



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 №3
з дисципліни “Математичні та алгоритмічні основи комп’ютерної графіки”
тема: “Структура файлів формату .bmp. Анімація примітивів за допомогою
засобів бібліотеки JavaFX”

Виконав
студент III курсу
групи КП-83
Ландо Максим Юрійович
варіант № 9

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

Київ 2021

Мета:

- 1) вивчення структури та особливостей використання файлів формату .bmp;
- 2) 2) вивчення стандартних засобів JavaFX для візуалізації зображення;
- 3) вивчення засобів анімації примітивів в JavaFX

Задання на лабораторну роботу

За допомогою примітивів JavaFX максимально реально зобразити персонажа за варіантом та виконати його 2D анімацію. Для анімації скористатися стандартними засобами бібліотеки JavaFX.

Обов'язковою є реалізація таких видів анімації:

- 1) переміщення;
- 2) поворот;
- 3) масштабування.

Варіант:

Лістинг коду програми

Main.java

```
package sample;

import javafx.animation.*;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.transform.Rotate;
import javafx.stage.Stage;
import javafx.scene.shape.*;
import javafx.scene.paint.Color;
import javafx.util.Duration;

public class Main extends Application {

    int width = 1000;
    int height = 700;
    int cx = width / 2;
    int cy = height / 2;

    @Override
    public void start(Stage primaryStage) throws Exception{

        Group gr = new Group();
        primaryStage.setScene(new Scene(gr, width, height));

        drawElephant(gr);

        TranslateTransition translateTransition = new TranslateTransition(Duration.millis(2000), gr);
        translateTransition.setFromX(50);
        translateTransition.setToX(350);
        translateTransition.setCycleCount(2);
        translateTransition.setAutoReverse(true);
        translateTransition.play();

        ScaleTransition scaleTransition = new ScaleTransition(Duration.millis(2000), gr);
        scaleTransition.setToX(320);
        scaleTransition.setToY(320);
        scaleTransition.setFromX(0.75);
        scaleTransition.setFromY(0.75);
        scaleTransition.setToX(1.5);
        scaleTransition.setToY(1.5);
        scaleTransition.setCycleCount(2);
        scaleTransition.setAutoReverse(true);
        scaleTransition.play();

        RotateTransition rotForArc1 =
            new RotateTransition(Duration.millis(500), gr);
        rotForArc1.setByAngle(20f);
        rotForArc1.setCycleCount(20);
        rotForArc1.setAutoReverse(true);
        rotForArc1.play();

        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }

    public void drawElephant(Group root) {
        Color elephantColor = Color.rgb(255,112,147);
        Color feetColor = Color.rgb(255,182,193);
        Color innerEarColor = Color.rgb(255,228,225);
        Color mouthColor = Color.rgb(201, 36, 35);

        Circle head = new Circle(cx, cy - 70, 50, elephantColor);
        head.setStroke(Color.BLACK);
        head.setStrokeWidth(1.5);

        int cxLeftEar = cx - 40;
        int cxRightEar = cx + 40;
        int cyEar = cy - 80;
        int earWidth = 30;
```

```

int earHeight = 60;
int innerEarWidth = 20;
int innerEarHeight = 40;

Ellipse leftEar = new Ellipse(cxLeftEar, cyEar, earWidth, earHeight);
leftEar.setFill(elephantColor);
leftEar.setStroke(Color.BLACK);
leftEar.setStrokeWidth(1.5);
Ellipse rightEar = new Ellipse(cxRightEar, cyEar, earWidth, earHeight);
rightEar.setFill(elephantColor);
rightEar.setStroke(Color.BLACK);
rightEar.setStrokeWidth(1.5);

Ellipse innerLeftEar = new Ellipse(cxLeftEar, cyEar, innerEarWidth, innerEarHeight);
innerLeftEar.setFill(innerEarColor);
Ellipse innerRightEar = new Ellipse(cxRightEar, cyEar, innerEarWidth, innerEarHeight);
innerRightEar.setFill(innerEarColor);

Ellipse body = new Ellipse(cx, cy, 60, 70);
body.setFill(elephantColor);
body.setStroke(Color.BLACK);
body.setStrokeWidth(1.5);

int xLeftArmDown = cx - 60;
int yLeftArmDown = cy;
QuadCurve leftArmDown = new QuadCurve();
leftArmDown.setStartX(xLeftArmDown);
leftArmDown.setStartY(yLeftArmDown);
leftArmDown.setEndX(xLeftArmDown + 25);
leftArmDown.setEndY(yLeftArmDown + 10);
leftArmDown.setControlX(xLeftArmDown + 5);
leftArmDown.setControlY(yLeftArmDown + 7);
leftArmDown.setFill(elephantColor);
leftArmDown.setStroke(Color.BLACK);
leftArmDown.setStrokeWidth(1.5);

int xLeftArmUp = xLeftArmDown + 15;
int yLeftArmUp = yLeftArmDown - 20;
QuadCurve leftArmUp = new QuadCurve();
leftArmUp.setStartX(xLeftArmUp);
leftArmUp.setStartY(yLeftArmUp);
leftArmUp.setEndX(xLeftArmUp + 10);
leftArmUp.setEndY(yLeftArmUp + 5);
leftArmUp.setControlX(xLeftArmUp + 8);
leftArmUp.setControlY(yLeftArmUp + 4);
leftArmUp.setFill(elephantColor);
leftArmUp.setStroke(Color.BLACK);
leftArmUp.setStrokeWidth(1.5);

int xMidArmStart = xLeftArmUp + 10;
int yMidArmStart = yLeftArmUp + 5;
int xMidArmEnd = xLeftArmDown + 25;
int yMidArmEnd = yLeftArmDown + 10;
QuadCurve armMiddle = new QuadCurve();
armMiddle.setStartX(xMidArmStart);
armMiddle.setStartY(yMidArmStart);
armMiddle.setEndX(xMidArmEnd);
armMiddle.setEndY(yMidArmEnd);
armMiddle.setControlX(xMidArmStart + 15);
armMiddle.setControlY(yMidArmStart + 15);
armMiddle.setFill(elephantColor);
armMiddle.setStroke(Color.BLACK);
armMiddle.setStrokeWidth(1.5);

int xRightArmDown = cx;
int yRightArmDown = cy;
QuadCurve rightArmDown = new QuadCurve();
rightArmDown.setStartX(xRightArmDown);
rightArmDown.setStartY(yRightArmDown);
rightArmDown.setEndX(xRightArmDown - 32);
rightArmDown.setEndY(yRightArmDown + 10);
rightArmDown.setControlX(xRightArmDown - 5);
rightArmDown.setControlY(yRightArmDown + 7);
rightArmDown.setFill(elephantColor);
rightArmDown.setStroke(Color.BLACK);
rightArmDown.setStrokeWidth(1.5);

```

```

int xRightArmUp = xRightArmDown - 15;
int yRightArmUp = yRightArmDown - 15;
QuadCurve rightArmUp = new QuadCurve();
rightArmUp.setStartX(xRightArmUp);
rightArmUp.setStartY(yRightArmUp);
rightArmUp.setEndX(xRightArmUp - 15);
rightArmUp.setEndY(yRightArmUp + 5);
rightArmUp.setControlX(xRightArmUp - 8);
rightArmUp.setControlY(yRightArmUp + 4);
rightArmUp.setFill(elephantColor);
rightArmUp.setStroke(Color.BLACK);
rightArmUp.setStrokeWidth(1.5);

int eyesY = cy - 90;
int deltaXEyes = 10;
Circle circleEye1 = new Circle(cx - deltaXEyes, eyesY, 8, Color.WHITE);
Circle circleInner1 = new Circle(cx - deltaXEyes, eyesY + 3, 4, Color.BLACK);
Circle circleEye2 = new Circle(cx + deltaXEyes, eyesY, 8, Color.WHITE);
Circle circleInner2 = new Circle(cx + deltaXEyes, eyesY + 3, 4, Color.BLACK);

int browY = eyesY - 10;
QuadCurve leftEyebrow = new QuadCurve();
leftEyebrow.setStartX(cx - 15);
leftEyebrow.setStartY(browY);
leftEyebrow.setEndX(cx - 5);
leftEyebrow.setEndY(browY);
leftEyebrow.setControlX(cx - 10);
leftEyebrow.setControlY(browY - 5);
leftEyebrow.setFill(elephantColor);
leftEyebrow.setStroke(Color.BLACK);
leftEyebrow.setStrokeWidth(1.5);

QuadCurve rightEyebrow = new QuadCurve();
rightEyebrow.setStartX(cx + 5);
rightEyebrow.setStartY(browY);
rightEyebrow.setEndX(cx + 15);
rightEyebrow.setEndY(browY);
rightEyebrow.setControlX(cx + 10);
rightEyebrow.setControlY(browY - 5);
rightEyebrow.setFill(elephantColor);
rightEyebrow.setStroke(Color.BLACK);
rightEyebrow.setStrokeWidth(1.5);

QuadCurve mouth = new QuadCurve();
mouth.setStartX(cx - 20);
mouth.setStartY(cy - 40);
mouth.setEndX(cx + 20);
mouth.setEndY(cy - 40);
mouth.setControlX(cx);
mouth.setControlY(cy - 15);
mouth.setFill(mouthColor);
mouth.setStroke(Color.BLACK);
mouth.setStrokeWidth(1.5);

Ellipse leftLeg = new Ellipse(cx - 25, cy + 65, 20, 15);
leftLeg.setFill(elephantColor);
leftLeg.setStroke(Color.BLACK);
leftLeg.setStrokeWidth(1.5);

Ellipse rightLeg = new Ellipse(cx + 25, cy + 65, 20, 15);
rightLeg.setFill(elephantColor);
rightLeg.setStroke(Color.BLACK);
rightLeg.setStrokeWidth(1.5);

Ellipse leftFeet = new Ellipse(cx - 45, cy + 65, 20, 15);
leftFeet.setFill(feetColor);
leftFeet.setStroke(Color.BLACK);
leftFeet.setStrokeWidth(1.5);

Ellipse rightFeet = new Ellipse(cx + 5, cy + 65, 20, 15);
rightFeet.setFill(feetColor);
rightFeet.setStroke(Color.BLACK);
rightFeet.setStrokeWidth(1.5);

int xPathStart = cx - 25;
int yPathStart = cy - 110;

```

```

Path path = new Path();
MoveTo moveTo = new MoveTo();
moveTo.setX(xPathStart);
moveTo.setY(yPathStart);

int xQ1 = xPathStart + 60;
int yQ1 = yPathStart + 50;
QuadCurveTo quad1 = new QuadCurveTo();
quad1.setControlX(xPathStart + 20);
quad1.setControlY(yPathStart - 10);
quad1.setX(xQ1);
quad1.setY(yQ1);

int xQ2 = xQ1 + 60;
int yQ2 = yQ1 + 50;
QuadCurveTo quad2 = new QuadCurveTo();
quad2.setControlX(xQ1 + 20);
quad2.setControlY(yQ1 + 40);
quad2.setX(xQ2);
quad2.setY(yQ2);

int xQ3 = xQ2;
int yQ3 = yQ2 + 30;
QuadCurveTo quad3 = new QuadCurveTo();
quad3.setControlX(xQ2 + 30);
quad3.setControlY(yQ2 + 15);
quad3.setX(xQ3);
quad3.setY(yQ3);

int xQ4 = xQ3 - 60;
int yQ4 = yQ3 - 10;
QuadCurveTo quad4 = new QuadCurveTo();
quad4.setControlX(xQ3 - 30);
quad4.setControlY(yQ3);
quad4.setX(xQ4);
quad4.setY(yQ4);

int xQ5 = xQ4 - 40;
int yQ5 = yQ4 - 30;
QuadCurveTo quad5 = new QuadCurveTo();
quad5.setControlX(xQ4 - 20);
quad5.setControlY(yQ4 - 10);
quad5.setX(xQ5);
quad5.setY(yQ5);

int xQ6 = xQ5 - 40;
int yQ6 = yQ5 - 20;
QuadCurveTo quad6 = new QuadCurveTo();
quad6.setControlX(xQ5 - 10);
quad6.setControlY(yQ5 - 10);
quad6.setX(xQ6);
quad6.setY(yQ6);

QuadCurveTo chain = new QuadCurveTo();
chain.setControlX(xPathStart - 30);
chain.setControlY(yPathStart + 15);
chain.setX(xPathStart);
chain.setY(yPathStart);

path.getElements().add(moveTo);
path.getElements().add(quad1);
path.getElements().add(quad2);
path.getElements().add(quad3);
path.getElements().add(quad4);
path.getElements().add(quad5);
path.getElements().add(quad6);
path.getElements().add(chain);

path.setFill(elephantColor);
path.setStrokeWidth(1.5f);
path.setScaleX(0.55);
path.setScaleY(0.55);

root.getChildren().add(leftEar);
root.getChildren().add(rightEar);

root.getChildren().add(innerLeftEar);

```

```
        root.getChildren().add(innerRightEar);

        root.getChildren().add(body);

        root.getChildren().add(leftArmUp);
        root.getChildren().add(leftArmDown);
        root.getChildren().add(armMiddle);
        root.getChildren().add(rightArmDown);
        root.getChildren().add(rightArmUp);

        root.getChildren().add(head);
        root.getChildren().add(mouth);
        root.getChildren().add(path);
        root.getChildren().add(leftLeg);
        root.getChildren().add(rightLeg);

        root.getChildren().add(leftEyebrow);
        root.getChildren().add(rightEyebrow);

        root.getChildren().add(circleEye1);
        root.getChildren().add(circleInner1);
        root.getChildren().add(circleEye2);
        root.getChildren().add(circleInner2);

        root.getChildren().add(leftFeet);
        root.getChildren().add(rightFeet);
    }
}
```

Результат

