

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-02  Товстига Максим Анатолійович (*прізвище, ім’я, по батькові*)  варіант № 17 | Зарахована “ ” “ ” 20 р.  викладачем  Шкурат Оксаною Сергіївною  (*прізвище, ім’я, по батькові*) |

Київ 2023

# Варіант завдання

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

**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:** Годинник

# Лістинг коду програми

|  |
| --- |
| **Lab4.java** |
| package lab4;  import javax.swing.\*;  import javax.vecmath.\*;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import javax.media.j3d.\*;  import com.sun.j3d.utils.geometry.\*;  import com.sun.j3d.utils.geometry.Box;  import com.sun.j3d.utils.universe.\*;  public class lab4 implements ActionListener  {      private TransformGroup phoneTransformGroup;      private Transform3D phoneTransform3D = new Transform3D();      private Timer timer;      private float angle = 0;        public static void main(String[] args) {          new lab4();      }        public lab4() {          timer = new Timer(50, this);          timer.start();          BranchGroup scene = createSceneGraph();          SimpleUniverse u = new SimpleUniverse();          u.getViewingPlatform().setNominalViewingTransform();          u.addBranchGraph(scene);          }      private BranchGroup createSceneGraph() {          BranchGroup objRoot = new BranchGroup();          phoneTransformGroup = new TransformGroup();          phoneTransformGroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_WRITE);          buildPhone();          objRoot.addChild(phoneTransformGroup);            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 buildPhone() {          TransformGroup tgFrame = new TransformGroup();          Transform3D transformFrame = new Transform3D();          Box boxFrame = FramePhone.getBox(0.22f, 0.4f, 0.03f);          Vector3f vectorFrame = new Vector3f(.0f, .0f, .0f);          transformFrame.setTranslation(vectorFrame);          tgFrame.setTransform(transformFrame);          tgFrame.addChild(boxFrame);          phoneTransformGroup.addChild(tgFrame);            TransformGroup tgScreen = new TransformGroup();          Transform3D transformScreen = new Transform3D();          Box boxScreen = ScreenPhone.getBox(0.17f, 0.2f, 0.001f, "");          Vector3f vectorScreen = new Vector3f(.0f, 0.18f, 0.03f);          transformScreen.setTranslation(vectorScreen);          tgScreen.setTransform(transformScreen);          tgScreen.addChild(boxScreen);          phoneTransformGroup.addChild(tgScreen);            createButton(0.15f, -0.11f, 0.02f, 0.05f, 0.05f, new Color3f(.0f, .0f, .0f));          createButton(.0f, -0.11f, 0.02f, 0.05f, 0.05f, new Color3f(.0f, .0f, .0f));          createButton(-0.15f, -0.11f, 0.02f, 0.05f, 0.05f, new Color3f(.0f, .0f, .0f));            createButton(0.15f, -0.19f, 0.02f, 0.05f, 0.05f, new Color3f(.0f, .0f, .0f));          createButton(.0f, -0.19f, 0.02f, 0.05f, 0.05f, new Color3f(.0f, .0f, .0f));          createButton(-0.15f, -0.19f, 0.02f, 0.05f, 0.05f, new Color3f(.0f, .0f, .0f));            createButton(0.15f, -0.27f, 0.02f, 0.05f, 0.05f, new Color3f(.0f, .0f, .0f));          createButton(.0f, -0.27f, 0.02f, 0.05f, 0.05f, new Color3f(.0f, .0f, .0f));          createButton(-0.15f, -0.27f, 0.02f, 0.05f, 0.05f, new Color3f(.0f, .0f, .0f));            createButton(0.15f, -0.35f, 0.02f, 0.05f, 0.05f, new Color3f(.0f, .0f, .0f));          createButton(.0f, -0.35f, 0.02f, 0.05f, 0.05f, new Color3f(.0f, .0f, .0f));          createButton(-0.15f, -0.35f, 0.02f, 0.05f, 0.05f, new Color3f(.0f, .0f, .0f));        }      private void createButton(float x, float y, float z, float radius, float height, Color3f emissiveColor) {          TransformGroup tg = new TransformGroup();          Transform3D transform = new Transform3D();          Cylinder cylinder = ButtonPhone.getCylinder(radius, height, emissiveColor);          Vector3f vector = new Vector3f(x, y, z);          transform.setTranslation(vector);          tg.setTransform(transform);          tg.addChild(cylinder);          phoneTransformGroup.addChild(tg);      }      @Override      public void actionPerformed(ActionEvent e) {          phoneTransform3D.rotY(angle);          phoneTransformGroup.setTransform(phoneTransform3D);          angle += 0.035;        }  } |

|  |
| --- |
| **ButtonPhone.java** |
| package lab4;  import javax.media.j3d.Appearance;  import javax.media.j3d.Material;  import javax.vecmath.Color3f;  import com.sun.j3d.utils.geometry.\*;  public class ButtonPhone {        public static Cylinder getCylinder(float radius, float height, Color3f emissiveColor) {          int primflags = Primitive.GENERATE\_NORMALS + Primitive.GENERATE\_TEXTURE\_COORDS;          return new Cylinder(radius, height, primflags, getButtonPhoneAppearence(emissiveColor));      }        private static Appearance getButtonPhoneAppearence(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));          return ap;      }  } |

|  |
| --- |
| **FramePhone.java** |
| package lab4;  import javax.media.j3d.\*;  import javax.vecmath.\*;  import com.sun.j3d.utils.geometry.\*;  public class FramePhone {      public static Box getBox(float length, float width, float height) {          int primflags = Primitive.GENERATE\_NORMALS + Primitive.GENERATE\_TEXTURE\_COORDS;          return new Box(length, width, height, primflags, getFramePhoneAppearence());      }      private static Appearance getFramePhoneAppearence() {          Appearance ap = new Appearance();          Color3f emissive = new Color3f(0.0f, 0.05f, 0.0f);          Color3f ambient = new Color3f(.0f, .0f, 1.0f);          Color3f diffuse = new Color3f(0.2f, 0.15f, .15f);          Color3f specular = new Color3f(0.0f, 0.8f, 0.0f);          ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));          return ap;      }  } |

|  |
| --- |
| **ScreenPhone.java** |
| package lab4;  import java.awt.Container;  import javax.media.j3d.Appearance;  import javax.media.j3d.Material;  import javax.media.j3d.Texture;  import javax.media.j3d.TextureAttributes;  import javax.vecmath.Color3f;  import javax.vecmath.Color4f;  import com.sun.j3d.utils.geometry.Box;  import com.sun.j3d.utils.geometry.Primitive;  import com.sun.j3d.utils.image.TextureLoader;  public class ScreenPhone {      public static Box getBox(float length, float width, float height, String picture) {          int primflags = Primitive.GENERATE\_NORMALS + Primitive.GENERATE\_TEXTURE\_COORDS;          return new Box(length, width, height, primflags, getScreenPhoneAppearence(picture));      }      private static Appearance getScreenPhoneAppearence(String picture) {          Appearance ap = new Appearance();          Color3f emissive = new Color3f(0.7f, 1.0f, 1.0f);          Color3f ambient = new Color3f(0.2f, 0.5f, 0.15f);          Color3f diffuse = new Color3f(0.2f, 0.15f, .15f);          Color3f specular = new Color3f(0.0f, 0.8f, 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;      }  } |

# Результат



