

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 № 4

Виконав:

студент 3-го курсу, групи КП-83,
Дереворіз Назар

Перевірив:

викладач
Шкурат Оксана Сергіївна

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 взаємодії з об'єктом за допомогою миші та клавіатури.

4. Олівець

Код програми:

pencil.java

```
public static TransformGroup getPencil() {
    TransformGroup pencil = new TransformGroup();
    pencil.addChild(getPencilGraphite());
    pencil.addChild(getPencilTop());
    pencil.addChild(getPencilCenter());
    pencil.addChild(getInnerPencil());
    pencil.addChild(getInnerGraphitePencil());
    return pencil;
}

private static TransformGroup getPencilGraphite() {
    TransformGroup pencilGraphite = new TransformGroup();
    Transform3D transformPencilGraphite = new
    Transform3D();
    Cone conePencilGraphite = new
    Cone(0.09f, 0.22f, Primitive.GENERATE_NORMALS +
    Primitive.GENERATE_TEXTURE_COORDS,
    getPencilGraphiteAppearance());
    Vector3f vectorPencilGraphite = new
    Vector3f(.0f, .2f, .0f);

    transformPencilGraphite.setTranslation(vectorPencilGraphite);
    pencilGraphite.setTransform(transformPencilGraphite);
    pencilGraphite.addChild(conePencilGraphite);
}
```

```

        return pencilGraphite;
    }

    private static TransformGroup getPencilTop() {
        TransformGroup pencilTop = new TransformGroup();
        Transform3D transformPencilTop = new Transform3D();
        Cone conePencilTop = new
Cone(0.1f,0.2f,Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS, getPencilTopAppearance());
        Vector3f vectorPencilTop = new Vector3f(.0f,.2f,.0f);
        transformPencilTop.setTranslation(vectorPencilTop);
        pencilTop.setTransform(transformPencilTop);
        pencilTop.addChild(conePencilTop);
        return pencilTop;
    }

    private static TransformGroup getPencilCenter() {
        TransformGroup pencilCenter = new TransformGroup();
        Transform3D transformPencilCenter = new Transform3D();
        Cylinder pencilCenterCylinder = new
Cylinder(0.1f,0.5f,Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS,
getPencilSideAppearance());
        Vector3f vectorPencilCenter = new
Vector3f(.0f,-0.15f,.0f);

        transformPencilCenter.setTranslation(vectorPencilCenter);
        pencilCenter.setTransform(transformPencilCenter);
        pencilCenter.addChild(pencilCenterCylinder);
        return pencilCenter;
    }

    private static TransformGroup getInnerPencil() {
        TransformGroup innerPencil = new TransformGroup();
        Transform3D transformInnerPencil = new Transform3D();
        Cylinder innerPencilCylinder = new
Cylinder(0.098f,0.51f,Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS, getPencilTopAppearance());
        Vector3f vectorInnerPencil = new
Vector3f(.0f,-0.15f,.0f);

        transformInnerPencil.setTranslation(vectorInnerPencil);
        innerPencil.setTransform(transformInnerPencil);
        innerPencil.addChild(innerPencilCylinder);
        return innerPencil;
    }

    private static TransformGroup getInnerGraphitePencil() {
        TransformGroup innerGraphitePencil = new
TransformGroup();
    }

```

```

        Transform3D transformInnerGraphitePencil = new
Transform3D();
        Cylinder innerGraphitePencilCylinder = new
Cylinder(0.025f,0.52f,Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS,
getPencilGraphiteAppearance());
        Vector3f vectorInnerGraphitePencil = new
Vector3f(.0f,-0.15f,.0f);

transformInnerGraphitePencil.setTranslation(vectorInnerGraphi
tePencil);

innerGraphitePencil.setTransform(transformInnerGraphitePencil
);

innerGraphitePencil.addChild(innerGraphitePencilCylinder);
        return innerGraphitePencil;
    }

    private static Appearance getPencilSideAppearance() {
        Appearance ap = new Appearance();
        Color3f emissive = new Color3f(0.0f, 0.05f, 0.0f);
        Color3f ambient = new Color3f(0.2f, 0.5f, 0.15f);
        Color3f diffuse = new Color3f(0.2f, 0.15f, .15f);
        Color3f specular = new Color3f(0.0f, 0.8f, 0.0f);
        ap.setMaterial(new Material(ambient, emissive,
diffuse, specular, 1.0f));
        return ap;
    }

    private static Appearance getPencilTopAppearance() {
        Appearance ap = new Appearance();
        Color3f emissive = new Color3f(0.9f, 0.6f, 0.4f);
        Color3f ambient = new Color3f(1.0f, 0.8f, 0.6f);
        Color3f diffuse = new Color3f(0.8f, 0.8f, 0.6f);
        Color3f specular = new Color3f(1.0f, 0.0f, 0.0f);
        ap.setMaterial(new Material(ambient, emissive,
diffuse, specular, 1.0f));
        return ap;
    }

    private static Appearance getPencilGraphiteAppearance() {
        Appearance ap = new Appearance();

        Color3f emissive = new Color3f(0.1f, 0.1f, 0.1f);
        Color3f ambient = new Color3f(0.1f, 0.1f, 0.1f);
        Color3f diffuse = new Color3f(0.1f, 0.1f, 0.1f);
        Color3f specular = new Color3f(0.1f, 0.1f, 0.1f);
        ap.setMaterial(new Material(ambient, emissive,
diffuse, specular, 1.0f));
    }

```

```
    return ap;  
}  
}
```

Результат роботи програми:

