# Nine Men Morris Project Report

We go over all the cases one by one.

Command to run the code:
python <file_name.py> <input_board.txt> <output_board_path.txt> <depth>

## 1. Running MiniMaxOpening.py

**Console output 1:** python MiniMaxOpening.py board.txt board1.txt 5

Input position: BBxxxxWBWWWxxxxxxxxxx
Output position: BBxWxxWBWWWxxxxxxxxxx

Positions evaluated by static estimation:  612636
MINIMAX estimate:  3

**Console output 2:** python MiniMaxOpening.py board2.txt board1.txt 5

Input position: BBBxWxWBWWWBBBxWxxxxW
Output position: BBBxWxWBWWWBBBWWxxxxW

Positions evaluated by static estimation:  59912
MINIMAX estimate:  2

## 2. Running ABOpening.py

**Console output 1:** python ABOpening.py board.txt board1.txt 5

Input position: BBxxxxWBWWWxxxxxxxxxx
Output position: BBxWxxWBWWWxxxxxxxxxx

Positions evaluated by static estimation:  7771
MINIMAX estimate:  3

**Console output 2:** python ABOpening.py board2.txt board1.txt 5

Input position: BBBxWxWBWWWBBBxWxxxxW
Output position: BBBxWxWBWWWBBBWWxxxxW

Positions evaluated by static estimation: 1614
MINIMAX estimate: 2

Savings using Alpha-Beta pruning method:

As we can see in the above ABOpening and MiniMaxOpening consoles output

Comparing console output 1:

Positions evaluated = 612636 (MiniMaxOpening)
Positions evaluated = 7771 (ABOpening)

Comparing console output 1:

Positions evaluated = 59912 (MiniMaxOpening)
Positions evaluated = 1614 (ABOpening)

Savings are 90% on an average for Alpha-beta pruning over MiniMax Algorithm

## 3. Running MiniMaxGame.py

**Console output 1:** python MiniMaxGame.py board.txt board1.txt 6

Input position: BBBxxxWBWWWBxxxxxxxxW
Output position: BBBxWxWBxWWBxxxxxxxxW

Positions evaluated by static estimation: 88186
MINIMAX estimate: -7

**Console output 2:** python MiniMaxGame.py board2.txt board1.txt 6

Input position: WBWxBxBWWWxxBxxBWBWxx
Output position: WBWxBxBWWxWxBxxBWBWxx

Positions evaluated by static estimation: 82833
MINIMAX estimate: 994

## 4. Running ABGame.py

### **Console output 1:** python ABGame.py board1.txt board1.txt 6

Input position: BBBxxxWBWWWBxxxxxxxxW
Output position: BBBxWxWBxWWBxxxxxxxxW

Positions evaluated by static estimation:  3765
MINIMAX estimate:  -7

### **Console output 1:** python ABGame.py board2.txt board1.txt 6

Input position: WBWxBxBWWWxxBxxBWBWxx
Output position: WBWxBxBWWxWxBxxBWBWxx

Positions evaluated by static estimation:  3167
MINIMAX estimate:  994

### **Savings using Alpha-Beta pruning method:**

As we can see in the above ABOpening and MiniMaxOpening consoles output
Comparing console output 1:
Positions evaluated = 88186 (MiniMaxOpening)
Positions evaluated = 3765 (ABOpening)

Comparing console output 1:
Positions evaluated = 82833 (MiniMaxOpening)
Positions evaluated = 3167 (ABOpening)

Savings are 90% on an average for Alpha-beta pruning over MiniMax Algorithm and varies as per the board and moves combinations and remaining pieces on the board.

## 5. Running MiniMaxOpeningBlack.py

**Console output 1:** python MiniMaxOpeningBlack.py board2.txt board1.txt 6

Input position: WBWxBxBWxxxxxxxxxxWxx
Output position: WBWxBxBWxxxxxxxBxxWxx

Positions evaluated by static estimation:  514028
MINIMAX estimate:  0

--- As we can see the program is playing as a Black piece and adding a black piece to the board

**Console output 1:** python MiniMaxOpeningBlack.py board.txt board1.txt 5

Input position: WBWxBxBWxxxxxxxxxxWxx
Output position: WBWxBxBWxxxxxxxBxxWxx

Positions evaluated by static estimation:  514028
MINIMAX estimate:  0

**Console output 2:** python MiniMaxOpeningBlack.py board2.txt board1.txt 5

Input position: Wxxxxxxxxxxxxxxxxxxxx
Output position: WBxxxxxxxxxxxxxxxxxxx

Positions evaluated by static estimation:  1917984
MINIMAX estimate:  0

## 6. Running MiniMaxGameBlack.py

**Console output 1:** python MiniMaxGameBlack.py board.txt board1.txt 6

Input position: WBWxBxBWWWxxBxxBWBWxx
Output position: WxWxBxBWWWxBBxxBWBWxx

Positions evaluated by static estimation: 98217
MINIMAX estimate: -1010

--- As we can see the program is playing as Black and moving a black piece on the board

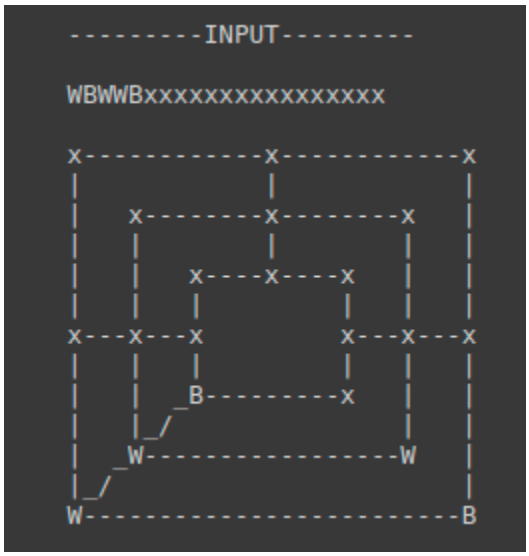
**Console output 1:** python MiniMaxGameBlack.py board2.txt board1.txt 6

Input position: WBWxBxBWWxxxBxxBxxWxx
Output position: WxWxBxBWWxxBBxxBxxWxx

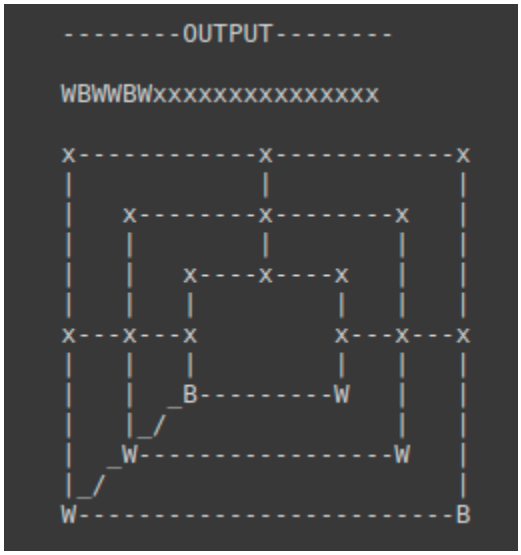
Positions evaluated by static estimation: 37275
MINIMAX estimate: -7

## 7. Opening on given static estimation:

When given the input: **WBWWBxxxxxxxxxxxxxxxx** to MiniMaxOpening with depth 3




Using the given static estimation the output which we get is:




```
Input position: WBWWBxxxxxxxxxxxxxxxx Output position: WBWWBWxxxxxxxxxxxxxxx
Positions evaluated by static estimation:  3528
MINIMAX estimate:  2
```

As we can see from the output that the next move for white is to place **W** in the **6th position.**

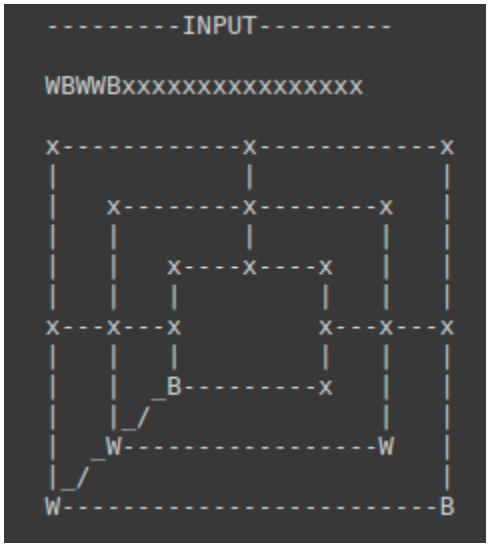
This does not seem to be a good choice. By placing W in the 6th position white is not getting any advantage. At least white should try forming a mill instead of wasting the move by placing W in the 6th position. This is because the static estimation given to us is very basic and just subtracts white pieces with black pieces which are currently on the board.

## Opening Improved static estimation:

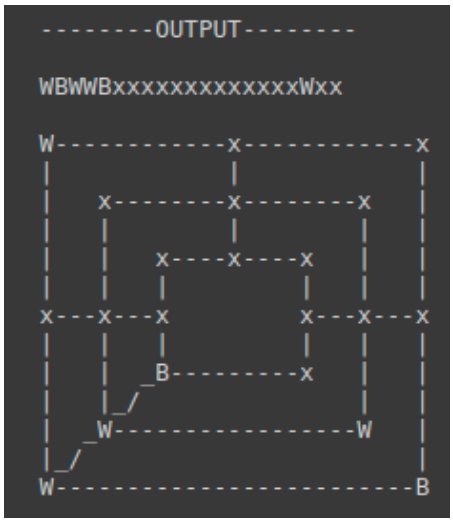
My improved static estimation takes into account DoubleMill forming, number of white pieces blocked, number of black pieces blocked and number of white and black pieces

**Example 1:**

When given the input: **WBWWBxxxxxxxxxxxxxxxx** to MiniMaxOpeningImproved with depth 3




Using the improved static estimation the output which we get is:

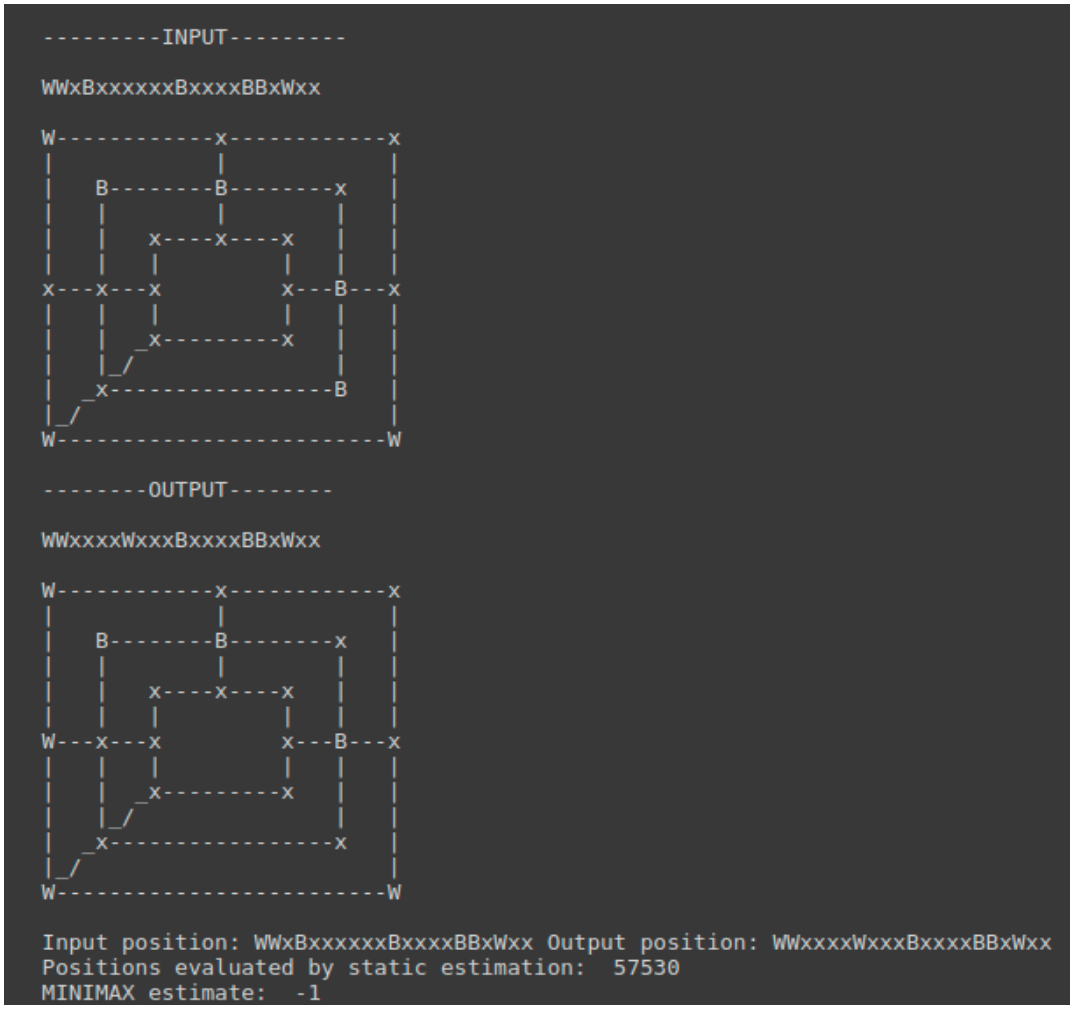


```
Input position: WBWWBxxxxxxxxxxxxxxx Output position: WBWWBxxxxxxxxxxxxxWxx
Positions evaluated by static estimation:  3528
MINIMAX estimate:  12
```

As we can see from the output that the next move for white using the improved static estimation is to place **W** in the **19th position.**
Placing W in the 19th position seems to be a better choice rather than the 6th since it provides white with the opportunity to form a mill or use that position to form two mills instead of the 6th position where you can form only one mill using that position.

**Example 2:**

When given the input: **WBWWBxxxxxxxxxxxxxxxx** to MiniMaxOpening with depth 4.
The output is closing a mill on position 6th and removing B from position 3.



This is a good move but we have a better move which leads us close to winning positions and prepares the white to be equipped with better moves when the game commences to midgame

Improved:  Same input to ImprovedMiniMaxOpening


In an improved static estimation case we can see that the white is blocking the formation of two mills. This case is crucial because if white would not have intervened and blocked the two mills there are high chances that during the midgame the black could move a black piece(from the double mill) back and forth and constantly form a mill which is a big disadvantage for White hence it first blocks the double mill formation by black so as not to be in a state where it has a disadvantage.

# 8. Midgame on given static estimation:

## Example1: Basic static estimation

When given input: **WBxBxWxWWWxxxxxxxxBxB** to MiniMaxMidgame with depth 3

```
---------INPUT---------

WBxBxWxWWWxxxxxxxxBxB

B------------x------------B
|            |            |
|   x--------x--------x   |
|   |        |        |   |
|   |   x----x----x   |   |
|   |   |         |   |   |
x---W---W         W---x---x
|   |   |         |   |   |
|   |  _x---------W   |   |
|   |_/               |   |
|  _x-----------------B   |
|_/                        |
W--------------------------B

--------OUTPUT--------

WBxBWxxWWWxxxxxxxxBxB

B------------x------------B
|            |            |
|   x--------x--------x   |
|   |        |        |   |
|   |   x----x----x   |   |
|   |   |         |   |   |
x---W---W         W---x---x
|   |   |         |   |   |
|   |  _W---------x   |   |
|   |_/               |   |
|  _x-----------------B   |
|_/                        |
W--------------------------B

Input position: WBxBxWxWWWxxxxxxxxBxB Output position: WBxBWxxWWWxxxxxxxxBxB
Positions evaluated by static estimation:  1963
MINIMAX estimate:  1961
```

We get output where white moves its piece from 5th position to 4th. Which isn't doing anything given the board. And does not form a mill which is possible in this case.

## Improved Midgame on given static estimation:

The improved static estimation takes into account number of mills formed by white and black pieces and number of white and black pieces blocked and remaining white and black pieces

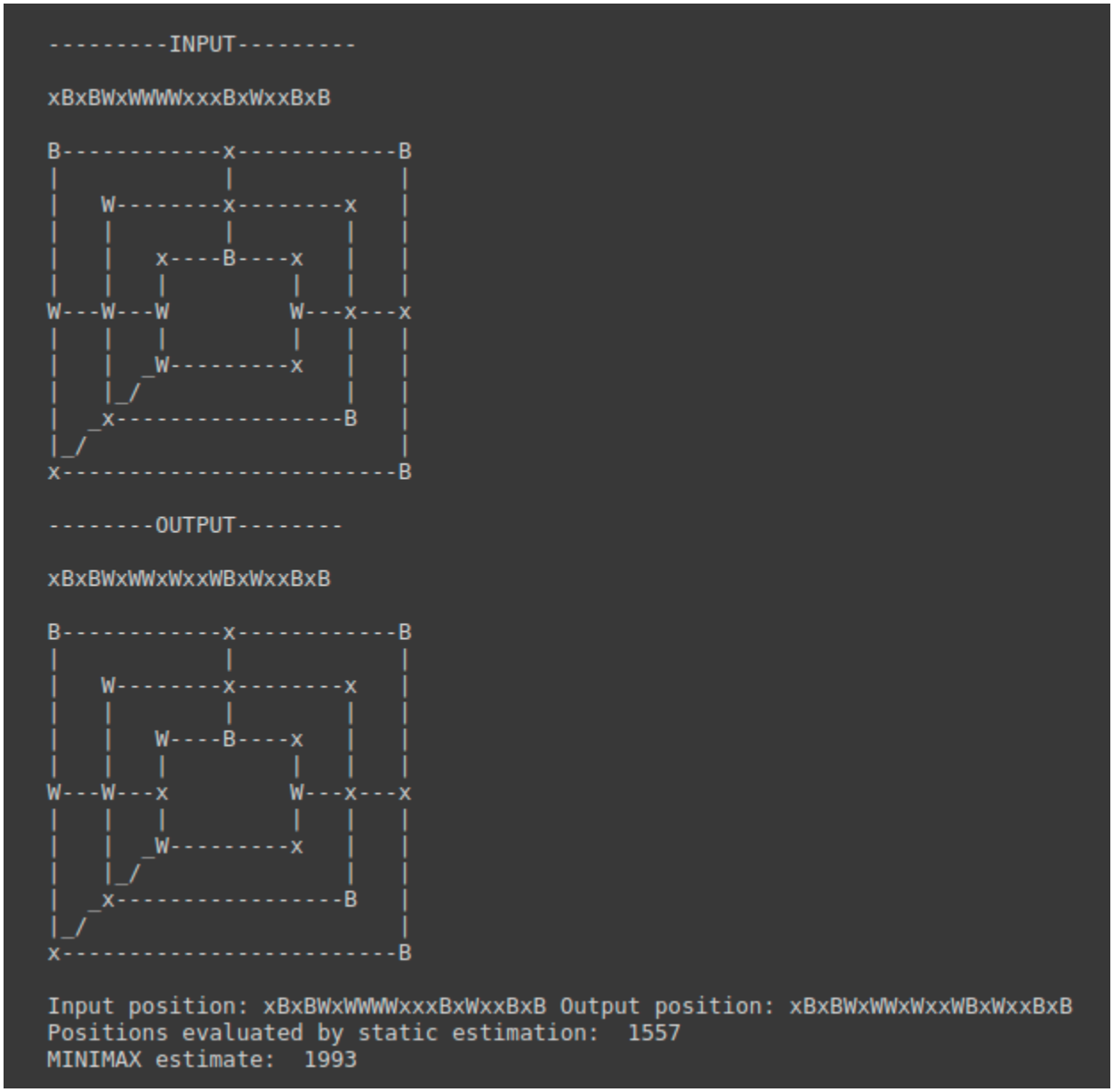
### Example 1:

When given input: **WBxBxWxWWWxxxxxxxxBxB** to MiniMaxMidgameImproved with depth 3

```
---------INPUT---------

WBxBxWxWWWxxxxxxxxBxB

B------------x------------B
|            |            |
|   x--------x--------x   |
|   |        |        |   |
|   |   x----x----x   |   |
|   |   |         |   |   |
x---W---W         W---x---x
|   |   |         |   |   |
|   |  _x---------W   |   |
|   |_/               |   |
|  _x-----------------B   |
|_/                        |
W--------------------------B

--------OUTPUT--------

xBxBxWWWWWxxxxxxxxBxx

B------------x------------x
|            |            |
|   x--------x--------x   |
|   |        |        |   |
|   |   x----x----x   |   |
|   |   |         |   |   |
W---W---W         W---x---x
|   |   |         |   |   |
|   |  _x---------W   |   |
|   |_/               |   |
|  _x-----------------B   |
|_/                        |
x--------------------------B

Input position: WBxBxWxWWWxxxxxxxxBxB Output position: xBxBxWWWWWxxxxxxxxBxx
Positions evaluated by static estimation:  1963
MINIMAX estimate:  150
```

We get output where white moves its piece from 1st position to 7th and closes a mill which is not the case in the basic static estimation. Also the white smartly removes the black piece from position 21st removing the possibility of mill forming in the potential future.

**Example 2: Basic static estimation for midgame**
When given input: **xBxBWxWWWWxxxBxWxxBxB** to MiniMaxMidgame with depth 3




We can see a unusual move of white moving from position 9th to 13th. Whereas it had a chance to form a mill by moving white from 5th position to 3rd position.

**Example 2: Improved static estimation for midgame**
When given input: **xBxBWxWWWWxxxBxWxxBxB** to MiniMaxMidgameImproved with depth 3

```
---------INPUT---------

xBxBWxWWWWxxxBxWxxBxB

B------------x------------B
|            |            |
|   W--------x--------x   |
|   |        |        |   |
|   |   x----B----x   |   |
|   |   |         |   |   |
W---W---W         W---x---x
|   |   |         |   |   |
|   |  _W---------x   |   |
|   |_/              |   |
|  _x----------------B   |
|_/                      |
x------------------------B

--------OUTPUT--------

xxWBxxWWWWxxxBxWxxBxB

B------------x------------B
|            |            |
|   W--------x--------x   |
|   |        |        |   |
|   |   x----B----x   |   |
|   |   |         |   |   |
W---W---W         W---x---x
|   |   |         |   |   |
|   |  _x---------x   |   |
|   |_/              |   |
|  _W----------------B   |
|_/                      |
x------------------------x

Input position: xBxBWxWWWWxxxBxWxxBxB Output position: xxWBxxWWWWxxxBxWxxBxB
Positions evaluated by static estimation:  1557
MINIMAX estimate:  280
```

We can see that the white moves to our desired position where it closes the mill instead of doing a move from 9th to 13th which was done previously when using basic static estimation. Here in this case the white moves its pieces from 5th position to 3rd position and thus closing the mill and removing black piece from position 2.

Most of the cases my improved static estimation gives same output as the given static estimation but there are some cases where basic static estimation does not take into consideration mill forming which is addressed in my static estimation