



## Artificial Intelligence, Wintersemester 2012/2013

### Übungsblatt 8

Abgabe: 20.12.2012, Besprechung: 10.01.2013

#### Aufgabe 1 Drei Gewinnt [6 Punkte]

Betrachten Sie folgende Spielsituation in **Drei Gewinnt** (Gewinner ist also der Spieler, der es als erster schafft, **drei** seiner Spielsteine waagrecht, senkrecht oder diagonal in eine Linie zu bringen). Spieler O ist am Zug:

```
| . . . . |
| X . . . |
| O . . . |
| O . . . |
| XX . . |
| OX . . |
12345
```

Der Minimax-Spieler O wählt als nächsten Zug wieder Spalte 1, obwohl er in Spalte 2 eine Gewinnmöglichkeit von Spieler X blockieren könnte.

- (a) Wieso wählt er diese Aktion nicht aus? Machen Sie sich hierzu klar, wie der Minimax-Algorithmus im Detail funktioniert (und welche Annahmen er macht).
- (b) Unterbreiten Sie einen Vorschlag, durch welche Modifikation(en) man das Verhalten des Algorithmus intuitiver erscheinen lassen könnte, ohne die Optimalität zu verlieren. (In diesem Fall: Spieler O wählt Spalte 2)

#### Aufgabe 2 Vier Gewinnt [6 Punkte]

Betrachten Sie nun die Erweiterung des vorherigen Spiels zu **Vier Gewinnt**.

- (a) Skizzieren Sie eine heuristische Evaluierungsfunktion für dieses Spiel. Erklären Sie dazu zunächst die zugrunde liegende Idee.
- (b) Spezifizieren Sie Ihre Funktion dann kurz, aber möglichst präzise in Form einer mathematischen Gleichung, also z. B. der Form:

$$eval(s) = a_1 f_1(s) + a_2 f_2(x) + \dots ,$$

wobei Sie die von Ihnen benutzten Hilfsfunktionen in Textform definieren können, z. B. „ $f_1(s)$  ist die Anzahl der Steine von Spieler 1 auf dem Feld.“

### **Aufgabe 3 Alpha-Beta-Pruning [8 Punkte]**

Es gibt viele mögliche Verbesserungen für den Alpha-Beta-Pruning-Algorithmus. In den folgenden Teilaufgaben sollen Sie einige davon diskutieren bzw. umsetzen. Laden Sie hierzu die Datei `uebung8.zip` von der Webseite herunter.

- (a) Geben Sie eine Move-Ordering-Strategie für **Tonylanche** an, und implementieren Sie diese, indem Sie die Methode `orderMoves` in der Klasse `TonylancheMinMaxPlayer` vervollständigen. Erklären Sie außerdem die Idee, die dahinter steckt.
- (b) Zeigen Sie die Überlegenheit Ihres Move-Orderings, indem Sie die Anzahl der erzeugten Spielzustände mit und ohne Move-Ordering vergleichen. Verwenden Sie hierfür die `main`-Methode der Klasse `Uebung8.java`.
- (c) Bei der in der Vorlesung vorgestellten Version des Alpha-Beta-Prunings werden  $\alpha$  und  $\beta$  mit  $-\infty$  und  $+\infty$  initialisiert. Da die Utility-Funktion von Tonylanche nur die Werte  $-1$  und  $+1$  zurückgibt, könnten  $\alpha = -1$  und  $\beta = +1$  als Startwerte wählen. Ändern Sie die Klasse `TonylancheMinMaxPlayer` entsprechend ab und vergleichen die Ergebnisse mit denen aus Aufgabe b), einmal mit und einmal ohne Move-Ordering. Was beobachten Sie? Ist diese Änderung sinnvoll? Erläutern Sie Ihre Antwort.

Senden Sie Ihre kommentierte Implementierung sowie Ihre Ergebnisse an [florian.mittag@uni-tuebingen.de](mailto:florian.mittag@uni-tuebingen.de) und [andreas.draeger@uni-tuebingen.de](mailto:andreas.draeger@uni-tuebingen.de) mit dem Betreff „Abgabe KI-Uebung 08“. Bitte geben Sie in jeder Datei Ihren vollständigen Namen an, und senden Sie nur die Java-Dateien ein, die Sie laut Aufgabenstellung modifiziert haben.