

**Submitted by**

Alexis GUERET

Meriadeg GUICHARD

Anaïs JARNO

Spring 2018

**Project – LP24**

**Snakes and ladders game**

*Written report*

**TABLE OF CONTENTS**

[1 Introduction 1](#_Toc517098014)

[1.1 Game’s Principles 1](#_Toc517098015)

[1.2 Aims and Objectives 2](#_Toc517098016)

[2 Program organization 3](#_Toc517098017)

[2.1 The Gameboard and its Frame 3](#_Toc517098018)

[2.2 The Multiplayer Configuration 4](#_Toc517098019)

[2.3 The system of Questions 4](#_Toc517098020)

[2.4 The movement of tokens 5](#_Toc517098021)

[2.5 The system of winning the game 5](#_Toc517098022)

[3 Implementation choices 6](#_Toc517098023)

[3.1 Directories, Packages Structure 6](#_Toc517098024)

[3.2 MVC Pattern 6](#_Toc517098025)

[4 Manual Guide of the game 7](#_Toc517098026)

[5 Global Evaluation 8](#_Toc517098027)

[5.1 Evaluation of Objectives and Aims 8](#_Toc517098028)

[5.2 Problems encountered 8](#_Toc517098029)

[5.3 Evaluation of Project Management 8](#_Toc517098030)

[5.4 Further Work 9](#_Toc517098031)

[6 Conclusion 10](#_Toc517098032)

[7 Bibliography 11](#_Toc517098033)

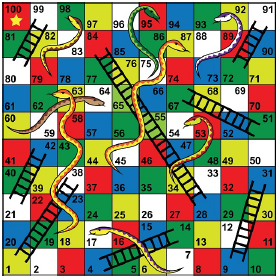
# Introduction

This report discusses the work done in development of the project about the Snake and Ladders Game.

## Game’s Principles

Snakes and Ladders is an ancient Indian board game that historically was based on morality lessons. The player's progression up the board represented a life journey complicated by virtues (ladders) and vices (snakes).

It is played between two or more players on a gameboard having numbered, gridded squares. Play begins on square number 1 at the bottom left-hand corner of the board and finishes on the last space of the board. Players take turns to roll a dice and move along the number of spaces rolled if they correctly answer a question. Several "ladders" and "snakes" are pictured on the board, each connecting two specific board squares.

The objective of the game is to be the first player who reaches the end navigating on the board, following the numbers of each gridded square, helped or hindered by ladders and snakes respectively.

## Aims and Objectives

To implement this game, the followed points needs to be approached:

First, create a gameboard with cells, snakes and ladders.

Next, configure the game to be multiplayer, so having turns, players and their corresponding tokens.

Then, configure the system of questions asked to the player during his turn.

Finally, the movement of the token and the system of winning the game.

Along with the above approached points, we’ll discuss the optimizations, the encountered difficulties and the result.

# Program organization

## The Gameboard and its Frame

When you load the game, a frame is opened and positioned right in the middle of your screen.

This frame contains a menu bar which allow the user to see the rules, play again or close the game. A controller is used to perform actions according to the chosen option.

There is in the center of the frame the grid display with tokens, snakes and ladders.

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevéRight after, you can find the button to allow you to advance in the game: it’s a button to roll the dice, another one to stop the rolling and then, a picture of the corresponding value of the dice is displayed.

## The Multiplayer Configuration

## The system of Questions

To make the questioning system easier, we decided to use a true-false type of questioning. So that there wouldn’t have any problems as deciding if the input would correspond to a satisfactory answer. That way, we just need to store some affirmations and their correction (TRUE or FALSE).

To store these data, we chose to use .properties files. In fact, this type of file managed as values such as key/value pairs. In each pair, the key and value are both String values. The key identifies and is used to retrieve the value.

To make beneficial use of this pairing system, we choose to use numbers converted into string as keys, so we would just have to generate a random number to get back a question and its corresponding answer. And so, our value contains the affirmation separated from the correction by a point.

After creating the .property file, we needed to create a java class which would get the values. To resolve this matter, we created a class with a ResourceBundle loading the storing file. In this class, we implemented a method to generate a pair question and its corresponding answer. So, inside this one, we get the value accordingly of the randomly generated integer used as key. Then, inside this value, we are looking for the index of the point to substring value into to strings: question and answer.

|  |  |  |
| --- | --- | --- |
| **VALUE** | | |
| **Question** | **.** | **answer** |
| *substring(0, index);* | *index = value.indexOf('.');* | *substring(index+1);* |

To use our questioning system, we simply need to instantiate an object of this class, then call the generation method each time we need a new question, with the getters for the question and the answer.

Moreover, by using .properties files it would allow us to improve our game by giving a multilanguage platform.

## The movement of tokens

## The system of winning the game

# Implementation choices

## Directories, Packages Structure

## MVC Pattern

# Manual Guide of the game

# Global Evaluation

## Evaluation of Objectives and Aims

## Problems encountered

## Evaluation of Project Management

To manage the coworking and the evolution of the project, we choose to use GitLab.

The tool GitLab is an online manager of source codes versions. It served us for the realization of the project to see the evolution of the program while having a visual management of the development process.

Thanks to GitLab the collaboration during such a project becomes more efficient and easier to manage.

Indeed, after changes files in our project, we simply put them online with Git commands and then, other users can quickly retrieve the latest changes to the program while working on their own tasks.

Some features allow you to define a role for users participating in the project, to create tasks to perform and to assign them to one of the users.

They can also be ranked in order of priority and accordingly to their status to which they are assigned. For example, for our project, each task had to go through the status "To do", then "Doing" (in progress) when the user concerned took care of it, followed by "Done" (done) when the completion of the task is finished, finally ending with the "Closed" status when all potential and task-related bugs are resolved.

Everything is displayed in a dashboard to get a global view of the tasks and their status.

To Do

Doing

Done

Closed

## Further Work

Certain elements in this project leave scope for further development. To highlight the general areas where extra work would benefit the project:

# Conclusion

# Bibliography