Morris Variant

Project Setup:

**Using IDE:**

* Keep all the files in a single folder along with input files. Make sure that **HelperClass.java** is present.
* Run each program in an IDE with run arguments as:  **<Input File Path> <Output File Path> <Depth>**

**Using Command Line:**

* Change current directory to this folder.
* Compile: javac cur\_dir/\*.java
* Run: java cur\_dir /**MiniMaxOpening** **<Input File Path> <Output File Path> <Depth>**

**Test Cases for AB Savings**

**1.MiniMaxOpening:**

Input Board Position : WWxxxxxWWWWBBBBxxBWBxB Depth : 5

Output Board Position : WWWxxxxWWWWBBxBxxBWBxB

Positions evaluated by static estimation : 841327

MINMAX Estimate : 2

**1.ABOpening:**

Input Board Position : WWxxxxxWWWWBBBBxxBWBxB Depth : 5

Output Board Position : WWWxxxxWWWWBBxBxxBWBxB

Positions evaluated by static estimation : 9812

MINMAX Estimate : 2

**2.MiniMaxOpening:**

Input Board Position : WWxxxxxWWWBBBxBxxBWBBB Depth : 3

Output Board Position : WWWxxxxWWWBBBxBxxBWBBB

Positions evaluated by static estimation : 637

MINMAX Estimate : -1

**2.ABOpening:**

Input Board Position : WWxxxxxWWWBBBxBxxBWBBB Depth : 3

Output Board Position : WWWxxxxWWWBBBxBxxBWBBB

Positions evaluated by static estimation : 67

MINMAX Estimate : -1

**We can observe that ABOpening is evaluating less number of positions when compared to MiniMaxOpening.**

**1.MiniMaxGame:**

Input Board Position : WWxxxxxWWWWBBBBxxBWBxB Depth : 2

Output Board Position : WWxxxxxWWWWBBBBWxxxBxB

Positions evaluated by static estimation : 410

MINMAX Estimate : 981

**1.ABGame:**

Input Board Position : WWxxxxxWWWWBBBBxxBWBxB Depth : 2

Output Board Position : WWxxxxxWWWWBBBBWxxxBxB

Positions evaluated by static estimation : 355

MINMAX Estimate : 981

**2.MiniMaxGame:**

Input Board Position : WWxxxxxWWWBBBxBxxBWBBB Depth : 3

Output Board Position : WWxWxxxWxWBBBxBxxBWBBB

Positions evaluated by static estimation : 755

MINMAX Estimate : -1009

**2.ABGame:**

Input Board Position : WWxxxxxWWWBBBxBxxBWBBB Depth : 3

Output Board Position : WWxWxxxWxWBBBxBxxBWBBB

Positions evaluated by static estimation : 343

MINMAX Estimate : -1009

**We can observe that ABGame is evaluating less number of positions when compared to MiniMaxGame.**

**Improved Static Estimation**

Let's define:

* noOfWhites = the number of white pieces on the board
* noOfWhites = the number of black pieces on the board
* numWhitePiecesAtRisk = the number of white pieces at risk of being removed if black makes a mill
* numBlackPiecesAtRisk = = the number of black pieces at risk of being removed if white makes a mill
* noOfWhtMillPcs = the number of white pieces that are part of a mill
* noOfBlcMillPcs = the number of black pieces that are part of a mill
* noOfWhiteMoves = the number of legal moves available to white
* noOfBlackMoves = the number of legal moves available to black

Improved Static Estimate Function:

int score = 100 \* (noOfWhites - noOfBlacks) + 100 \* (noOfWhiteMoves - noOfBlackMoves) + 50 \* (noOfWhtMillPcs - noOfBlcMillPcs) - 10 \* (numWhitePiecesAtRisk - numBlackPiecesAtRisk);  
if (noOfBlacks <= 2) return 10000;  
else if (noOfWhites <= 2) return -10000;  
else if (noOfBlackMoves == 0) return 10000;  
else return score;

The improved static estimate function consider various other factors like number of white moves, number of white/black pieces in mills , number of white/black pieces at risk of being removed if the opponents makes a mill inaddition to the given number of white/black pieces and number of black moves. Hence it gives better moves than the given simple function.

**Test Cases for Improved Static Estimation**

**1.MiniMaxOpening:**

Input Board Position : BWBWWWBBxWxxxBWxWxxBBW Depth : 4

Output Board Position : xWBWWWBBWWxxxBWxWxxBBW

Positions evaluated by static estimation : 5046

MINMAX Estimate : 2

**1. MiniMaxOpeningImproved:**

Input Board Position : BWBWWWBBxWxxxBWxWxxBBW Depth : 4

Output Board Position : BWBWWWBBxWxxxBWxWxWBBW

Positions evaluated by static estimation : 5046

MINMAX Estimate : 2270

We can see that the Improved program gives a better step for future where we can form 3 mills in the next step where as the normal program just forms a mill and removes a black piece which can’t be the best move considering next steps.

**1.MiniMaxGame:**

Input Board Position : xWBWWWBBWWxxxBWxWxxBBW Depth : 2

Output Board Position : WxBWWWBBWWxxxBWxWxxBBW

Positions evaluated by static estimation : 37

MINMAX Estimate : 2993

**1. MiniMaxGameImproved:**

Input Board Position : xWBWWWBBWWxxxBWxWxxBBW Depth : 2

Output Board Position : xWBWWWBBWWxxxBxxWWxBBW

Positions evaluated by static estimation : 37

MINMAX Estimate : 1170

We can see that the Improved Game produced a better step than the normal program as we can create a mill by moving W in 6g to 5f in next step where as there is no way to create a mill in normal program in future.