TCG Homework2 2016

Cheng Yueh

National Taiwan University

fenzhangs@gmail.com

November 23, 2016

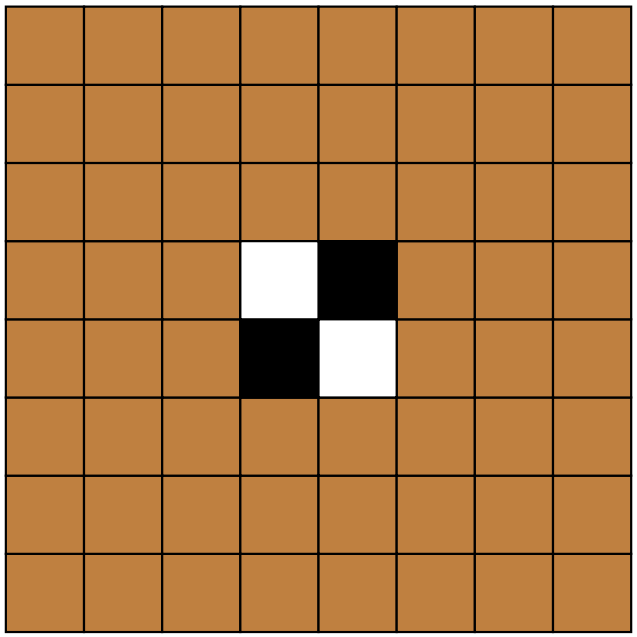
Description

Write a program that plays 8×8 Othello with Monte-Carlo game tree search techniques. The purpose of this homework is to ask you to master the basic techniques of Monte-Carlo tree search, not to test your efforts in designing GUI.

- **Due date: 23:59pm, December 15, 2016.**

Rules of Othello : Game Play

1. The board will start with 2 black discs and 2 white discs at the center of the board. They are arranged with black forming a North-East to South-West direction. White is forming a North-West to South-East direction.

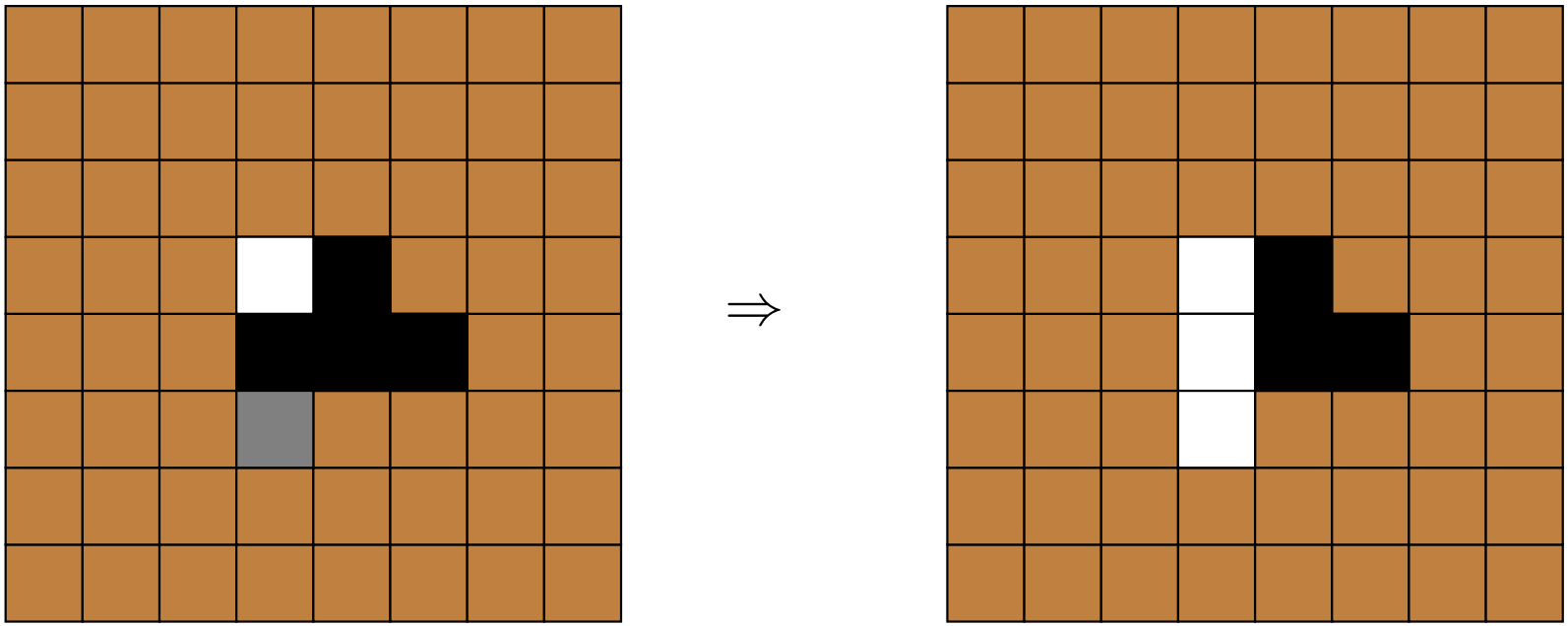


Rules of Othello : Game Play

2. The game alternates between white and black. Black always starts the game.
3. When a player has no valid moves, he passes his turn and the opponent continues. A player can not voluntarily forfeit his turn.
4. When 2 consecutive passes are made the game ends.
5. Discs are counted and the player with the majority of his or her color discs on the board is the winner.

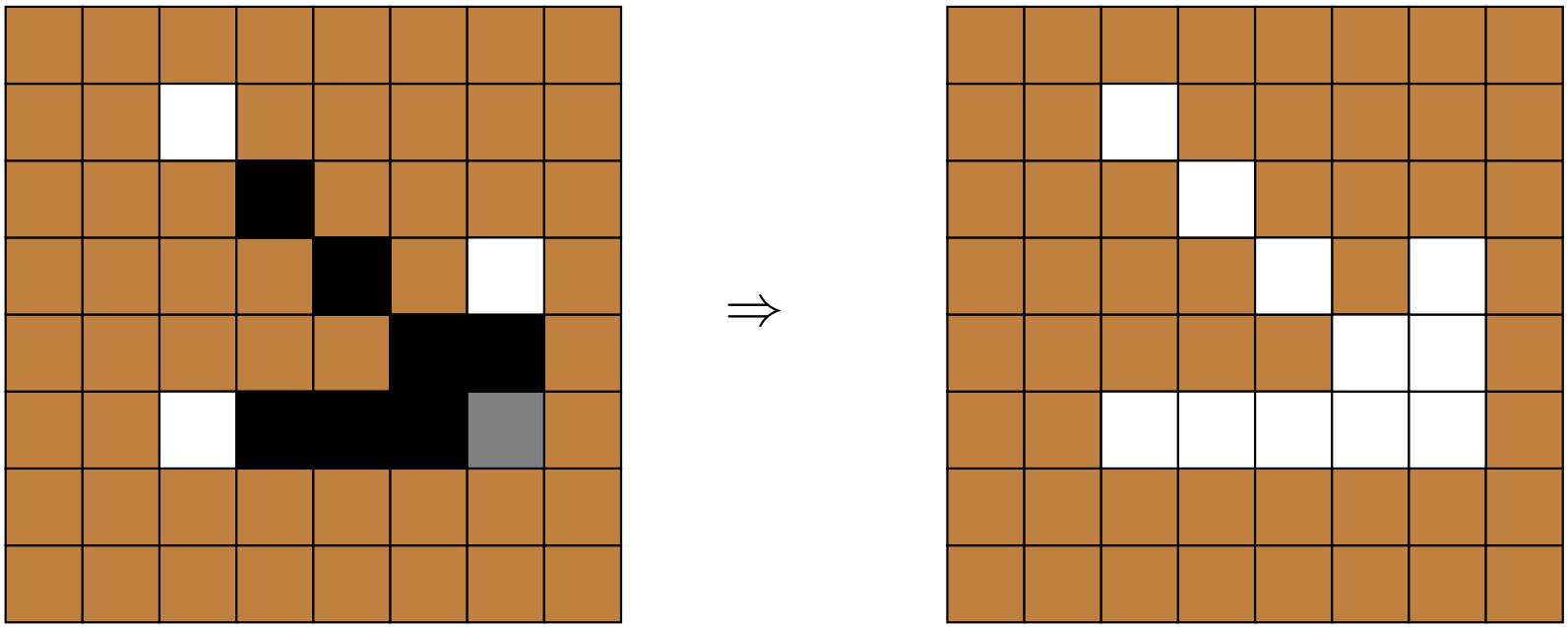
Rules of Othello : Legal Moves

1. To make a valid move, you must outflank and flip at least one opposing disc. To outflank means to place a disc on the board so that your opponent's row(s) of disc(s) is bordered at each end by a disc of your color. (A row is defined as one or more discs in a continuous straight line).



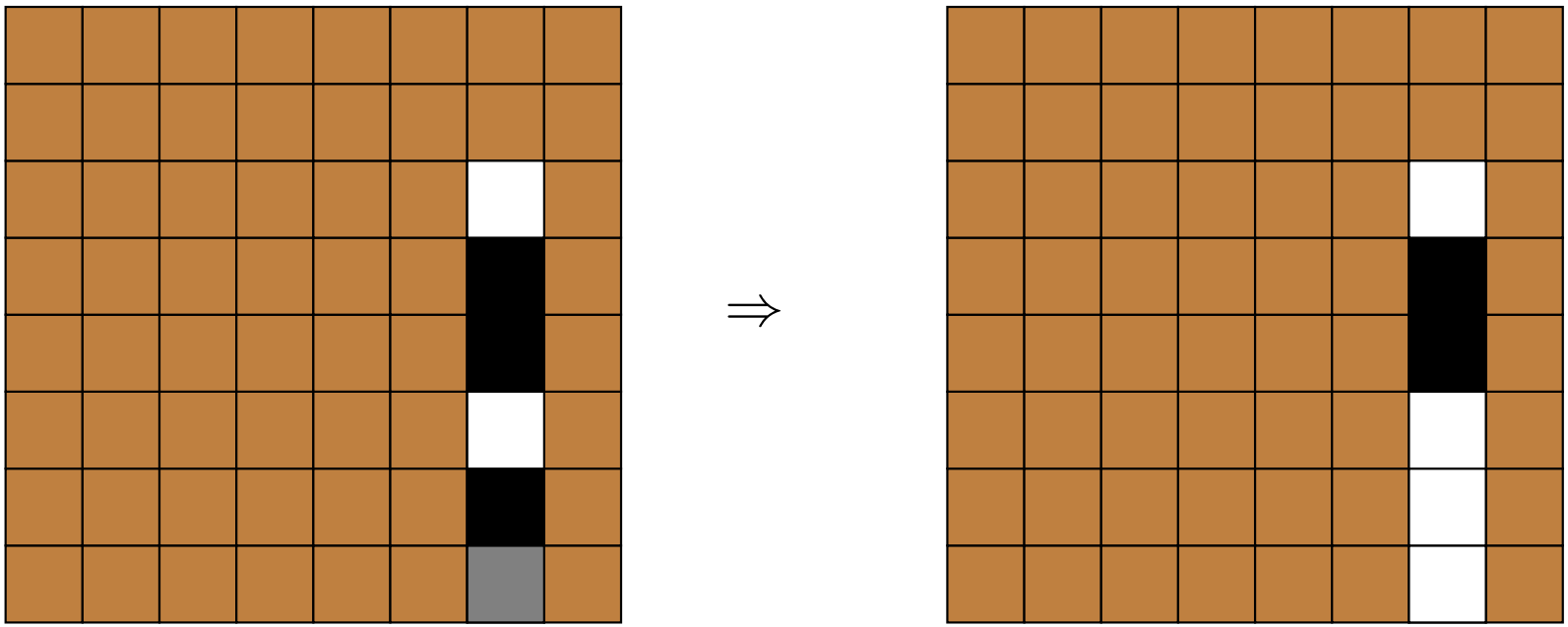
Rules of Othello : Legal Moves

2. A disc may outflank any number of discs in one or more rows in any number of directions at the same time - horizontally, vertically or diagonally.



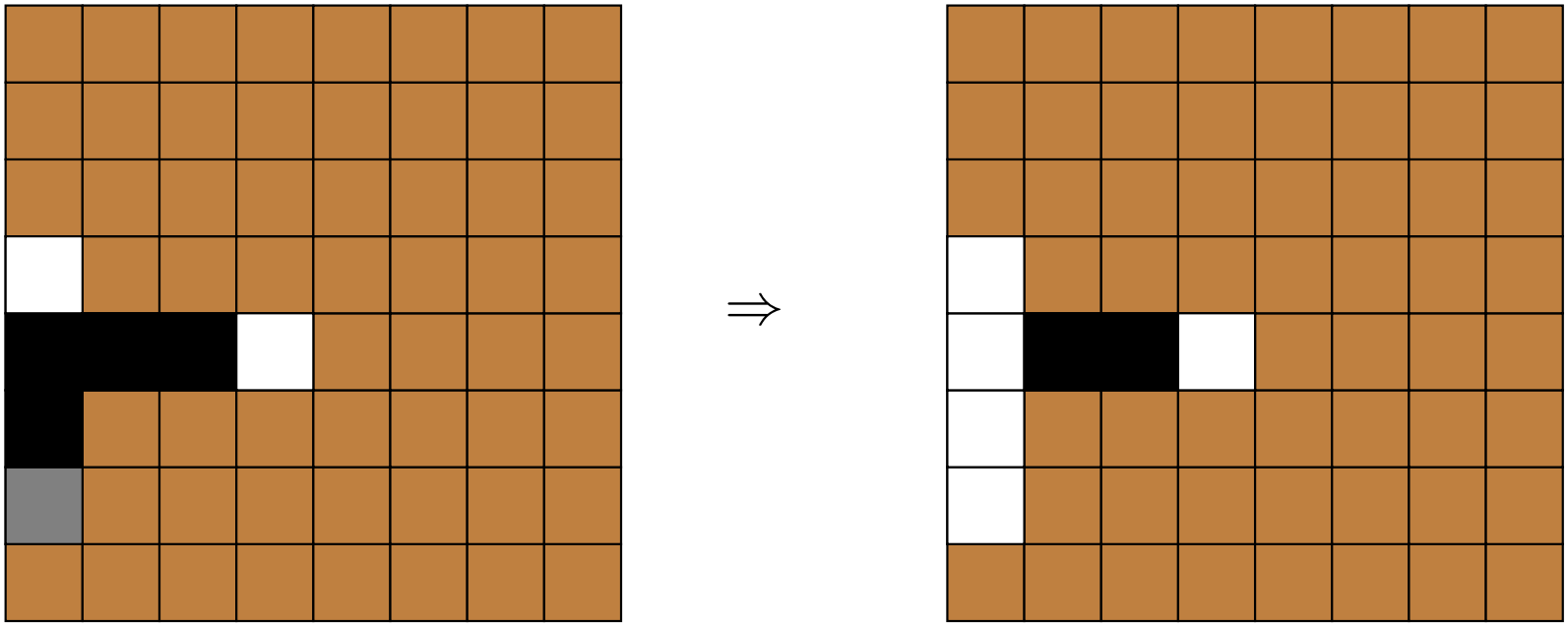
Rules of Othello : Legal Moves

3. You cannot skip over your own color disc to outflank an opposing disc.



Rules of Othello : Legal Moves

4. Discs may only be outflanked as a direct result of a move and must fall in the direct line of the disc placed down.
5. All discs outflanked in any one move must be flipped, even if it is to the player's advantage not to flip them at all.



Othello Text Protocol(OTP)

- **name**

arguments	none
effects	none
output	name str `str` - Name of the engine (at most 514 bytes)
comments	If the judge does not request name, then some people might make two connections to the judge and fake the result.

Othello Text Protocol(OTP)

- **quit**

arguments	none
effects	The session is terminated and the connection is closed.
output	quit
comments	The full response of this command must be sent before the engine closes the connection. The controller must receive the response before the connection is closed on its side.

Othello Text Protocol(OTP)

- **clear_board**

arguments	none
effects	The board is cleared and the move history is reset to empty.
output	clear_board
comments	

Othello Text Protocol(OTP)

- **play**

arguments	x y x - x value of the move y - y value of the move (x,y)= $(8, 0)$ represents pass
effects	A stone of the current player is played at the requested vertex. The flipped stones is updated if needed and the move is added to the move history.
output	play
comments	Pass while the current player has move are not considered illegal from the protocol point of view.

Othello Text Protocol(OTP)

(x,y)

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

(8,0) = PASS

Othello Text Protocol(OTP)

- **genmove**

arguments	none
effects	A move of the the current player is played where the engine chooses. The flipped stones is updated if needed and the move is added to the move history.
output	genmove x y x - x value of the move y - y value of the move (x,y)$= (8, 0)$ represents pass
comments	

Othello Text Protocol(OTP)

- **undo**

arguments	none
effects	The board configuration and the flipped stones are reset to the state before the last move. The last move is removed from the move history.
output	undo
comments	If you want to take back multiple moves, use this command multiple times. Use clear_board if you want to start over.

Othello Text Protocol(OTP)

- **final_score**

arguments	none
effects	none
output	final_score score `score` - (# of black) − (# of white)
comments	

- **showboard**

arguments	none
effects	none
output	showboard
comments	The engine may draw the board as it likes. This command is only intended to help humans with debugging.

computer vs computer

1. Compile search.cpp and judge.cpp. The code contains c++11 features so make sure your compiler supports at least c++11. And for windows, it needs to link ws2_32.lib.
2. Excute "judge port [# of round]". The default # of round is 2.
3. Excute "search1 ip port" which search1 is the first player program.
4. Excute "search2 ip port" which search2 is the second player program.

Examples are in cpu_vs_cpu.bat for windows. Examples are in cpu_vs_cpu.sh for linux.

human vs computer 1

1. Compile search.cpp
2. Execute “search”.
3. Command using OTP

human vs computer 2

1. Compile simple_http_UI.cpp
2. Execute “simple_http_UI 7122”.
3. Open Chrome.
4. Type 127.0.0.1:7122 in address bar

Rules of Othello : Additional Rules of Computer Othello

1. Your program needs to connect to judge within 10 seconds after it starts.
2. You have 10 seconds to respond to each genmove command, and 1 seconds to respond to each others. If one program does not respond in time, it will lose immediately.
3. If one program play an illegal move, it will lose immediately.
4. **single thread**, memory $\leq$ 4GiB, disk usage $\leq$ 8GiB

Requirement : Solution Package

1. Source code with implementation of required algorithm.
   - UCB
   - UCT
   - Progressive pruning
   - If your implementation of above algorithm is correct, then you will get full points. The rest are bonus.
2. report.**pdf** (the documentation)
   - Explain how to compile your program. List of available compiler: g++, clang++.
   - Explain the techniques you implemented.
   - Show the coefficients you picked and explain how they are picked.
   - Specify maximum memory usage, and disk usage of your program for scheduling the tournament.

Requirement : Program

1. You are allowed to throw the template code away and write it your own way, but **the program must be able to play with the given judge and search**.
2. Name your program your student id and respond to **name** command with your student id.
3. Your program must be compiled and run **under linux**.
4. Bonus points: win rate againt all other students

Where do I do?

1. In OTP.h, there is a do_genmove function. You need to replace the current algorithm (choose move randomly) with Monte-Carlo tree search.
2. See board.h for some function like get_valid_move, is_valid_move.

The End