Mitzi - Chess Engine

Christoph Hofer
Stefan Lew

20th January 2014

Outline

Outline

Chess Board

- Pieces and Sides are represented as Enums
- Value for a piece for a certain side is

$$10 \cdot \text{side.ordinal()} + \text{piece.ordinal()}$$

- Arrays of length 65 for Sides and Pieces
- Last element is an empty dummy position for pieces outside the board.
- Application of move via DoMove/UndoMove (no copies)

Outline

Board Evaluation

Value is computed in cp (*centipawns*), i.e. a pawns has a value of 100 cp

- Material value of pieces
- Position of pieces on the board. (Higher values in the center)
- Weak and Strong squares. (pieces covered by pawns get an higher value)
- Additional boni for rooks on a open/halfopen line and covering other pieces.
- Pawn structure: Multipawns, Twinpawns, Passed pawns, Isolated Pawns
- Bonus for castling

Outline

Basic NegaMax algorithm with $\alpha - \beta -$ pruning

```
1: if depth = 0 then
2:     return  evalBoard()
3: end if
4: value = − inf
5: moves = generateOrderedMoves()
6: for move ∈ moves do
7:     doMove(move)
8:     val = sign·NegaMax(depth - 1, −β, −α)
9:     undoMove(move)
10:    bestValue = max( bestValue, val )
11:    α = max( α, val )
12:    if α ≥ β then
13:        break
14:    end if
15: end for
16: return  sign*bestValue
```

Transposition Tables Lookup

```
1: entry = TranspositionTableLookup()
2: if entry ≠ null and entry.depth ≥ depth then
3:     if entry.flag = EXACT then
4:         return entry.value
5:     else if entry.flag = LOWERBOUND then
           α = max(α, entry.value)
6:     else if entry.flag = UPPERBOUND then
           β = min(β, entry.value)
7:     end if
8:     if α ≥ β then
9:         return entry.value
10:    end if
11: end if
12: proceed with NegaMax
```

Transposition Tables Storage

```
1:  entry.Value = bestValue
2:  if bestValue ≤ α_old then
3:      entry.Flag := UPPERBOUND
4:  else if bestValue ≥ β then
5:      entry.Flag := LOWERBOUND
6:  else
7:      entry.Flag := EXACT
8:  end if
9:  entry.depth = depth
10: TranspositionTableStore(entry)
11: return bestValue
```

Move Ordering

- If the position was found in the transposition table (but value could not used) use the saved and ordered moves. We only saved moves, which improved the best value in NegaMax.
- Rule of thumb: Most Valuable Victim - Least Valuable Aggressor.
- Killer moves: Moves, which produced a cutoff in the same depth, needs a check for legality. Usually only 2 are stored.
- Move order:
  1. from Transposition Table
  2. Killer Moves
  3. remaining moves (ordered by rule of thumb)

Further Improvements

Iterative Deepening Sequentially find the best move for depth $= 1, \ldots, n$.

Quiescence Search If base case is reached continue with NegaMax using all capture moves and promotions until either no capture or promotion is possible.

Aspiration Windows Instead of starting in each search depth with $\alpha = -\infty$ and $\beta = \infty$ use instead $\alpha = \mathit{val} - \epsilon$ and $\beta = \mathit{val} + \epsilon$.