

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

Факультет прикладної математики
Інженерія програмного забезпечення

ЛАБОРАТОРНА РОБОТА №6
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) навчитися анімувати складні об'єкти тривимірної сцени.

Завдання

Виконати анімацію тривимірної сцени за варіантом.

Варіант 13: (3)

Анімація риби fish.obj. Риба повинна рухати плавцями, хвостом, головою, рухатися по екрану.

Текст коду програм

Fish.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.Hashtable;
import java.util.Enumeraation;

public class Fish extends JFrame{
    public Canvas3D myCanvas3D;

    public Fish(){
        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("Fish");
        setSize(700,700);
        getContentPane().add("Center", myCanvas3D);
        setVisible(true);
    }

    public void createSceneGraph(SimpleUniverse
su){ // loading object
    ObjectFile f = new ObjectFile(ObjectFile.RESIZE);
    BoundingSphere bs = new BoundingSphere(new
Point3d(0.0,0.0,0.0),Double.MAX_VALUE);
    String name;
    BranchGroup fishBranchGroup = new BranchGroup();
    TextureLoader t = new TextureLoader("source_folder//ocean.jpg",
myCanvas3D); Background fishBackground = new Background(t.getImage());

    Scene fishScene = null;
    try{
        fishScene = f.load("models/fish.obj");
    }
    catch (Exception e){
        System.out.println("File loading failed:" + e);
    }
    Hashtable roachNamedObjects = fishScene.getNamedObjects();
```

```

Enumeration enumer = roachNamedObjects.keys();
while (enumer.hasMoreElements()){
    name = (String) enumer.nextElement();
    System.out.println("Name: " + name);
}

// start animation
Transform3D startTransformation = new Transform3D();
startTransformation.setScale(1.0/6);
Transform3D combinedStartTransformation = new Transform3D();
combinedStartTransformation.rotY(-3*Math.PI/2);
combinedStartTransformation.mul(startTransformation);

TransformGroup fishStartTransformGroup =
new TransformGroup(combinedStartTransformation);

    Appearance bodyApp = addAppearance();

    int movesCount = 100; // moves count
    int movesDuration = 500; // moves for 0,3 seconds
    int startTime = 0; // launch animation after timeStart seconds

    // fin 1
    Alpha fin1_1RotAlpha = new Alpha(movesCount, Alpha.INCREASING_ENABLE,
startTime, 0, movesDuration,0,0,0,0,0);

    Shape3D fin1 = (Shape3D) roachNamedObjects.get("fin1");
    fin1.setAppearance(bodyApp);
    TransformGroup fin1TG = new TransformGroup();
    fin1TG.addChild(fin1.cloneTree());

    Transform3D fin1RotAxis = new Transform3D();
    fin1RotAxis.set(new Vector3d(0, -0.101, 0.52));
    fin1RotAxis.setRotation(new AxisAngle4d(0, 0, 0, 0));

    RotationInterpolator fin1rot = new RotationInterpolator(fin1_1RotAlpha,
fin1TG, fin1RotAxis,(float) 0.0f, (float) Math.PI/3);
    fin1rot.setSchedulingBounds(bs);
    fin1TG.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
    fin1TG.addChild(fin1rot);

    // fin 2
    Alpha fin2_1RotAlpha = new Alpha(movesCount, Alpha.INCREASING_ENABLE,
startTime, 0, movesDuration,0,0,0,0,0);

    Shape3D fin2 = (Shape3D) roachNamedObjects.get("fin2");
    fin2.setAppearance(bodyApp);
    TransformGroup fin2TG = new TransformGroup();
    fin2TG.addChild(fin2.cloneTree());

    Transform3D fin2RotAxis = new Transform3D();
    fin2RotAxis.set(new Vector3d(0, -0.2, 0.3));
    fin2RotAxis.setRotation(new AxisAngle4d(0, 0, 0, 0));

    RotationInterpolator fin2rot = new RotationInterpolator(fin2_1RotAlpha,
fin2TG, fin2RotAxis,(float) 0.0f, (float) Math.PI/3);
    fin2rot.setSchedulingBounds(bs);
    fin2TG.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
    fin2TG.addChild(fin2rot);

    // tail
    Alpha tail_RotAlpha = new Alpha(movesCount, Alpha.INCREASING_ENABLE, startTime,

```

```

0, movesDuration,0,0,0,0,0);
    Shape3D tail = (Shape3D) roachNamedObjects.get("tail");
    tail.setAppearance(bodyApp);
    TransformGroup tailTG = new TransformGroup();

    tailTG.addChild(tail.cloneTree());

    Transform3D tailRotAxis = new Transform3D();
    tailRotAxis.set(new Vector3d(0, 0, 0));
    tailRotAxis.setRotation(new AxisAngle4d(0, 0, 0, 0));

    RotationInterpolator tailRot = new RotationInterpolator(tail_RotAlpha,
tailTG, tailRotAxis, (float) -Math.PI/20, (float) Math.PI/20); // Math.PI*2
    tailRot.setSchedulingBounds(bs);
    tailTG.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
    tailTG.addChild(tailRot);

    TransformGroup sceneGroup = new
    TransformGroup(); sceneGroup.addChild(fin1TG);
    sceneGroup.addChild(fin2TG);
    sceneGroup.addChild(tailTG);

    TransformGroup tgBody = new TransformGroup();

    Shape3D fishBodyShape = (Shape3D) roachNamedObjects.get("rt_body");
    fishBodyShape.setAppearance(bodyApp);
    tgBody.addChild(fishBodyShape.cloneTree());

    Shape3D headShape = (Shape3D) roachNamedObjects.get("head");
    headShape.setAppearance(bodyApp);
    tgBody.addChild(headShape.cloneTree());

    Shape3D eyeShape = (Shape3D) roachNamedObjects.get("rt_eye");
    tgBody.addChild(eyeShape.cloneTree());

    Shape3D ventrtalFinShape = (Shape3D) roachNamedObjects.get("ventral_fin2");
    ventrtalFinShape.setAppearance(bodyApp);
    tgBody.addChild(ventrtalFinShape.cloneTree());

    Shape3D ventrtalFin2Shape = (Shape3D) roachNamedObjects.get("ventral_finq");
    ventrtalFin2Shape.setAppearance(bodyApp);
    tgBody.addChild(ventrtalFin2Shape.cloneTree());

    sceneGroup.addChild(tgBody.cloneTree());

    TransformGroup whiteTransXformGroup =
        translate( fishStartTransformGroup,
        new Vector3f(0.0f,0.0f,0.5f));

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

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

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

```

```

private Appearance addAppearance(){
    Appearance boatAppearance = new Appearance();
    boatAppearance.setTexture(getTexture("source_folder//ocean.jpg"));
    TextureAttributes texAttr = new TextureAttributes();
    texAttr.setTextureMode(TextureAttributes.COMBINE);
    boatAppearance.setTextureAttributes(texAttr);
    boatAppearance.setMaterial(getMaterial()); return boatAppearance;
}

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 void addLight(SimpleUniverse su){
    BranchGroup bgLight = new BranchGroup();
    BoundingSphere bounds = new BoundingSphere(new Point3d(10.0,0.0,0.0), 100.0);
    Color3f lightColour1 = new Color3f(100/255f, 100/255f, 100/255f);
    Vector3f lightDir1 = new Vector3f(-1.0f,-1.0f,-1.0f);
    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 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;
}

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

Результати роботи програми



Висновки

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