## **Universidad ICESI**

Juan Daniel Torres Zúñiga – A00392872

W. Joseph Verdesoto Vélez – A00395664

**Requirements.**

| Client | Snakes and Ladders Inc. |
| --- | --- |
| Users | Players |
| Functional Requirements | R1: Register the players.  R2: Register the number of rows and columns that the game will have.  R3: Register the number of snakes and ladders that the game will have.  R4: Implement a turn-based system.  R5: Show the game board with the snakes and with the ladders.  R6: Record the time from the start to the end of the game.  R7: Implement the menu to start the game.  R8: Implement a menu to every turn of each player. |
| Context of the Problem | * To know the functioning of the game, as the rules and limitations of it. * To know the maximum number of players. * Study the possible limit to the amount of ladders and snakes of the game. * To know the mobility that the player will have. * How will the game be typecasting? |
| Non Functional Requirements | R: Implement Java as the programming language.  R: Save the player’s information.  R: Implement a points-based system to the final of the game.  R: Use linked lists to model the game.  R: Use recursive methods to the readability of the code. |

| Name or Identifier | **R1: Register Players.** | | |
| --- | --- | --- | --- |
| Abstract | The game has to be able to register the players and save his/her basic attributes. | | |
| Entry | **Entry Name** | **Type** | **Selection or Repetition Condition** |
| Name | String | None |
|  | Token | String | Different token from the rest of the players |
| General Activities necessary to obtain a result | The game has to get initialized, then it saves the data typed by the user in an object of type player. | | |
| Postcondition or Result | The player will be created | | |
| Output | **Output Name** | **Type** | **Selection or Repetition Condition** |
| P1 | Player | The attributes must not repeat. |

| Name or Identifier | **R2: Register the number of rows and columns that the game will have.** | | |
| --- | --- | --- | --- |
| Abstract | The game has to ask the initial user for the parameters to create the game board. | | |
| Entry | **Entry Name** | **Type** | **Selection or Repetition Condition** |
| Columns | Int | It has to be a positive integer number |
|  | Rows | Int | It has to be a positive integer number |
| General Activities necessary to obtain a result | The game has to be initialized, then creates a board used the number of rows and columns typed by the user. | | |
| Postcondition or Result | The gameboard will be created. | | |
| Output | **Output Name** | **Type** | **Selection or Repetition Condition** |
| message | String | Confirms the successful creation of the board. |

| Name or Identifier | **R3: Register the number of snakes and ladders that the game will have.** | | |
| --- | --- | --- | --- |
| Abstract | The game has to ask the initial user for the parameters to create the game board. | | |
| Entry | **Entry Name** | **Type** | **Selection or Repetition Condition** |
| Snakes | Int | It has to be a positive integer number |
|  | Ladders | Int | It has to be a positive integer number |
| General Activities necessary to obtain a result | The board and the players have to already created | | |
| Postcondition or Result | The snakes and ladders will be randomly created. | | |
| Output | **Output Name** | **Type** | **Selection or Repetition Condition** |
| S | Node | The tail and head of the snake must not share position with any ladder or snake |
|  | L | Node | The tail and head of the ladder must not share position with any ladder or snake |

| Name or Identifier | **R4: Implement a turn-based system.** | | |
| --- | --- | --- | --- |
| Abstract | The game will have a turn-based system in which every player will move their game piece. | | |
| Entry | **Entry Name** | **Type** | **Selection or Repetition Condition** |
| Option | Int | The number has to match with the chosen option |
| General Activities necessary to obtain a result | The player´s status has to be true in order for it to start its turn. | | |
| Postcondition or Result | The player will move their piece. | | |
| Output | **Output Name** | **Type** | **Selection or Repetition Condition** |
| Board | linkedList | No player has reached the end yet. |

| Name or Identifier | **R5: Show the game board with the snakes and with the ladders.** | | |
| --- | --- | --- | --- |
| Abstract | The game has to be able to show in the console symbols to represent the board with the snakes and ladders actualizing every move. | | |
| Entry | **Entry Name** | **Type** | **Selection or Repetition Condition** |
|  |  |  |
| General Activities necessary to obtain a result | The game read the last move made by the player and actualized the position of the player’s game pieces on the board. Then print in the console with symbols, numbers and letters a representation of the actual board. | | |
| Postcondition or Result | An actualized board is printed. | | |
| Output | **Output Name** | **Type** | **Selection or Repetition Condition** |
| Board | linkedList | The player has to make a move. |

| Name or Identifier | **R6: Record the time from the start to the end of the game.** | | |
| --- | --- | --- | --- |
| Abstract | The game has to be able to take the time of a game from the start to finish. | | |
| Entry | **Entry Name** | **Type** | **Selection or Repetition Condition** |
| initHour | Int | It has to be a positive integer number |
|  | initMinute | Int | It has to be a positive integer number |
|  | endHour | int | It has to be a positive integer number |
|  | endMinute | int | It has to be a positive integer number |
| General Activities necessary to obtain a result | The game uses the date library to save the initial hour and minute and the compare it with the final hour and minut, Finally the game prints the time passed. | | |
| Postcondition or Result | The game prints a message with the time passed. | | |
| Output | **Output Name** | **Type** | **Selection or Repetition Condition** |
| message | String | The players has to start and finish a game. |

| Name or Identifier | **R7: Implement the menu to start the game.** | | |
| --- | --- | --- | --- |
| Abstract | The game has to be able to show an initial menu to start. | | |
| Entry | **Entry Name** | **Type** | **Selection or Repetition Condition** |
| None | Void |  |
| General Activities necessary to obtain a result | The program prints in the console the initial menu. | | |
| Postcondition or Result | Prints the principal menu of the game | | |
| Output | **Output Name** | **Type** | **Selection or Repetition Condition** |
| initMenu | String | The user has to init the program. |

| Name or Identifier | **R8: Implement a menu to every turn of each player.** | | |
| --- | --- | --- | --- |
| Abstract | The game has to be able to show a menu after every move made by each player. | | |
| Entry | **Entry Name** | **Type** | **Selection or Repetition Condition** |
| None | Void |  |
| General Activities necessary to obtain a result | The game prints the gaming menu after every player move. | | |
| Postcondition or Result | The game prints the gaming menu. | | |
| Output | **Output Name** | **Type** | **Selection or Repetition Condition** |
| gamingMenu | String | the players has to make a move before. |