



OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

Entwicklung einer künstlichen Intelligenz für Brettspiele und deren Anbindung an eine Touch-Hardware über mobile Endgeräte

An der Fakultät für Informatik und Mathematik der
Ostbayerischen Technischen Hochschule Regensburg
im Studiengang
Technische Informatik

eingereichte

Bachelorarbeit

zur Erlangung des akademischen Grades des
Bachelor of Science (B.Sc.)

Vorgelegt von: Korbinian Federholzner
Matrikelnummer: 3114621

Erstgutachter: Prof. Dr. Carsten Kern
Zweitgutachter: Prof. Dr. Daniel Jobst

Abgabedatum: 31.08.2020

Erklärung zur Bachelorarbeit

1. Mir ist bekannt, dass dieses Exemplar der Abschlussarbeit als Prüfungsleistung in das Eigentum der Ostbayerischen Technischen Hochschule Regensburg übergeht.
2. Ich erkläre hiermit, dass ich diese Abschlussarbeit selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Regensburg, den 24. Mai 2020

Korbinian Federholzner

Zusammenfassung

In der folgenden Arbeit wird ...

Inhaltsverzeichnis

I	Abkürzungsverzeichnis	VII
1	Einleitung	1
1.1	Motivation	1
1.2	Aufgabenstellung	1
1.3	Struktur dieser Arbeit	1
2	Grundlagen	2
2.1	Dame	2
2.1.1	Internationale Dame	2
2.2	Minimax	3
2.3	Alpha-Beta Pruning	4
2.4	Monte Carlo Tree Search (MCTS)	4
3	Anforderungsanalyse	4
3.1	Anwendungsszenario	4
3.2	Anforderungen an die Software	4
3.2.1	Nichtfunktionale Anforderungen	4
3.2.2	Funktionale Anforderungen	4
3.2.3	Zusammenfassung der Anforderungen	4
4	Architektur	5
4.1	Überblick	5
4.2	Gameserver	5
4.3	Graphische Oberfläche	5
4.4	KI Client	5
5	Hardware	5
5.1	Raspberry Pi	5
5.2	Touch Monitor	5
6	Implementierung	5
6.1	Eingesetzte Softwarekomponenten	5
6.1.1	Programmiersprachen und Frameworks	5
6.1.2	Datentransferprotokolle	5
6.2	Gameserver	5
6.2.1	Netzwerkspezifikation des Gameservers	5
6.3	Graphische Oberfläche	5
6.3.1	Gegebene React Anwendung	5

6.3.2	Erweiterungen	5
6.4	KI Client	5
6.4.1	Vergleich der KI Algorithmen	5
7	Testing	5
7.1	Integrationstest	5
7.2	Ergebnisse	5
8	Fazit und Ausblick	6

Abbildungsverzeichnis

1	DameSpielfeld	3
2	minimax	3

Tabellenverzeichnis

I Abkürzungsverzeichnis

1 Einleitung

Das Thema dieser Arbeit ist die Implementierung einer künstlichen Intelligenz für diverse Brettspiele. Dabei soll die resultierende Software auf einem von einem Raspberry Pi gesteuerten Touch-Bildschirm laufen, durch welchen ein Benutzer die künstliche Intelligenz herausfordern kann. Außerdem gibt es die Möglichkeit, sich mit der Hardware über ein mobiles Endgerät wie einem Smartphone zu verbinden, um die KI oder den Spieler, der den Touch-Bildschirm bedient, herauszufordern.

1.1 Motivation

Das Feld der künstlichen Intelligenz ist momentan eines der sich am schnellsten entwickelnden Felder der Informatik. Dabei spielt die Spieltheorie schon seit Anfang eine große Rolle. So bieten klassische Brettspiele wie z. B. Schach, Dame oder Mühle nicht nur eine klar definierte Abstraktion von Problemen der realen Welt, sondern sie können auch von dem Großteil der Bevölkerung verstanden und gespielt werden. Das Meistern einer dieser Brettspiele wird auch oft mit hohem Grad an Intelligenz gleichgesetzt. Viele Algorithmen, die in der Spieltheorie entwickelt wurden, haben sich auch erfolgreich auf andere Felder der Informatik übertragen lassen. Ebenso hat sich die Art, wie Brettspiele gespielt werden, durch das Verwenden von künstlicher Intelligenz auch verändert, da die KI Züge in Betracht zieht, die auf den ersten Blick recht ungewöhnlich und nachteilhaft aussehen, sich aber als extrem stark herausstellen. Diese Arbeit versucht eine künstliche Intelligenz für Brettspiele, wie Dame, zu implementieren.

1.2 Aufgabenstellung

Im Rahmen der Bachelorarbeit soll eine künstliche Intelligenz entwickelt werden, welche Brettspiele, wie Dame, spielen kann. Gegeben ist eine Software, bei welcher man in der Lage ist, das Spiel ReversiXT (Reversi Extreme) gegen eine KI, sowie sich selbst zu spielen. Diese Software soll um einen Game Server, die KI und das neue Spiel in der GUI, erweitert werden. Außerdem soll ein Benutzer in der Lage sein, sich mit seinem Smartphone mit der Hardware zu verbinden, um neue, sowie die alte KI herausfordern zu können.

1.3 Struktur dieser Arbeit

Die Arbeit ist folgendermaßen aufgebaut:

Kapitel 2 befasst sich mit den Grundlagen zu Dame, sowie den verwendeten künstliche Intelligenz Algorithmen. Dabei werden zu Dame auch die Grundregeln der verwendeten Variante erklärt. Bei den KI Algorithmen handelt es sich um die Algorithmen, die in der Arbeit verwendet und miteinander verglichen werden.

In Kapitel 3 werden die Anforderungen, welche gefordert sind, vorgestellt.

2 Grundlagen

Dieses Kapitel gibt einen Überblick über die theoretischen Grundlagen, die für das Verständnis dieser Arbeit notwendig sind. Zunächst werden die Grundregeln des behandelten Brettspieles vorgestellt. Darauf folgt eine Erklärung der künstlichen Intelligenz Algorithmen, welche für folgenden Kapitel von großer Relevanz sind.

2.1 Dame

Dame ist eines der ältesten Brettspiele, wobei erste Varianten 3000 v. Chr. im irakischen Ur entdeckt wurden. In der heutigen Zeit werden verschiedene Varianten desselben Spieles weltweit gespielt. So wird in vielen englischsprachigen Ländern eine andere Version gespielt als im Rest der Welt, was auch als Internationale Dame bekannt ist. Unterschiede sind z. B. bei diesen Varianten, dass bei internationaler Dame, Damen beliebig viele Felder in alle Richtungen springen dürfen, in anderen Varianten jedoch nur 1 Feld. Da die Regeln der internationalen Dame weiter verbreitet sind und auch im Vereinssport praktiziert werden, wird sich diese Arbeit im folgenden auf diese Regeln fokussieren.

2.1.1 Internationale Dame

In dieser Variante des Spieles wird auf einem 10×10 Brett mit Schachbrett-Muster gespielt. Siehe Abbildung 1. Die Spielsteine sind scheibenförmig und in zwei Farben vorhanden, meist schwarz und weiß und dürfen nur auf den dunklen Feldern des Schachbrettes bewegt werden. Es gibt zwei Arten von Spielsteinen. Normale Spielsteine, welche nur in Richtung des Gegners bewegt werden, aber Rückwärts schlagen dürfen, und Damen, welche in alle Richtungen beliebig viele Felder fahren und schlagen dürfen. Allgemein herrscht Schlagzwang, was bedeutet, dass falls ein Spieler die Möglichkeit hat zu schlagen, er auch schlagen muss. Ein normaler Spielstein wird zur Dame, falls er in die hinterste Reihe des Gegners kommt. Ziel des Spieles ist es, entweder alle Steine des Gegners zu schlagen, oder den Gegner in eine Situation zu zwingen, in der er keine Züge mehr machen kann.

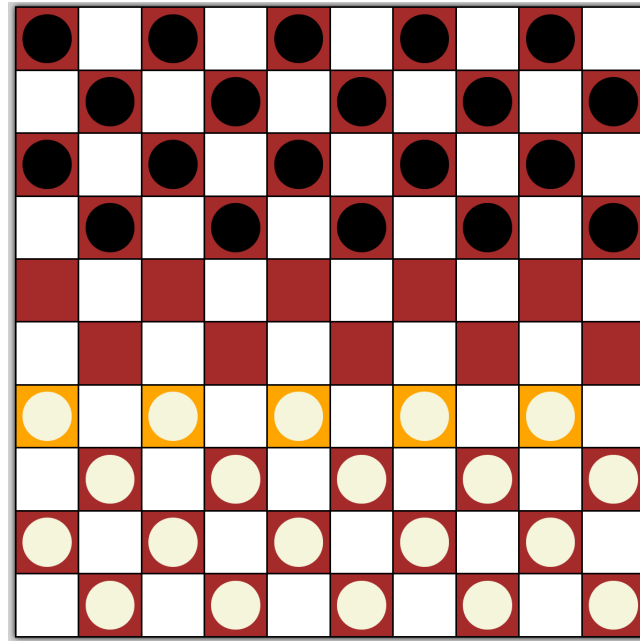


Abbildung 1: Das 10x10 Spielfeld aus der Anwendung

2.2 Minimax

Der Minimax Algorithmus wird verwendet, um einen optimalen Spielzug in Spielen mit perfekter Information zu finden. Allgemein wird eine Bewertungsfunktion verwendet, welche einen Zustand des Spieles bewertet und einen Integer Wert zurückgibt. Sehr niedrige Werte (negative) sind gut für Spieler A und sehr hohe Werte (positive) sehr gut für Spieler B. Um nun mehrere mögliche Zustände des Spieles zu berechnen und dann miteinander vergleichen zu können wird ein Suchbaum verwendet. siehe Abbildung 2. Der Algorithmus beginnt bei den untersten Blättern und geht nach oben bis zur Wurzel. Auf jeder Ebene wird versucht, je nach dem welcher Spieler an der Reihe ist, den Knoten zu maximieren oder zu minimieren. Der Wert, welcher der Wurzel zugewiesen wird, ist auch der Spielzug, der als Nächstes gespielt wird.

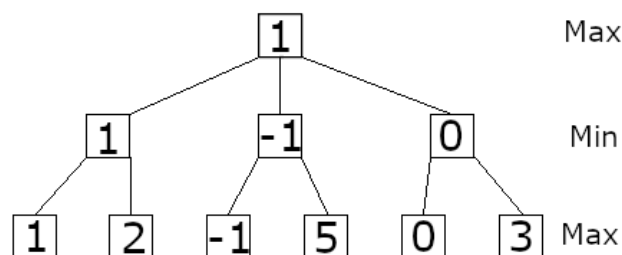


Abbildung 2: Beispiel eines Minimax Suchbaumes

2.3 Alpha-Beta Pruning

Das Alpha-Beta Pruning ist eine Optimierung zum Minimax Algorithmus. Die Idee des Algorithmus ist, dass manche Zweige des Suchbaums nicht untersucht werden müssen, da für den anderen Spieler diese Züge nicht in Frage kommen. Hierbei ist α der Wert für den Spieler, für den die niedrigen Werte besser sind und β für den anderen Spieler. Für jeden Knoten, je nach dem, ob er ein maximierender oder ein minimierender Knoten ist, wird überprüft, ob ein Kind-Knoten, welcher einen neuen Wert erhalten hat, nicht mehr vom Knoten beachtet werden muss. Z. B. ist der zu beachtende Knoten ein minimierender Knoten und hat aus seinem Linken Zweig eine 8 bekommen. Der rechte Kind-Knoten ist ein maximierender Knoten und hat von seinen Kindern eine 9 bekommen, sodass es nun egal ist, welche weiteren Werte er von seinen anderen Kindern noch bekommt. Er wird seinen Wert nicht mehr unter 8 ändern, was zur Folge hat, dass der rechte Zweig komplett ignoriert werden kann.

2.4 Monte Carlo Tree Search (MCTS)

Der Monte Carlo Tree Search Algorithmus, ist ein heuristischer Algorithmus, bei welchem von einem Zustand eines Spieles zufällig endlich viele Simulationen durchgeführt werden. Ein Vorteil des MCTS-Algorithmus gegenüber Minimax ist, dass erst am Ende eines Durchlaufs eine Bewertungsfunktion benötigt wird. Allgemein besteht der Algorithmus aus vier Schritten:

- *Selektion*: Versucht wird, einen Zustand zu finden der noch erweiterbar ist, also einen Zustand zu finden, der kein Endzustand ist und noch nicht besuchte Züge hat.
- *Expansion*: Der Spielbaum wird zufällig um einen noch nicht besuchten Zug erweitert.
- *Simulation*: Von dem gewählten Knoten aus wird nun ein Spiel zufällig bis zum Ende simuliert.
- *Backpropagation*: Das Ergebnis der Simulation wird den vorhergehenden Knoten mitgeteilt und diese werden mit diesem aktualisiert.

3 Anforderungsanalyse

3.1 Anwendungsszenario

3.2 Anforderungen an die Software

3.2.1 Nichtfunktionale Anforderungen

3.2.2 Funktionale Anforderungen

3.2.3 Zusammenfassung der Anforderungen

4 Architektur

4.1 Überblick

4.2 Gameserver

4.3 Graphische Oberfläche

4.4 KI Client

5 Hardware

5.1 Raspberry Pi

5.2 Touch Monitor

6 Implementierung

6.1 Eingesetzte Softwarekomponenten

6.1.1 Programmiersprachen und Frameworks

6.1.2 Datentransferprotokolle

6.2 Gameserver

6.2.1 Netzwerkspezifikation des Gameservers

6.3 Graphische Oberfläche

6.3.1 Gegebene React Anwendung

6.3.2 Erweiterungen

6.4 KI Client

6.4.1 Vergleich der KI Algorithmen

7 Testing

7.1 Integrationstest

7.2 Ergebnisse

8 Fazit und Ausblick