Project Part A: Finding Legal Moves

Chris Leckie and Sarah Erfani
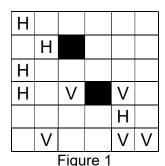Last updated: 18 March 2017

Task

The aim of this part of the project is to design and implement a Java 1.8 program to count the number of legal moves available to each player in a given board configuration for the game. This will help refresh your skills in Java, and provide you with important infrastructure that will be needed by your game playing agent in Part B. Before you read this specification, please ensure you have read "Project Specification: Rules of the Game of Slider".

Your program should read a board configuration from the standard input.
- The first line of input contains an integer corresponding to the dimension of the board. You can assume the given dimension of the board will be $N > 3$.
- The remaining lines of input will specify the contents of each board position, one row per line, starting from the top-left corner of the board. The contents of a board position will be encoded as one of the characters 'H', 'V', '+', or 'B', corresponding to a Horizontal piece, a Vertical piece, a free cell, or a Blocked cell, respectively. The contents of consecutive board positions in the input will be separated by a single whitespace character.

For example, the input below corresponds to the board position depicted in Figure 1:

```
6
H + + + + +
+ H B + + +
H + + + + +
H + V B V +
+ + + + H +
+ V + + V V
```


Figure 1

The output from your program should be on two lines, with the format:

```
numLegalHMoves
numLegalVMoves
```

where
- `numLegalHMoves` is a non-negative integer corresponding to the number of legal moves for Horizontal pieces from the given board configuration
- `numLegalVMoves` is a non-negative integer corresponding to the number of legal moves for Vertical pieces from the given board configuration

For example, given the board state depicted in Figure 1, your program should produce the output:

```
9
9
```

Assessment

Part A is intended as a progress submission for the larger project. The aim of Part A is to motivate you to get started on the project. Consequently, marks will be awarded for the quality of your code, the correct operation of your program, and the efficiency of your algorithm. Your program should include comments where appropriate. You will also need to provide a pseudocode description of the algorithm you used, and discuss its time complexity.

Part A will be marked out of 8 points, and contribute 8% to your final mark for the subject. Of the 8 points, 2 points will be for the quality of your code and comments, 4 points will be for the results of testing on a set of test cases, and 2 points will be for the pseudocode description of your algorithm and the discussion of its time complexity. Note that even if you don't have a function that works correctly by the time of the deadline, you should submit anyway, since you may be awarded some points for a reasonable attempt at the project.

Questions and answers pertaining to the project will be available in the FAQ page on the LMS and will be considered as part of the specification for the project.

There should be one submission per group. You are encouraged to discuss ideas with your fellow students, but your program should be entirely the work of your group. It is not acceptable to share code between groups, nor to use the code of someone else. You should not show your code to another group, nor ask another group to look at their code. You may make use of the library of classes provided by the Russell and Norvig textbook site if you wish, provided you make appropriate acknowledgements that you have made use of this library (a copy of their library of classes will soon be provided on the MSE Student Unix servers). If your program is found to be suspiciously similar to someone else's or a third party's software, you may be subject to investigation and, if necessary, formal disciplinary action.

Please refer to http://academichonesty.unimelb.edu.au/ if you need further clarification on this point.

Submission

The submission deadline for Part A is 4.00pm Friday 7th April 2017. Please see the separate document partA-submit-2017.pdf, to be released closer to the deadline, for detailed information on submission.