

Theory of Computer Games (Fall 2021) Final Project

NTU CSIE

Due: 14:20 (UTC+8), January 13, 2022

Outline

- 1 Game Description
- 2 Homework Requirements
- 3 Submission and Grading Policy

Chinese Dark Chess (CDC)



- The game rule could be found here

https:

[//homepage.iis.sinica.edu.tw/
~tshsu/tcg/2021/hwks/rules.pdf](https://homepage.iis.sinica.edu.tw/~tshsu/tcg/2021/hwks/rules.pdf)

Chinese Dark Chess (CDC) - Score

Score

- Win: $1.0 + \text{Bonus}$
- Draw: $0.2 + \text{Bonus}$
- Lose: 0

Bonus

- $\text{Bonus} = \min(\max(\text{Diff} \times \frac{0.3}{8}, 0), 0.3)$
- $\text{Diff} = \# \text{ of my alive piece} - \# \text{ of opponent's alive piece}$

Chinese Dark Chess (CDC) - Score



Draw

- Red Score
 $= 0.2 + \min(\max((2 - 6) \times \frac{0.3}{8}, 0), 0.3)$
 $= 0.2 + 0 = 0.2$
- Black Score
 $= 0.2 + \min(\max((6 - 2) \times \frac{0.3}{8}, 0), 0.3)$
 $= 0.2 + 0.15 = 0.35$

Requirements

HW Requirements

- Implement an agent of Chinese Dark Chess (CDC) using **NegaScout** algorithm.
- Write a report.

Part I: CDC Agent (Due: 2022/01/13 14:20 UTC+8)

CDC Agent

In your CDC agent, you need to implement following requirements:

- NegaScout algorithm
- Time control
- Transposition table with using Zobrist's hash as hash function
- Dynamic search extension
- Iterative deepening Aspiration search
- Star0, Star1 or Star2
- Opening strategy
- Evaluate function with dark pieces
- At least one of the following:
 - ▶ Refutation tables
 - ▶ Killer heuristic
 - ▶ History heuristic

Testing

Testing

- Your CDC agent will be against the baseline 20 times.
- Take turns to move first.
- Your score is the sum of the scores of all games.

Target

Your CDC agent needs to get scores from baseline as much as possible.

Baseline

Provide the baseline's source code and executable file

Baseline's specification:

- Nega Max algorithm
- Search depth: 3

Testing

Limit

- Thread limit: 1 thread
- Memory limit: 4 GB
- Pre-processing time limit: 30 minutes

Game Setting

- Time limit: 900 seconds each game (measured by server)
- Threefold repetition rule: A game is considered draw if the same position occurs three times.
- 60-ply rule: If no capture has occurred in the last 60 ply (by both players), the game is automatically a draw.

Competition

Competition Date

- 2022/01/13 (Thursday) 14:20 (UTC+8)
- 2022/01/14 (Friday) 14:20 (UTC+8)
- 2022/01/17 (Monday) 14:20 (UTC+8)

You need to submit the code before each competition.

Game Setting

- Tournament format: Swiss-system
- Round: 3 for each day, but the scores will be accumulated.
- Time limit: **900 seconds each game**
- Threefold repetition rule: A game is considered draw if the same position occurs **three times**.
- 60-ply rule: If no capture has occurred in the last **60 ply** (by both players), the game is automatically a draw.

Competition

Swiss System

- Players are never eliminated.
- In every round, a player is paired against an opponent who has the same or similar points.
- Each round consists of 2 games with alternating first player.
- You can get S , $S \in \{0, 0.5, 1\}$ points for each game.
 - ▶ Win: 1.0 point
 - ▶ Draw: 0.5 point
 - ▶ Lose: 0 point

Competition

Rules

- You lose if your program crash 2 times in a game.
- In case of any violation, the referee has the right to decide the result of a game or if it may be restarted.
- No appeal against the decision of the referee is allowed.

Part II: Report (Due: 2022/01/20 14:20 UTC+8)

Report Structure

Your report should include but not limited to:

- Implementation
 - ▶ How to compile and run your code in linux.
 - ▶ What technique did you implement.
- Experiments
 - ▶ Compare the winning rate by using different time control strategy.
- Discussion
 - ▶ Compare effectiveness of using different heuristics.
 - ▶ Give and explain method(s) introduced in alpha-beta has a hope to be used in MCTS (or vice versa).
 - ▶ Besides the goal of becoming the best game player, give and explain at least 3 different goals that a computer game programming can achieve.
 - ▶ Comment and feedbacks.

At least one page with suitable font size for each problem in discussion.

Submission

- Directory hierarchy:
 - ▶ student id // e.g. r09922026 (lowercase)
 - ★ Makefile // make your code
 - ★ src // a folder contains all your codes
 - ★ report.pdf // your report
- Compress your folder into a zip file and submit to <https://www.csie.ntu.edu.tw/~tcg/2021/final.php>
- Due to server limitation, the file size is restricted to 2 MB.

Grading Policy

Final Project

$$\text{Your Point} = P \times \left(\frac{\text{Your Score}}{\text{Boss Score}} \times W_1 + \frac{r}{R} \times W_2 \right) + \text{Report} + B$$

- P : 30
- Your Score : $\in [0, 26]$
- Boss Score : 22.8625
- W_1 : 0.2
- W_2 : 0.8
- r : # of requirements you implemented
- R : 9
- Report : $\in [0, 20]$
- B : Competition Bonus

Important Dates

Submission Deadline

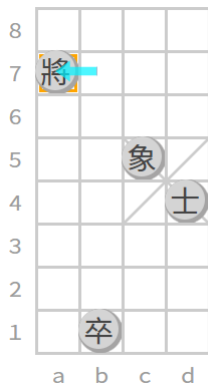
- Code: 2022/01/13 14:20 (UTC+8)
- Report: 2022/01/20 14:20 (UTC+8)

Competition

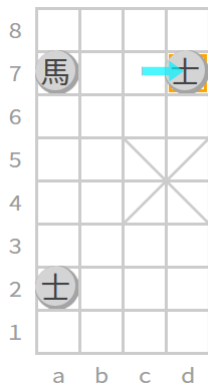
- 2022/01/13 (Thursday) 14:20 (UTC+8)
- 2022/01/14 (Friday) 14:20 (UTC+8)
- 2022/01/17 (Monday) 14:20 (UTC+8)

Appendix

Appendix



Baseline vs Boss
1.15 (W)



Boss vs Baseline
1.1125 (W)



Baseline vs Boss
1.3 (W)

Appendix



Boss vs Baseline
1.225 (W)



Baseline vs Boss
1.3 (W)



Boss vs Baseline
1.2625 (W)

Appendix



Baseline vs Boss
1.3 (W)



Boss vs Baseline
1.2625 (W)



Baseline vs Boss
1.3 (W)

Appendix



Boss vs Baseline
1.3 (W)

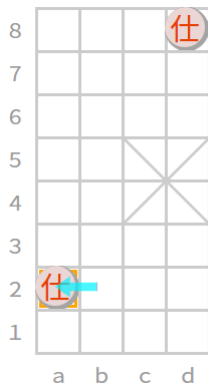


Baseline vs Boss
1.3 (W)



Boss vs Baseline
1.3 (W)

Appendix



Baseline vs **Boss**
1.075 (W)



Boss vs **Baseline**
1.15 (W)

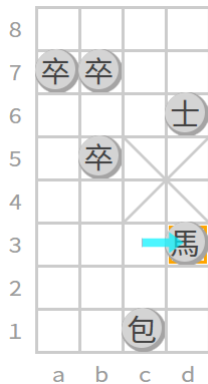


Baseline vs Boss
1.225 (W)

Appendix



Boss vs Baseline
1.225 (W)



Baseline vs Boss
1.225 (W)



Boss vs Baseline
0.35 (D)

Appendix



Baseline vs **Boss**
0.2 (D)



Boss vs Baseline
1.3 (W)