Increasing Assignment Motivation Using a Game AI Tournament

Øyvind Kolås and Ivar Farup
Gjøvik University College
P.O.Box 191
N-2802 Gjøvik, Norway
oeyvindk@hig.no, ivarf@hig.no

Introduction

Programming assignments are usually strict, which might be demotivating. In this note we describe a rather successful attempt at giving students in a secondary algorithms course a somewhat open assignment that required them to evaluate and use a broad range of programming concepts and ideas.

Othello is a zero-sum board game with simple rules but advanced strategy, and AIs for this game is a well researched topic [3, 1]. Creating a successful AI involves considerations of programming topics ranging from data structure optimization, through sorting, hashing and recursion to tree pruning and genetic programming.

The assignment

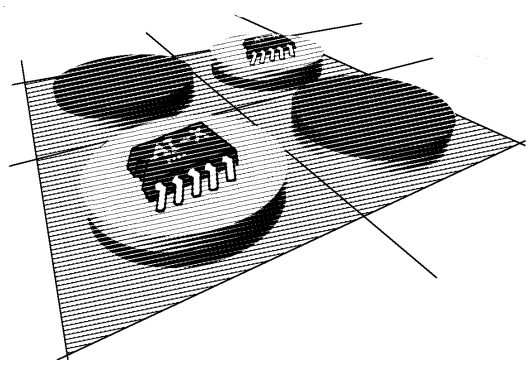
The assignment was to *individually* create a stand-alone application capable of playing a game of Othello.

The applications were to communicate on standard input and output through a simple text-based protocol. The protocol starts defining which color the AI should play, i.e., "`black`" or "`white`", followed by commands consisting of coordinates, e.g., "`a1`"–"`h8`", "`pass`" and "`quit`", separated by line breaks. Lines beginning with "`#`" are treated as comments.

To facilitate running the tournament, additional constraints were given: single binary file, less than 60 Mb resident size and maximum 2.5 minutes of CPU time spent calculating moves.

Precompiled tools for playing man vs. machine, machine vs. machine and running full tournaments were provided both with and without GUIs. The tools use the protocol above, and could thus be used by the students for, e.g., playing against their own AIs or playing their AIs against each other.

Some fast precompiled sample AIs ranging from plain stupid to quite strong were provided, thus making the students able to test their creations.



In order to pass the assignment, the most simple-minded sample AIs (not even considering future moves) had to be beaten. To encourage further interest, an award for the best AI beating the strongest provided sample was announced.

During the week after the AIs were turned in, a web page was continuously updated with the results from the continuously running round-robin tournament. For a full description of the assignment (in Norwegian) and the software tools developed, see [2].

Results

The assignment was evaluated by having the students fill in a form rating various aspects of the assignment. Even though the group was rather small, trends appeared in the results.

Surprisingly, *all* of the students rated this assignment as more motivating than "normal" programming assignments. This was ascribed to both the competitive aspect and the tools provided.

Some of the students would rather have preferred a group submission, but the ratings were not conclusive. Cooperation between individuals was observed, but fewer than hoped for researched and facilitated external information, even though references to good starting points on the web were given.

The majority felt that the level of difficulty matched the expertise one could expect them to have, whereas other ratings suggested that the needed skill level was not present. This was also indicated by the quality of most of the submissions.

Some of the students suggested that if they were provided with code for a basic AI, they could spend more time on advanced techniques, and less on basics such as checking for legal moves.

Conclusively, the experiment was successful and will be repeated next year. However, submissions will be accepted from pairs as well. Instead of providing code for sample AIs, it is probably advisable to have an even stronger emphasis on references and usage of relevant external resources, giving the students the opportunity to make a well founded choice of algorithms and data structures.

References

[1] Bruce Abramson Control Strategies for Two-Player Games *ACM Computing Surveys*, 21(2):137-161, 1989.

[2] Algoritmiske Metoder II. `http://www.hig.no/at/data/algmet2/`

[3] Noonian's Lab: Strategic Games: Theory & Programming. `http://home.pacifier.com/~noonian/GameTheory/games.html`