Project Summary Document

Raphael Puget

17 février 2011

Application of Monte Carlo tree search to a two-player game
Advisor : Jonathan Shapiro

Aims The principal aim of this project is to developed an AI of a the 2-players game *GO*. The major issue with designing an AI of a *GO* player is that it is very difficult to implement an Heuristic to describe and to evaluate a position of a *GO ban* (the board of the game) because of the *simplicity* of the rules and the relative huge *depth* of the game.

That is why recently some people try to use the methods of Monte-Carlo in order to construct the min-max graphs of the positions of the game. The principal idea is to simulate a big number of complete games and use the probability in order to assign a value of each node of the tree of the positions in function of the number of game won.

The aim of the project is to implement these methods of Monte-Carlo to improve results found nowadays.

Objectives There are different objectives when working on this kind of project. In order to well understand the problem, it is very important to proceed of a good investigation of what has been already done and what was a success. Then, one not so small task is to implement the game of *GO* in `C++`. This task has to be well thought because the algorithm based on Monte-Carlo will run a big number of entire games. With a bad implementation of this engine, it could takes twice more times to run a game (or more). With this engine ready, the next step would be to implement one recent algorithm already developed in the literature. Thus, some improvement would be revealed if I achieve to improve the results of the previous algorithms. This would be the last part and objective of this project.

First objective : Proceeding of a good literature survey.
Second objective : Development of the engine of a GO game.
Third objective : Implementation of a well known algorithm.
Fourth objective : Improvement of the previous algorithm.

Outcomes The final outcome of this project would be the improved algorithm using Monte-Carlo methods and the engine of the GO game.