

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“

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

|  |  |  |  |
| --- | --- | --- | --- |
| Виконала  студентка III курсу  групи КП-83  Дзьоба Уляна Олегівна  (*прізвище, ім’я, по батькові*)  варіант № 5 |  |  | Зарахована  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладачем  Шкурат Оксаною Сергіївною (*прізвище, ім’я, по батькові*) |

Київ 2021

**Варіант завдання**

**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.

**Варіант:**

Імпортувати модель паперового кораблика. Анімувати рух кораблика по весняній калюжі

**Лістинг коду програми**

**Main.java**

package lab5;

import com.sun.j3d.utils.universe.\*;

import java.awt.Color;

import javax.media.j3d.\*;

import javax.media.j3d.Material;

import javax.swing.\*;

import javax.vecmath.\*;

import javax.media.j3d.Background;

import com.sun.j3d.loaders.\*;

import com.sun.j3d.loaders.objectfile.ObjectFile;

import com.sun.j3d.utils.image.TextureLoader;

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.\*;

import java.net.URL;

import java.util.Map;

public class Main extends JFrame implements ActionListener, KeyListener {

private final static String boatModelLocation = "Paper\_Boat.obj";

private final static String backgroundLocation = "river.png";

private final BranchGroup root = new BranchGroup();

private final Canvas3D canvas = new Canvas3D(SimpleUniverse.getPreferredConfiguration());

private final TransformGroup boatGroup = new TransformGroup();

private final Transform3D transform3D = new Transform3D();

private final Transform3D rotateTransformY = new Transform3D();

private final Transform3D rotateTransformZ = new Transform3D();

private final ClassLoader classLoader = Thread.currentThread().getContextClassLoader();

private final Timer timer = new Timer(50, this);

private SimpleUniverse universe;

private Scene boatScene;

private Background background;

private Map<String, Shape3D> nameMap;

private float x\_location\_current = -3;

private float z\_location\_initial = -5;

private float z\_location\_current = -4;

float diff = 0.01f;

int swing = 0;

public static void main(String[] args) {

try {

Main window = new Main();

window.addKeyListener(window);

window.setVisible(true);

} catch (IOException e) {

System.err.println(e.getMessage());

}

}

public Main() throws IOException {

initialize();

addTexture();

addImageBackground();

addLight();

setInitialViewAngle();

setInitialLocation();

root.compile();

universe.addBranchGraph(root);

}

private void setInitialLocation() {

transform3D.setTranslation(new Vector3f(x\_location\_current, 0, 0));

boatGroup.setTransform(transform3D);

}

private void initialize() throws IOException {

// window settings

setTitle("Lab #5");

setSize(1000, 700);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

canvas.setDoubleBufferEnable(true);

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

universe = new SimpleUniverse(canvas);

universe.getViewingPlatform().setNominalViewingTransform();

timer.start();

canvas.addKeyListener(this);

boatScene = getSceneFromFile();

}

private void addLight() {

DirectionalLight 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);

AmbientLight ambientLight = new AmbientLight(new Color3f(Color.WHITE));

DirectionalLight directionalLight = new DirectionalLight(

new Color3f(Color.BLACK),

new Vector3f(-1F, -1F, -1F)

);

BoundingSphere 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 {

URL textureResource = classLoader.getResource(path);

if (textureResource == null) {

throw new IOException("Couldn't find texture: " + path);

}

return new TextureLoader(textureResource.getPath(), canvas);

}

private void addTexture() throws IOException {

Texture texture = getTextureLoader("paper3.jpg").getTexture();

texture.setBoundaryModeS(Texture.WRAP);

texture.setBoundaryModeT(Texture.WRAP);

TextureAttributes texAttr = new TextureAttributes();

texAttr.setTextureMode(TextureAttributes.MODULATE);

Appearance ap = new Appearance();

ap.setTexture(texture);

ap.setTextureAttributes(texAttr);

nameMap = boatScene.getNamedObjects();

System.out.println(nameMap.get("box001").getParent());

BranchGroup parent = (BranchGroup)(nameMap.get("box001").getParent());

parent.removeChild(nameMap.get("box001"));

nameMap.get("box001").setAppearance(ap);

boatGroup.addChild(nameMap.get("box001"));

boatGroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_WRITE);

root.addChild(boatGroup);

}

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() {

ViewingPlatform vp = universe.getViewingPlatform();

Transform3D transform = new Transform3D();

transform.lookAt(

new Point3d(-15f, 4f, 0f),

new Point3d(-1f, 0, 0),

new Vector3d(0f, 1, 0.1f)

);

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);

InputStream inputStream = classLoader.getResourceAsStream(boatModelLocation);

if (inputStream == null) {

throw new IOException("Resource " + boatModelLocation + " not found");

}

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

}

@Override

public void keyPressed(KeyEvent e) {

System.out.println(e.getKeyCode());

int keyCode = e.getKeyCode();

float diff = 0.05f;

switch (keyCode) {

case KeyEvent.VK\_LEFT: {

x\_location\_current += 0.05;

} break;

case KeyEvent.VK\_RIGHT: {

x\_location\_current -= 0.05;

} break;

case KeyEvent.VK\_A: {

rotateTransformY.rotY(diff);

transform3D.mul(rotateTransformY);

boatGroup.setTransform(transform3D);

} break;

case KeyEvent.VK\_S: {

rotateTransformY.rotY(-diff);

transform3D.mul(rotateTransformY);

boatGroup.setTransform(transform3D);

} break;

case KeyEvent.VK\_Z: {

transform3D.setTranslation(new Vector3f(0, 0, z\_location\_initial));

boatGroup.setTransform(transform3D);

z\_location\_current = z\_location\_initial;

} break;

}

}

@Override

public void keyReleased(KeyEvent e) { }

@Override

public void keyTyped(KeyEvent e) { }

@Override

public void actionPerformed(ActionEvent e) {

transform3D.setTranslation(new Vector3f(x\_location\_current, 0, z\_location\_current));

rotateTransformZ.rotZ(diff);

transform3D.mul(rotateTransformZ);

boatGroup.setTransform(transform3D);

z\_location\_current += 0.05;

swing++;

if(swing == 10){

diff \*= -1;

swing = 0;

}

}

}

**Результат**

****