# Assignment 1: The Reversi Game

This assignment is inspired by the TAI Game Playing Championship held for many years by prof. Jan Eric Larsson from the EIT department at LTH.

## In short:

Your task in this assignment consists in writing a program playing Reversi game.

## In detail:

The rules of reversi can be found e.g. in [http://en.wikipedia.org/wiki/Reversi](http://en.wikipedia.org/wiki/Reversi" \t "_blank). This version will be sufficient for our purposes. We will be playing according to the modern, so called Othello rules, i.e. diagonal beginning 2x2 and no limit of disks for players.

Two example sites (there are more, of course) giving you the possibility to get acquainted with the game interactively are[http://www.samsoft.org.uk/reversi/](http://www.samsoft.org.uk/reversi/" \t "_blank) and [http://www.haxx.se/home/games/othello/index.shtml](http://www.haxx.se/home/games/othello/index.shtml" \t "_blank), but there are lots more interesting resources on the web, of course.

In order to get the assignment accepted, your program must:

1. in every game situation suggest a legal move;
2. implement at least the classical mini-max adversarial search algorithm, preferably with the alpha-beta pruning introduced;
3. provide a response within a given, user-predefinable time limit;
4. (optional but strongly recommended) visualize the current game state. An ASCII-grid will be more than sufficient;
5. (optional) bookkeep the time used for reasoning, so that the time of the play may be traced and used for, e.g., detecting a time-over.

Please use the standard move notation, both for input and output, i.e. XY where X is one of 1-8 (row) and Y is one of a-h (column).

Moreover, the program must come with a reasonable description (i.e. the submitted report) regarding its design (what algorithm have you used, what kind of extensions or heuristics, if any, where in your code can one find the important elements used in your best move calculations, how is the game represented, etc.), usage (how can one run it in order to test it) and where can it be found on the student computer system, i.e., the access path. (**IMPORTANT NOTES:** The path quoted must include your username /e.g. ~/TAI/a1/myprog is NOT GOOD! Why?/ and you have to make sure it is accessible for the person evaluating your submission. Check the Linux chmod command for that purpose, please. The code should be accessible both for reading /i.e. source/ and running /i.e. executable/ without any additional demands like, e.g., compilation or usage of some particular IDE /read Eclipse/.) Please, typeset and format your report consistently using Latex for instance.

You may consult the code in the textbook code repository (or any other implementations), but the code you hand in must be primarily your work. You do not need to provide all or any code printout in the report - the code is available in your solution directory anyway - but only comments on its interesting parts.

The deadline for this assignment is Thursday, 11th of February, by which you MUST file in your working solution to the assignment, fulfilling requirements 1-3 (i.e., stand-alone testing should be possible).

**Important remark**: Remember that the time frame allotted to this assignment is approximately three days of your work, so please don't overdo it: a decent program will suffice to get a pass. You are expected to write a simple program that uses a well-known adversarial search algorithm, provide it with some evaluation functions for guiding the search, limit its search depth depending on time flow, and finally present the game in a simple manner.

If you decide to use someone else's code (e.g. some library found on the web), please mention it both in the code and in your report: it is a matter of academic honesty. Lund University is committed to fighting every case of dishonesty or plagiarism.

The report should be sent to tai@cs.lth.se as an attachment to a mail message with the following subject line:

Assignment 1 by username1 and username2,

where username is your user name in the student computer system, like ada09jam.

The resulting programs should remain in your directory until you have been notified of the result, e.g. on the notice board and/or web or by e-mail. You may expect that your report and implementation will be examined within two weeks. If your report or implementation is unsatisfactory you will be given one chance to make the corrections and then to hand it in within a week after you have been notified (on the notice board and/or web or by e-mail).

Please send any requests for clarification to Jacek.Malec@cs.lth.se or visit him personally at his office, preferably on Thursdays after the lecture, between 15 and 16.