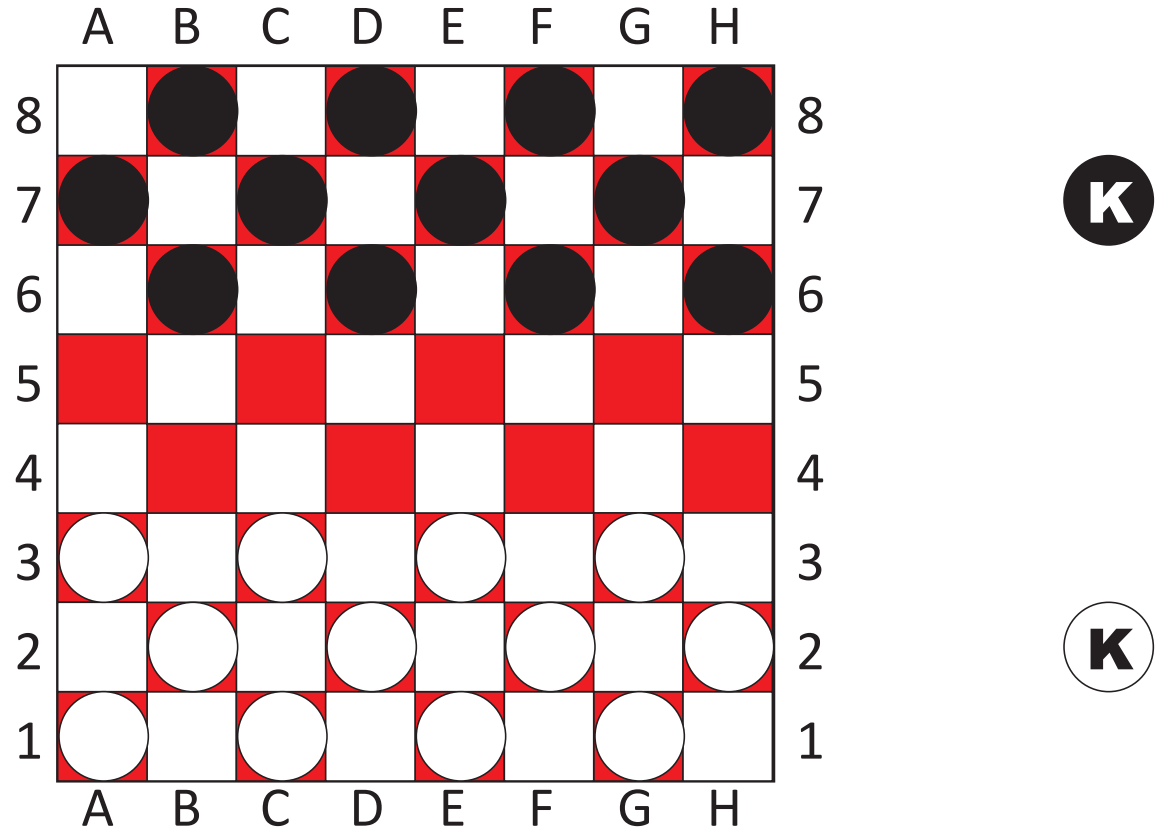
CS 2ME3 and SE 2AA4: Assignment 1 January – April 2014

Due: 28 February, 2014

Consider the following American Checkers board. It has been set up for the start of an American Checkers game. (The game is also referred to as English Draughts.) There are other variations on the game, but we will confine ourselves to 8-by-8 American Checkers/English Draughts. On the side of the board you can see two examples of "kings" or "crowned" pieces. (For rules of the game there are numerous descriptions on the internet.)



This assignment consists of generating the graphical representation of a checkers board, and being able to specify initial piece positions.

Specifically:

1. Initially set up an 8-by-8 checkers board with dark and light squares. There must be a light square in the bottom right corner. The normal convention is to label the columns as A, B, …, H starting from the left, and the rows as 1, 2, …, 8 starting from the bottom.
2. The user shall be able to set up an initial position of pieces on the board by specifying either the standard opening position (as shown in the figure), or by specifying positions for pieces using a notation such as A7=B (black piece on A7), or E3=W (white piece on E3), or F8=WK (white king on F8), etc. Users shall be warned if the position is illegal, and pieces must not be placed on illegal squares (white/light squares). A maximum of 12 white pieces and 12 black pieces may be placed on the board. Alternatively, pieces may be placed on the board using a graphical interface.

3. In the case that the standard opening position is not used, there shall be a way for the user to indicate that set up is complete.
4. The deliverables for the assignment include a design document as well as the code. The design document should include:
   4.1. a description of the classes/modules you have decided to use in your application, and your explanation of why you have decomposed the application into those classes
   4.2. for each class, a description of the interface (public entities), and make sure that there is a description of the semantics (behaviour) of each public method in the class, as well as a description of the syntax
   4.3. a view of the uses relationship
   4.4. include a trace back to requirements in each class interface
   4.5. for each class, a description of the implementation (private entities), including class variables. Include enough detail to show how the class variables are maintained by the methods in the class
   4.6. an internal review/evaluation of your design
   4.7. Document the code so that it is clear how the code follows its design, and also explain design decisions in the code that were not included in the design document.
5. Include a test report document that records how you tested your application (we have not discussed testing yet – so you are on your own with this document ☺).
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 wither 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.

Grading:

The assignment will be graded out of 50.

35 marks for the design document (decomposition – 10, public interface – 8, uses relationship – 4, private implementation – 6, traceability – 4, evaluation of the adequacy of the design – 3).

15 marks for the code (layout, variable names, comments, etc – 5, and "correctness" – 10).