

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

Факультет прикладної математики
Інженерія програмного забезпечення

ЛАБОРАТОРНА РОБОТА №5
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.

Мета:

1) Здобути навички імпорту моделей, побудованих у тривимірних редакторах, (об'єктів форматів .obj, .lwo, .3ds) до бібліотек.

Завдання

Імпортувати моделі тривимірних об'єктів форматів, що визначені варіантом. Створити реалістичну анімацію об'єкту. Додати до сцени фон, інші об'єкти для

надання сцені реалістичного вигляду. Для цього використати текстури, матеріали, імпортувати додаткові об'єкти з відкритих бібліотек, за бажанням створити прості об'єкти у графічному редакторі.

Студенти, які мають непарний номер варіанту у списку групи імпортують *моделі формату .obj*, парний

варіант – .lwo.obj, парний варіант – *.obj*, парний варіант – *.lwo.lwo.obj*, парний варіант – *.lwo*.

Варіант 13:

Використав свій варіант – модель кота.

Текст коду програм

Main.java

```
import javax.vecmath.*;

import com.sun.j3d.utils.image.TextureLoader;
import com.sun.j3d.utils.universe.*; import
javax.media.j3d.*;
import com.sun.j3d.utils.behaviors.vp.*;
import javax.swing.JFrame;
import com.sun.j3d.loaders.*;
import com.sun.j3d.loaders.objectfile.*;
import java.util.Map;

public class Cat extends JFrame{
    public Canvas3D myCanvas3D;
    static Map<String, Shape3D> nameMap;

    static TransformGroup wholeCat;
    static Transform3D transform3D;

    public Cat(){
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        myCanvas3D = new Canvas3D(SimpleUniverse.getPreferredConfiguration());
        SimpleUniverse simpUniv = new SimpleUniverse(myCanvas3D);
        simpUniv.getViewingPlatform().setNominalViewingTransform();

        // set the geometry and transformations
        createSceneGraph(simpUniv);
        addLight(simpUniv);

        OrbitBehavior ob = new OrbitBehavior(myCanvas3D);
        ob.setSchedulingBounds(new BoundingSphere(new Point3d(0.0,0.0,0.0),
Double.MAX_VALUE));
        simpUniv.getViewingPlatform().setViewPlatformBehavior(ob);

        setTitle("Cat");
        setSize(700,700);
        getContentPane().add("Center", myCanvas3D);
        setVisible(true);
    }

    public void createSceneGraph(SimpleUniverse
    su){ // loading object
        ObjectFile f = new ObjectFile(ObjectFile.RESIZE);
        BranchGroup CatBranchGroup = new BranchGroup();
        TextureLoader t = new TextureLoader("source_folder//ufo.jpg", myCanvas3D);
        Background CatBackground = new Background(t.getImage());

        Scene CatScene = null;
        try{
            CatScene = f.load("source_folder//cat_Cat.obj");
        }
        catch (Exception e){
            System.out.println("File loading failed:" + e);
        }
        nameMap = CatScene.getNamedObjects();
    }
}
```

```

wholeCat = new TransformGroup();

transform3D = new Transform3D(); transform3D.rotX(-
Math.PI / 2); wholeCat.setTransform(transform3D);
transform3D.setTranslation(new Vector3f(0, -1.3f,
0)); wholeCat.setTransform(transform3D);

transform3D.rotX(-Math.PI / 2);
wholeCat.setTransform(transform3D);

// start animation
Transform3D startTransformation = new Transform3D();
startTransformation.rotX(-Math.PI / 2);
startTransformation.setScale(1.0/6);
Transform3D combinedStartTransformation = new Transform3D();
combinedStartTransformation.mul(startTransformation);

TransformGroup CatStartTransformGroup = new
TransformGroup(combinedStartTransformation);

Appearance bodyApp = addAppearance();

TransformGroup sceneGroup = new TransformGroup();
TransformGroup tgBody = new TransformGroup();

Shape3D CatBodyShape = nameMap.get("19292_cat_Cat_v1");
CatBodyShape.setAppearance(bodyApp);
tgBody.addChild(CatBodyShape.cloneTree());

sceneGroup.addChild(tgBody.cloneTree());

// movement from start position
TransformGroup whiteTransXformGroup =
    translate( CatStartTransformGroup,
        new Vector3f(0.0f,0.0f,0.5f));

TransformGroup whiteRotXformGroup = rotate(whiteTransXformGroup,
new Alpha(10,5000));
CatBranchGroup.addChild(whiteRotXformGroup);
CatStartTransformGroup.addChild(sceneGroup);

// adding background to the branch group
BoundingSphere bounds = new BoundingSphere(new
Point3d(120.0,250.0,100.0),Double.MAX_VALUE);
CatBackground.setApplicationBounds(bounds);
CatBranchGroup.addChild(CatBackground);

CatBranchGroup.compile();
su.addBranchGraph(CatBranchGroup);
}

public void addLight(SimpleUniverse su){
    // Cat light
    BranchGroup bgLight = new BranchGroup();
    BoundingSphere bounds = new BoundingSphere(new Point3d(0.0,0.0,0.0), 100.0);
    Color3f lightColour1 = new Color3f(65/255f, 30/255f, 25/255f);
    Vector3f lightDir1 = new Vector3f(-1.0f,0.0f,-0.5f);
    DirectionalLight light1 = new DirectionalLight(lightColour1, lightDir1);
    light1.setInfluencingBounds(bounds); bgLight.addChild(light1);

```

```

        su.addBranchGraph(bgLight);
    }

    TransformGroup translate(Node node,Vector3f vector){
        Transform3D transform3D = new Transform3D();
        transform3D.setTranslation(vector);
        TransformGroup transformGroup =
            new TransformGroup();
        transformGroup.setTransform(transform3D);

        transformGroup.addChild(node);
        return transformGroup;
    }

    TransformGroup rotate(Node node,Alpha alpha){
        TransformGroup xformGroup = new
        TransformGroup(); xformGroup.setCapability(
            TransformGroup.ALLOW_TRANSFORM_WRITE);

        // Create an interpolator (calculates coordinates) for rotating the node
        RotationInterpolator interpolator =
            new RotationInterpolator(alpha,xformGroup);

        // Set the animation region for this interpolator
        interpolator.setSchedulingBounds(new BoundingSphere( new
            Point3d(0.0,0.0,0.0),1.0));

        //Populate the xform group.
        xformGroup.addChild(interpolator);
        xformGroup.addChild(node);

        return xformGroup;
    }

    private Appearance addAppearance(){
        Appearance CatAppearance = new Appearance();
        CatAppearance.setTexture(getTexture("source_folder//paper.jpg"));
        TextureAttributes texAttr = new TextureAttributes();
        CatAppearance.setTextureAttributes(texAttr);
        CatAppearance.setMaterial(getMaterial());
        Shape3D plane = nameMap.get("19292_cat_Cat_v1");
        plane.setAppearance(CatAppearance); return
        CatAppearance;
    }

    Texture getTexture(String path) {
        TextureLoader textureLoader = new TextureLoader(path,"LUMINANCE",myCanvas3D);
        Texture texture = textureLoader.getTexture();
        texture.setBoundaryModeS(Texture.WRAP);
        texture.setBoundaryModeT(Texture.WRAP);
        texture.setBoundaryColor( new Color4f( 0.0f, 1.0f, 0.0f, 0.0f )
        ); return texture;
    }

    Material getMaterial() {
        Material material = new Material();
        material.setAmbientColor ( new Color3f( 0.9f, 0.9f, 0.9f)
        ); material.setDiffuseColor ( new Color3f( 1f, 1f, 1f ) );
        material.setSpecularColor( new Color3f( 1f, 1f, 1f ) );
        material.setShininess( 0.3f );
        material.setLightingEnable(true); return material;
    }

```

```
    }

    public static void main(String[] args) {
        Cat start = new Cat();
    }
}
```

Результати роботи програми





Висновки

Виконавши дану лабораторну роботу я здобув навички імпорту моделей, побудованих у тривимірних редакторах, об'єктів форматів .obj до бібліотек.