

DIPLOMARBEIT

JAVACHESS, CHESSPI ANDCHESS

Höhere Technische Bundeslehr- und Versuchsanstalt Anichstrasse

Abteilung

Elektronik & Technische Informatik

Ausgefuehrt im Schuljahr 2017/18 von: Betreuer/Betreuerin:

Alexander Beiser 5CHEL
Marcel Huber 5CHEL

Ing. MSc. Signitzer Mar-
kus

Innsbruck, am 30.10.2017

Abgabevermerk:

Betreuer/in:

Datum:



Inhaltsverzeichnis

Erklärung der Eigenständigkeit der Arbeit	4
1 Zusammenfassung des Projektergebnisses	4
1.1 Kurzfassung /Abstract	4
1.1.1 Alexander Beiser	4
1.1.2 Marcel Huber	5
1.2 Projektergebnis	6
2 Lizenz:	8
1 EINLEITUNG	9
2 VERTIEFENDE AUFGABENSTELLUNG	9
2.1 Alexander Beiser	9
2.2 Marcel Huber	9
3 Schach, eine Erklärung	11
3.1 ist Schach?	11
3.2 Spielregeln:	11
3.2.1 Zugregel Bauer:	12
3.2.2 Zugregel Springer:	12
3.2.3 Zugregel Läufer:	12
3.2.4 Zugregel Turm:	13
3.2.5 Zugregel Dame:	13
3.2.6 Zugregel König:	13
3.3 Schachmaschinen:	13
4 Java Chess	14
4.1 Einführung:	14
4.2 Java Chess - Übersicht	14
4.2.1 Blockschaltbild	15
4.2.2 Initialisierung	15
4.3 Repräsentation der Figuren:	16
4.4 Zugmechanik und Local-Mode	16
4.5 LAN-Mode	17
4.6 AI-Mode	17
4.6.1 Prinzipielle Möglichkeiten einer AI	17
4.6.2 Verwendete Schach-AI	18
4.6.3 Code	18
5 FERTIGUNGSDOKUMENTATION	19
6 BENUTZERDOKUMENTATION	19
6.1 Installationsanleitung	19
6.2 Anwendungsbeispiele	19
6.3 Referenzhandbuch	19
6.4 Fehlermeldungen und Hinweise auf Fehlerursachen	19



I	ABBILDUNGSVERZEICHNIS	20
II	TABELLENVERZEICHNIS	20
III	LITERATURVERZEICHNIS	21
6	PFLICHTENHEFT	23
6 .1	Funktionale Anforderungen	23
6 .2	Schnittstellen	23
6 .3	Abnahmekriterien	23
6 .4	Dokumentationsanforderungen	23
6 .5	Qualitätsstandards	24
6 .6	Abwicklungsprozess	24
7	ZUSAMMENFASSUNG	26
7 .1	Schlussfolgerung / Projekterfahrung	26
7 .2	Projektterminplanung	26
7 .3	Projektpersonalplanung und Kostenplanung	26
7 .3.1	Projektkostenplan	26
7 .3.2	Arbeitsnachweis Diplomarbeit	26
	Tabelle 1: Arbeitsaufstellung	27
7 .3.3	Leistungscontrolling	27
	<i>Abzugeben sind:</i>	28



ERKLAERUNG DER EIGENSTAENDIGKEIT DER ARBEIT

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wortlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Ort, Datum

Verfasser, Verfasserinnen
Vor- und Zunamen

1 Zusammenfassung des Projektergebnisses

1.1 Kurzfassung /Abstract

Alexander Beiser und Marcel Huber entwickelten im Zuge ihrer Diplomarbeit 2017/18 ein Schachspiel, welches auf einem PC, Android-Smartphone und Raspberry PI spielbar ist. Das Schachspiel basierte auf einem bereits von ihnen geschriebenen rohen „Gerüst“.

Dieses Spiel wurde mit der Programmiersprache Java entwickelt, weiteres war für den Raspberry PI ein Gehäuse zu Designen und mit einem 3D Drucker zu realisieren. Um das Spielvergnügen für den Raspberry PI auch Unterwegs zu ermöglichen wurde eine Akkusteuerung entworfen und realisiert.

1.1.1 Alexander Beiser

Alexander Beiser war für große Teile des Backends zuständig. Ein Hauptteil bestand aus der Entwicklung einer Chess Engine, also eines Zugmechanismus welcher



speziell für die ebenso von Alexander Beiser erschaffene Künstliche Intelligenz performiert wurde. Er entwickelte auch die Akkusteuering für den Raspberry PI und designte das Gehäuse.

1.1.2 Marcel Huber

Marcel Huber war für weite Teile des Frontend Bereichs zuständig, die anderen Schwerpunkte bestanden in der Entwicklung des Netzwerkalgorithmus und der Entwicklung des Schachspiels für Android Geräte.



1.2 Projektergebnis

Allgemeine Beschreibung, was vom Projektziel umgesetzt wurde, in einigen kurzen Sätzen. Optional Hinweise auf Erweiterungen. Gut machen sich in diesem Kapitel auch Bilder vom Gerät (HW) bzw. Screenshots (SW).

Liste aller im Pflichtenheft aufgeführten Anforderungen, die nur teilweise oder gar nicht umgesetzt wurden (mit Begründungen).



Inhaltsverzeichnis



2 Lizenz:

Das Schachprogramm wird unter der „Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Public License“ entwickelt.

Dies räumt jeden Menschen folgende Rechte ein:

- **Teilen:** Das Programm darf frei kopiert und weiterverteilt werden.
- **Verändern:** Das Programm darf frei verändert werden. Somit dürfen natürlich Verbesserungen implementiert werden.

Allerdings muss dies unter den folgenden Bedingungen geschehen:

- **Zuschreibung:** Man muss die Namen der Entwickler entsprechend anführen und angeben, ob Veränderungen gemacht wurden.
- **Nicht kommerziell:** Das Programm darf nicht kommerziell benützt werden.
- **Gleiche Lizenz:** Sobald Veränderungen gemacht wurden, muss die original Lizenz weiter verwendet werden. Auch darf nicht von der obigen genannten Lizenz abgewichen werden.
- **Gesetzeskonform:** Das Programm darf nicht so verändert werden, dass die Nutzung illegal wird.



1 EINLEITUNG

Alexander und Marcel sind beide begeisterte Schachspieler, womit die Entwicklung eines Schachspiels nahe liegt. Gegen Ende des 4. Jahres der HTL trafen sie die Entscheidung ein Schachspiel selber zu entwickeln und keine Diplomarbeit von einer Firma anzunehmen. Für die Entwicklung ihres Schachspiels, werden sie von „Elektrotechnik Beiser“ unterstützt. Diese Firma übernimmt etwaige anfallende Kosten für die Hardwarekomponenten.

Anfang der 5. Klasse der HTL kamen die Sondierungsgespräche mit ihrem Betreuer Ing. MSc. Signitzer Markus, welcher Ihnen sagte was im Zuge dieser Diplomarbeit alles erledigt werden müsse.

Durch die Gespräche kam man zum Schluss, dass für die Diplomarbeit ein Schachspiel in Java entwickelt werden muss und dieses auf einen RaspberryPI, als auch auf Android Geräte portiert werden soll. Weiteres wird eine Akkusteuerung für den RaspberryPI entwickelt und ein Gehäuse designed und mittels Schuleigenen 3D-Drucker ausgedruckt.

Die GUI des Spiels soll mit JavaFX erstellt werden. Das Spiel soll gegen eine selbstentwickelte Künstliche Intelligenz spielbar sein, im Hot Seat Modus oder im Local Area Network.

Im Hot Seat Modus spielt man auf einem PC abwechselnd die Partien. Details werden in einem Pflichtenheft festgehalten, dieses Pflichtenheft befindet sich im Anhang.

In der Einleitung wird erklärt, wieso man sich für dieses Thema entschieden hat. (Zielsetzung und Aufgabenstellung des Gesamtprojekts, fachliches und wirtschaftliches Umfeld)

2 VERTIEFENDE AUFGABENSTELLUNG

2.1 Alexander Beiser

Überarbeitung des Schachmattalgorithmus, Entwicklung der Zugmechanik und Entwicklung einer künstlichen Intelligenz.

Implementierung des Schachspiels auf den Raspberry-PI, gleichzeitige Designen des Gehäuses für den Raspberry-PI und Entwicklung der Akkuansteuerungsschaltung.

2.2 Marcel Huber

Entwicklung der Netzwerkfähigkeit und Implementierung von Java FX. Verbesserung und Weiterentwicklung der audio- und visuellen Gestaltung. Entwicklung der



Android-App und Fertigung des Raspberry PI-Gehäuses



3 Schach, eine Erklärung

3.1 Was ist Schach?

Um den Aufbau des Programmes nachzuvollziehen zu können, sollten die Grundregeln des Schachspiels geläufig sein. Hier haben wir versucht, die wichtigsten kurz zusammenzufassen.

Was ist Schach?

Schach ist ein strategisches Brettspiel, indem es darum geht die feindliche Seite zuschlagen. Die feindliche Seite hat asbald verloren, wann der König gefallen ist.

Der Name Schach kommt aus dem persischen „Schah“ und bedeutet so viel wie König, woher der Name „königliches Spiel“ stammt.

Ursprünglich wurde das Spiel vermutlich in Nordindien erfunden und kam im Zuge der islamischen Expansion, von 630 bis ca. 750, nach Europa.

3.2 Spielregeln:

Nach der 1. Erklärung was Schach ist, kommen wir zu den Spielregeln. Schach wird aufm einem 8*8 karierten Feld gespielt. Die Nummerrierung erfolgt horizontal durch das Alphabet, a bis h und vertikal durch Ziffern, 1 bis 8. Zu Beginn gibt es zwei Teams, meist Weiß und Schwarz, mit jeweils 16 Figuren. Folgende Figuren sind zu Beginn am Feld:

- 8 Bauern
- 2 Springern
- 2 Läufern
- 2 Türmen
- 1 Dame
- 1 König

Das Ende des Spiels erfolgt entweder durch Schachmatt, Aufgabe oder durch ein Remis/Patt. Schachmatt bedeutet, dass der König bedroht wird und es dem Spieler nicht mehr möglich ist den König aus dieser Position zu befreien. Patt Möglichkeiten:

- entsteht wenn eine der Parteien keinen legalen Zug mehr ausführen kann
- Durch ein „technisches Remis“, wenn außer den beiden Königen nur mehr ein Läufer oder Springer am Feld ist.



- Wenn 50 Züge lang keine Spielfigur geschlagen oder ein Bauer bewegt wurde und der am Zug befindliche Spieler das Remis verkündet.
- Wenn eine identische Stellung drei mal mit identischen Zugmöglichkeiten mindestens drei mal vorkommt, kann ein Spieler ein Remis beantragen.
- Wenn 75 Züge lang keine Figur geschlagen bzw. ein Bauer gezogen wurde oder fünfmal diesselbe Stellung mit gleichen Zugmöglichkeiten entstanden ist. Hierbei wird das Remis automatisch vom Schiedsrichter/Schachspiel eingeleitet.

Nun folgen die Zugregeln:

3.2.1 Zugregel Bauer:

- Bauer darf einen Schritt nach vorne ziehen, wenn das Feld leer ist
- Befindet sich der Bauer in der Ausgangsposition und wurde noch nicht gezogen, kann er auch wahlweise zwei Schritte vorrücken.
- Der Bauer schlägt vorwärts diagonal ein Feld.
- Spezialzug: „En Passant“. Dies kann er als einzige Spielfigur, wenn ein feindlicher Bauer zuvor einen Doppelschritt gemacht hat und somit den eigenen Bauern die Option nimmt, den gegnerischen Bauern anzugreifen. Falls er ausgeführt wird, ist der feindliche Bauer vernichtet und der eigene rückt diagonal ein Feld hinter den nicht mehr existierenden Bauern.
- Sobald ein Bauer die gegnerische „Grundreihe“ erreicht, wird ein Bauerntausch durchgeführt. Hier muss der Bauer gegen Dame, Turm, Läufer oder einen Springer eingetauscht werden.

3.2.2 Zugregel Springer:

- Der Springer darf auf das Feld ziehen, dass zwei Felder horizontal bzw. diagonal und eines diagonal bzw. horizontal (gegengleich) versetzt ist. z.B.: Von b8 auf c6

3.2.3 Zugregel Läufer:

- Läufer dürfen diagonal, so weit wie sie wollen ziehen und schlagen. Jedoch darf er nicht über eine Figur ziehen.



3.2.4 Zugregel Turm:

- Ein Turm darf horizontal bzw. vertikal ziehen und schlagen wie weit er will, jedoch nicht über Figuren hinweg.

3.2.5 Zugregel Dame:

- Eine Dame darf horizontal, vertikal bzw. diagonal ziehen und schlagen so weit wie sie will, jedoch nicht über Figuren hinweg.

3.2.6 Zugregel König:

- Der König kann horizontal, vertikal bzw. diagonal ein Feld ziehen.
- Spezialzug: „Rochade“. Dabei wird der König entweder zwei Felder nach links, bzw. zwei Felder nach rechts bewegt. Der Turm bewegt sich dabei drei Felder nach rechts bzw. zwei Felder nach links. König und Turm dürfen bis zu diesem Zug noch nicht bewegt worden sein, weiters darf keines der Felder über das sie ziehen, der König oder der Turm bedroht werden.

3.3 Schachmaschinen:

Seit dem es die Möglichkeit gibt einen Schachspielenden Mechanismus zu bauen, hat man dies auch getan. Zu Anfang war dies noch der „schachspielende Türke“, welcher 1769 von Wolfgang von Kempelen konstruiert wurde.

Der richtige Durchbruch geschah aber erst durch die Erfindung des Computers. Die Hardware wurde immer leistungsfähiger, wodurch der Mensch als Gegner immer weiter in Bedrängung geriet. 1997 schlug der von IBM speziell entwickelte Schachcomputer Deep Blue, den damaligen Schachweltmeister Kasparow, wodurch die Künstliche Intelligenz in diesem Bereich offiziell den Menschen überholt hat.

Heutzutage wird gegen Schachcomputer vor allem zu Trainingszwecken gespielt. Solche Schachcomputer finden sich mittlerweile auf so ziemlich jedem Gerät, egal ob Smartphone, Tablet oder PC/Laptop. Meist sind diese Programme aber proprietär und „closed source“. Wir entwickeln deshalb ein „open Source“ Schachspiel, dass auf mehreren Devices spielbar ist.



4 Java Chess

4.1 Einführung:

Bevor mit der Dokumentation des Programmcodes begonnen werden kann, werden zuerst einige Möglichkeiten beschrieben, wie ein Schachprogramm prinzipiell programmiert werden kann.

Hierfür gibt es prinzipiell zwei Möglichkeiten:

1. Die Figuren kennen ihre Position
2. Das Brett kennt die Positionen der Figuren

Das die Figuren ihre Position kennen, klingt zuerst gar nicht so abwegig. Probleme treten aber auf, sobald Schachmatt überprüft werden soll. Hierfür muss überprüft werden ob irgendeine gegnerische Figur den König schlagen kann, wofür man aber das Objekt der Figur benötigt. Natürlich ist dies Programmiertechnisch kein Problem, dadurch entstehen aber längere Wartezeiten.

Falls das Brett die Position der Figuren kennt und diese Figuren lediglich über eine Zahlenmatrix dargestellt werden, ist das Spiel nicht nur sehr viel performanter, es ergeben sich auch große Vorteile beim entwickeln der Künstlichen Intelligenz. Wir entschieden uns für diese Lösung.

4.2 Java Chess - Übersicht

JavaChess ist in der Programmiersprache Java geschrieben. Java ist eine Objekt-orientierte, Klassenbasierte hochsprache der Informatik. Java hat den Vorteil, dass es nicht Hardware gebunden ist und somit ein Programm, geschrieben auf einer Distribution des Betriebssystems GNU/Linux auf (zumindest theoretisch) allen unterstützten Systemen läuft.

Somit können wir unser Spiel auch auf einem Raspberry-PI lauffähig machen.

Das von uns verwendete GUI Environment ist JavaFX. Es wurde erstmals im Dezember 2008 den Programmierern zugänglich gemacht und soll das bis dahin Standard Java GUI Environment „Swing“ ersetzen. Die Unterschiede bestehen im Aufbau, wie eine GUI realisiert werden kann bis hin zu den verbesserten grafischen Effekten, die durch JavaFX möglich sind.

Dadurch wurden die Entscheidungen gefällt Java mit JavaFX zu verwenden. Java Chess nützt in Folge einige dieser Vorteile aus, vorallem Objektorientiertes design.