CS 6613 **Fall 2013**
Project: Interactive *M,N,K* (4,5,4) Game **E. K. Wong**

Total points = 100. Extra credits: up to a maximum of 25 points.

Project Description: Design and implement an *Interactive M,N,K* (4,5,4) Game for a person to play against a computer. The game board consists of a 4 *X* 5 grid (as shown below). The computer and the human players take turns to mark the cells. The first player who gets 4 consecutive marks in a row, in a column, or in a diagonal wins. The diagonal could be a 45 degree diagonal or a 135 degree diagonal. Your program should have the following features:

- The computer uses the *X* and the human uses the *O* mark. The human player can choose to go first or second.
- The program declares who the winner is by displaying a message on the screen.
- The program stops when all the cells are marked and there is no winner. In this case, the program displays a message declaring a draw.

				O
	X		X	
X	O	O		

In designing your program, you should use the *Alpha-Beta-Search Algorithm* as discussed in lecture (and in the book.) If the computer cannot search all the way to the bottom in 10 seconds or less for the first move, you should cutoff the search early at some level and use an evaluation function to estimate a utility value for the board position (See Section 5.4 of text book.) Propose an evaluation function for this game if it is necessary to use cutoff. Exercise 5.9 on page 197 of the book contains an evaluation function for the 3 x 3 *tic-tac-toe* game. You can extend or modify the evaluation function for this project if suitable.

You can use Python, C++, C#, or Java to do the project. If you plan to use another language, send me an e-mail first.

Extra Credits: The following parts are *optional*. You can implement one or more of the following to get extra credits. The maximum number of points you can get as extra credits is 25, regardless of how many parts you implement.

- [15 points] Design and implement a graphical user interface to display the *X* 's and *O's* on the 4 x 5 grid. Human can mark the cells by using a mouse.
- [10 points] In the design above, the computer plays the best strategy. Design your program in such a way that the computer does not always play the best strategy so that it is easier for the human to win. The human player can enter the level of difficulty (say from 1 to 3).
- [10 points] If you do use an evaluation function as described above, it will be marked as correct if the evaluation function is a reasonable one. If your evaluation function is very good in terms of providing an estimate for the utility value, you may get up to 10 points in extra credits.

What do hand in:

1. Source code for your program. Documentation and in-line comments are required for your source code (Points will be taken off if you do not have this.)

2. An MS Words, PDF, or plain text file that contains:
   a. Instructions on how to compile and run your program.
   b. A high level description of your design and algorithm. If you use cutoff and an evaluation function, describe how the cutoff and the evaluation function work. If you implement the different levels of difficulty part for extra credits, describe how it works. No need to describe the basic *alpha-beta* algorithm (which I know.) However, if you modify the alpha-beta algorithm, you need to describe your modifications.