- *Maria Alejandra Mantilla Coral: A00395792*
- *Pablo Fernando Pineda Patiño: A00395831*

**SOFTWARE ENGINEERING PROBLEM SPECIFICATION TABLE**

| | |
|---|---|
| CLIENT | Snakes and Ladders Inc. |
| USER | Players. |
| FUNCTIONAL REQUIREMENTS | **R1:** Show start menu.<br>**R2:** Register players.<br>**R3:** Create the game board.<br>**R4:** Display game board with players' positions.<br>**R5:** Show turn menu.<br>**R6:** Roll dice randomly.<br>**R7:** Show game board with position of snakes and ladders.<br>**R8:** Calculate the score of the winner of each game.<br>**R9:** Display list of scores sorted in ascending order. |
| PROBLEM CONTEXT | To achieve proper software development, the following aspects must be taken into account during the process:<br><br>The goal is to create a game of Snakes and Ladders, which starts with a simple menu containing two options: Play and Quit. If the user chooses the first option, they will be asked for the parameters of the game (the size of the grid, the snakes, and the ladders). These ladders will be randomly placed on any square of the board, subject to the following restrictions: no ladder starts on |

| | |
|---|---|
| | square 1, no snake starts on square n x m, and no start or end square of a ladder or snake can coincide with another start or end of a ladder or snake.<br><br>Once the user has entered the game parameters, the program will display a grid formed by brackets, with the squares correctly numbered and the location of the ladders and snakes. After that, the game begins and each player has a menu of two options on each turn: roll the dice, or view the ladders and snakes.<br><br>With the option to roll the dice, the player rolls the dice and advances n squares forward. If the player lands on a snake, they move back along the board. If the player lands on a ladder, they move forward. If the player lands on a normal square, they remain there. On the other hand, with the second option, the player can view the ladders and snakes on the board.<br><br>After choosing one of the above options, the board, the player's current position, and the snakes and ladders on the board are shown.<br><br>After a player has reached the finish line, the game ends and each player is awarded points, and finally, a table is shown with the global rankings according to the points in ascending order. |
| NON-FUNCTIONAL REQUIREMENTS | **Usability:** The game's user interface should be easy to use and understandable by a wide audience. |
| PROCESS REQUIREMENTS | - The grid must be modeled and implemented using linked lists.<br><br>- It is not possible to use any arrays, arraylists, or Java collections in this program.<br><br>- It is not possible to use loops in this program. All iterations must be done using recursion.<br><br>- All ladders and snakes must be modeled as connections between nodes of the linked structure. |

**Functional Requirements Analysis Tables**

| | | | |
|---|---|---|---|
| Name or identifier | **R1:** Show start menu. | | |
| Summary | Present a simple menu to the user with options to choose from. The options are:<br>1. Play<br>2. Exit | | |
| Inputs | Input name | Data type | Condition of selection and repetition |
| | option | int | Only if the user chose an option less than 1 or greater than 2. |
| | | | |
| | | | |
| | | | |
| | | | |
| General activities necessary to obtain the result | - Display the list of options to the user.<br><br>- Allow the user to select an option. | | |
| Result or postcondition | The user's selected option is returned or performed. | | |
| Outputs | Output name | Data type | Condition of selection and repetition |
| | selected_option | int | Returned once the user selects an option. |
| | | | |
| | | | |
| | | | |

| | |
|---|---|
| Name or identifier | **R2:** Register players. |

| | | | |
|---|---|---|---|
| Summary | This feature allows players to register in the system in order to play. The user will type their name and it will be stored in a list of players. | | |
| Inputs | Input name | Data type | Condition of selection and repetition |
| | name | String | |
| | nickname | String | Only if the nickname is not already registered in the system. |
| | symbol | String | |
| | | | |
| General activities necessary to obtain the result | - Enter the player's name.<br>- Enter the player's nickname.<br>- Validate that the nickname does not already exist.<br>- Store the player in the list of players. | | |
| Result or postcondition | A message will be displayed on the screen informing that the user has registered successfully. In case there was an error or the nickname already existed during the registration process, an error message will be displayed. | | |
| Outputs | Output name | Data type | Condition of selection and repetition |
| | msg_succesfull | String | Only if all the steps of the registration were successful. |
| | msg_error | String | Only if there was an error or issue during the registration process. |
| | | | |
| | | | |

| | | | |
|---|---|---|---|
| Name or identifier | **R3:** Create the game board. | | |
| Summary | It allows the user to choose the size of the game board, i.e. the number of rows and columns it will have.<br>It also allows the user to choose the number of snakes and ladders and generates them randomly on the game board. | | |
| Inputs | Input name | Data type | Condition of selection and repetition |

| | | | |
|---|---|---|---|
| | rows | int | Only if the user does not type a positive integer. |
| | columns | int | Only if the user does not type a positive integer. |
| | ladders | int | Only if the user does not type a positive integer. |
| | snakes | int | Only if the user does not type a positive integer. |
| General activities necessary to obtain the result | - Display a message on the screen asking the user to enter the number of rows and columns desired.<br><br>- Display a message on the screen asking the user to enter the number of snakes and ladders desired.<br><br>- Validate that the entries are positive integers.<br><br>- Generate the game board with the given number of rows and columns.<br><br>- Randomly place the snakes and ladders on the game board. | | |
| Result or postcondition | The dimensions of the game board will have been defined and the board along with the snakes and ladders will have been randomly generated. | | |
| Outputs | Output name | Data type | Condition of selection and repetition |
| | game_board | matrix | |
| | | | |
| | | | |

| | | | |
|---|---|---|---|
| Name or identifier | **R4:** Display game board with players' positions. | | |
| Summary | Displays the current state of the game board, showing the positions of the players and any other relevant game information. | | |
| Inputs | Input name | Data type | Condition of selection and repetition |
| | | | |
| | | | |

| | | | |
|---|---|---|---|
| General activities necessary to obtain the result | - Access the current state of the game board.<br><br>- Create a visual representation of the game board that includes player positions and other game elements.<br><br>- Display the visual representation of the game board to the players. | | |
| Result or postcondition | The players will be able to see a visual representation of the game board that accurately reflects the current state of the game, including player positions and other game elements. This information can be used by the players to make decisions and take actions during the game. | | |
| Outputs | Output name | Data type | Condition of selection and repetition |
| | board_game | matrix | |
| | | | |
| | | | |

| | | | |
|---|---|---|---|
| Name or identifier | **R5:** Show turn menu. | | |
| Summary | Present a simple menu to the user with options to choose what they want to do on their turn. The options are:<br>1. Throw the dice<br>2. Show snakes and ladders | | |
| Inputs | Input name | Data type | Condition of selection and repetition |
| | option | int | Only if the user chose an option less than 1 or greater than 2. |
| | | | |
| | | | |
| General activities necessary to obtain the result | - Display the list of options to the user.<br><br>- Allow the user to select an option. | | |
| Result or postcondition | The user's selected option is returned or performed. | | |

| | | | |
|---|---|---|---|
| Outputs | Output name | Data type | Condition of selection and repetition |
| | selected_option | int | Returned once the user selects an option. |
| | | | |
| | | | |

| | | | |
|---|---|---|---|
| Name or identifier | **R6:** Roll dice randomly. | | |
| Summary | Simulates a random dice roll, to determine the number of squares the player will move on his turn. | | |
| Inputs | Input name | Data type | Condition of selection and repetition |
| | | | |
| | | | |
| | | | |
| General activities necessary to obtain the result | - Use a random number generation algorithm to simulate the die roll.<br><br>- Verify that the number generated is within the range of valid values for the die being used in the game. | | |
| Result or postcondition | The result will be a whole number, representing the number of squares the player will move in his turn. | | |
| Outputs | Output name | Data type | Condition of selection and repetition |
| | resultDice | Int | |
| | | | |
| | | | |

| | |
|---|---|
| Name or identifier | **R7:** Show game board with position of snakes and ladders. |
| Summary | Visually displays the snakes and ladders on the game board. To achieve this, it is required to obtain the previously generated information of the positions of the snakes and ladders on the board and present them clearly and concisely in the user interface. |

| | | | |
|---|---|---|---|
| Inputs | Input name | Data type | Condition of selection and repetition |
| | | | |
| | | | |
| | | | |
| General activities necessary to obtain the result | - Access the current state of the game board.<br><br>- Create a visual representation of the game board that includes the positions of snakes and ladders<br><br>- Display the visual representation of the game board with the snakes and ladders. | | |
| Result or postcondition | Players will be able to see a visual representation of the game board that accurately shows the positions of snakes and ladders. | | |
| Outputs | Output name | Data type | Condition of selection and repetition |
| | laddersAndSnakes | matrix | |
| | | | |
| | | | |

| | | | |
|---|---|---|---|
| Name or identifier | **R8:** Calculate the score of the winner of each game. | | |
| Summary | Calculate the score of the winner of each game based on the game rules. | | |
| Inputs | Input name | Data type | Condition of selection and repetition |
| | | | |
| | | | |
| | | | |
| General activities necessary to obtain the result | - Determine the game rules.<br>- Calculate the score for each player based on the rules.<br>- Determine the winner based on the score.<br>- Output the score of the winner. | | |
| Result or postcondition | The score of the winning player will be calculated and displayed to the user. | | |

| | | | |
|---|---|---|---|
| Outputs | Output name | Data type | Condition of selection and repetition |
| | winner_score | int | Only if a winner is determined based on the game rules. |
| | | | |

| | | | |
|---|---|---|---|
| Name or identifier | **R9:** Display list of scores sorted in ascending order. | | |
| Summary | Displays a list of the scores obtained by the players during the game, sorted in ascending order.<br>Example:<br>1. Pablo -> 5 points<br>2. Cristiano Ronaldo -> 3 points<br>3. Alejandra -> -10 points | | |
| Inputs | Input name | Data type | Condition of selection and repetition |
| | | | |
| | | | |
| | | | |
| General activities necessary to obtain the result | - Read game logs to retrieve scores for all players.<br><br>- Apply the formula to find the score to each player, which is: (600 - t in seconds) / 6<br><br>- Sort the scores in ascending order.<br><br>- Generate a formatted string displaying the sorted scores in ascending order. | | |
| Result or postcondition | The top players according to the score will be shown in ascending order. | | |
| Outputs | Output name | Data type | Condition of selection and repetition |
| | topScores | String | |
| | | | |
| | | | |