



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 № 5

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

Виконав

студент III курсу

групи КП-82

Мельничук Олексій Геннадійович

(прізвище, ім'я, по батькові)

Зарахована

“ ____ ” “ ____ ” 20__ р.

викладачем

Шкурат Оксаною Сергіївною

(прізвище, ім'я, по батькові)

варіант № 12

Тема: Імпорт тривимірних моделей у середовище програмування java3D, обробка та маніпуляція цих зображень

Мета:

1. Здобути навички імпорту моделей, побудованих у тривимірних редакторах, (об'єктів форматів .obj, .lwo, .3ds) до бібліотеки java3D
2. Навчитися анімувати імпортовані об'єкти.

Завдання

Імпортувати моделі тривимірних об'єктів форматів, що визначені варіантом.

Створити реалістичну анімацію об'єкту. Додати до сцени фон, інші об'єкти для надання сцені реалістичного вигляду. Для цього використати текстури, матеріали, імпортувати додаткові об'єкти з відкритих бібліотек, за бажанням створити прості об'єкти у графічному редакторі.

Студенти, які мають непарний номер варіанту у списку групи імпортують моделі формату .obj, парний варіант – .lwo.

Код програми

animatecar.java

```
package lab5;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import javax.media.j3d.*;
import javax.swing.JFrame;
import javax.swing.Timer;
import javax.vecmath.*;

public class AnimateCar implements ActionListener, KeyListener {
    private Button go;
    private TransformGroup wholeCar;
    private Transform3D translateTransform;
    private Transform3D rotateTransformX;
    private Transform3D rotateTransformY;

    private JFrame mainFrame;

    private float zoom=2.0f;
    private float xloc=0.0f;
    private float yloc=0.0f;
    private float zloc=0.0f;

    private int moveType=1;
    private Timer timer;

    public AnimateCar(TransformGroup wholeCar, Transform3D trans, JFrame frame){
        go = new Button("Go");
        this.wholeCar=wholeCar;
        this.translateTransform=trans;
        this.mainFrame=frame;

        rotateTransformX= new Transform3D();
        rotateTransformY= new Transform3D();

        FirstMainClass.canvas.addKeyListener(this);
        timer = new Timer(100, this);

        Panel p =new Panel();
        p.add(go);
        mainFrame.add("North",p);
        go.addActionListener(this);
        go.addKeyListener(this);
    }

    private void initialCarState(){
        xloc=0.0f;
        yloc=0.0f;
        zloc=0.0f;
        zoom=2.0f;

        moveType=1;

        rotateTransformY.rotY(-Math.PI/2.8);
        translateTransform.mul(rotateTransformY);
    }
}
```

```

        if(timer.isRunning()) timer.stop();

        go.setLabel("Go");
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        // start timer when button is pressed
        if (e.getSource()==go) {
            if (!timer.isRunning()) {
                timer.start();
                go.setLabel("Stop");
            }
            else {
                timer.stop();
                go.setLabel("Go");
            }
        }
        else {
            Move(moveType);
            translateTransform.setScale(new Vector3d(zoom, zoom, zoom));
            translateTransform.setTranslation(new Vector3f(xloc, yloc, zloc));
            wholeCar.setTransform(translateTransform);
        }
    }

    private void Move(int mType){
        if(mType==1)
        {
            //rotate
            rotateTransformY.rotY(Math.PI/240);
            translateTransform.mul(rotateTransformY);
        }
        if(mType==2){
            //do nothing
        }
        if(mType==3)
        {
            //rotate
            rotateTransformY.rotY(-Math.PI/30);
            translateTransform.mul(rotateTransformY);
        }
    }

    @Override
    public void keyTyped(KeyEvent e) {
        //Invoked when a key has been typed.
    }

    @Override
    public void keyPressed(KeyEvent e) {
        //Invoked when a key has been pressed.
        if (e.getKeyChar()=='d') {xloc = xloc + .05f;}
        if (e.getKeyChar()=='a') {xloc = xloc - .05f;}
        if (e.getKeyChar()=='w') {yloc = yloc + .05f;}
        if (e.getKeyChar()=='s') {yloc = yloc - .05f;}
        if (e.getKeyChar()=='z') {zoom = zoom + .05f;}
        if (e.getKeyChar()=='x') {zoom = zoom - .05f;}

        if (e.getKeyChar()=='t') {
            rotateTransformX.rotX(Math.PI/180);
            translateTransform.mul(rotateTransformX);
        }
    }

```

```

    }
    if (e.getKeyChar()=='g') {
        rotateTransformX.rotX(- Math.PI/180);
        translateTransform.mul(rotateTransformX);
    }
    if (e.getKeyChar()=='f') {
        rotateTransformY.rotY(Math.PI/180);
        translateTransform.mul(rotateTransformY);
    }
    if (e.getKeyChar()=='h') {
        rotateTransformY.rotY(-Math.PI/180);
        translateTransform.mul(rotateTransformY);
    }

    if (e.getKeyChar()=='1') {
        moveType = 1;
    }
    if (e.getKeyChar()=='2') {
        moveType = 2;
    }
    if (e.getKeyChar()=='3') {
        moveType = 3;
    }
}

@Override
public void keyReleased(KeyEvent e) {
    // Invoked when a key has been released.
}
}

```

firstmainclass.java

```
package lab5;

import com.sun.j3d.utils.universe.*;

import java.awt.Color;
import javax.media.j3d.*;
import javax.media.j3d.Material;
import javax.vecmath.*;
import javax.media.j3d.Background;

import com.sun.j3d.loaders.*;
import com.sun.j3d.loaders.objectfile.ObjectFile;
import com.sun.j3d.loaders.lw3d.Lw3dLoader;
import com.sun.j3d.utils.image.TextureLoader;
import java.awt.*;
import java.io.FileReader;
import java.io.IOException;
import java.util.Map;
import javax.swing.JFrame;

public class FirstMainClass extends JFrame {
    static SimpleUniverse universe;
    static Scene scene;
    static Map<String, Shape3D> nameMap;
    static BranchGroup root;
    static Canvas3D canvas;

    static String assetPath =
"C:\\\\Users\\datru\\Desktop\\study2021\\maokg\\lab5\\res\\";
    static String modelName = "camaro2.obj";
    static String bgName = "garage.jpg";

    static TransformGroup wholeCar;
    static Transform3D transform3D;

    public FirstMainClass() throws IOException {
        configureWindow();
        configureCanvas();
        configureUniverse();
        addModelToUniverse();
        setCarElementsList();
        addAppearance();
        addImageBackground();
        addLightToUniverse();
        addOtherLight();
        ChangeViewAngle();
        root.compile();
        universe.addBranchGraph(root);
    }

    private void configureWindow() {
        setTitle("Car Animation Example");
        setSize(760, 640);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    private void configureCanvas() {
        canvas = new Canvas3D(SimpleUniverse.getPreferredConfiguration());
        canvas.setDoubleBufferEnable(true);
    }
}
```

```

        getContentPane().add(canvas, BorderLayout.CENTER);
    }

    private void configureUniverse() {
        root = new BranchGroup();
        universe = new SimpleUniverse(canvas);
        universe.getViewingPlatform().setNominalViewingTransform();
    }

    private void addModelToUniverse() throws IOException {
        scene = getSceneFromFile(assetPath + modelName);
        root = scene.getSceneGroup();
    }

    private void addLightToUniverse() {
        Bounds bounds = new BoundingSphere();
        Color3f color = new Color3f(96 / 255f, 96 / 255f, 96 / 255f);
        Vector3f lightdirection = new Vector3f(0f, -1f, 0f);
        DirectionalLight dirlight = new DirectionalLight(color, lightdirection);
        dirlight.setInfluencingBounds(bounds);
        root.addChild(dirlight);
    }

    private void printModelElementsList(Map<String, Shape3D> nameMap) {
        for (String name : nameMap.keySet()) {
            System.out.printf("Name: %s\n", name);
        }
    }

    private void setCarElementsList() {
        nameMap = scene.getNamedObjects();

        // Print elements of your model:
        printModelElementsList(nameMap);

        wholeCar = new TransformGroup();
        transform3D = new Transform3D();
        transform3D.setScale(new Vector3d(1, 1, 1));
        wholeCar.setTransform(transform3D);

        for (String name : nameMap.keySet()) {
            root.removeChild(nameMap.get(name));
            wholeCar.addChild(nameMap.get(name));
            wholeCar.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        }

        root.addChild(wholeCar);
    }

    Texture getTexture(String path) {
        TextureLoader textureLoader = new TextureLoader(path, canvas);
        Texture texture = textureLoader.getTexture();
        texture.setBoundaryModeS(Texture.WRAP);
        texture.setBoundaryModeT(Texture.WRAP);
        texture.setBoundaryColor(new Color4f(0.0f, 1.0f, 0.0f, 0.0f));
        return texture;
    }

    Material getMaterial() {
        Material material = new Material();

        Color3f ambient = new Color3f(1.0f, 1.0f, 1.0f);

```

```

        Color3f diffuse = new Color3f(1.0f, 1.0f, 1.0f);
        Color3f specular = new Color3f(1.0f, 1.0f, 1.0f);

//      material.setAmbientColor(new Color3f(new Color(221, 221, 221)));
//      material.setDiffuseColor(new Color3f(new Color(200, 200, 200)));
//      material.setSpecularColor(new Color3f(new Color(200, 200, 200)));

        material.setAmbientColor(ambient);
        material.setDiffuseColor(diffuse);
        material.setSpecularColor(specular);
        material.setShininess(1f);
        material.setLightingEnable(true);
        return material;
    }

    private void addAppearance() throws IOException {
        for (String name : nameMap.keySet()) {
            Shape3D car = nameMap.get(name);

            Appearance carAppearance = new Appearance();
            carAppearance.setTexture(getTexture(assetPath + ".png/" + name +
".png"));

            TextureAttributes texAttr = new TextureAttributes();
            texAttr.setTextureMode(TextureAttributes.COMBINE);
            carAppearance.setTextureAttributes(texAttr);
            carAppearance.setMaterial(getMaterial());

            car.setAppearance(carAppearance);
        }

        private void addColorBackground() {
            Background background = new Background(new Color3f(Color.CYAN));
            BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0),
100.0);

            background.setApplicationBounds(bounds);
            root.addChild(background);
        }

        private void addImageBackground() {
            TextureLoader t = new TextureLoader(assetPath + bgName, canvas);
            Background background = new Background(t.getImage());
            background.setImageScaleMode(Background.SCALE_FIT_ALL);
            BoundingSphere bounds = new BoundingSphere(new Point3d(0.0, 0.0, 0.0),
100.0);

            background.setApplicationBounds(bounds);
            root.addChild(background);
        }

        private void ChangeViewAngle() {
            ViewingPlatform vp = universe.getViewingPlatform();
            TransformGroup vpGroup = vp.getMultiTransformGroup().getTransformGroup(0);
            Transform3D vpTranslation = new Transform3D();
            Vector3f translationVector = new Vector3f(0.0F, 0.0F, 6F);
            vpTranslation.setTranslation(translationVector);
            vpGroup.setTransform(vpTranslation);
        }

        private void addOtherLight() {
            Color3f directionalLightColor = new Color3f(Color.BLACK);
            Color3f ambientLightColor = new Color3f(Color.WHITE);
            Vector3f lightDirection = new Vector3f(-1F, -1F, -1F);

```



```

        AmbientLight ambientLight = new AmbientLight(ambientLightColor);
        DirectionalLight directionalLight = new
DirectionalLight(directionalLightColor, lightDirection);

        Bounds influenceRegion = new BoundingSphere();

        ambientLight.setInfluencingBounds(influenceRegion);
        directionalLight.setInfluencingBounds(influenceRegion);
        root.addChild(ambientLight);
        root.addChild(directionalLight);
    }

    public static Scene getSceneFromFile(String location) throws IOException {
        ObjectFile file = new ObjectFile(ObjectFile.RESIZE);
        file.setFlags(ObjectFile.RESIZE | ObjectFile.TRIANGULATE |
ObjectFile.STRIPIFY);
        return file.load(new FileReader(location));
    }

    // Not always works
    public static Scene getSceneFromLwoFile(String location) throws IOException {
        Lw3dLoader loader = new Lw3dLoader();
        return loader.load(new FileReader(location));
    }

    public static void main(String[] args) {
        try {
            FirstMainClass window = new FirstMainClass();
            AnimateCar carMovement = new AnimateCar(wholeCar, transform3D, window);
            window.addKeyListener(carMovement);
            window.setVisible(true);
        } catch (IOException ex) {
            System.out.println(ex.getMessage());
        }
    }
}

```

Результати роботи програми

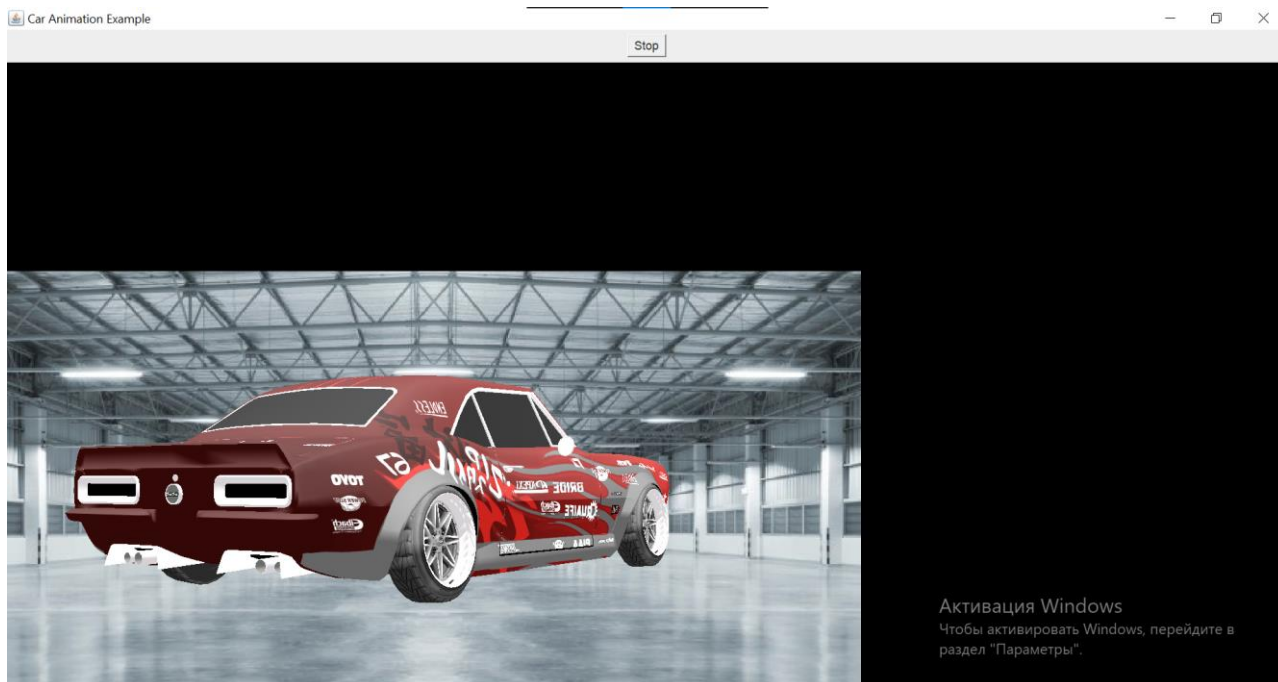


Рис.1. Автомобіль обертається

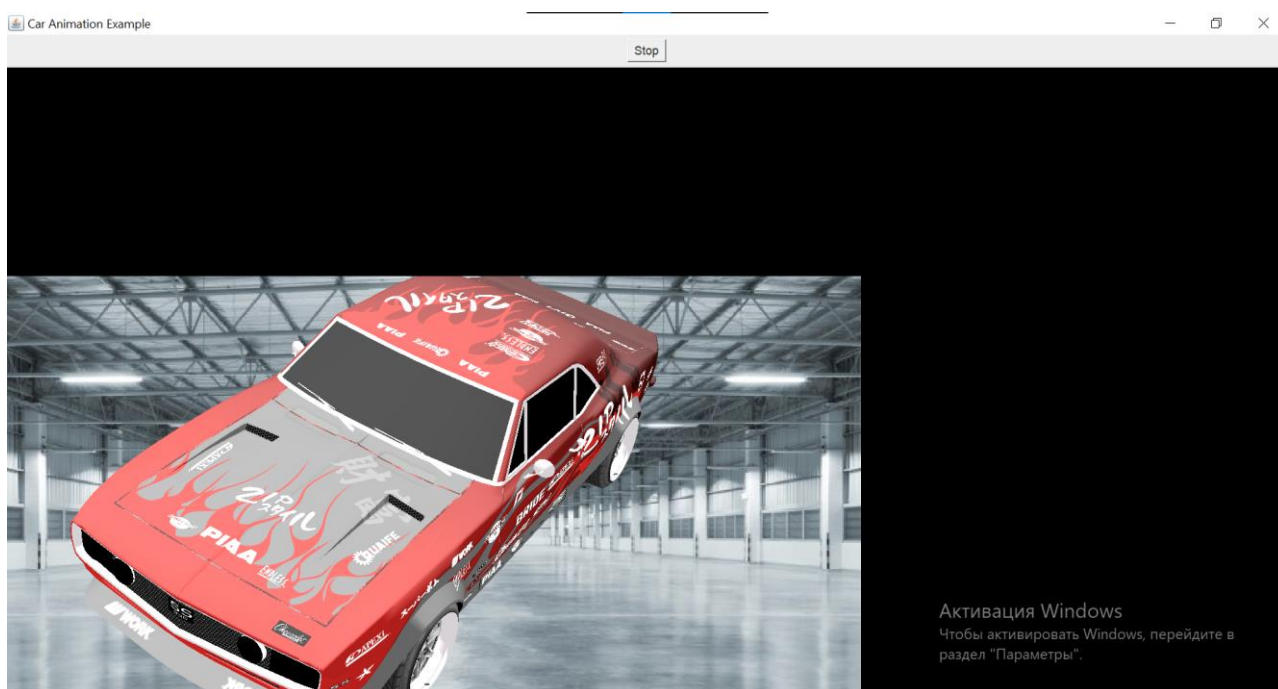


Рис.2. Автомобіль можна оглядати

Висновки

Виконавши дану лабораторну роботу я навчився імпортувати моделі, побудованих у тривимірних редакторах, (в даному разі Blender) до бібліотеки java3D та анімувати їх.