[AI] Project Documentation
Sudarshana Jagadeeshi

Several Examples of Running:
Example 1
input position:
['x','x','W','W','x','x','W','W','x','x','B','W','x','B','x','B','W','B','B','W','B']
output position:
['x', 'W', 'W', 'W', 'x', 'x', 'W', 'W', 'x', 'x', 'B', 'x', 'x', 'B', 'x', 'B', 'W', 'B', 'B', 'W', 'B']
evaluation after minimaxMidgameEndgameAB depth 3:
996 AFTER CALLING MINIMAX 91 TIMES

Example 2
input position:
['x','B','B','x','x','x','x','x','W','x','W','W','x','x','x','x','B','x','x','B','x']
output position:
['x', 'B', 'B', 'x', 'x', 'x', 'x', 'x', 'W', 'x', 'W', 'W', 'x', 'x', 'x', 'x', 'B', 'x', 'x', 'B', 'x']
evaluation after minimaxMidgameEndgameAB depth 2:
-3021 after calling minimax 257 times

Example 3
input position:
['W','x','x','x','x','x','B','B','B','x','B','x','B','x','W','W','x','x','W','W','x']
output position:
['W', 'x', 'x', 'x', 'x', 'x', 'B', 'B', 'B', 'x', 'x', 'x', 'B', 'x', 'W', 'W', 'x', 'x', 'W', 'W', 'W']

evaluation after minimaxOpeningAB depth 2:
1 after calling minimax 44 times

Examples where alphabeta produced savings over minimax:
Example 1 (depth 3 midgameendgame):
input position:
['W', 'B', 'W', 'x', 'x', 'x', 'W', 'W', 'B', 'x', 'B', 'B', 'x', 'x', 'W', 'x', 'x', 'x', 'B', 'B', 'W']
output position:
['W', 'B', 'W', 'x', 'W', 'x', 'W', 'W', 'B', 'x', 'x', 'B', 'x', 'x', 'W', 'x', 'x', 'x', 'B', 'B', 'W']
ALPHABETA: 102 TIMES
REGULAR MINIMAX: 313 times

Example 2 (depth 2 opening)
input position:
['x', 'x', 'x', 'x', 'x', 'W', 'B', 'x', 'x', 'B', 'B', 'W', 'W', 'x', 'x', 'x', 'x', 'x', 'B', 'W', 'B']
output position:
['W', 'x', 'x', 'x', 'x', 'W', 'B', 'x', 'x', 'B', 'B', 'W', 'W', 'x', 'x', 'x', 'x', 'x', 'B', 'W', 'B']
ALPHABETA: 35 times
REGULAR MINIMAX: 189 times

My new evaluation function for the opening is the following:

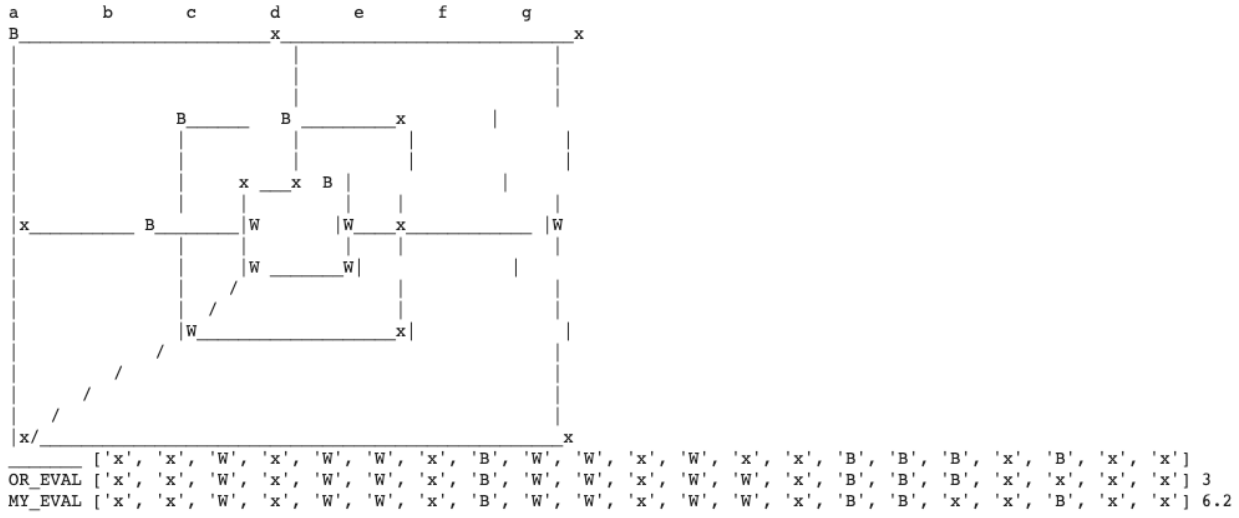
diffpiece + (morrisways) + 1.2*(diffmorris) + 0.25*(whitesturn)

My new evaluation function for the middlegame has only one difference- opponentmoves:
diffpiece + (morrisways) + 1.2*(diffmorris) + 0.25*(whitesturn)- 0.1*(opponentmoves)

> morrisways- difference in # of ways to make a morris on the next turn between you and opponent
> diffpiece- difference in piece count
> whitesturn- a penalty term that is -1 when it is not whites turn, and 0 otherwise.
> diffmorris- difference in the number of morrises
> opponentmoves- number of possible opponent moves

The original function did not place enough emphasis on the morris/mill. I believe a morris is worth the equivalent of a piece, even perhaps slightly more.

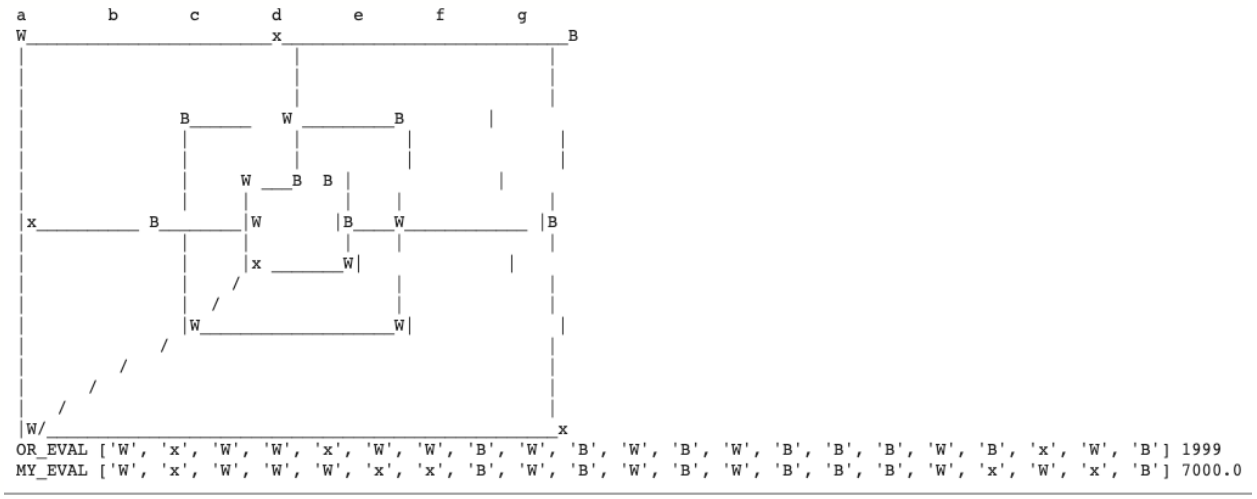
Example position 1 (Opening):
['x','x','W','x','W','W','x','B','W','W','x','W','x','x','B','B','B','x','B','x','x'] with White to move



Note: or_eval- original_eval

White is in a good spot once he completes the mill. However, the two algorithms differ in which piece to remove afterwards. My algorithm correctly recognizes that removing the center piece (rather than the top-left one) will limit blacks potential mills.

Example position 2 (Middlegame):
['W','x','W','W','x','W','x','B','W','B','W','B','W','B','B','B','W','B','W','x','B'] with white to move



The original evaluation misses the fact that a mill can be created, instead opting to move the top left piece. I’m not sure why it misses such a simple 1-mover, but it is clear that my move is better.