Games.04: Suchtiefe begrenzen

Evaluierungsfunktion

Wir definieren für das Spiel Tic-Tac-Toe:

$X_n(s)$ = Anzahl der Reihen/Spalten/Diagonalen mit genau n X's und *keinem* O

$O_n(s)$ = Anzahl der Reihen/Spalten/Diagonalen mit genau n O's und *keinem* X

Für Terminalstellungen gilt:

$$\text{Utility}(s) = \begin{cases} +1 & \text{falls } X_3(s) \geq 1, \\ -1 & \text{falls } O_3(s) \geq 1, \\ 0 & \text{sonst.} \end{cases}$$

Für Nicht-Terminalstellungen:

$$\text{Eval}(s) = 3\,X_2(s) + X_1(s) - \big(3\,O_2(s) + O_1(s)\big).$$

Beispielwerte

#	Zustand (vereinfacht)	X_2	X_1	O_2	O_1	Eval(s)
1	X gewinnt (XXX in einer Linie)	0	0	0	0	$+1$ (Utility)
2	O gewinnt (OOO in einer Spalte)	0	0	0	0	-1 (Utility)
3	Voll besetzt, keiner gewinnt (Unentschieden)	0	0	0	0	0 (Utility)
4	Zwei X in einer Reihe, kein O dort	1	1	0	1	$3 \cdot 1 + 1 - (0 + 1) = 3$
5	Eine X in mehreren Linien, ein O aktiv	0	2	0	1	$0 + 2 - (0 + 1) = 1$
6	Zwei O in einer Spalte, sonst leer	0	1	1	0	$0 + 1 - (3 \cdot 1 + 0) = -2$

Begründung

Diese Evaluierungsfunktion ist im Zusammenhang mit Tic-Tac-Toe sinnvoll, weil sie:

- vorrangig Stellungen bewertet, in denen ein Spieler unmittelbar vor dem Gewinn steht (z. B. X_2 oder O_2),
- die Anzahl potentieller Gewinnlinien berücksichtigt (X_1, O_1),
- dadurch ermöglicht, dass auch bei begrenzter Suchtiefe („Cut-Off“) sinnvolle Abschätzungen getroffen werden.

Somit kann der Algorithmus auch dann gute Züge wählen, wenn nicht bis zur echten Spielendung (Terminalstellung) gesucht wird.