CS 6364 Project 2

Project 2 Report

By:

DHRUV MITTAL

DXM220015

How to run any file:

The codebase is in python

Open the folder in an IDE, and cd into the folder. In our case DXM220015 is folder name. Open the terminal in the IDE and do the following options…

Cmd to write in terminal (when you navigate into the suitable folder) :-

* python “file\_path” input\_board.txt output\_board.txt depth

for example:-



PART 1:-

1. MiniMaxOpening

Navigate to the folder Part1 using cmd: cd Part1

Then write the following cmd:

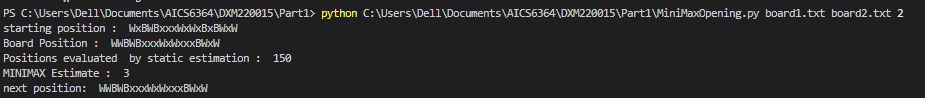
python C:\Users\Dell\Documents\AICS6364\DXM220015\Part1\miniMaxOpening.py board1.txt board2.txt 2

**Input (board1.txt)** – WxBWBxxxWxWxBxBWxW

**Output (board2.txt)** - Board Position : WWBWBxxxWxWxxxBWxW

Positions evaluated by static estimation : 150

MINIMAX Estimate : 3

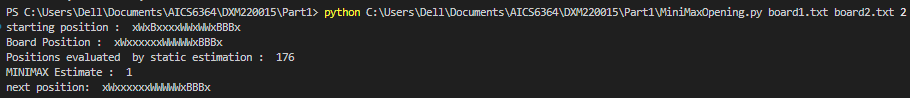


**Input (board1.txt)** – xWxBxxxxWWxWWxBBBx

**Output (board2.txt)** - Board Position : xWxxxxxxWWWWWxBBBx

Positions evaluated by static estimation : 176

MINIMAX Estimate : 1



1. MiniMaxGame

Navigate to the folder Part1 using cmd: cd Part1

Then write the following cmd:

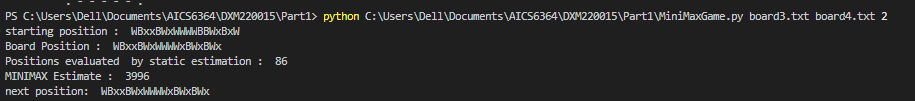
python C:\Users\Dell\Documents\AICS6364\DXM220015\Part1\miniMaxGame.py board3.txt board4.txt 2

**Input (board3.txt)** – WBxxBWxWWWWBBWxBxW

**Output (board4.txt)** - Board Position : WBxxBWxWWWWxBWxBWx

Positions evaluated by static estimation : 86

MINIMAX Estimate : 3996

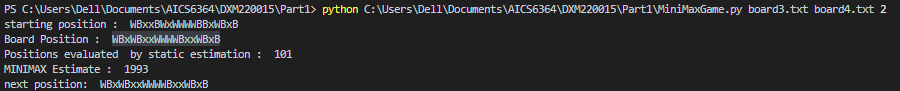


**Input (board3.txt)** – WBxxBWxWWWWBBxWBxB

**Output (board4.txt)** - Board Position : WBxWBxxWWWWBxxWBxB

Positions evaluated by static estimation : 101

MINIMAX Estimate : 1993



PART 2:-

1. ABOpening

Navigate to the folder Part2 using cmd: cd Part2

Then write the following cmd:

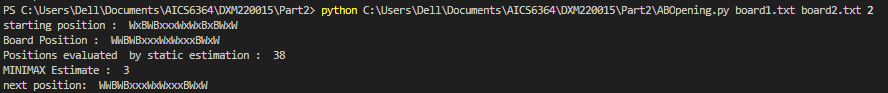
python C:\Users\Dell\Documents\AICS6364\DXM220015\Part2\ABOpening.py board1.txt board2.txt 2

**Input(board1.txt)** – WxBWBxxxWxWxBxBWxW

**Output (board2.txt)** - Board Position : WWBWBxxxWxWxxxBWxW

Positions evaluated by static estimation : 38

MINIMAX Estimate : 3

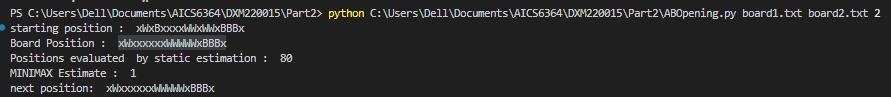


**Input(board1.txt)** – xWxBxxxxWWxWWxBBBx

**Output (board2.txt)** - Board Position : xWxxxxxxWWWWWxBBBx

Positions evaluated by static estimation : 80

MINIMAX Estimate : 1



* Results of MiniMaxOpening in Part1 (A) and ABOpening in Part2 (A) are the same, and we see number of positions evaluated in ABOpening is less because we use pruning method.

1. ABGame

Navigate to the folder Part2 using cmd: cd Part2

Then write the following cmd:

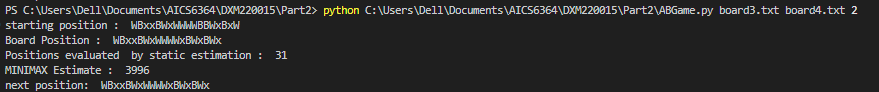
python C:\Users\Dell\Documents\AICS6364\DXM220015\Part2\ABGame.py board3.txt board4.txt 2

**Input (board3.txt)** – WBxxBWxWWWWBBWxBxW

**Output (board4.txt)** - Board Position : WBxxBWxWWWWxBWxBWx

Positions evaluated by static estimation : 31

MINIMAX Estimate : 3996

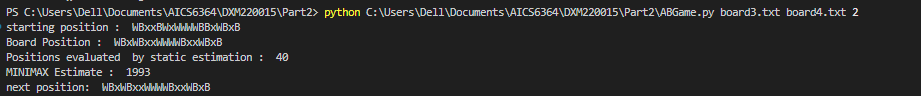


**Input (board3.txt)** – WBxxBWxWWWWBBxWBxB

**Output (board4.txt)** - Board Position : WBxWBxxWWWWBxxWBxB

Positions evaluated by static estimation : 40

MINIMAX Estimate : 1993



* Result of MiniMaxGame in Part1 (B) and ABGame in Part2 (B) are the same, and we see number of positions evaluated in ABGame is less because we use pruning method.

PART 3:-

1. MiniMaxOpeningBlack

Navigate to the folder Part3 using cmd: cd Part3

Then write the following cmd:

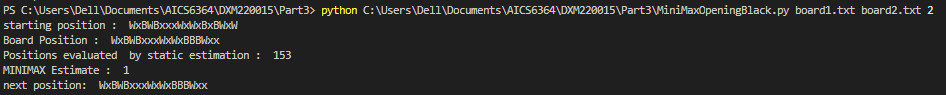
python C:\Users\Dell\Documents\AICS6364\DXM220015\Part3\MiniMaxOpeningBlack.py board1.txt board2.txt 2

**Input (board1.txt)** – WxBWBxxxWxWxBxBWxW

**Output (board2.txt)** - Board Position : WxBWBxxxWxWxBBBWxx

Positions evaluated by static estimation : 153

MINIMAX Estimate : 1

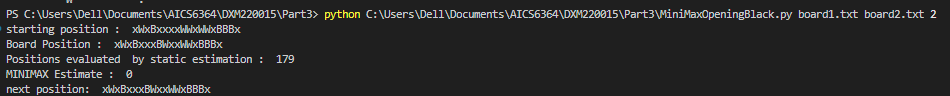


**Input (board1.txt)** – xWxBxxxxWWxWWxBBBx

**Output (board2.txt)** - Board Position : xWxBxxxBWxxWWxBBBx

Positions evaluated by static estimation : 179

MINIMAX Estimate : 0



1. MiniMaxGameBlack

Navigate to the folder Part3 using cmd: cd Part3

Then write the following cmd:

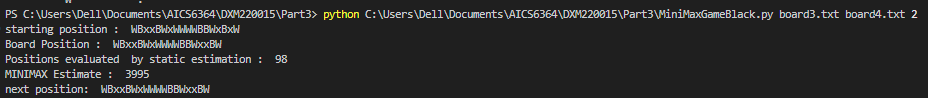
python C:\Users\Dell\Documents\AICS6364\DXM220015\Part3\MiniMaxGameBlack.py board3.txt board4.txt 2

**Input (board3.txt)** – WBxxBWxWWWWBBWxBxW

**Output (board4.txt)** - Board Position : WBxxBWxWWWWBBWxxBW

Positions evaluated by static estimation : 98

MINIMAX Estimate : 3995

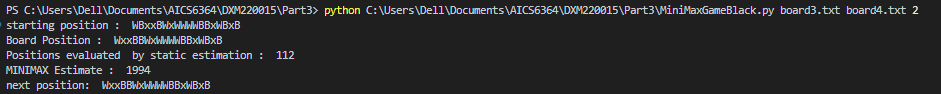


**Input (board3.txt)** – WBxxBWxWWWWBBxWBxB

**Output (board4.txt)** - Board Position : WxxBBWxWWWWBBxWBxB

Positions evaluated by static estimation : 112

MINIMAX Estimate : 1994



PART 4:-

1. MiniMaxOpeningImproved

Navigate to the folder Part4 using cmd: cd Part4

Then write the following cmd:

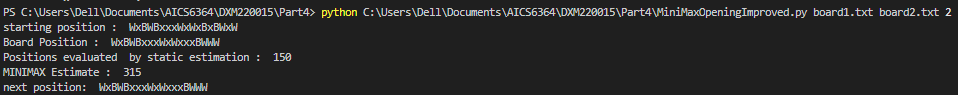
python C:\Users\Dell\Documents\AICS6364\DXM220015\Part4\miniMaxOpeningImproved.py board1.txt board2.txt 2

**Input (board1.txt)** – WxBWBxxxWxWxBxBWxW

**Output (board2.txt)** - Board Position : WxBWBxxxWxWxxxBWWW

Positions evaluated by static estimation : 150

MINIMAX Estimate : 315

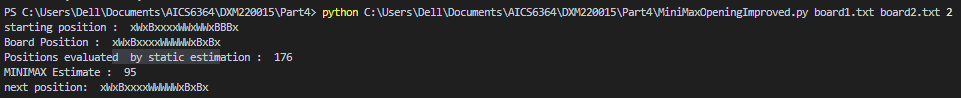


**Input (board1.txt)** – xWxBxxxxWWxWWxBBBx

**Output (board2.txt)** - Board Position : xWxBxxxxWWWWWxBxBx

Positions evaluated by static estimation : 176

MINIMAX Estimate : 95



* In MiniMaxOpening Part1(A) and MiniMaxOpeningImproved Part4(A), we see that we have different results. The result in Part4(A) is different than the result in Part1(A) even though the input is same, it is because of the improved static estimation function in Part4(A) for opening game.

Here is the improved static estimation function:-

return 100\*(numWhitePieces-numBlackPieces) + 5\*(numberOfMills(boardPos,'W'))-10\*(numberOfMills(boardPos,'B'))

 - this gives me number of mills that can be formed using White pieces.

 - this gives me the number of mills that can be formed using black pieces.

My improved static estimation function for the opening move is better than given in the document as reward is given when mill is made for white player and points are deducted when black player makes a mill. So this helps to make a move which in turn helps to avoid black player making mills. According to my understanding number of pieces predicts the winner at the end, more the pieces more the chances of you winning.



1. MiniMaxGameImproved

Navigate to the folder Part4 using cmd: cd Part4

Then write the following cmd:

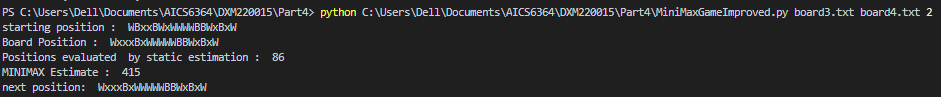
python C:\Users\Dell\Documents\AICS6364\DXM220015\Part4\MiniMaxGameImproved.py board3.txt board4.txt 2

**Input (board3.txt)** – WBxxBWxWWWWBBWxBxW

**Output (board4.txt)** - Board Position : WxxxBxWWWWWBBWxBxW

Positions evaluated by static estimation : 86

MINIMAX Estimate : 415

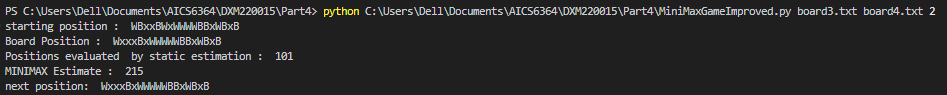


**Input (board3.txt)** – WBxxBWxWWWWBBxWBxB

**Output (board4.txt)** - Board Position : WxxxBxWWWWWBBxWBxB

Positions evaluated by static estimation : 101

MINIMAX Estimate : 215



* In MiniMaxGame Part1(B) and MiniMaxGameImproved Part4(B), we see that we have different results. The result in Part4(B) is different than the result in Part1(B) even though the input is same, it is because of the improved static estimation function in Part4(B) for opening game.

Here is the improved static estimation function:-

return (1000\*(numWhitePieces-numBlackPieces)-len(blackMoves) + len(whiteMoves) + 5\*(numberOfMills(boardPos,'W'))-10\*(numberOfMills(boardPos,'B')) )

 - this gives me possible black player moves.

 - this gives me possible white player moves.

My improved static estimation function for during the mid and end game is better than given in the document as reward is given when mill is made for white player and points are deducted when black player makes a mill. So this helps to make a move which in turn helps to avoid black player making mills. Also I am considering the available number of white moves, which means move in such a way that white player has more moves to make than the black player. According to my understanding number of pieces predicts the winner at the end, more the pieces more the chances of you winning.