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 № 5

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

Виконала
студентка III курсу
групи КП-82
Морщак Каріна
(*прізвище, ім’я, по батькові*)
варіант № 14

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

Київ 2021

Варіант завдання

Тема: Імпорт тривімирних моделей у середовище програмування java3D, обробка та маніапуляція цих зображень.

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

Лістинг коду програми

Main.java

```
package main;

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 javax.media.j3d.*;
import javax.swing.*;
import javax.vecmath.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Map;

public class Main extends JFrame implements ActionListener,
KeyListener {
    private final static String trexModelLocation = "cat.obj";
    private final static String backgroundLocation = "field.jpg";
    private final BranchGroup root = new BranchGroup();
    private final Canvas3D canvas = new
Canvas3D(SimpleUniverse.getPreferredConfiguration());
    private final TransformGroup trexGroup = new TransformGroup();
    private final Transform3D transform3D = new Transform3D();
    private final Transform3D rotateTransformX = new Transform3D();
    private final Transform3D rotateTransformY = new Transform3D();
    private final Transform3D rotateTransformZ = new Transform3D();
    private final ClassLoader classLoader =
Thread.currentThread().getContextClassLoader();
    private SimpleUniverse universe;
```

```
    private Scene trex;
    private Background background;
    private Map<String, Shape3D> nameMap;

    //розмір та позиція
    private float x_location_current = -3;
    private float scale_cur = 0.3f;


    public static void main(String[] args) {
        try {
            var window = new Main();
            window.addKeyListener(window);
            window.setVisible(true);
        } catch (IOException e) {
            System.err.println(e.getMessage());
        }
    }

    public Main() throws IOException {
        initialize();
        addTexture();
        printModelElementsList(nameMap);
        addImageBackground();
        addLight();
        setTexture();
        setInitialViewAngle();
        setInitialLocation();
        root.compile();
        universe.addBranchGraph(root);
    }

    private void setInitialLocation() {
        transform3D.setTranslation(new Vector3f(x_location_current,
0, 0));
        transform3D.setScale(scale_cur);
        trexGroup.setTransform(transform3D);
    }
```

```
    private void initialize() throws IOException {
        // window settings
        setTitle("Cat");
        setSize(1000, 700);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        canvas.setDoubleBufferEnable(true);
        getContentPane().add(canvas, BorderLayout.CENTER);

        universe = new SimpleUniverse(canvas);
        universe.getViewingPlatform().setNominalViewingTransform();

        canvas.addKeyListener(this);
        trex = getSceneFromFile();
    }

    //налаштування освітлення
    private void addLight() {
        var dirLight = new DirectionalLight(
                new Color3f(Color.WHITE),
                new Vector3f(4.0f, -7.0f, -12.0f)
        );

        dirLight.setInfluencingBounds(new BoundingSphere(new
Point3d(), 1000));
        root.addChild(dirLight);

        var ambientLight = new AmbientLight(new
Color3f(Color.WHITE));
        var directionalLight = new DirectionalLight(
                new Color3f(Color.BLACK),
                new Vector3f(-1F, -1F, -1F)
        );
        var influenceRegion = new BoundingSphere(new Point3d(),
1000);
        ambientLight.setInfluencingBounds(influenceRegion);
        directionalLight.setInfluencingBounds(influenceRegion);
```

```
        root.addChild(ambientLight);
        root.addChild(directionalLight);
    }

    private TextureLoader getTextureLoader(String path) throws
IOException {
        var textureResource = classLoader.getResource(path);
        if (textureResource == null) {
            throw new IOException("Couldn't find texture: " + path);
        }
        return new TextureLoader(textureResource.getPath(), canvas);
    }

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

    private Shape3D getModelShape3D(String name, String path) throws
IOException {
     Scene scene = getSceneFromFile(path);
     Map<String, Shape3D> map = scene.getNamedObjects();
     printModelElementsList(map);
     Shape3D shape = map.get(name);
     scene.getSceneGroup().removeChild(shape);
     return shape;
    }

    private void setTexture() throws IOException {
     Shape3D shape1 = getModelShape3D("cat", "Lab5/src/cat.obj");
     Transform3D transform3D = new Transform3D();
     transform3D.setScale(new Vector3d(0.7, 0.7, 0.7));
        Transform3D rotationY = new Transform3D();
        rotationY.rotY(Math.PI/2);
        transform3D.mul(rotationY);
        Appearance appearance = new Appearance();
```

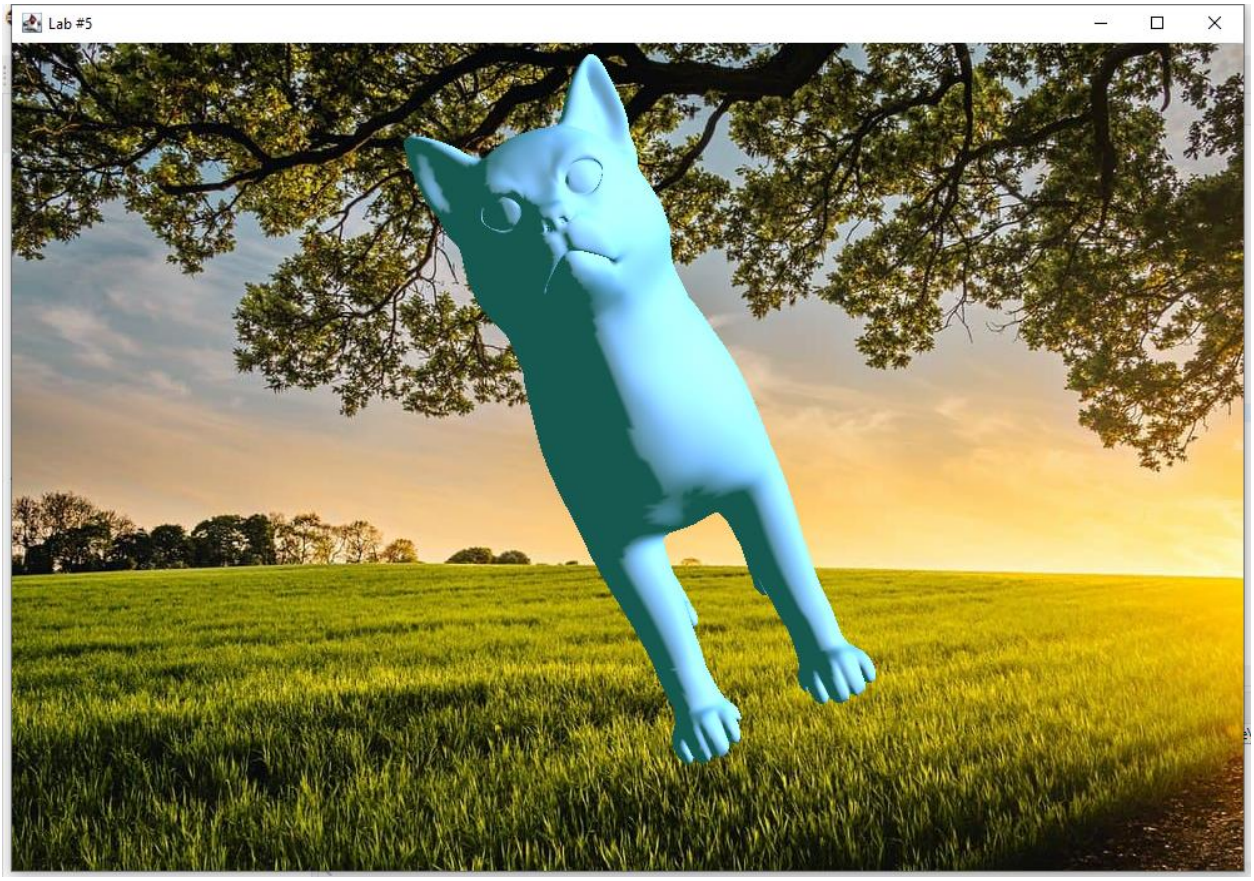
```
        TextureLoader textureLoader = new
TextureLoader("Lab5/src/fur.jpg", "LUMINANCE", canvas);
        Texture texture = textureLoader.getTexture();
        texture.setBoundaryModeS(Texture.WRAP);
        texture.setBoundaryModeT(Texture.WRAP);
        texture.setBoundaryColor(new Color4f(0.0f, 1.0f, 0.0f,
0.0f));
        TextureAttributes attrs = new TextureAttributes();
        attrs.setTextureMode(TextureAttributes.MODULATE);
        appearance.setTextureAttributes(attrs);
        Color3f emissive = new Color3f(new Color(0, 0, 0));
           Color3f ambient = new Color3f(new Color(22, 89, 80));
           Color3f diffuse = new Color3f(new Color(69, 127, 233));
           Color3f specular = new Color3f(new Color(106, 77, 113));
           appearance.setMaterial(new Material(ambient, emissive,
diffuse, specular, 1.0f));
        shape1.setAppearance(appearance);
     trexGroup.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
     trexGroup.addChild(shape1);
     trexGroup.setTransform(transform3D);
    }
    //додамо матеріали
    private Material getMaterial() {
        var material = new Material();
        material.setAmbientColor(new Color3f(new Color(243, 142,
221)));
        material.setDiffuseColor(new Color3f(new Color(255, 133,
207)));
        material.setSpecularColor(new Color3f(new Color(255, 103,
195)));
        material.setLightingEnable(true);
        return material;
    }

    private void addTexture() throws IOException {
        nameMap = trex.getNamedObjects();
        trexGroup.addChild(trex.getSceneGroup());
```

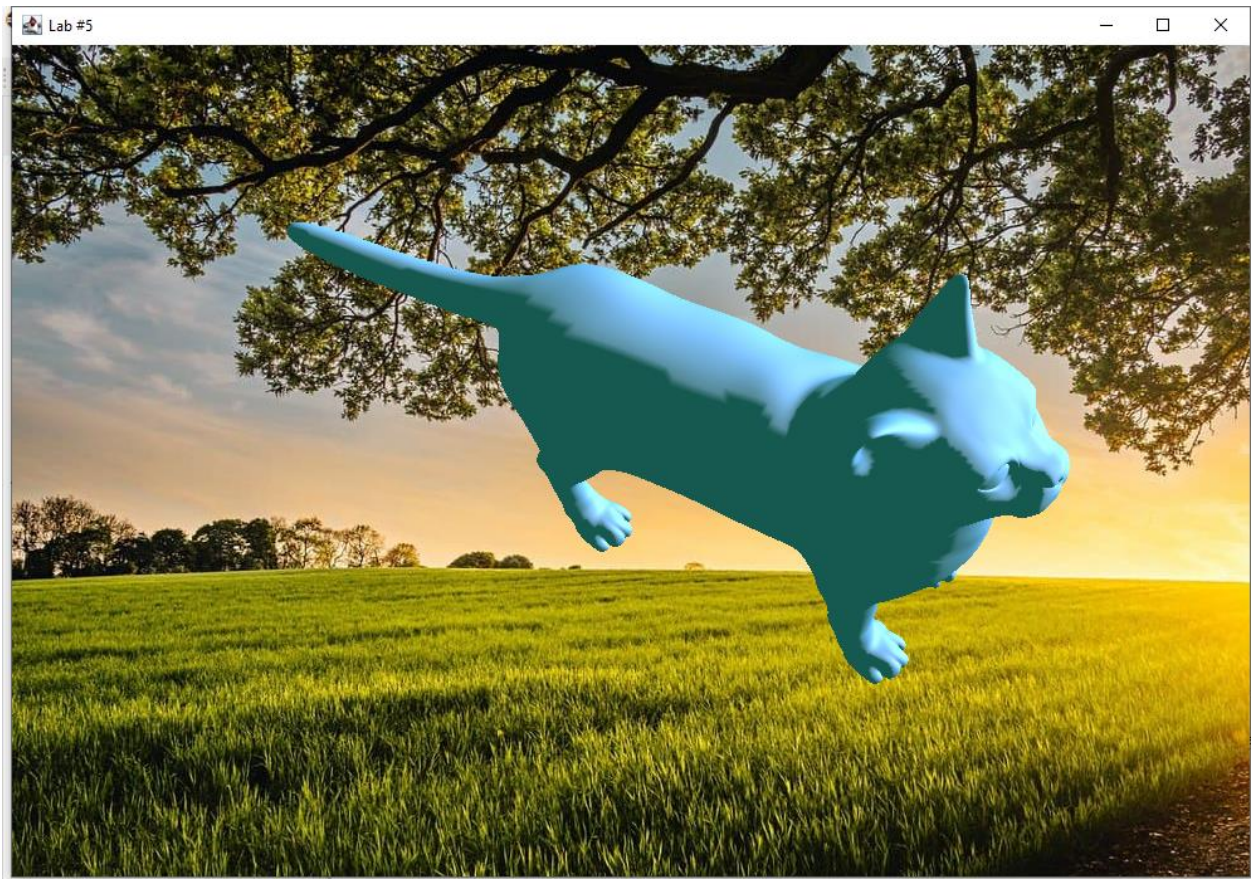
```
trexGroup.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        root.addChild(trexGroup);

    }

    //додамо картинку поля на фон
    private void addImageBackground() throws IOException {
        background = new
Background(getTextureLoader(backgroundLocation).getImage());
        background.setImageScaleMode(Background.SCALE_FIT_MAX);
        background.setApplicationBounds(new BoundingSphere(new
Point3d(),1000));
        background.setCapability(Background.ALLOW_IMAGE_WRITE);
        root.addChild(background);
    }

    private void setInitialViewAngle() {
        var vp = universe.getViewingPlatform();
        var transform = new Transform3D();
        transform.lookAt(
                new Point3d(-3.985f, -0.07f, 0),
                new Point3d(-0.7, 0, 0),
                new Vector3d(0, 1, 0)
        );
        transform.invert();
        vp.getViewPlatformTransform().setTransform(transform);
    }

    private Scene getSceneFromFile() throws IOException {
        ObjectFile file = new ObjectFile(ObjectFile.RESIZE);
        file.setFlags(ObjectFile.RESIZE | ObjectFile.TRIANGULATE |
ObjectFile.STRIPIFY);
        var inputStream =
classLoader.getResourceAsStream(trexModelLocation);
        if (inputStream == null) {
            throw new IOException("Resource " + trexModelLocation + "
not found");
```

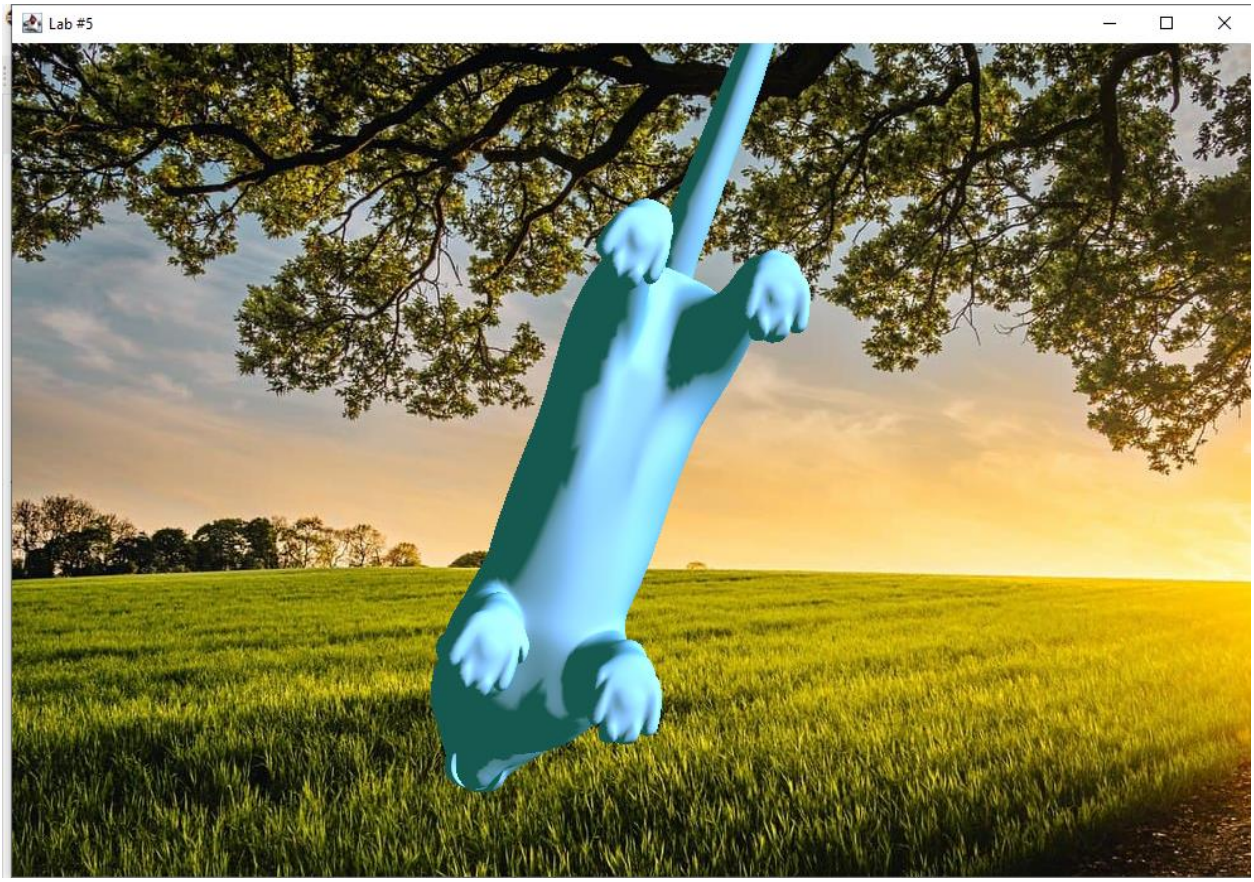
```
        }
        return file.load(new BufferedReader(new
InputStreamReader(inputStream)));
    }

    private void printModelElementsList(Map<String, Shape3D> map) {
      for (String name : map.keySet()) {
            System.out.println("Name: " + name);
      }
    }

    //налаштування анімації повертання моделі

    @Override
    public void keyPressed(KeyEvent e) {
        System.out.println(e.getKeyCode());
        int keyCode = e.getKeyCode();
        float diff = 0.05f;

        switch (keyCode) {
            case KeyEvent.VK_X: {
                rotateTransformX.rotX(diff);
                transform3D.mul(rotateTransformX);
                trexGroup.setTransform(transform3D);
            } break;
            case KeyEvent.VK_Y: {
                rotateTransformY.rotY(diff);
                transform3D.mul(rotateTransformY);
                trexGroup.setTransform(transform3D);
            } break;
            case KeyEvent.VK_Z: {
                rotateTransformZ.rotZ(diff);
                transform3D.mul(rotateTransformZ);
                trexGroup.setTransform(transform3D);
            } break;
        }
    }
```

```
    @Override
    public void keyReleased(KeyEvent e) { }

    @Override
    public void keyTyped(KeyEvent e) { }

    @Override
    public void actionPerformed(ActionEvent actionEvent) {}
}
```

Результат






Lab #5