

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-83  Линь Дмитро Іванович  варіант № 10 |  |  | Зарахована  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладачем  Шкурат Оксаною Сергіївною |

Київ 2021

**Варіант завдання**

**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 10:**

Паровоз.

**Лістинг коду програми**

import com.sun.j3d.utils.applet.MainFrame;  
import com.sun.j3d.utils.geometry.\*;  
import com.sun.j3d.utils.geometry.Box;  
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;  
  
public class Engine extends Applet implements ActionListener {  
  
 private final TransformGroup engineTG = new TransformGroup();  
 private final Transform3D engineT3d = new Transform3D();  
 private final Timer timer = new Timer(50, this);  
 private float angle = 0;  
  
 public static void main(String[] args) {  
 Engine eng = new Engine();  
 MainFrame mf = new MainFrame(eng, 700, 700);  
 mf.run();  
 }  
  
 private Engine() {  
 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();  
  
 engineTG.setCapability(TransformGroup.*ALLOW\_TRANSFORM\_WRITE*);  
 root.addChild(engineTG);  
 buildEngine();  
  
 TextureLoader loader = new TextureLoader("./assets/zhd.JPG", new Container());  
 ImageComponent2D texture = loader.getImage();  
  
 Background background = new Background(texture);  
 background.setImageScaleMode(Background.*SCALE\_FIT\_MAX*);  
 background.setCapability(Background.*ALLOW\_IMAGE\_WRITE*);  
 BoundingSphere sphere = new BoundingSphere(new Point3d(0,0,0), 100000);  
 background.setApplicationBounds(sphere);  
 root.addChild(background);  
  
 BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0), 100);  
  
 Color sunLightColor = new Color(242, 255, 0);  
 DirectionalLight lightDirect = new DirectionalLight(new Color3f(sunLightColor), new Vector3f(0, 0, 0));  
 lightDirect.setInfluencingBounds(bounds);  
 root.addChild(lightDirect);  
  
 AmbientLight ambientLightNode = new AmbientLight(new Color3f(new Color(100, 255, 255)));  
 ambientLightNode.setInfluencingBounds(bounds);  
 root.addChild(ambientLightNode);  
  
 return root;  
 }  
  
 private TransformGroup getWheel(float x, float y, float z, Color color) {  
 TransformGroup wheel = new TransformGroup();  
 Transform3D t = new Transform3D();  
 Transform3D x1 = new Transform3D();  
 x1.rotX(Math.*PI* / 2);  
 t.mul(x1);  
 Cylinder cylinder = EngineBody.*getCylinder*(0.1f, 0.05f, color);  
 Vector3f vector = new Vector3f(x, y, z);  
 t.setTranslation(vector);  
 wheel.setTransform(t);  
 wheel.addChild(cylinder);  
 return wheel;  
 }  
  
 private TransformGroup getBigWheel(float x, float y, float z, Color color) {  
 TransformGroup wheel = new TransformGroup();  
 Transform3D t = new Transform3D();  
 Transform3D x1 = new Transform3D();  
 x1.rotX(Math.*PI* / 2);  
 t.mul(x1);  
 Cylinder cylinder = EngineBody.*getCylinder*(0.13f, 0.05f, color);  
 Vector3f vector = new Vector3f(x, y, z);  
 t.setTranslation(vector);  
 wheel.setTransform(t);  
 wheel.addChild(cylinder);  
 return wheel;  
 }  
  
 private void buildEngine() {  
  
 TransformGroup corps = new TransformGroup();  
 Transform3D t1 = new Transform3D();  
 Transform3D y1 = new Transform3D();  
 y1.rotZ(Math.*PI* / 2);  
 t1.rotX(Math.*PI* / 2);  
 t1.mul(y1);  
 Cylinder cylinder = EngineBody.*getCylinder*(0.15f, 0.7f, Color.*CYAN*);  
 Vector3f vector = new Vector3f(0f, .0f, 0f);  
 t1.setTranslation(vector);  
 corps.setTransform(t1);  
 corps.addChild(cylinder);  
 engineTG.addChild(corps);  
  
 TransformGroup chimney = new TransformGroup();  
 Transform3D t3 = new Transform3D();  
 Cylinder cylinder3 = EngineBody.*getCylinder*(0.1f, 0.2f, Color.*YELLOW*);  
 Vector3f vector3 = new Vector3f(0f, 0.2f, 0f);  
 t3.setTranslation(vector3);  
 chimney.setTransform(t3);  
 chimney.addChild(cylinder3);  
 engineTG.addChild(chimney);  
  
 TransformGroup cabin = new TransformGroup();  
 Transform3D t4 = new Transform3D();  
 Box box1 = EngineBody.*getBox*(0.22f, 0.22f, 0.15f, Color.*MAGENTA*);  
 Vector3f vector4 = new Vector3f(0.5f, 0.08f, 0f);  
 t4.setTranslation(vector4);  
 cabin.setTransform(t4);  
 cabin.addChild(box1);  
 engineTG.addChild(cabin);  
  
 TransformGroup plate1 = new TransformGroup();  
 Transform3D t5 = new Transform3D();  
 Box box2 = EngineBody.*getBox*(0.55f, 0.04f, 0.15f, Color.*BLUE*);  
 Vector3f vector5 = new Vector3f(0.15f, -0.17f, 0f);  
 t5.setTranslation(vector5);  
 plate1.setTransform(t5);  
 plate1.addChild(box2);  
 engineTG.addChild(plate1);  
  
 TransformGroup wheel1 = getWheel(-0.25f, -0.2f, -0.2f, Color.*BLACK*);  
 engineTG.addChild(wheel1);  
 TransformGroup wheel2 = getWheel(0.1f, -0.2f, -0.2f, Color.*BLACK*);  
 engineTG.addChild(wheel2);  
 TransformGroup wheel4 = getBigWheel(0.5f, -0.17f, -0.2f, Color.*BLACK*);  
 engineTG.addChild(wheel4);  
  
 TransformGroup wheel5 = getWheel(-0.25f, -0.2f, 0.2f, Color.*BLACK*);  
 engineTG.addChild(wheel5);  
 TransformGroup wheel6 = getWheel(0.1f, -0.2f, 0.2f, Color.*BLACK*);  
 engineTG.addChild(wheel6);  
 TransformGroup wheel8 = getBigWheel(0.5f, -0.17f, 0.2f, Color.*BLACK*);  
 engineTG.addChild(wheel8);  
  
 TransformGroup con = new TransformGroup();  
 Transform3D t2 = new Transform3D();  
 Transform3D y2 = new Transform3D();  
 y2.rotZ(Math.*PI* / 2);  
 t2.rotX(Math.*PI* / 2);  
 t2.mul(y2);  
 Cone cone1 = EngineBody.*getCone*(0.2f, 0.15f, Color.*PINK*);  
 Vector3f vector2 = new Vector3f(-0.45f, 0f, 0f);  
 t2.setTranslation(vector2);  
 con.setTransform(t2);  
 con.addChild(cone1);  
 engineTG.addChild(con);  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 engineT3d.rotY(angle);  
 angle += 0.05;  
 if (angle >= 25) {  
 angle = 0;  
 }  
  
  
 engineTG.setTransform(engineT3d);  
 }  
}

**Результат у директорії разом з цим документом**