

МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота № 4**

з дисципліни “Математичні та алгоритмічні основи компютерної графіки”

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

|  |  |  |
| --- | --- | --- |
| Виконав:  студент III курсу  групи КП-71  Мікулін Володимир Олександрович  варіант № 13  Сумарний бал: |  | Перевірила  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_”  2020р.  викладач  Шкурат Оксана Сергіївна |
|  |  |  |

|  |  |
| --- | --- |
|  |  |

Київ 2020

**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 13: Книга**

**Текст коду програм**

|  |
| --- |
| Book.java |
| public class Book implements ActionListener {  private final float higherEyeLimit = 15.0f;  private final float lowerEyeLimit = 5.0f;  private final float furthestEyeLimit = 28.0f;  private final float closestEyeLimit = 25.0f;  private TransformGroup treeTransformGroup;  private TransformGroup viewingTransformGroup;  private Transform3D treeTransform3D = new Transform3D();  private Transform3D viewingTransform = new Transform3D();  private float angle = 0;  private float eyeHeight;  private float eyeDistance;  private boolean descending = true;  private boolean approaching = true;  public static void main(String[] args) {  Book book = new Book();  }  private Book() {  Timer timer = new Timer(50, this);  SimpleUniverse universe = new SimpleUniverse();  viewingTransformGroup = universe.getViewingPlatform().getViewPlatformTransform();  universe.addBranchGraph(createSceneGraph());  eyeHeight = higherEyeLimit;  eyeDistance = furthestEyeLimit;  timer.start();  }  private Appearance getPagesMaterials() {  Appearance ap = new Appearance();  Color3f emissive = new Color3f(new Color(0, 0, 0));  Color3f ambient = new Color3f(Color.WHITE);  Color3f diffuse = new Color3f(Color.WHITE);  Color3f specular = new Color3f(new Color(0, 0, 0));  ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));  return ap;  }  private Appearance getCoverMaterials() {  Appearance ap = new Appearance();  Color3f emissive = new Color3f(new Color(0, 0, 0));  Color3f ambient = new Color3f(Color.BLACK);  Color3f diffuse = new Color3f(Color.blue);  Color3f specular = new Color3f(Color.BLUE);  ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));  return ap;  }  private Appearance getStripeMaterials() {  Appearance ap = new Appearance();  Color3f emissive = new Color3f(new Color(0, 0, 0));  Color3f ambient = new Color3f(Color.RED);  Color3f diffuse = new Color3f(Color.RED);  Color3f specular = new Color3f(Color.RED);  ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));  return ap;  }  private Appearance getWritingMaterials() {  Appearance ap = new Appearance();  Color3f emissive = new Color3f(new Color(0, 0, 0));  Color3f ambient = new Color3f(Color.BLACK);  Color3f diffuse = new Color3f(Color.GREEN);  Color3f specular = new Color3f(Color.GREEN);  ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));  return ap;  }  private BranchGroup createSceneGraph() {  BranchGroup objRoot = new BranchGroup();  treeTransformGroup = new TransformGroup();  treeTransformGroup.setCapability(TransformGroup.ALLOW\_TRANSFORM\_WRITE);  buildBook();  objRoot.addChild(treeTransformGroup);  createBackground(objRoot);  BoundingSphere bounds = createBoundingSphere(objRoot);  createAmbientLight(objRoot, bounds);  return objRoot;  }  private void createAmbientLight(BranchGroup objRoot, BoundingSphere bounds) {  Color3f ambientColor = new Color3f(1.0f, 1.0f, 1.0f);  AmbientLight ambientLightNode = new AmbientLight(ambientColor);  ambientLightNode.setInfluencingBounds(bounds);  objRoot.addChild(ambientLightNode);  }  private BoundingSphere createBoundingSphere(BranchGroup objRoot) {  BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0), 100000.0);  Color3f light1Color = new Color3f(1.0f, 1, 1);  Vector3f light1Direction = new Vector3f(4.0f, -7.0f, -12.0f);  DirectionalLight light1 = new DirectionalLight(light1Color, light1Direction);  light1.setInfluencingBounds(bounds);  objRoot.addChild(light1);  return bounds;  }  private void createBackground(BranchGroup objRoot) {  Background background = new Background(new Color3f(0.9f, 0.9f, 0.9f)); // white color  BoundingSphere sphere = new BoundingSphere(new Point3d(0, 0, 0), 1000000000);  background.setApplicationBounds(sphere);  objRoot.addChild(background);  }  private TransformGroup buildTG() {  return buildTG(new Vector3f(), new Transform3D());  }  private TransformGroup buildTG(Vector3f translation) {  return buildTG(translation, new Transform3D());  }  private TransformGroup buildTG(Vector3f translation, Transform3D rotation) {  Transform3D transform = new Transform3D();  TransformGroup transformG = new TransformGroup();  transform.setTranslation(translation);  transform.mul(rotation, transform);  transformG.setTransform(transform);  return transformG;  }  private void buildBook() {  TransformGroup pagesTG = buildPages();  buildCover(pagesTG);  buildCover2(pagesTG);  buildCover3(pagesTG);  buildWritings(pagesTG);  buildStripe(pagesTG);  treeTransformGroup.addChild(pagesTG);  }  private void buildStripe(TransformGroup pagesTG) {  Box stripe = new Box(1f, 0.2f, 8.5f, getStripeMaterials());  TransformGroup stripeTG = buildTG(new Vector3f(3, 0, -3));  stripeTG.addChild(stripe);  pagesTG.addChild(stripeTG);  }  private void buildWritings(TransformGroup pagesTG) {  Box writing1 = new Box(4f, 2f, 1f, getWritingMaterials());  TransformGroup writing1TG = buildTG(new Vector3f(0, 0.4f, 3));  writing1TG.addChild(writing1);  pagesTG.addChild(writing1TG);  Box writing2 = new Box(4f, 1f, 4f, getWritingMaterials());  TransformGroup writing2TG = buildTG(new Vector3f(2.55f, 0, 0));  writing2TG.addChild(writing2);  pagesTG.addChild(writing2TG);  }  private void buildCover3(TransformGroup pagesTG) {  Box cover3 = new Box(0.5f, 2.4f, 8.5f, getCoverMaterials());  TransformGroup cover3TG = buildTG(new Vector3f(6, 0, 0));  cover3TG.addChild(cover3);  pagesTG.addChild(cover3TG);  }  private void buildCover2(TransformGroup pagesTG) {  Box cover2 = new Box(6.5f, 0.2f, 8.5f, getCoverMaterials());  TransformGroup cover2TG = buildTG(new Vector3f(0, -2, 0));  cover2TG.addChild(cover2);  pagesTG.addChild(cover2TG);  }  private void buildCover(TransformGroup pagesTG) {  Box cover = new Box(6.5f, 0.2f, 8.5f, getCoverMaterials());  TransformGroup coverTG = buildTG(new Vector3f(0, 2, 0));  coverTG.addChild(cover);  pagesTG.addChild(coverTG);  }  private TransformGroup buildPages() {  Box pages = new Box(6, 2, 8, getPagesMaterials());  TransformGroup pagesTG = buildTG();  pagesTG.addChild(pages);  return pagesTG;  }  // ActionListener interface  @Override  public void actionPerformed(ActionEvent e) {  float delta = 0.02f;  // rotation of the castle  treeTransform3D.rotZ(angle);  treeTransformGroup.setTransform(treeTransform3D);  angle += delta;  // movement of the camera up and down  if (eyeHeight > higherEyeLimit)  descending = true;  else if (eyeHeight < lowerEyeLimit)  descending = false;  if (descending)  eyeHeight -= delta;  else  eyeHeight += delta;  // change camera distance to the scene  if (eyeDistance > furthestEyeLimit)  approaching = true;  else if (eyeDistance < closestEyeLimit)  approaching = false;  if (approaching)  eyeDistance -= delta;  else  eyeDistance += delta;  Point3d eye = new Point3d(eyeDistance, eyeDistance, eyeHeight); // eye  Point3d center = new Point3d(.0f, .0f, 0.1f); // target  Vector3d up = new Vector3d(.0f, .0f, 1.0f);  viewingTransform.lookAt(eye, center, up);  viewingTransform.invert();  viewingTransformGroup.setTransform(viewingTransform);  }  } |

**Висновки**

Виконавши дану лабораторну роботу я вивчила стандарі засоби Java3D для візуалізації зображення, вивчила засоби анімації примітивів та складених об'єктів в Java3D.