IA -Atari Go

Peter MOUEZA

May 25, 2010

In order to be used with main algorithms, a project of Atari-Go has been given. Specifically : we have to implement Min-Max and Alpha-Beta algos. We speak in English because of time sparing because

Technics and Developpment

lecture has imposed it; and all informaticians are supposed to speak English.

Chapter 1

User Manual

We suppose you use Eclipse to load the project The project is trunk with directory src, and package myPack. The main class is JeuConsole.java, so select it and Run with green left arrow. It requests a choice : set 2 (for now) Presently, it just prints the file src/myPack/snapfile.txt of poners to start the party with, under an array of -1 Whites, 0 not filled, or 1 Blacks.

Chapter 2

Assumptions

Because of the 2b th rule of TP_ d_IA.pdf , we consider first: are Whites capturing ?

- if no $\Rightarrow$ are the White captured
- if yes $\Rightarrow$ sweep away the gotten poners , then compute back are Whites captured, from this new situation.

Precision of the remarks about the 37 captured White stones: there are captured by Blacks because if Black move on bottom-left corner, the private Whites from any move.

Chapter 3

Rules

Let's see the most interesting class : Rules.java

Concepts of Image synthesis can give ideas for he concept of capture : outline of a stain.

This is done for each Connected Component(CC.java) : for each we look if tere is no an opposite color poner, which is a capture.

Chapter 4

Algorithms

4.1 Tree structure

It's virtualized by Vector type simulating a stack.

4.2 Depth First Search

4.3 Breadth First Search

Only in case where the time per move is imposed.

4.4 Min-Max

4.5 Alpha-Beta

Chapter 5

Conclusion

It has allowed us to see the mervelous world of A.I. and to have to write a real program has pointed out that it's not that easy to program algorithms which seem easy on the paper. And the real wrinklest think is to find an accurate eval function.

A possibility to enhance the behaviour would perhaps be to implement genetic algorithms which enable to select the best evaluation function, without to have the problem of local and global minimums problems. But, if still nowadays no efficient program has succeeded in Go game, it's probably that even this way is not a successful solution.

This topic has really interested me, so perhaps I will fill this topic a day(http://code.google.com/p/alma-go/source/checkout or

http://membres.multimania.fr/mouezapetero/atarigo.php) for further development.