



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”

Виконав
студент III курсу
групи КП-83

Палій Дмитро Володимирович
(прізвище, ім'я, по батькові)

варіант № 16

Зарахована
“ ____ ” “ _____ ” 20__ р.
викладачем

Шкурат Оксаною Сергіївною
(прізвище, ім'я, по батькові)

Варіант завдання

Завдання: 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.

Студентам пропонується скористатися розглянутими класами для читання, обробки та збереження зображень формату .bmp з метою використання рисунку для створення траєкторії руху або меж, в яких дозволений рух об'єктів. В даному випадку рекомендується використовувати кольори великої контрастності для різних призначень (наприклад, чорний колір відповідатиме за траєкторію руху, а інші кольори – заборонятимуть рух).

Варіант:

16



Лістинг коду програми

```
package sample;
import javafx.animation.*;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.StackPane;
import javafx.scene.shape.*;
import javafx.stage.Stage;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.image.PixelReader;
import javafx.scene.paint.Color;
import javafx.scene.image.PixelWriter;
import javafx.scene.image.WritableImage;
import javafx.util.Duration;

public class ReadingFromFileBMPWithStandart extends Application {

    @Override
    public void start(Stage primaryStage) {
        Image image = new
Image("file:C:\\Users\\dpali\\Documents\\маокр\\lab3\\picture.bmp");

        PixelReader pixelReader = image.getPixelReader();
        WritableImage wImage = new WritableImage((int)image.getWidth(),
(int)image.getHeight());
        PixelWriter pixelWriter = wImage.getPixelWriter();

        for (int readY = 0; readY < image.getHeight(); readY++) {
            for (int readX = 0; readX < image.getWidth(); readX++) {
                Color color = pixelReader.getColor(readX, readY);
                if (color.getBlue() != 1 && color.getGreen() != 1 &&
color.getRed() != 1) {
                    pixelWriter.setColor(readX, readY, color);
                }
            }
        }

        ImageView imageView = new ImageView();
        imageView.setImage(wImage);

        StackPane root = new StackPane();
        root.getChildren().add(imageView);
        // animation
        final Duration SEC_2 = Duration.millis(2000);
        final Duration SEC_3 = Duration.millis(3000);

        Path path = new Path();
        path.getElements().add(new MoveTo(20,20));
        path.getElements().add(new LineTo(20,image.getHeight()));
        path.getElements().add(new
LineTo(image.getWidth(),image.getHeight()));
        path.getElements().add(new LineTo(image.getWidth(),20));
        path.getElements().add(new LineTo(20,20));

        PathTransition pt = new PathTransition();
        pt.setDuration(Duration.millis(4000));
        pt.setPath(path);
        pt.setNode(imageView);

        pt.setOrientation(PathTransition.OrientationType.ORTHOGONAL_TO_TANGENT);
        pt.setCycleCount(Timeline.INDEFINITE);
        pt.setAutoReverse(true);
    }
}
```

```

RotateTransition rt = new RotateTransition(SEC_3);
rt.setByAngle(180f);
rt.setCycleCount(Animation.INDEFINITE);
rt.setAutoReverse(true);

ScaleTransition st = new ScaleTransition(SEC_2);
st.setByX(-0.5f);
st.setByY(-0.5f);
st.setCycleCount(Animation.INDEFINITE);
st.setAutoReverse(true);

ParallelTransition animation = new ParallelTransition(imageView, pt,
rt, st);
animation.play();
// animation end
Scene scene = new Scene(root, 2 * image.getWidth(), 2 *
image.getHeight());
scene.setFill(Color.BLACK);
primaryStage.setTitle("lab3");
primaryStage.setScene(scene);
primaryStage.show();
}

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

```

Результат



