GAME

LADDER AND SNAKE – Program 2

Cao, Lisi

2023

Table of Content

[Project Introduction: 1](#_Toc139913085)

[Software Structure Diagram: 3](#_Toc139913086)

[Flowing Diagram: 4](#_Toc139913087)

[Coding 5](#_Toc139913088)

[Proofreading 5](#_Toc139913089)

[Restructure 5](#_Toc139913090)

[Finish 5](#_Toc139913091)

# Project Introduction:

This project aims to develop a console-based "Snakes and Ladders" game. The game can accommodate 2 to 4 players, with players moving their pieces by rolling a dice. The game rules follow the traditional "Snakes and Ladders" game, with the first player to reach the 100th square declared as the winner.

**Target Audience:** The target users for the game are those looking for a simple, quick game to pass time.

**Functional Requirements:**

**Player Creation:** The system should allow users to input player names and create 2 to 4 players. The input is the player names, and the output is the created players.

**Dice Rolling:** Players should be able to roll a dice, getting a random number between 1 and 6. The input is the player's action, and the output is the result of the dice.

**Piece Movement:** The system should move the player's piece according to the result of the dice. If the piece lands at the bottom of a ladder, it should be moved to the top of the ladder; if it lands on the head of a snake, it should be moved to the tail of the snake. The input is the result of the dice, and the output is the new position of the piece on the board.

**Game Over Determination:** When a player's piece reaches the 100th square, the system should determine the game to be over and declare the player as the winner. The input is the position of the piece, and the output is the result of the game.

**Non-Functional Requirements:**

**Usability:** The game should have simple, clear instructions so that users can easily understand how to play the game.

**Response Speed:** The system should immediately show the result after the player rolls the dice.

Constraints: Due to the project's time and budget constraints, we will only develop a console-based version instead of a graphical interface version.

**Acceptance Criteria:** At the end of the project, we will conduct a series of tests to ensure all functional requirements have been implemented. For example, we will check whether players can be created, whether dice can be rolled, whether pieces move correctly, and whether the system correctly determines the end of the game, etc.

# Software Structure Diagram:

**Software Introduction:** Name, author, flow introduction, welcome words.

**Game Players:** Player class (Player), which saves each player's basic information, such as name, current position, etc.

**Game Board:** GameBoard class (GameBoard), which saves the basic information of the game, such as the positions of the snake's head and tail, the positions of the top and bottom of the ladder, etc.

**Game Class:** Game class (Game), which will control the main logic of the game, such as deciding the order of players, handling the movements of players, etc.

**Dice Class:** Dice class (Dice), which will return a random value from 1 to 6 each time it is called.

# Flowing Diagram:

图示

描述已自动生成

# Coding

# Proofreading

# Restructure

# Finish