

Kurzbeschreibung des Entwicklungsansatzes

Einleitung

Im Rahmen des Wahlpflichtmoduls KI (Künstliche Intelligenz) im Wintersemester 2009/2010 sollte eine KI für das 1911 entwickelte Brettspiel Laska realisiert werden.

Zugberechnung

Die Berechnung der Zugmöglichkeiten erfolgt über eine Analyse der Nachbarfelder. Diese sind in Form von Fakten zu jedem Feld hinterlegt und können somit effizient abgefragt werden.

Bewertung

Um durchgeführte Züge bewerten und diese anschließend miteinander vergleichen zu können wird jeder Figur auf dem Spielfeld ein bestimmter Wert zugewiesen.

Grundsätzlich werden Soldaten mit 30 Punkten und Offiziere mit 100 bewertet. Zusätzlich werden jeder Figur fünf weitere Punkte pro möglichem Zug angerechnet.

Figuren der eigenen Farbe werden mit positiven Zahlen und Figuren der gegnerischen Farbe mit negativen Zahlen bewertet.

Beispiel:

- Eigener Soldat mit einem möglichen Zug	= + 35 Punkte
- Eigener Offizier mit drei möglichen Zügen	= + 115 Punkte
- Gegn. Soldat mit zwei möglichen Zügen	= - 40 Punkte
- Gegn. Offizier ohne möglichen Zug	= - 100 Punkte

Endet ein Zug in diesem Spielbrett würde dieser mit +10 bewertet werden.

Sind keine Züge mehr möglich (keine Figur mehr vorhanden oder alle vorhandenen Figuren blockiert) wird die Wertung auf -10000 gesetzt. Somit bekommt die KI einen Anreiz das Spiel zu gewinnen.

Suchalgorithmus

Als Suchalgorithmus wird der α - β -Algorithmus in der NegaMax-Variante verwendet.

Der erste Ansatz eine reine Breitensuche zu verwenden, stellte sich als zu unperformant heraus, da bei steigender Suchtiefe eine riesige Anzahl an Stacks aufgebaut wurde, welche komplett durchlaufen werden musste. Bei dem α - β -Algorithmus werden nicht optimale Zweige nicht weiter untersucht, so dass der Stack nicht unnötig wächst und so die Suchzeit stark verringert wird.

Quellen

- Umsetzung vom α - β -Algorithmus in Prolog für das Spiel TicTacToe
http://www.csupomona.edu/~jrfisher/www/prolog_tutorial/5_3.html
- Sehr ausführliche Erläuterung zum α - β -Algorithmus
<http://www-i1.informatik.rwth-aachen.de/~algorithmus/algo19.php>
- Wikipedia : Artikel zur Alpha-Beta-Suche
<http://de.wikipedia.org/wiki/Alpha-Beta-Suche>
- Skript zur Vorlesung : „Begrenzte Suche zum Fällen von Entscheidungen“



Programmstruktur

Dateien

- Prädikate
 - boardreduce.pl
 - calculator.pl
 - expert.pl
 - gamelogic.pl
 - main.pl
 - ui.pl
 - weighing.pl
- Fakten
 - facts.pl

Wichtigste Prädikate

- **Ziehen einer eigenen Figur durch die KI**
Zum Starten der Zugberechnung dienen folgende Prädikate:
 - `mbk (Depth)`
Berechnung des besten Zuges für die schwarzen Spielsteine.
Beispielaufruf: `mbk (8)` .
 - `mwk (Depth)`
Berechnung des besten Zuges für die weißen Spielsteine.
Beispielaufruf: `mwk (8)` .Der Parameter `Depth` dient der Übergabe der zu verwendeten Suchtiefe.
- **Ziehen einer gegnerischen Figur**
Um die vom Gegner durchgeführten Züge eingeben zu können dienen folgende Prädikate:
 - `mw (StartColumn, StartRow, TargetColumn, TargetRow)`
Ziehen eines Spielsteines von `(StartColumn, StartRow)` nach `(TargetColumn, TargetRow)`.
Beispielaufruf: `mw (1, a, 2, b)` .
 - `jw (StartColumn, StartRow, TargetColumn, TargetRow)`
Schlagen eines Spielsteines in dem dieser Übersprungen wird.
Beispielaufruf: `jw (1, a, 3, c)` .
In diesem Beispiel wird der Stein auf dem Feld `(2, b)` geschlagen.
- **Zugbewertung**
Für die Bewertung eines durchzuführenden Zuges sind folgende Prädikate hervorzuheben:
 - `abSearch (Board, MyColor, CurrentColor, First, Depth, Alpha, Beta, Move, Rating)`
Dieses Prädikat realisiert den α - β -Algorithmus.
 - `getBoardValue (Board, Board, CurrentColor, Rating)`
Mit Hilfe dieses Prädikates wird das übergebene Spielbrett bewertet.

