

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  Науменко Артур Сергійович  (*прізвище, ім’я, по батькові*)  варіант № 9 |  |  | Зарахована  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладачем  Шкурат Оксаною Сергіївною (*прізвище, ім’я, по батькові*) |

Київ 2023

**Варіант 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:** 9. Автомобіль

**Лістинг коду програми**

import javax.media.j3d.BranchGroup;  
import com.sun.j3d.utils.geometry.Box;  
import com.sun.j3d.utils.geometry.Cylinder;  
import com.sun.j3d.utils.universe.SimpleUniverse;  
import javax.media.j3d.BoundingSphere;  
import javax.media.j3d.DirectionalLight;  
import javax.vecmath.Color3f;  
import javax.vecmath.Point3d;  
import javax.vecmath.Vector3f;  
import javax.media.j3d.Appearance;  
import javax.media.j3d.\*;  
import javax.swing.Timer;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class Car implements ActionListener{  
 private TransformGroup carTransformGroup;  
 private Transform3D carTransform3D = new Transform3D();  
 private float y\_angle = 0;  
 private Timer timer;  
 public static void main(String[] args){  
 new Car();  
 }  
 public Car()  
 {  
 SimpleUniverse universe = new SimpleUniverse();  
 universe.getViewingPlatform().setNominalViewingTransform();  
  
 BranchGroup group = new BranchGroup();  
 carTransformGroup = new TransformGroup();  
 carTransformGroup.setCapability(TransformGroup.*ALLOW\_TRANSFORM\_WRITE*);  
 group.addChild(carTransformGroup);  
 carTransformGroup.setTransform(carTransform3D);  
  
 timer = new Timer(50, this);  
 timer.start();  
  
 Box body = get\_body();  
 carTransformGroup.addChild(body);  
  
 TransformGroup upper\_body = get\_upper\_body();  
 carTransformGroup.addChild(upper\_body);  
  
 TransformGroup[] whs = get\_wheels();  
 for (int i=0;i<whs.length; i++){  
 carTransformGroup.addChild(whs[i]);  
 }  
  
 TransformGroup[] ws = get\_windows();  
 for (int i=0;i<ws.length; i++){  
 carTransformGroup.addChild(ws[i]);  
 }  
 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);  
 group.addChild(light1);  
 universe.getViewingPlatform().setNominalViewingTransform();  
 universe.addBranchGraph(group);  
 }  
 public Box get\_body(){  
 Appearance ap = get\_body\_material();  
 return new Box(0.4f, 0.1f, 0.16f, ap);  
 }  
 public TransformGroup get\_upper\_body(){  
 Appearance ap = get\_body\_material();  
 Box upper\_body = new Box(0.25f, 0.07f, 0.16f, ap);  
 TransformGroup tgTop = new TransformGroup();  
 Transform3D transformTop = new Transform3D();  
 Vector3f vectorTop = new Vector3f(.07f, 0.17f, .0f);  
 transformTop.setTranslation(vectorTop);  
 tgTop.setTransform(transformTop);  
 tgTop.addChild(upper\_body);  
 return tgTop;  
 }  
 public TransformGroup[] get\_wheels(){  
 float x\_ch = 0.2f;  
 float z\_ch = -0.15f;  
 float y\_ch = -0.1f;  
 TransformGroup[] tr = new TransformGroup[4];  
 for (int i=0;i<4;i++){  
 int x = i<2 ? 1: -1;  
 int z = i%2==0 ? 1:-1;  
 Cylinder cyl = new Cylinder(0.05f, 0.05f, get\_wheels\_material());  
 tr[i] = new TransformGroup();  
 Transform3D transformTop = new Transform3D();  
 transformTop.rotX(1.57);  
 Vector3f vectorTop = new Vector3f(x\*x\_ch, y\_ch, z\*z\_ch);  
 transformTop.setTranslation(vectorTop);  
 tr[i].setTransform(transformTop);  
 tr[i].addChild(cyl);  
 }  
 return tr;  
 }  
 public TransformGroup[] get\_windows(){  
 Appearance ap = get\_windows\_material();  
 TransformGroup[] tr = new TransformGroup[4];  
 tr[0] = new TransformGroup();  
 Box box = new Box(0.005f,0.05f, 0.14f, ap);  
 Transform3D transformTop = new Transform3D();  
 Vector3f vectorTop = new Vector3f(-0.18f,0.17f,0);  
 transformTop.setTranslation(vectorTop);  
 tr[0].setTransform(transformTop);  
 tr[0].addChild(box);  
 for (int i=1;i<3;i++){  
 tr[i] = new TransformGroup();  
 Box ibox = new Box(0.23f, 0.05f, 0.005f, ap);  
 Transform3D itransformTop = new Transform3D();  
 int koef = i<2? 1 : -1;  
 Vector3f ivectorTop = new Vector3f(.07f, 0.17f, koef\*0.16f);  
 itransformTop.setTranslation(ivectorTop);  
 tr[i].setTransform(itransformTop);  
 tr[i].addChild(ibox);  
 }  
 tr[3] = new TransformGroup();  
 Box аbox = new Box(0.005f,0.05f, 0.14f, ap);  
 Transform3D аtransformTop = new Transform3D();  
 Vector3f аvectorTop = new Vector3f(0.32f,0.17f,0);  
 аtransformTop.setTranslation(аvectorTop);  
 tr[3].setTransform(аtransformTop);  
 tr[3].addChild(аbox);  
 return tr;  
 }  
 public Appearance get\_windows\_material() {  
 Appearance ap = new Appearance();  
 Color3f emissive = new Color3f(0.2f, 0.02f, 0.02f);  
 Color3f ambient = new Color3f(0.2f, 0.2f, 0.2f);  
 Color3f diffuse = new Color3f(0.2f, 0.2f, .2f);  
 Color3f specular = new Color3f(0.0f, 0.1f, 0.1f);  
 ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));  
 return ap;  
 }  
  
 public Appearance get\_body\_material(){  
 Appearance ap = new Appearance();  
 Color3f emissive = new Color3f(0.2f, 0.02f, 0.08f);  
 Color3f ambient = new Color3f(0.2f, 0.2f, 0.8f);  
 Color3f diffuse = new Color3f(0.5f, 0.2f, .8f);  
 Color3f specular = new Color3f(0.3f, 0.1f, 0.15f);  
 ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));  
 return ap;  
 }  
 public Appearance get\_wheels\_material(){  
 Appearance ap = new Appearance();  
 Color3f emissive = new Color3f(0.0f, 0.0f, 0.0f);  
 Color3f ambient = new Color3f(0.f, 0.f, 0.f);  
 Color3f diffuse = new Color3f(0.f, 0.f, 0.f);  
 Color3f specular = new Color3f(0.2f, 0.2f, 0.2f);  
 ap.setMaterial(new Material(ambient, emissive, diffuse, specular, 1.0f));  
 return ap;  
 }  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 carTransform3D.rotY(y\_angle);  
 carTransformGroup.setTransform(carTransform3D);  
 y\_angle += 0.05;  
 }  
}

**Результат**

