



# LASKA - KI

Semester:	Wintersemester 14/15
Dozent:	Prof. Dr. Meyer
Projektteam:	Jan Eichel
	Chris Ehrenberg
	Sven Schilling

# 1 Inhalt

2	Entwicklungsansatz .....	2
2.1	Zugberechnung .....	2
2.2	Bewertung:.....	2
2.3	Suchalgorithmus: .....	2
3	Ergebnis und Ausblick: .....	2
4	Quellenverzeichnis .....	3

## 2 Entwicklungsansatz

### 2.1 Zugberechnung

Das Board sowie die Nachbarfelder werden als Regeln abgelegt. Durch das Abfragen dieser Regeln können die möglichen Züge berechnet werden, welche ebenfalls in Form von Regeln abgelegt werden. Wenn die Möglichkeit des Schlagens besteht werden die Regeln für die "normalen" Züge gelöscht.

### 2.2 Bewertung:

Um das Spielbrett bewerten zu können, wird jeder Art Figur ein bestimmter Wert zugeteilt. Auch die Anzahl der möglichen Züge fließt mit in die Bewertung ein. Die Werte wurden in Testspielen exemplarisch ermittelt. Ein Soldat wird mit 20 Punkten und ein Offizier mit 65 Punkten bewertet. Jeder mögliche Zug wird mit 3 Punkten bewertet. Dabei werden eigene Figuren und Züge mit positiven Punkten und gegnerische mit negativen Punkten bewertet. Existieren keine eigenen Züge mehr wird die Wertung auf -10000 gesetzt, sollte der Gegner keine Züge mehr haben wird die Wertung auf 10000 gesetzt, somit erhält die KI ein Anreiz das Spiel zu gewinnen und achtet zugleich darauf nicht zu verlieren.

### 2.3 Suchalgorithmus:

Als Suchalgorithmus wird der Alpha-Beta-Algorithmus verwendet, welcher eine verbesserte Variante des Mini-Max-Algorithmus darstellt, da Zweige die nicht mehr in Frage kommen abgeschnitten werden. Hiervon wurde sich eine erhöhte Suchtiefe versprochen. Zudem werden die einzelnen Knoten und ihre Bewertung in Regeln, anstatt einer Liste, abgespeichert, da das Hinzufügen von Regeln nur zwei Interferenzen kostet. Auch hiervon wurde sich eine höhere Suchtiefe versprochen da die Annahme getroffen wurde, dass die Anzahl der Interferenzen mit der benötigten Zeit einhergeht. Zudem wird anhand von Spielfeldkriterien die Suchtiefe eingeschränkt. Die Kriterien sind wie folgt festgelegt:

- Startsuchtiefe = 8
- Es sind mehr als 2 Offiziere vorhanden, Suchtiefe = 4
- Es sind mehr als 12 leere Felder vorhanden, Suchtiefe = 5
- Es sind mehr als 10 leere Felder vorhanden, Suchtiefe = 6
- Es sind mehr als 8 leere Felder vorhanden, Suchtiefe = 7

Damit soll das Zeit überschreiten bei wachsendem Baum verhindert werden.

## 3 Ergebnis und Ausblick:

Die Annahme das die Anzahl an Interferenzen mit der Zeit einhergeht hat sich als falsch herausgestellt, da andere KIs die Listen verwendet haben deutlich schneller waren. Ein weiterer Schritt um den Suchalgorithmus zu verbessern wäre das Sortieren von vielversprechenden Zweigen nach oben. Das reduzieren der Suchtiefe im Spielverlauf hat sich als sehr nützliches System herausgestellt, da so oft Spiele durch Zeit-K.O. gewonnen werden konnten. Auch die Bewertungsfunktion scheint sinnvoll gewählt zu sein, da auch gegen andere KIs gewonnen wurde die höhere Suchtiefen hatten.

## 4 Quellenverzeichnis

Alpha-Beta-Algorithmus in Prolog am Beispiel Tic-Tac-Toe

[http://www.csupomona.edu/~jrfisher/www/prolog\\_tutorial/5\\_3.html](http://www.csupomona.edu/~jrfisher/www/prolog_tutorial/5_3.html)

Alpha-Beta-Algoritsmus in einem Video

<https://www.youtube.com/watch?v=xBXHtz4Gbdo>