



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

Мричко Богдан
(прізвище, ім'я, по батькові)

Зарахована
“ ____ ” “ ____ ” 20 ____ р.
викладачем

Шкурат Оксаною Сергіївною
(прізвище, ім'я, по батькові)

варіант № 15

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.

Завдання за варіантом: Качелі

Код програми

Swing.java

```
package sample;

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 Swing extends Applet implements ActionListener {

    private static String assetsDir = System.getProperty("user.dir") +
"\\assets\\";

    private final TransformGroup animationGroup = new TransformGroup();
    private final Transform3D animationTransform = new Transform3D();

    private double dAngleX = 0.01;
    private double dAngleY = 0.02;

    private double angleX = 0;
    private double angleY = 0;
    private final Timer timer = new Timer(10, this);

    public static void main(String[] args) {
        new Swing();
    }

    public Swing()
    {
        SimpleUniverse universe = new SimpleUniverse();
        BranchGroup group = new BranchGroup();
        universe.getViewingPlatform().setNominalViewingTransform();
        animationGroup.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);

        addLight(group);
        addBackground(group);

        animationGroup.addChild(SwingBody.getBody());
        group.addChild(animationGroup);

        universe.addBranchGraph(group);

        timer.start();
    }

    public void addLight(BranchGroup group) {
        Color3f light1Color = new Color3f(0.8f, 1.1f, 0.1f);
        BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0,
0.0), 1000);
        Vector3f light1Direction = new Vector3f(4.0f, -7.0f, -12.0f);
```

```

        DirectionalLight light1 = new DirectionalLight(light1Color,
light1Direction);
        light1.setInfluencingBounds(bounds);
        group.addChild(light1);
    }

    public void addBackground(BranchGroup group) {
        ImageComponent2D texture = new TextureLoader(assetsDir +
"background.jpg", new Container()).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);
        group.addChild(background);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        animationTransform.rotY(angleX);
        animationGroup.setTransform(animationTransform);
        angleX += dAngleX;
        angleY += dAngleY;
        angleX = circleNumber(angleX, 0, Math.PI * 2);
        angleY = circleNumber(angleY, 0, Math.PI * 2);
    }

    private static double circleNumber(double num, double left, double right)
    {
        if (num < left) {
            return right;
        }
        if (num > right) {
            return left;
        }
        return num;
    }
}

```

SwingBody.java

```

package sample;

import com.sun.j3d.utils.geometry.Primitive;
import com.sun.j3d.utils.geometry.Cylinder;
import com.sun.j3d.utils.geometry.Box;
import com.sun.j3d.utils.image.TextureLoader;

import javax.media.j3d.*;
import javax.vecmath.*;
import java.awt.*;

public class SwingBody {

```

```

    private static String assetsDir = System.getProperty("user.dir") +
    "\\assets\\";
    private static int primFlags = Primitive.GENERATE_NORMALS +
    Primitive.GENERATE_TEXTURE_COORDS;

    public static TransformGroup getBody() {
        TransformGroup transformGroup = new TransformGroup();
        transformGroup.addChild(getSit(-0.5f));
        transformGroup.addChild(getBack(-0.4f));
        transformGroup.addChild(getArmrest(-0.42f, 0.2f));
        transformGroup.addChild(getArmrest(-0.42f, -0.2f));
        transformGroup.addChild(getRope(1.0f, 0f, 0.2f));
        transformGroup.addChild(getRope(1.0f, 0f, -0.2f));
        transformGroup.addChild(getTrunk(0.5f));
        return transformGroup;
    }

    public static TransformGroup getTrunk(float y) {
        Box trunk = new Box(0.4f, .1f, .1f, primFlags,
        getComponentAppearance("wood.jpgg"));
        Vector3f vectorTop = new Vector3f(.0f, y, .0f);
        return getTGTemplate(trunk, vectorTop);
    }

    public static TransformGroup getRope(float h, float y, float xMove) {
        Cylinder rope = new Cylinder(0.01f, h, primFlags,
        getComponentAppearance("rope.jpgg"));
        Vector3f vectorTop = new Vector3f(xMove, y, .0f);
        return getTGTemplate(rope, vectorTop);
    }

    public static TransformGroup getArmrest (float y, float zMove) {
        Box sit = new Box(0.01f, .08f, .1f, primFlags,
        getComponentAppearance("pillow.jpgg"));
        Vector3f vectorTop = new Vector3f(zMove, y, 0.0f);
        return getTGTemplate(sit, vectorTop);
    }

    public static TransformGroup getBack(float y) {
        Box sit = new Box(0.2f, .1f, .01f, primFlags,
        getComponentAppearance("pillow.jpgg"));
        Vector3f vectorTop = new Vector3f(.0f, y, .1f);
        return getTGTemplate(sit, vectorTop);
    }

    public static TransformGroup getSit(float y) {
        Box sit = new Box(0.2f, .03f, .1f, primFlags,
        getComponentAppearance("pillow.jpgg"));
        Vector3f vectorTop = new Vector3f(.0f, y, .0f);
        return getTGTemplate(sit, vectorTop);
    }

    public static TransformGroup getTGTemplate(javax.media.j3d.Node node,
    Vector3f vector) {
        TransformGroup tg = new TransformGroup();
        Transform3D transformTop = new Transform3D();
        transformTop.setTranslation(vector);
        tg.setTransform(transformTop);
        tg.addChild(node);
        return tg;
    }

    private static Appearance getComponentAppearance(String resource) {

```

```
TextureLoader loader = new TextureLoader(assetsDir + resource, new  
Container());  
Texture texture = loader.getTexture();  
Appearance appearance = new Appearance();  
appearance.setTexture(texture);  
return appearance;  
}  
}
```

Результат



