[R] [L]ELEN2004/ELEN2020 Software Development I Assignment

*Introduction *Snakes And Ladders Snakes and ladders is a multiplayer board game on a grid with dimensions (n,n).

Take for example a 10 *by* 10 grid numbered from 1 to 100 in order. To make things a little bit more interesting, let us
terminated with a draw.
That is how the game works in a nutshell.

*Board and Game Design The number of snakes and ladders are dependent on the binary encoding of my student num

*Possible game termination outcomes The game has one of three outcomes, either it is terminated by a single player re

*Considerations of design Ideally we want the game to go as long as possible to afford us the chance of getting each of

*The Solution The idea here is to use the object oriented design paradigm. In this paradigm, entities that are interact

Player: A player who's attributes might include but not limited to the player's position, player's recent/current round and

Snake and Ladder Objects: I put this in the same bullet point as they serve the same purpose just in opposite direction

Tile: The tile object is probably my favorite designed object, a tile represents each block on the grid or board. The interest

Board: This is the biggest piece of the puzzle. The reason the board is an object of its own is because we can read in mult

Each board can be seen as a game it self. The main section in the implementation will consist of three parts only and t

Reading input: I imagine it might be difficult to in interact directly with the input file. In this section, I will read in the i

Boards initialization: This is the next middle piece of the whole puzzle and very crucial, no mistakes can be afforded her

Playing the game: This is adding meat to the bones. At this point we've read in the data and initialized all the boards. T
Upon playing the game, we proceed to the very next board if any exists, otherwise we terminate.

*Results In this section I discuss the results that came from our codes in the form of tables. The results here are a res

*Results on boards designed from the student number Below are tables for 3 separate trials. Here we expected a winn

[scale=0.5]threeOrig.png

If you look at the input file, I have designed 2 boards of different sizes and different number of players. This is approximate
[scale=0.5]sixOrig.png

*Results on boards designed to test conditions of winning In this section we doctored the design of the boards and the
[scale=0.45]three.png
[scale=0.45]six.png
Here we can then deduce that upon many trials and most of which are not included in the report, we can confidently say th

*Discussion *The Approach The aim is to complete a given task by minimizing the amount of code one will write. Hav
current implementation is that if I add a function to the snake class, I need to add the exact same function to the ladder cla
One more improvement I could have added is dynamically creating the input of snakes and ladders and the writing it to a fi

This assignment wasn't at all easy, the only assumptions that were made were ones in relation to the time taken to complet

*Conclusion All boxes have been checked, code works exactly as intended, or so I believe. There are many design patt

*References

*Appendix
*A: Project Time Management

*B: Flow Chart [scale=0.5]SnL.png