AI Report

Go game (Wei Qi)
Artificial Intelligence

Author:
Thomas Wickham

Instructor:
Pr. Liqing Zhang

May 15^{th} 2012

Contents

Introduction

This report is about the GO contest of the AI course.

It contains detailed description of the structure of our AI, the algorithms we have implemented, description of some of our choices, and an analyze of the result we obtain in our tests.

1 Common Part

In this part, we will discut about the common work of the group.

Firstly, we will discuss our choice to use OCaml, a functionnal language. Then, we will see the differents algorithms we choose to implement and, last but not least, we will analyze our results.

1.1 The choice of a functionnal Language

1.1.1 Explanations

The classics way for programming this project was C or C++. It's the general choice, the choice that a lot of groups make.

Otherwise, these languages are lacking a lot on sides that matters from an strictly algorithmic point of view.

The most important problem with C-based languages is that they are state-based. A variable is a location on the memory and the programmer have to manage this memory himself, changing value by hand.

In this way, programming have a lot of side-effects, and the whole program is weakened by an approximative memory gestion.

Good programmers use the manual memory gestion as an advantage in order to gain efficiencity, but our project is not about memory gestion but Articifial Intelligence algorithms.

On the other way, functionnal programming avoid mutable variables and changing states but use functionnal application of contants values.

This permit a way of programming with very few side-effects, with less

memory management, and permit to be more focused on the algorithm itself.

1.1.2 Why OCaml

There is differents functionnal programming langagues : Lisp, Haskell, OCaml, Erlang...

We choose OCaml because:

- It is really fast
- It is Object Oriented ready
- It have many mature tools helping for project management
- And, mainly, because we were all already familar with OCaml

1.2 Algortihms

1.2.1 Upper bound for Confidence Tree (Uct)

1.2.2 Negascout

1.2.3 Genetic Algorithm

1.3 Results

2 Individual Part

2.1 Tasks

2.2 Personnal value in the project

2.3 Individual Conclusion

Conclusion

plap