МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота № 5
з дисципліни “ МАОКГ”

Виконала
студентка III курсу
групи КП-83
Снітко Маріанна Дмитрівна
(*прізвище, ім’я, по батькові*)
Варіант № 19

Зарахована
“____” “_____” 2021 р.
викладачем
Шкурат О. С.
(*прізвище, ім’я, по батькові*)

Київ 2021

Лістинг коду програми

AnimationFish.java

```
package application;

import javax.media.j3d.Transform3D;
import javax.media.j3d.TransformGroup;
import javax.swing.*;
import javax.vecmath.Vector3d;
import javax.vecmath.Vector3f;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;

public class AnimationFish implements ActionListener, KeyListener {

    private TransformGroup fish;
    private Transform3D translateTransform;
    private Transform3D rotateTransformX;
    private Transform3D rotateTransformY;
    private Transform3D rotateTransformZ;

    private float zoom = 0.5f;
    private float xloc = 0.3f;
    private float yloc = 0.3f;
    private float zloc = 0.2f;
    private float delta = 0.03f;

    private boolean pressedW = false;
    private boolean pressedS = false;
    private boolean pressedA = false;
    private boolean pressedD = false;

    public AnimationFish(TransformGroup fish, Transform3D trans){
        this.fish = fish;
        this.translateTransform = trans;

        rotateTransformX= new Transform3D();
        rotateTransformY= new Transform3D();
        rotateTransformZ= new Transform3D();

        Main.canvas.addKeyListener(this);
        Timer timer = new Timer(20, this);
        timer.start();
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        Move();
    }

    private void Move(){
        if(pressedA){
            if (xloc*2 >= -3) {
                xloc -= delta;
            }
        } else if(pressedS){
            if(yloc*2 >= -4){
                yloc -= delta;
            }
        } else if(pressedW){
            if(yloc <= 0.03){
                yloc += delta;
```

```
            }
        } else if(pressedD){
            if(xloc <= 1.5){
                xloc += delta;
            }
        }
        translateTransform.setScale(new Vector3d(zoom,zoom,zoom));
        translateTransform.setTranslation(new Vector3f(xloc,yloc,zloc));
        fish.setTransform(translateTransform);
    }

    @Override
    public void keyTyped(KeyEvent e) {
        //Invoked when a key has been typed.
    }

    @Override
    public void keyPressed(KeyEvent e) {
        switch (e.getKeyCode()) {
            case 87: // W
                pressedW = true;
                if(pressedA){
                    rotateTransformZ.rotZ(-Math.PI/2);
                    rotateTransformY.rotY(Math.PI/2);
                    translateTransform.mul(rotateTransformZ);
                    translateTransform.mul(rotateTransformY);
                    pressedA = false;
                } else if(pressedD){
                    rotateTransformZ.rotZ(Math.PI/2);
                    rotateTransformY.rotY(Math.PI/2);
                    translateTransform.mul(rotateTransformZ);
                    translateTransform.mul(rotateTransformY);
                    pressedD = false;
                } else if(pressedS){
                    rotateTransformZ.rotZ(Math.PI);
                    translateTransform.mul(rotateTransformZ);
                    pressedS = false;
                }
                break;
            case 83: // S
                pressedS = true;
                if(pressedA){
                    rotateTransformZ.rotZ(Math.PI/2);
                    rotateTransformY.rotY(-Math.PI/2);
                    translateTransform.mul(rotateTransformZ);
                    translateTransform.mul(rotateTransformY);
                    pressedA = false;
                } else if(pressedW){
                    rotateTransformZ.rotZ(Math.PI);
                    translateTransform.mul(rotateTransformZ);
                    pressedW = false;
                } else if(pressedD){
                    rotateTransformZ.rotZ(-Math.PI/2);
                    rotateTransformY.rotY(-Math.PI/2);
                    translateTransform.mul(rotateTransformZ);
                    translateTransform.mul(rotateTransformY);
                    pressedD = false;
                }
                break;
            case 65: // A
                pressedA = true;
                if(pressedW){
                    rotateTransformX.rotX(-Math.PI/2);
                    rotateTransformY.rotY(-Math.PI/2);
                    translateTransform.mul(rotateTransformX);
                    translateTransform.mul(rotateTransformY);
                    pressedW = false;
```

```
                } else if(pressedD){
                    rotateTransformZ.rotZ(Math.PI);
                    translateTransform.mul(rotateTransformZ);
                    pressedD = false;
                } else if(pressedS){
                    rotateTransformX.rotX(-Math.PI/2);
                    rotateTransformY.rotY(Math.PI/2);
                    translateTransform.mul(rotateTransformX);
                    translateTransform.mul(rotateTransformY);
                    pressedS = false;
                }
                break;
            case 68: // D
                pressedD = true;
                if(pressedW){
                    rotateTransformZ.rotZ(-Math.PI/2);
                    rotateTransformX.rotX(Math.PI/2);
                    translateTransform.mul(rotateTransformZ);
                    translateTransform.mul(rotateTransformX);
                    pressedW = false;
                } else if(pressedA){
                    rotateTransformZ.rotZ(Math.PI);
                    translateTransform.mul(rotateTransformZ);
                    pressedA = false;
                } else if(pressedS){
                    rotateTransformZ.rotZ(Math.PI/2);
                    rotateTransformX.rotX(Math.PI/2);
                    translateTransform.mul(rotateTransformZ);
                    translateTransform.mul(rotateTransformX);
                    pressedS = false;
                }
                break;
        }
    }

    @Override
    public void keyReleased(KeyEvent e) {

    }
}
```

Main.java

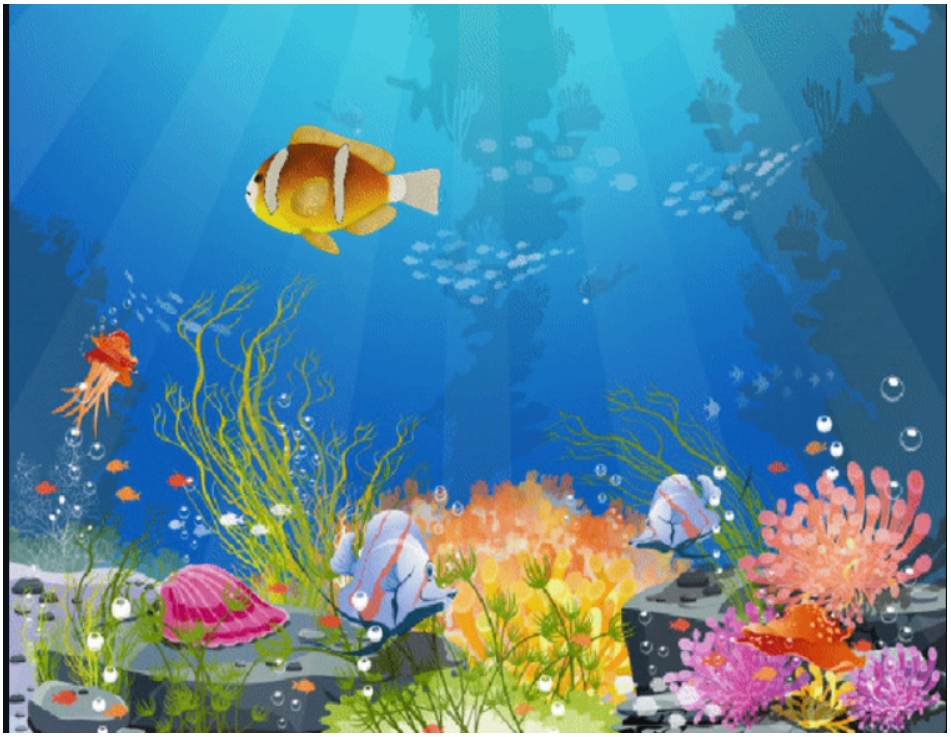
```
package application;

import com.sun.j3d.loaders.Scene;
import com.sun.j3d.loaders.objectfile.ObjectFile;
import com.sun.j3d.utils.image.TextureLoader;
import com.sun.j3d.utils.universe.SimpleUniverse;
import com.sun.j3d.utils.universe.ViewingPlatform;

import javax.media.j3d.Appearance;
import javax.media.j3d.Background;
import javax.media.j3d.BoundingSphere;
import javax.media.j3d.Bounds;
import javax.media.j3d.BranchGroup;
import javax.media.j3d.Canvas3D;
import javax.media.j3d.DirectionalLight;
import javax.media.j3d.Shape3D;
import javax.media.j3d.Texture;
```

```
import javax.media.j3d.Transform3D;
import javax.media.j3d.TransformGroup;
import javax.swing.*;
import javax.vecmath.Color3f;
import javax.vecmath.Color4f;
import javax.vecmath.Point3d;
import javax.vecmath.Vector3f;
import java.awt.*;
import java.io.FileReader;
import java.io.IOException;
import java.util.Map;

public class Main extends JFrame {
    static SimpleUniverse universe;
    static Scene scene;
    static Map<String, Shape3D> nameMap;
    static BranchGroup root;
    static Canvas3D canvas;

    static TransformGroup fish;
    static Transform3D transform3D;

    public Main() throws IOException {
        configureWindow();
        configureCanvas();
        configureUniverse();
        addModelToUniverse();
        setFishElementsList();
        addAppearance();
        addImageBackground();
        addLightToUniverse();
        changeViewAngle();
        root.compile();
        universe.addBranchGraph(root);
    }

    private void configureWindow()  {
        setTitle("Fish animation");
        setSize(760,640);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    private void configureCanvas(){
        canvas = new Canvas3D(SimpleUniverse.getPreferredConfiguration());
        canvas.setDoubleBufferEnable(true);
        getContentPane().add(canvas, BorderLayout.CENTER);
    }

    private void configureUniverse(){
        root = new BranchGroup();
        universe = new SimpleUniverse(canvas);
        universe.getViewingPlatform().setNominalViewingTransform();
    }

    private void addModelToUniverse() throws IOException{
        scene = getSceneFromFile("source_folder/12265_Fish_v1_L2.obj");
        root = scene.getSceneGroup();
    }

    public static Scene getSceneFromFile(String location) throws IOException {
        ObjectFile file = new ObjectFile(ObjectFile.RESIZE);
        file.setFlags (ObjectFile.RESIZE | ObjectFile.TRIANGULATE |
ObjectFile.STRIPIFY);
        return file.load(new FileReader(location));
    }

    private void addLightToUniverse(){
```

```
        Bounds bounds = new BoundingSphere();
        Color3f color = new Color3f(65/255f, 30/255f, 25/255f);
        Vector3f lightdirection = new Vector3f(-1f,-1f,-1f);
        DirectionalLight dirlight = new DirectionalLight(color,lightdirection);
        dirlight.setInfluencingBounds(bounds);
        root.addChild(dirlight);
    }

    private void changeViewAngle(){
        ViewingPlatform vp = universe.getViewingPlatform();
        TransformGroup vpGroup =
vp.getMultiTransformGroup().getTransformGroup(0);
        Transform3D vpTranslation = new Transform3D();
        Vector3f translationVector = new Vector3f(0.0F, -1.2F, 6F);
        vpTranslation.setTranslation(translationVector);
        vpGroup.setTransform(vpTranslation);
    }

    private void printModelElementsList(Map<String,Shape3D> nameMap){
        for (String name : nameMap.keySet()) {
            System.out.printf("Name: %s\n", name);}
    }

    private void setFishElementsList() {
        nameMap = scene.getNamedObjects();
        //Print elements of your model:
        printModelElementsList(nameMap);

        fish = new TransformGroup();
        transform3D = new Transform3D();
        transform3D.rotX(-Math.PI/2);
        fish.setTransform(transform3D);
        root.removeChild(nameMap.get("12265_fish"));
        fish.addChild(nameMap.get("12265_fish"));
        fish.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        root.addChild(fish);
    }

    private Texture getTexture(String path) {
        TextureLoader textureLoader = new TextureLoader(path,"RGP", new
Container());
        Texture texture = textureLoader.getTexture();
        texture.setBoundaryModeS(Texture.WRAP);
        texture.setBoundaryModeT(Texture.WRAP);
        texture.setBoundaryColor( new Color4f( Color.BLUE ) );
        return texture;
    }

    private void addAppearance(){
        Appearance fishAppearance = new Appearance();
        fishAppearance.setTexture(getTexture("source_folder/fish.jpg"));
        Shape3D fish = nameMap.get("12265_fish");
        fish.setAppearance(fishAppearance);
    }

    private void addImageBackground(){
        TextureLoader t = new TextureLoader("source_folder/water.jpg", canvas);
        Background background = new Background(t.getImage());
        background.setImageScaleMode(Background.SCALE_FIT_ALL);
        BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0,
0.0),100.0);
        background.setApplicationBounds(bounds);
        root.addChild(background);
    }

    public static void main(String[]args){
        try {
```

```
            Main window = new Main();
            AnimationFish fishMovement = new AnimationFish(fish, transform3D);
            window.addKeyListener(fishMovement);
            window.setVisible(true);
        }
        catch (IOException ex) {
            System.out.println(ex.getMessage());
        }
    }

}
```

Результати роботи програми



Висновки

Виконавши дану лабораторну роботу, я здобула навички імпорту моделей, побудованих у тривимірних редакторах, (об’єктів форматів .obj, .lwo, .3ds) до бібліотеки java3D. Також навчилася анімувати імпортовані об’єкти.

Система була написана на мові програмування **Java.**