

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  Василець Данило Андрійович  (*прізвище, ім’я, по батькові*)  варіант № 2 |  |  | Зарахована  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладачем  Шкурат Оксаною Сергіївною (*прізвище, ім’я, по батькові*) |

Київ 2021

**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.

**Варіант:**



**Лістинг коду програми**

**public** **class** Main **extends** Application {

    Color background = Color.WHITE;

**int** width = 700;

**int** height = 700;

**int** cx = width / 2;

**int** cy = height / 2;

**public** **static** **void** main(String[] args) {

        launch(args);

    }

    @Override

**public** **void** start(Stage primaryStage) {

        Group root = **new** Group();

        Scene scene = **new** Scene(root, width, height);

        scene.setFill(background);

*//build*

        buildChristmasTree(root);

*//*

**int** time = 3000;

*//*

        RotateTransition rotate = **new** RotateTransition(Duration.millis(time), root);

        rotate.setByAngle(-360f);

        rotate.setCycleCount(Timeline.INDEFINITE);

*//*

        ScaleTransition scaleFrom = **new** ScaleTransition(Duration.millis(time), root);

        scaleFrom.setToX(1);

        scaleFrom.setToY(1);

        ScaleTransition scaleTo = **new** ScaleTransition(Duration.millis(time), root);

        scaleTo.setToX(0.3);

        scaleTo.setToY(0.3);

        SequentialTransition scale = **new** SequentialTransition();

        scale.getChildren().addAll(

                scaleTo,

                scaleFrom

        );

        scale.setCycleCount(Timeline.INDEFINITE);

*//*

        TranslateTransition translateTo = **new** TranslateTransition(Duration.millis(time), root);

        translateTo.setFromY(-150);

        translateTo.setToY(250);

        translateTo.setFromX(-150);

        translateTo.setToX(250);

        translateTo.setCycleCount(Timeline.INDEFINITE);

        translateTo.setAutoReverse(**true**);

        TranslateTransition translateFrom = **new** TranslateTransition(Duration.millis(time), root);

        translateFrom.setFromY(-150);

        translateFrom.setToY(250);

        translateTo.setFromX(-150);

        translateTo.setToX(250);

        translateFrom.setCycleCount(Timeline.INDEFINITE);

        translateFrom.setAutoReverse(**true**);

        SequentialTransition translate = **new** SequentialTransition();

        translate.getChildren().addAll(

                translateTo,

                translateFrom

        );

        translate.setCycleCount(Timeline.INDEFINITE);

        ParallelTransition animation = **new** ParallelTransition();

        animation.getChildren().addAll(

                rotate,

                scale,

                translate

        );

*//*

        animation.play();

*//*

        primaryStage.setTitle("Christmas Tree");

        primaryStage.setScene(scene);

        primaryStage.show();

    }

**public** **void** buildChristmasTree(Group root) {

*/////COLOURS*

        Stop[] green\_stops = **new** Stop[]{**new** Stop(0, Color.LIGHTGREEN), **new** Stop(0.3, Color.DARKGREEN)};

        Stop[] brown\_stops = **new** Stop[]{**new** Stop(0, Color.rgb(151, 117, 112)), **new** Stop(0.4, Color.BROWN)};

        Stop[] red\_stops = **new** Stop[]{**new** Stop(0, Color.rgb(255, 183, 182)), **new** Stop(0.4, Color.RED)};

        Stop[] blue\_stops = **new** Stop[]{**new** Stop(0, Color.LIGHTBLUE), **new** Stop(0.4, Color.BLUE)};

        Stop[] yellow\_stops = **new** Stop[]{**new** Stop(0, Color.rgb(255, 255, 220)), **new** Stop(0.4, Color.YELLOW)};

        Stop[] pink\_stops = **new** Stop[]{**new** Stop(0, Color.LIGHTPINK), **new** Stop(0.4, Color.PINK)};

        Stop[] orange\_stops = **new** Stop[]{**new** Stop(0, Color.rgb(255, 217, 182)), **new** Stop(0.4, Color.ORANGE)};

        Stop[] magenta\_stops = **new** Stop[]{**new** Stop(0, Color.MAGENTA), **new** Stop(0.4, Color.DARKMAGENTA)};

        LinearGradient green\_linear = **new** LinearGradient(0, 0, 1, 0, **true**, CycleMethod.NO\_CYCLE, green\_stops);

        LinearGradient brown\_linear = **new** LinearGradient(0, 0, 1, 0, **true**, CycleMethod.NO\_CYCLE, brown\_stops);

        LinearGradient red\_linear = **new** LinearGradient(0, 0, 1, 0, **true**, CycleMethod.NO\_CYCLE, red\_stops);

        LinearGradient blue\_linear = **new** LinearGradient(0, 0, 1, 0, **true**, CycleMethod.NO\_CYCLE, blue\_stops);

        LinearGradient yellow\_linear = **new** LinearGradient(0, 0, 1, 0, **true**, CycleMethod.NO\_CYCLE, yellow\_stops);

        LinearGradient pink\_linear = **new** LinearGradient(0, 0, 1, 0, **true**, CycleMethod.NO\_CYCLE, pink\_stops);

        LinearGradient orange\_linear = **new** LinearGradient(0, 0, 1, 0, **true**, CycleMethod.NO\_CYCLE, orange\_stops);

        LinearGradient magenta\_linear = **new** LinearGradient(0, 0, 1, 0, **true**, CycleMethod.NO\_CYCLE, magenta\_stops);

*/////SHAPES*

        Ellipse floor1 = **new** Ellipse(cx, cy + 150, 60, 30);

        Ellipse floor2 = **new** Ellipse(cx, cy + 145, 60, 30);

        Ellipse foot\_floor = **new** Ellipse(cx, cy + 143, 15, 5);

        Rectangle foot = **new** Rectangle(cx - 15, cy + 43, 30, 100);

        Rectangle mash = **new** Rectangle(cx - 14, cy + 44, 29, 101);

        Polygon triangle\_L = **new** Polygon(

                cx, cy - 40,

                cx + 150, cy + 50,

                cx + 130, cy + 70,

                cx + 100, cy + 90,

                cx + 50, cy + 105,

                cx, cy + 110,

                cx - 50, cy + 105,

                cx - 100, cy + 90,

                cx - 130, cy + 70,

                cx - 150, cy + 50,

                cx, cy - 40

        );

        Polygon triangle\_M = **new** Polygon(

                cx, cy - 100,

                cx + 130, cy,

                cx + 115, cy + 12,

                cx + 80, cy + 30,

                cx + 40, cy + 45,

                cx, cy + 50,

                cx - 40, cy + 45,

                cx - 80, cy + 30,

                cx - 115, cy + 12,

                cx - 130, cy,

                cx, cy - 100

        );

        Polygon triangle\_S = **new** Polygon(

                cx, cy - 150 - 10,

                cx + 30, cy - 100 - 10,

                cx + 60, cy - 75 - 10,

                cx + 110, cy - 50 - 10,

                cx + 95, cy - 38 - 10,

                cx + 73, cy - 20 - 10,

                cx + 37, cy - 5 - 10,

                cx, cy - 10,

                cx - 37, cy - 5 - 10,

                cx - 73, cy - 20 - 10,

                cx - 95, cy - 38 - 10,

                cx - 110, cy - 50 - 10,

                cx - 60, cy - 75 - 10,

                cx - 30, cy - 100 - 10,

                cx, cy - 150 - 10

        );

        Polygon star = **new** Polygon(

                cx - 20, cy - 160 + 17 - 25,

                cx + 15 - 20, cy - 160 + 15 - 25,

                cx + 20 - 20, cy - 160 + 2 - 25,

                cx + 25 - 20, cy - 160 + 15 - 25,

                cx + 40 - 20, cy - 160 + 17 - 25,

                cx + 30 - 20, cy - 160 + 25 - 25,

                cx + 32 - 20, cy - 160 + 38 - 25,

                cx + 20 - 20, cy - 160 + 30 - 25,

                cx + 8 - 20, cy - 160 + 38 - 25,

                cx + 10 - 20, cy - 160 + 25 - 25,

                cx - 20, cy - 160 + 17 - 25);

        Circle toy1 = **new** Circle(cx, cy, 12, Color.RED);

        Circle toy2 = **new** Circle(cx - 40, cy - 50, 12, blue\_linear);

        Circle toy3 = **new** Circle(cx + 30, cy - 90, 12, yellow\_linear);

        Circle toy4 = **new** Circle(cx - 40, cy + 70, 12, orange\_linear);

        Circle toy5 = **new** Circle(cx + 50, cy - 60, 12, red\_linear);

        Circle toy6 = **new** Circle(cx - 60, cy + 5, 12, yellow\_linear);

        Circle toy7 = **new** Circle(cx + 50, cy + 40, 12, magenta\_linear);

        Circle toy8 = **new** Circle(cx - 80, cy + 75, 12, red\_linear);

        Circle toy9 = **new** Circle(cx + 90, cy - 20, 12, magenta\_linear);

        Circle toy10 = **new** Circle(cx - 60, cy + 30, 12, pink\_linear);

        Circle toy11 = **new** Circle(cx + 55, cy + 70, 12, yellow\_linear);

        Circle toy12 = **new** Circle(cx - 90, cy, 12, pink\_linear);

        Circle toy13 = **new** Circle(cx + 78, cy + 30, 12, pink\_linear);

        Circle toy14 = **new** Circle(cx - 20, cy - 100, 12, yellow\_linear);

        Circle toy15 = **new** Circle(cx + 14, cy + 50, 12, pink\_linear);

*/////STROKES*

        star.setStroke(Color.BLACK);

        toy1.setStroke(Color.BLACK);

        toy2.setStroke(Color.BLACK);

        toy3.setStroke(Color.BLACK);

        toy4.setStroke(Color.BLACK);

        toy5.setStroke(Color.BLACK);

        toy6.setStroke(Color.BLACK);

        toy7.setStroke(Color.BLACK);

        toy8.setStroke(Color.BLACK);

        toy9.setStroke(Color.BLACK);

        toy10.setStroke(Color.BLACK);

        toy11.setStroke(Color.BLACK);

        toy12.setStroke(Color.BLACK);

        toy13.setStroke(Color.BLACK);

        toy14.setStroke(Color.BLACK);

        toy15.setStroke(Color.BLACK);

        floor1.setStroke(Color.BLACK);

        floor2.setStroke(Color.BLACK);

        foot\_floor.setStroke(Color.BLACK);

        foot.setStroke(Color.BLACK);

        triangle\_L.setStroke(Color.BLACK);

        triangle\_M.setStroke(Color.BLACK);

        triangle\_S.setStroke(Color.BLACK);

*/////FILLS*

        star.setFill(red\_linear);

        floor1.setFill(brown\_linear);

        floor2.setFill(brown\_linear);

        foot\_floor.setFill(brown\_linear);

        foot.setFill(brown\_linear);

        mash.setFill(brown\_linear);

        triangle\_L.setFill(green\_linear);

        triangle\_M.setFill(green\_linear);

        triangle\_S.setFill(green\_linear);

*/////ADD BUILD*

        root.getChildren().add(floor1);

        root.getChildren().add(floor2);

        root.getChildren().add(foot\_floor);

        root.getChildren().add(foot);

        root.getChildren().add(mash);

        root.getChildren().add(triangle\_L);

        root.getChildren().add(triangle\_M);

        root.getChildren().add(triangle\_S);

        root.getChildren().add(toy1);

        root.getChildren().add(toy2);

        root.getChildren().add(toy3);

        root.getChildren().add(toy4);

        root.getChildren().add(toy5);

        root.getChildren().add(toy6);

        root.getChildren().add(toy7);

        root.getChildren().add(toy8);

        root.getChildren().add(toy9);

        root.getChildren().add(toy10);

        root.getChildren().add(toy11);

        root.getChildren().add(toy12);

        root.getChildren().add(toy13);

        root.getChildren().add(toy14);

        root.getChildren().add(toy15);

        root.getChildren().add(star);

    }

}

**Результат**

****