Program Assignment

1. What is Omok (Gomoku)?
In Korea, Omok (五目), also called Gomoku (五目) in Japan and Wuziqi (五子棋
) in China, is a traditionally played with Go pieces on a Go board (15x15). Basically, the rule is almost same as Tic-Tac-Toe.

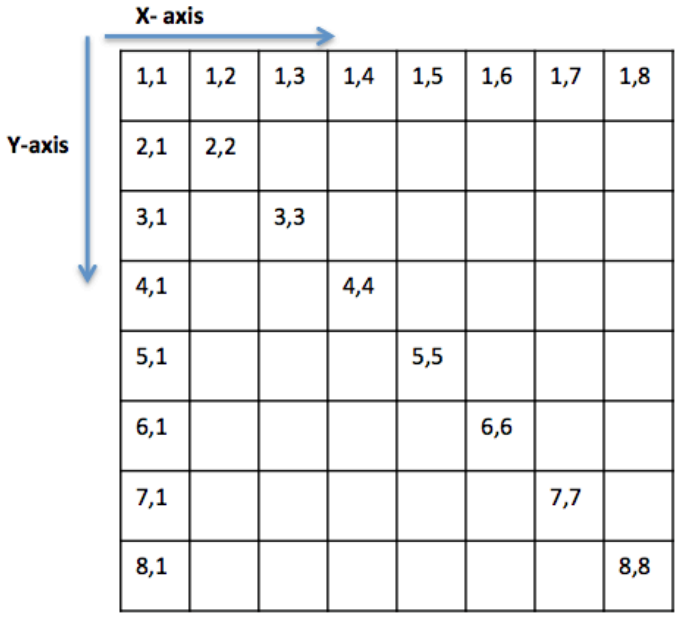
2. Objective
Your goal is to write a simple AI (client) program to win the game.

3. Game Rule
Here are some ground rules:

- The Omok is a turn-based game, and it requires two players.
- The game is played on a 15x15 grid matrix.
- The x-axis of the row is indexed from 1 to 15.
- The y-axis of the column is indexed from 1 to 15.
- A player who picked the black Go piece plays first and the opponent who picked the white Go piece plays afterward.
- Once you place the piece, you should wait until the opponent move.
- You can place your move anywhere in the board if the space is empty.
- To win the game, you need to make an unbroken row of five Go pieces vertically, horizontally, and diagonally.

4. Brief description of the implemented game
Here are some brief descriptions of the implemented game:

- The game is designed a GUI-based server and client.
- The game is designed simple local TCP/IP network using Java.
- To start the game, you need to run a server first and run two clients. The first connector to the server will be Black, and the following will be White. Black plays first, then white plays afterward. (Generally, black player takes advantage of the game because it starts first.)
- Once you start the game, you need to place your Go piece by assigning a two-dimensional grid coordinate (x, y). For example, if you want to place a piece in the second row fifth column, the grid coordinate of the piece is (2,5). If you want to place a piece in the forth row second column, the grid coordinate of the piece is (4,5). See the following figure for the details.
- If Black wins, the server will send a message “WIN BLACK” and print out “black wins”, and vice versa.
- The server and clients are automatically shut down after finished the game.



4. Program structure

- Java package: omok_csce625 (server)
  - MainControl.java (the game server)
  - Display.java (Layout of the game board)
    - DollType[][] is a two-dimensional array variable that defines Go board such as array size and grid size, and etc.
    - DollType is a enumeration variable is consist of 'Black, White, and Empty'.
  - GameBoard.java (game rule and validation check)

- Java package: omok_client_package (client)
  - OmokClient (the game client)
  - Display.java (Layout of the game board)
  - GameBoard.java (game rule and validation check)
  - **OmokAI.java (AI class)**
    - getNextMove(DollType[][] board, DollType myDollType) is a string type method that is defined in OmokAi class. Return type of the getNextMove is a string.

(Appendix)

In OmokClient.java,

```
while (true) {
          input = in.readLine();
           …

          } else if (input.startsWith("ENTER ")) {
                    System.out.println("Waiting for a piece position (e.g. 2,3):");


                    ////////////////////////////////////////////////
                    //select next piece location ///////////////////
                    //and then send the location to the server ///
                    //e.g. sr = "2,3" for the location x=2, y=3 //
                    ////////////////////////////////////////////////
                    sr = omokAI.getNextMove(gameBoard.getBoard(), myDollType);

                    out.println(sr);
          } else if (input.startsWith("WIN ")) {
                    System.out.println("Game Ends. Systems ends.");
                    System.exit(1);

 ...
```

In OmokAI.java,

```
public class OmokAI {
          public String getNextMove(GameBoard.DollType[][] board, GameBoard.DollType myDollType) {
                    //Find best location for next piece based on the current board

                    return "2,3";
                    //e.g. return "2,3" for the location x=2, y=3
          }

}
```