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 № 6

з дисципліни “Математичні та алгоритмічні основи комп’ютерної графіки”

Виконав
студент III курсу
групи КП-81
Дикий Ілля
(*прізвище, ім’я, по батькові*)

Зарахована
“____” “_____” 20___ р.
викладачем
Шкурат Оксаною Сергіївною
(*прізвище, ім’я, по батькові*)

варіант № 3

Київ 2021

Завдання

Виконати анімацію тривимірної сцени за варіантом.

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

Лістинг коду програми

```
import com.sun.j3d.loaders.Scene;
import com.sun.j3d.loaders.objectfile.ObjectFile;
import com.sun.j3d.utils.behaviors.vp.OrbitBehavior;
import com.sun.j3d.utils.image.TextureLoader;
import com.sun.j3d.utils.universe.SimpleUniverse;

import javax.media.j3d.*;
import javax.swing.*;
import javax.vecmath.Color3f;
import javax.vecmath.Point3d;
import javax.vecmath.Vector3d;
import javax.vecmath.Vector3f;
import java.util.Enumeration;
import java.util.Hashtable;

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();

        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){
        ObjectFile f = new ObjectFile(ObjectFile.RESIZE);
        BoundingSphere bs = new BoundingSphere(new
Point3d(0.0,0.0,0.0),Double.MAX_VALUE);
        String name;
        BranchGroup trainerBranchGroup = new BranchGroup();
        TextureLoader t = new
TextureLoader("/home/stilpert/Education/MAOKG/lab6/data/background.jpg",
myCanvas3D);
        Background trainerBackground =  new Background(t.getImage());

        Scene trainerScene = null;
        try{
            trainerScene =
f.load("/home/stilpert/Education/MAOKG/lab6/data/fish.obj");
        }
        catch (Exception e){
            System.out.println("File loading failed:" + e);
        }
        Hashtable roachNamedObjects = trainerScene.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(2.0/6);
        Transform3D combinedStartTransformation = new Transform3D();
        combinedStartTransformation.mul(startTransformation);

        TransformGroup scratStartTransformGroup = new
TransformGroup(combinedStartTransformation);
```

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

        // head
        Appearance headApp = new Appearance();
        setToMyDefaultAppearance(headApp, new Color3f(0.1f, 0.2f, 0.1f));

        Alpha headRotAlpha = new Alpha(movesCount, Alpha.INCREASING_ENABLE,
startTime, 0, movesDuration,0,0,0,0,0);

        Shape3D head = (Shape3D) roachNamedObjects.get("head");
        head.setAppearance(headApp);
        TransformGroup headTG = new TransformGroup();
        headTG.addChild(head.cloneTree());

        Transform3D headRotAxis = new Transform3D();
        headRotAxis.setTranslation(new Vector3f(0.0f, 0.0f, 0.5f));

        RotationInterpolator headRot = new RotationInterpolator(headRotAlpha,
headTG, headRotAxis, (float) -Math.PI/4, (float) Math.PI/4);
        headRot.setSchedulingBounds(bs);
        headTG.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        headTG.addChild(headRot);

        // ventralFin
        Appearance ventralFinApp = new Appearance();
        setToMyDefaultAppearance(ventralFinApp, new Color3f(0.9f, 0.0f,
0.0f));

        Alpha ventralFinAlpha = new Alpha(movesCount,
Alpha.INCREASING_ENABLE, startTime, 0, movesDuration,0,0,0,0,0);

        Shape3D ventralFin = (Shape3D) roachNamedObjects.get("ventral_finq");
        ventralFin.setAppearance(ventralFinApp);
        TransformGroup ventralFinTG = new TransformGroup();
        ventralFinTG.addChild(ventralFin.cloneTree());

        Transform3D ventralFinRotAxis = new Transform3D();
```

```
        RotationInterpolator ventralFinrot = new
RotationInterpolator(ventralFinAlpha, ventralFinTG, ventralFinRotAxis, 0.0f,
(float) Math.PI/3); // Math.PI*2
        ventralFinrot.setSchedulingBounds(bs);
        ventralFinTG.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        ventralFinTG.addChild(ventralFinrot);

        // ventralFin2

        Shape3D ventralFin2 = (Shape3D) roachNamedObjects.get("ventral_fin2");
        ventralFin2.setAppearance(ventralFinApp);
        TransformGroup ventralFin2TG = new TransformGroup();
        ventralFin2TG.addChild(ventralFin2.cloneTree());

        Transform3D ventralFin2RotAxis = new Transform3D();

        RotationInterpolator ventralFin2rot = new
RotationInterpolator(ventralFinAlpha, ventralFin2TG, ventralFin2RotAxis,
0.0f, (float) Math.PI/3); // Math.PI*2
        ventralFin2rot.setSchedulingBounds(bs);
        ventralFin2TG.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        ventralFin2TG.addChild(ventralFin2rot);

        // fin1
        Appearance finApp = new Appearance();
        setToMyDefaultAppearance(finApp, new Color3f(0.9f, 0.0f, 0.0f));

        Alpha finAlpha = new Alpha(movesCount, Alpha.INCREASING_ENABLE,
startTime, 0, movesDuration,0,0,0,0,0);

        Shape3D fin = (Shape3D) roachNamedObjects.get("fin1");
        fin.setAppearance(finApp);
        TransformGroup finTG = new TransformGroup();
        finTG.addChild(fin.cloneTree());

        Transform3D finRotAxis = new Transform3D();
        finRotAxis.setTranslation(new Vector3f(0.0f, 0.0f, 0.5f));

        RotationInterpolator finrot = new RotationInterpolator(finAlpha,
finTG, finRotAxis, 0.0f, (float) Math.PI/3); // Math.PI*2
        finrot.setSchedulingBounds(bs);
```

```
        finTG.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        finTG.addChild(finrot);

        // fin2

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

        Transform3D fin2RotAxis = new Transform3D();
        fin2RotAxis.setTranslation(new Vector3f(0.0f, 0.0f, 0.5f));

        RotationInterpolator fin2rot = new RotationInterpolator(finAlpha,
fin2TG, fin2RotAxis, 0.0f, (float) -Math.PI/3); // Math.PI*2
        fin2rot.setSchedulingBounds(bs);
        fin2TG.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        fin2TG.addChild(fin2rot);

        // tail
        Appearance tailApp = new Appearance();
        setToMyDefaultAppearance(tailApp, new Color3f(0.9f, 0.0f, 0.0f));

        Alpha tailAlpha = new Alpha(movesCount, Alpha.INCREASING_ENABLE,
startTime, 0, movesDuration,0,0,0,0,0);

        Shape3D tail = (Shape3D) roachNamedObjects.get("tail");
        tail.setAppearance(tailApp);
        TransformGroup tailTG = new TransformGroup();
        tailTG.addChild(tail.cloneTree());

        Transform3D tailRotAxis = new Transform3D();
        Vector3f vectorTail = new Vector3f(0.0f, 0.0f, -0.6f);
        tailRotAxis.setTranslation(vectorTail);

        RotationInterpolator tailrot = new RotationInterpolator(tailAlpha,
tailTG, tailRotAxis, (float) -Math.PI/3, (float) Math.PI/3);
        tailrot.setSchedulingBounds(bs);
        tailTG.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        tailTG.addChild(tailrot);
```

```
        // body
        Appearance bodyApp = new Appearance();
        setToMyDefaultAppearance(bodyApp, new Color3f(0.1f, 0.2f, 0.1f));

        TransformGroup sceneGroup = new TransformGroup();
        sceneGroup.addChild(headTG);
        sceneGroup.addChild(ventralFinTG);
        sceneGroup.addChild(ventralFin2TG);
        sceneGroup.addChild(finTG);
        sceneGroup.addChild(fin2TG);
        sceneGroup.addChild(tailTG);

        TransformGroup tgBody = new TransformGroup();
        Shape3D nShape = (Shape3D) roachNamedObjects.get("rt_body");
        nShape.setAppearance(bodyApp);
        tgBody.addChild(nShape.cloneTree());
        sceneGroup.addChild(tgBody.cloneTree());

        TransformGroup whiteTransXformGroup = translate(
                scratStartTransformGroup,
                new Vector3f(0.0f,0.0f,0.7f));

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

        BoundingSphere bounds = new BoundingSphere(new
Point3d(120.0,250.0,100.0),Double.MAX_VALUE);
        trainerBackground.setApplicationBounds(bounds);
        trainerBranchGroup.addChild(trainerBackground);

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

    public void addLight(SimpleUniverse su){
        BranchGroup bgLight = new BranchGroup();
        BoundingSphere bounds = new BoundingSphere(new Point3d(0.0,0.0,0.0),
100.0);
        Color3f lightColour1 = new Color3f(1.0f,1.0f,1.0f);
```

```
        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);
    }

    private TransformGroup translate(Node node, Vector3f vector){

        Transform3D transform3D = new Transform3D();
        Transform3D rotY = new Transform3D();
        transform3D.setTranslation(vector);
        rotY.rotY(Math.PI/2);
        transform3D.mul(rotY);
        TransformGroup transformGroup =
                new TransformGroup();
        transformGroup.setTransform(transform3D);

        transformGroup.addChild(node);
        return transformGroup;
    }

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

        RotationInterpolator interpolator = new
RotationInterpolator(alpha,xformGroup);

        interpolator.setSchedulingBounds(new BoundingSphere( new
Point3d(0.0,0.0,0.0),1.0));

        xformGroup.addChild(interpolator);
        xformGroup.addChild(node);

        return xformGroup;
    }

    public static void setToMyDefaultAppearance(Appearance app, Color3f col) {
        app.setMaterial(new Material(col, col, col, col, 150.0f));
```

```
    }

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

}
```

Результат