

АНОТАЦІЯ

У даній курсовій роботі розглядається проект гри “Лабіринт”.

Робота виконана на 20 сторінках, містить 3 рисунка. Робота має 1 програму, з текстом програми та результатами.

Робота виконана українською мовою.

ABSTRACT

In this term paper, the project of the game "Labyrinth" is considered.

The work is executed on 20 pages, contains 3 pictures. The work has 1 program, with the text of the program and the results.

Work is executed in the Ukrainian language.

					122.2141.01. КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		1

ЗМІСТ

ВСТУП.....	3
1.АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	4
2.ПОСТАНОВКА ЗАДАЧІ.....	5
3.ДІАГРАМА КЛАСІВ.....	6
4.ДІАГРАМА СТАНІВ.....	7
5.ЗАСОБИ РОЗРОБКИ.....	8
6.РЕЗУЛЬТАТ РОЗРОБКИ.....	9
ВИСНОВОК.....	10
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	11
ДОДАТОК А.....	12

					122.2141.01. КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

З розвитком цифрових технологій комп'ютери все більше вливаються в життя людини. Якщо раніше ЕОМ використовувалися виключно для складних математичних обчислень, то сьогодні сфера їх застосування істотно розширилася. Комп'ютерні ігри - одне з найбільш масових застосувань електронних обчислювальних машин. Розвиток ігрової індустрії йшло стрімким темпом, і особливо користувалося популярністю у підлітків. Перші ігри відрізнялися простотою інтерфейсу і логіки, але з часом вони ставали все складніше і складніше, над їх створенням працювала вже не одна людина, а ціла команда розробників. Сучасні ігри вимагають досить великої продуктивності від комп'ютера, і не кожна офісна машина в силах відтворити їх. Однак для відпочинку від монотонної роботи часто досить простої, не вимогливою до техніки, гри. Саме такий розробці присвячений даний курсовий проект - гра «Лабіринт».

Ігри жанру Лабіринт (maze) характеризуються тим, що успіх гравця визначається в основному навігацією і орієнтацією в лабіринті. Деякі з них менше орієнтовані на навігацію, але більше на те, якою послідовністю дій дістатися до тих чи інших місць ігрового світу, зібрати предмети, розкидані на карті. Часто виконання завдання навігації і орієнтації відбувається в умовах, коли гравця переслідують вороги, але це не є обов'язковою умовою.

					122.2141.01. КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Метою даної курсової роботи є розробка гри „Лабіринт”. Вона відноситься до аркадних ігор, яка не потребує багато часу, але з інтенсивним ігровим процесом. Для цієї мети не підходять ігри типу стратегій, бо для них потрібно багато часу, а для 5-10 хвилинного відпочинку цілком підійде саме «Лабіринт». У стандартній поставці операційних систем даної гри немає, тому виникла необхідність у її створенні.

До ігор такого типу пред'являються такі вимоги: використання простих засобів управління; зручний графічний інтерфейс; поступове ускладнення гри. Під час виконання програма повинна виконуватися коректно і не приводити до збоїв.

Аналіз продуктів

Pacman – аркадна гра, основна мета якої є зібрати всю їжу на карті, уникаючи привидів. Коли Пакман досягає стіни, він з'являється з протилежної сторони поля. Може збирати об'єкти, які приносять бонуси. Перехід на наступний рівень відбувається, коли вся їжа зібрана. Складність гри зростає з кожним рівнем.

Diamond Rush – гра для мобільних телефонів, мета якої є зібрати якомога більше діамантів та дістатися виходу. Герой рухається по ігровому полю, уникає пастки та небезпечних істот. Герой має декілька «життів» – спроби, протягом яких гравець може продовжити гру. Також, за ігрову валюту можна придбати броню – одяг, в якому герою завдається менша шкода від навколишнього середовища.

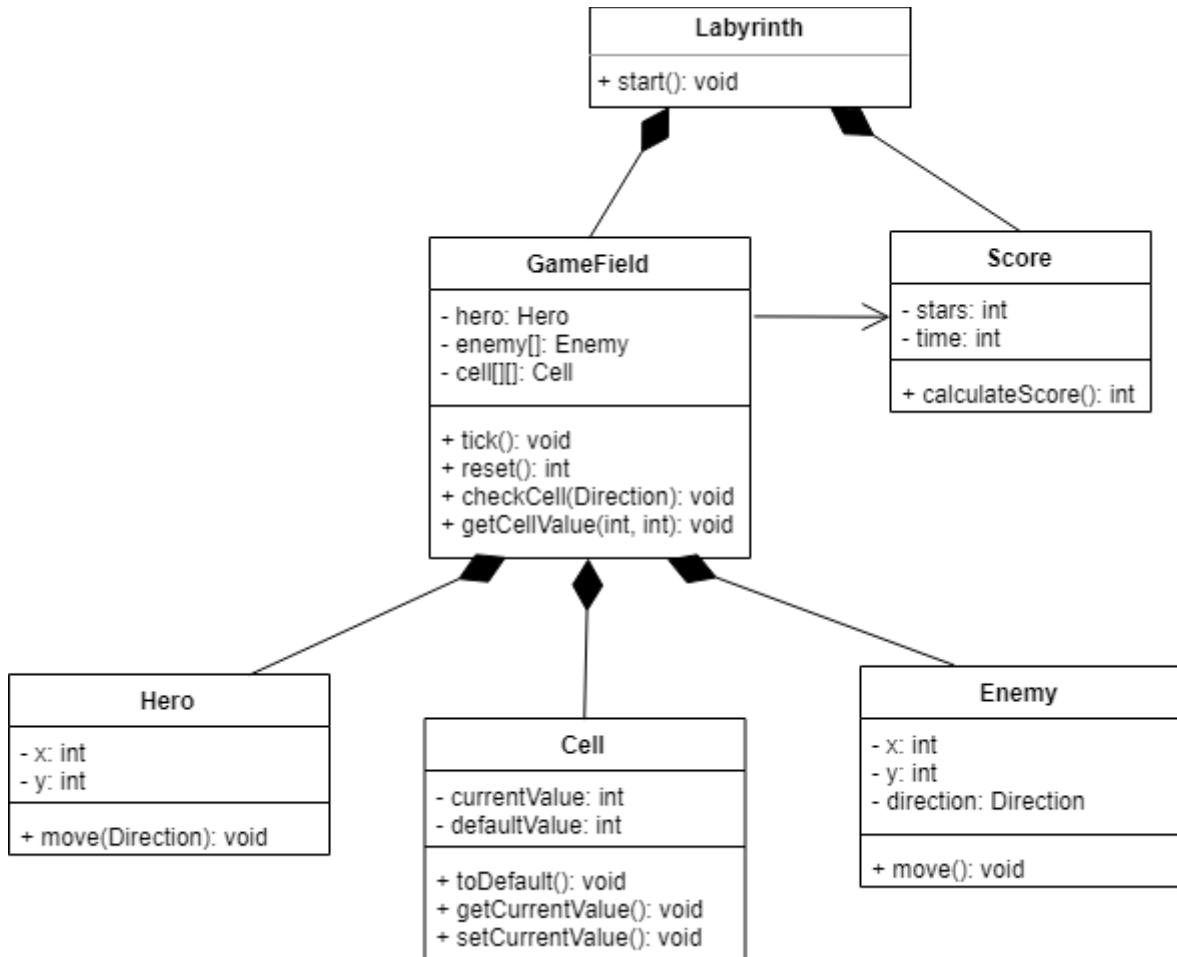
									Арк.
									4
Змн.	Арк.	№ докум.	Підпис	Дата	122.2141.01. КР				

ПОСТАНОВКА ЗАДАЧІ

Розробити програму “Лабіринт”, яка виконує переміщення героя по екрану, координату його місцезнаходження відповідно до дій гравця. Зміст гри полягає у тому, щоб дістатися виходу, збираючи зірки та уникаючи ворогів та ям. Гра складається з 3 рівнів, кожен з яких містить 5 зірок та 3 вороги. Закінчення гри – перемога (герой дійшов до останньої двері) або програш (контакт з ворогом або герой потрапив у яму). Програма починається з вибору складності. Від рівня складності залежить швидкість руху ворогів: Easy – раз у 0.5 секунди, Medium – 0.3, Hard – 0.1. Після цього гра починається. Напрямок героя можна керувати за допомогою клавіш-стрілок, герой переміщається на одну клітинку поля.

					122.2141.01. КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

ДІАГРАМА КЛАСІВ



Labyrinth – головний об’єкт.

GameField – основна панель, на якій будуть відбуватися дії гри.

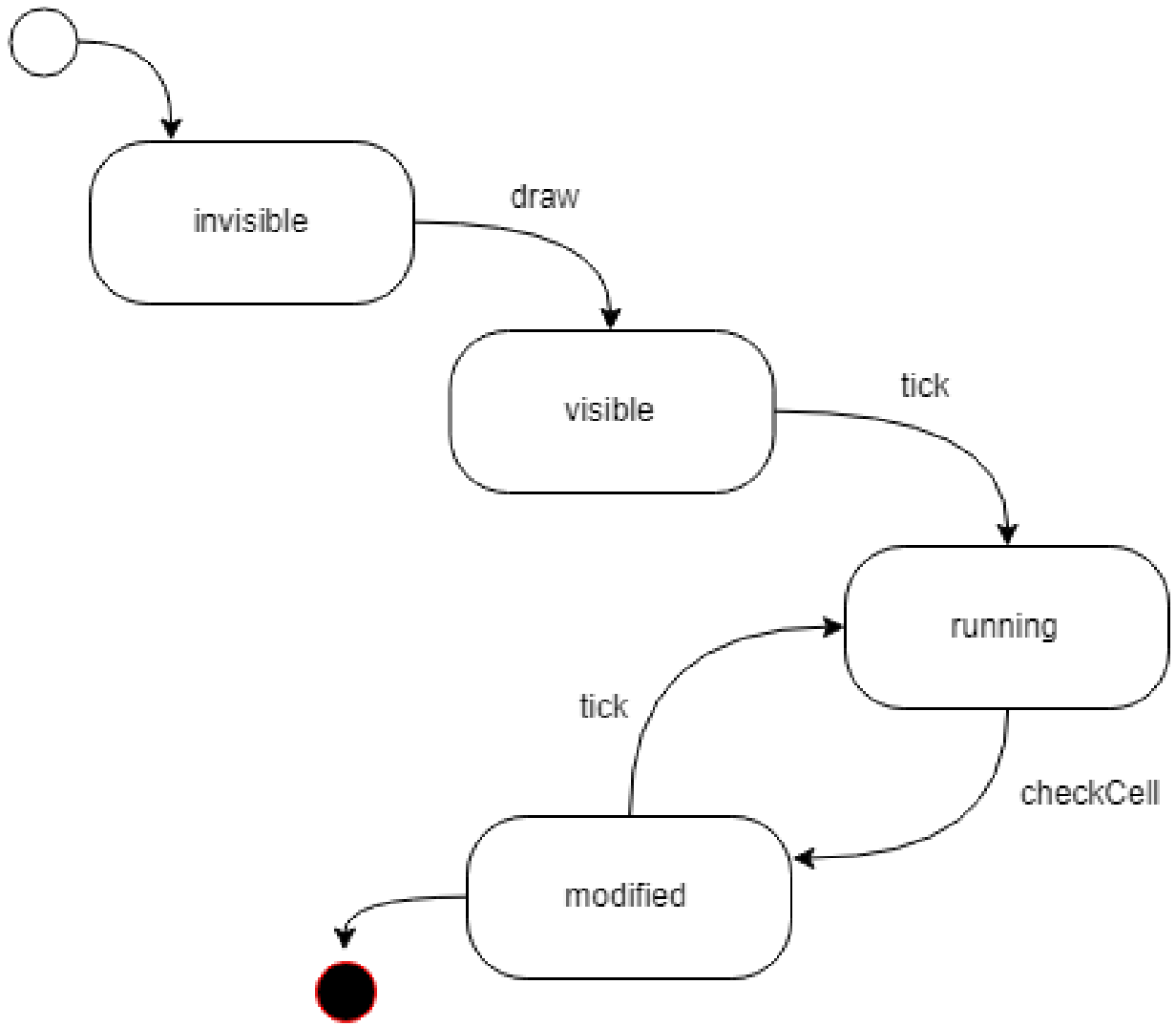
Score – об’єкт для зберігання кількості зібраних зірок та часу гри. **GameField** повідомляє йому про зібрану зірку.

Cell – обов’язкова складова **GameField**, містить інформацію про те, що знаходиться в даній області поля.

Hero – об’єкт, яким керує гравець. Обов’язкова складова **GameField**.

Enemy – об’єкт, який самостійно рухається з постійною швидкістю вперед та назад. Обов’язкова складова **GameField**.

ДІАГРАМА СТАНІВ



ЗАСОБИ РОЗРОБКИ

1) Мова програмування Java — об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині.

2) Середовище розробки IntelliJ IDEA - є інтегрованим середовищем розробки Java (IDE) для розробки комп'ютерного програмного забезпечення. Вона розробляється JetBrains (раніше відомим як IntelliJ), і є як Apache 2 Ліцензійного Community Edition, і в власній комерційній версії. Обидва можна використовувати для комерційного розвитку.

3) Бібліотека JavaFX - платформа та набір інструментів для створення насичених інтернет-застосунків (Rich Internet Applications, RIA) з можливістю підвантаження медіа та змісту. JavaFX включає в себе набір утиліт, за допомогою яких веб-розробники та дизайнери можуть швидко створювати та надавати розвинуті інтернет-застосунки для десктопів, мобільних пристроїв, телебачення та інших платформ. JavaFX складається з JavaFX Script і JavaFX Mobile. Починаючи з випуску JavaFX 2.0 забезпечено можливість створення JavaFX-застосунків, написаних цілком мовою Java.

									Арк.
									8
Змн.	Арк.	№ докум.	Підпис	Дата	122.2141.01. КР				

РЕЗУЛЬТАТ РОЗРОБКИ



					122.2141.01. КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

ВИСНОВОК

В даній роботі був розроблений проект комп'ютерної гри “Лабіринт”.

В результаті роботи була створена комп'ютерна гра “Лабіринт”. У процесі роботи було розроблено 2 діаграми: діаграма класів та діаграма станів.

Код програми представлено у додатку А.

					122.2141.01. КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Методичні вказівки до виконання курсових робіт з дисципліни "Мова програмування Java" / Є. Ю. Беркунський. – Миколаїв : НУК, 2015. – 29 с.
- 2) Бібліотека JavaFX: вступ [Електронний ресурс]. – Режим доступу: <https://metanit.com/java/javafx/1.1.php> - Заголовок з екрану.

					122.2141.01. КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11


```

    }

    public void tick() {
        for (int i = 0; i < ENEMIES; i++) {
            currentEnemyPosition.set(i, enemy[i].move());
            if (
hero.getPosition().equals(currentEnemyPosition.get(i)) ) {
                controller.loss();
            }
        }
    }

    public void reset() {
        stars = 0;
        for (int i = 1; i < SIZE-1; i++) {
            for (int j = 1; j < SIZE-1; j++) {
                cell[i][j].toDefault();
            }
        }
        hero = new Hero(heroStartPosition);
        initializeEnemies();
    }

    public void checkCell(Direction direction) {
        Position position = direction.next(new Position(hero.getX(),
hero.getY()));
        int x = position.getX();
        int y = position.getY();
        int value = cell[y][x].getCurrentValue();

        for (int i = 0; i < ENEMIES; i++) {
            if ( position.equals(enemy[i].getPosition()) ) {
                controller.loss();
            }
        }

        if ( value == 1 ) {
            hero.move(direction);
        } else if ( value == 2 ) {
            takeStar(x, y);
            hero.move(direction);
        } else if ( value == 3 ) {
            hero.move(direction);
            controller.draw();
            currentLevel++;
            if ( currentLevel < LEVELS ) {
                nextLevel();
            } else {
                controller.win();
            }
        } else if ( value == -1 ) {
            hero.move(direction);
            controller.draw();
            controller.loss();
        }
    }

    public int getCellValue(int x, int y) {
        return cell[y][x].getCurrentValue();
    }

```

										Арк.
										13
Змн.	Арк.	№ докум.	Підпис	Дата						

122.2141.01. KP

```

}

public Hero getHero() {
    return hero;
}

public Enemy getEnemy(int number) {
    return enemy[number];
}

private void nextLevel() {
    score.setStars(stars);
    stars = 0;
    controller.nextLevel();
    score.nextLevel();
    cell = map[currentLevel].getArray();
    hero = new Hero(heroStartPosition);
    initializeEnemies();
}

private void takeStar(int x, int y) {
    cell[y][x].setCurrentValue(1);
    stars++;
    score.setStars(stars);
}

private void initializeEnemies() {
    if ( currentLevel > 0 ) {
        for (int i = 0; i < ENEMIES; i++) {
            enemy[i] = new Enemy(enemyStartPosition[i],
enemyStartDirection[i], this);
        }
    } else {
        enemy[0] = new Enemy(enemyStartPosition[0],
enemyStartDirection[0], this);
        enemy[1] = new Enemy(new Position(6, 1),
enemyStartDirection[1], this);
        enemy[2] = new Enemy(enemyStartPosition[2],
enemyStartDirection[2], this);
    }

    for (int i = 0; i < ENEMIES; i++) {
        currentEnemyPosition.add(enemy[i].getPosition());
    }
}

private static Position heroStartPosition = new Position(1, 10);
private static Position[] enemyStartPosition = { new Position(1,
1), new Position(5, 1), new Position(10, 10) };
private static Direction[] enemyStartDirection = {
Direction.RIGHT, Direction.DOWN, Direction.UP };
private static Map[] map = new Map[LEVELS];

static {
    int[][] arr = {
        {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
        {0, 1, 2, 1, 1, 1, 1, 1, 1, 1, 3, 0},
        {0, 1, 1, -1, 1, 0, 1, 0, 2, -1, 1, 0},
        {0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0},
    }
}

```

						122.2141.01. KP	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			14

```

        {0, 1, 1, 1, 1, 0, 1, 0, 2, 1, 1, 0},
        {0, 1, 0, -1, 1, 1, 1, 0, -1, 0, 1, 0},
        {0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0},
        {0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0},
        {0, 2, 0, 1, 1, 0, 1, -1, 1, 0, 1, 0},
        {0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 2, 0},
        {0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0},
        {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
    };
    map[0] = new Map(arr);
    arr = new int[][] {
        {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
        {0, 1, 1, 1, 1, 1, 1, 1, 2, 1, 3, 0},
        {0, -1, 1, 0, 2, 1, 0, 1, 0, 1, 1, 0},
        {0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0},
        {0, 1, 0, 1, -1, 1, 1, 1, 1, 1, 1, 0},
        {0, 2, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0},
        {0, 0, 0, 1, 0, 1, 2, 0, 1, 0, 1, 0},
        {0, 1, 1, 1, 0, 1, 1, 0, 1, -1, 1, 0},
        {0, 1, 1, -1, 0, 0, 0, 0, 2, 1, 1, 0},
        {0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0},
        {0, 1, 1, 1, 1, -1, 1, 1, 1, 1, 1, 0},
        {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
    };
    map[1] = new Map(arr);
    arr = new int[][] {
        {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
        {0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 0},
        {0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0},
        {0, 1, 0, 2, 0, 1, 1, 1, 2, 0, 1, 0},
        {0, 1, 1, 1, 0, 1, -1, 1, 0, 0, 1, 0},
        {0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0},
        {0, 1, 1, 1, -1, 1, 1, 0, 2, -1, 1, 0},
        {0, 1, -1, 1, 1, 1, 1, 0, 0, 0, 1, 0},
        {0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 2, 0},
        {0, 1, 0, 0, 0, 0, 0, -1, 1, 0, 1, 0},
        {0, 1, 0, 2, 1, 1, 1, 1, 1, 1, 1, 0},
        {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
    };
    map[2] = new Map(arr);
}
}

```

Enemy.java:

```

package game.logic;

public class Enemy {
    private int x;
    private int y;
    private Direction direction;
    private final boolean isVertical;
    private GameField field;

    public Enemy(Position position, Direction initialDirection,
GameField field) {
        x = position.getX();
        y = position.getY();
    }
}

```

										Арк.
										15
Змн.	Арк.	№ докум.	Підпис	Дата	122.2141.01. КР					

```

        direction = initialDirection;
        isVertical = initialDirection == Direction.UP ||
initialDirection == Direction.DOWN;
        this.field = field;
    }

    public Position move() {
        Position next = direction.next(new Position(x, y));
        int value = field.getCellValue(next.getX(), next.getY());
        if ( value == 0 || value == 3 ) {
            if (isVertical) {
                direction = ( direction == Direction.UP)?
Direction.DOWN : Direction.UP);
            } else {
                direction = ( (direction == Direction.RIGHT)?
Direction.LEFT : Direction.RIGHT);
            }
            next = direction.next(new Position(x, y));
        }
        x = next.getX();
        y = next.getY();
        return new Position(x, y);
    }

    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }

    public Position getPosition() {
        return new Position(x, y);
    }
}

```

Controller.java:

```

package game.controller;

import game.logic.Direction;
import game.logic.GameField;
import game.logic.Score;
import game.view.GameView;
import javafx.animation.Animation;
import javafx.animation.KeyFrame;
import javafx.animation.Timeline;
import javafx.application.Platform;
import javafx.fxml.FXML;
import javafx.scene.canvas.Canvas;
import javafx.scene.control.Alert;
import javafx.scene.control.ButtonType;
import javafx.scene.input.KeyEvent;
import javafx.scene.layout.Pane;
import javafx.scene.text.Text;
import javafx.util.Duration;
import java.util.Optional;

```

									Арк.
									16
Змн.	Арк.	№ докум.	Підпис	Дата					


```

public class Controller {
    @FXML private Text starsText;
    @FXML private Text timeText;
    @FXML private Pane pane;
    @FXML private Canvas canvas;

    private GameView view;
    private GameField field;
    private Score score = new Score(GameField.LEVELS);

    private Timeline timeline;
    private Timeline enemyTimeline;
    private Timeline heroTimeline;
    private final int[] seconds = {0};
    private int enemySpeed;

    @FXML
    public void initialize() {
        canvas.widthProperty().bind(pane.widthProperty());
        canvas.heightProperty().bind(pane.heightProperty());
        field = new GameField(score, this);
        view = new GameView(field, canvas);
        difficultyDialog();
        initializeTimelines();
        start();
    }

    public void processKey(KeyEvent keyEvent) {
        switch (keyEvent.getCode()) {
            case LEFT: field.checkCell(Direction.LEFT); break;
            case RIGHT: field.checkCell(Direction.RIGHT); break;
            case UP: field.checkCell(Direction.UP); break;
            case DOWN: field.checkCell(Direction.DOWN); break;
            case E: exit(); break;
            case R: restartLevel(); break;
        }
    }

    public void draw() {
        if (view!=null) {
            view.draw();
        }
        starsText.setText("☆ "+ score.getStars());
    }

    public void exit() {
        Platform.exit();
    }

    public void restartLevel() {
        stop();
        score.setStars(0);
        score.setTime(0);
        seconds[0] = 0;
        field.reset();
        start();
    }
}

```

										Арк.
										17
Змн.	Арк.	№ докум.	Підпис	Дата						

122.2141.01. KP

```

public void nextLevel() {
    stop();
    score.setTime(seconds[0]);
    System.out.println(seconds[0]+" seconds");
    seconds[0] = 0;
    start();
}

public void win() {
    stop();
    score.setTime(seconds[0]);
    System.out.println(seconds[0]+" seconds");
    Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
    alert.setTitle("You win!");
    alert.setHeaderText("Score:
"+score.calculateScore()+"\nCollected stars:
"+score.allStars()+"\nTime: "+score.generalTime()+" seconds");
    alert.setResizable(false);
    alert.setContentText("Play again?");
    Optional<ButtonType> result = alert.showAndWait();
    ButtonType button = result.orElse(ButtonType.CANCEL);
    if (button == ButtonType.OK) {
        restartGame();
    } else {
        exit();
    }
}

public void loss() {
    stop();
    Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
    alert.setTitle("You lost!");
    alert.setHeaderText("Play again?");
    alert.setResizable(false);
    Platform.runLater( () -> {
        Optional<ButtonType> result = alert.showAndWait();
        ButtonType button =
result.orElse(ButtonType.CANCEL);
        if (button == ButtonType.OK) {
            restartLevel();
        } else {
            exit();
        }
    }
);
}

private void restartGame() {
    score = new Score(GameField.LEVELS);
    field = new GameField(score, this);
    view = new GameView(field, canvas);
    seconds[0] = 0;
    start();
}

private void start() {
    timeline.play();
    heroTimeline.play();
    enemyTimeline.play();
}

```

										Арк.
										18
Змн.	Арк.	№ докум.	Підпис	Дата						

```

    }

    private void stop() {
        timeline.stop();
        heroTimeline.stop();
        enemyTimeline.stop();
    }

    private void initializeTimelines() {
        timeline = new Timeline(new KeyFrame(Duration.millis(1000), e-
>timerProcessTick()));
        heroTimeline = new Timeline(new KeyFrame(Duration.millis(100),
e->heroProcessTick()));
        enemyTimeline = new Timeline(new
KeyFrame(Duration.millis(enemySpeed), e->enemyProcessTick()));
        timeline.setCycleCount(Animation.INDEFINITE);
        heroTimeline.setCycleCount(Animation.INDEFINITE);
        enemyTimeline.setCycleCount(Animation.INDEFINITE);
    }

    private void timerProcessTick() {
        timeText.setText(String.format("%02d:%02d", (seconds[0])/60,
(seconds[0])%60));
        seconds[0]++;
    }

    private void enemyProcessTick() {
        field.tick();
        draw();
    }

    private void heroProcessTick() {
        draw();
    }

    private void difficultyDialog() {
        Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
        alert.setTitle("Difficulty");
        alert.setHeaderText("Select difficulty:");
        alert.setResizable(false);

        ButtonType easyButton = new ButtonType("Easy");
        ButtonType mediumButton = new ButtonType("Medium");
        ButtonType hardButton = new ButtonType("Hard");
        ButtonType exitButton = new ButtonType("Exit game");
        alert.getButtonTypes().setAll(easyButton, mediumButton,
hardButton, exitButton);

        Optional<ButtonType> result = alert.showAndWait();
        if (result.get() == easyButton) {
            enemySpeed = 500;
        } else if (result.get() == mediumButton) {
            enemySpeed = 300;
        } else if (result.get() == hardButton) {
            enemySpeed = 100;
        } else if (result.get() == exitButton) {
            exit();
        }
    }
}

```

									Арк.
									19
Змн.	Арк.	№ докум.	Підпис	Дата	122.2141.01. КР				

