

# **Untersuchung der Lernfähigkeit verschiedener Verfahren am Beispiel von Computerspielen**

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

Thilo Stegemann  
s0539757  
Angewandte Informatik

12. Januar 2017



**Hochschule für Technik  
und Wirtschaft Berlin**

*University of Applied Sciences*

Erstprüfer: Prof. Dr. Burkhard Messer  
Zweitprüferin: Prof. Dr. Adrianna Alexander

# Todo list

Schreiben der Einführung des Kapitels 2 Grundlagen . . . . .	3
Schreibe die Einführung von Kapitel 3! . . . . .	4
Grundlagen maschinelles Lernen einfügen in Kapitel 2! . . . . .	4
Erarbeiten der Eigenschaften von überwachten und nicht überwachtem ma- schinellem Lernen In Kapitel 2 Grundlagen . . . . .	4
Figure: Hier muss noch eine passende Grafik erstellt werden. . . . .	11

# Abkürzungsverzeichnis

**ID**    Identifikator

**UI**    User Interface

# Abbildungsverzeichnis

4.1	Tic Tac Toe horizontaler Sieg . . . . .	8
4.2	Tic Tac Toe vertikaler Sieg . . . . .	9
4.3	Tic Tac Toe diagonaler Sieg . . . . .	9
4.4	Tic Tac Toe Spielfiguren setzen . . . . .	10

# Tabellenverzeichnis

3.1	Funktionale Anforderungen . . . . .	6
3.2	Nichtfunktionale Anforderungen . . . . .	6

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>iii</b>
<b>Abbildungsverzeichnis</b>	<b>iv</b>
<b>Tabellenverzeichnis</b>	<b>v</b>
<b>1 Projektvision</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Vorläufige Zielsetzung . . . . .	1
1.3 Nutzen/Zweck des Projektes . . . . .	2
1.4 Zielgruppe . . . . .	2
<b>2 Grundlagen</b>	<b>3</b>
2.1 Lernfähige Algorithmen . . . . .	3
2.1.1 Überwachtes Lernen . . . . .	3
2.1.2 Unüberwachtes Lernen . . . . .	3
2.1.3 Bestärkendes Lernen . . . . .	3
2.2 Spielentwicklung . . . . .	3
2.3 Lineare Algebra . . . . .	3
2.4 Heuristik . . . . .	3
<b>3 Problemanalyse und Anforderungsdefinition</b>	<b>4</b>
3.1 Problemanalyse . . . . .	4
3.1.1 Die Problematik . . . . .	4
3.1.2 Bereits existierende Softwarelösungen . . . . .	5
3.1.3 Gegenstandsbereich des Projektes abgrenzen . . . . .	5
3.1.4 Abgrenzung gegenüber der künstlichen Intelligenz . . . . .	5
3.2 Anforderungsdefinition . . . . .	5
3.2.1 Funktionale Anforderungen . . . . .	5
3.2.2 Nicht-funktionale Anforderungen . . . . .	5
<b>4 Modellierung und Entwurf</b>	<b>7</b>
4.1 Computerspiele . . . . .	7
4.1.1 Tic Tac Toe . . . . .	7
4.1.2 Reversi . . . . .	10
4.2 Lernverfahren . . . . .	11
4.2.1 Analyse und Auswahl der lernfähigen Algorithmen . . . . .	11

## Inhaltsverzeichnis

4.2.2	Anwendung der Algorithmen auf Computerspiele . . . . .	11
4.2.3	Konzeptuelles Training der Algorithmen . . . . .	11
4.2.4	Persistenz der Trainingsdaten . . . . .	11
<b>5</b>	<b>Implementierung</b>	<b>12</b>
5.1	Computerspiele . . . . .	12
5.2	Lernverfahren . . . . .	12
5.3	Alternative Lernverfahren . . . . .	12
<b>6</b>	<b>Validierung</b>	<b>13</b>
6.1	Messbare Testkriterien . . . . .	13
6.2	Modultests . . . . .	13
6.2.1	TicTacToe . . . . .	13
6.2.2	Reversi . . . . .	13
6.2.3	Lernverfahren 1 . . . . .	13
6.2.4	Lernverfahren 2 . . . . .	13
6.2.5	Persistenz . . . . .	13
6.3	Systemtest . . . . .	13
<b>7</b>	<b>Auswertung</b>	<b>14</b>
7.1	Belastbarkeit und Grenzen der Lernverfahren . . . . .	14
7.2	Optimale Anwendungsspiele für die Lernverfahren . . . . .	14
7.3	Gegenüberstellung der Lernverfahren . . . . .	14
7.4	Bewertung der Strategien . . . . .	14
7.5	Menschlicher oder mechanischer Trainer? . . . . .	14
<b>A</b>	<b>I am an appendix!</b>	<b>16</b>
A.1	Section in appendix! . . . . .	16
<b>B</b>	<b>Another appendix? What the heck?</b>	<b>18</b>
B.1	Section in appendix! . . . . .	18

# Projektvision

## 1.1 Motivation

Sind Sie ein (angehender) Softwareentwickler und programmieren aktuell ein Computerspiel, welches lernfähige Verfahren unterstützen soll? Benötigen Sie innerhalb einer beliebigen Anwendung einen lernfähigen Algorithmus und sie kennen die Schwächen, Stärken, Grenzen und Anwendungsgebiete der Lernverfahren nicht?

Haben Sie sich auch schon mal eine der nachfolgenden Fragen gestellt oder interessieren Sie diese Fragen generell?

Wie lernt ein Programm Strategien? Was sind die elementaren Schritte die ein Programm während des Lernprozesses durchläuft? Wie anwendbar und leistungsfähig sind die Lernverfahren hinsichtlich verschiedener Spielgrundlagen? In wie fern wird ein Lernverfahren von einem Computerspiel angereizt? Wenn zwei unterschiedliche Lernverfahren untersucht und verglichen werden, welches Lernverfahren ist dann effizienter, schneller oder besser?

Diese wissenschaftliche Arbeit könnte dann sehr interessant für Sie sein. Innerhalb dieser Arbeit werden bestimmte Lernverfahren, am Beispiel verschiedener Computerspiele, auf Ihre Funktionsweise, Schwächen, Stärken und Grenzen untersucht, implementiert, und getestet.

## 1.2 Vorläufige Zielsetzung

Das Ziel der Arbeit ist die Untersuchung des Lernverhaltens, der Grenzen, der Schwächen und der Stärken verschiedener Lernverfahren am Beispiel von mindestens zwei eigens implementierten Computerspielen. Die Lernverfahren sollen trainiert werden und dadurch mehr oder weniger eigenständige Siegesstrategien und Spielzugmuster entwickeln. Die Lernverfahren könnten sich gegenseitig trainieren



oder sie trainieren indem sie gegen einen Menschen spielen. Der Fokus der wissenschaftlichen Arbeit liegt hierbei auf der Untersuchung der verschiedenen Lernverfahren und nicht auf der Implementierung besonders komplexer Computerspiele, daher sollen nur sehr simple Computerspiele implementiert werden. Ein vollständiges Dame Spiel wird zum Beispiel nicht implementiert, aber eine absichtlich verkleinerte Dame Variante mit veränderten Spielregeln, für ein schnelleres Spielende, wäre durchaus möglich. Zudem wären auch ein vier mal vier Tic-Tac-Toe ein Vier Gewinnt oder ein Black Jack Computerspiel

### **1.3 Nutzen/Zweck des Projektes**

### **1.4 Zielgruppe**

# **Grundlagen**

Schreiben der Einführung des Kapitels 2 Grundlagen

## **2.1 Lernfähige Algorithmen**

### **2.1.1 Überwachtes Lernen**

### **2.1.2 Unüberwachtes Lernen**

### **2.1.3 Bestärkendes Lernen**

## **2.2 Spielentwicklung**

## **2.3 Lineare Algebra**

## **2.4 Heuristik**

# Problemanalyse und Anforderungsdefinition

In diesem Kapitel:

Grundlagen maschinelles Lernen einfügen in Kapitel 2!

Schreibe  
die Ein-  
führung  
von  
Kapitel  
3!

## 3.1 Problemanalyse

### 3.1.1 Die Problematik

Welches lernfähige Verfahren ist auf die Brettspiele anwendbar?

Welche Daten müssen die Brettspiele liefern, sodass die lernfähigen Verfahren diese Daten verwenden können?

Wie muss das Format dieser Daten angepasst werden?

Wie wird eine Spielsituation dargestellt?

Eine Schwierigkeit besteht darin herauszufinden welche lernfähigen Verfahren anhand der Daten der Brettspiele angewendet werden können. Die Daten welche von den Brettspielen erzeugt werden, sind besonders ausschlaggebend für die Auswahl der lernfähigen Verfahren. Produziert ein implementiertes Brettspiel ein Datenset bestehend aus Eigenschaften(features) und Zielwerten(target values) so könnte ein überwachtes lernfähiges Verfahren mittels diesen Datensets trainiert werden. Ist es nicht möglich sinnvolle Zielwerte aus den Ausgabedaten der Brettspiele zu extrahieren, dann kann kein überwachtes Lernverfahren eingesetzt werden.

Erarbeiten der Eigenschaften von überwachten und nicht überwachtem maschinellem Lernen In Kapitel 2 Grundlagen

### **3.1.2 Bereits existierende Softwarelösungen**

### **3.1.3 Gegenstandsbereich des Projektes abgrenzen**

### **3.1.4 Abgrenzung gegenüber der künstlichen Intelligenz**

## **3.2 Anforderungsdefinition**

Innerhalb dieses Abschnittes werden die funktionalen und nicht-funktionalen Anforderungen für die Software definiert. Die funktionalen Anforderungen sollen das Verhalten der Software festlegen, sprich welche Aktionen die Software ausführen kann. In verschiedenen Softwareprojekten können sich die funktionale Anforderungen stark voneinander unterscheiden, dahingegen sind nicht-funktionale Anforderungen in verschiedenen Softwareprojekten oftmals sehr ähnlich.

Nichtfunktionale Anforderungen beschreiben wie gut eine Software seine Leistung erbringen soll.

### **3.2.1 Funktionale Anforderungen**

Eine User Interface (UI) ist eine Benutzerschnittstelle diese definiert die Interaktion zwischen dem Benutzer der Anwendung und der Anwendung. Die Identifikator (ID) soll die nachfolgend aufgestellten Anforderungen eindeutig identifizieren.

### **3.2.2 Nicht-funktionale Anforderungen**

**Tabelle 3.1** Funktionale Anforderungen

ID	Titel	Beschreibung
1	Brettspiele	Es sollen zwei Brettspiele implementiert werden.
2	Brettspiel UI	Es soll eine Benutzersteuerung für die Brettspiele implementiert werden.
3	Gleichartigkeit der Brettspiele	Die Spielweise der Brettspiele soll hinsichtlich bestimmter Punkte gleich sein.
4	Eindeutigkeit der Brettspiele	Die Brettspiele sollen immer einen Gewinner und einen Verlierer oder ein Unentschieden hervorbringen.
5	Endlichkeit der Brettspiele	Die Brettspiele sollen immer nach einer angemessenen endlichen Anzahl von Spielzügen enden.
6	Spieleranzahl der Brettspiele	Die Brettspiele sollen genau von zwei Spielern in einer direkten Konfrontation(Eins-gegen-Eins-Situation) ausgetragen werden.
7	Lernfähige Verfahren	Es sollen zwei lernfähige Verfahren implementiert werden. Diese Verfahren sollen Strategien, wie die Brettspiele gewonnen werden können, entwickeln.
8	Training der Lernverfahren	Die lernfähigen Verfahren sollen durch einen menschlichen Spieler trainiert werden oder durch das jeweils andere Lernverfahren.
9	Wissen Speichern	Das von den Lernverfahren ermittelte Wissen(z.B. mögliche Spielzüge, Gewichtungen besserer und schlechterer Spielzüge) soll persistent gespeichert werden.

**Tabelle 3.2** Nichtfunktionale Anforderungen

ID	Titel	Beschreibung
1	Programmlogik	Die Programmlogik der Brettspiele und der lernfähigen Verfahren soll in der Programmiersprache Python geschrieben sein.
2	Testen der Programmlogik	Die Tests der Programmlogik sollen mit den, in der Standard-Bibliothek von Python enthaltenen, Modultests(Unit testing framework) durchgeführt werden.
3	Brettspielgrafik	Die Grafik der Brettspiele soll mit der Bibliothek PyGameimplementiert werden.

# Modellierung und Entwurf

In diesem Kapitel werden die funktionalen Anforderungen aus dem Abschnitt 3.2 spezifiziert. Modelle für die einzelnen funktionalen Anforderungen sollen entwickelt werden. Die Modelle veranschaulichen das geforderte funktionale Verhalten der Software. Voraussetzung für die Entwicklung der Modelle ist die Konkretisierung der funktionalen Anforderungen. Diese Konkretisierung beinhaltet die Festlegung der Bestandteile die für eine Implementierung der funktionalen Anforderung benötigt werden. Ziel des Kapitels ist es, die wichtigsten Bestandteile einer funktionalen Anwendung herauszubilden, diese Bestandteile zu definieren und zu veranschaulichen.

## 4.1 Computerspiele

In diesem Abschnitt werden die Bestandteile der Computerspiele festgelegt und veranschaulicht.

### 4.1.1 Tic Tac Toe

Das klassische Tic Tac Toe ist ein Spiel, welches mit genau zwei Spielern gespielt wird. Jeder dieser Spieler zeichnet abwechselnd entweder ein Kreuz oder einen Kreis in eine Matrix auf ein Blatt Papier. Während eines gesamten Spiels darf ein Spieler nur Kreuze zeichnen und der andere Spieler nur Kreise. Das Spielfeld ist eine drei mal drei große Matrix, also können maximal neun Symbole in diese Matrix eingetragen werden. Um die Anzahl der möglichen Spielzüge zu erhöhen wird das Spielfeld des klassischen Tic Tac Toe auf eine vier mal vier Matrix erweitert.

#### Spielregeln

Ziel des Spiels ist es vier Kreuze oder vier Kreise in einer bestimmten Position anzuordnen. Im nachfolgenden wird davon ausgegangen, dass der menschliche Spieler Kreuze verwendet und der Computergegner Kreise. Die Kreise und Kreuze sind

Spielfiguren, welche den jeweiligen Spieler repräsentieren. Der menschliche Spieler hat zusätzlich, in den Nachfolgenden Siegesszenarien, das Anrecht auf den ersten Zug. Es existieren drei unterschiedliche Anordnungen von Spielfiguren, die das Spiel beenden und einen Sieg herbeiführen. Gewinnt ein Spieler mit einer Siegesanordnung seiner Spielfiguren, dann verliert der andere Spieler dadurch automatisch.

Eine horizontale Siegesanordnung entsteht, wenn vier Spielfiguren eines Spielers in einer horizontalen Reihe, veranschaulicht in Abbildung 4.1, angeordnet sind. In jeder Reihe des Spielbretts ist ein horizontaler Sieg möglich.

X	X	X	X
	O		
		O	
			O

**Abbildung 4.1** Tic Tac Toe horizontaler Sieg

In Abbildung 4.2 gewinnt der menschliche Spieler knapp gegen den Computergegner mit einer ununterbrochenen vertikalen Reihe. Der Computergegner hätte fast eine diagonale Reihe aus Kreisen verbunden, die jedoch von dem menschlichen Spieler mit einer Spielfigur geblockt wurde. Zudem hätte der Computergegner auch fast eine vertikale Reihe ohne Unterbrechungen vervollständigt.

Die dritte und letzte Anordnungsvariante der Spielfiguren, welche zu einem Sieg eines Spielers führt, ist die diagonale Verbindung von vier Spielfiguren eines Spielers. In Abbildung 4.3 gewinnt der Computergegner mit einer diagonalen Anordnung von vier Spielfiguren ohne Unterbrechung einer gegnerischen Spielfigur.

Insgesamt existieren vier vertikale, vier horizontale und zwei diagonale Anordnungen der Spielfiguren, welche einen Sieg herbeiführen würden, also zehn verschiedene Siegesanordnungen. Was passiert jedoch, wenn keine der zehn möglichen Siegesanordnungen auftritt?

Dann gewinnt bzw. verliert keiner der beiden Spieler und es entsteht ein Unentschieden. Sind die beiden Kontrahenten gleich gut, erfahren oder verwenden die selben Strategien, dann tritt ein Unentschieden möglicherweise öfter oder andauernd ein.

O	X		
	X	O	X
	X	O	X
	X	O	O

**Abbildung 4.2** Tic Tac Toe vertikaler Sieg

X		X	O
	X	O	
	O	X	
O		X	O

**Abbildung 4.3** Tic Tac Toe diagonaler Sieg



## Benutzerschnittstellen

Der Benutzer kann seine Kreuze auf das Spielbrett setzen indem er ein vorher vom Spiel definiertes Zahlentupel über die Tastatur eingibt. Welche Zahlentupel ein Kreuz an welche Stelle setzt ist in Abbildung 4.4 definiert. Sollte der menschliche Spieler keines der erlaubten Zahlentupel eingeben, dann wird er darauf hingewiesen, welche Steuerungsmöglichkeiten zum setzen der Spielfiguren ihm zur Verfügung stehen.

00	01	02	03
10	11	12	13
20	21	22	23
30	31	32	33

Abbildung 4.4 Tic Tac Toe Spielfiguren setzen

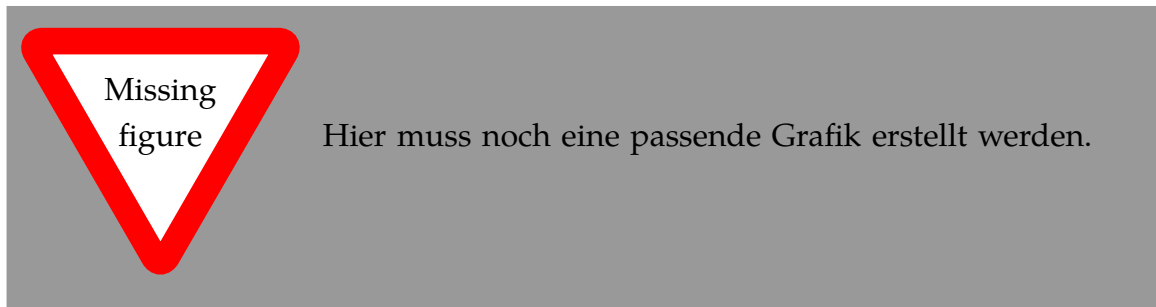
### 4.1.2 Reversi

#### Spielprinzipien

#### Spielregeln

#### Benutzerschnittstellen

## **4.2 Lernverfahren**



### **4.2.1 Analyse und Auswahl der lernfähigen Algorithmen**

### **4.2.2 Anwendung der Algorithmen auf Computerspiele**

### **4.2.3 Konzeptuelles Training der Algorithmen**

### **4.2.4 Persistenz der Trainingsdaten**

# Implementierung

In diesem Kapitel: //TODO Einführung in das Kapitel

## 5.1 Computerspiele

## 5.2 Lernverfahren

## 5.3 Alternative Lernverfahren

# Validierung

In diesem Kapitel: //TODO Einführung in das Kapitel

## 6.1 Messbare Testkriterien

## 6.2 Modultests

### 6.2.1 TicTacToe

### 6.2.2 Reversi

### 6.2.3 Lernverfahren 1

Empirisches Protokoll

### 6.2.4 Lernverfahren 2

Empirisches Protokoll

### 6.2.5 Persistenz

## 6.3 Systemtest

# Auswertung

In diesem Kapitel: //TODO Einführung in das Kapitel

## 7.1 Belastbarkeit und Grenzen der Lernverfahren

## 7.2 Optimale Anwendungsspiele für die Lernverfahren

## 7.3 Gegenüberstellung der Lernverfahren

## 7.4 Bewertung der Strategien

## 7.5 Menschlicher oder mechanischer Trainer?

# Literatur

- [Alp08] Ethem Alpaydm. *Maschinelles Lernen*. 1. Aufl. Oldenbourg, 2008.
- [Bei14] Christoph Beierle. *Methoden wissensbasierter Systeme: Grundlagen, Algorithmen, Anwendungen*. 5. Aufl. Springer, 2014.
- [Ert16] Wolfgang Ertel. *Grundkurs Künstliche Intelligenz: Eine praxmatische Einführung*. 4. Aufl. Springer, 2016.
- [Har12] Peter Harrington. *Machine Learning: IN ACTION*. 1. Aufl. Manning, 2012.
- [Lö93] Jan Löschner. *Künstliche Intelligenz: Ein Handwörterbuch für Ingenieure*. 1. Aufl. VDI, 1993.
- [Ras16] Sebastian Raschka. *Machine Learning mit Python*. 1. Aufl. MIT Press, 2016.
- [Rus12] Stuart J. Russell. *Künstliche Intelligenz: Ein moderner Ansatz*. 3. Aufl. Pearson, 2012.

# I am an appendix!

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradisiacal country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas, wild Question Marks and devious Semikoli, but the Little Blind Text didn't listen. She packed her seven versalia, put her initial into the belt and made herself on the way. When she reached the first hills of the Italic Mountains, she had a last view back on the skyline of her hometown Bookmarksgrove, the headline of Alphabet Village and the subline of her own road, the Line Lane. Pityful a rethoric question ran over her cheek, then

## A.1 Section in appendix!

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradisiacal country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas, wild Question Marks and devious Semikoli, but the Little Blind Text didn't listen. She packed her seven versalia, put her initial into the belt and made herself on the way. When she reached the first hills of the Italic Mountains, she had a last view back on the skyline of her

*Anhang A:* I am an appendix!

hometown Bookmarksgrove, the headline of Alphabet Village and the subline of her own road, the Line Lane. Pityful a rethoric question ran over her cheek, then



# Another appendix? What the heck?

## B.1 Section in appendix!

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regalia. It is a paradisiacal country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life. One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas, wild Question Marks and devious Semikoli, but the Little Blind Text didn't listen. She packed her seven versalia, put her initial into the belt and made herself on the way. When she reached the first hills of the Italic Mountains, she had a last view back on the skyline of her hometown Bookmarksgrove, the headline of Alphabet Village and the subline of her own road, the Line Lane. Pityful a rethoric question ran over her cheek, then



## **Acknowledgements**

Acknowledgements go to the back!