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 №6

з дисципліни “Математичні та алгоритмічні основи комп'ютерної графіки”

Варіант 7

Виконав
студент 3 курсу
групи КП-81
Каснер Максим

Київ 2021

7. Анімація одноокого циклопа Майка (із мультфільму) mike.obj. Повинен рухати руками і ногами, пересуватися по екрану.

Код програми

Main.java

```
import javax.vecmath.*;
import com.sun.j3d.utils.universe.*;
import javax.media.j3d.*;
import com.sun.j3d.utils.behaviors.vp.*;
import com.sun.j3d.utils.image.TextureLoader;
import javax.swing.JFrame;
import com.sun.j3d.loaders.*;
import com.sun.j3d.loaders.objectfile.*;

import java.util.Hashtable;


public class Main extends JFrame
{
    public Canvas3D myCanvas3D;

    public Main() {

        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        myCanvas3D = new Canvas3D(SimpleUniverse.getPreferredConfiguration());

        SimpleUniverse simpUniv = new SimpleUniverse(myCanvas3D);

        simpUniv.getViewingPlatform().setNominalViewingTransform();

        createSceneGraph(simpUniv);

        addLight(simpUniv);

        OrbitBehavior ob = new OrbitBehavior(myCanvas3D);
        ob.setSchedulingBounds(new      BoundingSphere(new      Point3d(0.0,      0.0,      0.0),
Double.MAX_VALUE));
        simpUniv.getViewingPlatform().setViewPlatformBehavior(ob);

        setTitle("Lab6");
```

```
        setSize(700, 700);
        getContentPane().add("Center", myCanvas3D);
        setVisible(true);
    }
    public static void main(String[] args)
    {
        new Main();
    }

    public void createSceneGraph(SimpleUniverse su)
    {

        ObjectFile f = new ObjectFile(ObjectFile.RESIZE);
        Scene mikeScene = null;
        try
        {
            mikeScene = f.load("files/mike.obj");
        }
        catch (Exception e)
        {
            System.out.println("File loading failed:" + e);
        }

        Transform3D scaling = new Transform3D();
        scaling.setScale(1.0/6);
        Transform3D tfMike = new Transform3D();
        tfMike.rotX(Math.PI/3);
        tfMike.mul(scaling);
        TransformGroup tgMike = new TransformGroup(tfMike);
        TransformGroup sceneGroup = new TransformGroup();


        Hashtable mikeNamedObjects = mikeScene.getNamedObjects();

        Shape3D leftLeg = (Shape3D) mikeNamedObjects.get("left_leg");
        Shape3D rightLeg = (Shape3D) mikeNamedObjects.get("right_leg");
        Shape3D leftHand = (Shape3D) mikeNamedObjects.get("left_hand");
        Shape3D rightHand = (Shape3D) mikeNamedObjects.get("right_hand");
        Shape3D monstr = (Shape3D) mikeNamedObjects.get("monstr");

        TextureAttributes texAttr = new TextureAttributes();
        texAttr.setTextureMode(TextureAttributes.MODULATE);

        TransformGroup transformGroup = new TransformGroup();
        transformGroup.addChild(monstr.cloneTree());
```

```
        TransformGroup leftLegGr = new TransformGroup();
        TransformGroup rightLegGr = new TransformGroup();
        TransformGroup leftHandGr = new TransformGroup();
        TransformGroup rightHandGr = new TransformGroup();
        leftLegGr.addChild(leftLeg.cloneTree());
        rightLegGr.addChild(rightLeg.cloneTree());
        leftHandGr.addChild(leftHand.cloneTree());
        rightHandGr.addChild(rightHand.cloneTree());


        BoundingSphere bounds = new BoundingSphere(new Point3d(120.0, 250.0, 100.0),
Double.MAX_VALUE);
        BranchGroup theScene = new BranchGroup();
        Transform3D tCrawl = new Transform3D();
        Transform3D tCrawl1 = new Transform3D();
        tCrawl.rotY(-90D);
        tCrawl1.rotX(-90D);
        long crawlTime = 10000;
        Alpha crawlAlpha = new Alpha(1,
                Alpha.INCREASING_ENABLE,
                0,
                0, crawlTime, 0, 0, 0, 0, 0);
        float crawlDistance = 3.0f;
        PositionInterpolator posICrawl = new PositionInterpolator(crawlAlpha,
                sceneGroup, tCrawl, -9.0f, crawlDistance);

        long crawlTime1 = 300000000;
        Alpha crawlAlpha1 = new Alpha(1,
                Alpha.INCREASING_ENABLE,
                crawlTime1,
                0, crawlTime1, 0, 0, 0, 0, 0);
        float crawlDistance1 = 15.0f;
        PositionInterpolator posICrawl1 = new PositionInterpolator(crawlAlpha1,
                sceneGroup, tCrawl1, -9.0f, crawlDistance1);


        BoundingSphere bs = new BoundingSphere(new Point3d(0.0, 0.0, 0.0), Double.MAX_VALUE);
        posICrawl.setSchedulingBounds(bs);
        posICrawl1.setSchedulingBounds(bs);
        sceneGroup.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        sceneGroup.addChild(posICrawl);


        int timeStart = 500;
        int timeRotationHour = 500;

        Transform3D leftLegRotationAxis = new Transform3D();
```

```
        leftLegRotationAxis.rotZ(Math.PI / 2);
        Alpha    leftLegRotationAlpha    =    new    Alpha(-1,    Alpha.INCREASING_ENABLE    |
Alpha.DECREASING_ENABLE, timeStart, 0,
                timeRotationHour, 0, 0, timeRotationHour, 0, 0);
        RotationInterpolator  leftLegRotation  =  new  RotationInterpolator(leftLegRotationAlpha,
leftLegGr,
                leftLegRotationAxis, (float) Math.PI / 4, 0.0f);
        leftLegRotation.setSchedulingBounds(bounds);



        Transform3D rightHandRotationAxis = new Transform3D();
        rightHandRotationAxis.rotZ(Math.PI / 2);
        RotationInterpolator rightHandRotation = new RotationInterpolator(leftLegRotationAlpha,
rightHandGr,
                rightHandRotationAxis, (float) Math.PI / 4, 0.0f);
        rightHandRotation.setSchedulingBounds(bounds);



        Transform3D rightLegRotationAxis = new Transform3D();
        rightLegRotationAxis.rotZ(Math.PI / 2);
        Alpha    rightLegRotationAlpha    =    new    Alpha(-1,    Alpha.INCREASING_ENABLE    |
Alpha.DECREASING_ENABLE, 0, 0,
                timeRotationHour, 0, 0, timeRotationHour, 0, 0);
        RotationInterpolator rightLegRotation = new RotationInterpolator(rightLegRotationAlpha,
rightLegGr,
                rightLegRotationAxis, (float) Math.PI / 4, 0.0f);
        rightLegRotation.setSchedulingBounds(bounds);

        Transform3D leftHandRotationAxis = new Transform3D();
        leftHandRotationAxis.rotZ(Math.PI / 2);
        RotationInterpolator leftHandRotation = new RotationInterpolator(rightLegRotationAlpha,
leftHandGr,
                leftHandRotationAxis, (float) Math.PI / 4, 0.0f);
        leftHandRotation.setSchedulingBounds(bounds);



        leftLegGr.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        rightLegGr.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        leftHandGr.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        rightHandGr.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        leftLegGr.addChild(leftLegRotation);
        rightLegGr.addChild(rightLegRotation);
        leftHandGr.addChild(leftHandRotation);
        rightHandGr.addChild(rightHandRotation);

        sceneGroup.addChild(transformGroup);
        sceneGroup.addChild(leftLegGr);
```

```
        sceneGroup.addChild(rightLegGr);
        sceneGroup.addChild(leftHandGr);
        sceneGroup.addChild(rightHandGr);
        tgMike.addChild(sceneGroup);
        theScene.addChild(tgMike);

        Background bg = new Background(new Color3f(0.5f,0.5f,0.5f));

        TextureLoader myLoader = new  TextureLoader("files/mikee.png",this);
        ImageComponent2D myImage = myLoader.getImage( );
        bg.setImage(myImage);
        bg.setApplicationBounds(bounds);
        theScene.addChild(bg);
        theScene.compile();

        su.addBranchGraph(theScene);
    }


    public void addLight(SimpleUniverse su)
    {
        var bgLight = new BranchGroup();
        var bounds = new BoundingSphere(new Point3d(0.0,0.0,0.0), 100.0);
        var lightColour1 = new Color3f(0.5f,1.0f,1.0f);
        var lightDir1 = new Vector3f(-1.0f,0.0f,-0.5f);
        var light1 = new DirectionalLight(lightColour1, lightDir1);
        light1.setInfluencingBounds(bounds);
        bgLight.addChild(light1);
        su.addBranchGraph(bgLight);
    }
}
```

Результати роботи програми :