

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ
УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

Лабораторна робота №4

з дисципліни “Побудова найпростіших тривимірних об'єктів за
допомогою бібліотеки Java3D та їх анімація”

Виконав
студент III курсу

групи КП-83

Коваль Андрій Олександрович

(прізвище, ім'я, по батькові)

варіант № 8

Зарахована

“ _____ ” “ _____ ” 20__ р.

викладачем Шкурат Оксаною Сергіївною

(прізвище, ім'я, по батькові)

Київ 2021

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 взаємодії з об'єктом за допомогою миші та клавіатури.

Варіант 8 - Сніговик

Лістинг коду програми

SnowMan.java

```
package com.lab4;

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 SnowMan 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 SnowMan();
}

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

    addLight(group);
    addBackground(group);

    animationGroup.addChild(SnowmanBody.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.rotX(angleX);
//    doesn't work ??
//    animationTransform.rotY(angleY);
    animationGroup.setTransform(animationTransform);
    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;
    }
}

```

SnomanBody.java

```

package com.lab4;

import com.sun.j3d.utils.geometry.*;
import com.sun.j3d.utils.image.TextureLoader;

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

public class SnowmanBody {

    private static float bodyOffset = -0.1f;

    private static float topBallY = 0.65f + bodyOffset;
    private static float middleBallY = 0.45f + bodyOffset;
    private static float bottomBallY = 0.1f + bodyOffset;

    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(getSnowball(0.1f, topBallY));
        transformGroup.addChild(getSnowball(0.15f, middleBallY));
        transformGroup.addChild(getSnowball(0.25f, bottomBallY));
        transformGroup.addChild(getNose());
        transformGroup.addChild(getEye(new Vector3f(0.05f, topBallY + 0.05f,
0.05f)));
        transformGroup.addChild(getEye(new Vector3f(0.07f, topBallY + 0.05f,
-0.02f)));
        return transformGroup;
    }

    public static TransformGroup getSnowball(float r, float y) {
        TransformGroup tg = new TransformGroup();
    }
}

```

```

        Transform3D transformTop = new Transform3D();
        Sphere sphere = new Sphere(r, primFlags, getSnowballAppearance());
        Vector3f vectorTop = new Vector3f(.0f, y, .0f);
        transformTop.setTranslation(vectorTop);
        tg.setTransform(transformTop);
        tg.addChild(sphere);
        return tg;
    }

    public static TransformGroup getNose() {
        TransformGroup tg = new TransformGroup();
        Transform3D transformTop = new Transform3D();
        transformTop.setRotation(new Matrix3d(new double[] {
            0.5, 0.866, 0,
            0, 0, 1,
            -0.8661, 0.5, 0,
        }));
        Cone carrot = new Cone(0.03f, 0.3f, primFlags, getCarrotAppearance());
        Vector3f vectorTop = new Vector3f(0.2f, topBallY, 0.1f);
        transformTop.setTranslation(vectorTop);
        tg.setTransform(transformTop);
        tg.addChild(carrot);
        return tg;
    }

    public static TransformGroup getEye(Vector3f translation) {
        TransformGroup tg = new TransformGroup();
        Transform3D transformEye = new Transform3D();
        Sphere eye = new Sphere(0.03f, primFlags, getEyesAppearance());
        transformEye.setTranslation(translation);
        tg.setTransform(transformEye);
        tg.addChild(eye);
        return tg;
    }

    private static Appearance getEyesAppearance() {
        Appearance ap = new Appearance();
        Color3f emissive = new Color3f(new Color(0,0, 0));
        Color3f ambient = new Color3f(new Color(100,38, 38));
        Color3f diffuse = new Color3f(new Color(178,38, 38));
        Color3f specular = new Color3f(new Color(0,0, 0));
        ap.setMaterial(new Material(ambient, emissive, diffuse, specular,
1.0f));
        return ap;
    }

    private static Appearance getSnowballAppearance() {
        TextureLoader loader = new TextureLoader(assetsDir + "snow.jpg", 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);
        Appearance appearance = new Appearance();
        appearance.setTexture(texture);
        appearance.setTextureAttributes(texAttr);
    }

```

```
        return appearance;
    }

    private static Appearance getCarrotAppearance() {
        TextureLoader loader = new TextureLoader(assetsDir + "carrot.jpg", new
Container());
        Texture texture = loader.getTexture();
        texture.setBoundaryModeS(Texture.WRAP);
        texture.setBoundaryModeT(Texture.WRAP);
        TextureAttributes texAttr = new TextureAttributes();
        texAttr.setTextureMode(TextureAttributes.MODULATE);
        Appearance appearance = new Appearance();
        appearance.setTexture(texture);
        appearance.setTextureAttributes(texAttr);
        return appearance;
    }
}
```

Приклад результату

