



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”

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)

Варіант завдання

Тема: Анімація тривимірних об'єктів.

Завдання: Виконати анімацію тривимірної сцени за варіантом.

Варіант: Анімація гусака `goose.obj`. Гусак повинен рухати ногами, ходити по екрану, з поворотами.

Лістинг коду програми

Main.java

```
package main;

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.awt.Container;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Hashtable;
import java.util.Enumeration;

public class Main extends JFrame{
    public Canvas3D myCanvas3D;

    public Main() throws IOException {
        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("Goose");
        setSize(600,600);
        getContentPane().add("Center", myCanvas3D);
        setVisible(true);
    }

    public void createSceneGraph(SimpleUniverse su) throws
IOException {
        ObjectFile f = new ObjectFile(ObjectFile.RESIZE);
        BoundingSphere bs = new BoundingSphere(new
Point3d(0.0,0.0,0.0),Double.MAX_VALUE);
        String name;
        BranchGroup scratBranchGroup = new BranchGroup();
        Background scratBackground = new Background(new Color3f(-
1.0f,-1.0f,1.0f));

        ClassLoader classLoader =
Thread.currentThread().getContextClassLoader();
        var inputStream =
classLoader.getResourceAsStream("goosee.obj");
        Scene scratScene = f.load(new BufferedReader(new
InputStreamReader(inputStream)));

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

        //додавання текстури
        Appearance bodyAppearance = new Appearance();
        setToMyDefaultAppearance(bodyAppearance, new Color3f(0.3f,
0.3f, 0.3f));
    }

```

```
TextureLoader loader = new
TextureLoader("Lab6/src/textute.jpg", "LUMINANCE", new Container());
Texture texture1 = loader.getTexture();
texture1.setBoundaryModeS(Texture.WRAP);
texture1.setBoundaryModeT(Texture.WRAP);
texture1.setBoundaryColor(new Color4f(0.0f, 1.0f, 1.0f,
0.0f));

TextureAttributes texAttr = new TextureAttributes();
texAttr.setTextureMode(TextureAttributes.MODULATE);
bodyAppearance.setTexture(texture1);
bodyAppearance.setTextureAttributes(texAttr);

//додавання фону
var background = new
Background(getTextureLoader("field.jpg").getImage());
background.setImageScaleMode(Background.SCALE_FIT_MAX);
background.setApplicationBounds(new BoundingSphere(new
Point3d(),1000));
background.setCapability(Background.ALLOW_IMAGE_WRITE);

//анімація
Transform3D startTransformation = new Transform3D();
startTransformation.setScale(0.5);
Transform3D rotation1 = new Transform3D();
rotation1.rotX(-1.3f);
startTransformation.mul(rotation1);
Transform3D combinedStartTransformation = new Transform3D();
combinedStartTransformation.mul(startTransformation);

TransformGroup scratStartTransformGroup = new
TransformGroup(combinedStartTransformation);

int movesCount = 200;
int movesDuration = 600;
int startTime = 0;

// права лапа
Appearance legApp = new Appearance();
```

```

        setToMyDefaultAppearance(legApp, new Color3f(0.3f, 0.2f,
0.1f));

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

        Shape3D rightLeg = (Shape3D)
roachNamedObjects.get("rightleg");
        rightLeg.setAppearance(legApp);
        TransformGroup rightLegTG = new TransformGroup();
        rightLegTG.addChild(rightLeg.cloneTree());

        Transform3D rightLegRotAxis = new Transform3D();
        rightLegRotAxis.rotZ(Math.PI / 2);

        RotationInterpolator rightLegRot = new
RotationInterpolator(rightLegAlpha, rightLegTG, rightLegRotAxis,
0.0f, (float) Math.PI/8);
        rightLegRot.setSchedulingBounds(bs);

        rightLegTG.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        rightLegTG.addChild(rightLegRot);

        // лівá лапа
        Alpha leftLegAlpha = new Alpha(movesCount,
Alpha.INCREASING_ENABLE, startTime, 0, movesDuration,0,0,0,0,0);

        Shape3D leftLeg = (Shape3D) roachNamedObjects.get("leftleg");
        leftLeg.setAppearance(legApp);
        TransformGroup leftLegTG = new TransformGroup();
        leftLegTG.addChild(leftLeg.cloneTree());

        Transform3D leftLegRotAxis = new Transform3D();
        leftLegRotAxis.rotZ(Math.PI / 2);

        RotationInterpolator leftLegRot = new
RotationInterpolator(leftLegAlpha, leftLegTG, leftLegRotAxis,
0.0f, (float) -Math.PI/8);
        leftLegRot.setSchedulingBounds(bs);

```

```

leftLegTG.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
    leftLegTG.addChild(leftLegRot);

    // тіло
    TransformGroup sceneGroup = new TransformGroup();
    sceneGroup.addChild(rightLegTG);
    sceneGroup.addChild(leftLegTG);
    sceneGroup.addChild(background);

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

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

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

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

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

//налаштування освітлення
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.setTranslation(vector);
        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;
    }

```



```

    }

    private TextureLoader getTextureLoader(String path) throws
IOException {
        ClassLoader classLoader =
Thread.currentThread().getContextClassLoader();
        var textureResource = classLoader.getResource(path);
        if (textureResource == null) {
            throw new IOException("Couldn't find texture: " +
path);
        }
        return new TextureLoader(textureResource.getPath(),
myCanvas3D);
    }

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

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

```

Результат



