Artificial Intelligence Project

Morris Variant Game

Morris Game

Game is played between two players each having 9 pieces each. The Player with 2 pieces left or all movable position are blocked, that player loses. Game is divided into three phases

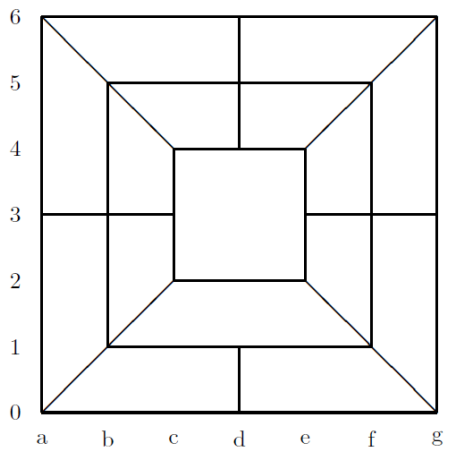
a) Opening: Players take turns placing their 9 pieces - one at a time - on any vacant board intersection spot.

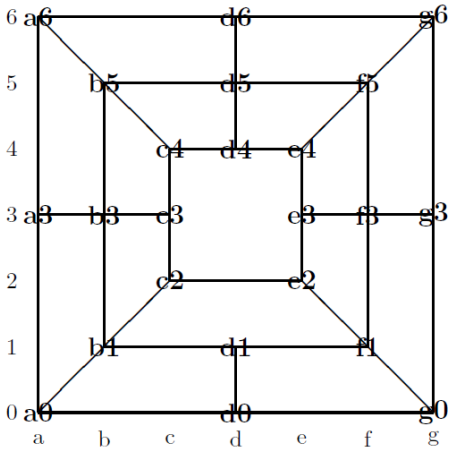
b) MidGame: Players take turns moving one piece along a board line to any adjacent vacant spot.

c) EndGame: A player down to only three pieces may move a piece to any open spot, not just an adjacent one (hopping).

Below is the Morris board and its positions.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
a0	d0	g0	b1	d1	f1	c2	e2	a3	b3	c3	e3	f3	g3	c4	d4	e4	b5	d5	f5	a6	d6	g6





Mills: Any time in game if a player gets three of their pieces on the same straight board line, then one of the opponent's isolated pieces is removed from the board.

Project Details

Programming Language: Java

OS: Windows 10

IDE: Eclipse IDE for Java Developers

Results

Part 1: Minimax algorithm

MinimaxOpening

Input

```
Board Position:       XXXWXXXBXWXBXXXXXXXXBWW

Depth:                3
```

Output

```
BoardPosition:        XXXWXXXBXWXBXXXXWXXXBWW
Positions Evaluated:  5831
MINIMAX estimate:     3
```

MinimaxGame

Input

```
Board Position:       XXXWXXXBXWXBXXXXXXXXBWW

Depth:                3
```

Output

```
BoardPosition:        WXXXXXXBXWXBXXXXXXXXBWW
Positions Evaluated:  6267
MINIMAX estimate:     -51
```

Part 2: Alpha-Beta(optimization for MiniMax)

ABOpening

Input

```
Board Position:       XXXWXXXBXWXBXXXXXXXXBWW

Depth:                3
```

Output

```
BoardPosition:        XXXWXXXBXWXBXXXXWXXXBWW
Positions Evaluated:  1848
MINIMAX estimate:     3
```

ABGame

Input

```
Board Position:       XXXWXXXBXWXBXXXXXXXXBWW

Depth:                3
```

Output

```
BoardPosition:        WXXXXXXBXWXBXXXXXXXXBWW
Positions Evaluated:  1253
MINIMAX estimate:     -51
```

Part 3: Play a game for Black using MiniMax

MinimaxOpening

Input

```
Board Position:       XXXWXXXBXWXBXXXXXXXXBWW

Depth:                3
```

Output

```
BoardPosition:        XXXXXXXBXWXBXXXXBXXXBWW
Positions Evaluated:  6004
MINIMAX estimate:     1
```

MinimaxGame

Input

```
Board Position:       XXXWXXXBXWXBXXXXXXXXBWW

Depth:                     3
```

Output

```
BoardPosition:        XXXWXXXXXWXBXBXXXXXXBWW
Positions Evaluated:  30801
MINIMAX estimate:     -50
```

Part 4: Improved Static Estimation for MiniMax

MinimaxOpeningImproved

Input

```
Board Position:       XXXWXXXBXWXBXXXXXXXXBWW

Depth:                3
```

Output

```
BoardPosition:        XWXWXXXBXWXBXXXXXXXXBWW
Positions Evaluated:  5831
MINIMAX estimate:     1
```

MinimaxGameImproved

Input

```
Board Position:       XXXWXXXBXWXBXXXXXXXXBWW

Depth:                3
```

Output

```
BoardPosition:        XXXWXXXBXWXBXXXXXXWXBXW
Positions Evaluated:  6267
MINIMAX estimate:     1949
```

Inferences from above results:

1) **Minimax and Alpha-Beta Comparison**

   For same input minimax opening and ABOpening have same output board position but different in number of positions evaluated. Same is situation for MiniMaxGame and ABGame.

   a) Example 1 for Opening game Phase

   Input

   Board Position: XXXWXXXBXWXBXXXXXXXXBWW Depth: 3

   Output for MinimaxOpening

   BoardPosition: XXXWXXXBXWXBXXXXWXXXBWW
   Positions Evaluated: 5831
   MINIMAX estimate: 3

   Output for Alpha Beta Opening

   BoardPosition: XXXWXXXBXWXBXXXXWXXXBWW
   Positions Evaluated: 1848
   MINIMAX estimate: 3

   b) Example 2 for opening game Phase

   Input

   Board Position: XBBXXXWWXXXXXXXXXXXXXXX Depth: 3

   Output for MinimaxOpening

   BoardPosition: WBBXXXWWXXXXXXXXXXXXXXX
   Positions Evaluated: 6672
   Static estimate: 1

   Output for Alpha Beta Opening

   BoardPosition: WBBXXXWWXXXXXXXXXXXXXXX
   Positions Evaluated: 356
   Static estimate: 1

a) Example 1 for MidEndGame Phase

Input

Board Position: XXXWXXXBXWXBXXXXXXXXBWW Depth: 3

Output for MiniMaxGame

```
BoardPosition:        WXXXXXXBXWXBXXXXXXXXBWW
Positions Evaluated:  6267
MINIMAX estimate:     -51
```

Output for ABGame

```
BoardPosition:        WXXXXXXBXWXBXXXXXXXXBWW
Positions Evaluated:  1253
MINIMAX estimate:     -51
```

b) Example 2 for MidEndGame phase

Input

Board Position: WXWWBBBXBXXXWWXXBWWWWWX Depth: 3

Output for MiniMaxGame

```
BoardPosition:        WXWWXBBXBXXXWXXXBWWWWWW
Positions Evaluated:  3103
Static estimate:      6970
```

Output for ABGame

```
BoardPosition:        WXWWXBBXBXXXWXXXBWWWWWW
Positions Evaluated:  550
Static estimate:      6970
```

We can see that Alpha Beta optimizes the minimax algorithm.

2) **Standard Static Estimation and Improved Static Estimation**

Improved static estimation produces different board positions compared to standard Static estimation positions but number of positions evaluated is same.

a) Example 1 Opening Phase

Input

```
Board Position: XXXWXXXBXWXBXXXXXXXXBWW Depth: 3
```

Output for MinimaxOpening

```
BoardPosition:      XXXWXXXBXWXBXXXXWXXXBWW
Positions Evaluated:  5831
MINIMAX estimate:     3
```

Output for MiniMaxOpening Improved

```
BoardPosition:      XWXWXXXBXWXBXXXXXXXXBWW
Positions Evaluated:  5831
MINIMAX estimate:     1
```

b) Example 2 Opening phase

```
Board Position: XXXWBXXBXWXBXXXXXXXXBWW Depth: 3
```

Output for MinimaxOpening

```
BoardPosition:      XXXWBXXBXWXBXXXXWXXXBWW
Positions Evaluated:  5427
Static estimate:      2
```

Output for MiniMaxOpening Improved

```
BoardPosition:      XWXWBXXBXWXBXXXXXXXXBWW
Positions Evaluated:  5427
Static estimate:      0
```

a) Example 1 for MidEndGame Phase

Input

```
Board Position: XXXWXXXBXWXBXXXXXXXXBWW Depth: 3
```

Output for MiniMaxGame

```
BoardPosition:      WXXXXXXBXWXBXXXXXXXXBWW
Positions Evaluated:  6267
```

MINIMAX estimate: -51

Output for MiniMaxGame Improved

BoardPosition: XXXWXXXBXWXBXXXXXXWXBXW
Positions Evaluated: 6267
MINIMAX estimate: 1949

b)Example 2 for MidEndGame Phase

Input

Board Position: WXWWBBBXBXXXWWXXBWWWWWX Depth: 3

Output for MiniMaxGame

BoardPosition: WXWWXBBXBXXXWXXXBWWWWWW
Positions Evaluated: 3103
Static estimate: 6970

Output for ABGame

BoardPosition: WXWWBBBXBXXWXWXXBWWWWWX
Positions Evaluated: 3103
Static estimate: 993

Why Improved static estimation is better than given standard static estimation?

In Standard Estimation, static estimation is calculated by taking the difference between the number of white pieces and black pieces. In the mid- end game phase it is calculated by considering the number of white pieces, black pieces and number of black positions available. The standard estimation is calculating the optimal based on only on the current board positions nowhere it is considering the next best possible moves.

According to handouts given, In the opening phase the static estimation is calculated by taking difference of number of white and black pieces. Further, for Mid End Game Static estimation is calculated based on the number of available black moves, white pieces and black pieces that are currently on the board.

However, in Improved static estimation, there we add other parameters like number of adjacent blocked position for all pieces on the board, which gives more insight of the game and gives better optimal board positions than standard static estimation.