

John Nickle

September 24, 2015

In this project I created Alpha-Beta and an AdvancedSearch that used Alpha-Beta along with a singular extension and a quiescent search extension. In addition to these searches, I also improved upon the BasicMaterials by adding zones of interest, adding pawn promotion incentive, and giving knights higher scores for being in more opportune positions. Overall, the AI performed very well. It was difficult to beat until late game because I had no end-game heuristics that were able to help the AI win.

DATA:

When set to depth 6, Advanced Search does not finish searching in 10 seconds because of the singular extension and quiescent search additions cause it to go further than 6, often much too far to be beneficial in the opening game.

Game status

Computer move

Boards generated: 802756  
Time (ms): 42384  
Boards/ms: 18.940096

Restart game

Replace game

a b c d e f g h

8



8

7

6

5

4

3

2

1

a b c d e f g h

b4

Black searcher

AdvancedSearch

White searcher

AdvancedSearch

Black evaluator

AdvancedMaterial

White evaluator

AdvancedMaterial

Black depth

6

White depth

1

It works exceptionally well at a depth of 4.

Game status

Computer move

Boards generated: 0  
Time (ms): 0  
Boards/ms: NaN

Restart game

Replace game

Na3 c6  
Nb5 cxb5  
Rb1 Na6  
Ra1 Rb8  
Rb1 Ra8  
Ra1 Rb8  
b4 Nxb4  
Rb1 Nxa2  
Rxb5 Nxc1  
Qxc1 Ra8  
Rxb7 Bxb7  
Qa3 Bxg2  
Bxg2 Rb8  
Qxa7 Rc8  
Qxd7+ Qxd7  
Ba8 Rxa8  
Kd1 Qxd2+  
Kxd2 Rb8  
Kc3 Ra8  
Kb4 Rb8+  
Ka5 Ra8+  
Kb6 Rb8+  
Ka7 Ra8+  
Kxa8 Kd8  
Kb8 Ke8  
Ka8 Kd8  
Kb8 Ke8  
Ka8 Kd8  
Kb8 Ke8

Black searcher

AdvancedSearch

White searcher

AdvancedSearch

Black evaluator

AdvancedMaterial

White evaluator

AdvancedMaterial

Black depth

4

White depth

1

The AdvancedSearch takes a little bit longer than Alpha-Beta because of the two extensions, but makes much more informed decisions. After the openings, AdvancedSearch outperforms both MiniMax and Alpha-Beta significantly in play, and greatly outperforms MiniMax in time efficiency.




Game status

Computer move

Boards generated: 3866  
Time (ms): 250  
Boards/ms: 15.464

Restart game

Replace game

	a	b	c	d	e	f	g	h	
8									8
7									7
6									6
5									5
4									4
3									3
2									2
1									1
	a	b	c	d	e	f	g	h	

Black searcher

Black evaluator

Black depth

AdvancedSearch

AdvancedMaterial

4

White searcher

White evaluator

White depth

AlphaBeta

AdvancedMaterial

4

Alpha Beta first Move Against Advanced

Game status

Computer move

Boards generated: 9848  
Time (ms): 502  
Boards/ms: 19.61753

Restart game

Replace game

Nc3 c6

Black searcher

AdvancedSearch

Black evaluator

AdvancedMaterial

Black depth

4

White searcher

AlphaBeta

White evaluator

AdvancedMaterial

White depth

4

## Advanced Search against Alpha Beta

As you can see, even though AdvancedSearch uses Alpha-Beta as its core searcher, it doesn't even use the same opening move as Alpha-Beta by itself.

Game status

Computer move

Boards generated: 3866  
Time (ms): 602  
Boards/ms: 6.421927

Restart game

Replace game

Nc3

a b c d e f g h

8

8

7

7

6

6

5

5

4

4

3

3

2

2

1

1

a b c d e f g h

Black searcher

Minimax

Black evaluator

AdvancedMaterial

Black depth

4

White searcher

AlphaBeta

White evaluator

AdvancedMaterial

White depth

4

Alpha Beta Opening Move White. Note the time of 602 ms at depth 4.

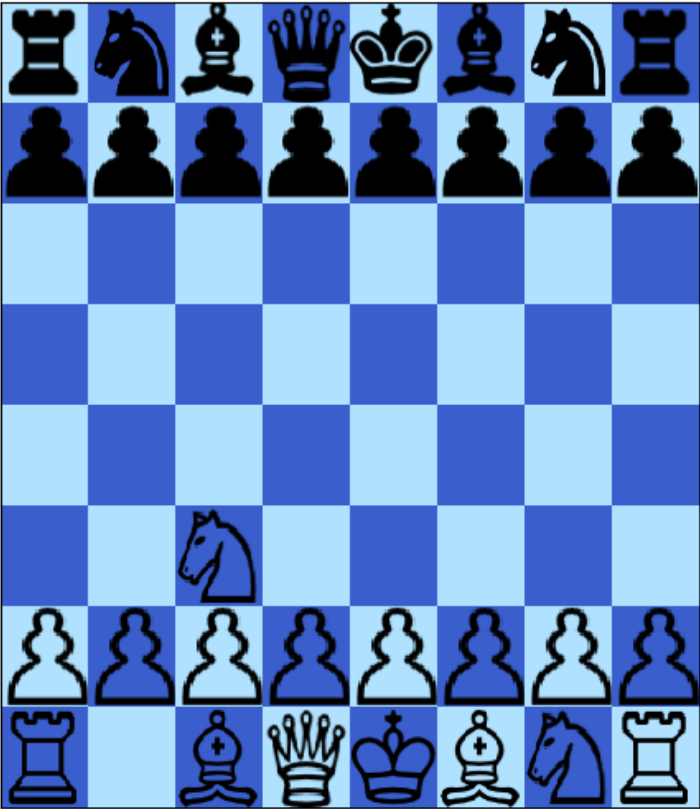
Game status

Computer move

Boards generated: 244780  
Time (ms): 14129  
Boards/ms: 17.324652

a b c d e f g h

8  
7  
6  
5  
4  
3  
2  
1



8  
7  
6  
5  
4  
3  
2  
1

a b c d e f g h

Restart game

Replace game

Nc3

Black searcher: Minimax

White searcher: AlphaBeta

Black evaluator: AdvancedMaterial

White evaluator: AdvancedMaterial

Black depth: 4

White depth: 4

MiniMax First Move Time and Node. Note that the move has still not been made after 14129 ms at a depth of 4. Minimax was greatly outperformed by both AlphaBeta and AdvancedSearch.

AdvancedSearch was able to make decisions based on a greater depth than the other searches, and was able to do so in a timely manner. This led to an incredible foresight for the AI when playing, making it very difficult to trick into making bad moves. However, some odd moves were seen while playing, often made out of nowhere for seemingly no good reason. This could be due to a bug in my code, or maybe the AI thought it had a great set of moves planned way ahead and it just didn't play out. Either way, the AI performed exceptionally well and I am very happy with it.