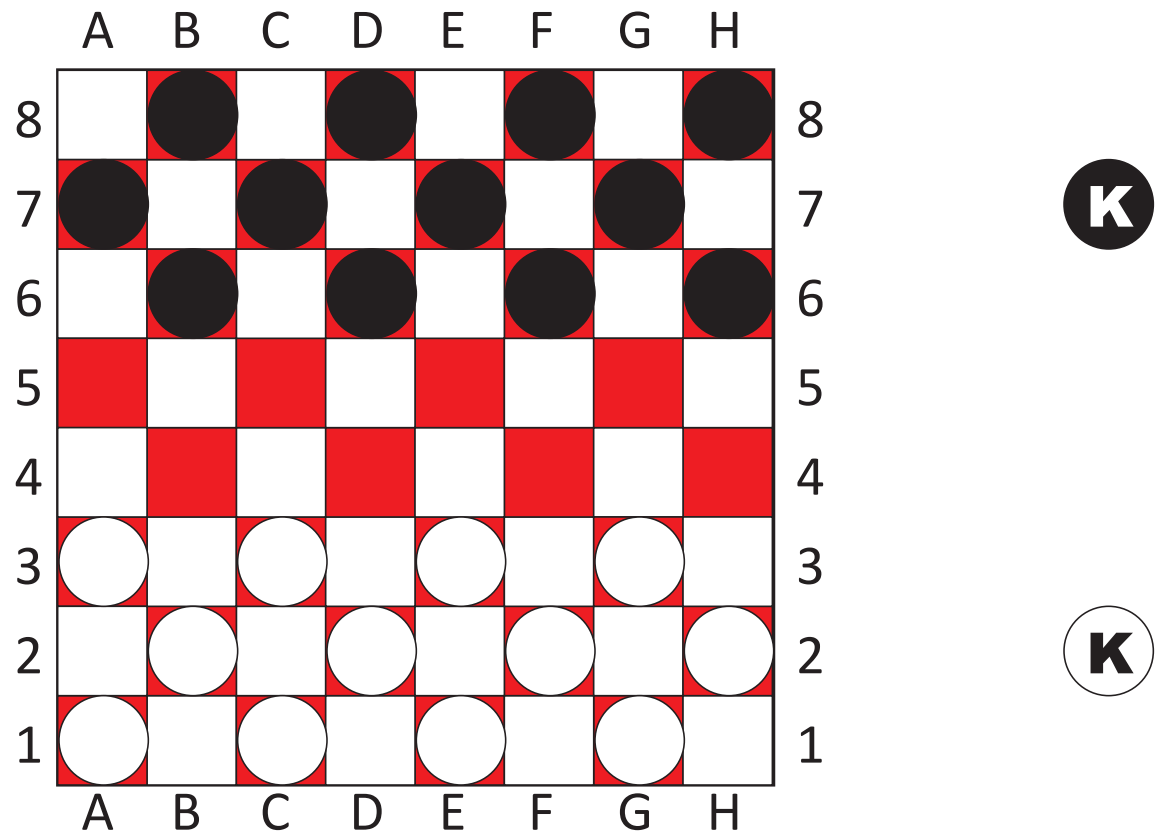
CS 2ME3 and SE 2AA4: Assignment 2 January – April 2014

Due: 18 March, 2014

Assignment 2 builds on the work you did in Assignment 1.



Repeated from Assignment 1.

6. In order to help you with the decomposition, here is some indication of what is to come in assignments 2 and 3:
   6.1. Assignment 2 will require you to be able to move pieces on the board. The moves have to be legal moves. You can move pieces using a graphical interface, or by using accepted checkers terminology, i.e. E3-D4 (move the piece on E3 to its new position, D4).
   6.2. Assignment 3 will require you to provide two modes of operation: 2 player checkers where 2 people can play against each other; or 1 player against the computer. The latter mode will require you to include an automated mode in which algorithms are used to determine the moves for the computer.

Specific requirements for Assignment 2.

1. Include choices that enable users to:
   1.1. Start a game from original starting positions.
   1.2. Start a game from a previously stored state (the state should be saved within a file – and you can decide whether you want to allow more than 1 saved game).
   1.3. Make moves – you don't need to be able to play a complete game (yet) – just move pieces from one position to another. The moves must be legal moves. You should be able to make moves that: simply move a piece to another square; jump the opponent's piece (so that piece is removed from the board); convert a piece to a "king"; move kings in both directions (forwards and backwards). You can decide how you want the user to indicate moves – graphically or by code (E3-D4 etc), or both.
   1.4. Save a game to be resumed later.
2. The deliverables for the assignment include:
   2.1. Extensions to your existing design documents. Make sure you find a way to tell the reader (and yourselves) what has changed in your design. A revision history is a good idea.
   2.2. Extensions to your existing code. Make sure you find a way to tell the reader (and yourselves) what has changed in your design. A revision history is a good idea.
   2.3. Include a test report document that records how you tested your application.

Grading.

The assignment will be graded out of 50.

30 marks for the design document (description of changes from assignment 1 – 5, decomposition – 10, public interface – 4, uses relationship – 2, private implementation – 4, traceability – 2, evaluation of the adequacy of the design – 3).
10 marks for the code (description of changes from assignment 1 – 4, layout, variable names, comments, etc – 4, and quality of the code – 2.
10 marks for testing.

Special note on documentation.

When you did assignment 1 you had not yet seen ways of documenting MIS and MID (and Module Guides). So, you are unlikely to have prepared your documentation for assignment 1 using preferred ways of displaying MIS and MID. Assignment 3 will expect documentation in line with (not necessarily exactly the same) as described in class. You can start making that change now if you like, or you can wait until assignment 3.