

Project developers: Miguel Gonzalez, Alexandro Cabezas, Esteban Gaviria.

Requirements Analysis Table.

CLIENT	Snakes and Ladders Inc.
USERS	Gamers
FUNCTIONAL REQUIREMENTS	<ol style="list-style-type: none"> 1. Generate game board. 2. Roll dice and move player. 3. Use a ladder. 4. Use snake. 5. Show ladders and snakes. 6. Calculate the score of the match winner. 7. Show scoreboard.
CONTEXT OF THE PROBLEM	<p>The famous company Snakes and Ladders Inc. wants a program that allows play and also simulates the famous game Snakes and Ladders. The program can have a text console interface. The game should present the user with a grid or table of n rows by m columns, within which are s snakes and e ladders.</p>
NON-FUNCTIONAL REQUIREMENTS	<ol style="list-style-type: none"> 1. The software must be programmed on java. 2. The grid must be modeled and implemented using linked lists. 3. It is not possible to use any array, or array list, or any Java collection in this program. 4. It is not possible to use cycles in this program. All iterations must be done using recursion. 5. All ladders and snakes must be modeled as connections between nodes of the linked structure

Functional Requirements Analysis Tables.

Name or identifier	R1. Generate game board.		
Summary	The system must generate a game board that must be presented to the user as a grid or table of n rows by m columns, within which there are s snakes and e ladders (both the rows, columns, stairs and snakes are given by the user). Once generated, the system generates the turn system.		
Inputs	Input name	Datatype	Selection or repetition condition
	rows	int	The number entered must be greater than 0
	columns	int	The number entered must be greater than 0
	snakes	int	The number entered must be greater than 0
	ladders	int	The number entered must be greater than 0
General activities necessary to obtain the results	<ol style="list-style-type: none"> 1. Read the rows 2. Check that the entered number is valid 3. Read the columns 4. Check that the entered number is valid 5. Read the snakes 6. Check that the entered number is valid 7. Read the ladders 8. Check that the entered number is valid 9. Generate game board 10. Add ladders and snakes to the game board (Restrictions: no straight starts at square 1, no snake starts at square n x m, and no straight or snake start or end square must coincide with another straight or snake start or end). 11. Assign random symbol to each player and put it on space 1 of game 12. Generate the turn system 13. Show game board 		

Result or postcondition	The game board of the current match and the turn system is generated with each of the players on square 1.		
Outputs	Output name	Datatype	Selection or repetition condition
	gameBoard	String	

Name or identifier	R2. Roll dice and move player.		
Summary	The system must allow the player to roll the dice (which generates a random number from 1 to 6) and move it on the game board, the squares indicated by it.		
Inputs	Input name	Datatype	Selection or repetition condition
	option	int	
General activities necessary to obtain the results	<ol style="list-style-type: none"> 1. The player selects the roll dice option 2. Generate a random number from 1 to 6 3. Move the player on the game board the number of squares indicated by the dice. 4. Use ladder or snake (if there is one on the square where the player landed) 5. Refresh game board. 		
Result or postcondition	The player changes its position on the game board.		
Outputs	Output name	Datatype	Selection or repetition condition
	gameBoard	String	

Name or identifier	R3. Use a ladder.		
Summary	If the player lands on a space where there is a ladder, the system advances them on the board to the space the top of the ladder points to.		
Inputs	Input name	Datatype	Selection or repetition condition

	none	none	none
General activities necessary to obtain the results	1. Find the top end of the ladder on the game board. 2. Move player to found square		
Result or postcondition	The player is moved to the top square of the stairs.		
Outputs	Output name	Datatype	Selection or repetition condition
	none	none	none

Name or identifier	R4. Use snake.		
Summary	If the player lands on a square where there is a snake, the system will delay them on the board to the square that the snake's tail is pointing to.		
Inputs	Input name	Datatype	Selection or repetition condition
	none	none	none
General activities necessary to obtain the results	1. Find the tail of the snake on the game board. 2. Move player to found square		
Result or postcondition	The player is moved to the snake tail.		
Outputs	Output name	Datatype	Selection or repetition condition
	none	none	none

Name or identifier	R5. Show ladders and snakes.
---------------------------	-------------------------------------

Summary	The system allows the player to see the position of the ladders and snakes located on the game board.		
Inputs	Input name	Datatype	Selection or repetition condition
	option	int	
General activities necessary to obtain the results	<ol style="list-style-type: none"> 1. The player selects the option to show ladders and snakes 2. Generate a String with brackets [] that represent each square on the game board, and place on it the number or letter corresponding to the ladder or snake located in the corresponding square (If there is). 3. Display the game board only with the positions of the ladders and snakes that are distributed on it. 		
Result or postcondition	The system displays the game board with the positions of the ladders and snakes that are distributed on it.		
Outputs	Output name	Datatype	Selection or repetition condition
	gameBoard	String	

Name or identifier	R6. Calculate the score of the match winner.		
Summary	The system should calculate the score of the player who win the match (The player who reaches the m*n square of the board).		
Inputs	Input name	Datatype	Selection or repetition condition
	winnerName	String	
General activities necessary to obtain the results	<ol style="list-style-type: none"> 1. Read the name of the winner match. 2. Calculate the winner match score based on game time. 3. Set the player on the scoreboard depending on his total score. 4. Show the player score. 		
Result or postcondition	A message with the name and total score of the match winner is displayed.		
Outputs	Output name	Datatype	Selection or repetition condition

	playerScore	String	
--	-------------	--------	--

Name or identifier	R7. Show scoreboard.		
Summary	The system should display a message with the updated scoreboard after a match is over.		
Inputs	Input name	Datatype	Selection or repetition condition
	none	none	none
General activities necessary to obtain the results	1. Display a message with the updated scoreboard after a match is over (names and scores of the matches winners).		
Result or postcondition	A message with the updated scoreboard is displayed.		
Outputs	Output name	Datatype	Selection or repetition condition
	scoreBoard	String	