

BACHELORARBEIT

Entwicklung einer flexiblen Schach-Engine zur Unterstützung nicht-standardisierter Brettkonfigurationen mit Untersuchung der Nutzerakzeptanz

vorgelegt am 3. August 2026
Lennard Burrmann
Matrikel-Nr.: XXXXXXXXXX
Studiengang Media Systems (B.Sc.)

Erstprüferin: Prof. Dr. Larissa Putzar
Zweitprüfer: Sune Maute

**HOCHSCHULE FÜR ANGEWANDTE
WISSENSCHAFTEN HAMBURG**
Fakultät Informatik und Digitale Gesellschaft
Finkenau 35
22081 Hamburg

Zusammenfassung

Diese Arbeit untersucht, wie sich grundlegende Verfahren von Schachengines auf nicht standardisierte Brettgeometrien übertragen lassen und wie sich das Spielen auf polymorphen Brettern hinsichtlich subjektiver Belastung, Spielverhalten und Spielerleben vom Spiel auf einem Standardbrett unterscheidet. Hierzu wurde mit *Uncy* eine in C#/.NET implementierte Engine entwickelt, die unterschiedliche Brettgrößen und inaktive Felder zur Laufzeit verarbeitet. Sie kombiniert eine flexible Brettrepräsentation mit geometrieabhängiger Zuggenerierung, etablierten Suchverfahren und einer adaptiven Bewertungsfunktion. Ein Vergleich der Zuggenerierung mit der etablierten Variantenengine *Fairy-Stockfish* ergab für exemplarische Brettkonfigurationen übereinstimmende Knotenzahlen. Laufzeitmessungen zeigten außerdem, dass die Suchzeiten für den Einsatz als interaktiver Spielgegner ausreichend kurz waren. Eine Webanwendung diente zugleich als Spieloberfläche und Erhebungsinstrument. In einer Remote-Nutzerstudie im Within-Subject-Design spielten 23 Personen zunächst auf einem Standardbrett und anschließend auf polymorphen Brettern. Dabei war die subjektiv empfundene Belastung in der polymorphen Phase signifikant höher. Entgegen der Erwartung trafen die Teilnehmenden ihre Entscheidungen jedoch schneller; eine statistisch belastbare Verschlechterung der Zugqualität zeigte sich nicht. Der durchschnittliche Spielspaß blieb nahezu unverändert, während die individuellen Bewertungen stärker auseinanderfielen. Die Ergebnisse zeigen, dass flexible Schachengines für polymorphe Brettgeometrien technisch realisierbar sind und solche Geometrien vor allem das subjektive Spielerleben verändern. Aufgrund der begrenzten Stichprobe und der festen Versuchsreihenfolge sind die empirischen Befunde als erste Hinweise zu verstehen.

Abstract

This thesis investigates how fundamental chess-engine methods can be transferred to non-standard board geometries and how playing on polymorphic boards differs from playing on a standard board in terms of subjective workload, playing behavior, and player experience. To this end, *Uncy* was developed, a C#/.NET chess engine capable of processing different board sizes and inactive squares at runtime. It combines a flexible board representation with geometry-dependent move generation, established search methods, and an adaptive evaluation function. A comparison of its move generation with the established chess-variant engine *Fairy-Stockfish* produced identical node counts for representative board configurations. Runtime measurements further showed that search times were sufficiently short for the engine to serve as an interactive opponent. A web application functioned as both the playing interface and the study instrument. In a remote user study with a within-subject design, 23 participants first played on a standard board and then on polymorphic boards. Subjective

workload was significantly higher during the polymorphic phase. Contrary to expectations, participants made their decisions faster, while no statistically reliable deterioration in move quality was observed. Average enjoyment remained almost unchanged, although individual ratings became more polarized. The findings demonstrate that flexible chess engines for polymorphic board geometries are technically feasible and that such geometries primarily alter the subjective experience of play. Given the limited sample and fixed order of conditions, the empirical findings should be regarded as preliminary evidence.

Inhaltsverzeichnis

Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
Hilfsmittelverzeichnis	VI
1. Einleitung	1
1.1. Motivation	1
1.2. Problemstellung	1
1.3. Zielsetzung der Arbeit	2
1.3.1. Entwicklung der Engine	2
1.3.2. Entwicklung einer Nutzerschnittstelle	2
1.3.3. Nutzerstudie zu polymorphen Brettgeometrien	3
1.4. Aufbau der Arbeit	3
2. Theoretische Grundlagen	4
2.1. Grundlagen von Schach-Engines	4
2.1.1. Spielzustände, Suchbaum und Bewertungsfunktionen	4
2.1.2. Alpha-Beta-Suche und Zugsortierung	5
2.1.3. Iterative Tiefensuche	5
2.1.4. Brettrepräsentationen	6
2.1.5. Hashing und Transpositionstabellen	6
2.2. Schachvarianten und flexible Brettrepräsentationen	7
2.2.1. Begriffliche Abgrenzung	7
2.2.2. Andere spezialisierte Spielsysteme	8
2.3. Grundlagen der Nutzerstudie	8
2.3.1. Mustererkennung im Schachspiel	8
2.3.2. Subjektive Beanspruchung und Entscheidungsverhalten	9
2.4. Zusammenfassung und Bezug zur Forschungsfrage	9
3. Konzeption	10
3.1. Anforderungsanalyse	10
3.1.1. Funktionale Anforderungen	10

3.1.2.	Nicht-funktionale Anforderungen	11
3.2.	Systemarchitektur	12
4.	Implementierung	14
4.1.	Technischer Gesamtaufbau	14
4.2.	Polymorphe Zustandsrepräsentation	15
4.2.1.	Brett- und Feldkodierung	15
4.2.2.	Erweiterte FEN-Verarbeitung	15
4.2.3.	Dimensionsabhängige Lookup Tabellen	16
4.3.	Zuggenerierung auf polymorphen Geometrien	17
4.3.1.	Repräsentation und Legalitätsprüfung von Zügen	17
4.3.2.	Figurenspezifische Behandlung der Geometrie	17
4.3.3.	Validierung der Zuggenerierung	18
4.4.	Stellungsgedächtnis, Suche- und Bewertungsfunktionen	19
4.4.1.	Zobrist-Hashing und Transpositionstabelle	19
4.4.2.	Iterative Suche und Baumreduktion	20
4.4.3.	Zugsortierung	21
4.4.4.	Geometrieabhängige Stellungsbewertung	21
4.4.5.	Suchleistung auf den Studienbrettern	22
4.5.	Bereitstellung als Web- und Studienanwendung	22
4.5.1.	Darstellung	22
4.5.2.	Kommunikation zwischen Oberfläche und Engine während einer Partie	23
4.5.3.	Aufbau der Nutzerstudie und Datenlogging	23
5.	Methodik der Nutzerstudie	25
5.1.	Forschungsdesign und Versuchsaufbau	26
5.2.	Stichprobe	28
5.3.	Messinstrumente	29
5.4.	Datenaufbereitung und Auswertungsstrategie	30
6.	Ergebnisse der Nutzerstudie	32
6.1.	Datenbasis und Stichprobe	32
6.2.	Subjektive kognitive Belastung	33
6.3.	Analyse des Spielverhaltens	34
6.3.1.	Entscheidungsdauer	34
6.3.2.	Zugqualität	35
6.3.3.	Partieausgänge	36
6.4.	Nutzerakzeptanz und qualitatives Feedback	36
6.4.1.	Akzeptanzurteile	36
6.4.2.	Freitextfeedback	37

6.5. Zusammenfassung der Ergebnisse	38
7. Diskussion	39
7.1. Interpretation der zentralen Befunde	39
7.1.1. Subjektive Belastung und vertraute Schachmuster	39
7.1.2. Kürzere Bedenkzeiten trotz höherer Belastung	40
7.1.3. Zugqualität und Partieausgänge	41
7.1.4. Spielspaß und weitere Einzelurteile	41
7.2. Implikationen für die Weiterentwicklung von Uncy	42
7.3. Limitationen der Untersuchung	43
7.3.1. Stichprobe und Übertragbarkeit	43
7.3.2. Versuchsdesign und Operationalisierung	43
7.3.3. Statistische und qualitative Aussagekraft	44
7.4. Zusammenfassung der Diskussion	45
8. Fazit und Ausblick	46
8.1. Fazit	46
8.2. Ausblick	48
Literatur	49
A. Erweiterte PERFT Vergleiche	51
B. Materialien zur qualitativen Inhaltsanalyse	53
B.1. Kategoriensystem	54
B.2. Fallbezogene Kurzparaphrasen	55
C. Verzeichnis des elektronischen Anhangs	57
D. Kurzanleitungen	58
D.1. Ausführung von Uncy.	58
D.2. Ausführung von Fairy-Stockfish	58

Abbildungsverzeichnis

3.1.	Komponenten und zentrale Datenflüsse von Uncy. Eigene Darstellung. . . .	13
4.1.	Visualisierung der getesteten Brettgeometrien zur Validierung der Zuggenerierung.	18
5.1.	Anfangsstellungen der vier Studienpartien in ihrer festen Reihenfolge. . . .	27
5.2.	Ablauf des Within-Subject-Designs. Eigene Darstellung.	28
6.1.	Verteilung der Probanden nach Alter (links) und Spielstärke (rechts)	32
6.2.	Mittlere subjektive Belastung nach RTLX-Dimension und Bedingung. Das positiv formulierte Leistungsitem wurde invertiert. Quelle: Eigene Darstellung; $n = 23$	33
6.3.	Mittlere Bedenkzeit je Zug nach Partie. Sämtliche Mittelwerte beruhen auf $n = 23$. Quelle: Eigene Darstellung.	34
6.4.	Mittlerer Centipawn Loss der menschlichen Züge nach Partie. Niedrigere Werte stehen für geringere Bewertungsverluste. Quelle: Eigene Darstellung. . . .	35
6.5.	Verteilung der Partieausgänge nach Phase. Quelle: Eigene Darstellung. . . .	36
6.6.	Verteilung der fünfstufigen Einzelitems. Quelle: Eigene Darstellung; $n = 23$	37
6.7.	Verteilung der meistgenannten Freitextaussagen. Quelle: Eigene Darstellung; $n = 23$	38

Tabellenverzeichnis

4.1.	Komponenten der Implementierung von <i>Uncy</i>	14
4.2.	Vergleich der PERFT-Knotenanzahl (Tiefe 5) zwischen Uncy und der Referenz-Engine Fairy-Stockfish.	19
4.3.	Rohsuchzeit auf den polymorphen Studienanfangsstellungen bei Tiefe sechs ($n = 10$ je Stellung).	22
A.1.	Gegenüberstellung der PERFT-Knotenzahlen von Uncy und Fairy-Stockfish fuer 20 zufällig aus den Studiendaten ausgewählte Stellungen bei Tiefe 5 . . .	52
B.1.	Kategoriensystem der vereinfachten qualitativen Inhaltsanalyse	54
B.2.	Kurzparaphrasen des pseudonymisierten Freitextfeedbacks	55

Hilfsmittelverzeichnis

Hilfsmittel	Version		Verwendungszweck
ChatGPT (OpenAI)	Modell	GPT-5.6 (https://chatgpt.com/)	Unterstützung bei der Gliederung einzelner Abschnitte, bei der Formulierung und sprachlichen Überarbeitung einzelner Textpassagen sowie bei der Erstellung und Anpassung einzelner LaTeX-Codefragmente für Abbildungen und Diagramme. Sämtliche verwendeten Ausgaben wurden fachlich geprüft, eigenständig ausgewählt und überarbeitet.
Gemini (Google)	Gemini	3.1 Pro (https://gemini.google.com/app)	Unterstützung bei der Gliederung und sprachlichen Überarbeitung einzelner Textpassagen sowie bei der Fehlersuche und der Erarbeitung einzelner Lösungsvorschläge beziehungsweise Codefragmente für den Softwareprototyp. Sämtliche übernommenen Vorschläge wurden nachvollzogen, getestet und eigenständig angepasst.
JASP	Version 0.98.1		Deskriptive und inferenzstatistische Auswertung der Nutzerstudie, insbesondere Normalitätsprüfungen, verbundene Mittelwertvergleiche, Effektstärken und Konfidenzintervalle.
Fairy-Stockfish	Fairy-Stockfish 230726 LB		Referenzengine zur Überprüfung der Zuggenerierung durch PERFT-Vergleiche.

1. Einleitung

1.1. Motivation

Die Entwicklung von Algorithmen, die es Computern ermöglichen, Schach erfolgreich zu spielen, hat das Spiel auch für Menschen verändert. Sogenannte „Engines“ haben schon lange die Oberhand über die besten menschlichen Spieler bekommen (Campbell et al., 2002, S. 57) und beeinflussen das menschliche Spielverhalten, das Studium der Schachtheorie sowie die allgemeine Perspektive auf das Spiel (Bilalić et al., 2026, S. 589–590, 602–605). Denn auch schon vor den Engines wurde die Eröffnungsvorbereitung, also das Auswendiglernen der ersten Zugabfolgen, auf dem 8x8 feldrigem Spielbrett und seiner starren Startposition hinterfragt. Mit Engines konnte das Ausrechnen von Eröffnungstheorie erweitert und vertieft werden. So war ein Lösungsansatz die Schachvariante des Chess960, welches die Startposition der Figuren auf eine Anzahl von 960 Variationen randomisierte (Polgar, 2025). Das Aufbrechen der starren Startposition reduzierte die direkte Übertragbarkeit auswendig gelernter Varianten. Eine weiterführende Möglichkeit könnte es sein, dies weiter zu gehen und neben der Startposition der Figuren, auch die Geometrie des Bretts selber aufzubrechen. Dies soll Mensch als auch Maschine zwingen, das Spiel wieder vermehrt auf Basis der Berechnung und fundamentalen Prinzipien wie der Mobilität der Figuren zu spielen. Diese Arbeit widmet sich diesem Ansatz und versucht, die Herausforderung der Berechnung eines solchen Vorhabens mittels einer dafür entwickelten, flexiblen Engine, sowie einer Nutzerstudie zur Untersuchung der Akzeptanz zu untersuchen. Für Brettkonfigurationen mit variablen Abmessungen und optional als unbetretbar markierten Feldern wird im Folgenden der Arbeitsbegriff „polymorphe Brettgeometrien“ verwendet. Die ausführliche begriffliche Abgrenzung erfolgt in Abschnitt 2.2.1.

1.2. Problemstellung

Moderne Schachengines sind auf das Spiel mit dem konventionellem 8×8 Brett optimiert. Sie nutzen Techniken, wie z.B. das sogenannte „Bitboard“, welche die 64-Bit-Länge eines Integers nutzt, um den Zustand eines Schachbretts darauf abzubilden, was auch in einer der bekanntesten und performantesten Engine, Stockfish, verwendet wird (Stockfish Developers,

2026). Diese Technik wäre für ein Brett mit mehr als 64 Feldern nicht ohne weitere Anpassungen anwendbar. Zudem erfordern inaktive Felder in der Spielfläche - umgesetzt durch unerreichbare, inaktive Felder - eine neue Interpretation der Mobilität einzelner Figurentypen. Und wo eine klassische Engine Piece-Square-Tables für die Evaluation der Stellung nutzt, um die Wertigkeit einer Figur, basierend auf seiner Position auf dem Brett, einzuschätzen, muss dieses Konzept neu gedacht werden, um einen Nutzen auf polymorphen Brettgeometrien zu haben. Die Herausforderung ist also eine Engine, die performant genug ist, um tiefe Suchbäume aufzubauen und gleichzeitig flexibel genug, um die Spielregeln und Heuristiken auf, vor der Laufzeit unbekannte, und dynamische Spielbrettkonfigurationen, anzuwenden.

1.3. Zielsetzung der Arbeit

Das technische Hauptziel ist die Entwicklung von 'Uncy', einer C# programmierten Schachengine, die nicht-standardisierte Spielbrettkonfigurationen dynamisch einlesen und interpretieren kann. Die Engine soll dabei nicht nur die Einhaltung der Regeln auf polymorphen Brettern gewährleisten, sondern auch selbstständig nach eigener Suche und Bewertung einen bestgefundenen Zug berechnen. Daraus leiten sich folgende Teilziele ab:

1.3.1. Entwicklung der Engine

Das erste Teilziel besteht in der Entwicklung einer Schachengine, deren Brettrepräsentation nicht fest an die Geometrie eines klassischen 8×8 -Bretts gebunden ist. Die neue Brettrepräsentation soll so entworfen werden, dass unterschiedliche Brettgrößen, sowie inaktive und damit nicht betretbare Felder abgebildet werden. Auf dieser Grundlage muss insbesondere die Zuggenerierung an die neuen geometrischen Bedingungen umgedacht werden. Auch die Stellungsbewertung sollte um Methoden ergänzt werden, welche dynamisch an die neue Herausforderung angepasst sind. Der Schwerpunkt liegt nicht auf einem Spielstärkevergleich mit bereits entwickelten Engines für konventionelles Schach. Stattdessen soll gezeigt werden, wie grundlegende Verfahren einer Schachengine auf, zur Laufzeit eingelesene, polymorphe Brettgeometrien übertragen werden können, um ausreichend Spielstärke in angemessener Zeit für interaktive Partien zu liefern.

1.3.2. Entwicklung einer Nutzerschnittstelle

Das zweite Teilziel ist die Entwicklung einer webbasierten Nutzerschnittstelle, über die die Funktionen der Engine zugänglich gemacht wird. Die Oberfläche soll Stellungen darstellen, Nutzereingaben verarbeiten und entsprechend Züge ausführen lassen können. Dadurch

soll es möglich sein, beliebige polymorphe Schachstellungen gegen Uncy zu spielen. Das Spielen in der Oberfläche dient dem übergeordneten Ziel, es als Erhebungsinstrument für die Nutzerstudie zu nutzen. Sie führt die Teilnehmenden durch die Registrierung, die Spielphasen und die Beantwortung der Fragebögen. Außerdem erfasst sie Interaktionsdaten, welche für die Auswertung relevant sind.

1.3.3. Nutzerstudie zu polymorphen Brettgeometrien

Neben der technischen Realisierung soll das Konzept von nicht-standardisierten Brettkonfigurationen empirisch überprüft werden. Ziel ist es, in einer Nutzerstudie die Akzeptanz der Spielenden auf solchen Brettern anhand exemplarischer, polymorpher Brettvarianten zu erforschen. In diesem Kontext soll gemeinsam mit der Engine aus dem technischen Hauptziel die übergeordnete Forschungsfrage erkundet werden, welche Unterschiede sich beim Spielen gegen diese Engine zwischen dem konventionellen 8×8 Standardbrett und polymorphen Brettgeometrien hinsichtlich subjektiver kognitiver Belastung, dem Spielerleben und dem individuellem Spielverhalten ergeben. Die genaueren Hypothesen und eine explorative Forscherfrage wird in Kapitel 5 der Methodik vorgestellt.

1.4. Aufbau der Arbeit

Die vorliegende Arbeit gliedert sich in acht Kapitel. Im Anschluss an diese Einleitung beleuchtet Kapitel 2 die theoretischen Grundlagen zu der Programmierung von Schachengines. In Kapitel 3 werden die Anforderungen an die Architektur von Uncy definiert, bevor Kapitel 4 detailliert auf die Implementierung der dynamischen Brettrepräsentation, der Zuggenerierung und der adaptiven Heuristiken eingeht. Kapitel 5 beschreibt die Methodik der durchgeführten Nutzerstudie zur Evaluation der polymorphen Geometrien. Die Ergebnisse dieser Untersuchung werden in Kapitel 6 deskriptiv und analytisch aufbereitet. In Kapitel 7 erfolgt eine Interpretation und kritische Diskussion der Ergebnisse. Die Arbeit schließt in Kapitel 8 mit einem Fazit und einem Ausblick auf zukünftige Forschungsmöglichkeiten ab.

2. Theoretische Grundlagen

Die Entwicklung einer Schachengine für polymorphe Spielbretter involviert sowohl technische als auch nutzerbezogene Fragestellungen. Auf technischer Ebene sollte geklärt werden, welche etablierten Verfahren des Computerschachs unabhängig von dem konventionellen 8×8-Brett eingesetzt werden können und an welchen Stellen Anpassungen erforderlich sind. Auf nutzerbezogener Ebene stellt sich die Frage, wie Spielende auf polymorphe Spielbretter reagieren, die von den vertrauten räumlichen Strukturen abweichen. Dieses Kapitel führt die dafür benötigten Grundlagen zusammen

Zunächst werden zentrale Algorithmen beleuchtet, die bei der Entwicklung von Schachengines geläufig sind. Danach werden Schachvarianten und unterschiedliche Ansätze zur flexiblen Repräsentation von Spielfeldern eingeordnet. Aus diesen Erkenntnissen werden die Folgen hinzu variablen Brettgrößen und inaktiven Feldern abgeleitet. Abschließend werden kognitionspsychologischen Grundlagen dargestellt, die später Teil der Nutzerstudie sind.

2.1. Grundlagen von Schach-Engines

2.1.1. Spielzustände, Suchbaum und Bewertungsfunktionen

Das Schachspiel kann als deterministisches Nullsummenspiel zwischen zwei Spielenden mit perfekter Information beschrieben werden (Mann, 2019, S. 12). Jeder legale Zug kann dabei eine neue Stellung erzeugen und somit eine neue Zustandsänderung auf dem Brett. Wenn man diese Zustandsänderungen rekursiv ausspielen lässt, also ausgehend von einer Initialstellung, jeden möglichen Zug rekursiv ausspielt und dies beliebig oft wiederholt, entsteht ein Suchbaum. Die Knoten repräsentieren Stellungen, während die Kanten Züge darstellen. Da sich die beiden Spielenden abwechseln und gegensätzliche Ziele verfolgen, kann die Zugauswahl als Minimax Problem formuliert werden. Es wird vorausgesetzt, dass eine Stellung bewertet werden kann und beide Seiten mittels Auswahl der besten Zügen versuchen, den Wert zu ihren Gunsten zu bewegen. Dieses Grundprinzip beschreibt Shannon (1950, S. 261–262) bereits in einer Arbeit zum Computerschach .

Weil die Untersuchung des daraus entstehenden Suchbaums praktisch nicht durchführbar wäre (Shannon, 1950, S. 259), müssen Schachengines die Suche begrenzen bis zu einer Suchgrenze, bzw. einer Suchtiefe. Um eine Stellung zu bewerten, wird eine Bewertungsfunktion geschaffen, welche eine Stellung auf konkrete Qualitätsmerkmale (vorhandenes Figurmateriale, Aktivität der Figuren und andere strukturelle Beschaffenheiten eines Bretts) untersuchen kann. Die Suche ist dabei von der Bewertung zu unterscheiden. Die Suche entscheidet, welche Knotenpunkte im Suchbaum betrachtet werden, während die Bewertungsfunktion die erreichte Stellung einschätzt und einen numerischen Wert für sie liefert, der dann verglichen werden kann (Campbell et al., 2002, S. 60–63, 73).

2.1.2. Alpha-Beta-Suche und Zugsortierung

Wenn der Spielbaum eine gleichmäßige Verzweigung hat, wächst die Zahl der betrachteten Knoten näherungsweise mit $O(b^d)$, wobei b den Verzweigungsfaktor, also die Anzahl an Zugmöglichkeiten von einer Stellung aus, und d die Suchtiefe bezeichnet. Hier kann das Alpha-Beta-Pruning genutzt werden, um den tatsächlich durchsuchten Teil des Spielbaums zu kürzen, ohne das Ergebnis der Suche durch Minimax zu verändern (Knuth & Moore, 1975, S. 296–301). Um dies zu realisieren, werden während der Suche Schranken für den bereits bekannten Wert einer Stellung mitgeführt. Wenn im weiteren Verlauf feststeht, dass von einer Stellung aus die Entscheidung eines übergeordneten Knotens nicht mehr beeinflusst werden kann, muss sie nicht weiter untersucht werden und der gesamte Teilbaum kann abgeschnitten werden.

Die Wirksamkeit des Alpha-Beta-Prunings hängt stark von der Reihenfolge der Züge ab, die von einer Stellung aus durchsucht werden. Wenn aussichtsreiche Züge für die Seite, die am Zug ist, zuerst betrachtet werden, können nicht gewinnbringende Teilbäume eher abgeschnitten werden. Die theoretische Analyse von Knuth und Moore (1975, S. 305–306, 309) bestätigt, dass sich der Aufwand unter idealisierten Bedingungen gegenüber einer ungekürzten Suche von $O(b^d)$ in Richtung $O(b^{d/2})$ verringern kann, wohingegen bei ungünstiger Reihenfolge der Aufwand nahe an der vollständigen Suche liegt. Um die Sortierung der Züge in der Praxis möglichst effizient zu gestalten, können mehrere Informationen kombiniert werden. Dazu zählen die Ergebnisse vorheriger Suchen, Einträge einer Transpositionstabelle oder eine Evaluation der Züge selber mittels Prüfung auf Schlagzüge oder Schachgebote.

2.1.3. Iterative Tiefensuche

Das bis hierhin beschriebene Verfahren zur Traversierung des Suchbaums mittels einer Tiefensuche wird durch die iterative Tiefensuche in einen anderen Rahmen gesetzt. Statt einer bloßen Suche zu einer bestimmten Tiefe führt die iterative Tiefensuche nacheinander

tiefenbegrenzte Suchen mit zunehmender Tiefe aus. Die Suche wird also immer wiederholt. Die Ergebnisse einer Iteration von dieser können aber dann zur Sortierung der Iteration bei der nächsthöheren Tiefe wieder aufgerufen werden. Der zusätzliche Aufwand ist bei exponentiell wachsenden Bäumen begrenzt, da ein großer Anteil der Knoten auf der jeweils tiefsten Ebene liegt (Korf, 1985, S. 100–101, 108). Für eine interaktive Engine besitzt das Verfahren den praktischen Nutzen, dass nach jeder abgeschlossenen Iteration für eine jeweilige Tiefe ein verwendbares Zwischenergebnis vorliegt. Wenn eine Engine ein Zeitlimit für ihre Berechnung erreicht, kann sie auf das zuletzt gespeicherte Ergebnis zurückgreifen.

2.1.4. Brettrepräsentationen

Die Brettrepräsentation beeinflusst die Generierung von legalen Zügen und sowie die Stellungsbewertung. In der Schachprogrammierung gibt es mehrere Ansätze, wie ein Spielfeld implementiert werden kann. Neben einer Vielzahl an Möglichkeiten, widmet sich dieses Unterkapitel der Vorstellung der zwei Alternativen, die im Recherche- und Entwicklungsprozess des Autors am präsentesten waren:

Bei einem klassischen Bitboard entspricht jedes Bit eines Maschinenworts, wie z.B. ein 64-Bit Integer, einem Brettfeld. Auf einem 8×8-Brett passen die 64 Felder genau auf diesen Datentyp. Wenn nun für jede Figur (weißer Bauer, schwarzer Bauer, weißer Läufer, etc.) ein Integer genutzt wird, kann im konventionellen Schach die Position der Figuren auf 12 Integer Werten abgebildet werden (Vučković, 2008, S. 157–158). Daraus folgt jedoch nicht, dass Bitboards grundsätzlich auf 64 Felder begrenzt sind. Doch wenn man die Felderanzahl erweitern will, muss das Datenlayout angepasst werden, was die Entwicklung erschweren würde.

Eine Brettrepräsentation mittels Array ordnet dagegen jedem Feld ein Element in dem Array zu, das den aktuellen Belegzustand kodiert. Die Größe des Arrays kann dann aus den Brettdimensionen abgeleitet werden. Für *Uncy* ist insbesondere relevant, dass die Repräsentation nicht strukturell an 64 Felder gebunden ist und die in der Implementation festgelegte maximale Brettgröße abgebildet werden kann. Die Entscheidung wird in Kapitel 4 begründet.

2.1.5. Hashing und Transpositionstabellen

Im Schach können unterschiedliche Zugfolgen in derselben Stellung enden. Das Phänomen, bei dem dies passiert, wird Transposition genannt. Um nicht jede Transposition im Suchbaum sich neu bewerten zu lassen, wäre es hilfreich ein Hashingverfahren zu nutzen, das eine Stellung identifizieren kann, sodass frühere Suchergebnisse für die Transposition erneut verwendet werden können. Das Zobrist Hashing ordnet einer Stellung einen Schlüssel zu, indem für relevante Merkmalsausprägungen vorbereitete Zufallswerte per XOR-Operation

kombiniert werden. Dadurch lässt sich der Schlüssel bei jedem Zug inkrementell aktualisieren. Der entstehende Schlüssel ist kein mathematisch eindeutiger Stellungsbezeichner. Kollisionen sind möglich und muss in der Folge bei Gestaltung und Weiterarbeit berücksichtigt werden (Zobrist, 1990, S. 69–72).

Mithilfe des Schlüssels können nun Transpositionstabellen genutzt werden, welche zu diesem die erreichte Suchtiefe, den jeweiligen Stellungswert, die Art der gespeicherten Schranke und den bevorzugten Zug (aus der aktuellen Stellung heraus für die Seite, die am Zug ist) speichern (Marsland, 1986, S. 12–15). Sollte nun eine Stellung während der Suche erreicht werden, die bereits einmal durchsucht worden ist, kann die Transpositionstabelle nun nützliche Informationen über diese Stellung preisgeben. Eine Transpositionstabelle hat eine begrenzte physische Größe im Speicher, weshalb eine Ersetzungsstrategie erforderlich ist. Es entsteht ein Konflikt zwischen tief untersuchten, aber älteren Einträgen und flacheren Suchen, aber dafür womöglich aktuelleren Einträgen. Untersuchungen verschiedener Ersetzungsstrategien zeigen, dass die Wahl der Strategie die Suchleistung beeinflussen kann und nicht unabhängig von Tabellengröße und Suchverhalten betrachtet werden sollte (Breuker et al., 1996, S. 175–180).

2.2. Schachvarianten und flexible Brettrepräsentationen

2.2.1. Begriffliche Abgrenzung

Als Schachvarianten werden Spiele bezeichnet, die mit Schach verwandt, daraus abgeleitet oder davon inspiriert sind. Abweichungen können beispielsweise das Spielbrett, die Zugweise der Figuren oder das Spielziel betreffen (Pritchard, 2007, S. 13). Für diese Arbeit sind Varianten relevant, bei denen die Bewegungsmuster der Figuren erhalten bleiben, aber die spielbare Fläche verändert wird.

Im Rahmen dieser Arbeit bezeichnet der projektspezifische Begriff „polymorphe Brettgeometrien“ nicht-standardisierte Brettkonfigurationen, deren Abmessungen vom konventionellen 8×8 Brett abweichen und/oder die inaktive, nicht betretbare Felder enthalten. Die Bezeichnungen „polymorph“ und „nicht-standardisiert“ werden im weiteren Verlauf für diese Konfigurationen synonym verwendet. Das konventionelle 8×8 Brett kann zwar ebenfalls durch das flexible Brettmodell von Uncy abgebildet werden, wird im empirischen Vergleich jedoch ausschließlich als Standardbedingung und nicht als polymorphe Brettgeometrie eingeordnet. Die Spielfelder bleiben dabei stets in einen zweidimensionalen, rechtwinkligen Koordinatenraum eingebettet.

2.2.2. Andere spezialisierte Spielsysteme

Systeme des General Game Playing verfolgen das Ziel, Spiele auf Grundlage einer formalen Regelbeschreibung zu verarbeiten, ohne für jedes Spiel einen vollständig spezialisierten Spieler vorauszusetzen. Ein GGP-System erhält unter anderem Informationen über Rollen, mögliche Aktionen, Zustandsübergänge und Zielbedingungen und muss daraus selbstständig ein Spielverfahren ableiten. Obwohl dieser Ansatz Flexibilität bietet, ein Brettspiel nach eigener Gestaltung zu entwickeln, stellt es das System vor die Aufgabe, aus der puren Beschreibung der Regeln spielrelevante Strukturen und Heuristiken zu gewinnen (Genesereth et al., 2005, S. 62–64).

An dieser Stelle lassen sich spezialisierte Engines für Schachvarianten abgrenzen, die das Problem spezifizieren. Sie verwenden schachspezifische Such- und Bewertungsverfahren und erweitern zugleich den Regelumfang. Fairy-Stockfish ist ein praktisches Beispiel für diesen Ansatz. Es handelt sich auch um eine Schachengine. Das Projekt unterstützt zahlreiche Schachvarianten und bietet Konfigurationsmöglichkeiten, das Spielfeld mitsamt Deaktivierung von Feldern und selbst die Figuren zu modifizieren. Jedoch limitiert sich diese Engine auf Spielfelder mit maximal 120 Feldern (12×10) (Fichter & Fairy-Stockfish Contributors, n.d.). Damit unterscheidet sich Fairy-Stockfish in der Zielsetzung von einer Engine wie *Uncy*. In dieser Arbeit bleiben die grundlegenden Schachregeln weitgehend fest, während die Brettgröße wesentlich größer sein darf.

Die hier vorgenommene Gegenüberstellung dient der Einordnung und nicht dem Nachweis einer Forschungslücke. Um eine absolute Aussage über die Neuartigkeit von *Uncy* zu treffen, würde es eine systematische Untersuchung bestehender Engines benötigen. Der Beitrag in dieser Arbeit wird daher vorrangig als Entwicklung mit empirischer Studie einer konkreten Lösung für einen definierten Anwendungsbereich verstanden.

2.3. Grundlagen der Nutzerstudie

2.3.1. Mustererkennung im Schachspiel

Chase und Simon zeigten, dass erfahrene Spielende Figurenkonstellationen nicht als eine Menge von isolierten Einzelfiguren verarbeiten. Stattdessen erkennen sie bedeutungshaltige Gruppen beziehungsweise *Chunks*, die mit im Langzeitgedächtnis gespeichertem Schachwissen integriert sind (Chase & Simon, 1973, S. 56, 80–81). Gobet und Simon erweiterten diesen Ansatz um das Konzept größerer und flexiblerer Templates, mit denen erfahrene Spielende wiederkehrende Stellungsstrukturen effizient aufnehmen können (Gobet & Simon, 1996b, S. 29, 31–33).

Dies hat die Folge, dass unter Zeitbeschränkung die schnelle Wiedererkennung bekannter Strukturen einen wesentlichen Beitrag zur Zugauswahl leisten kann (Gobet & Simon, 1996a, S. 52–54). Die Mustererkennung ergänzt also die vorausschauende Suche. In dieser Arbeit soll untersucht werden, ob veränderte Brettgeometrien die Übertragbarkeit dieser erlernten Muster beeinträchtigen können und was die potentiellen Auswirkungen daraus für das Spielerlebnis bedeuten.

2.3.2. Subjektive Beanspruchung und Entscheidungsverhalten

Subjektiv wahrgenommene Beanspruchung kann mehrdimensional erfasst werden. Ein verbreiteter Ansatz ist der NASA Task Load Index, der unterschiedliche Anforderungen und Reaktionen auf eine Aufgabe getrennt betrachtet (Hart & Staveland, 1988, S. 169; Hart, 2006, S. 904–906). In dieser Arbeit dient ein an den Untersuchungskontext angepasstes, ungewichtetes Instrument zur Messung der subjektiv empfundenen Belastung. Der Aufbau, sowie die Modifikationen und Berechnung werden in Kapitel 5 beschrieben.

Bedenkzeit und Entscheidungsqualität bilden unterschiedliche Aspekte des Entscheidungsverhaltens ab. Eine längere Entscheidungsdauer ist weder ein eindeutiger Beleg für höhere Belastung noch zwangsläufig mit einer besseren Entscheidung verbunden. Die Beziehung zwischen Geschwindigkeit und Genauigkeit hängt von der jeweiligen Aufgabe und Entscheidungssituation ab (Heitz, 2014, S. 1, 4). Beide Größen werden daher in dieser Arbeit getrennt untersucht.

2.4. Zusammenfassung und Bezug zur Forschungsfrage

Die technischen Grundlagen zeigen, dass zentrale Methoden der Schachprogrammierung nicht grundsätzlich an ein 8×8 Brett gebunden sind, wohingegen vor allem die Brettrepräsentation und zum Teil auch die Bewertungsfunktion Anpassungen benötigen, um auf polymorphen Brettern zu funktionieren. Die Konzeption und Implementation dieser Elemente ist die technische Herausforderung der folgenden Arbeit.

Die kognitionspsychologischen Grundlagen zeigen wiederum, dass Grund zur Annahme besteht, dass menschliches Schachspiel durch polymorphe Bretter die Spielenden zu neuen Orientierungs- und Entscheidungsanforderungen nötigt. Ob diese Anforderungen tatsächlich zu höherer subjektiver Belastung, zu längeren Bedenkzeiten und zu geringerer Zugqualität führen, wird dann Gegenstand der Nutzerstudie sein.

3. Konzeption

3.1. Anforderungsanalyse

Aus den Zielen dieser Arbeit ergeben sich Anforderungen an die generalisierte Schachengine, sowie ihre technische Bereitstellung und die Webanwendung als Schnittstelle zur Nutzerstudie. Die Anforderungen sind iterativ während des Entwicklungsprozesses entstanden und werden in diesem Kapitel aber nun zusammengefasst formuliert. Im Mittelpunkt stehen die Eigenschaften, die für polymorphe Bretter und die Durchführung der Nutzerstudie erforderlich sind. Eine vollständige Produktspezifikation ist daher nicht Gegenstand dieser Arbeit. Um die spätere Überprüfung zu erleichtern, werden die Anforderungen im Folgenden mit Kennungen versehen. Dies dient der Zuordnung bei der Verifikation der Anforderungen in Kapitel 4.

3.1.1. Funktionale Anforderungen

F1: Verarbeitung polymorpher Brettgeometrien Die Engine muss zweidimensionale Spielflächen mit einer Größe von bis zu 30×30 Feldern verarbeiten können. Innerhalb dieses Koordinatenraums kann ein Feld aktiv oder inaktiv sein.

F2: Einlesen vollständiger Spielzustände Stellungen müssen einschließlich ihrer Geometrie und der für die Zugregeln benötigten Beschreibung initialisiert werden können. Als Grundlage dient die Forsyth-Edwards-Notation (FEN). Diese beschreibt im Standardschach die Figurenbelegung, sowie spezielle Zustandseigenschaften (Edwards, 1994). Die Figurenbelegung muss polymorphe Bretter entgegennehmen können, was eine größere Brettgröße, sowie als inaktiv deklarierte Felder involviert. Für das En-Passant Zielfeld wird ein Koordinatenpaar verwendet (z.B. 5,3 entspricht dem Feld e3).

F3: Regelkonforme Zuggenerierung Die bekannten Bewegungsmuster der Schachfiguren bleiben erhalten (FIDE, 2023), müssen jedoch auf die jeweils eingelesene Geometrie angewendet werden können. Keine Figur darf ein inaktives Feld betreten. Gleitfiguren dürfen ein inaktives Feld nicht überspringen. Springer dürfen ein inaktives Feld überspringen. Bauern ziehen, je nach Farbe, hoch oder runter, wie im konventionellen Schach und schlagen diagonal. Wenn sie das letzte Feld einer Spalte erreicht haben, wandeln sie sich in eine andere Figur um. Für eine Rochade des Königs, muss der Weg zum beteiligten Turm frei sein, damit der König dann zwei Felder in dessen Richtung zieht, um die Rochade auszuführen.

F4: Berechnung eines Zugs der Engine Für jede gültige, nichtterminale (Matt oder Patt) Stellung muss die Engine mithilfe von Such- und Bewertungsfunktionen einen legalen Zug geben können, den sie nach eigenen Maßstäben für die beste Entscheidung hält. Sie durchsucht dabei den Suchbaum aus der Initialstellung heraus bis zu einer gewissen Tiefe und gibt den Zug zurück.

F5: Technische Bereitstellung Die Schachlogik der polymorphen Bretter (was das konventionelle 8×8 Brett inkludiert) muss auf einer interaktiven Oberfläche im Web nutzbar sein. In dieser soll ein Spiel gegen *Uncy* gespielt werden können auf einem beliebigem polymorphen Brett. Sowohl der Spieler als auch die Engine sind limitiert mit einer Bedenkzeit pro Person/Computer, die vorher ausgewählt werden kann.

F6 Durchführung und Protokollierung der Nutzerstudie Die Webanwendung muss die Teilnehmenden durch Einwilligung, die Spielphasen und die Fragebögen führen. Alle Benutzereingaben, als auch Nutzungsdaten, die in Kapitel 5 konkret beschrieben werden, müssen pseudonymisiert und akkurat festgehalten werden.

3.1.2. Nicht-funktionale Anforderungen

N1: Korrektheit und Testbarkeit Sowohl für das Standardbrett als auch für polymorphe Bretter muss die Implementierung die Anzahl aller Knoten des Suchbaums bis zu einer bestimmten Tiefe benennen können, um die Korrektheit ihrer Zustandsverarbeitung und Zuggenerierung zu zeigen.

N2: Reaktions- und Suchleistung Die Zeit, welche die Engine benötigt, um eine Suche von einer Stellung aus zu tätigen, muss schnell genug sein, damit sie praktisch gegen einen Menschen konkurrieren kann, wenn beide Seiten mit einer limitierenden Bedenkzeit spielen.

Eine durchschnittliche Bedenkzeit von unter einer Sekunde auf einer Auswahl von polymorphen Brettern, die später in der Studie genutzt werden, bei einer Tiefe von sechs Halbzügen, ist angestrebt.

N3: Modularität Der Kern der Engine, sowie ein dazugehöriges Auswertungstool und ein Frontend, inklusive Server für die Benutzung müssen bereitgestellt sein und modular verfügbar. Die Regeln der Engine werden nur einmal implementiert und gelten dann systemweit. Alle Module haben einen konkreten Zuständigkeitsrahmen.

N4: Verständlichkeit der Bedienung Usability ist vom Nutzungskontext und den Nutzenden abhängig. Eine vollständig kognitiv neutrale Oberfläche wird nicht vorausgesetzt. Stattdessen sollen vermeidbare Bedienprobleme als Störfaktor begrenzt werden. Figuren, aktive und inaktive Felder, Auswahlzustände und ausgeführte Züge, sowie möglich ausführbare Züge sollen erkennbar sein. Baseline- und Interventionsphase verwenden dieselben Interaktionsmechanismen, sodass die Bedingungen sich während der Studie hauptsächlich in den Spielinhalten unterscheiden.

Die Anforderungen werden nicht durch eine große Evaluation geprüft. Vielmehr ist es vorgesehen, dass im Kapitel 4 die Erfüllung eines Großteils der Anforderungen gezeigt werden kann.

3.2. Systemarchitektur

Dieses Unterkapitel soll im Allgemeinen die Systemarchitektur vorstellen, um einen groben Überblick über die konzeptuelle Aufteilung der Verantwortlichkeiten zu liefern. In Abbildung 3.1 wird diese visualisiert. Es wird dabei nicht ins Detail gegangen, um nicht die Inhalte aus Kapitel 4 vorwegzunehmen.

Den Kern des Systems bildet Uncy.Shared. Dies ist eine Bibliothek auf Basis von .Net 9, welche die gesamte Verarbeitung der polymorphen Bretter, sowie die Berechnung mitsamt aller Komponenten, übernimmt. Sie besitzt selber keine graphische Oberfläche und wird von den anderen Modulen angesteuert, um in deren Ausführungsumgebungen eingebunden zu werden.

Uncy.Server stellt die Engine über eine ASP.NET-Core Anwendung bereit. Der Server vermittelt die Kommunikation zwischen Weboberfläche und Kernbibliothek über einen SignalR-Hub. Für jede Verbindung wird ein eigener Engine- und Spielzustand verwaltet, was das Spielen von mehreren Clients gleichzeitig ermöglicht. Hier findet mithilfe des StudyLogger auch die Protokollierung der Studiendaten statt. Der Server liefert das gebaute Frontend aus.

Das Frontend befindet sich im Projekt `uncy-frontend` und basiert auf React und Vite. Es enthält den Studienablauf, die Spieloberfläche sowie ergänzende Test- und Editorfunktionen. Ein Vite-Build wird erstellt und in ein Verzeichnis abgelegt, um dann durch den Server ausgeliefert zu werden an den Endclient. Daten der Teilnehmenden werden über die bestehende SignalR-Verbindung an den Server übermittelt.

Die Konsolenanwendung `Uncy.Engine` greift ebenfalls auf `Uncy.Shared` zu. Sie dient technischen Experimenten, zum Debugging und Benchmarks. Sie wird vor allem im Entwicklungsprozess genutzt, und um die Funktionsfähigkeit der einzelnen Komponenten aus `Uncy.Shared` zu testen.

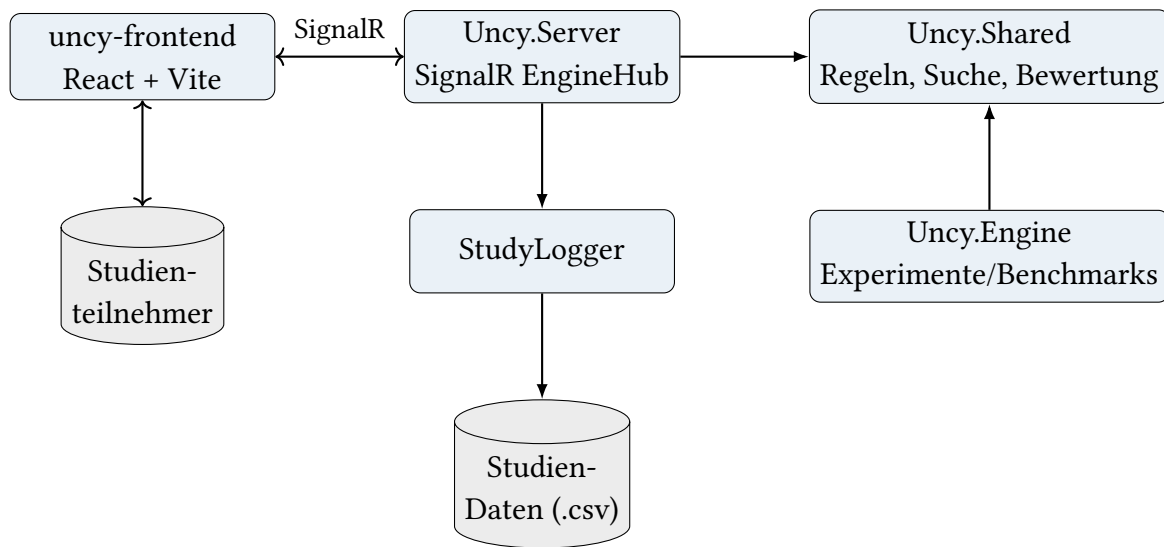


Abbildung 3.1.: Komponenten und zentrale Datenflüsse von Uncy. Eigene Darstellung.

4. Implementierung

4.1. Technischer Gesamtaufbau

Die aktuelle Anwendung ist in mehrere Projekte mit getrennten Verantwortlichkeiten aufgeteilt. Tabelle 4.1 fasst die für den Betrieb und die Entwicklung relevanten Komponenten zusammen.

Tabelle 4.1.: Komponenten der Implementierung von *Uncy*.

Komponente	Verantwortung
Uncy.Shared	.NET-9-Kernbibliothek für Brettrepräsentation, Zustandsübergänge, Zuggenerierung, Suche, Bewertung, Hashing und Perft. Die Spielregeln werden hier zentral implementiert.
Uncy.Server	ASP.NET-Core-Anwendung, welche die Kernbibliothek einbindet, den SignalR-Hub /enginehub bereitstellt, das gebaute Frontend ausliefert und die im Studienbetrieb anfallenden Daten protokolliert.
uncy-frontend	Aktuelle React-Oberfläche auf Basis von Vite. Sie enthält den Studienablauf, die Spielansicht sowie ergänzende Entwickleransichten für freie Partien, Tests und die Erstellung von Brettstellungen.
Uncy.Engine	Konsolenprojekt für technische Experimente, Perft-Aufrufe, Suchmessungen, Selbstspiele und die Untersuchung einzelner Stellungen. Es ist nicht Teil der Benutzeroberfläche.

Vom Userflow ausgehend, beginnt das Projekt in uncy-frontend. Nutzereingaben werden hier entgegengenommen. Eine Verbindung über SignalR besteht mit Uncy.Server. Dieser verarbeitet die Nutzereingabe mithilfe der Bibliothek Uncy.Shared, welche die Engine Logik enthält. Der Vite-Build des Frontends wird in Uncy.Server/wwwroot abgelegt und vom Server gemeinsam mit dem SignalR Endpunkt ausgeliefert. Diese Modulaufteilung erfüllt den wesentlichen Gedanken hinter der Anforderung N3, dass Server, Oberfläche und technische Werkzeuge dieselbe Regelimplementierung aus Uncy.Shared verwenden.

4.2. Polymorphe Zustandsrepräsentation

4.2.1. Brett- und Feldkodierung

Die Klasse Board speichert die Felder eines Bretts in einem eindimensionalen Array vom Typ `byte[]`. Das erste Element im Array bildet das Feld unten links ab und geht von dort aus reihenweise aufwärts, sodass das letzte Arrayelement das Feld oben rechts abbildet. Seine Länge wird beim Initialisierungsprozess in der Methode `InitializeBoard()` als Produkt aus der erkannten Reihen und Spalten bestimmt. Es enthält also keine Festlegung auf eine gewisse Anzahl an Feldern. Für ein Brett mit 30×30 Feldern würden beispielsweise 900 Elemente im Array reserviert.

Aus dem Bytewert eines Felds lässt sich seine Art und seine Belegung auslesen, was in der Klasse: `Piece` definiert wird. Die Bits 0 bis 2 kodieren die sechs Figurentypen und das leere Feld, die Bits 3 und 4 die Farbe. Das höchstwertigste Bit kennzeichnet ein inaktives Feld.

Bits 0–2 (PieceTypeMask 0b00000111): Definiert den Figurtyp (0 = Empty, 1 = Pawn, 2 = Knight, 3 = Bishop, 4 = Rook, 5 = Queen, 6 = King).

Bits 3–4 (ColorMask 0b00011000): Definiert die Farbe (8 = White, 16 = Black).

Bit 7 (InactiveMask 0b10000000): Der Dezimalwert 128 markiert ein Feld als inaktiv.

Abfragen nach Figurtyp, der Farbe, ob ein Feld frei ist oder sogar inaktiv können dann über bitweise Maskierungen durchgeführt werden. Da die Abfrage der Brettbelegung während der Suche sehr häufig genutzt wird, ist diese Bitmaskierung vor allem eine Optimierungsscheidung gewesen.

Mit dieser Implementierung ist die Anforderung F1 zum Teil erfüllt, dass die Engine eine zweidimensionale Spielfläche aus beliebigen Dimensionsgrößen (bis 30×30) erstellen kann und sie aktiv oder inaktive Felder beinhalten kann. Es ist an der Stelle noch anzumerken, dass die Anforderung eine zweidimensionale Spielfläche vorschreibt und in Board nur ein eindimensionales Array genutzt wird. Dies geschieht aus Effizienzgründen. Dass dieses eindimensionale Array korrekt den zweidimensionalen Raum abbildet, wird mit der Belegung von F3 (Regelkonforme Zuggenerierung) dann noch gezeigt.

4.2.2. Erweiterte FEN-Verarbeitung

Die Stellung für das polymorphe Brett wird über eine erweiterte Form der Forsyth-Edwards-Notation eingelesen (Edwards, 1994). Wie bei üblichen FEN besteht die Eingabe auch hier aus sechs durch Leerzeichen getrennte Bereiche. Doch es kommen eigenständig entwickelte Anpassungen hinzu, um polymorphe Bretter einlesen zu können. Es ist damit möglich, beliebig

viele Schrägstriche anzugeben und auch mehrere Figuren pro Reihe zu definieren, um größere Bretter zu definieren.

Das Einlesen des FEN ist vor allem beim Initialisierungsprozess wichtig. Die Entgegennahme und das Auslesen der FEN übernimmt die Klasse `Fen`, die gemeinsam mit der Klasse `FenParser` arbeitet. Die Bereiche des FEN Strings werden in ihre einzelnen Informationen zerlegt. Eine Klasse `BoardInitializer` nimmt dann diese extrahierten Daten entgegen und passt die Instanz von `Board` daran an.

Neben dem Initialisierungsprozess dient die erweiterte FEN auch der Kommunikation von Stellungen auf polymorphen Brettern zwischen Frontend und Server, bzw. der Engine-Logik.

Mit dieser Implementierung ist die Anforderung an F2 (Einlesen vollständiger Spielzustände) zunächst einmal erfüllt. Es ist allerdings aus softwaretechnischer Sicht anzumerken, dass der Parser, sowie das System bezüglich der erweiterten FEN, gutmütig entwickelt worden ist. Es findet beim Initialisierungsprozess eines Bretts kein Kontrollprozess statt, der genauestens untersucht, ob die Stellung, die ein FEN String beschreibt, tatsächlich Sinn ergibt. Im Rahmen der durchgeführten Studie sollte dies kein Problem darstellen, da die Teilnehmenden nicht mit den FEN Strings selber konfrontiert sind und wird daher hier der Vollständigkeit halber erwähnt.

4.2.3. Dimensionsabhängige Lookup Tabellen

Da die Brettrepräsentation in *Uncy* ein zweidimensionalen Raum auf ein eindimensionales Array abbildet, muss mit Offsets gearbeitet werden, um benachbarte Felder zu lokalisieren. So entspricht eine Bewegung auf dem Brett nach Norden dem Offset $+w$, wobei w die Brettweite ist. Eine diagonale Bewegung nach Nordosten hat den Offset $+(w + 1)$. Da die Spielbretter immer unterschiedlich sein können und w nicht gleich bleiben muss, ist der Offset variabel und muss jedes mal neu ausgerechnet werden. Hinzu kommt die Herausforderung, dass die Logik mit dem Aufaddieren des Offsets auf das aktuelle Feld nicht mehr funktioniert, wenn es sich um ein Spielfeld am linken oder rechten Rand handelt, da eine Bewegung weiter nach außen auf der anderen Seite der nächsten Reihe landen würde, was kein benachbartes Feld ist.

Da während der Suche sehr häufig darauf geprüft werden müsste, ob ein Rand bereits erreicht ist und welche Felder im Allgemeinen erreichbar sind, wird diese Überprüfung direkt bei der Initialisierung einer neuen `Board` Instanz einmal im Voraus erledigt. Dies geschieht in der Klasse `MoveLookupTables`, die für die erkannten Dimensionen die passenden Bewegungsmuster für alle Figuren in Zieltabellen abspeichert, sodass zur Laufzeit bei der Suche direkt aus der Tabelle die Daten gezogen werden können. Für König- und Springerbewegungen werden dabei die direkten Zielindizes verwendet. Bei Gleitfiguren auf potentiell sehr großen Brettern

hätte das Abspeichern von Zielindizes den Nachteil, dass sehr viele Indizes gespeichert werden müssten. Statt diese also alle abzuspeichern, wird lediglich die Distanz bis zum äußeren Brettrand gespeichert. Gleitfiguren „wissen“ damit, wie weit sie von einem Feld in eine Richtung iterieren können.

4.3. Zuggenerierung auf polymorphen Geometrien

4.3.1. Repräsentation und Legalitätsprüfung von Zügen

Ein Zug wird in der Implementierung als unveränderlicher Werttyp (Struct) `Move` repräsentiert. Er enthält alle Daten, die benötigt werden, um einen Zug zweifelsfrei zu identifizieren. Die Klasse `MoveGenerator` kann pseudo-legale Züge, für die Seite die am Zug ist, erzeugen. Dabei werden die Bewegungsregeln der Figuren berücksichtigt, aber ein Zug kann den eigenen König im Schach lassen, was ihn zu einem illegalen Zug macht. Daher muss nach der Generierung der Pseudo-legalen Züge eine Legalitätsprüfung stattfinden, die in *Uncy* in der `MakeMove()` Methode erfolgt. Ein Pseudo-legaler Zug wird hier erstmal ausgeführt, danach wird überprüft, ob der eigene König im Schach steht und falls dies der Fall ist, wird der Zug per `UnmakeMove()` Methode wieder zurückgenommen. Falls der Zug tatsächlich legal war, könnte die Suche dann von diesem Punkt aus weiter fortfahren, da die entstandene Stellung gültig ist. Dies hat den Vorteil, dass nicht überprüft werden muss, *wie* der Zug dafür sorgt, dass der eigene König im Schach steht, sondern nur pauschal bewertet, *dass* der eigene König im Schach steht.

4.3.2. Figurenspezifische Behandlung der Geometrie

Die Berechnung der möglichen Züge für jede Figur werden im `MoveGenerator` wesentlich mithilfe der `MoveLookUpTable` implementiert. Dabei verhält es sich für die einzelnen Figurentypen wie folgt:

- Königszüge verwenden die vorberechneten benachbarten Zielindizes. Ein inaktives Ziel wird verworfen. Ob das aktive Zielfeld angegriffen ist, entscheidet die nachgelagerte Legalitätsprüfung.
- Springer verwenden ebenfalls vorberechnete Ziele. Nur Ausgangs- und Zielfeld sind für ihre Bewegung relevant. Inaktive Felder zwischen beiden Punkten werden nicht betrachtet, sodass der Springer sie entsprechend F3 überspringen kann. Ein inaktives Zielfeld bleibt unzulässig.

- Türme, Läufer und Damen folgen den vorberechneten Richtungsvektoren bis zur äußeren Brettgrenze. Eine eigene Figur, eine gegnerische Figur oder ein inaktives Feld beendet den Strahl. Eine gegnerische Figur kann als letztes Feld geschlagen werden; ein inaktives Feld kann weder betreten noch übersprungen werden.
- Bauern verwenden farbabhängige Vorwärts- und Diagonalrichtungen. Ein Vorwärtzug setzt ein leeres aktives Zielfeld voraus, während ein Diagonalzug eine gegnerische Figur oder das aktuelle En-passant-Ziel benötigt. Der Doppelschritt wird von der dimensionsabhängigen zweiten beziehungsweise vorletzten Reihe aus erzeugt. Beim Erreichen des äußeren letzten Rangs werden vier Umwandlungen in Dame, Turm, Läufer und Springer angelegt.

4.3.3. Validierung der Zuggenerierung

Um die korrekte Regelimplementierung und infolge dessen die Zuggenerierung zu verifizieren, wird das etablierte Performance Test (PERFT)-Verfahren angewandt. Hierbei durchsucht eine Engine den Spielbaum bis zu einer definierten Tiefe und summiert die exakte Anzahl der generierten Blattknoten. Es findet dabei kein Abschneiden eines Zweigs oder von Transpositionen statt. Für das konventionelle Schach gelten universelle Standards bis zu gewissen Tiefen. Es ist naheliegend, dass dies für alle möglichen Varianten an polymorphen Brettern nicht existiert. Die Ergebnisse aus den PERFT Tests der Engine Uncy werden mit den validierten Outputs der etablierten Variante Fairy-Stockfish verglichen. Es wird dabei eine Tiefe von 5 Halbzügen gewählt. Im Folgenden wird nur eine kleine Anzahl an Brettern (Abbildung 4.1) exemplarisch verglichen. Bei der Entwicklung der Engine ist dieses Verfahren intensiv eingesetzt worden, um die Fehler in der Zuggenerierung zu beheben.

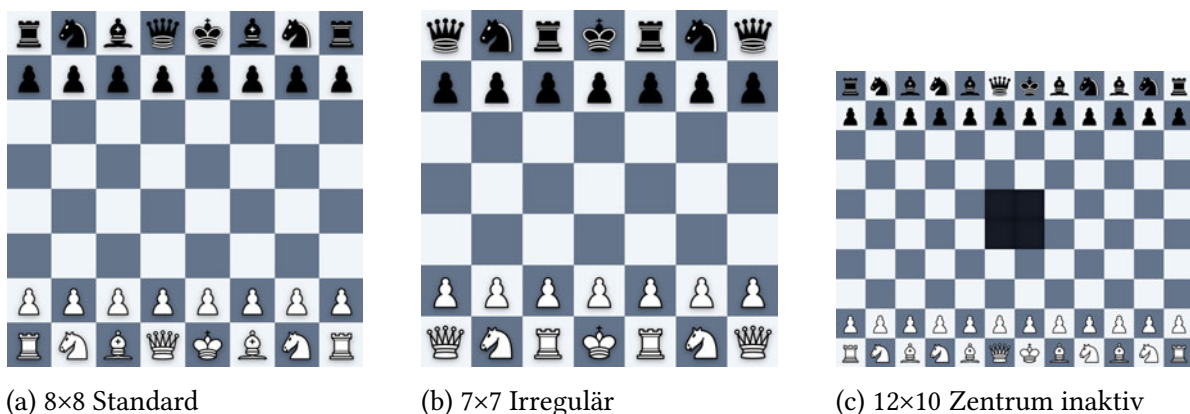


Abbildung 4.1.: Visualisierung der getesteten Brettgeometrien zur Validierung der Zuggenerierung.

Die Tests umfassen drei Varianten. ein klassisches 8x8 Schachbrett, ein verkleinertes 7x7 Brett, sowie eines mit einer 12x10 Struktur, welches inaktive Felder in der Mitte besitzt. Eine größere Struktur als 12x10 zu testen, lässt die Engine Fairy-Stockfish, aus architektonischen Gründen, nicht zu.

Tabelle 4.2.: Vergleich der PERFT-Knotenanzahl (Tiefe 5) zwischen Uncy und der Referenz-Engine Fairy-Stockfish.

Brettgeometrie	Spezifikation	Uncy	Fairy-Stockfish
8×8	(a) Standard	4.865.609	4.865.609
7×7	(b) Irregulär	2.486.479	2.486.479
12×10	(c) Zentrum inaktiv	45.533.161	45.533.161

Die Gegenüberstellung zeigt in Tabelle 4.2 eine Übereinstimmung der Knotenzahlen in allen Varianten. Weil die Logik in der Implementation der Engine Uncy dimensionsagnostisch berechnet wird, wird abgeleitet, dass die Zuggenerierung auch auf noch größeren Geometrien korrekt skaliert und damit die Anforderung F3 (Regelkonforme Zuggenerierung) bestätigt und sogleich auch die nicht-funktionale Anforderung N1, dass eine Korrektheit und Testbarkeit der Züge gewährleistet wird.

Nach Durchführung der Studie wurden aus dem Datensatz 20 zufällige Stellungen ausgewählt, um auf ihnen auch nochmal die Knotenanzahl zu messen, nach dem selben Schema wie in Tabelle 4.2. Die Ergebnisse waren auch in dem Fall fehlerfrei und sind im Anhang A abgebildet.

4.4. Stellungsgedächtnis, Suche- und Bewertungsfunktionen

4.4.1. Zobrist-Hashing und Transpositionstabelle

Da unterschiedliche Zugfolgen dieselbe Stellung erreichen können, verwendet *Uncy* das im Theoriekapitel beschriebene Zobrist-Hashing. Die Schlüssel für dieses Verfahren werden nach Ermittlung der Brettdimensionen erzeugt. Eine Board Instanz hält eine Instanz der *ZobristKeys* Klasse. Für jedes Feld existieren dort Einträge für die möglichen Figurenkodierungen und sämtliche Spezialschlüssel für Zug- und Rochaderecht, sowie En-Passant Ziel. Der initiale Schlüssel wird auch bei der Brettinitialisierung erstellt und bildet sich aus der XOR-Verknüpfung von allen zutreffenden Einträgen. Wenn dann ein normaler Zug später ausgeführt wird, werden nur die Schlüsselmerkmale per XOR verknüpft, die tatsächlich

betroffen sind. Wenn eine weiße Dame auf einem 8×8 Brett von a1 auf a2 ziehen würde, dann würde der `currentZobristKey` mit dem Schlüssel von einer weißen Dame auf a1 XOR verknüpft. Das Ergebnis wird dann wieder XOR verknüpft mit dem Schlüssel für eine weiße Dame auf a2. Nach diesem Muster entsteht nach jeder Veränderung auf dem Brett ein neuer Schlüssel, der als Identifikator für diese Stellung dient.

Die Transpositionstabelle wird als Array mit einer Zweierpotenz als Eintragszahl angelegt. Der Index eines Zobrist-Schlüssels kann dadurch mit einer Bitmaske statt einer Division bestimmt werden. Ein Eintrag enthält den vollständigen 64-Bit-Schlüssel zur Kollisionsprüfung, den Stellungswert, die erreichte Suchtiefe, die Art des Werts als exakter Wert beziehungsweise obere oder untere Schranke sowie den besten gefundenen Zug. Ein vorhandener Eintrag wird nur ersetzt, wenn das neue Ergebnis mindestens dieselbe Suchtiefe besitzt. Die Tabellengröße ist bei der Erzeugung einer Engine konfigurierbar; der Server verwendet pro Verbindung 32 MB.

4.4.2. Iterative Suche und Baumreduktion

Die Klasse `Search` setzt die im Theoriekapitel eingeführte Minimax Suche mit Alpha-Beta-Pruning um. Positive Bewertungswerte begünstigen Weiß, negative Werte Schwarz. Die Wurzelsuche maximiert oder minimiert, je nachdem, welche Seite gerade am Zug ist. Um bei einer Zeitbegrenzung nach mindestens einer abgeschlossenen Iteration ein Ergebnis bereitzuhalten, beginnt `FindBestMove()` mit Tiefe eins und erhöht dieses Limit schrittweise bis zur angeforderten Maximaltiefe. Außerdem werden ergänzende Verfahren genutzt, um den untersuchten Suchraum zu verringern oder an vielversprechenden Stellen sogar zu erweitern:

- Das Null-Move-Pruning untersucht in geeigneten Stellungen eine reduzierte Suche nach einem ausgelassenen Zug. Es wird nicht ohne weiteres in reinen Bauernendspielen eingesetzt, um typische Zugzwangfehler zu begrenzen (David-Tabibi & Netanyahu, 2002, S. 153–154).
- Late Move Reductions reduzieren ab dem fünften betrachteten Zug die Tiefe ruhiger, spät sortierter Züge. Überschreitet ein reduziertes Ergebnis die aktuelle Schranke, wird der Zug mit voller Tiefe erneut untersucht (Chessprogramming Wiki, [n. d.](#)).
- Erreicht die Hauptsuche ihre Grenze in einer Schachstellung, wird sie um einen Halbzug verlängert. Andernfalls folgt eine Ruhesuche, die Schlagzüge und Umwandlungen weiterverfolgt, bevor die statische Bewertung zurückgegeben wird (Marsland, 1986, S. 6–7).

4.4.3. Zugsortierung

Während, die oben genannten, separaten Reduktionstechniken dafür sorgen, dass viele unnötige Zweige des Suchbaums geschnitten werden können, bildet das Alpha-Beta Pruning die Basis der Suche. Für dieses Verfahren ist es besonders wichtig, dass die möglichen Züge, durch die bei jedem Knoten iteriert wird, optimal sortiert sind. Im Idealfall ist der beste Zug auch der erste Zug, der weiter durchsucht wird. Darum wird dies in *Uncy* auch in der separaten Klasse *MoveSorter* ausgelagert. Dieser bedient die Suche in einer priorisierten Reihenfolge mit Zügen. Jeder Zug erhält einen Score und wird mit Selection Sort sortiert. Der Score eines Zugs wird nach folgender Priorität gewichtet.

1. **Hash Move (Score 2.000.000):** Der beste Zug aus der Transpositionstabelle wird ohne erneute Evaluierung immer als Erstes durchsucht.
2. **Schlagzüge via MVV-LVA (Score 1.000.000 + Δ):** Nach dem Prinzip *Most Valuable Victim – Least Valuable Attacker* werden Schläge bevorzugt, bei denen eine schwache Figur eine wertvolle gegnerische Figur schlägt. Das Delta stellt die Differenz aus MVV-LVA dar.
3. **Killer Moves (Score 800.000 / 900.000):** Zwei Slots für ruhige Züge, die in anderen Geschwisterknoten auf derselben Suchtiefe erfolgreich einen Beta-Cutoff verursacht haben.
4. **History Heuristic:** Alle übrigen ruhigen Züge erhalten einen Score basierend auf vergangenen Cutoffs. Die Gewichtung skaliert quadratisch mit der Tiefe ($depth^2$), indiziert über das Muster [movedPiece, toSquare].

4.4.4. Geometrieabhängige Stellungsbewertung

Die Bewertungsfunktion von *Uncy* kombiniert zwei Bewertungsanteile. Zum einen existiert die Klasse *MaterialEvaluator*, welche die auf dem Brett vorhandenen Figuren zählt und gegenüberstellt. Zusätzlich dazu wird die Klasse *PieceSquareTable* genutzt, die bei der Brettinitialisierung positionsabhängige Werte für die einzelnen Felder und für jede Figur auf dem Spielbrett ermittelt. Die Felder folgen für die Figuren der Grundregel, dass mehr Mobilität belohnt wird. Ein Feld, das unterdurchschnittlich viele Felder erreichen kann, wird dementsprechend als schlechter gewertet für die Figur aus deren Perspektive man die Mobilität betrachtet. Bauern orientieren sich nicht an der Mobilität, sondern an ihrem Fortschritt bis zum Ende des Bretts.

Es ist an dieser Stelle anzumerken, dass die Bewertungsfunktion simpel gehalten worden ist und viele weitere Techniken, die auch die unterschiedlichen Spielphasen betrachten bei ihrer Einschätzung noch möglich wären. Um *Uncy* grob einschätzen lassen zu können mag das

ausgereicht haben, aber hier wäre noch Verbesserungspotential und das Forschungspotential mit anderen Technologien, die auf Reinforcement Learning basieren, bietet hier noch ein spannendes Feld, das aber nicht Gegenstand dieser Arbeit sein soll.

4.4.5. Suchleistung auf den Studienbrettern

Zur Prüfung des Zielwerts aus N2 wurde die reine Dauer von `GetBestMove()` auf den drei polymorphen Anfangsstellungen der Studie gemessen. Die Messung erfolgte auf einem AMD Ryzen 7 7800X3D mit Windows in 64-Bit-Ausführung. Pro Geometrie wurde ein Aufwärmrunden verworfen. Danach wurden zehn unabhängige Engine-Instanzen mit jeweils 32 MB Transpositionstabelle bis Tiefe sechs ohne Zeitabbruch gesucht. Gemessen wurde ausschließlich die Suchmethode; Initialisierung, SignalR-Übertragung und die künstliche Serververzögerung waren nicht Teil der Zeit.

Tabelle 4.3.: Rohsuchzeit auf den polymorphen Studienanfangsstellungen bei Tiefe sechs ($n = 10$ je Stellung).

Geometrie	Mittelwert	Std.-Abw.	Minimum	Maximum
12×12	42,9 ms	28,3 ms	23 ms	96 ms
9×13	20,8 ms	0,9 ms	19 ms	22 ms
7×7	34,6 ms	2,0 ms	32 ms	39 ms

Alle 30 Suchen lieferten einen nichtleeren Zugwert. Tabelle 4.3 zeigt, dass alle Mittelwerte unter den beschriebenen Bedingungen deutlich unter dem Zielwert von einer Sekunde liegen. Damit ist N2 für diese Anfangsstellungen und als grober Richtwert erfüllt. Die Ergebnisse sind nicht als allgemeine Skalierbarkeitsaussage zu verstehen: Andere Stellungen derselben Geometrie können einen anderen Verzweigungsfaktor besitzen, und Messungen auf anderen Systemen können abweichen. In diesem Zuge wird auch F4 (Berechnung eines Zugs der Engine) als bestätigt vermerkt.

4.5. Bereitstellung als Web- und Studienanwendung

4.5.1. Darstellung

uncy-frontend bietet neben dem geführten Studienmodus eine freie Partie gegen *Uncy*, einen Playground für technische Tests und einen Bretteditor zur Erstellung polymorpher Stellungen. Für freie Partien können unter anderem Ausgangsstellung, Spielerfarbe, Suchtiefe

und Bedenkzeit gewählt werden. Der Bretteditor erzeugt aus Figuren, leeren und inaktiven Feldern eine Stellung in der erweiterten FEN. Dies hilft insbesondere, um die polymorphen Bretter zu designen, die für die Studie genutzt werden sollten.

Die Komponente Play erzeugt die Brettdarstellung als CSS-Grid aus den in der FEN erkannten Dimensionen. Inaktive Felder werden gesondert dargestellt und als visuell mittels Schattierung als nicht begehbar erkenntlich gemacht. Für die ausgewählte Figur, legale Zielfelder, mögliche Schlagzüge und den letzten Enginezug verwendet die Oberfläche unterschiedliche Markierungen. Baseline- und Interventionspartien greifen auf dieselbe Komponente und dieselben Interaktionsmechanismen zurück. Damit ist der technische Teil von N4 weitgehend erfüllt. Ob die Darstellung tatsächlich verständlich ist, lässt sich jedoch erst durch die Nutzung der Oberfläche bewerten.

4.5.2. Kommunikation zwischen Oberfläche und Engine während einer Partie

Das React Frontend in `uncy-frontend` öffnet eine SignalR-Verbindung zur Klasse `EngineHub` in `Uncy.Server`. Die wichtigsten Aufrufe bilden einen wiederkehrenden Ablauf:

1. `SetupBoard` initialisiert die Engine mit der gewählten FEN.
2. Bei der Auswahl einer Figur fordert `GetLegalMovesForSquare` die legalen Zielfelder an.
3. `PlayerMove` rekonstruiert die übermittelte Stellung, sucht den passenden legalen Zug und führt ihn serverseitig aus.
4. Der Server sendet die aktualisierte FEN sowie Kennzeichen für Schlagzug, Schach und Rochade zurück.
5. Ist anschließend die Engine am Zug, ruft das Frontend `PlayEngineMove` mit Suchtiefe und verbleibender Zeit auf.

Dadurch bleibt die Regelprüfung auf dem Server und verwendet dieselbe Kernbibliothek (`Uncy.Shared`).

4.5.3. Aufbau der Nutzerstudie und Datenlogging

Die Komponente Study führt durch Studieninformation und Einwilligung, Registrierung, vier Partien und die vorgesehenen Fragebögen. Auf die Baseline mit dem 8×8 -Brett folgen die drei polymorphen Bretter mit 12×12 , 9×13 und 7×7 Feldern. Die angegebene Schacherfahrung bestimmt eine Suchtiefe zwischen eins und fünf; pro Seite stehen fünf Minuten zur Verfügung.

Der Server vergibt eine zufällige GUID als Teilnehmendenkennung. Die Klasse `StudyLogger` in `Uncy.Server` speichert Registrierungsangaben, Züge und resultierende FENs, Partieergebnisse sowie Fragebogenantworten in vier CSV-Dateien. Damit sind der geführte Ablauf und die in F6 vorgesehenen Datenarten implementiert. In `CSVEvaluator` werden die separaten Rohstudienendaten bereinigt und vorbereitet in einen gemeinsamen Analysedatensatz, der dann von einer Auswertungssoftware analysiert wird. Dies wird in Kapitel [5](#) genauer beschrieben.

5. Methodik der Nutzerstudie

Nachdem in den vorangegangenen Kapiteln die technischen Hintergründe zu „Uncy“ beleuchtet worden sind, beschreibt dieser Teil der Arbeit die Methodik hinter der empirischen Nutzerstudie. Die Untersuchung orientiert sich an der übergeordneten Forschungsfrage, welche Unterschiede sich beim Spielen gegen *Uncy* zwischen dem konventionellen 8×8-Standardbrett und polymorphen Brettgeometrien hinsichtlich der subjektiv empfundenen kognitiven Belastung, des Spielerlebens und des individuellen Spielverhaltens ergeben.

Den theoretischen Ausgangspunkt bilden Erkenntnisse zur Mustererkennung im Schach. Erfahrene Spielende verarbeiten wiederkehrende Figurenkonstellationen unter anderem als bedeutungshaltige Einheiten und greifen auf im Langzeitgedächtnis gespeichertes Schachwissen zurück (Chase & Simon, 1973, S. 56, 80–81; Gobet & Simon, 1996b, S. 29, 31–33; Gobet & Simon, 1996a, S. 52–54). Wie in Abschnitt 2.3 dargestellt, könnten veränderte Brettgeometrien die Übertragbarkeit dieser erlernten Muster beeinträchtigen. Ob daraus tatsächlich eine höhere Belastung, längere Bedenkzeiten oder eine geringere Entscheidungsqualität folgen, ist jedoch nicht vorwegzunehmen, sondern Gegenstand der empirischen Untersuchung.

Um die Annahme überprüfbar zu machen, werden folgende Hypothesen aufgestellt:

1. **H1 (Kognitive Belastung):** Das Spielen auf polymorphen Brettern führt zu einer signifikant höheren subjektiv empfundenen kognitiven Belastung im Vergleich zum 8x8 Standardbrett
2. **H2 (Entscheidungsdauer):** Das Spielen auf polymorphen Brettern führt zu einer signifikant höheren Bedenkzeit pro menschlichem Zug im Vergleich zum 8x8 Standardbrett.
3. **H3 (Entscheidungsqualität):** Aufgrund der höheren kognitiven Beanspruchung beim Spiel auf polymorphen Brettern, sinkt die Qualität der Entscheidungen während des Spiels im Vergleich zum 8x8 Standardbrett.

Die Hypothesen H1 bis H3 beziehen sich auf die gerichteten Vergleiche zwischen der Standardbedingung und der Interventionsbedingung. Daneben soll auch untersucht werden, ob die ungewohnte Brettkonfiguration als Störgefühl oder als motivierende Herausforderung wahrgenommen wird. Dies gibt Anlass zur Ergänzung durch die Formulierung einer explorativen Teilforschungsfrage:

1. **TFF1 (Spielerlebnis):** Wie wirken sich die polymorphen Brettgeometrien auf den subjektiven Spielspaß aus und welche Implikationen ergeben sich daraus für die Weiterentwicklung des Konzepts?

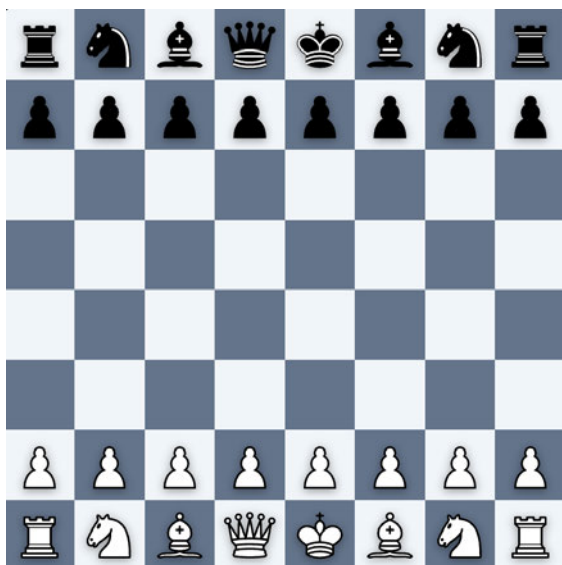
5.1. Forschungsdesign und Versuchsaufbau

Der Versuch ist einem Within-Subject-Design nach aufgebaut. Alle Teilnehmenden durchlaufen alle Bedingungen und dienen damit jeweils als eigene Vergleichsperson. Der grobe Ablauf ist in Abbildung 5.2 veranschaulicht. Dies hat den Vorteil, dass Varianzen durch unterschiedliche individuelle Eigenschaften (Alter, Spielerfahrung) besser kontrolliert werden können.

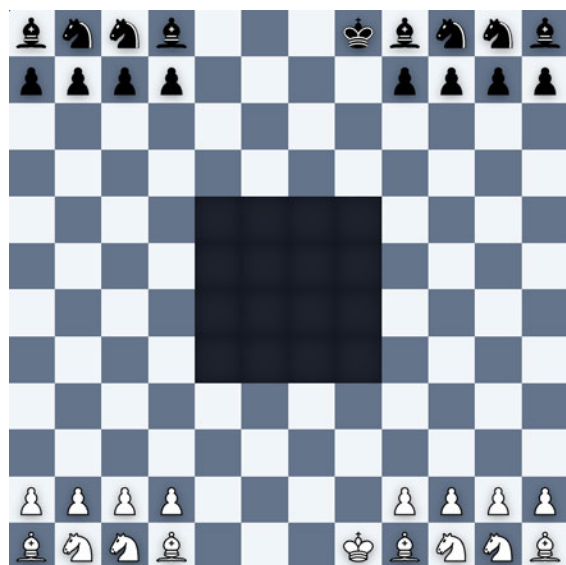
Der Versuchsaufbau ist konzeptionell in zwei aufeinanderfolgende Phasen unterteilt. In der Baselinephase absolvieren die Teilnehmenden eine Partie auf dem gewohnten 8×8 Brett (Abbildung (a) in 5.1). Dies dient zum einen der Ermittlung der Basismetriken (Referenzwerte für Zeit und Evaluation). Zusätzlich kann diese Partie als Akklimatisierungsphase verstanden werden, in der die Probanden sich, in einem bekannten Kontext, an das UI gewöhnen können. Anschließend folgt ein Zwischenfragebogen zur Feststellung der Baseline.

In der darauffolgenden Interventionsphase wird auf polymorphen Brettern gespielt. Sie absolvieren drei Partien auf fest vorgegebenen, jeweils unterschiedlichen Brettern, die in Abbildung 5.1 visualisiert sind ((b), (c), (d)). Verwendet wurden ein 12×12 -Brett mit einem inaktiven Bereich von 4×4 Feldern in der Brettmitte, ein 9×13 -Brett, das mittig gespalten ist, sowie ein verkleinertes 7×7 -Brett ohne inaktive Felder. Die drei Stellungen unterschieden sich neben der Brettgröße und der Anordnung inaktiver Felder auch im Figurenbestand und in der Startaufstellung. Die Entscheidung, drei strukturell unterschiedliche Bretter zu verwenden wird in einem Dilemma aus zwei Interessen getroffen. Zum einen soll versucht werden, dass die gemessenen Effekte auf das abstrakte Konzept der polymorphen Spielbretter zurückzuführen ist und nicht auf ein einzelnes Brett. Andererseits muss der zeitliche Aufwand limitiert werden, um die Frustrationsgrenze der Probanden nicht zu überschreiten. Nach Abschluss aller Partien folgt ein Abschlussfragebogen, der die Erfahrung aller drei Bretter bewertet.

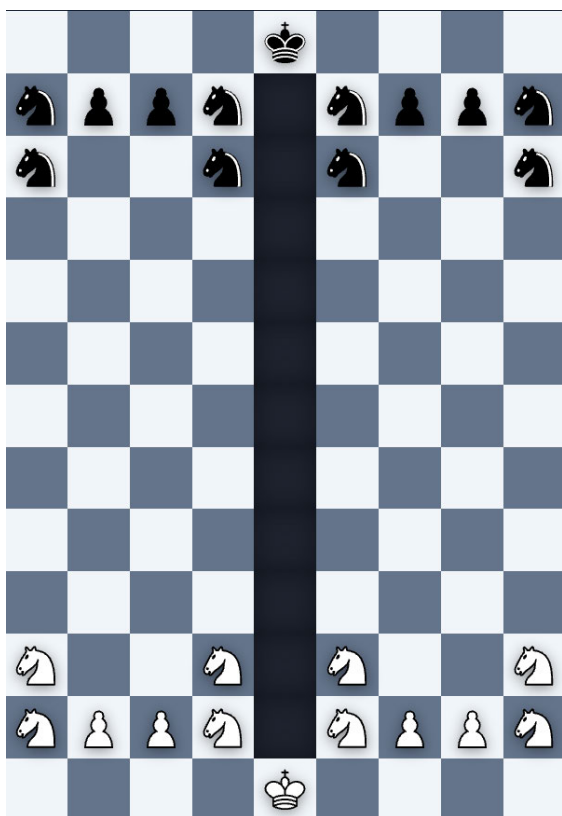
Die Spielstärke der Engine (ihre Berechnungstiefe) passt sich an die Angaben an, welche die Probanden am Anfang der Studie über ihre individuelle Schacherfahrung angeben. Die Angaben unterscheiden sich in fünf Klassifizierungsstufen von „Anfänger“ bis „Experte“ und repräsentieren eine Berechnungstiefe von eins bis fünf Halbzügen. Sämtliche Partien werden aus Aufwandsgründen auf fünf Minuten Bedenkzeit für jede Seite beschränkt. Dies beabsichtigt den Mittelweg, um zum einen die Aufwand- und Frustrationsgrenzen der Probanden



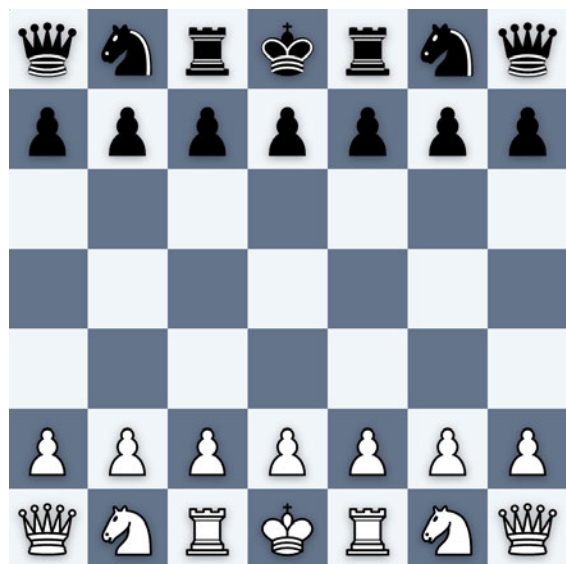
(a) Standardbedingung: 8×8



(b) Polymorph 1: 12×12 mit inaktivem Zentrum:
„Ringbrett“



(c) Polymorph 2: 9×13 : „Pferderennen“



(d) Polymorph 3: 7×7 : „Kompakt“

Abbildung 5.1.: Anfangsstellungen der vier Studienpartien in ihrer festen Reihenfolge.

gering zu halten und zum Anderen genug Zeit, um sich angemessen mit dem ungewöhnlichen Brett auseinanderzusetzen.

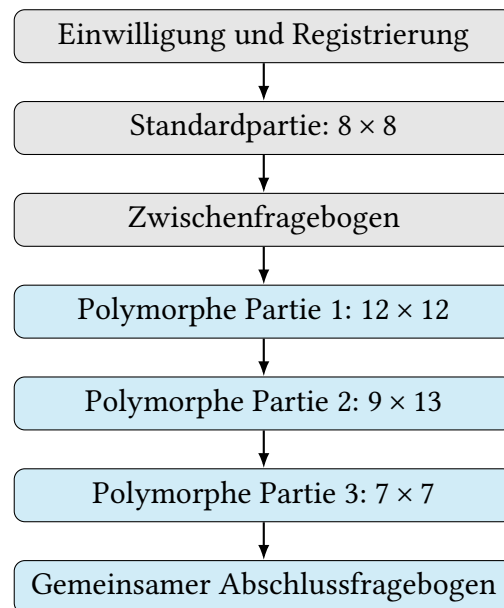


Abbildung 5.2.: Ablauf des Within-Subject-Designs. Eigene Darstellung.

Sollte ein Proband sich in einer verlorenen Stellung während der Partie wiederfinden, wird ihm die Möglichkeit gegeben, die Partie aufzugeben. Das Aufgeben im Schach zur frühzeitigen Beendung der Partie ist eine gängige und etablierte Umgangsform und daher wurde sich entschieden, diese Möglichkeit nicht auszulassen. Es sollte an dieser Stelle konkret darauf hingewiesen werden, dass alle Personen die Standardpartie zuerst und die drei Interventionspartien anschließend in derselben Reihenfolge spielten. Dadurch fand keine Randomisierung oder Gegenbalancierung statt. Mögliche Lern-, Anpassungs- und Ermüdungseffekte lassen sich deshalb nicht vollständig von den Unterschieden zwischen den Bedingungen trennen und werden in der Diskussion als Limitation berücksichtigt.

5.2. Stichprobe

Die Rekrutierung der Probanden erfolgt im Rahmen einer anfallenden Stichprobe. Es wurden primär Menschen aus dem universitären Umfeld oder dem erweiterten Bekanntenkreis des Autors zur Teilnahme aufgerufen. Zusätzlich wurde der Studienaufruf in einem Schachverein verbreitet. Diese Methode wurde gewählt, um im zeitlichen Rahmen der Bachelorarbeit eine ausreichende Gruppengröße zu gewährleisten.

Um valide Daten zu gewährleisten, wurde ein grundlegendes Einschlusskriterium definiert. Die Teilnehmenden mussten die fundamentalen Regeln des Schachspiels beherrschen. Jedoch

war Wissen darüber hinaus, keine Voraussetzung, da die Studie explizit untersuchen soll, wie Spieler unterschiedlicher Erfahrungsstufen auf die Veränderung reagieren. Personen, ohne jeglichen Schachkenntnisse, wurden aus dem Versuch ausgeschlossen, weil die Einarbeitung in die Grundregeln womöglich die Messung der kognitiven Last zu stark beeinflusst hätte. Vor Beginn der Studie wurden Daten zum Alter der Person, sowie zur Selbsteinschätzung der Schacherfahrung erhoben. Eine detailliertere Auswertung dieser Stichprobendaten erfolgt dann in Kapitel 6.

Die Teilnahme an der Studie erfolgte vollständig freiwillig und war frei zugänglich für alle Personen, die über den Link verfügten. In die Analysestichprobe wurden jedoch ausschließlich volljährige Teilnehmende aufgenommen. Es gab keine Belohnung oder Aufwandsentschädigung. Sie konnte jederzeit abgebrochen werden. Alle verwendeten Daten wurden erst nach einer informierten Einwilligung pseudonymisiert gesammelt.

5.3. Messinstrumente

Um die Hypothesen und Forschungsfragen zu beantworten, werden objektive, systemseitig erfasste Nutzungsdaten mit subjektiven Selbstauskünften der Teilnehmenden kombiniert. Für jeden legal ausgeführten Halbzug protokollierte das System die pseudonyme Teilnehmendenkennung, die Partie, die Nummer des Halbzugs, die ziehende Farbe, die Einordnung als menschlicher oder Engine-Zug, das gemessene Zeitintervall und die nach dem Zug entstandene Stellung. Zusätzlich wurden das Partieergebnis, die gewinnende Seite und die Anzahl der Halbzüge gespeichert.

Die Entscheidungsdauer für H2 wurde anhand der durchschnittlichen Zeit bestimmt, die ein Teilnehmender brauchte, um zu ziehen. Das protokollierte Zeitintervall beginnt mit der zuvor empfangenen Brettaktualisierung und endet mit dem Empfang der Aktualisierung nach Ausführung des menschlichen Zugs. Die Entscheidungsqualität für H3 wurde nach Abschluss der Erhebung mithilfe einer separaten Analysekonfiguration von *Uncy* quantifiziert. Die protokollierten Stellungen wurden mit einer für alle Personen und Bretter konstanten Suchtiefe von sechs Halbzügen bewertet. Als Indikator dient dabei der durchschnittliche Qualitätsverlust pro Zug. Dieser berechnet sich aus der Differenz der Stellungsbewertung vor und nach dem ausgeführten menschlichen Zug. Höhere Qualitätsverluste deuten auf eine geringere Entscheidungsqualität hin. An dieser Stelle sollte erwähnt werden, dass die Bestimmung der Entscheidungsqualität also von den Messwerten von der Such- und Bewertungsfunktion von *Uncy* abhängig ist.

Die subjektiv empfundene Belastung wurde mit einem modifizierten, ungewichteten Raw TLX erhoben (Hart & Staveland, 1988, S. 169; Hart, 2006, S. 904–906). Die vorgenommenen Anpassungen stellen eine studienspezifische Abweichung vom Originalinstrument dar. Die

Dimension der physischen Anforderungen wurde entfernt, da sie für die vorwiegend kognitive Interaktion mit der Weboberfläche nur eine geringe inhaltliche Relevanz aufweist. Die verbleibenden fünf Dimensionen mentale Anforderung, zeitliche Anforderung, wahrgenommene Leistung, Anstrengung und Frustration wurden sprachlich an den Schachkontext angepasst und über 20-stufige Schieberegler von 1 bis 20 erfasst. Das Leistungsitem ist im eingesetzten Fragebogen positiv gepolt, sodass ein hoher Wert eine hohe wahrgenommene Leistung ausdrückte. Vor der Bildung des Belastungswerts wurde dieses Item daher mit $x' = 21 - x$ umgepolt. Anschließend wurde der ungewichtete arithmetische Mittelwert der fünf Items gebildet. Höhere Werte stehen dadurch bei allen Bestandteilen für eine höhere subjektive Beanspruchung. Der so gebildete Wert diente als Indikator für H1.

Neben dem modifizierten, ungewichteten Raw TLX werden im Kontext des Game Designs die Fragebögen durch Single-Item Skalen ergänzt (5-stufige Likert Skala). Grund dafür ist es, die Ergebnisse aus der subjektiven kognitiven Belastung als negativen Stress oder als motivierende Herausforderung einordnen zu können. In diesen Single-Item Fragen geben die Probanden an, inwiefern sie das polymorphe Schach als spannend empfanden und wie stark sie die Notwendigkeit zur Adaption ihres Wissens aus dem konventionellem Schach schätzten. Außerdem werden zwei weitere Likert Items zur Validierung zukünftiger Entwicklungsziele integriert. Diese erheben den potentiellen Wunsch der Probanden nach einem Editor, um eigene Brettgeometrien zu erstellen, sowie ihr Interesse an einem Spiel gegen menschliche Gegner anstelle der Engine.

Der Fragebogen schließt ab mit einem offenem Textfeld, das für qualitatives Feedback genutzt werden kann. Probanden haben hier die Möglichkeit, die größte Umstellung oder Herausforderung beim Spielen auf dem polymorphen Brett zu benennen.

5.4. Datenaufbereitung und Auswertungsstrategie

Vor der eigentlichen Analyse erfolgt eine Bereinigung des Rohdatensatzes. Als vollständig gilt ein Durchlauf, wenn die Registrierung, die Ergebnisdatensätze aller vier Partien sowie der Zwischen- und der Abschlussfragebogen vorlagen. Unvollständige Durchläufe wurden aus den verbundenen Vergleichen ausgeschlossen. Da es sich um eine Remotestudie handelt, können externe Ablenkungen auf den Probanden nicht kontrolliert werden. Eine Bedenkzeit für einen Zug von über einer Minute (Gesamtbedenkzeit ist fünf Minuten) wird in der Berechnung der durchschnittlichen Entscheidungsdauer als Ausreißer betrachtet und von der Berechnung ausgeschlossen.

Die vollständigen Durchläufe wurden zusätzlich einer manuellen Plausibilitätsprüfung unterzogen. Dabei wurde ein Durchlauf ausgeschlossen, bei dem sämtliche drei Interventionspartien nach höchstens zwei protokollierten Halbzügen beendet worden waren. Für diesen

Durchlauf lag damit keine hinreichende Bearbeitung der Interventionsphase und keine belastbare Grundlage für die Analyse der Bedenkzeit und Zugqualität vor. Eine einzelne frühe Aufgabe war dagegen kein Ausschlussgrund, da das Aufgeben einer Partie im Studienablauf erlaubt war. Der manuelle Ausschluss erfolgte vor der inferenzstatistischen Auswertung und unabhängig von den Fragebogenwerten.

Die anschließende Datenauswertung gliedert sich dann in drei methodischen Ansätzen. Die deskriptiven und inferenzstatistischen Analysen werden mit JASP (Version 0.98.1) durchgeführt. Zunächst wird die deskriptive Statistik vorgestellt. Das beinhaltet die grundlegende Beschreibung des Datensatzes für alle erhobenen Variablen, also aus den Fragebögen, sowie aus dem Logging der Nutzungsdaten im Hintergrund. Diese werden primär in Form von Mittelwerten und Standardabweichungen, sowie relativen und absoluten Häufigkeiten dargestellt. Die Nutzungsdaten aus den drei Partien auf den polymorphen Brettern werden pro teilnehmender Person zu einem individuellen Mittelwert aggregiert. Im darauffolgenden Schritt wird zur Überprüfung der Hypothesen H1, H2 und H3 der Vorher-Nachher-Vergleich zwischen der Baseline- und Interventionsphase auf statistische Signifikanz geprüft. Da es sich um verbundene Stichproben handelt, kommt hierfür der gepaarte t-Test zum Einsatz. Sollte die Voraussetzung der Normalverteilung bei diesem verletzt sein, wird auf den nicht-parametrischen Wilcoxon-Vorzeichen-Rang-Test ausgewichen. Als Signifikanzniveau wird ein α -Fehler von 5 Prozent ($p < 0,05$) festgelegt. Außerdem wird das qualitative Feedback aus dem Freitextblock mittels einer vereinfachten qualitativen Inhaltsanalyse ausgewertet. Die Antworten werden kategorisiert und anschließend ihrer Häufigkeit deskriptiv aufbereitet. Prägnante Zitate können in der späteren Diskussion der inhaltlichen Fundierung der gemessenen Effekte dienen.

6. Ergebnisse der Nutzerstudie

Dieses Kapitel berichtet die Ergebnisse der Nutzerstudie in der Reihenfolge der in Kapitel 5 formulierten Hypothesen und der explorativen Forschungsfrage. Zunächst werden Datenbasis und Stichprobe beschrieben. Anschließend folgen die Analysen der subjektiven kognitiven Belastung, der Bedenkzeit und der Zugqualität. Abschließend werden die Akzeptanzurteile sowie das Freitextfeedback dargestellt. Die inhaltliche Einordnung der Befunde erfolgt dann in Kapitel 7.

6.1. Datenbasis und Stichprobe

Insgesamt registrierte das System 51 Starts der Studie. Nach Anwendung der in Kapitel 5 beschriebenen Bereinigungskriterien verblieben 23 auswertbare Personen (45,1 %). Der bereinigte Datensatz umfasst 92 Partieergebnisse, 7.113 protokollierte Halbzüge und 368 Fragebogenantworten. Sämtliche 23 Personen beantworteten sowohl den Fragebogen nach dem Standardspiel als auch den Abschlussfragebogen nach den drei polymorphen Spielen. Die gepaarten Zugdaten- und Fragebogenanalysen umfassen daher jeweils $n = 23$.

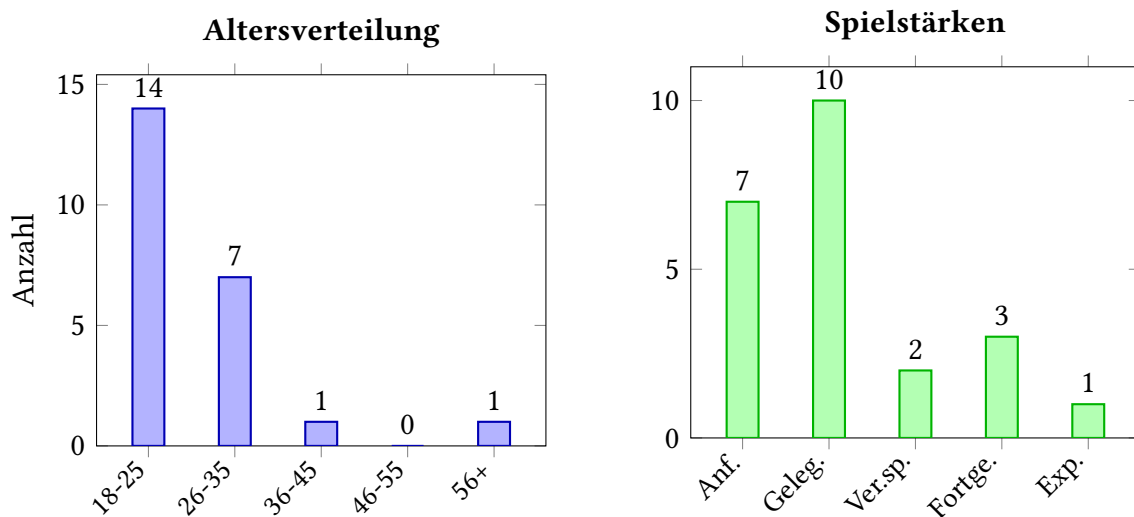


Abbildung 6.1.: Verteilung der Probanden nach Alter (links) und Spielstärke (rechts)

Abbildung 6.1 zeigt die Altersverteilung und die Spielstärkenverteilung. Daraus lässt sich ablesen, dass das mittlere Alter 26,30 Jahre ($SD = 9,38$; Median = 25; Spannweite: 18–62 Jahre) betrug. Die Stichprobe deckte alle erhobenen Erfahrungsstufen ab, war jedoch durch Anfängerinnen und Anfänger sowie Gelegenheitsspielende geprägt (Anfänger: 7; Gelegenheitsspieler: 10; Vereinsspieler: 2; Fortgeschritten: 3; Experte: 1).

6.2. Subjektive kognitive Belastung

Zur Prüfung von H1 wurde aus den fünf erhobenen RTLX-Dimensionen je Person und Phase ein ungewichteter Mittelwert auf der Skala von 1 bis 20 gebildet. Da das Leistungsitem positiv als subjektiver Erfolg formuliert war, wurde es für den Belastungsindex invertiert (21 minus Rohwert). Höhere Werte kennzeichnen damit in allen Dimensionen eine höhere Belastung. Sämtliche Teilskalen lagen in der polymorphen Bedingung über den Werten der Standardbedingung, wie in Abbildung 6.2 zu erkennen ist. Der stärkste mittlere Anstieg zeigte sich bei der geistigen Anforderung (+3,70 Punkte), gefolgt vom Zeitdruck (+3,35 Punkte).

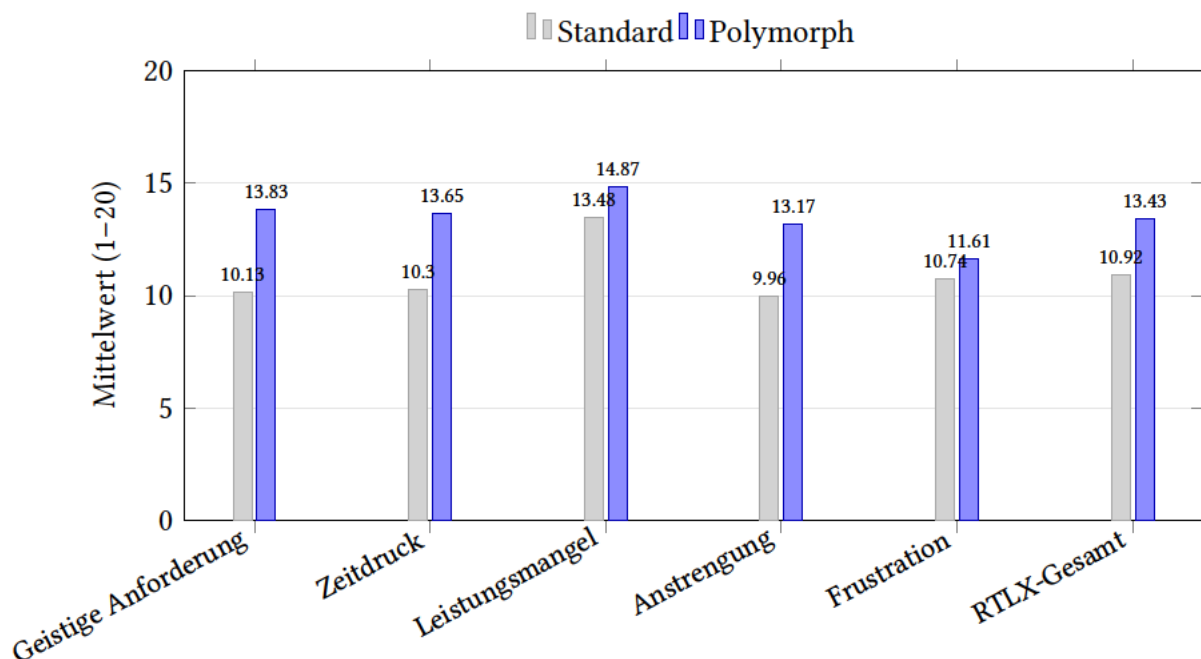


Abbildung 6.2.: Mittlere subjektive Belastung nach RTLX-Dimension und Bedingung. Das positiv formulierte Leistungsitem wurde invertiert. Quelle: Eigene Darstellung; $n = 23$.

Die Differenzwerte des RTLX-Gesamtwerts weichen nicht signifikant von einer Normalverteilung ab (Shapiro–Wilk: $W = 0,984$; $p = 0,964$). Aus diesem Grund wurde der in Kapitel 5 vorgesehene gepaarte t -Test verwendet. Der mittlere RTLX-Wert stieg von $M = 10,92$

($SD = 4,75$) auf $M = 13,43$ ($SD = 3,05$). Die mittlere Differenz betrug 2,50 Punkte (95 %-KI [0,43; 4,58]) und war statistisch signifikant, $t(22) = 2,499$; $p = 0,020$; $d_z = 0,52$. H1 wird damit durch die vorliegenden Daten unterstützt.

6.3. Analyse des Spielverhaltens

6.3.1. Entscheidungsdauer

Für die Analyse der Bedenkzeit wurden ausschließlich menschliche Züge berücksichtigt. Wie in Kapitel 5 festgelegt, wurden vier Einzelzüge mit einer erfassten Dauer von mehr als 60 Sekunden von der Mittelwertbildung ausgeschlossen. Pro Person wurde zunächst für jede Partie die mittlere Bedenkzeit je Zug berechnet. Der Interventionswert entspricht dem ungewichteten Mittel der drei polymorphen Partiemittel, sodass jede Geometrie gleich stark in den Phasenwert eingeht.

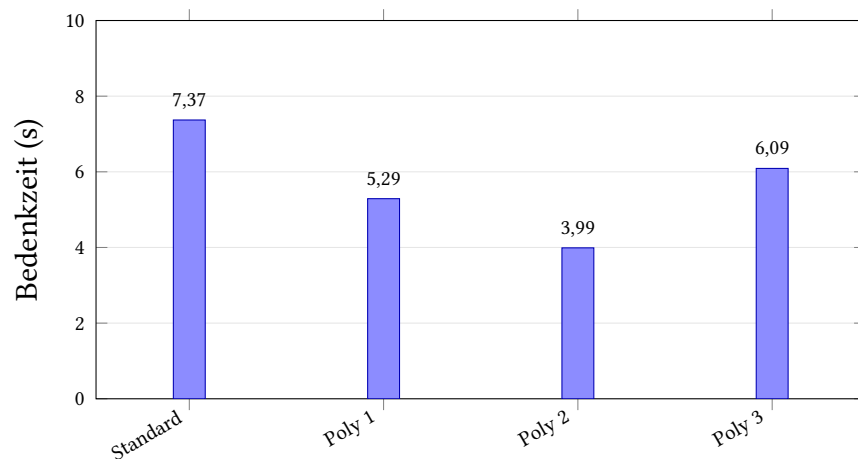


Abbildung 6.3.: Mittlere Bedenkzeit je Zug nach Partie. Sämtliche Mittelwerte beruhen auf $n = 23$. Quelle: Eigene Darstellung.

Die deskriptiven Mittelwerte, die in Abbildung 6.3 visualisiert werden, betrugen 7,37 Sekunden in Partie 1 ($SD = 3,19$; $n = 23$), 5,29 Sekunden in Partie 2 ($SD = 1,87$; $n = 23$), 3,99 Sekunden in Partie 3 ($SD = 1,75$; $n = 23$) und 6,09 Sekunden in Partie 4 ($SD = 2,49$; $n = 23$). Für den gepaarten Vergleich ($n = 23$) betrug die mittlere Bedenkzeit in der Standardbedingung $M = 7,37$ Sekunden ($SD = 3,19$) und über die verfügbaren polymorphen Partien $M = 5,12$ Sekunden ($SD = 1,72$). Die Differenzen waren mit einer Normalverteilung vereinbar (Shapiro-Wilk: $W = 0,980$; $p = 0,900$). Der gepaarte t -Test ergab eine signifikante Verringerung um durchschnittlich 2,24 Sekunden je Zug (95 %-KI [-3,24; -1,25]), $t(22) = -4,682$; $p < 0,001$;

$d_z = -0,98$. Damit zeigt sich ein signifikanter Effekt entgegen der in H2 erwarteten Richtung. H2 wird daher nicht unterstützt.

6.3.2. Zugqualität

Die Zugqualität wurde als Centipawn Loss (CPL) der menschlichen Züge interpretiert. Dabei stehen niedrigere Werte für geringere Bewertungsverluste. In der Evaluationsdatei erzeugten Übergänge zu oder aus mattkodierte Bewertungen 61 Differenzwerte in der Größenordnung von rund 21 Millionen. Da diese Werte durch die schwache Repräsentation von Matt- oder forcierten Mattstellungen in der Engine entstehen und keine regulären, auf zentibauern basierenden (CPL) Stellungsbewertungen darstellen, wurden sie vor der Mittelwertbildung ausgeschlossen. Von 3.571 menschlichen Zügen verblieben 3.510 Züge für die CPL-Auswertung.

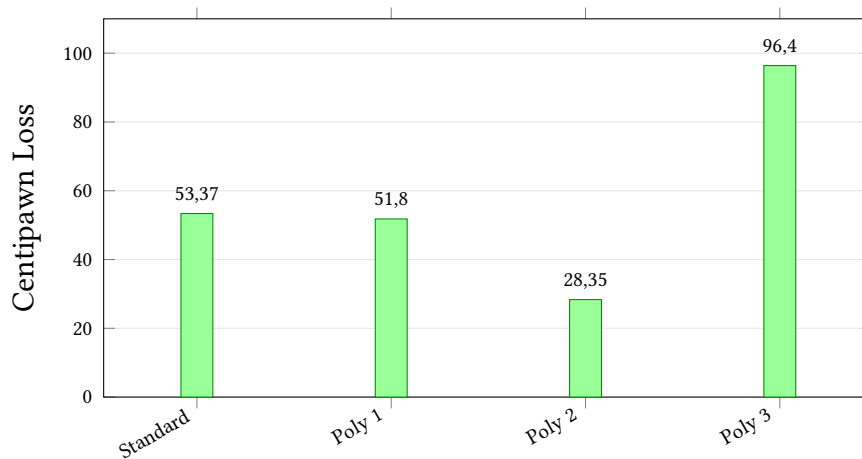


Abbildung 6.4.: Mittlerer Centipawn Loss der menschlichen Züge nach Partie. Niedrigere Werte stehen für geringere Bewertungsverluste. Quelle: Eigene Darstellung.

Die mittleren CPL-Werte, die in Abbildung 6.4 visualisiert werden, betrugen 53,37 in Partie 1 ($SD = 42,67$), 51,80 in Partie 2 ($SD = 37,83$), 28,35 in Partie 3 ($SD = 22,60$) und 96,40 in Partie 4 ($SD = 69,57$). Im gepaarten Datensatz ($n = 23$) lag der mittlere CPL in der Standardbedingung bei $M = 53,37$ ($SD = 42,67$) und im ungewichteten Mittel der verfügbaren polymorphen Partien bei $M = 58,85$ ($SD = 31,67$). Die Differenzen waren mit einer Normalverteilung vereinbar (Shapiro–Wilk: $W = 0,923$; $p = 0,078$). Der Unterschied von 5,49 CPL (95 %-KI $[-12,24; 23,22]$) war nicht signifikant, $t(22) = 0,642$; $p = 0,528$; $d_z = 0,13$. H3 wird daher nicht unterstützt.

6.3.3. Partieausgänge

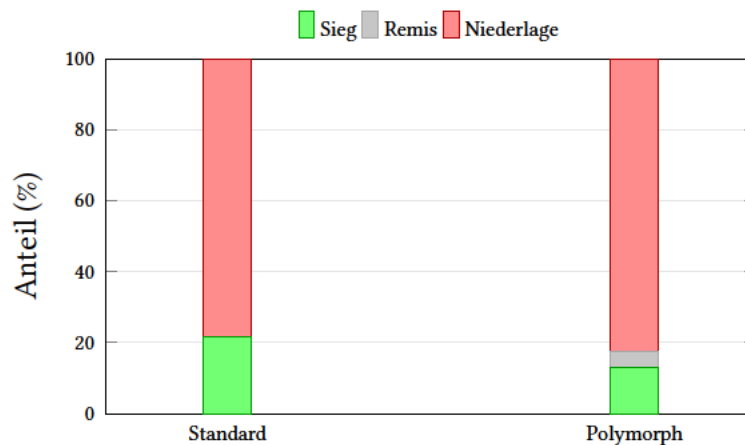


Abbildung 6.5.: Verteilung der Partieausgänge nach Phase. Quelle: Eigene Darstellung.

Abbildung 6.5 zeigt, dass in der Standardpartie die Teilnehmenden fünf von 23 Partien (21,7 %) gewannen. Über die drei polymorphen Bedingungen hinweg gab es neun Siege (13,0 %), drei Remis (4,3 %) und 57 Niederlagen (82,6 %). Wegen der unterschiedlichen Zahl von Partien pro Phase und der wiederholten Beobachtungen werden diese Häufigkeiten hier nicht inferenzstatistisch getestet.

6.4. Nutzerakzeptanz und qualitatives Feedback

6.4.1. Akzeptanzurteile

Der wahrgenommene Spielspaß lag für die Standardpartie bei $M = 3,30$ ($SD = 1,15$) und für die polymorphen Partien bei $M = 3,52$ ($SD = 1,62$). Ein explorativer Wilcoxon-Vorzeichen-Rang-Test ergab keinen systematischen Unterschied, $W = 89,0$; $p = 0,563$; $r_{rb} = 0,16$. Die nahezu identischen Mittelwerte verdecken eine stärkere Polarisierung, die vor allem durch die Veranschaulichung in Abbildung 6.6 zur Geltung kommt. Denn nach den polymorphen Partien: 65,2 % stimmten der Spaßaussage zu, während 30,4 % sie ablehnten; nur eine Person wählte die neutrale Kategorie.

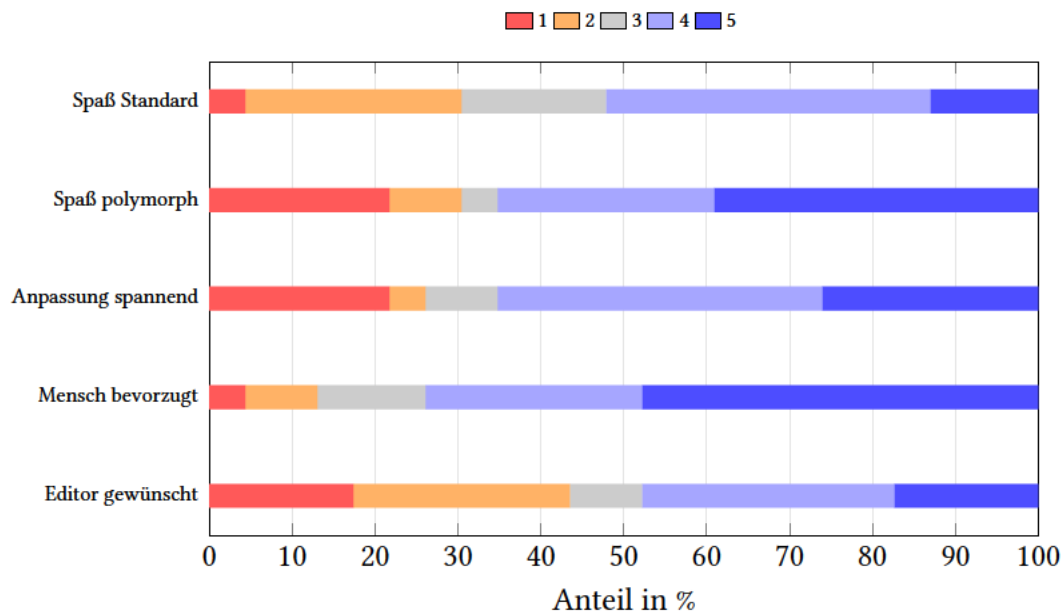


Abbildung 6.6.: Verteilung der fünfstufigen Einzelitems. Quelle: Eigene Darstellung; $n = 23$.

Eine Mehrheit von jeweils 65,2 % empfand sowohl die polymorphen Partien als unterhaltsam als auch die notwendige Anpassung gewohnter Strategien als spannend. Am deutlichsten fiel die Präferenz für einen menschlichen Gegner aus (73,9 % Zustimmung). Der Wunsch nach einem Editor für eigene Brettformen erhielt dagegen keine Mehrheit (47,8 % Zustimmung).

6.4.2. Freitextfeedback

Alle 23 Personen hinterließen eine Freitextantwort. Die Antworten wurden in einer vorläufigen, vereinfachten Inhaltsanalyse mehrfach kodiert. Eine Antwort konnte daher mehreren Kategorien zugeordnet werden. Die Häufigkeiten einzelner Aspekte werden in 6.7 gelistet. Am häufigsten wurden der Verlust gewohnter räumlicher Intuition und die notwendige Anpassung von Strategien angesprochen (14 Personen; 60,9 %). Ebenfalls häufig waren Schwierigkeiten, auf großen oder ungewöhnlich strukturierten Brettern den Überblick über bedrohte Felder zu behalten (11 Personen; 47,8 %). Weitere Kategorien waren Springerbewegungen und das Überspringen inaktiver Felder (8; 34,8 %), Zeitdruck durch die unmittelbar ziehende Engine (7; 30,4 %), positive Neugier oder Abwechslung (7; 30,4 %), technische oder regelbezogene Aspekte (6; 26,1 %) sowie explizite Frustration oder Ablehnung (4; 17,4 %).

Ein prägnanter Kommentar bündelt den wiederkehrenden Intuitionsverlust: „Beim normalen Schach hat man eine Intuition dafür, welche Figur welches Feld deckt. Bei den polymorphen Brettern fällt diese Intuition weg.“ Mehrere Antworten beschrieben zugleich einen eigenständigen Reiz der neuen Geometrien. Daneben wurde wiederholt kritisiert, dass die

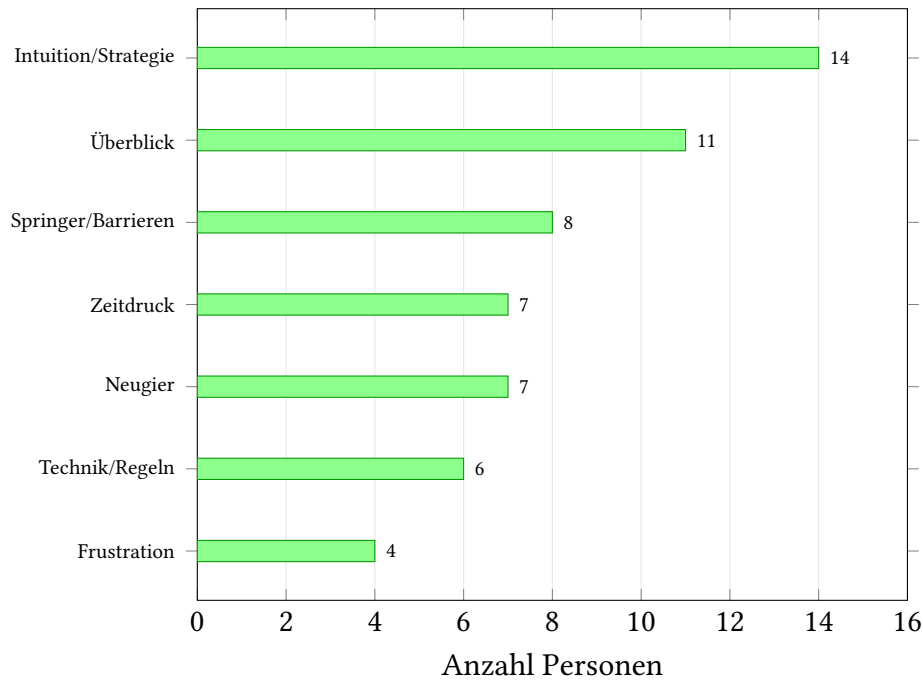


Abbildung 6.7.: Verteilung der meistgenannten Freitextaussagen. Quelle: Eigene Darstellung; $n = 23$.

nahezu unmittelbar ziehende Engine den subjektiven Zeitdruck verstärke. Das qualitative Material zeichnet somit kein einheitlich positives oder negatives Bild, sondern verbindet erhöhte Orientierungsanforderungen mit Neugier und einer deutlich polarisierten Akzeptanz. Das vollständige Kategoriensystem sowie die fallbezogenen Kurzparaphrasen mit den zugeordneten Codes sind in Anhang B dokumentiert (siehe Tabellen B.1 und B.2).

6.5. Zusammenfassung der Ergebnisse

H1 wird unterstützt: Die subjektive kognitive Belastung war in der polymorphen Phase signifikant höher. H2 wird nicht unterstützt, da die beobachtete Bedenkzeit signifikant in die entgegengesetzte Richtung sank. Für H3 zeigte sich kein statistisch belastbarer Unterschied der Zugqualität. Hinsichtlich der explorativen Forschungsfrage blieb der mittlere Spielspaß nahezu unverändert, während die Einzelurteile deutlich auseinanderlagen.

7. Diskussion

Ziel der Nutzerstudie war es, Unterschiede zwischen dem Spielen auf dem konventionellen 8×8 -Standardbrett und auf polymorphen Brettern zu untersuchen. Im Mittelpunkt standen die subjektiv empfundene Belastung, die durchschnittliche Bedenkzeit pro menschlichem Zug, die mit *Uncy* bestimmte Zugqualität sowie der wahrgenommene Spielspaß. Die Ergebnisse zeigen keine einheitliche Erschwerung in allen untersuchten Bereichen. Während die polymorphe Phase zwar als belastender wahrgenommen wurde, sank zugleich die durchschnittliche Bedenkzeit. Für die Zugqualität ergab sich kein statistisch belastbarer Unterschied. Auch der mittlere Spielspaß veränderte sich kaum, wobei eine Polarisierung vorliegt, da die individuellen Bewertungen nach der polymorphen Phase deutlich auseinanderlagen.

Im Folgenden werden die Befunde zunächst im Hinblick auf die drei Hypothesen und die explorative Teilforschungsfrage eingeordnet. Anschließend werden mögliche Konsequenzen für die Weiterentwicklung von *Uncy* abgeleitet und zum Ende die Grenzen der Untersuchung erläutert.

7.1. Interpretation der zentralen Befunde

7.1.1. Subjektive Belastung und vertraute Schachmuster

Die Ergebnisse unterstützen H1. Der mittlere Belastungswert des modifizierten, ungewichteten Raw TLX stieg von $M = 10,92$ in dem Fragebogen nach den Standardbrettern auf $M = 13,43$ nach den Polymorphen. Die mittlere Differenz von 2,50 Punkten war statistisch signifikant, $t(22) = 2,499$, $p = 0,020$. Auch auf Ebene der fünf Teilbereiche lagen die Mittelwerte in der polymorphen Phase durchgehend höher. Die größten deskriptiven Anstiege zeigten sich beim wahrgenommenen Zeitdruck und bei der geistigen Anforderung.

Der Befund ist mit der theoretischen Annahme vereinbar, dass Schachspielende bei der Stellungsbeurteilung auf vertraute räumliche Muster zurückgreifen. Chase und Simon beschreiben, dass sich Schachexpertise unter anderem in der Wahrnehmung bedeutungsvoller Figurenkonstellationen zeigt Chase und Simon (1973, S. 56, 80–81). Werden Brettgröße, Feldanordnung und Figurenkonstellation verändert, hat es dies möglicherweise zur Folge, dass

sich solche bekannten Muster möglicherweise weniger unmittelbar übertragen lassen. Dazu passen die Freitextantworten: 14 der 23 Teilnehmenden beschrieben einen Verlust gewohnter räumlicher Intuition oder die notwendige Anpassung bisheriger Strategien. Elf Personen nannten Schwierigkeiten, bedrohte oder kontrollierte Felder auf den ungewohnten Brettern zu überblicken.

Da die Studie nicht konkret Mustererkennung oder Chunking direkt untersucht hat, kann die erhöhte subjektive Beanspruchung daher nicht eindeutig auf diesen Mechanismus zurückgeführt werden. Denn neben der ungewohnten Geometrie kommen weitere Einflüsse infrage, wie veränderte Figurenbestände und ihre Ausgangsstellungen sowie die spezielle Zeitsituation. Untersucht wurde somit die Wirkung der vollständigen Spielkonfigurationen und nicht die einzelnen geometrischen Eigenschaften und ihre Einflüsse.

Da eine Dimension des ursprünglichen NASA-TLX entfernt und die verbleibenden Items sprachlich angepasst wurden, handelt es sich um einen modifizierten, ungewichteten Raw TLX (Hart & Staveland, 1988, S. 169; Hart, 2006, S. 904–906). Der Wert erfasst eine breitere subjektive Aufgabenbeanspruchung und ist weder ein direktes Maß eines einzelnen kognitiven Prozesses noch mit objektiv schlechterer Spielleistung gleichzusetzen. Unter den Bedingungen der Studie lässt sich dennoch festhalten, dass die polymorphe Phase von den Teilnehmenden insgesamt als anspruchsvoller erlebt wurde.

7.1.2. Kürzere Bedenkzeiten trotz höherer Belastung

H2 wird durch die Daten nicht unterstützt. Entgegen der gerichteten Erwartung sank die durchschnittliche Bedenkzeit von $M = 7,37$ Sekunden in der Standardbedingung auf $M = 5,12$ Sekunden in der polymorphen Phase. Die Verringerung um durchschnittlich 2,24 Sekunden war statistisch signifikant, $t(22) = -4,682$, $p < 0,001$, und ging mit $d_z = -0,98$ einher.

Es sollte daraus allerdings nicht automatisch geschlossen werden, dass die polymorphen Bretter leichter zu verarbeiten waren. Eine solche Deutung wäre mit der höheren subjektiven Belastung und dem qualitativen Feedback nur schwer vereinbar. Ein möglicher Erklärungsansatz ist eine veränderte Zeitstrategie. Unter einem festen Zeitbudget können Entscheidungen beschleunigt werden, indem weniger Varianten geprüft oder niedrigere Entscheidungsschwellen akzeptiert werden. Außerdem zu berücksichtigen wäre in dem Zug, dass den Teilnehmenden pro Partie insgesamt fünf Minuten zur Verfügung standen. Die durchschnittliche Zeit pro Zug hing damit auch von der Zahl der gespielten Züge und vom Umgang mit der verbleibenden Gesamtzeit ab. Da die Engine stets schnell antwortete, entstand während der gegnerischen Züge kaum eine zusätzliche Orientierungsphase. Sieben Personen erwähnten diesen Zeitdruck ausdrücklich im Freitext. Außerdem war die Standardpartie stets die erste Partie und diente auch der Eingewöhnung in die Oberfläche. Kürzere Zeiten in späteren

Partien können deshalb teilweise auf eine zunehmende Vertrautheit mit der Bedienung zurückgehen oder schlichtweg eine tendenzielle Ermüdungserscheinung.

Auch die deskriptiven Werte der einzelnen Partien sprechen gegen eine einfache Erklärung allein durch die Unterscheidung zwischen Standard- und polymorphem Brett. Die mittlere Bedenkzeit sank von der Standardpartie über die ersten beiden polymorphen Partien, stieg in der letzten Partie jedoch wieder an. Mehrere der bis hierhin genannten möglichen Einflussfaktoren könnten sich hierbei überlagert haben. Das Ergebnis widerlegt also die in H2 formulierte Richtung, belegt aber keine grundsätzlich beschleunigte Verarbeitung polymorpher Stellungen.

7.1.3. Zugqualität und Partieausgänge

Für H3 zeigte sich kein statistisch belastbarer Unterschied. Der mittlere Centipawn Loss lag in der Standardbedingung bei $M = 53,37$ und im ungewichteten Mittel der verfügbaren polymorphen Partien bei $M = 58,85$. Die geschätzte Differenz von 5,49 CPL war nicht signifikant, $t(22) = 0,642$, $p = 0,528$, $d_z = 0,13$. Das 95 %-Konfidenzintervall von $-12,24$ bis $23,22$ CPL umfasst sowohl eine mögliche Verbesserung als auch eine Verschlechterung. Dass dieses Ergebnis nicht Signifikant ist, bedeutet nicht, dass es eine Gleichwertigkeit zwischen den Bedingungen des Standardbretts und der polymorphen Bretter gibt. H3 wird nicht unterstützt, eine Beeinträchtigung der Zugqualität ist mit den vorliegenden Daten aber auch nicht sicher ausgeschlossen.

Zwischen den drei polymorphen Partien unterschieden sich die Mittelwerte deutlich. Mit 51,80 CPL lag die erste polymorphe Partie nahe am Standardwert, die zweite lag mit 28,35 CPL darunter und die dritte mit 96,40 CPL deutlich darüber. Diese Heterogenität in den Ergebnissen deutet darauf hin, dass es wesentliche Unterschiede in den Brettern gibt.

Deskriptiv gewannen die Teilnehmenden einen geringeren Anteil der polymorphen Partien als der Standardpartien. Dies kann auf Anpassungsschwierigkeiten hindeuten, ist aber nicht unmittelbar mit dem durchschnittlichen CPL gleichzusetzen. Ein Partieausgang kann von wenigen kritischen Entscheidungen abhängen, die dann gepaart sein können mit einem früheren Partieabbruch, während der CPL Bewertungsverluste über alle auswertbaren menschlichen Züge mittelt.

7.1.4. Spielspaß und weitere Einzelurteile

Die explorative Teilforschungsfrage ergibt ein gemischtes Bild. Der mittlere Spielspaß lag nach der Standardpartie bei $M = 3,30$ und nach den polymorphen Partien bei $M = 3,52$. Der explorative Vergleich ergab keinen systematischen Unterschied. Die nahezu identischen

Mittelwerte verdecken allerdings eine stärkere Polarisierung nach der polymorphen Phase: 65,2 % stimmten der positiven Spaßaussage zu, während 30,4 % sie ablehnten. Nur eine Person wählte die mittlere Kategorie.

Auch andere Teile des Feedbacks zeigen diese Ambivalenz. Einerseits empfanden 65,2 % die notwendige Anpassung gewohnter Strategien als spannend, und sieben Personen beschrieben im Freitext Neugier oder Abwechslung. Andererseits wurden Orientierungsprobleme, Zeitdruck und in vier Antworten auch ausdrückliche Frustration oder Ablehnung genannt. Die höhere Beanspruchung war somit nicht für alle Personen gleichbedeutend mit einem schlechteren Spielerlebnis. Für einen Teil der Stichprobe stellte sie eine interessante Herausforderung dar, für einen anderen Teil eher eine Überforderung. Das im Gameflow Modell beschriebene Verhältnis zwischen Herausforderung und vorhandenen Fähigkeiten bietet einen passenden Interpretationsrahmen für dieses Muster (Sweetser & Wyeth, 2005, S. 3, 5–7). Da die Studie jedoch weder Flow noch eine Übereinstimmung zwischen Anforderung und Fähigkeit mit einer entsprechenden Skala erfasst hat, bleibt dieser Bezug eine theoretische Einordnung.

Die Aussagen zum Spielerleben beruhen auf einzelnen fünfstufigen Items und nicht auf einer validierten Akzeptanz- oder Spielerlebnisskala. Sie erlauben daher eine erste explorative Einschätzung, aber kein umfassendes Urteil über die Akzeptanz des Systems. Aus diesen Gründen lässt sich TFF1 entsprechend vorsichtig beantworten: Im Mittel veränderte sich der berichtete Spielspaß kaum, die polymorphen Konfigurationen riefen jedoch deutlich unterschiedlichere Reaktionen hervor als das Standardbrett.

7.2. Implikationen für die Weiterentwicklung von Uncy

Aus den Ergebnissen lassen sich mehrere Ansatzpunkte für die weitere Entwicklung ableiten. Sie sind als Hinweise aus einer kleinen Gelegenheitsstichprobe und nicht als abschließend validierte Produktanforderungen zu verstehen. Um direktere Aussagen zu treffen, wären weitere Untersuchungen, die spezifischer auf das große Konzept der polymorphen Bretter eingehen, sinnvoll.

Zunächst erscheint eine stärkere Unterstützung der räumlichen Orientierung zumindest erwähnenswert. Da häufig der Überblick über bedrohte Felder, Springerbewegungen und Barrieren durch inaktive Felder genannt wurde, könnte das optionale Hervorheben kontrollierter Felder geprüft werden. Außerdem könnte der Einstieg in neue Geometrien schrittweise gestaltet werden. Kurze Erläuterungen der jeweiligen Brettbesonderheiten oder eine Übungsstellung ohne Zeitdruck könnten den anfänglichen Orientierungsaufwand reduzieren.

Es stimmten 73,9 % der Aussage zu, bei solchen Varianten lieber gegen einen menschlichen Gegner als gegen die Engine zu spielen. Dem Wunsch nach einem Editor für eigene Brettformen stimmten 47,8 % zu. Diese beiden getrennt erhobenen Einzelitems bilden keine direkte Rangfolge von Entwicklungsfunktionen. Sie legen jedoch nahe, einen Mehrspielermodus als möglichen nächsten Prototyp zu untersuchen und einen Bretteditor zunächst als ergänzendes Angebot für eine interessierte Teilgruppe zu betrachten.

7.3. Limitationen der Untersuchung

7.3.1. Stichprobe und Übertragbarkeit

Die ausgewertete Stichprobe umfasste 23 Personen und wurde überwiegend im universitären Umfeld sowie im erweiterten Bekanntenkreis des Autors gewonnen. Sie war vergleichsweise jung und mehrheitlich durch Anfängerinnen und Anfänger sowie Gelegenheitsspielende geprägt. Die Ergebnisse lassen sich deshalb nicht ohne Weiteres auf die allgemeine Schachpopulation oder auf erfahrene Vereinsspielende und Expertinnen beziehungsweise Experten übertragen. Gerade für die angenommene Rolle erlernter Schachmuster wären Unterschiede nach Erfahrungsniveau theoretisch plausibel. Für eine solche Untersuchung sind die jeweiligen Teilgruppen allerdings als zu klein befunden worden.

Von 51 registrierten Studienstarts gingen 23 Durchläufe in die Auswertung ein. Die festgelegten Vollständigkeits- und Plausibilitätskriterien erhöhen die Vergleichbarkeit der analysierten Fälle, können aber mit einer Selektionsverzerrung verbunden sein. Falls insbesondere überforderte, technisch beeinträchtigte oder wenig interessierte Personen die Studie vorzeitig beendet haben, könnten Belastung und Spielerleben in der ausgewerteten Stichprobe verzerrt sein. Gleichzeitig könnte ein Schließen des Tabs und ein neuer Start der Studie der Grund für die hohen Abbruchquoten sein. Da die Abbruchgründe nicht erhoben wurden, lassen sich Richtung und Ausmaß eines solchen Effekts nicht bestimmen.

Die Durchführung als Remotestudie erleichterte die Teilnahme stark, entzog jedoch Kontrollen über Eingabegerät, Netzwerkverbindung, sowie Umgebung und mögliche Ablenkungen. Technische oder darstellungsbezogene Schwierigkeiten können daher nicht vollständig von den Effekten der Spielkonfiguration getrennt werden, wurden allerdings auch nicht im Allgemeinen im Freitextfeedback genannt.

7.3.2. Versuchsdesign und Operationalisierung

Das Within-Subject-Design reduziert den Einfluss stabiler Unterschiede zwischen den Personen. Die Reihenfolge der Bedingungen war jedoch für alle Teilnehmenden gleich: Auf

die Standardpartie folgten stets dieselben drei polymorphen Partien. Damit ist der Bedingungsvergleich mit Übung, Gewöhnung, Ermüdung und möglichen Frustrationseffekten verbunden. Weil die Standardpartie zugleich der Eingewöhnung in die Oberfläche diene, können insbesondere die späteren Bedenkzeiten durch zunehmende Bedienerfahrung beeinflusst worden sein. Eine Randomisierung oder Gegenbalancierung wäre in einer Folgestudie erforderlich, um diese Einflüsse besser zu trennen. Die nicht vorhandene Randomisierung der Reihenfolge wird in der Selbstreflektion im Nachhinein auch als eine der größten Schwächen im Versuchsdesign betrachtet.

Des weiteren waren Baseline- und Interventionsphase nicht symmetrisch, da in der Interventionsphase von jedem Probanden auf drei Brettern gespielt worden ist. Dies hat zur Folge, dass im Abschlussfragebogen alle Bretter gleichzeitig bewertet worden sind. So wie die Auswertung der quantitativen Daten teils verschieden war auf den verschiedenen Brettern, könnte auch die subjektive Wahrnehmung der Teilnehmenden unterschiedlich gewesen sein. Auf einen Fragebogen nach jedem Brett wurde aufgrund von Ermüdungsrisiko während der Studie verzichtet, hätte aber womöglich noch bessere Aufschlüsse ermöglichen können.

Der modifizierte Raw TLX und die Einzelitems wurden an den konkreten Studienkontext angepasst. Das verbessert ihre Verständlichkeit für die Aufgabe, schränkt aber die Vergleichbarkeit mit Untersuchungen ein, die unveränderte und umfassender validierte Instrumente verwenden. Die Einzelitems zu Spielspaß, Strategieanpassung, menschlichen Gegnern und Bretteditor bilden keine gemeinsame Akzeptanzskala, sondern wurden eher genutzt, um Tendenzen bezüglich der Nutzerakzeptanz eines abstrakten Konzepts herauszuarbeiten.

7.3.3. Statistische und qualitative Aussagekraft

Für die Hauptanalysen wurden die drei polymorphen Partien je Person zu einem ungewichteten Mittel ihrer Partiemittelwerte zusammengefasst. Dadurch ging jede Konfiguration unabhängig von Zugzahl und Partiedauer gleich stark ein. Dieses Vorgehen passt zum globalen Vergleich der beiden Phasen, reduziert jedoch die sichtbaren Unterschiede zwischen den einzelnen Brettern.

Die Zugqualität wurde nachträglich durch dieselbe Engine bestimmt, die auch als Spielgegner diente. Die ausgeschlossenen mattkodierte Werte zeigen zusätzlich, dass die Bewertungsskala an terminalen Stellungen nicht wie eine gewöhnliche Centipawn-Skala behandelt werden kann. Außerdem ist die Aussagekraft des CPL zusätzlich dadurch begrenzt, dass die Zugbewertungen mit *Uncy* selbst berechnet wurden, da für die untersuchten nicht-standardisierten Stellungen keine etablierte Referenzengine zur Verfügung stand. Der Evaluationswert der Stellung bleibt jedoch von der Such- und Bewertungsfunktion von *Uncy* abhängig.

Die Freitextantworten lieferten hilfreiche Erklärungsansätze, wurden jedoch im Rahmen einer vereinfachten qualitativen Inhaltsanalyse ausgewertet. Die Kategorien sind daher als strukturierende, explorative Zusammenfassung und nicht als abschließend abgesicherte qualitative Theorie zu verstehen. Zudem geht jede Person unabhängig von Länge und Detailgrad ihrer Antwort mit höchstens einer Nennung pro Kategorie in die Häufigkeitsdarstellung ein.

7.4. Zusammenfassung der Diskussion

Unter den Bedingungen der Studie wurden die polymorphen Spielkonfigurationen subjektiv als belastender erlebt als das Standardbrett. Gleichzeitig trafen die Teilnehmenden ihre Entscheidungen im Mittel schneller, ohne dass sich im aggregierten CPL-Vergleich eine statistisch belastbare Verschlechterung der Zugqualität zeigte. Der durchschnittliche Spielspaß blieb nahezu unverändert, die individuellen Urteile fielen nach der polymorphen Phase jedoch deutlich gegensätzlicher aus. H1 wird damit unterstützt, H2 und H3 werden nicht unterstützt. Die explorative Teilforschungsfrage weist auf ein polarisiertes Spielerleben hin, bei dem neuartige Herausforderung und Orientierungsverlust nebeneinander auftreten.

Die Ergebnisse sprechen somit weder für eine pauschale Überforderung noch für eine durchgehend positive Aufnahme polymorphen Schachs. Sie liefern allerdings erste Hinweise darauf, dass die konkrete Brettgeometrie und der Komplexitätsgrad der untersuchten Bedingungen berücksichtigt werden müssen, was für zukünftige Forschungsvorhaben bedacht werden kann. Aufgrund der kleinen Gelegenheitsstichprobe, der festen Reihenfolge und der eingeschränkten Operationalisierungen sind die Befunde als erste Evaluation des Konzepts und nicht als allgemeingültiger Nachweis zu verstehen.

8. Fazit und Ausblick

8.1. Fazit

Diese Arbeit untersuchte, wie sich grundlegende Verfahren einer Schachengine auf variable und nicht rechteckige Brettgeometrien übertragen lassen und wie Spielende auf solche Veränderungen reagieren. Dafür wurde mit *Uncy* eine Engine entwickelt, die unterschiedliche Brettgrößen sowie inaktive Felder zur Laufzeit verarbeitet. Ergänzt wurde sie durch eine Webanwendung, die zugleich als Spieloberfläche und als Erhebungsinstrument der Nutzerstudie diente.

Die technische Umsetzung zeigt, dass zentrale Verfahren des Computerschachs nicht grundsätzlich an das konventionelle 8×8 -Brett gebunden sind. Notwendig waren vor allem Anpassungen der Zustandsrepräsentation, der Zuggenerierung und der Stellungsbewertung. Mithilfe einer erweiterten FEN-Notation können polymorphe Stellungen beschrieben werden, während dimensionsabhängige Lookup-Tabellen die Bewegungsmöglichkeiten der Figuren an die jeweils eingelesene Geometrie anpassen. Darauf aufbauend konnten etablierte Verfahren wie Alpha-Beta-Suche, iterative Tiefensuche, Zobrist-Hashing und Transpositionstabellen weiterverwendet werden.

Die PERFT-Vergleiche mit Fairy-Stockfish ergaben für die drei geprüften Brettkonfigurationen übereinstimmende Knotenzahlen. Auch die gemessenen Suchzeiten lagen auf den Studienanfangsstellungen deutlich unter dem angestrebten Grenzwert von einer Sekunde. Damit konnte für den untersuchten Rahmen gezeigt werden, dass *Uncy* regelkonforme Züge erzeugt und schnell genug reagiert, um als interaktiver Spielgegner eingesetzt zu werden. Daraus folgt jedoch weder eine vollständige Korrektheitsgarantie für sämtliche denkbaren Geometrien noch eine allgemeine Aussage über die Spielstärke oder Skalierbarkeit der Engine. Der technische Beitrag der Arbeit liegt daher weniger in der Entwicklung eines neuen Suchverfahrens als in der Übertragung bestehender Verfahren auf eine dynamisch einlesbare Brettstruktur.

Die Nutzerstudie ergab kein einheitliches Bild einer allgemeinen Erschwerung. Die subjektiv empfundene Belastung stieg auf den polymorphen Brettern signifikant an. Der Mittelwert des modifizierten, ungewichteten Raw TLX erhöhte sich von 10,92 auf 13,43, wodurch H1 unterstützt wird. Besonders deutlich nahmen die wahrgenommene geistige Anforderung

und der Zeitdruck zu. Auch die Freitextantworten sprechen dafür, dass die ungewohnten Geometrien den Zugriff auf vertraute räumliche Vorstellungen erschwerten. Häufig genannt wurden der Verlust gewohnter Intuition, Probleme beim Überblick über bedrohte Felder und die notwendige Anpassung bisheriger Strategien.

Diese höhere Belastung spiegelte sich allerdings nicht in allen Verhaltensmaßen wider. Entgegen H2 sank die durchschnittliche Bedenkzeit von 7,37 auf 5,12 Sekunden pro Zug. Für die anhand des Centipawn Loss bestimmte Zugqualität zeigte sich ebenfalls kein statistisch belastbarer Unterschied, sodass auch H3 nicht unterstützt wird. Das bedeutet nicht, dass polymorphe Stellungen gleich gut oder grundsätzlich schneller verarbeitet wurden. Die kürzeren Bedenkzeiten können ebenso auf eine veränderte Zeitstrategie, zunehmende Vertrautheit mit der Oberfläche oder Ermüdung zurückgehen. Aufgrund der festen Reihenfolge der Partien lassen sich diese Einflüsse nicht eindeutig voneinander trennen. Ebenso kann aus dem nicht signifikanten Unterschied der Zugqualität keine Gleichwertigkeit der Bedingungen abgeleitet werden.

Beim Spielerleben zeigte sich eine ähnliche Ambivalenz. Der durchschnittliche Spielspaß blieb nahezu unverändert, die einzelnen Bewertungen fielen nach den polymorphen Partien jedoch deutlich gegensätzlicher aus. Einige Teilnehmende empfanden die notwendige Umstellung als interessante Abwechslung, während andere sie als frustrierend oder überfordernd wahrnahmen. Eine höhere Beanspruchung war somit nicht automatisch mit einem schlechteren Spielerlebnis verbunden.

Die Forschungsfrage lässt sich daher dahingehend beantworten, dass polymorphe Brettgeometrien unter den Bedingungen dieser Studie vor allem die subjektive Wahrnehmung des Spiels veränderten. Sie erhöhten die empfundene Belastung und führten zu stärker auseinanderliegenden Bewertungen des Spielerlebnisses. Ein entsprechender Rückgang der gemessenen Zugqualität konnte dagegen nicht nachgewiesen werden, und die Entscheidungen wurden im Mittel sogar schneller getroffen. Die Wirkung polymorphen Schachs lässt sich somit nicht auf eine einfache Erschwerung reduzieren. Sie entsteht vielmehr aus dem Zusammenspiel von Brettgeometrie, Ausgangsstellung, Zeitbedingungen, Spielerfahrung und Gestaltung der Oberfläche.

Diese Einordnung ist vor dem Hintergrund der begrenzten Stichprobe und des Versuchsdesigns zu betrachten. Die ausgewerteten 23 Personen bildeten eine Gelegenheitsstichprobe, und alle absolvierten die Partien in derselben Reihenfolge. Zudem unterschieden sich die polymorphen Stellungen nicht nur in ihrer Geometrie, sondern auch im Figurenbestand und in der Ausgangsstellung. Die Ergebnisse sind deshalb als erste empirische Hinweise und nicht als allgemeingültiger Nachweis über die Wirkung veränderter Brettgeometrien zu verstehen.

8.2. Ausblick

Für die weitere technische Entwicklung bietet sich zunächst eine breitere Validierung von *Uncy* an. Weitere Brettgrößen, Anordnungen inaktiver Felder sowie Mittelspiel- und Endspielstellungen könnten zeigen, wie sich Laufzeit, Speicherbedarf und Verzweigungsfaktor unter unterschiedlichen Bedingungen entwickeln. Gleichzeitig könnte die noch vergleichsweise einfache Stellungsbewertung erweitert werden, um strategische Eigenschaften wie Königsicherheit, Bauernstrukturen und kontrollierte Räume stärker an die jeweilige Geometrie anzupassen.

Da eine deutliche Mehrheit Interesse an Partien gegen menschliche Gegner äußerte, erscheint auch ein Mehrspielermodus als sinnvolle Weiterentwicklung. Ein Bretteditor könnte ergänzend dazu dienen, eigene Geometrien zu entwerfen und das Konzept über die in dieser Arbeit verwendeten Konfigurationen hinaus zu erproben.

Eine Folgestudie sollte eine größere und stärker nach Schacherfahrung differenzierte Stichprobe untersuchen. Durch eine randomisierte oder gegenbalancierte Reihenfolge ließen sich Geometrieeffekte besser von Übung und Ermüdung trennen. Ebenso wäre es sinnvoll, die einzelnen Bretter getrennt zu bewerten und Figurenbestand sowie Ausgangsstellung stärker zu kontrollieren. Dadurch könnte genauer bestimmt werden, welche geometrischen Eigenschaften eine interessante Herausforderung darstellen und ab welchem Punkt sie vorwiegend zu Orientierungsproblemen führen.

Uncy stellt damit eine Grundlage bereit, polymorphe Formen des Schachs technisch umzusetzen und systematisch zu untersuchen. Die Arbeit zeigt, dass eine flexible Schachengine realisierbar ist und dass nicht standardisierte Brettgeometrien ein eigenständiges, jedoch unterschiedlich wahrgenommenes Spielerlebnis erzeugen können. Wie solche Geometrien gestaltet werden müssen, damit Herausforderung, Verständlichkeit und Spielspaß in einem ausgewogenen Verhältnis stehen, bleibt eine weiterführende Forschungsfrage.

Literatur

- Bilalić, M., Graf, M., & Vaci, N. (2026). Computers and chess masters: The role of AI in transforming elite human performance. *British Journal of Psychology*, 117(2), 585–609. <https://doi.org/10.1111/bjop.12750>
- Breuker, D. M., Uiterwijk, J. W. H. M., & van den Herik, H. J. (1996). Replacement Schemes and Two-Level Tables. *ICCA Journal*, 19(3), 175–180. <https://doi.org/10.3233/ICG-1996-19304>
- Campbell, M., Hoane, A. J., Jr., & Hsu, F.-h. (2002). Deep Blue. *Artificial Intelligence*, 134(1–2), 57–83. [https://doi.org/10.1016/S0004-3702\(01\)00129-1](https://doi.org/10.1016/S0004-3702(01)00129-1)
- Chase, W. G., & Simon, H. A. (1973). Perception in Chess. *Cognitive Psychology*, 4(1), 55–81. [https://doi.org/10.1016/0010-0285\(73\)90004-2](https://doi.org/10.1016/0010-0285(73)90004-2)
- Chessprogramming Wiki. (n. d.). *Late move reductions*. Verfügbar 5. Juli 2026 unter https://www.chessprogramming.org/Late_Move_Reductions
- David-Tabibi, O., & Netanyahu, N. S. (2002). Verified Null-Move Pruning. *ICGA Journal*, 153–161. [arXiv:0808.1125v1](https://arxiv.org/abs/0808.1125v1)
- Edwards, S. J. (1994, 12. März). *Portable Game Notation Specification and Implementation Guide*. Verfügbar 1. August 2026 unter <https://www.saremba.de/chessgml/standards/pgn/pgn-complete.htm#c16.1>
- Fichter, F., & Fairy-Stockfish Contributors. (n. d.). *Fairy-Stockfish: Chess Variant Engine* (Version 230726 LB). Verfügbar 1. August 2026 unter <https://github.com/fairy-stockfish/Fairy-Stockfish>
- FIDE. (2023, 1. Januar). *FIDE Laws of Chess Taking Effect from 1 January 2023*. Verfügbar 15. Juli 2026 unter <https://handbook.fide.com/chapter/E012023>
- Genesereth, M., Love, N., & Pell, B. (2005). General Game Playing: Overview of the AAAI Competition. *AI Magazine*, 26(2), 62–72. https://www.researchgate.net/publication/220604585_General_Game_Playing_Overview_of_the_AAAI_Competition
- Gobet, F., & Simon, H. A. (1996a). The Roles of Recognition Processes and Look-Ahead Search in Time-Constrained Expert Problem Solving: Evidence from Grand-Master-Level Chess. *Psychological Science*, 7(1), 52–55. <https://doi.org/10.1111/j.1467-9280.1996.tb00666.x>
- Gobet, F., & Simon, H. A. (1996b). Templates in Chess Memory: A Mechanism for Recalling Several Boards. *Cognitive Psychology*, 31(1), 1–40. <https://doi.org/10.1006/cogp.1996.0011>

- Hart, S. G. (2006). NASA-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(9), 904–908. <https://doi.org/10.1177/154193120605000909>
- Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In P. A. Hancock & N. Meshkati (Hrsg.), *Human Mental Workload* (S. 139–183, Bd. 52). North-Holland. [https://doi.org/10.1016/S0166-4115\(08\)62386-9](https://doi.org/10.1016/S0166-4115(08)62386-9)
- Heitz, R. P. (2014). The Speed-Accuracy Tradeoff: History, Physiology, Methodology, and Behavior. *Frontiers in Neuroscience*, 8, Artikel 150. <https://doi.org/10.3389/fnins.2014.00150>
- Knuth, D. E., & Moore, R. W. (1975). An Analysis of Alpha-Beta Pruning. *Artificial Intelligence*, 6(4), 293–326. [https://doi.org/10.1016/0004-3702\(75\)90019-3](https://doi.org/10.1016/0004-3702(75)90019-3)
- Korf, R. E. (1985). Depth-First Iterative-Deepening: An Optimal Admissible Tree Search. *Artificial Intelligence*, 27(1), 97–109. [https://doi.org/10.1016/0004-3702\(85\)90084-0](https://doi.org/10.1016/0004-3702(85)90084-0)
- Mann, C. (2019). *Schach: Die Welt auf 64 Feldern* (1. Aufl., Bd. 2899). C.H.BECK Literatur - Sachbuch - Wissenschaft.
- Marsland, T. (1986). A Review of Game-Tree Pruning1. *ICGA Journal*, 9(1), 3–19. <https://doi.org/10.3233/ICG-1986-9102>
- Polgar, S. (2025, 8. März). *History of Chess960*. International Chess Federation (FIDE). Verfügbar 2. August 2026 unter <https://www.fide.com/history-of-chess960/>
- Pritchard, D. B. (2007). *The Classified Encyclopedia of Chess Variants* (J. Beasley, Hrsg.; 2. Aufl.). John Beasley.
- Shannon, C. E. (1950). XXII. Programming a Computer for Playing Chess. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41(314), 256–275. <https://doi.org/10.1080/14786445008521796>
- Stockfish Developers. (2026, 31. Januar). *Stockfish* (Version 18) [Release sf_18, Commit cb3d4ee]. Verfügbar 2. August 2026 unter https://github.com/official-stockfish/Stockfish/tree/sf_18
- Sweetser, P., & Wyeth, P. (2005). GameFlow: A Model for Evaluating Player Enjoyment in Games. *Computers in Entertainment*, 3(3), Artikel 3. <https://doi.org/10.1145/1077246.1077253>
- Vučković, V. (2008). The Compact Chessboard Representation. *ICGA Journal*, 31(3), 157–164. <https://doi.org/10.3233/ICG-2008-31304>
- Zobrist, A. L. (1990). A New Hashing Method with Application for Game Playing [Nachdruck des Technical Report No. 88, University of Wisconsin, 1970]. *ICCA Journal*, 13(2), 69–73. <https://doi.org/10.3233/ICG-1990-13203>

A. Erweiterte PERFT Vergleiche

Tabelle A.1.: Gegenüberstellung der PERFT-Knotenzahlen von Uncy und Fairy-Stockfish fuer 20 zufällig aus den Studiendaten ausgewählte Stellungen bei Tiefe 5

Nr.	FEN	Uncy	Fairy-Stockfish	Differenz
1	r3kb1r/p1p2ppp/1p2b3/2PNP3/3B1P2/5N2/PP4PP/R3K2R w KQkq - 1 1	37840949	37840949	0
2	4kb1r/2p1p2p/1pN1p3/6B1/1p2p3/2n1NnP1/P4PKP/r6R b k - 2 1	28117569	28117569	0
3	B4b1r/3k1pp1/5n2/2p1Pb1p/p1P4P/1P6/P3N1R1/R1BK4 b - - 0 1	21608194	21608194	0
4	4q2r/3kp1b1/1Bp1b1p1/3r4/1Q4p1/8/8/1KR5 w - - 0 1	70899516	70899516	0
5	3q3r/3kppb1/4bn2/1N1p4/B1pP3B/2P1PN2/2Q2PPP/5RK1 b - - 1 1	27660288	27660288	0
6	r7/2p1k1p1/2p5/p2qp2Q/1p1b4/1P1P2P1/P1P4P/3K4 w - - 1 1	14934868	14934868	0
7	r2q1b1r/2pkpppp/2n2n2/pp1p1b2/5P2/PP3NP1/2PPP2P/RNBQKB1R w KQ - 4 1	19493209	19493209	0
8	4kr2/2p1p3/1pN1p2N/7p/7P/2n3P1/5p2/3n1K2 b - - 1 1	1538230	1538230	0
9	r6r/4pkb1/2p1b1p1/1p1pPp1p/P2q1P2/1P2NQ2/1P4PP/2R2RK1 w - - 0 1	59850380	59850380	0
10	r1b2b2/2ppkp2/2n2Np1/pp2Pp2/3PQB1q/2P5/PP3P2/R3KB1r b Q - 3 1	64589177	64589177	0
11	r1bqkb1r/1pp1pppp/2P2n2/pB6/8/2N2p2/PPP2PPP/R1BQK2R b KQkq - 0 1	46266695	46266695	0
12	r1bqkb1r/pppp1ppp/1nn1p3/4P3/2PP4/2N2N2/PP3PPP/R1BQKB1R b KQkq - 1 1	44079714	44079714	0
13	4kb1r/2p1p2p/1pN1p3/6n1/1p6/2n1NpP1/P6P/7K w k - 0 1	4020107	4020107	0
14	8/3k1p2/8/7p/2r1K2P/2b5/b7/8 w - - 2 1	86991	86991	0
15	r3kb1r/ppp1pppp/4b3/4P3/3P4/1Qn2N2/1P3PPP/R1B1K2R w KQkq - 0 1	63967948	63967948	0
16	5b1r/2p1p1p1/3qpn2/4k2p/1BQ1P2P/5R2/5PP1/r3K3 w - - 1 1	1749904	1749904	0
17	r2q1b1r/1p1k1ppp/2p2n2/p2p1b2/Q2P4/N1PnP1P1/PP2RPPK/R1B5 b - - 2 1	72401088	72401088	0
18	r3kb1r/4pppp/p2qbn2/3pN3/3p1P2/p2P4/1P2N1PP/R1BQ1RK1 w kq - 0 1	42330436	42330436	0
19	r1bqkb1r/1pp1pppp/2n2n2/pB1p4/3P1B2/4PN2/PPP2PPP/RN1QK2R b KQkq - 1 1	32555877	32555877	0
20	r3k2r/2p2pp1/2pb4/p2qpb1p/1p2N3/1P1PBNP1/P1PQ3P/2KR1R2 b kq - 1 1	96852377	96852377	0

B. Materialien zur qualitativen Inhaltsanalyse

B.1. Kategoriensystem

Tabelle B.1.: Kategoriensystem der vereinfachten qualitativen Inhaltsanalyse

Code	Kategorie	Kodierregel	Ankerbeispiel
A	Verlust gewohnter Intuition und Strategieanpassung	Aussagen über nicht mehr übertragbare Schachmuster, Eröffnungen, Figurenwerte oder die notwendige Entwicklung neuer Strategien.	P25: „Bei den polymorphen Brettern fällt diese Intuition weg.“
B	Überblick über Felder und Bedrohungen	Schwierigkeiten, auf großen oder ungewöhnlich strukturierten Brettern kontrollierte, bedrohte oder unsichere Felder zu erkennen.	P03: „Es ist schwerer einen Überblick zu behalten.“
C	Springerbewegungen und Barrieren	Aussagen zur großen Zahl von Springern, zu ihrer Bewegungslogik oder zum Überspringen inaktiver Felder.	P22: „Überraschend war, dass man beim Pferderennen über die Mitte konnte.“
D	Zeitdruck durch die Engine	Wahrgenommener Zeitdruck, der ausdrücklich mit der unmittelbar oder sehr schnell ziehenden Engine beziehungsweise der Zeitbegrenzung verknüpft wird.	P23: „Keine Zeit zum Nachdenken, da die KI instant setzt.“
E	Positive Neugier und Abwechslung	Positive Aussagen über den Reiz, die Abwechslung oder interessante neue Möglichkeiten der Brettvarianten.	P02: „Die Abwechslung gegenüber dem klassischen Spiel ist hoch und macht Spaß.“
F	Technische oder regelbezogene Aspekte	Hinweise auf Darstellung, Engineverhalten oder als fehlend beziehungsweise unerwartet wahrgenommene Regeln.	P11: „Kein Unentschieden bei Zugwiederholungen.“
G	Explizite Frustration oder Ablehnung	Eindeutig negative Gesamturteile, ausdrücklich benannte Frustration oder fehlender Spielspaß.	P18: „Ich fand es furchtbar frustrierend.“

Eine Antwort konnte mehreren Kategorien zugeordnet werden. Quelle: Eigene Auswertung.

B.2. Fallbezogene Kurzparaphrasen

Tabelle B.2.: Kurzparaphrasen des pseudonymisierten Freitextfeedbacks

Fall	Kurzparaphrase	Codes
P01	Viele Springer und ihre Bewegungen waren unter Zeitdruck schwer vorausszusehen; die als schwächer wahrgenommene Engine erhöhte Hoffnung und Spaß.	A, C, D, E
P02	Veränderte Geometrie und Brettgröße erschwerten den Überblick und führten zu unerwarteten Figurenverlusten; die Abwechslung wurde positiv bewertet.	A, B, C, E
P03	Auf den großen Brettern wurden mehr schlechte Züge gemacht, weil der Überblick schwerer zu behalten war.	A, B
P04	Die große Zahl von Springern wurde als unangenehm und schwierig erlebt.	A, C
P05	Figurenwerte und unsichere Felder waren schwer einzuschätzen; Springer über Barrieren wurden als interessant erlebt; die schnell ziehende Engine erzeugte Zeitdruck.	A, B, C, D, E
P06	Der Überblick auf großen Brettern und die Anpassung der aus normalem Schach gewohnten Züge fielen schwer.	A, B
P07	Die Zeit reichte nicht aus, um die vielen neuen Möglichkeiten auszuprobieren.	A, D, E
P08	Figurenbewegungen und Bedrohungen wurden wiederholt übersehen, besonders bei ungewöhnlich vielen gleichen Figuren.	A, B
P09	Die größte Herausforderung bestand darin, beide Brettseiten gleichzeitig im Blick zu behalten.	B
P10	Gewohnte Situationen und Eröffnungen waren nicht nutzbar; die sehr schnell ziehende Engine verstärkte den Zeitdruck.	A, D, F
P11	Genannt wurden fehlende Zugwiederholungsremis, große Bretter, Zeitbegrenzung, Springer über der Mitte und eine ungewohnte Figurenzusammensetzung.	A, B, C, D, F
P12	Eine Variante mit einer Barriere in der Mitte wurde sogar als einfacher empfunden.	E
P13	Die eigene Anpassung an bekannte, aber veränderte Muster und das Festhalten an gewohnten Zügen wurden als interessant beschrieben.	A, E
P14	Die Antwort enthielt nur ein positives Emoticon und keine inhaltlich codierbare Aussage.	–

Fortsetzung auf der nächsten Seite

Fall	Kurzparaphrase	Codes
P16	Gleich aussehende Figurenfarben führten zu Verwechslungen und Figurenverlusten.	F
P17	Ausführliche Reflexion über Strategie, Feldfarbenkontrolle, Springerbarrieren, Spielreiz, Engineverhalten und den durch die sofortigen Enginezüge entstehenden Zeitdruck.	A, B, C, D, E, F
P18	Die Spiele wurden ausdrücklich als stark frustrierend bewertet.	G
P19	Die Brettmodifikation wurde wegen der Größe und geringer wahrgenommener Ungleichgewichte als nicht interessanter bewertet.	G
P20	Auf großen Brettern waren die von Läufern kontrollierten Felder schwer zu erkennen.	B
P21	Eigene geringe Spielstärke, wiederholtes Aufgeben und Frustration wurden ausdrücklich genannt.	G
P22	Es bestanden Unsicherheiten bei der Entwicklung der Figuren und beim Berechnen starker Figuren; das Überspringen der Mitte durch Springer überraschte.	A, B, C
P23	Die unmittelbar ziehende Engine erzeugte starken Zeitdruck und nahm den Spielspaß; außerdem wurde wiederholtes Engineverhalten kritisiert.	D, F, G
P25	Gewohnte Intuition für gedeckte Felder fiel weg; Läuferkontrolle und das Überspringen einer Begrenzung durch einen Springer wurden übersehen.	A, B, C, F

C. Verzeichnis des elektronischen Anhangs

Der elektronische Anhang befindet sich auf dem beigefügten Datenträger und besitzt folgende Verzeichnisstruktur:

01_Bachelorarbeit/ Finale Fassung der Bachelorarbeit als PDF.

02_Quellcode/ Quellcode von Uncy einschließlich Projekt- und Abhängigkeitsdateien.

03_Reproduktion/ Anleitungen zur Bedienung, um Softwareergebnisse zu reproduzieren.

04_Studiendaten/ JASP-Datei, vorbereinigte & bereinigte Datensätze

D. Kurzanleitungen

D.1. Ausführung von Uncy.

Uncy benötigt das .NET 9 SDK. Für einen Perft-Test werden die gewünschte FEN-Stellung und Suchtiefe in `Uncy.Engine/Program.cs` eingetragen; anschließend wird das Konsolenprojekt ausgeführt. Die Weboberfläche ist bereits in `Uncy.Server` enthalten und wird gemeinsam mit dem Server gestartet:

```
dotnet run --project Uncy.Engine -c Release
dotnet run --project Uncy.Server --launch-profile http
```

Die Perft-Knotenzahlen und Laufzeiten erscheinen in der Konsole. Nach dem Start des Servers ist die Weboberfläche unter `http://localhost:5000` erreichbar.

D.2. Ausführung von Fairy-Stockfish

Fairy-Stockfish ist eine Schachengine für benutzerdefinierte Varianten. Die Software steht unter der GNU GPLv3. Lizenz und zugehöriger Quellcode befinden sich im elektronischen Anhang.

Die hinterlegte .exe starten und anschließend diese Befehle einzeln eingeben:

```
uci
```

Lade die `variants.ini` Datei ein. Muss im Optimalfall nur einmal getan werden. Ist im Verzeichnis vorhanden.

```
setoption name VariantPath value variants.ini
```

Wähle die Brettgeometrie aus. In diesem Fall „standard“.

```
setoption name UCI_Variant value standard
```

Laden der konventionellen 8x8 Stellung

```
position fen rnbqkbnr/pppppppp/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1
```

Der Befehl `d` zeigt das Brett in der Konsole an. Hier einmal überprüfen, ob es korrekt aussieht
`d`

Perft-Test bis Tiefe 5:

`go perft 5`

Eigenständigkeitserklärung

Hiermit versichere ich, Lennard Burrmann, dass ich die vorliegende Bachelorarbeit mit dem Titel

**Entwicklung einer flexiblen Schach-Engine zur Unterstützung
nicht-standardisierter Brettkonfigurationen mit Untersuchung der
Nutzerakzeptanz**

selbstständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die ich wörtlich aus der Literatur oder aus anderen Quellen wie z. B. Internetseiten übernommen habe, habe ich deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.

Hamburg, 03. August 2026