

CS 6364 – Artificial Intelligence
Programming Project –Nine men Morris

Name: Revanth Segu

Given below are few input board scenarios and output given by

1. java MinMaxOpening board1.txt board2.txt 5

a)

Input board is: xxxxxxxxWWWxWWxBBBBxxxx

Board Position: xxxxxxxxWWWWWxBBBBxxxx

Positions evaluated by static estimation: 993198

MINMAX estimate: 2.0

b)

Input board is: xxBBxxxxWWBxxxxBBxxWxxB

Board Position: WxBBxxxxWWBxxxxBBxxWxxB

Positions evaluated by static estimation: 906166

MINMAX estimate: -3.0

2. java MinMaxGame board1.txt board2.txt 5

a)

Input board is: xxxxxxxxWWWxWWxBBBBxxxx

Board Position: xxxxxxxxWWWxWxBBBBxxxx

Positions evaluated by static estimation: 160524

MINMAX estimate: -4.0

b)

Input board is: xxBBxxxxWWBxxxxBBxxWxxB

Board Position: xxBBxxxxWBxxxWBBxxWxxB

Positions evaluated by static estimation: 22960

MINMAX estimate: -3013.0

3. java ABOpening board1.txt board2.txt 2

a)

Input board is: xxxxxxxxWWWxWWxBBBBxxxx

Board Position: xxxxxxxxWWWWWxBBBBxxxx

Positions evaluated by static estimation: 490903

MINMAX estimate: 2.0

b)

Input board is: xxBBxxxxWWBxxxxBBxxWxxB

Board Position: xxBBxxxxWWBxxxWBBxxWxxB

Positions evaluated by static estimation: 368032

MINMAX estimate: -3.0

4. java ABGame board1.txt board2.txt 2

a)

Input board is: xxxxxxxxWWWxWWxBBBBxxxx

Board Position: xxxxxxxxWWWxWxBBBBWxxx

Positions evaluated by static estimation: 25170

MINMAX estimate: -4.0

b)

Input board is: xxBBxxxxWWBxxxxBBxxWxxB

Board Position: xxBBxxxxxWBxxxWBBxxWxxB

Positions evaluated by static estimation: 6168

MINMAX estimate: -3013.0

5. java MinMaxOpeningBlack board1.txt board2.txt 2

a)

Input board is: xxxxxxxxWWWxWWxBBBBxxxx

Board Position: xxxxxxxxWWWxWBBBBBxxxx

Positions evaluated by static estimation: 890312

MINMAX estimate: 3.0

b)

Input board is: xxBBxxxxWWBxxxxBBxxWxxB

Board Position: xxBBxxxxWWBxxxBBBxxxxxB

Positions evaluated by static estimation: 1157734

MINMAX estimate: 7.0

6. java MinMaxGameBlack board1.txt board2.txt 2

a)

Input board is: xxxxxxxxWWWxWWxBBBBxxxx

Board Position: xxxxxxxxWWWxWWBxBBBxxxx

Positions evaluated by static estimation: 118979

MINMAX estimate: -10.0

b)

Input board is: xxBBxxxxWWBxxxxBBxxWxxB

Board Position: xxBBxxxxxWxxxxBBBxxWxxB

Positions evaluated by static estimation: 7553

MINMAX estimate: 10000.0

7. java MinMaxOpeningImproved board1.txt board2.txt 5

a)

Input board is: xxxxxxxxWWWxWWxBBBBxxxx

Board Position: xxxxxxxxWWWWWxBBBBxxxx

b)

Input board is: xxBBxxxxWWBxxxxBBxxWxxB

Board Position: xxBBxxxxWWBxxxWBBxxWxxB

8. java MinMaxGameImproved board1.txt board2.txt 5

a)

Input board is: xxxxxxxxWWWxWWxBBBBxxxx

Board Position: xxxxxxxxWWWxWxBBBBxxxx

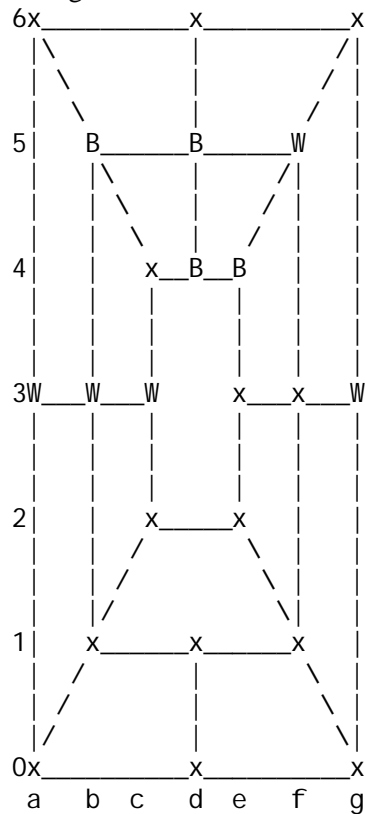
b)

Input board is: xxBBxxxxWWBxxxxBBxxWxxB

Board Position: xxBBxxxxWWBxxxWBBxxxxxB

Other than that there is a method which prints input and output board structure which can easily be understood by humans one example is shown below.

String board: xxxxxxxxWWWxxWxBBBBwxxx



Also there is a method to print tree constructed using minmax though the tree is very big and complex to be printed we can always print the tree up to depth two or three to have clear idea of what is happening.

All we need to do to print the tree is uncomment a line in code which is shown below

```
//printTreeNode(root);
```

Example of a tree printed for depth 1(the below tree shows all the child nodes and board positions)

Input board: xxxxxxxxWWWxWWxBBBBxxxx

```
StatEst: 3.0; Board i s: xxxxxxxxWWWWWxxBBBxxxx
| StatEst: 2.0; Board i s: WxxxxxxxWWWxWWxBBBBxxxx
| StatEst: 2.0; Board i s: xWxxxxxxxWWWxWWxBBBBxxxx
| StatEst: 2.0; Board i s: xxWxxxxxWWWxWWxBBBBxxxx
| StatEst: 2.0; Board i s: xxxWxxxxWWWxWWxBBBBxxxx
| StatEst: 2.0; Board i s: xxxxWxxxWWWxWWxBBBBxxxx
| StatEst: 2.0; Board i s: xxxxxWxxWWWxWWxBBBBxxxx
| StatEst: 2.0; Board i s: xxxxxxWxWWWxWWxBBBBxxxx
| StatEst: 2.0; Board i s: xxxxxxxWWWxWWxBBBBxxxx
| StatEst: 3.0; Board i s: xxxxxxxxWWWWWxxBBBxxxx
| StatEst: 3.0; Board i s: xxxxxxxxWWWWWxBxBBBxxxx
| StatEst: 3.0; Board i s: xxxxxxxxWWWWWxBBxBxxxx
| StatEst: 3.0; Board i s: xxxxxxxxWWWWWxBBBxxxxx
| StatEst: 2.0; Board i s: xxxxxxxxWWWxWWWBBBxxxxx
```

```
|StatEst: 2.0; Board i s: xxxxxxxxWWWxWWxBBBBWxxx  
|StatEst: 2.0; Board i s: xxxxxxxxWWWxWWxBBBBxWxx  
|StatEst: 2.0; Board i s: xxxxxxxxWWWxWWxBBBBxxWx  
|StatEst: 2.0; Board i s: xxxxxxxxWWWxWWxBBBBxxxW
```

The current static estimation does not consider number of mills in the board. It only considers number of white and black coins and number of black moves. It is also important that we progress to more number of mills in the opening game(non blockable), and in mid game it progresses to make mills which can be opened and close again(non blockable) and in the end game where hopping is allowed the heuristic focuses on number of non- blockable mills just by three coins. It also considers number of non blockable mills of the opponent. More about these will be shown in demo.