НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ І. СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

ЛАБОРАТОРНА РОБОТА №4

з дисципліни

"МАТЕМАТИЧНІ ТА АЛГОРИТМІЧНІ ОСНОВИ КОМП’ЮТЕРНОЇ ГРАФІКИ"

ТЕМА: «Побудова найпростіших тривимірних об'єктів за допомогою бібліотеки Java3D та їх анімація»

Підготував:

студент групи КП-71

Пилипенко І.О.

Київ-2020

**Код**

package sample;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import com.sun.j3d.utils.geometry.\*;  
import com.sun.j3d.utils.universe.SimpleUniverse;  
import javax.media.j3d.\*;  
import javax.swing.Timer;  
import javax.vecmath.\*;  
  
public class ComplexObject implements ActionListener {  
 private TransformGroup treeTransformGroup;  
 private Transform3D treeTransform3D = new Transform3D();  
 private Timer timer;  
 private float angle = 0;  
 public static void main(String[] args) {  
 new ComplexObject();  
 }  
 public ComplexObject() {  
 timer = new Timer(50, this);  
 timer.start();  
 BranchGroup scene = createSceneGraph();  
 SimpleUniverse u = new SimpleUniverse();  
 u.getViewingPlatform().setNominalViewingTransform();  
 u.addBranchGraph(scene);  
 }  
 public BranchGroup createSceneGraph() {  
 BranchGroup objRoot = new BranchGroup();  
 treeTransformGroup = new TransformGroup();  
 treeTransformGroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_WRITE);  
 buildObj();  
 objRoot.addChild(treeTransformGroup);  
  
 BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0),100.0);  
 Color3f light1Color = new Color3f(1.0f, 0.5f, 0.4f);  
 Vector3f light1Direction = new Vector3f(4.0f, -7.0f, -12.0f);  
 DirectionalLight light1 = new DirectionalLight(light1Color,  
 light1Direction);  
 light1.setInfluencingBounds(bounds);  
 objRoot.addChild(light1);  
  
 Color3f ambientColor = new Color3f(1.0f, 1.0f, 1.0f);  
 AmbientLight ambientLightNode = new AmbientLight(ambientColor);  
 ambientLightNode.setInfluencingBounds(bounds);  
 objRoot.addChild(ambientLightNode);  
 return objRoot;  
 }  
 private void buildObj() {  
  
 createBall(0.2f, .0f, -0.05f, .0f, "", new Color3f(1.0f, 0.5f, 0.0f));  
 TransformGroup cone1 = new TransformGroup();  
 Transform3D transformTop = new Transform3D();  
 Cone coneTop = XMassCone.getCone(-0.25f, 0.1f);  
 Vector3f vectorTop = new Vector3f(.0f, -0.2f, .0f);  
 transformTop.setTranslation(vectorTop);  
 cone1.setTransform(transformTop);  
 cone1.addChild(coneTop);  
  
 treeTransformGroup.addChild(cone1);  
  
 TransformGroup cone2 = new TransformGroup();  
 Transform3D transformTop2 = new Transform3D();  
 Cone coneTop2 = XMassCone.getCone(-0.20f, 0.19f);  
 Vector3f vectorTop2 = new Vector3f(.0f, -0.2f, .0f);  
 transformTop2.setTranslation(vectorTop2);  
 cone2.setTransform(transformTop2);  
 cone2.addChild(coneTop2);  
 treeTransformGroup.addChild(cone2);  
  
 TransformGroup tg = new TransformGroup();  
 Transform3D transform = new Transform3D();  
 Cylinder cylinder = new Cylinder(0.01f, 0.09f);  
 Vector3f vector = new Vector3f(.0f, .2f, .0f);  
 transform.setTranslation(vector);  
 tg.setTransform(transform);  
 tg.addChild(cylinder);  
 treeTransformGroup.addChild(tg);  
  
 createBall(0.18f, .0f, 0.03f, .0f, "./dub.jpg", new Color3f(0.5f, 0.0f, 0.0f));  
  
 }  
 private void createBall(float radius, float x, float y, float z, String picture,  
 Color3f emissive) {  
 TransformGroup tg = new TransformGroup();  
 Transform3D transform = new Transform3D();  
 Sphere cone = XMassBall.getSphere(radius, picture, emissive);  
 Vector3f vector = new Vector3f(x, y, z);  
 transform.setTranslation(vector);  
 tg.setTransform(transform);  
 tg.addChild(cone);  
 treeTransformGroup.addChild(tg);  
 }  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 treeTransform3D.rotY(angle);  
 treeTransformGroup.setTransform(treeTransform3D);  
 angle += 0.05;  
 }  
}

package sample;  
  
import java.awt.Container;  
import javax.media.j3d.\*;  
import javax.vecmath.Color3f;  
import javax.vecmath.Color4f;  
import com.sun.j3d.utils.geometry.\*;  
import com.sun.j3d.utils.image.TextureLoader;  
import com.sun.javafx.geom.Shape;  
import javax.vecmath.\*;  
  
public class XMassBall {  
 public static Sphere getSphere(float radius, String picture, Color3f emissiveColor) {  
 int primflags = Primitive.*GENERATE\_NORMALS* + Primitive.*GENERATE\_TEXTURE\_COORDS*;  
 return new Sphere(radius, primflags, *getXMassBallsAppearence*(picture,  
 emissiveColor));  
 }  
  
 public static Sphere getEllipsoid(float radius, String picture, Color3f emissiveColor) {  
 int primflags = Primitive.*GENERATE\_NORMALS*;  
 return new Sphere(radius, primflags, *getXMassBallsAppearence*(picture,  
 emissiveColor));  
 }  
  
  
 private static Appearance getXMassBallsAppearence(String picture, Color3f emissive) {  
 Appearance ap = new Appearance();  
 Color3f ambient = new Color3f(0.2f, 0.15f, .15f);  
 Color3f diffuse = new Color3f(1.2f, 1.15f, .15f);  
 Color3f specular = new Color3f(0.0f, 0.0f, 0.0f);  
 ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));  
 if (picture != "") {  
  
 TextureLoader loader = new TextureLoader(picture, "LUMINANCE", new  
 Container());  
 Texture texture = loader.getTexture();  
  
 texture.setBoundaryModeS(Texture.*WRAP*);  
 texture.setBoundaryModeT(Texture.*WRAP*);  
 texture.setBoundaryColor(new Color4f(0.0f, 1.0f, 1.0f, 0.0f));  
 TextureAttributes texAttr = new TextureAttributes();  
 texAttr.setTextureMode(TextureAttributes.*MODULATE*);  
 ap.setTexture(texture);  
 ap.setTextureAttributes(texAttr);  
 }  
 return ap;  
 }  
}

package sample;  
  
import javax.media.j3d.Appearance;  
import javax.media.j3d.ColoringAttributes;  
import javax.media.j3d.Material;  
import javax.vecmath.Color3f;  
import com.sun.j3d.utils.geometry.Cone;  
import com.sun.j3d.utils.geometry.Primitive;  
  
public class XMassCone {  
 public static Cone getCone(float height, float radius) {  
 int primflags = Primitive.*GENERATE\_NORMALS* + Primitive.*GENERATE\_TEXTURE\_COORDS*;  
 return new Cone(radius, height, primflags, *getXMassTreeAppearence*());  
 }  
 private static Appearance getXMassTreeAppearence() {  
 Appearance ap = new Appearance();  
 Color3f emissive = new Color3f(1.0f, 0.5f, 0.0f);  
 Color3f ambient = new Color3f(0.2f, 0.15f, .15f);  
 Color3f diffuse = new Color3f(1.2f, 1.15f, .15f);  
 Color3f specular = new Color3f(0.0f, 0.0f, 0.0f);  
 ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));  
 return ap;  
 }  
}

**Малюнки та анімації**

****

Рис1. Малюнок