flags,
get_Button_Appearance());
        }
        public static Box getCD(float x, float y, float z) {
                int primitive_flags = Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS;
                return new Box(x,y,z, primitive_flags, get_CD_Appearance());
        }

        private static Appearance getTableAppearance() {
                TextureLoader loader = new
TextureLoader("src\\textures\\wood.jpg", "LUMINANCE", new Container());
                Appearance ap = new Appearance();

                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);

                ap.setTexture(texture);
                ap.setTextureAttributes(texAttr);

                Color3f emissive = new Color3f(new Color(69, 29, 0));
                Color3f ambient = new Color3f(new Color(119, 41, 0));
                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 get_PC_box_Appearance() {
                TextureLoader loader = new
TextureLoader("src\\textures\\plastic_for_pc.jpg", "LUMINANCE", new Container());
                Appearance ap = new Appearance();

                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);

                ap.setTexture(texture);
```

```
                ap.setTextureAttributes(texAttr);

                Color3f emissive = new Color3f(new Color(99, 99, 101));
                Color3f ambient = new Color3f(new Color(0, 0, 0, 210));
                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 get_PC_box_back_Appearance() {
                TextureLoader loader = new
TextureLoader("src\\textures\\metal.jpg", "LUMINANCE", new Container());
                Appearance ap = new Appearance();

                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);

                ap.setTexture(texture);
                ap.setTextureAttributes(texAttr);

                Color3f emissive = new Color3f(new Color(71, 71, 73));
                Color3f ambient = new Color3f(new Color(179, 179, 179, 210));
                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 get_Monitor_Appearance() {
                TextureLoader loader = new
TextureLoader("src\\textures\\plastic.jpg", "LUMINANCE", new Container());
                Appearance ap = new Appearance();

                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);

                ap.setTexture(texture);
                ap.setTextureAttributes(texAttr);

                Color3f emissive = new Color3f(new Color(71, 71, 73));
                Color3f ambient = new Color3f(new Color(6, 6, 6, 210));
                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 get_Push_Appearance() {
                TextureLoader loader = new
TextureLoader("src\\textures\\push.jpg", "LUMINANCE", new Container());
                return getAppearance(loader);
        }

        private static Appearance getAppearance(TextureLoader loader) {
```

```
                Appearance ap = new Appearance();

                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);

                ap.setTexture(texture);
                ap.setTextureAttributes(texAttr);

                Color3f emissive = new Color3f(new Color(3, 3, 4));
                Color3f ambient = new Color3f(new Color(255, 0, 0, 255));
                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 get_CD_Appearance() {
                TextureLoader loader = new TextureLoader("src\\textures\\cd.jpg",
"LUMINANCE", new Container());
                return getAppearance(loader);
        }

        private static Appearance get_Button_Appearance() {
                TextureLoader loader = new
TextureLoader("src\\textures\\buttons.jpg", "LUMINANCE", new Container());
                Appearance ap = new Appearance();

                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);

                ap.setTexture(texture);
                ap.setTextureAttributes(texAttr);

                Color3f emissive = new Color3f(new Color(87, 87, 87));
                Color3f ambient = new Color3f(new Color(140, 0, 0, 246));
                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 get_Screen_Appearance() {
                TextureLoader loader = new
TextureLoader("src\\textures\\screen.png", "LUMINANCE", new Container());
                Appearance ap = new Appearance();

                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);

                ap.setTexture(texture);
                ap.setTextureAttributes(texAttr);
```

```
                Color3f emissive = new Color3f(new Color(0, 3, 119));
                Color3f ambient = new Color3f(new Color(0, 3, 50, 238));
                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;
        }
}
```

Результати роботи програми

Висновки

Виконавши дану лабораторну роботу, я вивчила стандартні засоби Java3D для візуалізації зображення. Крім цього, опанувала засоби анімації примітивів та складених об'єктів в Java3D.

Система була написана на мові програмування **Java.**