

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 №** **4**

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

|  |  |  |  |
| --- | --- | --- | --- |
| Виконав  студент III курсу  групи КП-82  Мельничук Олексій Геннадійович  (*прізвище, ім’я, по батькові*)  варіант № 12 |  |  | Зарахована  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладачем  Шкурат Оксаною Сергіївною (*прізвище, ім’я, по батькові*) |

Київ 2021

**Тема**: «Побудова найпростіших тривимірних об'єктів за допомогою бібліотеки Java3D та їх анімація

**Мета**:

1. вивчення стандартних засобів Java3D для візуалізація зображення;
2. вивчення засобів анімації примітивів та складених об'єктів в Java3D.

**Завдання**

За допомогою засобів, що надає бібліотека Java3D, побудувати тривимірний об’єкт. Для цього скористатися основними примітивами, що буде доцільно використовувати згідно варіанту: сфера, конус, паралелепіпед, циліндр. Об'єкт має складатися з 5-15 примітивів. Задати матеріал кожного примітиву, в разі необхідності накласти текстуру. В сцені має бути мінімум одне джерело освітлення.

Виконати анімацію сцени таким чином, щоб можна було розглянути об'єкт з усіх сторін. За бажанням можна виконати інтерактивні взаємодію з об'єктом за допомогою миші та клавіатури.

**Варіант: 12**

12. Мухомор

**Код програми**

|  |
| --- |
| **Amanita.java** |
| package lab4;  import com.sun.j3d.utils.applet.MainFrame;  import com.sun.j3d.utils.geometry.\*;  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.applet.Applet;  import java.awt.\*;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import javax.media.j3d.AmbientLight;  import javax.media.j3d.Appearance;  import javax.media.j3d.BoundingSphere;  import javax.media.j3d.BranchGroup;  import javax.media.j3d.DirectionalLight;  import javax.media.j3d.Material;  import javax.media.j3d.Texture;  import javax.media.j3d.TextureAttributes;  import javax.vecmath.Color3f;  import javax.vecmath.Color4f;  import javax.vecmath.Point3d;  import javax.vecmath.Vector3f;  import com.sun.j3d.utils.geometry.Primitive;  import com.sun.j3d.utils.geometry.Sphere;  import com.sun.j3d.utils.image.TextureLoader;  import com.sun.j3d.utils.universe.SimpleUniverse;  import com.sun.j3d.utils.geometry.Box;  @SuppressWarnings("deprecation")  public class Amanita extends Applet implements ActionListener {  private final TransformGroup amanitaTransformGroup = new TransformGroup();  private final Transform3D amanitaTransform3d = new Transform3D();  private final Timer timer = new Timer(50, this);    private float angle = 0;  private boolean rotateY = true;    public static String fullPath = "C:/Users/datru/Desktop/study2021/maokg/lab4/assets/";  public static void main(String[] args) {    MainFrame mf = new MainFrame(new Amanita(), 640, 480);  mf.run();  }  private Amanita() {  setLayout(new BorderLayout());  GraphicsConfiguration config = SimpleUniverse.getPreferredConfiguration();  Canvas3D c = new Canvas3D(config);  add("Center", c);  SimpleUniverse universe = new SimpleUniverse(c);  timer.start();  universe.getViewingPlatform().setNominalViewingTransform();  universe.addBranchGraph(createSceneGraph());  }      private BranchGroup createSceneGraph() {  BranchGroup root = new BranchGroup();  amanitaTransformGroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_WRITE);    buildAmanita();    root.addChild(amanitaTransformGroup);    TextureLoader loader = new TextureLoader(fullPath + "shroom\_forest.jpg", new Container());  ImageComponent2D texture = loader.getImage();  Background background = new Background(texture);  background.setImageScaleMode(Background.SCALE\_FIT\_ALL);  // background.setImageScaleMode(Background.SCALE\_FIT\_ALL);  background.setCapability(Background.ALLOW\_IMAGE\_WRITE);    BoundingSphere sphere = new BoundingSphere(new Point3d(0,0,0), 1);  background.setApplicationBounds(sphere);  root.addChild(background);    Color3f light1Color = new Color3f(1f, 1f, 1f);  BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0),  100.0);  Vector3f light1Direction = new Vector3f(4.0f, -7.0f, -12.0f);  DirectionalLight light1 = new DirectionalLight(light1Color, light1Direction);  light1.setInfluencingBounds(bounds);  root.addChild(light1);  AmbientLight ambientLight = new AmbientLight(new Color3f(.5f, .5f, .5f));  ambientLight.setInfluencingBounds(bounds);  root.addChild(ambientLight);  return root;  }    private void buildAmanita2() {    TransformGroup amanitaHatGroup1 = new TransformGroup();  TransformGroup amanitaHatGroup2 = new TransformGroup();  TransformGroup amanitaHatGroup3 = new TransformGroup();  TransformGroup amanitaHatGroup4 = new TransformGroup();  TransformGroup amanitaHatGroup5 = new TransformGroup();  Transform3D transformHat1 = new Transform3D();  Transform3D transformHat2 = new Transform3D();  Transform3D transformHat3 = new Transform3D();  Transform3D transformHat4 = new Transform3D();  Transform3D transformHat5 = new Transform3D();  //width height length  Box hatSideLeft = AmanitaParts.getBox(.03f\* 1.5f, .06f\* 1.5f, .1f\* 1.5f, true);  Box hatSideRight = AmanitaParts.getBox(.03f\* 1.5f, .06f\* 1.5f, .1f\* 1.5f, true);  Box hatSideNear = AmanitaParts.getBox(.1f\* 1.5f, .06f\* 1.5f, .03f\* 1.5f, true);  Box hatSideFar = AmanitaParts.getBox(.1f\* 1.5f, .06f\* 1.5f, .03f\* 1.5f, true);    Box hatTop = AmanitaParts.getBox(.1f\* 1.5f, .03f\* 1.5f, .1f\* 1.5f, true);    //left-right top-bottom near-far  transformHat1.setTranslation(new Vector3f( .125f\* 1.5f, .22f\* 1.5f, .0f));  transformHat2.setTranslation(new Vector3f(-.125f\* 1.5f, .22f\* 1.5f, .0f));  transformHat3.setTranslation(new Vector3f( .0f, .22f\* 1.5f, .125f\* 1.5f));  transformHat4.setTranslation(new Vector3f( .0f, .22f\* 1.5f, -.125f\* 1.5f));  transformHat5.setTranslation(new Vector3f( .0f, .31f\* 1.5f, .0f));  amanitaHatGroup1.setTransform(transformHat1);  amanitaHatGroup1.addChild(hatSideLeft);  amanitaTransformGroup.addChild(amanitaHatGroup1);    amanitaHatGroup2.setTransform(transformHat2);  amanitaHatGroup2.addChild(hatSideRight);  amanitaTransformGroup.addChild(amanitaHatGroup2);    amanitaHatGroup3.setTransform(transformHat3);  amanitaHatGroup3.addChild(hatSideNear);  amanitaTransformGroup.addChild(amanitaHatGroup3);    amanitaHatGroup4.setTransform(transformHat4);  amanitaHatGroup4.addChild(hatSideFar);  amanitaTransformGroup.addChild(amanitaHatGroup4);    amanitaHatGroup5.setTransform(transformHat5);  amanitaHatGroup5.addChild(hatTop);  amanitaTransformGroup.addChild(amanitaHatGroup5);            TransformGroup amanitaLegGroup1 = new TransformGroup();  TransformGroup amanitaLegGroup2 = new TransformGroup();  TransformGroup amanitaLegGroup3 = new TransformGroup();  Transform3D transformLeg1 = new Transform3D();  Transform3D transformLeg2 = new Transform3D();  Transform3D transformLeg3 = new Transform3D();  //width height length  Box legStem = AmanitaParts.getBox(.04f \* 1.5f, .32f\* 1.5f, .04f\* 1.5f, false);  Box legSkirt = AmanitaParts.getBox(.06f\* 1.5f, .02f\* 1.5f, .06f\* 1.5f, false);  Box legRoot = AmanitaParts.getBox(.06f\* 1.5f, .08f\* 1.5f, .06f\* 1.5f, false);    //left-right top-bottom near-far  transformLeg1.setTranslation(new Vector3f( .0f, .0f, .0f));  transformLeg2.setTranslation(new Vector3f( .0f, .12f\* 1.5f, .0f));  transformLeg3.setTranslation(new Vector3f( .0f, -.26f\* 1.5f, .0f));      amanitaLegGroup1.setTransform(transformLeg1);  amanitaLegGroup1.addChild(legStem);  amanitaTransformGroup.addChild(amanitaLegGroup1);    amanitaLegGroup2.setTransform(transformLeg2);  amanitaLegGroup2.addChild(legSkirt);  amanitaTransformGroup.addChild(amanitaLegGroup2);    amanitaLegGroup3.setTransform(transformLeg3);  amanitaLegGroup3.addChild(legRoot);  amanitaTransformGroup.addChild(amanitaLegGroup3);    }      private void buildAmanita() {  Transform3D transformRotate = new Transform3D();    TransformGroup amanitaHatGroup1 = new TransformGroup();  TransformGroup amanitaHatGroup2 = new TransformGroup();  Transform3D transformHat1 = new Transform3D();  Transform3D transformHat2 = new Transform3D();  Sphere hat = AmanitaParts.getSphere(0.3f, true);  transformHat1.setTranslation(new Vector3f(.0f, 0.5f, .0f));  amanitaHatGroup1.setTransform(transformHat1);  amanitaHatGroup1.addChild(hat);    Cylinder hat2 = AmanitaParts.getCylinder(0.3f, 0.2f, true);  transformHat2.setTranslation(new Vector3f(.0f, 0.35f, .0f));  amanitaHatGroup2.setTransform(transformHat2);  amanitaHatGroup2.addChild(hat2);    amanitaTransformGroup.addChild(amanitaHatGroup1);  amanitaTransformGroup.addChild(amanitaHatGroup2);    TransformGroup amanitaLegGroup1 = new TransformGroup();  TransformGroup amanitaLegGroup2 = new TransformGroup();  TransformGroup amanitaLegGroup3 = new TransformGroup();  Transform3D transformLeg1 = new Transform3D();  Transform3D transformLeg2 = new Transform3D();  Transform3D transformLeg3 = new Transform3D();    Cylinder leg1 = AmanitaParts.getCylinder(0.1f, 0.7f, false);  transformLeg1.setTranslation(new Vector3f(.0f, -.1f, .0f));  amanitaLegGroup1.setTransform(transformLeg1);  amanitaLegGroup1.addChild(leg1);    Cone leg2 = AmanitaParts.getCone(0.4f, 0.2f, false);  transformLeg2.setTranslation(new Vector3f(.0f, .25f, .0f));  amanitaLegGroup2.setTransform(transformLeg2);  amanitaLegGroup2.addChild(leg2);    Sphere leg3 = AmanitaParts.getSphere(0.1f, false);  transformLeg3.setTranslation(new Vector3f(.0f, -0.45f, .0f));  amanitaLegGroup3.setTransform(transformLeg3);  amanitaLegGroup3.addChild(leg3);    amanitaTransformGroup.addChild(amanitaLegGroup1);  amanitaTransformGroup.addChild(amanitaLegGroup2);  amanitaTransformGroup.addChild(amanitaLegGroup3);  }  @Override  public void actionPerformed(ActionEvent e) {  amanitaTransform3d.rotY(angle);  angle += 0.05;  if (angle >= 25) {  rotateY = !rotateY;  angle = 0;  }  amanitaTransformGroup.setTransform(amanitaTransform3d);  }  } |

# 

|  |
| --- |
| **AmanitaParts.java** |
| package lab4;  import com.sun.j3d.utils.geometry.\*;  import com.sun.j3d.utils.image.TextureLoader;  import javax.media.j3d.\*;  import javax.vecmath.\*;  import java.awt.\*;  public class AmanitaParts  {  public static String fullPath = "C:/Users/datru/Desktop/study2021/maokg/lab4/assets/";    public static int getPrimFlags()  {  return Primitive.GENERATE\_NORMALS + Primitive.GENERATE\_TEXTURE\_COORDS;  }    public static Cone getCone(float height, float radius, boolean top)  {  int primflags = getPrimFlags();  return top ? new Cone(radius, height, primflags, getAppearence("top1.jpg"))  : new Cone(radius, height, primflags, getAppearence("leg.jpg"));  }    public static Cylinder getCylinder(float radius, float height, boolean top)  {  int primflags = getPrimFlags();  return top ? new Cylinder(radius, height, primflags, getAppearence("top1.jpg"))  : new Cylinder(radius, height, primflags, getAppearence("leg.jpg"));  }  public static Sphere getSphere(float radius, boolean top)  {  int primflags = getPrimFlags();  return top ? new Sphere(radius, primflags, getAppearence("top1.jpg"))  : new Sphere(radius, primflags, getAppearence("leg.jpg"));  }    public static Box getBox(float l, float w, float h, boolean top)  {  int primflags = getPrimFlags();  return top ? new Box(l,w,h, primflags, getAppearence("mushroom\_hat\_hq.jpg"))  : new Box(l,w,h, primflags, getAppearence("mushroom\_stem\_hq.jpg"));  }    public static Appearance getAppearence(String textureName) {  TextureLoader loader = new TextureLoader(fullPath + textureName, 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);  Appearance ap = new Appearance();  ap.setTexture(texture);  ap.setTextureAttributes(texAttr);    Color3f emissive = new Color3f(new Color(255, 255, 255));  Color3f ambient = new Color3f(new Color(0, 0, 0));  Color3f diffuse = new Color3f(new Color(0, 0, 0));  Color3f specular = new Color3f(0.0f, 0.0f, 0.0f);    ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));  return ap;  }  } |

# **Результати роботи програми**

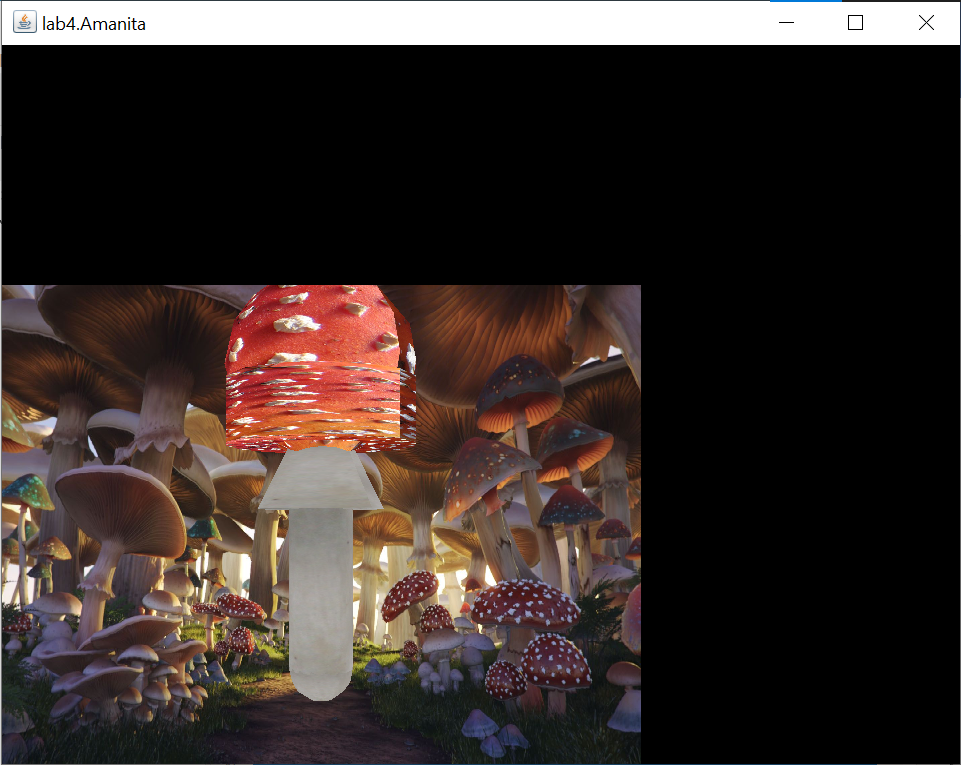


Рис.1. Мухомор з примітивів

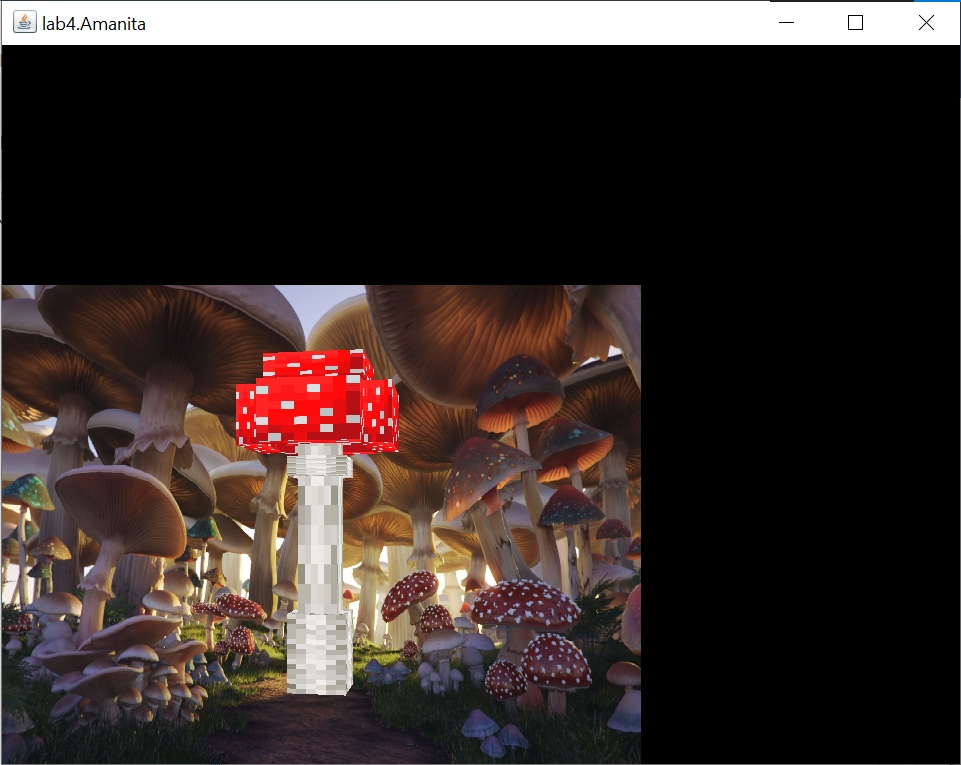


Рис.2. Стилізований мухомор

**Висновки**

Виконавши дану лабораторну роботу я опрацював стандартні засоби Java3D для візуалізація зображення та примінив навички для анімації примітивів та складених об'єктів в Java3D.