TCG Final Project Report

B04705003 資工四 林子雋

1. How to compile and run your code?

$ cd b04705003/source
$ make [search_attackMove|search]
$ cp search.exe [Search Folder]/search

2. What algorithms and heuristic you've implemented?

- NegaScout: at SearchEngine.h:262
- Transposition Table: at TranspositionTable.hh and TranspositionTable.cc
  - Thanks TA's black board code suggestion.
- Chance Node Search(I implement it but haven't thoroughly debug it): SearchEngine.h:424
- Heuristic Evaluation Function: SearchEngine.h:477, 512
  - I use static board evaluation function. (http://www.csie.ntnu.edu.tw/~linss/Students_Thesis/2011_06_29_Lao_Yung_Hsiang.pdf)
    - King: 6095
    - Guard: 3047
    - Minister: 1523
    - Rook: 761
    - Knight: 380
    - Canno: 420
    - Pawn: 200
- Attack Move Heuristic: SearchEngine.h:372
  - Intuition: the sooner you capture a piece, the better state you are in .
  - I will give some bonus(100 points) if an attack move is in move candidates.

3. The experiments you've done

Experiment Setting:

- Depth = 7
- NGS: NegaScout
- TTB: Transposition Table
- AM: Attack Move Heuristic

Versus Template Code

Algorithm	Win Rate	Draw Rate
NGS + TTB	30%	66%
NGS + TTB + AM	27%	66%

Explanation:

It seems that Attack Move Heuristic does not really work or Dark Chess randomness factor is enormous.

4. What I learned from this project?

- I found that when my AI is in almost winning state, I will start wandering. I think this behavior is derived from static evaluation function.

- I finally found out(2019/01/18 22:27) that transposition table will be “poisoned” if I set timeout if negaScout algorithm. For example, this timeout break will let transposition table believe the return negaScout value depth is 7, however, because timeout break, the value can actually be only searched depth, say 5.