# Dokumentation der Computerspieler „CPU Shallow“, „CPU Deep“ und „CPU Deeper“

Neben dem Spielen gegen einen menschlichen Gegner bietet das Schachspiel zusätzlich eine Reihe an computergesteuerten Spielern an. Folglich wird die Funktionsweise und die Softwarearchitektur dreier dieser Computerspieler beschrieben.

## Die Evaluationsfunktion

Den Kern der KI liegt die Evaluationsfunktion, die die Güte einer Situation im Spiel bewertet und es somit ermöglicht, den besten Zug zu ermitteln. Die statische Funktion ist in der *GameStateAnalyzer*-Klasse unter dem Namen *analyzeGameState* zu finden. Sie liefert einen numerischen Wert, der größer wird, umso vorteilhafter die gegebene Situation für den ausgewählten Spieler ist, und kleiner, je mehr der gegnerische Spieler bevorteilt ist. Die Situation wird anhand zweier Merkmale bewertet: Der noch im Spiel vorhandenen, d.h. nicht geschmissenen Spielfiguren, und deren Position auf dem Spielfeld. Jeder Spielfigur wird hierbei ein Wert zugeschrieben, wie aus der folgenden Tabelle ersichtlich wird:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 10 | 50 | 30 | 30 | 90 | 900 |

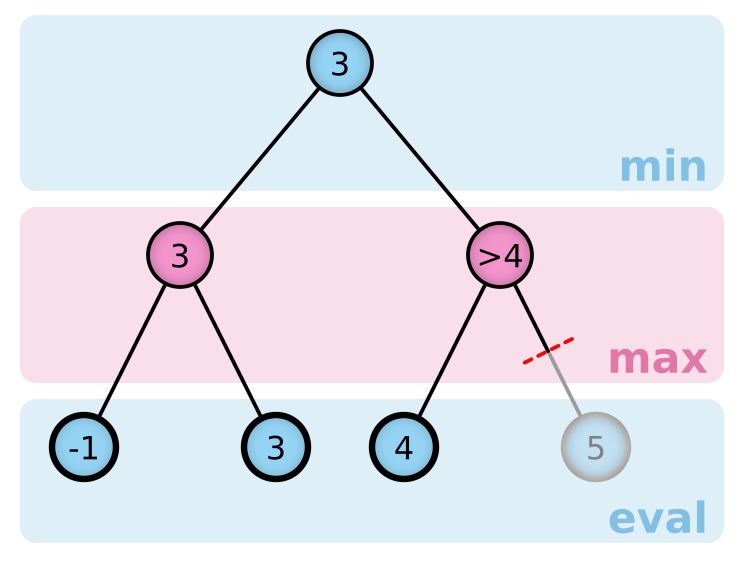
Zusätzlich zu dem statischen Wert, wird jeder Figur eine Punktezahl zugeschrieben, die abhängig von ihrer Position ist. Diese wird auf den Basiswert hinzuaddiert und ist von Figur zu Figur unterschiedlich.

Die folgenden zwei Beispiele zeigen die Positionsevaluationsmatrizen für den weißen Läufer und das Pferd:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** |  | | **8** | -2.0 | -1.0 | -1.0 | -1.0 | -1.0 | -1.0 | -1.0 | -2.0 | **8** | | **7** | -1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -1.0 | **7** | | **6** | -1.0 | 0.0 | 0.5 | 1.0 | 1.0 | 0.5 | 0.0 | -1.0 | **6** | | **5** | -1.0 | 0.5 | 0.5 | 1.0 | 1.0 | 0.5 | 0.5 | -1.0 | **5** | | **4** | -1.0 | 0.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.0 | -1.0 | **4** | | **3** | -1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | -1.0 | **3** | | **2** | -1.0 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.5 | -1.0 | **2** | | **1** | -2.0 | -1.0 | -1.0 | -1.0 | -1.0 | -1.0 | -1.0 | -2.0 | **1** | |  | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** |  | | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** |  | | **8** | -5.0 | -4.0 | -3.0 | -3.0 | -3.0 | -3.0 | -4.0 | -5.0 | **8** | | **7** | -4.0 | -2.0 | 0.0 | 0.0 | 0.0 | 0.0 | -2.0 | -4.0 | **7** | | **6** | -3.0 | 0.0 | 1.0 | 1.5 | 1.5 | 1.0 | 0.0 | -3.0 | **6** | | **5** | -3.0 | 0.5 | 1.5 | 2.0 | 2.0 | 1.5 | 0.5 | -3.0 | **5** | | **4** | -3.0 | 0.0 | 1.5 | 2.0 | 2.0 | 1.5 | 0.0 | -3.0 | **4** | | **3** | -3.0 | 0.5 | 1.0 | 1.5 | 1.5 | 1.0 | 0.5 | -3.0 | **3** | | **2** | -4.0 | -2.0 | 0.0 | 0.5 | 0.5 | 0.0 | -2.0 | -4.0 | **2** | | **1** | -5.0 | -4.0 | -3.0 | -3.0 | -3.0 | -3.0 | -4.0 | -5.0 | **1** | |  | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** |  | |

## Das Entscheidungsfällen

Mithilfe der oben beschriebenen Evaluierungsfunktion kann die KI entscheiden, welcher Zug gespielt werden soll. Die „CPU Shallow“ tut dies auf eine sehr einfache Art und Weise. Jeder mögliche Zug wird simuliert und die darauf folgende Situation mithilfe der Evaluationsfunktion bewertet. Gibt es einen eindeutigen besten Zug, so wird dieser gespielt. Ansonsten wird aus der Menge der besten Züge ein Zug per Zufall gewählt.

Die „CPU Deep“ geht einen Schritt weiter. Mithilfe des sog. „Minimax“-Algorithmus kann sie auch die in der Zukunft zu spielenden Züge berücksichtigen, um somit eine bessere Entscheidung zu fällen, als dies die „CPU Shallow“ tut. Dieser Algorithmus funktioniert, indem er einen Entscheidungsbaum generiert, bei dem sowohl die möglichen Züge der KI, als auch des Gegners in jeweils abwechselnd aufeinander folgenden Knoten repräsentiert werden. Vorgesehen werden ein sog. „maximizing player“, der einen möglichst hohen Evaluationswert anstrebt, sowie ein „minimizing player“, der niedrige Werte anstrebt. Da die Evaluationsfunktion den zu betrachtenden Spieler (also die KI) entgegennimmt und den für ihn vorteilhaften Situationen höhere Werte zuschreibt, ist der „maximizing player“ immer die KI selbst und der „minimizing player“ ihr Gegner. Um jetzt den besten Zug zu ermitteln, wird jedem Knoten ein Wert zugeschrieben, der jeweils der höchste bzw. niedrigste Wert (je nach Player) aller seiner Kinderknoten ist. Die Blätter erhalten ihren Wert durch die Evaluationsfunktion. Das bedeutet, dass der Minimax-Algorithmus die Bewertung des besten Zuges zurückgibt, unter der Annahme, dass sowohl die KI als auch ihr Gegner immer ihren besten Zug spielen.

Um die Performanz des Algorithmus weiter zu optimieren, wird zudem die unter dem Namen „Alpha-beta pruning“ bekannte Optimierungsmethode angewendet. Äste des Suchbaums, die zu keiner besseren Situation führen können, als bereits vorher evaluierte Äste, und somit das Resultat des „Minimax“-Algorithmus nicht beeinflussen, werden abgeschnitten, um Rechenzeit zu sparen.

Da das Durchführen des Algorithmus für alle möglichen Züge trotz dieser Optimierung immer noch einige Sekunden dauert, geschieht die Evaluation parallel auf mehreren CPU-Threads. Zu diesem Zweck besitzt die für die „CPU Deep“ zuständige Klasse *ComputerPlayerDeep* eine Subklasse namens *MoveEvaluator.* Am Anfang der Berechnung wird für jeden möglichen Zug ein eben solcher *MoveEvaluator* erstellt, der jeweils einen Zug auf seinem eigenen Thread evaluieren soll. Dadurch kann in einem Bruchteil einer Sekunde eine Entscheidung getroffen werden, wodurch eine flüssige Spielerfahrung ermöglicht wird.

Die „CPU Deeper“ ist der „CPU Deep“ sehr ähnlich. Tatsächlich wird sie von der gleichen Klasse realisiert. Die Klasse hat zwei Eigenschaften, *minDepth* und *maxDepth*, die für die „CPU Deep“ in beiden Fällen 3 betragen. *minDepth* gibt die Tiefe des initialen Suchbaums an. Wird mehr als ein bester Zug gefunden und ist *maxDepth* größer als *minDepth*, so wird die Tiefe graduell erhöht, bis entweder ein eindeutiger bester Zug ermittelt, oder die maximale Tiefe erreicht worden ist. Züge, die schlechter sind als der jeweils beste Zug werden dabei bei jeder Iteration aussortiert, sodass die Anzahl der zu evaluierenden Züge mit zunehmender Tiefe sinkt. Die maximale Tiefe der „CPU Deeper“ beträgt 5.