CHEVA: Erschaffung eines Schachbewertungs-Programms

Zweite Bachelorarbeit

Ausgeführt zum Zweck der Erlangung des akademischen Grades   
**Bachelor of Science in Engineering**

am Bachelorstudiengang Medientechnik  
an der Fachhochschule St. Pölten

von:  
**Stefan Dürr**MT191087

Betreuer/in: Markus Seidl

Wien, 01.03.22

Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.

- ich dieses Thema bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der vom Begutachter bzw. der Begutachterin beurteilten Arbeit überein.

.................................................. ................................................

Ort, Datum Unterschrift

Kurzfassung

In der Arbeit geht es um die Erschaffung von CHEVA, ein Programm, das Schachpositionen analysiert und anschließend bewertet. Der Leser wird mit einer Einführung an die Thematik herangeführt. Anschließend wird im Praxisteil der Coding-Prozess erklärt und das Programm anhand berühmter Schachstellungen auf Herz und Nieren geprüft.

This paper is about the creation of CHEVA, a program that analyses and evaluates chess positions. In the first part the reader is introduced to the topic. In the second part, the coding process is being explained and CHEVA gets its first use: It is going to analyse chess positions of famous matches, some of them changed the way humans play chess forever.

Inhaltsverzeichnis

Ehrenwörtliche Erklärung II

Kurzfassung III

Inhaltsverzeichnis IV

1. Einleitung 5

2. Hintergründe und Anforderungen 6

2.1 Schach: Eine kurze „Eröffnung“ 6

2.2 Wie wird eine Schachposition bewertet? 6

2.3 Gegenwärtige Schachprogramme 6

2.4 Welche Anforderungen muss CHEVA erfüllen? 6

3. Herstellung des Prototyps 7

3.1 Verwendete Tools 7

3.2 Der Codingprozess 7

3.3 Hindernisse 7

3.4 Parameteranpassung 7

4. Evaluation des Prototyps 8

4.1 Vorstellung und Bewertung der Positionen 8

4.1.1 Erste Stellungsbewertung: „Mikhail Tal’s Immortal Game (1960)“ 8

4.1.2 Zweite Stellungsbewertung: „World Chess Championship: Magnus Carlsen vs. Ian Nepomniatchtchi, Game 9 (2022)“ 8

4.1.3 Dritte Stellungsbewertung: „Nezhmetdinov vs Chernikov (1962)“ 8

4.2 Analyse 8

4.2.1 Analyse der Präzision 8

4.2.2 Analyse der Laufzeit 8

4.2.3 Gegenüberstellung 9

4.2.4 Persönliche Bewertung der Stellungen 9

5. Fazit / Diskussion 10

5.1 Gewonnene Erkenntnisse 10

5.2 Bewertung der Arbeit aus Sicht des Autors 10

Literaturverzeichnis 11

Abbildungsverzeichnis 12

Tabellenverzeichnis 13

Listingverzeichnis 14

Anhang 15

1. Einleitung

In dieser Bachelorarbeit geht es um die Erschaffung von „CHEVA“, einem Programm, das Schachpositionen bewertet und analysiert. Das Ziel dieser Publikation ist, in zwei Teilen die folgenden zwei Forschungsfragen ausführlich zu beantworten. Im ersten Teil wird auf die erste Forschungsfrage, die für grundlegendes Verständnis der Thematik unabdingbar ist, detailliert eingegangen:

**Wie wird eine Schachposition bewertet?**

Der erste Teil soll dem Leser einen groben Überblick über die Materie verschaffen. Als Beispiel werden hier unter anderem die Praktiken des momentan spielstärksten Schachprogramms („Stockfish“) verwendet. Weiters wird konkretisiert, welche Anforderungen „CHEVA“ schlussendlich erfüllen muss.

Im zweiten Teil, dem Praxisteil, wird das Programm mittels der Programmiersprache „Python“ erstellt. Der Codingprozess wird kurz dokumentiert. Nach Abschluss der Entwicklung erhält „CHEVA“ zu Testzwecken interessante, berühmte Schachstellungen als Input und muss sie möglichst präzise bewerten.

Anhand der Ergebnisse kann die zweite Forschungsfrage beantwortet werden: **Wie gut kann ein selbst erstelltes Programm, das Schachstellungen importieren, analysieren und bewerten kann, mit State-of-the-art-Schachprogrammen mithalten?**

„Easy to learn, hard to master“: Auf kaum einen Anwendungsbereich ist dieser Satz wohl so zutreffend wie auf das jahrtausendalte Spiel des Schachs. Ein Großteil der Motivation, die der Themenauswahl zugrunde liegt, ist meine starke Leidenschaft für ebendieses Brettspiel. Auf den ersten Blick unersichtlich, weist Schach eine enorme Komplexität auf, die selbst für heutige Computer noch nicht vollständig zu beherrschen ist.

In meinem Studium konnte ich mir Programmierkenntnisse in einigen „schachaffinen“ Sprachen aneignen (Python, Java, etc.), die meine Entscheidung, meine Bachelorarbeit über dieses Thema zu schreiben, endgültig besiegelt haben. Aufgrund der programmierlastigen Themenstellung ist die Relevanz für den Studiengang „BMT“ klar ersichtlich.

1. Hintergründe und Anforderungen
   1. Schach: Eine kurze „Eröffnung“

Hier folgt eine kurze Erklärung des Spiels.

* 1. Wie wird eine Schachposition bewertet?

Eine Beispielposition bekommt eine Bewertung von „Stockfish“. Die Bewertung wird anschließend anhand Kriterien verständlich gemacht.

* 1. Gegenwärtige Schachprogramme

Welche Schachprogramme gibt es? Worin unterscheiden sie sich?

* 1. Welche Anforderungen muss CHEVA erfüllen?

…um mit anderen Programmen konkurrieren zu können?

1. Herstellung des Prototyps
   1. Verwendete Tools

Eine Auflistung verwendeter Tools.

<https://www.chessprogramming.org/Evaluation#Where_to_Start>

<https://stackoverflow.com/questions/17379849/simple-minimax-evaluation-function-for-chess-position>

<https://chess.stackexchange.com/questions/35621/quick-method-for-generating-fen-strings-from-pgn-using-python-chess>

<https://stackoverflow.com/questions/44445051/creating-a-2d-numpy-array-to-hold-characters>

<https://www.chessprogramming.org/Simplified_Evaluation_Function>

<https://www.geeksforgeeks.org/extract-data-from-pgn-files-using-the-chess-library-in-python/?ref=rp>

<https://www.youtube.com/watch?v=1_u5VYBm3D4>

<https://jupyter.brynmawr.edu/services/public/dblank/CS371%20Cognitive%20Science/2016-Fall/Programming%20a%20Chess%20Player.ipynb>

* 1. Der Codingprozess

Eine Step-by-step-Dokumentation des Codingprozesses.

* 1. Hindernisse

Etwaige Hindernisse, die während der Praxisarbeit auftraten.

* 1. Parameteranpassung

Feintuning der Parameter, mit anschließender Protokollierung. Wurde die Präzision erhöht?

1. Evaluation des Prototyps
   1. Vorstellung und Bewertung der Positionen
      1. Erste Stellungsbewertung: „Mikhail Tal’s Immortal Game (1960)“

CHEVA bekommt mit dieser Stellung, die als „Klassiker“ gilt, einen ersten Input.

* + 1. Zweite Stellungsbewertung: „World Chess Championship: Magnus Carlsen vs. Ian Nepomniatchtchi, Game 9 (2022)“

Eine sehr interessante, turnierentscheidende Stellung ist CHEVAs zweiter Input.

* + 1. Dritte Stellungsbewertung: „Nezhmetdinov vs Chernikov (1962)“

Ein unvergessliches Damenopfer ist der dritten Stellung unmittelbar voraus.

* 1. Analyse
     1. Analyse der Präzision

Wurde die Position korrekt analysiert?

Stellung 1

Stellung 2

Stellung 3

* + 1. Analyse der Laufzeit

Bei einer derart umfangreichen Bewertung spielt die Laufzeit eine entscheidende Rolle. Wie gut ist CHEVAs Performance? Kann sie durch Änderungen im Code noch verbessert werden?

* + 1. Gegenüberstellung

Wie nah konnte CHEVA an Stockfish heranreichen? Welche Unterschiede in Bewertung und Laufzeit gibt es?

* + 1. Persönliche Bewertung der Stellungen

Hier kommt ein drittes Assessment des schachaffinen Autors hinzu.

1. Fazit / Diskussion
   1. Gewonnene Erkenntnisse

Zusammenfassung der Beantwortung der zwei Forschungsfragen:

Wie wird eine Schachposition bewertet?

Wie gut kann ein selbst erstelltes Programm, das Schachstellungen importiert, analysiert und bewertet, mit State-of-the-art-Schachprogrammen mithalten?

Sonstige Erkenntnisse werden ebenfalls hier beleuchtet. Diese Sektion könnte in naher Zukunft noch ausgebaut werden.

* 1. Bewertung der Arbeit aus Sicht des Autors

Persönliche Meinung

Literaturverzeichnis

Lai, M. (2015). *Giraffe: Using Deep Reinforcement Learning to Play Chess*. https://arxiv.org/abs/1509.01549v2

Tesauro, G. (2001). Comparison training of chess evaluation functions. In *Machines that learn to play games* (pp. 117-130).

Kendall, G., & Whitwell, G. (2001, May). An evolutionary approach for the tuning of a chess evaluation function using population dynamics. In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546)* (Vol. 2, pp. 995-1002). IEEE.

Thrun, S. (1994). Learning to play the game of chess. *Advances in neural information processing systems*, *7*.

Mehr Literatur folgt.

Abbildungsverzeichnis

Tabellenverzeichnis

Listingverzeichnis

Anhang