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”

Варіант 7

Виконав
студент 3 курсу
групи КП-81
Каснер Максим

Київ 2021

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 7 :

7. Морозиво

Код програми

Main.java

```
import com.sun.j3d.utils.geometry.*;
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 Main implements ActionListener {
    private final float upperEyeLimit = 6.0f;
    private final float lowerEyeLimit = 5.0f;
    private final float farthestEyeLimit = 6.0f;
    private final float nearestEyeLimit = 5.0f;

    private TransformGroup treeTransformGroup;
    private final TransformGroup viewingTransformGroup;
    private final Transform3D treeTransform3D = new Transform3D();
    private final Transform3D viewingTransform = new Transform3D();
    private float angle = 0;
    private float eyeHeight;
    private float eyeDistance;
    private boolean descend = true;
```

```
    private boolean approaching = true;

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

    private Main() {
        Timer timer = new Timer(50, this);
        SimpleUniverse universe = new SimpleUniverse();

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

        eyeHeight = upperEyeLimit;
        eyeDistance = farthestEyeLimit;
        timer.start();
    }

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

        // створюємо об'єкт, що будемо додавати до групи
        treeTransformGroup = new TransformGroup();
        treeTransformGroup.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        buildIceCream();
        objRoot.addChild(treeTransformGroup);

        Background background = new Background(new Color3f(0.9f, 0.9f, 0.9f)); // grey color
        BoundingSphere sphere = new BoundingSphere(new Point3d(0, 0, 0), 100000);
        background.setApplicationBounds(sphere);
        objRoot.addChild(background);

        // налаштовуємо освітлення
        BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0), 100.0);
        Color3f light1Color = new Color3f(1.0f, 1.0f, 1.f);
        Vector3f light1Direction = new Vector3f(4.0f, -7.0f, -12.0f);
        DirectionalLight light1 = new DirectionalLight(light1Color, light1Direction);
        light1.setInfluencingBounds(bounds);
        objRoot.addChild(light1);

        // встановлюємо навколишнє освітлення
        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 buildIceCream() {
    Cylinder body = new Cylinder(1, 3, Utils.getBodyAppearence());
    Transform3D bodyT = new Transform3D();
    bodyT.setTranslation(new Vector3f());
    bodyT.rotX(Math.PI / 2);
    TransformGroup bodyTG = new TransformGroup();
    bodyTG.setTransform(bodyT);
    bodyTG.addChild(body);

    Sphere ball = new Sphere(1.1f, Utils.getBallAppearence());
    Transform3D ballT = new Transform3D();
    ballT.setTranslation(new Vector3f(0, 2, 0));
    TransformGroup ballTG = new TransformGroup();
    ballTG.setTransform(ballT);
    ballTG.addChild(ball);
    bodyTG.addChild(ballTG);

    Sphere choco1 = new Sphere(0.2f, Utils.getChocoAppearence());
    Transform3D choco1T = new Transform3D();
    choco1T.setTranslation(new Vector3f(0, 1f, 0));
    TransformGroup choco1TG = new TransformGroup();
    choco1TG.setTransform(choco1T);
    choco1TG.addChild(choco1);


    Sphere choco2 = new Sphere(0.2f, Utils.getChocoAppearence());
    Transform3D choco2T = new Transform3D();
    choco2T.setTranslation(new Vector3f(0.7f, 0.8f, 0));
    TransformGroup choco2TG = new TransformGroup();
    choco2TG.setTransform(choco2T);
    choco2TG.addChild(choco2);

    Sphere choco3 = new Sphere(0.2f, Utils.getChocoAppearence());
    Transform3D choco3T = new Transform3D();
    choco3T.setTranslation(new Vector3f(0, 0.7f, -0.7f));
    TransformGroup choco3TG = new TransformGroup();
    choco3TG.setTransform(choco3T);
    choco3TG.addChild(choco3);

    Sphere choco4 = new Sphere(0.2f, Utils.getChocoAppearence());
    Transform3D choco4T = new Transform3D();
    choco4T.setTranslation(new Vector3f(-0.5f, 0.6f, -0.6f));
    TransformGroup choco4TG = new TransformGroup();
    choco4TG.setTransform(choco4T);
    choco4TG.addChild(choco4);
```

```
        Sphere choco5 = new Sphere(0.2f, Utils.getChocoAppearence());
        Transform3D choco5T = new Transform3D();
        choco5T.setTranslation(new Vector3f(-0.4f, 0.7f, 0.5f));
        TransformGroup choco5TG = new TransformGroup();
        choco5TG.setTransform(choco5T);
        choco5TG.addChild(choco5);

        ballTG.addChild(choco1TG);
        ballTG.addChild(choco2TG);
        ballTG.addChild(choco3TG);
        ballTG.addChild(choco4TG);
        ballTG.addChild(choco5TG);


        treeTransformGroup.addChild(bodyTG);

    }

    // ActionListener interface
    @Override
    public void actionPerformed(ActionEvent e) {
        float delta = 0.03f;

        // rotation
        treeTransform3D.rotZ(angle);
        treeTransformGroup.setTransform(treeTransform3D);
        angle += delta;

        // change of the camera position up and down within defined limits
        if (eyeHeight > upperEyeLimit) {
            descend = true;
        } else if (eyeHeight < lowerEyeLimit) {
            descend = false;
        }
        if (descend) {
            eyeHeight -= delta;
        } else {
            eyeHeight += delta;
        }

        // change camera distance to the scene
        if (eyeDistance > farthestEyeLimit) {
            approaching = true;
        } else if (eyeDistance < nearestEyeLimit) {
            approaching = false;
        }
        if (approaching) {
```

```
            eyeDistance -= delta;
        } else {
            eyeDistance += delta;
        }

        Point3d eye = new Point3d(eyeDistance, eyeDistance, eyeHeight); // spectator's eye
        Point3d center = new Point3d(.0f, .0f, .0f); // sight target
        Vector3d up = new Vector3d(.0f, .0f, 1.0f);
        viewingTransform.lookAt(eye, center, up);
        viewingTransform.invert();
        viewingTransformGroup.setTransform(viewingTransform);
    }
}
```

Utils.java

```
import javax.media.j3d.*; // for transform
import javax.vecmath.Color3f;
import java.awt.Color;

public class Utils {
    public static Appearance getBodyAppearence() {
        Appearance ap = new Appearance();

        Color3f emissive = new Color3f(Color.BLACK);
        Color3f ambient = new Color3f(Color.PINK);
        Color3f diffuse = new Color3f(Color.PINK);
        Color3f specular = new Color3f(Color.BLACK);
        // ambient, emissive, diffuse, specular, 1.0f
        ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));

        return ap;
    }

    public static Appearance getBallAppearence() {
        Appearance ap = new Appearance();

        Color3f emissive = new Color3f(Color.BLACK);
        Color3f ambient = new Color3f(Color.WHITE);
        Color3f diffuse = new Color3f(Color.WHITE);
        Color3f specular = new Color3f(Color.WHITE);
        // ambient, emissive, diffuse, specular, 1.0f
        ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));

        return ap;
```

```
    }

    public static Appearance getChocoAppearence() {
        Appearance ap = new Appearance();

        Color3f emissive = new Color3f(Color.BLACK);
        Color3f ambient = new Color3f(Color.BLACK);
        Color3f diffuse = new Color3f(Color.BLACK);
        Color3f specular = new Color3f(Color.BLACK);
        // ambient, emissive, diffuse, specular, 1.0f
        ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));

        return ap;
    }
}
```

Результати роботи програми :