**Tarea integradora I**

|  |  |
| --- | --- |
| Customer | *Snakes and Ladders Inc* |
| User | Player(s) |
| Functional requirements | * + **R1:** Show the main options menu.   + **R2:** Construction of the game grid.   + **R3:** Display the game board with player’s positions.   + **R4:** Show the game options menu.   + **R5:** Show the game board with ladders and snakes.   + **R6:** Simulate rolling a die.   + **R7:** Linear movement of the player.   + **R8:** Movement through stairs.   + **R9:** Movement through snakes.   + **R10:** Completion of the game.   + **R11:** Calculation of the player's final score based on game time. |
| Context of the problem | The game must present the user with a grid or table of ***n*** rows by ***m*** columns, within which there are ***s*** snakes and ***e*** ladders.  The squares must be identified through numbers and the numbering begins at the bottom left of the board, having a straight movement to the final column, then moving up a row and reversing the direction of movement. |
| Non-functional requirements | * The game must be a text interface per console. |
| Process requirements | * Linked lists should be used for the game grid. * Preloaded Java data structures should not be used. * Cycles should not be used. * All ladders and snakes should be modeled as connections between nodes of the linked structure. |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R1:** Main menu display | | |
| Summary | The menu must be displayed to the user with the corresponding options for him to decide himself | | |
| Inputs | **Entry name** | **Data type** | **Selection or repetition condition** |
| option | Int | Number between 1 and 2 |
| General activities required to achieve results | When entering the menu, the player must choose between the two options that will be displayed, either to start the game or to exit the menu. | | |
| Result or Postcondition | The game will be started or quit. | | |
| Outputs | **Output name** | **Data type** | **Selection or repetition condition** |
| message | String | The option entry was among the required parameters |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R2:** Construction of the game grid. | | |
| Summary | It receives the size values of the game grid, as well as the number of snakes and ladders to be generated, and starts the general game board. | | |
| Inputs | **Entry name** | **Data type** | **Selection or repetition condition** |
| rows | Int | Integer number greater than 1. |
| Columns | Int | Integer greater than 1. |
| snakes | Int | Integer greater than or equal to 0. |
| ladders | Int | Integer greater than or equal to 0. |
| General activities required to achieve results | Add a number (rows\*columns) of linked levels in a linear structure.  Assign the relevant connections of the respective ladder snakes (randomly generated positions).  Randomly select the game figures that will represent the players. | | |
| Result or Postcondition | Initialized game board. | | |
| Outputs | **Output name** | **Data type** | **Selection or repetition condition** |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R3:** Display the game board with player’s positions. | | |
| Summary | The game board with all player’s position must be shown at the beginning of the game as well as each time a player makes a move. | | |
| Inputs | **Entry name** | **Data type** | **Selection or repetition condition** |
|  |  |  |
| General activities required to achieve results | The data structure will be traversed saving the info of each level in a single String which will finally be printed. | | |
| Result or Postcondition | The game grid will be displayed. | | |
| Outputs | **Output name** | **Data type** | **Selection or repetition condition** |
| board | String |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R4:** Show the game options menu. | | |
| Summary | The play menu must be displayed to the user with the corresponding options for him to decide what he wants. | | |
| Inputs | **Entry name** | **Data type** | **Selection or repetition condition** |
| option | int | Number between 1 and 2 |
| General activities required to achieve results | When entering the menu, the player must choose between the two options that will be displayed, either to throw dice or to show snakes and ladders. | | |
| Result or Postcondition | The player will either start its movement or simply will be displayed the snakes and ladders board. | | |
| Outputs | **Output name** | **Data type** | **Selection or repetition condition** |
| message | String | The option entry was among the required parameters |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R5:** Show the game board with ladders and snakes | | |
| Summary | The game board with all snake’s and ladder’s positions must be displayed. | | |
| Inputs | **Entry name** | **Data type** | **Selection or repetition condition** |
|  |  |  |
| General activities required to achieve results | The data structure will be traversed saving the info of each level in two different Strings (one for ladders and the other for snakes). | | |
| Result or Postcondition |  | | |
| Outputs | **Output name** | **Data type** | **Selection or repetition condition** |
| board | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R6:** Simulate rolling a die | | |
| Summary | Returns, randomly, an integer between 1 and 6. | | |
| Inputs | **Entry name** | **Data type** | **Selection or repetition condition** |
|  |  |  |
| General activities required to achieve results | Generate the number. | | |
| Result or Postcondition |  | | |
| Outputs | **Output name** | **Data type** | **Selection or repetition condition** |
| result | Int | Number between 1 and 6 |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R7:** Move the player | | |
| Summary | Receive the number of squares to run and the player who will do it. | | |
| Inputs | **Entry name** | **Data type** | **Selection or repetition condition** |
| n | Int | Positive integer |
|  | Player | Char | It must be a special character ("\*! OX%$#+&") |
| General activities required to achieve results | Find the square the player is in.  Get the number given by the die.  Move the player the designated number of squares.  Perform the movement through ladder or snake if appropriate. | | |
| Result or Postcondition | The player will possibly be in a new position. | | |
| Outputs | **Output name** | **Data type** | **Selection or repetition condition** |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R8:** Move the player through ladders. | | |
| Summary | It puts the player in a higher position on the board in case the player is standing on an entrance stair. | | |
| Inputs | **Entry name** | **Data type** | **Selection or repetition condition** |
| Player | Char | It must be a special character ("\*! OX%$#+&") |
| General activities required to achieve results | It must be compared if the final position of the player, after having rolled the die, is equal to the position of a ladder, then the player will be led to that corresponding square. As each level node, which contains an entrance ladder, will be pointing its exit node level, it will simply translate the player to that level in a single operation. | | |
| Result or Postcondition | The player will be in a position ahead of the original. | | |
| Outputs | **Output name** | **Data type** | **Selection or repetition condition** |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R9:** Move the player through snakes | | |
| Summary | It puts the player in a lower position on the board in case the player is standing on an entrance snake. | | |
| Inputs | **Entry name** | **Data type** | **Selection or repetition condition** |
| Player | Char | It must be a special character ("\*! OX%$#+&") |
| General activities required to achieve results | It must be compared if the final position of the player, after having rolled the die, is equal to the position of a snake that connects to another position further back on the board, then the player will be led towards that corresponding square. As each level node, which contains an entrance snake, will be pointing its exit node level, it will simply translate the player to that level in a single operation. | | |
| Result or Postcondition | The player will be in a position further back than the original. | | |
| Outputs | **Output name** | **Data type** | **Selection or repetition condition** |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | **R10:** Game Completion | | |
| Summary | The game ends due to a player who reached the last square. | | |
| Inputs | **Entry name** | **Data type** | **Selection or repetition condition** |
| n | Int |  |
|  | Player | Char | It must be a special character ("\*! OX%$#+&") |
| General activities required to achieve results | It needs to be compared whether the number thrown, and the number of squares moved by the player match exactly the last square on the board. If so, when the player is in the last square of the board, then a congratulations message will be printed. | | |
| Result or Postcondition | The game is finished. | | |
| Outputs | **Output name** | **Data type** | **Selection or repetition condition** |
| message | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | R11: Calculating the player's score | | |
| Summary | Through the formula provided (P = [600 - <seconds>]/6), the score of the players will be estimated at the end of the game. | | |
| Inputs | **Entry name** | **Data type** | **Selection or repetition condition** |
| time | int | Positive integer |
| General activities required to achieve results | Calculate the new player’s score. | | |
| Result or Postcondition | The score will be saved. | | |
| Outputs | **Output name** | **Data type** | **Selection or repetition condition** |
|  |  |  |