Final Project Description

Due date: 09:00 (UTC+8), Janunray 18, 2018

Final Project Description

- Implement the Chinese Dark Chess using NegaScout.
- For rule of Chinese Dark Chess, see `http://www.iis.sinica.edu.tw/~tshsu/tcg/2017/hwks/rules.pdf`
- Check your school e-mail address for password of template code (Provide by former TA, used in class only) and CDC_interface.

Final Project Requirements

- Implement NegaScout.
- Implement Transposition Table(Hash table).
- Other improved techniques.
- Provide your Makefile or specfied how to compile your codes (Step-by-Step) in the report.
- Your program must has the ability to complete any legal games normally via the given CDC_interface.

Solution Package

- Submit page: `http://w.csie.org/~tcg/2017/`
- Package structure:
  - Your ID [R05xxxxxx/B02xxxxxx/...]
    - code // A folder contains all your codes and makefile
    - report.pdf // Your report
- Compress your folder into a "zip" file
- Due to server limitation, the file size is restricted to 2M bytes

Report content

The report must contain

1. How to compile your program (makefile recommand)
2. What you have implemented.
   - please state the line number of the implementation

The report can contain but not limit to

1. The heuristic you use.
2. The motivation of using these heuristics.
3. The experiment data which shows that your techniques improve your program.

Grading policy

- Final score = Coding score + Document score + Bonus
  - Coding score + Document score is at most 40.
  - Bonus score depends on the tournament performance.
  - Bonus does not include in the 40% final project score
- In the following situation, your score will be very low:
  1. You claim something you have done but you didn't.
  2. Your program can not be compiled.
  3. Your program can not be executed.
  4. Your program can not complete a game normally.
  5. You didn't do anything and just upload the template code.

Brief Rules for Final Project: Swiss-system tournament

1. 抽籤決定第一輪對手。(請提早到場抽籤)
2. while(1){
3. 跟你的對手先後手各下一盤
4. if(裁判覺得可以)break;
5. 根據目前勝負決定下一輪對手。
6. } //大約 $\lceil \log_2$ 參賽人數 $\rceil$ 輪，預估六至八輪

算分方法：

- 勝一局加 2 大分，和一局加 1 大分，負一局加 0 大分。
- 小分 $=$ 對戰過的對手之大分總和。
- 先比大分再比小分，若都平手再比兩程式之間的勝負，若再平手就同名次。

Brief Rules for Final Project: Swiss-system tournament

ID\Round	1		2		3		對手分	Rank
1	1 v.s. 2 2W	4	1 v.s. 7 1W1L	6	1 v.s. 4 2D	8	22	2
2	2 v.s. 1 2L	0	2 v.s. 8 2D	2	2 v.s. 6 1W1D	5	14	5
3	3 v.s. 4 1W1L	2	3 v.s. 5 2D	4	3 v.s. 7 2L	4	25	6
4	4 v.s. 3 1W1L	2	4 v.s. 6 1W1D	5	4 v.s. 1 2D	7	16	4
5	5 v.s. 6 2D	2	5 v.s. 3 2D	4	5 v.s. 8 2W	8	10	3
6	6 v.s. 5 2D	2	6 v.s. 4 1D1L	3	6 v.s. 2 1D1L	4	20	7
7	7 v.s. 8 2W	4	7 v.s. 1 1W1L	6	7 v.s. 3 2W	10	14	1
8	8 v.s. 7 2L	0	8 v.s. 2 2D	2	8 v.s. 5 2L	2	23	8

Brief Rules for Final Project

- 一個程式一盤棋可用計算時間 900 秒，超時立刻算負。
- 三循環算和。
- 雙方連續 30 步皆無吃無翻算和。
- 單局犯規 (輸出不合法步、程式崩潰等等) 發生第二次立刻算輸。
- 裁判擁有所有棋局最終判決權以及犯規時棋局繼續的方式之決定權。

GUI usage

- See “暗棋對弈平台 _ 使用手冊 _windows.pdf” or “User manual of Chinese dark chess client_linux.pdf”
- 思考模式
  - 讀檔模式
    - 讀 Board.txt，寫 Move.txt。
    - 每下一步程式就會關閉。
  - 背景模式
    - 透過 socket 讀寫。
    - 每下一步程式不會結束。
    - 每下完一場介面還是會砍掉你的程式。
- 目前不支援本地自對戰，必須連 server。
- If you want to compare the search engine for two different version, you need to create another folder contains the GUI interface. Use one GUI interface to create the game room, and use the other to join the game room.
- 帳號/密碼：大寫學號

Template Code

- final project template
  - main.cc, anqi.cc, anqi.h
  - Protocol.h, Protocol.cpp, ClientSocket.h, ClientSocket.cpp
- You only need to modify codes in main.cc, anqi.cc and anqi.h
- Need -static -s when compiling in all system
- Need -lwsock32 when compiling in windows system
- Can use -D WINDOWS to enable extra WINDOWS only feature.
- 它的檔案儲存格式是 big5，若需要請自行轉。

Steps of Protocol: 讀檔模式

1. Initial the game state according to board.txt
2. Generate a move
3. Write the move to move.txt

```
1  BOARD B;
2  if (argc<=1) {
3      TimeOut=(B.LoadGame("board.txt")-3)*1000;
4      if(!B.ChkLose())Output(Play(B));
5      return 0;
6  }
```

Steps of Protocol: 背景模式

1. Initial the socket connection
2. Initial the game state through the socket
3. While(true)
   a. If it's your turn
      1. Generate a move and send it through the socket
      2. Update the local game state
      3. Receive the message from the server
      4. Change the turn
   b. If it's opponent's turn
      1. Wait for opponents' move
      2. Receiving Opponent's move and update the game state.
      3. Change the turn

Board Index

Template Code

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31

Protocal

28	29	30	31
24	25	26	27
20	21	22	23
16	17	18	19
12	13	14	15
8	9	10	11
4	5	6	7
0	1	2	3

Protocal (x,y)

(0,7)	(1,7)	(2,7)	(3,7)
(0,6)	(1,6)	(2,6)	(3,6)
(0,5)	(1,5)	(2,5)	(3,5)
(0,4)	(1,4)	(2,4)	(3,4)
(0,3)	(1,3)	(2,3)	(3,3)
(0,2)	(1,2)	(2,2)	(3,2)
(0,1)	(1,1)	(2,1)	(3,1)
(0,0)	(1,0)	(2,0)	(3,0)

Frequently Asked Questions

- For MAC:
  - Q: Is there a GUI version for MAC?
    A: Currently there is no MAC version.
- For Linux:
  - Q: What does permission denied means?
    A: Your "search" file needs to be executable.
  - Q: Why my GUI interface does not work?
    A: Make sure to add the LD_LIBRARY_PATH=. to include the GameDLL.so
- Q: Any hardware limit?
  A: In the tournament, you need to prepare your own hardware. You can use 16-core workstation or GPU or TPU or so on. But you need to support CPU only mode for testing purpose.