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”
на тему “Імпорт тривімирних моделей у середовище програмування java3D, обробка та маніапуляція цих зображень”

Виконав
студент ІІІ курсу
групи КП-82
Новохацький Владислав
(*прізвище, ім’я, по батькові*)

Зарахована
“____” “_____” 20___ р.
викладачем
Шкурат Оксаною Сергіївною
(*прізвище, ім’я, по батькові*)

варіант № 15

Київ 2021

Варіант завдання

Варіант 15:

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.

5. Анімувати рух автомобіля по трасі, повороти, збільшення, зменшення швидкості, зупинку.

Лістинг коду програми

```
package lab5;

import com.sun.j3d.loaders.Scene;
import com.sun.j3d.loaders.objectfile.ObjectFile;
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.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.io.FileReader;
import java.io.IOException;

public class Main extends JFrame implements ActionListener, KeyListener {
```

```
    private final static String carModelLocation =
"D:\\graphic\\lab5\\nissan-gt-r-nismo2.obj";
    private final static String backgroundLocation = "D:\\graphic\\lab5\\road.jpg";
    private final BranchGroup root = new BranchGroup();
    private final Canvas3D canvas = new
Canvas3D(SimpleUniverse.getPreferredConfiguration());
    private final TransformGroup carGroup = new TransformGroup();
    private final Transform3D transform3D = new Transform3D();
    private SimpleUniverse universe;
    private Scene car;
    private Background background;

    private float width = canvas.getWidth();
    private float x = width/2+0.3f;
    private float y = -.5f;
    private float z = 0;
    private float yk = 0.0017f;
    private float zk = 0.025f;
    private float xk = 0.0014f;
    private float a = 0;
    boolean start = true;

    private float scale = 0.5f;

    public static void main(String[] args) {
        try {
            var window = new Main();
            window.addKeyListener(window);
            window.setVisible(true);
        } catch (IOException e) {
            System.err.println(e.getMessage());
        }
    }

    public Main() throws IOException {
        init();
        addCar();
        setBackground();
        setLight();
        setPos();
        root.compile();
        universe.addBranchGraph(root);
    }

    private void setPos() {
        transform3D.setTranslation(new Vector3f(x, y, z));
        Transform3D transform2 = new Transform3D();
```

```
        transform2.rotY(Math.PI+Math.PI/36);
             transform3D.mul(transform2);
        transform3D.setScale(scale);
        carGroup.setTransform(transform3D);
    }

    private void init() throws IOException {
        setSize(1280, 760);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        canvas.setDoubleBufferEnable(true);
        getContentPane().add(canvas, BorderLayout.CENTER);

        universe = new SimpleUniverse(canvas);
        universe.getViewingPlatform().setNominalViewingTransform();

        canvas.addKeyListener(this);
        car = getScene();
    }

    private void setLight() {
       AmbientLight ambientLight = new AmbientLight(new Color3f(Color.WHITE));
       BoundingSphere influenceRegion = new BoundingSphere(new Point3d(), 1000);
       DirectionalLight directionalLight = new DirectionalLight(new
Color3f(Color.WHITE), new Vector3f(4.0f, -7.0f, -12.0f));

       directionalLight.setInfluencingBounds(new BoundingSphere(new Point3d(), 1000));
        ambientLight.setInfluencingBounds(influenceRegion);

        root.addChild(ambientLight);
        root.addChild(directionalLight);

    }

    private void setBackground() throws IOException {
       TextureLoader loader = new TextureLoader(backgroundLocation, canvas);
        background = new Background(loader.getImage());
        background.setImageScaleMode(Background.SCALE_FIT_MAX);
        background.setApplicationBounds(new BoundingSphere(new Point3d(),1000));
        background.setCapability(Background.ALLOW_IMAGE_WRITE);
        root.addChild(background);
    }

    private void addCar() throws IOException {
        carGroup.addChild(car.getSceneGroup());
        carGroup.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        root.addChild(carGroup);
```

```
}


private Scene getScene() throws IOException {
   ObjectFile loader = new ObjectFile(ObjectFile.RESIZE);
   loader.setBasePath("D:\\graphic\\lab5\\");
    return loader.load(new FileReader(carModelLocation));
}

@Override
public void keyPressed(KeyEvent e) {
    int keyCode = e.getKeyCode();
    switch (keyCode) {
        case KeyEvent.VK_UP: {
         if (start) {
                start = false;
                while (scale>0.05f) {
                       a += 0.1f;
                       y -= a*yk;
                          z -= a*zk;
                          x -= a*xk;
                       scale -= a*0.0017f;

                          transform3D.setTranslation(new Vector3f(x, y, z));
                          transform3D.setScale(scale);
                          carGroup.setTransform(transform3D);
                          try {
                                         Thread.sleep(10);
                                   } catch (InterruptedException e1) {
                                         e1.printStackTrace();
                                   }
                }
                   Transform3D transform3 = new Transform3D();
                   transform3.rotY(Math.PI-Math.PI/36);
                   transform3D.mul(transform3);
                   carGroup.setTransform(transform3D);
                while (a > 0){
                       a -= 0.1f;
                       y += a*yk;
                          z += a*zk;
                          x -= a*xk;
                          scale += a*0.0017f;

                          transform3D.setTranslation(new Vector3f(x, y, z));
                          transform3D.setScale(scale);
                          carGroup.setTransform(transform3D);
```

```
                try {
                                Thread.sleep(10);
                        } catch (InterruptedException e1) {
                                e1.printStackTrace();
                        }
        }
} else {
        start = true;
        while (a<7){
                a += 0.1f;
                y += a*yk;
                   z += a*zk;
                   x -= a*xk;
                   scale += a*0.0017f;

                   transform3D.setTranslation(new Vector3f(x, y, z));
                   transform3D.setScale(scale);
                   carGroup.setTransform(transform3D);
                   try {
                                Thread.sleep(10);
                        } catch (InterruptedException e1) {
                                e1.printStackTrace();
                        }
        }

        Transform3D transform3 = new Transform3D();
        a = 4f;
        x = width/2+0.3f+a*xk*40-800*xk*0.1f;
           y = -.5f+a*yk*40-800*yk*0.1f;
           z = zk*a*40-800*zk*0.1f;
           scale = 0.5f+a*0.0017f*40-800*0.0017f*0.1f;
           transform3D.setTranslation(new Vector3f(x, y, z));
           transform3D.setScale(scale);
           transform3.rotY(-Math.PI+Math.PI/36);
           transform3D.mul(transform3);
           carGroup.setTransform(transform3D);


           for (int i=0; i<40; i++) {
               a -= 0.1f;
               y -= a*yk;
                   z -= a*zk;
                   x -= a*xk;
               scale -= a*0.0017f;

                   transform3D.setTranslation(new Vector3f(x, y, z));
                   transform3D.setScale(scale);
```

```
                                carGroup.setTransform(transform3D);
                                try {
                                                Thread.sleep(10);
                                        } catch (InterruptedException e1) {
                                                e1.printStackTrace();
                                        }
                        }
                    }
                  }
              }
    }

    @Override
    public void keyReleased(KeyEvent e) { }

    @Override
    public void keyTyped(KeyEvent e) { }

    @Override
    public void actionPerformed(ActionEvent actionEvent) {

    }
}
```

Результат