AI Draughts

Martijn Gösgens Student number: xxxxxxx
Dominique Sommers Student number: 0895679

March 1, 2016

1 Algorithms

1.1 Alpha-Beta Search

1.2 Iterative Deepening

1.3 Evaluation Function

Algorithm 1 State Evaluation

```
if AI has no pieces left then
    Return $Integer.MINVALUE$
else if Opponent has no pieces left then
    Return $Integer.MAXVALUE$
else
    for all $piecesp$ do
        $pieceValue = 100$
        if $p$ is a king then
            $p.pieceValue* = 2$
            if $p$ is on the side then
                $p.pieceValue+ = 20$
        else if $p$ is black then
            if $p$ is on row 9 then
                $p.pieceValue+ = 20$
            else
                $p.pieceValue+ = (9 - rowNumber) * 3$
        else
            if $p$ is on row 0 then
                $p.pieceValue+ = 20$
            else
                $p.pieceValue+ = rowNumber * 3$
    blackValue $\sum_{p \epsilon pieces} p.pieceValue$ where $p$ is black.
    whiteValue $\sum_{p \epsilon pieces} p.pieceValue$ where $p$ is white.
    if AI is playing black then
        return $blackValue/whiteValue$
    else
        return $whiteValue/blackValue$
```

For the evaluation function, a score system is made for the draughts pieces. Initially, every piece is worth 100 points. A king is worth twice as much as a normal piece. Now, for kings it is better to stay at the side of the board rather than somewhere in the middle [1]. So there are 20 points added to the score of a king when it is positioned at the side in order to give this state of position a higher score. For normal pieces, the defense line is valuable to keep the opponent from getting a king. So again, 20 points are added to the score of pieces positioned on the first line. Pieces further on the board are more likely to get jumped, but is also more likely to jump it self. Also, the closer it is to the opponent's side, the closer it is to become a king. For every line it is moved forward, the piece receives an extra 3 points to add to it's score. So, a maximum of 24 points can be added, when the piece is one row from becoming a king.
After evaluating each piece value, the final value of the state is calculated as the ratio between the value of the AI's pieces and the value of the opponent's pieces. This will give a better approximation to the value of the state than taking the difference between the two, since for example the state with 5 pieces to 1 is much strong than with 20 pieces to 16, although the difference is the same.

2 Results

blablabla

3 References

[1] http://www.checkerstip.com/handling-a-king-piece-in.html

4 Contribution

Martijn Gösgens: Alpha-Beta seach, iterative deepening. Dominique Sommers: I designed the evaluation function for the AI. In order to do this, I studied good draughts patterns and searched for strong positions of particular pieces. These could be defended pieces (having a piece behind them, so they can not be jumped), defending pieces (having a piece in front of them), elbow pattern (pieces defending and being defended). However, I decided to keep the evaluation simple, since the method is called many times.

5 Manual

blablabla