



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 тривімірних моделей у середовище програмування java3D,  
обробка та маніпуляція цих зображень.”

Виконав(ла)  
студент(ка) III курсу  
групи КП-83  
Симонюк Володимир Павлович  
варіант №18

Зарахована  
“ \_\_\_\_ ” “ \_\_\_\_ ” 20 \_\_\_\_ р.  
Шкурат Оксаною Сергіївною

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

### Лістинг коду програми

#### Main.java

```
public class Main extends JFrame {
    private static Canvas3D canvas;
    private static SimpleUniverse universe;
    private static BranchGroup root;
    private static TransformGroup bear;
    private static int TextureFlags = Primitive.GENERATE_NORMALS +
Primitive.GENERATE_TEXTURE_COORDS;

    public Main() throws IOException {
        configureWindow();
        configureCanvas();
        configureUniverse();
        root = new BranchGroup();
        addImageBackground();
        addDirectionalLightToUniverse();
        addAmbientLightToUniverse();
        bear = getBearGroup();
        root.addChild(bear);
        root.compile();
        universe.addBranchGraph(root);
    }

    private void configureWindow() {
        setTitle("Lab5");
        setExtendedState(JFrame.MAXIMIZED_BOTH);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    private void configureCanvas() {
        canvas = new Canvas3D(SimpleUniverse.getPreferredConfiguration());
        canvas.setDoubleBufferEnable(true);
        getContentPane().add(canvas, BorderLayout.CENTER);
    }

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

    private void addImageBackground() {
        TextureLoader t = new
TextureLoader("C:/Users/izogi/IdeaProjects/lab5/src/assets/WOODS.jpg", 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 addDirectionalLightToUniverse() {
        BoundingSphere bounds = new BoundingSphere();
        bounds.setRadius(100);
        DirectionalLight light = new DirectionalLight(new Color3f(1, 1, 1), new
Vector3f(-1, -1, -1));
        light.setInfluencingBounds(bounds);
        root.addChild(light);
    }

    private void addAmbientLightToUniverse() {
        AmbientLight light = new AmbientLight(new Color3f(1, 1, 1));
        light.setInfluencingBounds(new BoundingSphere());
        root.addChild(light);
    }

    private TransformGroup getBearGroup() throws IOException {
        Shape3D shape = getModelShape3D("bear",
"C:/Users/izogi/IdeaProjects/lab5/src/assets/MODEL.obj");
        Transform3D transform3D = new Transform3D();
        transform3D.setScale(new Vector3d(0.4, 0.4, 0.4));
        Transform3D rotationY = new Transform3D();
        rotationY.rotY(Math.PI / 4);
        transform3D.mul(rotationY);
        TransformGroup group = getModelGroup(shape);
        group.setTransform(transform3D);
        return group;
    }

    private TransformGroup getModelGroup(Shape3D shape) {
        TransformGroup group = new TransformGroup();
        group.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
        group.addChild(shape);
        return group;
    }

    private Shape3D getModelShape3D(String name, String path) throws IOException {
        Scene scene = getSceneFromFile(path);
        Map<String, Shape3D> map = scene.getNamedObjects();
        Shape3D shape = map.get(name);
        scene.getSceneGroup().removeChild(shape);
        return shape;
    }

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

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

```

## Animation.java

```
public class Animation implements ActionListener, KeyListener {
    private TransformGroup bear;
    private Transform3D transform3D = new Transform3D();

    private float x = 0;
    private float y = 0;

    private boolean w = false;
    private boolean s = false;
    private boolean a = false;
    private boolean d = false;
    private boolean e = false;
    private boolean q = false;

    Animation(TransformGroup bear) {
        this.bear = bear;
        this.bear.getTransform(this.transform3D);
        Timer timer = new Timer(20, this);
        timer.start();
    }

    private void Move() {
        if (w) {
            y += 0.02f;
            if (y > 0.2f) y = 0.2f;
        }

        if (s) {
            y -= 0.02f;
            if (y < -0.3f) y = -0.3f;
        }

        if (a) {
            x -= 0.02f;
            if (x < -0.8f) x = -0.8f;
        }

        if (d) {
            x += 0.02f;
            if (x > 0.8f) x = 0.8f;
        }

        transform3D.setTranslation(new Vector3f(x, y, 0));

        if (e) {
            Transform3D rotation = new Transform3D();
            rotation.rotY(0.05f);
            transform3D.mul(rotation);
        }

        if (q) {
            Transform3D rotation = new Transform3D();
            rotation.rotY(-0.05f);
            transform3D.mul(rotation);
        }
        bear.setTransform(transform3D);
    }
}
```

```
@Override
public void actionPerformed(ActionEvent e) {
    Move();
}

@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

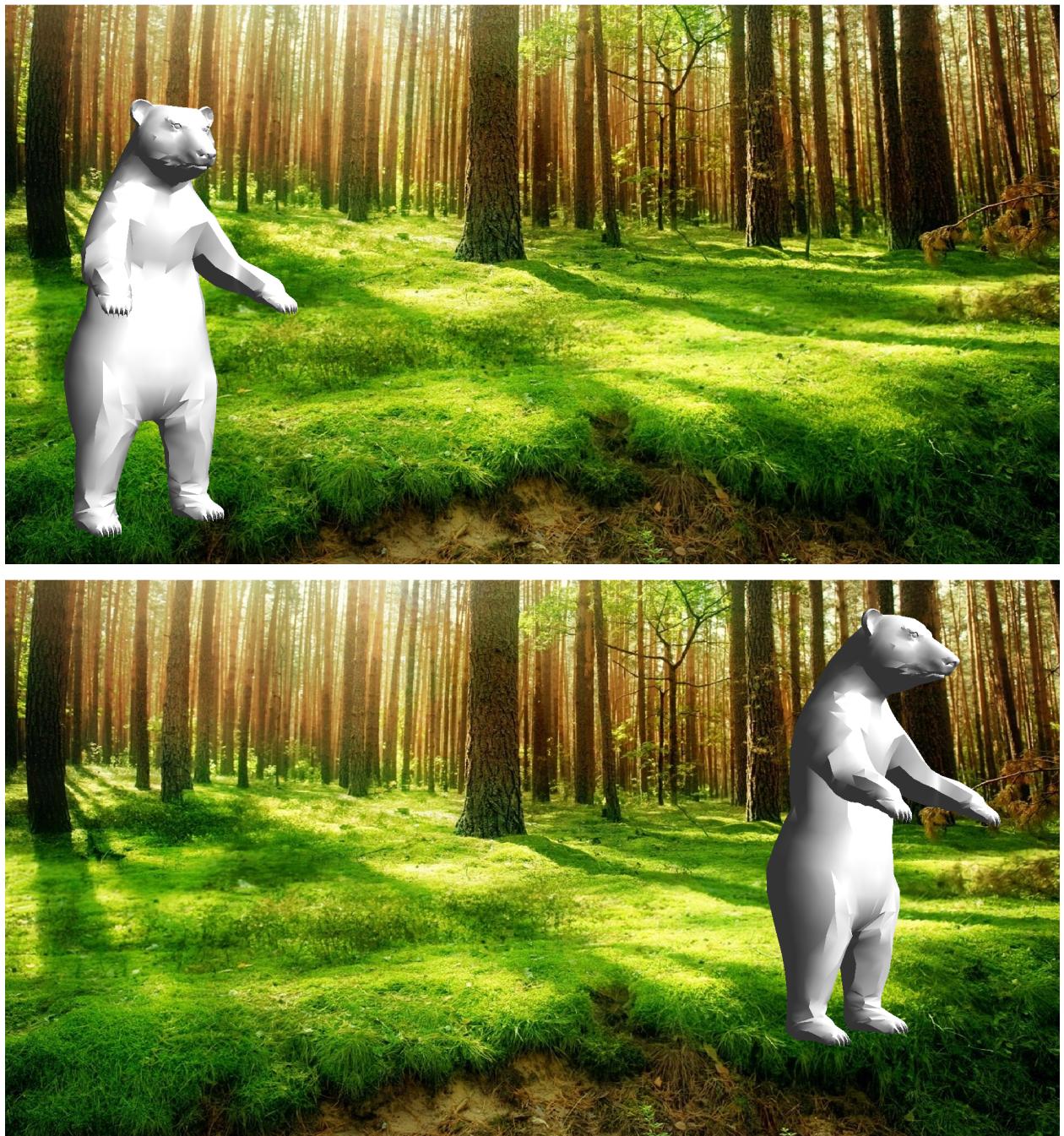


Рис. 1-2. Результати роботи програми