



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”
тема “Імпорт тривимірних моделей у середовище програмування java3D,
обробка та маніпуляція цих зображень”

Виконав(ла)
студент(ка) III курсу
групи КП-83
Матіюк Дарина Андріївна

Зарахована
“ ____ ” “ _____ ” 20__ р.
Шкурат Оксаною Сергіївною

варіант №12

Київ 2021

Варіант завдання

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.

Варіант: кінь.

Лістинг коду програми

Main.java

```

public class Main extends JFrame {
    private static SimpleUniverse universe;
    private static Canvas3D canvas;
    private static BranchGroup root;
    private static TransformGroup horse;
    private static int TextureFlags = Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS;
    public static void main(String[] args) {
        try {
            Main window = new Main();
            Animation action = new Animation(horse);
            canvas.addKeyListener(action);
            window.setVisible(true);
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }
    public Main() throws IOException {
        configurations();
        root = new BranchGroup();
        addBackground();
        addLight();
        horse = getHorseGroup();
        root.addChild(horse);
        root.compile();
        universe.addBranchGraph(root);
    }
    private void configurations() throws IOException {
        //configure window
        setTitle("lab5");
        setSize(800, 600);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //configure canvas
        canvas = new Canvas3D(SimpleUniverse.getPreferredConfiguration());
        canvas.setDoubleBufferEnable(true);
        getContentPane().add(canvas, BorderLayout.CENTER);
        //configure universe
        universe = new SimpleUniverse(canvas);
        universe.getViewingPlatform().setNominalViewingTransform();
    }
    private void addBackground() {
        TextureLoader tl = new
TextureLoader("C:\\Users\\rocke\\IdeaProjects\\lab5\\assets\\back.jpg",
canvas);
        Background back = new Background(tl.getImage());
        back.setImageScaleMode(Background.SCALE_FIT_ALL);
        BoundingSphere bs = new BoundingSphere(new Point3d(0.0, 0.0, 0.0),
100.0);
        back.setApplicationBounds(bs);
        root.addChild(back);
    }
    private void addLight() {
        //directional light
        BoundingSphere bs = new BoundingSphere();
        bs.setRadius(100);
    }
}

```

```

        DirectionalLight dl = new DirectionalLight(new Color3f(1, 1, 1), new
Vector3f(-1, -1, -1));
        dl.setInfluencingBounds(bs);
        root.addChild(dl);
        //ambient light
        AmbientLight al = new AmbientLight(new Color3f(1, 1, 1));
        al.setInfluencingBounds(new BoundingSphere());
        root.addChild(al);
    }
    private void addTexture(Shape3D shape, String path) {
        TextureLoader tl = new TextureLoader(path, "RGP", new Container());
        Texture t = tl.getTexture();
        t.setBoundaryModeS(Texture.WRAP);
        t.setBoundaryModeT(Texture.WRAP);
        t.setBoundaryColor(new Color4f(0.0f, 1.0f, 0.0f, 0.0f));
        TextureAttributes ta = new TextureAttributes();
        ta.setTextureMode(TextureAttributes.MODULATE);
        Appearance a = new Appearance();
        a.setTexture(t);
        a.setTextureAttributes(ta);
        shape.setAppearance(a);
    }
    private TransformGroup getHorseGroup() throws IOException {
        ObjectFile of = new ObjectFile(ObjectFile.RESIZE);
        of.setFlags(ObjectFile.RESIZE | ObjectFile.TRIANGULATE |
ObjectFile.STRIPIFY);
        Scene s = of.load(new
FileReader("C:\\Users\\rocke\\IdeaProjects\\lab5\\assets\\horse.obj"));
        Map<String, Shape3D> map = s.getNamedObjects();
        Shape3D s3d = map.get("horse");
        s.getSceneGroup().removeChild(s3d);
        addTexture(s3d,
"C:\\Users\\rocke\\IdeaProjects\\lab5\\assets\\texture.jpg");
        Transform3D t3D = new Transform3D();
        t3D.setScale(new Vector3d(0.4, 0.4, 0.4));
        Transform3D rY = new Transform3D();
        rY.rotY(150);
        t3D.mul(rY);
        TransformGroup tg = new TransformGroup();
        tg.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        tg.addChild(s3d);
        tg.setTransform(t3D);
        return tg;
    }
}

```

Animation.java

```

public class Animation implements ActionListener, KeyListener {
    private Transform3D t3d = new Transform3D();
    private TransformGroup horse;
    private boolean w = false;
    private boolean a = false;
    private boolean s = false;
    private boolean d = false;
    private boolean e = false;
    private boolean q = false;
    private float x = 0;
    private float y = 0;
    Animation(TransformGroup horse) {
        this.horse = horse;
        this.horse.getTransform(this.t3d);

        Timer t = new Timer(40, this);
        t.start();
    }
    private void Action() {
        if (w) {
            y += 0.03f;
            if (y > 0.4f) {
                y = 0.4f;
            }
        }
        if (s) {
            y -= 0.03f;
            if (y < -0.4f) {
                y = -0.4f;
            }
        }
        if (a) {
            x -= 0.03f;
            if (x < -0.7f) {
                x = -0.7f;
            }
        }
        if (d) {
            x += 0.03f;
            if (x > 0.8f) {
                x = 0.8f;
            }
        }
        t3d.setTranslation(new Vector3f(x, y, 0));
        if (e) {
            Transform3D rotation = new Transform3D();
            rotation.rotY(0.05f);
            t3d.mul(rotation);
        }
        if (q) {
            Transform3D rotation = new Transform3D();
            rotation.rotY(-0.05f);
            t3d.mul(rotation);
        }
        horse.setTransform(t3d);
    }
}

```

```
@Override
public void actionPerformed(ActionEvent e) {
    Action();
}
@Override
public void keyPressed(KeyEvent ev) {
    switch (ev.getKeyChar()) {
        case 'w':
            w = true;
            break;
        case 's':
            s = true;
            break;
        case 'a':
            a = true;
            break;
        case 'd':
            d = true;
            break;
        case 'e':
            e = true;
            break;
        case 'q':
            q = true;
            break;
    }
}
@Override
public void keyTyped(KeyEvent e) { }
@Override
public void keyReleased(KeyEvent ev) {
    switch (ev.getKeyChar()) {
        case 'w':
            w = false;
            break;
        case 's':
            s = false;
            break;
        case 'a':
            a = false;
            break;
        case 'd':
            d = false;
            break;
        case 'e':
            e = false;
            break;
        case 'q':
            q = false;
            break;
    }
}
}
```

Результат



Рис. 1-2. Результати роботи програми.