

МIНIСТЕРСТВО ОСВIТИ 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”

|  |  |
| --- | --- |
| Виконав  студент III курсу  групи КП-73  Булаєвський Ігор Олегович  (*прізвище, ім’я, по батькові*)  Варіант №3 |  |

Київ - 2020Лістинг програми

**Animation.java**

**import java.applet.Applet;**

**import java.awt.\*;**

**import java.awt.event.\*;**

**import java.util.Collections;**

**import javax.media.j3d.\*;**

**import javax.swing.\*;**

**import javax.swing.event.MouseInputListener;**

**import javax.vecmath.\*;**

**import com.sun.j3d.utils.applet.MainFrame;**

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

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

**public class Animation extends Applet implements ActionListener, MouseInputListener, MouseWheelListener {**

**private Point2d mouseLastPosition = null;**

**private double angleX = 0;**

**private double angleY = 0;**

**private double scale = 0.5;**

**Transform3D trans3d;**

**TransformGroup transGroup, bugGroup = new TransformGroup(), bugRotate = new TransformGroup();**

**double bugAngle = Math.PI / 4;**

**BranchGroup table;**

**Point2d bugPoisition = new Point2d(0f, 0f);**

**Vector2d bugVelocity = new Vector2d(0.01f, 0.01f);**

**double step = 0;**

**public BranchGroup createSceneGraph() {**

**BranchGroup group = new BranchGroup();**

**transGroup = new TransformGroup();**

**transGroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_WRITE);**

**trans3d = new Transform3D();**

**transGroup.setTransform(trans3d);**

**group.addChild(transGroup);**

**BoundingSphere bound = new BoundingSphere(new Point3d(0.0, 0.0, 0.0), 100.0);**

**Color3f bgColor = new Color3f(0.05f, 0.05f, 0.2f);**

**TextureLoader t = new TextureLoader("textures/background.jpg", this);**

**Background bg = new Background(t.getImage());**

**bg.setImageScaleMode(Background.SCALE\_FIT\_ALL);**

**bg.setApplicationBounds(bound);**

**group.addChild(bg);**

**Color3f lightColor = new Color3f(1.0f, 1.0f, 0.9f);**

**Vector3f lightDirection = new Vector3f(4.0f, -7.0f, -12.0f);**

**DirectionalLight light = new DirectionalLight(lightColor, lightDirection);**

**light.setInfluencingBounds(bound);**

**group.addChild(light);**

**TransformGroup objTrans = new TransformGroup();**

**objTrans.setCapability(TransformGroup.ALLOW\_TRANSFORM\_WRITE);**

**Appearance tableApp = loadTexture("textures/wood.jpg");**

**table = new ObjFileReader("models/table.obj", "table", tableApp);**

**BranchGroup g = (BranchGroup) table.getChild(0);**

**objTrans.addChild(table);**

**bugGroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_WRITE);**

**bugRotate.setCapability(TransformGroup.ALLOW\_TRANSFORM\_WRITE);**

**Appearance bugApp = loadTexture("textures/bug.jpg");**

**BranchGroup bug = new ObjFileReader("models/ROACH.OBJ", "roach\_body", bugApp);**

**TransformGroup bugGroupScaleGroup = new TransformGroup();**

**bugGroupScaleGroup.addChild(bug);**

**scaleObject(bugGroupScaleGroup, 0.2);**

**bugRotate.addChild(bugGroupScaleGroup);**

**moveObject(bugGroup, 0.f, 0.5f, 0f);**

**bugGroup.addChild(bugRotate);**

**objTrans.addChild(bugGroup);**

**transGroup.addChild(objTrans);**

**group.compile();**

**return group;**

**}**

**void moveObject(TransformGroup o, float x, float y, float z) {**

**Transform3D move = new Transform3D();**

**move.setTranslation(new Vector3d(x, y, z));**

**o.setTransform(move);**

**}**

**void scaleObject(TransformGroup o, double scale) {**

**Transform3D scaleGroup = new Transform3D();**

**scaleGroup.setScale(scale);**

**o.setTransform(scaleGroup);**

**}**

**public Animation() {**

**setLayout(new BorderLayout());**

**GraphicsConfiguration config = SimpleUniverse.getPreferredConfiguration();**

**Canvas3D canvas = new Canvas3D(config);**

**add("Center", canvas);**

**BranchGroup scene = createSceneGraph();**

**SimpleUniverse universe = new SimpleUniverse(canvas);**

**universe.getViewingPlatform().setNominalViewingTransform();**

**universe.addBranchGraph(scene);**

**canvas.addMouseMotionListener(this);**

**canvas.addMouseListener(this);**

**canvas.addMouseWheelListener(this);**

**Timer timer = new Timer(10, this);**

**timer.start();**

**}**

**public static void main(String[] args) {**

**new MainFrame(new Animation(), 800, 600);**

**}**

**Point2d tableBounds = new Point2d(0.9f, 0.4f);**

**Appearance loadTexture(String path) {**

**Texture tex = new TextureLoader(path, this).getTexture();**

**tex.setBoundaryModeS(Texture.WRAP);**

**tex.setBoundaryModeT(Texture.WRAP);**

**TextureAttributes texAttr = new TextureAttributes();**

**texAttr.setTextureMode(TextureAttributes.COMBINE);**

**Appearance ap = new Appearance();**

**ap.setTexture(tex);**

**ap.setTextureAttributes(texAttr);**

**Material material = new Material();**

**material.setSpecularColor(new Color3f(Color.WHITE));**

**material.setDiffuseColor(new Color3f(Color.WHITE));**

**ap.setMaterial(material);**

**return ap;**

**}**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**Transform3D temp = new Transform3D();**

**temp.rotX(angleY);**

**trans3d.rotY(angleX);**

**trans3d.mul(temp);**

**trans3d.setScale(scale);**

**transGroup.setTransform(trans3d);**

**step += 0.02;**

**double scaleFactor = (Math.cos(step) + 2) / 3;**

**bugPoisition.x += bugVelocity.x \* scaleFactor;**

**bugPoisition.y += bugVelocity.y \* scaleFactor;**

**if (bugPoisition.y > tableBounds.y) {**

**bugVelocity.y = -bugVelocity.y;**

**bugAngle += Math.PI / 2 \* (bugVelocity.x > 0 ? 1 : -1);**

**} else if (bugPoisition.y < -tableBounds.y) {**

**bugVelocity.y = -bugVelocity.y;**

**bugAngle += 3 \* Math.PI / 2 \* (bugVelocity.x > 0 ? 1 : -1);**

**}**

**if (bugPoisition.x > tableBounds.x) {**

**bugVelocity.x = -bugVelocity.x;**

**bugAngle += Math.PI / 2 \* (bugVelocity.y < 0 ? 1 : -1);**

**} else if (bugPoisition.x < -tableBounds.x) {**

**bugVelocity.x = -bugVelocity.x;**

**bugAngle += 3 \* Math.PI / 2 \* (bugVelocity.y < 0 ? 1 : -1);**

**}**

**rotateBug();**

**moveObject(bugGroup, (float) bugPoisition.x, 0.5f, (float) bugPoisition.y);**

**}**

**void rotateBug() {**

**Transform3D rotate = new Transform3D();**

**rotate.rotY(bugAngle);**

**bugRotate.setTransform(rotate);**

**}**

**@Override**

**public void mouseClicked(MouseEvent e) {**

**}**

**@Override**

**public void mousePressed(MouseEvent e) {**

**}**

**@Override**

**public void mouseReleased(MouseEvent e) {**

**mouseLastPosition = null;**

**}**

**@Override**

**public void mouseEntered(MouseEvent e) {**

**}**

**@Override**

**public void mouseExited(MouseEvent e) {**

**}**

**@Override**

**public void mouseDragged(MouseEvent e) {**

**Point2d mouseCurrentPosition = new Point2d(e.getX(), e.getY());**

**if (mouseLastPosition != null) {**

**double dx = mouseCurrentPosition.x - mouseLastPosition.x;**

**double dy = mouseCurrentPosition.y - mouseLastPosition.y;**

**angleX += dx / 150;**

**angleY += dy / 150;**

**}**

**mouseLastPosition = mouseCurrentPosition;**

**}**

**@Override**

**public void mouseMoved(MouseEvent e) {**

**}**

**@Override**

**public void mouseWheelMoved(MouseWheelEvent e) {**

**scale \*= 1 - e.getPreciseWheelRotation() / 10;**

**}**

**}**

Скріншоти результатів

