BLM19202E Data Structures PROJECT #1
SOLO NOBLE Game with Multi Linked List
(18 March 2022, 23:59)

In this project, you are expected to develop a program in Java programming language that imitates the game known as "SOLO NOBLE", using linked list data structure.

HOW TO PLAY THE GAME

Game board is an *n***n* sized matrix as illustrated in Figure 1 (*n* shall be an even number). The specialized game fills the entire board with pegs except for the central four holes. The objective is, making valid moves, to empty the entire board.

You can see a board of size 8*8 below. An example move is shown in Figure 2; peg located in D2 aimed to move to D4, peg in D3 cell will be removed by this movement (Figure 3). Another move is given in Figure 4; the peg in B2 cell aimed to move to D2, peg in C2 cell will be removed by this movement (Figure 5).

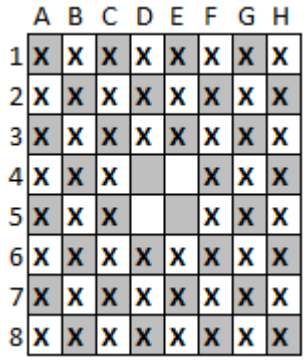

Figure 1- Board before the game

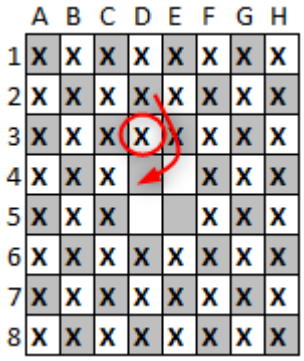

Figure 2- First movement plan

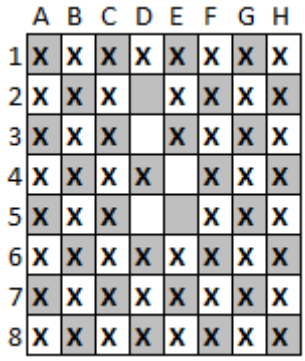

Figure 3- After first movement

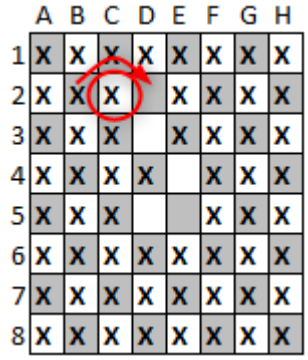

Figure 4- Second movement plan

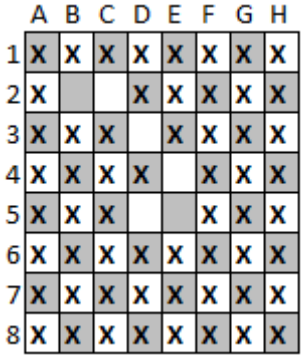

Figure 5- After second movement

APPLICATION DETAILS

You are supposed to use *n* linked lists in the implementation (*n* is the input size given by the user). Each linked list node will correspond to a cell on the board that has peg on. You should not store the empty cells.

The nodes of the linked lists must include at least the location (A1, A2, .., B1, B2,…) and the next node information. Additionally, the header nodes of the linked lists should also be connected in a linked list. That is, the header nodes should also have the information for the header of the next linked list. Your multi-linked list structure will be similar with the illustration in Figure 6. Corresponding game board is shown in Figure 7.

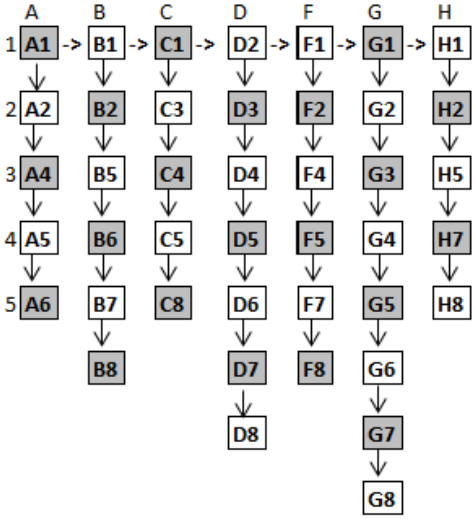

Figure 6- Multi linked list illustration

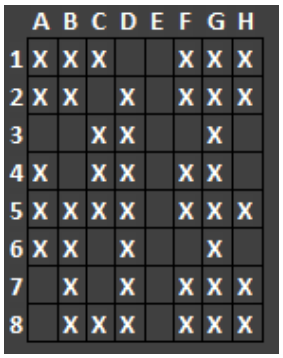

Figure 7- Corresponding game board

Game will be played in Java Frame; application interface must be prepared properly like in Figure 8 and Figure 9. You must design and implement your own GUI.

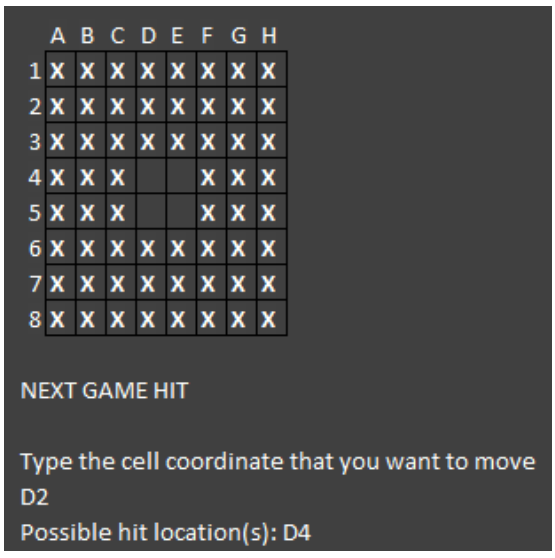

Figure 8

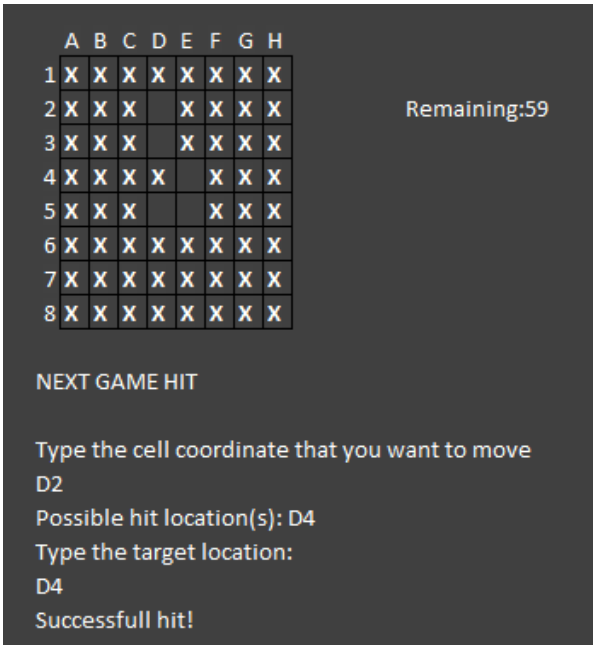

Figure 9

Current game board will be shown to the Frame during the game. User should be prompted to decide the hits.

1. Firstly user types the location of the peg that will move.
2. According to the first coordinate information, application will find the possible move locations by searching the linked list. Results will be printed on the frame.
3. If there is no move option, an error message will be printed like "You can't move the peg in C2", than the next input is expected. (Go to step 1)
4. User selects one of the possible hit locations listed in step 2.
5. If the selection is valid, then linked list and the game board will be updated and "Successful hit!" message will be printed. **Remaining peg number also will be calculated by counting nodes in entire linked list each time and printed to the screen.**
6. If the selection is not valid, then print a message like "Invalid move", go to step 4 and try again.
7. If there is no available move, a message will be printed like "Game over".

All operations will be done using linked lists. Please don't use arrays instead, **the projects prepared with array or other data structures will not be evaluated**.

GRADING:

1. User interface (30 points)
2. Multi linked list class (15 points)
3. Update linked list (55 points)

BONUS (15 points): (a) (10 pts.) You can design your GUI with button/circle/rectangle objects to represent a cell. You can first the click on the cell coordinate that you want to move, and then click on the target cell coordinate. **(b) (5 pts.)** Add a new button or menu to start a new game.

Notes:
By the due date, please submit the source code of your program, on the submission on LMS. And please name your files as follows: "*yournamesurnameXXX*.java", for example "BernaKirazMainClass.java".

Note that projects submitted after the project's due date will not be accepted and evaluated. Please keep this in mind and **promptly start working on your projects**!

You are required to exhibit an ***individual effort*** on this project. Similarity test is applied with all projects. Any potential violation of this rule will lead everyone involved to failing from all projects.

Good luck ☺