

# Handbuch

## Hammerschach-Analyzer

*PGN-Verwaltung, Bedienung, Technik und Engine-Analyse*





# Inhaltsverzeichnis

1. Einleitung
2. Bedienung
3. Technik
4. Resümee

## Hinweis zur Datei

Dieses Dokument ist als DOCX erstellt, damit es direkt in Apple Pages geöffnet und dort bei Bedarf als natives Pages-Dokument gesichert werden kann. Alle Formatvorlagen verwenden Times New Roman.

## 1. Einleitung

Der Hammerschach-Analyzer ist eine browserbasierte Analyse- und Verwaltungsoberfläche für Schachpartien im PGN-Format. Er richtet sich an Spieler, Trainer, Vereinsmitglieder und Turnierorganisatoren, die Partien erfassen, nachspielen, importieren, exportieren und mit einer Schach-Engine untersuchen möchten.

Im Mittelpunkt stehen drei Aufgaben:

1. **Partien verwalten:** Einzelne oder mehrere Partien werden mit klassischen PGN-Kopfdaten wie Event, Site, Date, Round, White, Black, Result, ECO, WhiteElo, BlackElo und PlyCount geführt.
2. **Partien am Brett nachspielen und eingeben:** Über ein grafisches Schachbrett werden Züge per Mausklick eingegeben. Der Analyzer prüft die Zuglegalität und berücksichtigt Sonderregeln wie Rochade, en passant und Bauernumwandlung.
3. **Stellungen mit Engine-Unterstützung analysieren:** Die integrierte Engine-Analyse arbeitet über Stockfish-Dateien und zeigt Bewertung, WDL-Werte, Tiefe, NPS, Bestzug und Hauptvarianten.

Die Anwendung ist als HTML-, CSS- und JavaScript-Anwendung aufgebaut und läuft direkt im Browser. Für die Engine-Funktion werden zusätzlich die Dateien stockfish.js und stockfish.wasm benötigt. Diese Dateien müssen im gleichen Veröffentlichungsumfeld erreichbar sein, damit die Engine als Web Worker gestartet werden kann.

### 1.1 Zielgruppe

- Vereinskrieger, die eigene Partien schnell nachspielen und prüfen möchten.
- Trainer, die kritische Stellungen mit Engine-Varianten besprechen möchten.
- Turnierorganisatoren, die mehrere Partien in einer PGN-Sammlung verwalten möchten.
- Schachinteressierte, die ohne separate Desktopdatenbank Partien importieren und exportieren möchten.

### 1.2 Grundidee des Workflows

Der typische Arbeitsablauf lautet: PGN laden oder Partie eingeben, Stellung navigieren, Engine-Analyse auswerten, Korrekturen vornehmen und anschließend als PGN exportieren. Da die Anwendung im Browser arbeitet, ist der Export ein wichtiger Schritt zur dauerhaften Sicherung der Arbeit.

## Merksatz

Importieren, prüfen, analysieren, korrigieren und anschließend konsequent exportieren.

## 2. Bedienung

### 2.1 Aufbau der Oberfläche

Die Oberfläche gliedert sich in vier Hauptbereiche: Kopfdatenbereich, Brettbereich, Steuerungsbereich sowie Analyse- und Notationsbereich. Auf kleineren Bildschirmen passt sich die Anordnung automatisch an. Die Seitenleiste wird dann unter das Brett verschoben und die Kopfdaten werden in weniger Spalten dargestellt.

#### Hauptbereiche der Oberfläche

Kopfdatenbereich	Logo und PGN-Felder wie Event, Site, Date, Round,
------------------	---



Hauptbereiche der Oberfläche	
	White, Black, Result und Elo-Angaben.
<b>Brettbereich</b>	Grafisches Schachbrett mit Figuren, Koordinaten, Markierungen und illegalem Zugfeedback.
<b>Steuerungsbereich</b>	Schaltflächen für PGN-Import, Partieliste, Navigation, Rückgängig, Flip und Speichern.
<b>Analysebereich</b>	Engine-Status, Suchoptionen, Bewertung, WDL, Tiefe, NPS, Bestzug und Hauptvarianten.
<b>Zugliste</b>	SAN-Notation der Partie mit Sprungfunktion zu jeder Stellung nach einem Zug.

## 2.2 Kopfdaten einer Partie

Die Kopfdaten werden im oberen Bereich bearbeitet. Sie werden beim Export in die PGN-Datei geschrieben und sollten vor dem Speichern geprüft werden.

Feld	Bedeutung	Hinweis
Event	Name des Turniers, Matches oder Trainingsanlasses	Freitext
Site	Spielort	Freitext
Date	Datum der Partie	Üblich ist JJJJ.MM.TT
Round	Runde	Freitext oder Zahl
White / Black	Spielernamen	Für Export und Partieliste wichtig
Result	Ergebnis	1-0, 0-1, 1/2-1/2 oder *
ECO	Eröffnungscodex	Optional
WhiteElo / BlackElo	Elo-Zahlen	Optional
PlyCount	Anzahl der Halbzüge	Wird automatisch geführt

## 2.3 Neue Partie anlegen

- Auf „**+** **Partie hinzufügen**“ klicken.
- Der aktuelle Stand der bisherigen Partie wird intern gesichert.
- Eine neue leere Partie mit Startstellung und Standard-Kopfdaten wird geöffnet.
- Kopfdaten ergänzen und Züge am Brett eingeben.

Eine neue Partie startet immer mit der normalen Schachanfängsstellung. Die Standardwerte in den Kopfdaten lauten beispielsweise „?“ , „????.??.??“ und „\*“.

## 2.4 PGN-Datei öffnen

Mit „**📁 PGN öffnen**“ wird eine lokale PGN- oder Textdatei geladen. Beim Öffnen wird die aktuelle Sammlung überschrieben. Deshalb erscheint vor dem Dateidialog eine Sicherheitsabfrage.


- Auf „**📁 PGN öffnen**“ klicken.
- Den Hinweis bestätigen, dass die aktuelle Sammlung überschrieben wird.
- Datei auswählen.
- Die importierten Partien werden geladen; die erste Partie wird angezeigt.

Der Import kann mehrere Partien aus einer Datei erkennen. Die Trennung erfolgt anhand neuer Event-Blöcke in der PGN-Struktur.



## 2.5 PGN-Text einfügen


Mit „ **PGN einfügen**“ wird ein Dialog geöffnet, in den PGN-Text direkt kopiert werden kann. Diese Funktion ist nützlich für Partien aus Webseiten, Schachservern, Datenbanken oder E-Mails.

12. Auf „ **PGN einfügen**“ klicken.
13. PGN-Text in das Textfeld kopieren.
14. Auf „**Importieren**“ klicken.
15. Die Überschreiben-Abfrage bestätigen.
16. Die importierte Sammlung wird angezeigt.

### Praxis-Tipp

Vor einem Import sollte eine bereits bearbeitete Sammlung exportiert werden, wenn sie erhalten bleiben soll.

## 2.6 Partieliste verwenden

Über „ **Partieliste**“ öffnet sich eine Übersicht aller geladenen Partien. Dort kann zwischen Partien gewechselt, gesucht und gelöscht werden. Die Suchfunktion durchsucht die Kopfdaten, zum Beispiel Spielernamen, Event, Datum und Ergebnis.

- Ein Klick auf eine Zeile öffnet die entsprechende Partie.
- Die aktuelle Partie ist hervorgehoben.
- Die Papierkorb-Schaltfläche löscht eine Partie nach Bestätigung.
- Wenn die letzte Partie gelöscht wird, erzeugt der Analyzer automatisch wieder eine leere Partie.

## 2.7 Züge am Brett eingeben

17. Figur anklicken.
18. Zielfeld anklicken.
19. Der Zug wird ausgeführt, wenn er legal ist.
20. Die Zugliste, der PlyCount und die Statusanzeige werden aktualisiert.

Nach Auswahl einer Figur markiert der Analyzer die möglichen Zielfelder. Es können nur Figuren der Seite gezogen werden, die am Zug ist.

## 2.8 Illegale Züge

Illegale Züge werden nicht ausgeführt. Die Oberfläche zeigt kurz eine Meldung und markiert das Brett visuell. Häufige Ursachen sind:

- Die Figur kann nach den Schachregeln nicht auf das gewählte Feld ziehen.
- Der eigene König würde nach dem Zug im Schach stehen.
- Es ist die falsche Farbe am Zug.
- Eine Rochade ist wegen Angriff, Figurenbewegung oder blockierter Felder nicht mehr erlaubt.
- Ein en-passant-Zug ist in der aktuellen Stellung nicht möglich.

## 2.9 Bauernumwandlung

Erreicht ein Bauer die letzte Reihe, öffnet sich ein Dialog zur Auswahl der Umwandlungsfigur. Zur Verfügung stehen Dame, Turm, Läufer und Springer. Wird der Dialog abgebrochen, wird der Zug nicht ausgeführt.

## 2.10 Navigation, Rückgängig und Flip

Schaltfläche / Taste	Funktion	Bemerkung
« / Home	Zum Anfang springen	Zeigt die Ausgangsstellung
◀ / Pfeil links	Einen Halbzug zurück	Verändert die Partie nicht
▶ / Pfeil rechts	Einen Halbzug vor	Bis maximal zum Ende der Historie
» / End	Zum Ende springen	Aktuelle Endstellung



Schaltfläche / Taste	Funktion	Bemerkung
Rückgängig / U	Zug entfernen oder Fortsetzung abschneiden	Je nach aktueller Ansichtsstellung
Flip / F	Brett drehen	Nur Anzeige, keine Änderung der Partie
Escape	Dialog schließen	PGN-Import oder Umwandlung

Wird an einer früheren Stellung ein neuer Zug eingegeben, wird die bisherige Fortsetzung ab dieser Stelle ersetzt. Diese Logik eignet sich zum Korrigieren falscher Eingaben oder zum Verwerfen einer Variante.

## 2.11 Speichern und Exportieren

Mit „**Speichern**“ wird die aktuelle Partie als PGN-Datei heruntergeladen. Der Dateiname wird aus Partienummer, Datum und Spielernamen gebildet, soweit diese Angaben vorhanden sind.

Mit „**PGN erstellen**“ wird die gesamte aktuelle Sammlung als eine PGN-Datei exportiert. Diese Funktion ist sinnvoll, wenn mehrere Partien importiert, bearbeitet oder neu erfasst wurden.

## 2.12 Zugliste verwenden

Die Zugliste zeigt die Partie in SAN-Notation, zum Beispiel e4, Nf3, O-O, exd5, Qh5+ oder Rxe8#. Ein Klick auf einen Zug springt zur Stellung nach diesem Zug. Der aktive Zug wird hervorgehoben.

## 2.13 Statusanzeige

Die Statusanzeige unter dem Brett meldet unter anderem, welche Seite am Zug ist, ob Schach vorliegt und ob die Partie durch Matt oder Patt beendet ist. Außerdem wird die aktuelle Partienummer innerhalb der Sammlung angezeigt.

## 2.14 Engine-Analyse verwenden

Die Engine analysiert die aktuell angezeigte Stellung. Standardmäßig ist die Auto-Analyse aktiv. Wenn sich die Stellung ändert, wird automatisch eine neue Analyse angefordert. Bei deaktivierter Auto-Analyse erscheint die Schaltfläche „**Analysieren**“.

Engine-Bedienung	
Auto-Analyse	Startet nach Stellungswechsel automatisch eine neue Analyse.
Analysieren	Startet manuell eine Analyse, wenn Auto-Analyse deaktiviert ist.
Stop	Beendet eine laufende Suche.
Suchmodus Tiefe	Analysiert bis zur gewählten Suchtiefe.
Suchmodus Zeit	Analysiert für die angegebene Zeit in Millisekunden.
MultiPV	Zeigt eine oder mehrere Hauptvarianten, maximal vier.

## 2.15 Engine-Anzeigen verstehen

Anzeige	Bedeutung
Bewertung (am Zug)	Stellungsbewertung aus Sicht der Seite, die am Zug ist. Positive Werte bedeuten Vorteil für diese Seite.
WDL	Win/Draw/Loss-Werte, sofern von der Engine geliefert.
Tiefe	Erreichte Rechentiefe der Engine.
NPS	Berechnete Knoten pro Sekunde.



Anzeige	Bedeutung
Bestzug	Von der Engine gemeldeter bester Zug, nach Möglichkeit in SAN und UCI.
Variantenliste	Hauptvariante oder mehrere Hauptvarianten abhängig von MultiPV.

## 2.16 Empfohlener Analyseablauf

21. PGN-Datei öffnen oder PGN-Text einfügen.
22. In der Partieliste die gewünschte Partie auswählen.
23. Kopfdaten prüfen und gegebenenfalls ergänzen.
24. Mit Navigationsschaltflächen oder Pfeiltasten zur kritischen Stellung gehen.
25. Engine-Bewertung und Varianten betrachten.
26. Bei Bedarf MultiPV erhöhen, um mehrere Kandidatenzüge zu vergleichen.
27. Korrekturen oder neue Fortsetzungen eingeben.
28. Aktuelle Partie speichern oder alle Partien als PGN exportieren.

## 3. Technik

### 3.1 Grundaufbau

Der Hammerschach-Analyzer ist als Webanwendung in HTML, CSS und JavaScript umgesetzt. Die HTML-Struktur enthält Kopfdaten, Brett, Schaltflächen, Dialoge, Engine-Anzeige und Zugliste. CSS steuert Layout, Farben, Responsivität und Markierungen. JavaScript übernimmt Schachregeln, PGN-Verarbeitung, Zustandsverwaltung, Rendering und Engine-Kommunikation.

Im vorliegenden Aufbau gibt es keine serverseitige Datenbanklogik. Partien werden während der Nutzung im Arbeitsspeicher der Browserseite gehalten und müssen für eine dauerhafte Sicherung exportiert werden.

### 3.2 Datenmodell

Die Partiensammlung wird als Liste geführt. Jede Partie besteht aus Kopfdaten, Zughistorie und Brettausrichtung. Beim Wechsel zwischen Partien wird der aktuelle Stand gesichert und die neue Partie geladen.

Element	Technische Rolle
headers	PGN-Kopfdaten der Partie
history	Liste der ausgeführten Halbzüge
orientationWhite	Ausrichtung des Brettes aus weißer oder schwarzer Sicht
currentGameIndex	Index der aktuell geöffneten Partie
viewIndex	Angezeigte Stellung innerhalb der Zughistorie

### 3.3 Interne Brett- und Figurenlogik

Die Schachlogik arbeitet mit einer Game-Struktur. Das Brett wird als 8x8-Array gespeichert. Großbuchstaben repräsentieren weiße Figuren, Kleinbuchstaben schwarze Figuren und ein Punkt steht für ein leeres Feld.

Zeichen	Figur
P / p	weißer / schwarzer Bauer
R / r	weißer / schwarzer Turm
N / n	weißer / schwarzer Springer
B / b	weißer / schwarzer Läufer



Zeichen	Figur
Q / q	weiße / schwarze Dame
K / k	weißer / schwarzer König
.	leeres Feld

Zusätzlich verwaltet die Game-Struktur das Zugrecht, das en-passant-Feld, die Rochaderechte, den Halbzugzähler und den Vollzugzähler.

### 3.4 Zugerzeugung und Legalitätsprüfung

Zunächst werden pseudo-legale Züge erzeugt. Anschließend simuliert der Analyzer den Zug und prüft, ob der eigene König danach angegriffen ist. Nur Züge, die diese Prüfung bestehen, werden als legal angeboten.

- Bauernzüge einschließlich Doppelschritt, Schlagzug, en passant und Umwandlung.
- Springerzüge.
- Läufer-, Turm- und Damenzüge entlang ihrer Linien.
- Königszüge und Rochade.
- Prüfung von Schach, Matt und Patt.

Bei der Rochade wird nicht nur geprüft, ob die Felder frei sind. Es wird auch geprüft, ob Ausgangsfeld, Durchgangsfeld und Zielfeld des Königs angegriffen sind.

### 3.5 Ausführung eines Zuges

Beim Ausführen eines Zuges aktualisiert der Analyzer mehrere Zustände. Dazu gehören Ausgangs- und Zielfeld, geschlagene Figuren, en-passant-Schläge, Rochadeturn, Umwandlungsfigur, en-passant-Feld, Rochaderechte, Halbzugzähler, Vollzugzähler und Zugrecht.

#### Technischer Nutzen

Diese vollständige Aktualisierung ist notwendig, damit spätere Legalitätsprüfungen, FEN-Erzeugung und PGN-Ausgabe konsistent bleiben.

### 3.6 Schach, Matt und Patt

Die Spielendprüfung untersucht, ob die Seite am Zug noch legale Züge besitzt. Gibt es keine legalen Züge, entscheidet der Schachstatus des Königs: Steht der König im Schach, ist es Matt; steht er nicht im Schach, ist es Patt.

### 3.7 Rendering des Brettes

Das Brett wird bei jeder relevanten Änderung neu aufgebaut. Für jedes Feld wird ein HTML-Element erzeugt. Figuren werden als Bilddateien eingebunden. CSS-Klassen markieren ausgewählte Figuren, legale Zielfelder, Engine-Vorschläge und illegale Zugversuche.

Die Koordinaten werden abhängig von der Brettausrichtung angezeigt. Bei normaler Ausrichtung erscheinen Dateien und Reihen aus weißer Sicht; nach dem Flip entsprechend aus schwarzer Sicht.

### 3.8 PGN-Import

Der PGN-Import normalisiert den Text, trennt Partien anhand von Event-Blöcken, liest Kopfdaten aus und verarbeitet anschließend die Zugliste. Kommentare in geschweiften Klammern und Varianten in runden Klammern werden entfernt. Importiert wird die Hauptlinie.

29. PGN-Text normalisieren.
30. Partien anhand von Event-Blöcken trennen.
31. Header-Zeilen auslesen.
32. Kommentare und Varianten entfernen.
33. Zugtokens erzeugen.
34. SAN-Züge gegen legale Züge der aktuellen Stellung abgleichen.
35. Interne Zughistorie aufbauen.



36. PlyCount berechnen.

Wenn ein SAN-Token nicht legal aufgelöst werden kann, bricht der Import der Zugfolge an dieser Stelle ab. In der Browser-Konsole wird ein Hinweis ausgegeben.

### 3.9 PGN-Export

Beim Export werden die Kopfdaten in festgelegter Reihenfolge geschrieben. Die Züge werden aus der internen Historie in SAN-Notation rekonstruiert, mit Zugnummern versehen und zusammen mit dem Ergebnis ausgegeben.

- Export einer einzelnen Partie über „Speichern“.
- Export der gesamten Sammlung über „PGN erstellen“.
- Automatische Aktualisierung des PlyCount.
- Zeilenumbruch der Zugliste bei ungefähr 80 Zeichen.

### 3.10 FEN-Erzeugung

Für die Engine wird aus der aktuell angezeigten Stellung eine FEN-Zeichenkette erzeugt. Sie enthält Figurenstellung, aktive Farbe, Rochaderechte, en-passant-Feld, Halbzugzähler und Vollzugzähler. Dadurch analysiert die Engine exakt die sichtbare Stellung.

### 3.11 Engine-Integration

Die Engine wird als Web Worker geladen. Der Analyzer erwartet die Dateien stockfish.js und stockfish.wasm. Vor dem Start prüft die Anwendung, ob beide Dateien erreichbar sind. Wenn eine Datei fehlt oder nicht geladen werden kann, erscheint eine Fehlermeldung im Engine-Bereich.

Nach dem Laden sendet der Analyzer den UCI-Befehl uci. Die Engine muss mit uciok antworten. Danach wird isready gesendet. Nach readyok gilt die Engine als bereit.

UCI-Schritt	Bedeutung
uci	Start des UCI-Handshakes
uciok	Engine bestätigt UCI-Kommunikation
isready	Analyzer fragt Bereitschaft ab
readyok	Engine ist bereit
position fen ...	Aktuelle Stellung wird gesetzt
go depth ...	Analyse bis zu einer Tiefe
go movetime ...	Analyse für eine Zeitspanne
bestmove ...	Engine meldet den besten Zug

### 3.12 Engine-Konfiguration und Ausgabe

Vor einer Analyse setzt der Analyzer UCI-Optionen. UCI\_ShowWDL wird aktiviert, MultiPV wird gemäß Benutzerauswahl gesetzt, Threads werden aus der Browserangabe zur Hardware abgeleitet und auf maximal 8 begrenzt, Hash wird auf 64 gesetzt.

Während der Suche werden info-Zeilen verarbeitet. Daraus liest der Analyzer unter anderem depth, seldepth, multipv, nodes, nps, time, score, mate score, wdl und pv. Die Hauptvarianten werden nach MultiPV-Index gespeichert und angezeigt.

### 3.13 UCI- und SAN-Notation

Die Engine liefert Züge in UCI-Notation, zum Beispiel e2e4 oder e7e8q. Für die Benutzeranzeige werden diese Züge gegen die legalen Züge der Stellung geprüft und nach Möglichkeit in SAN-Notation umgerechnet. Dadurch bleibt die Engine-Kommunikation technisch korrekt, während die Anzeige für Schachspieler lesbar ist.



### 3.14 Responsives Layout

Das Layout reagiert auf Bildschirmbreiten und Fensterhöhe. Bei kleineren Bildschirmen wird die Seitenleiste unter das Brett gesetzt, Kopfdaten wechseln auf zwei oder eine Spalte und die Bedienelemente werden kompakter. Eine Layoutfunktion berechnet die verfügbare Höhe und setzt die Brettgröße über die CSS-Variable `--board-size`.

### 3.15 Grenzen und Hinweise

- Ohne Export gehen Änderungen beim Neuladen der Seite verloren.
- Der PGN-Import konzentriert sich auf die Hauptlinie und entfernt Kommentare sowie Varianten.
- Komplexe PGN-Sonderfälle können scheitern, wenn sie nicht auf einen legalen SAN-Zug abgebildet werden können.
- Die Engine funktioniert nur, wenn `stockfish.js` und `stockfish.wasm` korrekt erreichbar sind.
- Im vorliegenden Aufbau gibt es keine Benutzerverwaltung und keine serverseitige Speicherung.

## 4. Resümee

Der Hammerschach-Analyzer verbindet eine kompakte PGN-Verwaltung mit direkter Bretteingabe und Engine-Analyse. Er eignet sich für Vereine, Training, Turniernachbereitung und private Partieanalyse.

Die Stärken liegen in der einfachen Bedienung: Partien können per Datei oder Text importiert werden, mehrere Partien lassen sich in einer Sammlung verwalten, Züge werden am Brett eingegeben und auf Legalität geprüft, die Zugliste erlaubt schnelles Springen durch die Partie und die Engine analysiert die aktuelle Stellung automatisch oder auf Knopfdruck.

Technisch ist der Analyzer übersichtlich aufgebaut. Schachlogik, PGN-Verarbeitung, FEN-Erzeugung und Engine-Kommunikation sind direkt in der Browseranwendung umgesetzt. Dadurch ist die Anwendung leicht zu verteilen, benötigt aber für dauerhafte Speicherung den PGN-Export.

#### **Empfehlung für den produktiven Einsatz**

Nach jedem Bearbeitungsschritt, der erhalten bleiben soll, sollte die aktuelle Partie oder die gesamte Sammlung als PGN exportiert werden.

Mit einem klaren Arbeitsablauf wird der Hammerschach-Analyzer zu einem zuverlässigen Werkzeug für strukturierte Arbeit mit Schachpartien: importieren, prüfen, analysieren, korrigieren und exportieren.

# Handbuch

## Hammerschach-Trainer

*Taktiktraining, Aufgabenfilter, Rating und Fortschritt*





# Inhaltsverzeichnis

1. Einleitung
2. Bedienung
3. Technik
4. Resümee

## Dokumentaufbau

Dieses Handbuch beschreibt den Hammerschach-Trainer aus Anwendersicht und ergänzt die Bedienung um technische Hinweise zur HTML-, CSS- und JavaScript-Umsetzung. Das Layout ist auf DIN A4 Hochformat optimiert.

## 1. Einleitung

Der Hammerschach-Trainer ist eine browserbasierte Trainingsoberfläche für Schachtaktik. Er lädt Aufgaben, stellt die Ausgangsstellung auf einem grafischen Brett dar und prüft anschließend, ob die vom Nutzer gespielte Fortsetzung der hinterlegten Lösung entspricht.

Der Trainer verbindet drei Kernbereiche: Taktikaufgaben, Filtersteuerung und Fortschrittsbewertung. Damit eignet er sich sowohl für freies Training als auch für ein strukturiertes Vereins- oder Einzeltraining.

1. **Taktiktraining:** Aufgaben werden aus einer Aufgabenliste geladen, am Brett gelöst und Schritt für Schritt geprüft.
2. **Filterung:** Schwierigkeit, Themen, maximale Zugzahl und Popularität können eingegrenzt werden.
3. **Fortschritt:** Nach Login kann ein Taktik-Rating geladen und bei Erfolg oder Fehlversuch aktualisiert werden.

### 1.1 Zielgruppe

- Spieler, die regelmäßig Schachtaktik trainieren möchten.
- Trainer, die Aufgaben nach Thema und Schwierigkeit auswählen wollen.
- Vereine, die ein kompaktes Online-Training in ihre Webseite integrieren möchten.
- Anwender, die ohne separates Schachprogramm direkt im Browser trainieren wollen.

### 1.2 Grundprinzip des Trainings

Eine Aufgabe besteht aus einer FEN-Stellung und einer hinterlegten Zugfolge in UCI-Notation. Beim Start einer Aufgabe wird die Stellung geladen. Der erste Zug der Aufgabenfolge wird automatisch ausgeführt; danach muss der Nutzer die beste Fortsetzung finden. Nach jedem korrekten Nutzerzug spielt der Trainer die Antwortseite automatisch weiter, bis die Kombination abgeschlossen ist.

#### Merksatz

Aufgabe wählen, Stellung verstehen, Kandidatenzug berechnen, am Brett ausführen und die Rückmeldung auswerten.

## 2. Bedienung

### 2.1 Aufbau der Oberfläche

Die Oberfläche ist in einen Brettbereich und eine Seitenleiste gegliedert. Oberhalb des Brettes befinden sich Logo, Taktik-Rating, Login-Schaltfläche, Statusanzeige sowie die Bedienelemente fuer Brettfarbe und Anleitung. Rechts beziehungsweise auf kleinen Bildschirmen unterhalb des Brettes befinden sich Filter und Informationen zur aktuellen Aufgabe.



Hauptbereiche der Bedienoberfläche	
Kopfleiste	Logo, Taktik-Rating, Login-Schaltfläche und Statusmeldung.
Brett	Interaktives Schachbrett mit Figuren, Koordinaten und Klicksteuerung.
Steuerung	Schaltflächen Nächste Aufgabe, Wiederholen und Flip.
Filter	Rating-Bereich, automatische Schwierigkeit, Themen, maximale Halbzüge und Popularität.
Aktuelle Aufgabe	Aufgaben-ID, Rating, Themen-Badges, Popularität und Lösung in SAN-Notation.

## 2.2 Start und Aufgabenladung

Beim Start der Seite wird das Brett gerendert und anschließend eine Aufgabenliste geladen. Standardmäßig erwartet die Anwendung eine Datei puzzles.json im gleichen Veröffentlichungsumfeld. Ist diese Datei nicht erreichbar oder leer, wird eine eingebaute Fallback-Aufgabe verwendet.

4. Die Seite öffnen.
5. Warten, bis eine Aufgabe geladen ist.
6. Statusmeldung lesen: Dort steht, welche Seite am Zug ist.
7. Auf dem Brett den berechneten Zug eingeben.

## 2.3 Login, Registrierung und Logout

Über die Login-Schaltfläche wird ein Anmeldedialog geöffnet. Dort können Benutzername und Passwort eingegeben werden. Zusätzlich steht eine Registrierungsfunktion zur Verfügung. Nach erfolgreichem Login wird der Benutzername in der Schaltfläche angezeigt; ein weiterer Klick meldet den Nutzer wieder ab.

Benutzerkonto	
Login	Meldet einen vorhandenen Nutzer an und speichert Sitzungsdaten im Browser.
Registrieren	Legt einen neuen Nutzer an, sofern die Backend-Schnittstelle die Registrierung akzeptiert.
Logout	Löscht Sitzungstoken und Benutzernamen aus dem lokalen Browser-Speicher.
Session-Check	Beim Start wird ein vorhandenes Token geprüft, damit der Nutzer automatisch angemeldet bleiben kann.

## 2.4 Taktik-Rating

Nach dem Login lädt der Trainer ein Taktik-Rating. Ohne Login wird kein persönliches Rating angezeigt. Bei einer normal gelösten Aufgabe wird ein Erfolg gespeichert; bei einem falschen Zug wird ein Fehlversuch gespeichert.

Das Rating wird nicht gespeichert, wenn die Aufgabe im Wiederholen-Modus gespielt wird oder wenn zuvor die Lösung eingeblendet wurde. Dadurch wird verhindert, dass Wiederholungen oder bereits aufgedeckte Aufgaben die Bewertung verfälschen.

### Fairness-Regel

Ratingänderungen entstehen nur bei regulären Lösungsversuchen ohne vorherige Lösungsanzeige und nicht beim Wiederholen.



## 2.5 Schwierigkeit automatisch bestimmen

Die Option "Schwierigkeit automatisch bestimmen" ist nur nach Login verfügbar. Ist sie aktiv, werden die Rating-Schieberegler gesperrt und der Trainer wählt den passenden Ratingbereich anhand des persönlichen Taktik-Ratings.

Eigenes Rating	Automatischer Aufgabenbereich
unter 1200	800 bis 1300
1200 bis unter 1600	1100 bis 1800
ab 1600	1400 bis 2400

## 2.6 Manuelle Filter verwenden

Ohne automatische Schwierigkeit werden die Aufgaben über mehrere Filter eingegrenzt. Die Anzeige "Gefilterte Aufgaben" zeigt, wie viele Aufgaben aktuell zu den Kriterien passen. Falls keine Aufgabe passt, greift die Anwendung auf die Fallback-Aufgabe zurück.

Filter	Funktion
Rating-Bereich	Minimum und Maximum zwischen 800 und 2400 Elo.
Themen	Mehrfachauswahl, zum Beispiel Matt, Gabel, Fesselung, Opfer oder Endspiel.
Maximale Züge	Begrenzt die Länge der Lösungsfolge in Halbzügen.
Popularität	Legt fest, wie beliebt eine Aufgabe mindestens sein soll.
Leere Themenauswahl	Bedeutet: alle Themen sind erlaubt.

## 2.7 Brett bedienen

Züge werden per Mausklick eingegeben. Zuerst wird die eigene Figur ausgewählt, danach das Zielfeld. Der Trainer akzeptiert nur Züge der Seite, die am Zug ist, und nur legale Züge der aktuellen Stellung.

- Eine ausgewählte Figur wird visuell hervorgehoben.
- Ungültige Züge lösen eine kurze optische Rückmeldung und eine Statusmeldung aus.
- Bei Bauernumwandlung wird automatisch zur Dame umgewandelt, sofern die Aufgabenlösung dies nicht anders spezifiziert.
- Mit "Flip" wird das Brett aus der anderen Perspektive angezeigt.

## 2.8 Brettfarbe wählen

Über "Brettfarbe" öffnet sich ein Auswahlfeld mit mehreren Farbschemata. Die Auswahl wird im lokalen Browser-Speicher gesichert und bleibt daher beim nächsten Öffnen erhalten.

Auswahl	Beschreibung
Basis	Rotbraune Hammerschach-Optik mit hellen Feldern.
Klassisch Braun	Traditionelle Braun- und Beigetöne.
Grau / Neutral	Zurückhaltende neutrale Darstellung.
Grün / Turnier	Grünliches Turnierbrett nach verbreitetem Online-Stil.

## 2.9 Eine Aufgabe lösen

Nach dem Laden einer Aufgabe zeigt die Statuszeile, welche Seite am Zug ist. Der Nutzer berechnet die beste Fortsetzung und spielt den Zug auf dem Brett. Der Trainer vergleicht den gespielten Zug mit dem nächsten erwarteten Zug aus der Lösungslinie.

8. Aufgabe laden oder mit "Nächste Aufgabe" eine neue Aufgabe wählen.
9. Stellung ansehen und Kandidatenzüge berechnen.
10. Figur anklicken und Zielfeld anklicken.



11. Bei korrektem Zug spielt der Trainer die Antwort automatisch weiter.
  12. Falls weitere eigene Züge notwendig sind, die Fortsetzung zu Ende lösen.
  13. Nach Abschluss erscheint eine Erfolgsmeldung.
- Wird ein falscher Zug gespielt, wird die Aufgabe nicht fortgesetzt. Die Statusmeldung weist darauf hin, dass der Zug nicht der Lösung entspricht. Bei angemeldetem Nutzer kann dieser Fehlversuch in die Fortschrittsbewertung einfließen.

## 2.10 Nächste Aufgabe und Wiederholen

Steuerung unter dem Brett	
Nächste Aufgabe	Wählt zufällig eine Aufgabe aus der aktuell gefilterten Aufgabenliste.
Wiederholen	Startet dieselbe Aufgabe erneut. In diesem Modus wird kein Rating gespeichert.
Flip	Dreht die Brettansicht, ohne die Stellung oder die Lösung zu verändern.

## 2.11 Lösung anzeigen

Mit "Lösung" wird die hinterlegte Lösung in SAN-Notation eingeblendet. Der erste automatisch ausgeführte Zug wird dabei nicht angezeigt; sichtbar ist die Fortsetzung, die der Nutzer lösen soll. Ein weiterer Klick blendet die Lösung wieder aus.

Sobald die Lösung angezeigt wurde, setzt der Trainer intern einen Lösungsabbruch. Danach wird für diese Aufgabe kein Rating gespeichert. Das schützt die Bewertung vor Aufgaben, die bereits aufgedeckt wurden.

## 2.12 Informationen zur aktuellen Aufgabe

Die Box "Aktuelle Aufgabe" zeigt die Aufgaben-ID, die ungefähre Schwierigkeit in Elo, Themenmarkierungen und gegebenenfalls die Popularität. Die Themen helfen, typische taktische Motive zu erkennen und gezielt zu trainieren.

- Matt, Matt in 1 und Matt in 2 für direkte Mattmotive.
- Gabel, Fesselung und Spieß für taktische Standardmotive.
- Opfer, Endspiel, Eröffnung und Mittelspiel für thematische Einordnung.
- Popularität als zusätzlicher Qualitäts- oder Auswahlhinweis.

## 2.13 Empfohlener Trainingsablauf

14. Mit Login starten, wenn das persönliche Rating genutzt werden soll.
15. Entweder automatische Schwierigkeit aktivieren oder einen manuellen Ratingbereich wählen.
16. Ein bis zwei Themen auswählen, wenn gezielt trainiert werden soll.
17. Aufgabe vollständig ohne Lösungshilfe berechnen.
18. Zugfolge am Brett ausführen und Rückmeldungen beachten.
19. Bei Fehlern zuerst selbst nachrechnen, dann gegebenenfalls die Lösung anzeigen.
20. Regelmäßig Aufgaben wiederholen, ohne das Rating als Hauptziel zu betrachten.

# 3. Technik

## 3.1 Grundaufbau

Der Hammerschach-Trainer ist als einzelne HTML-Anwendung mit CSS und JavaScript aufgebaut. Die HTML-Struktur stellt Logo, Status, Brett, Steuerbuttons, Filter, Aufgabeninformationen und Login-Dialog bereit. CSS definiert Layout, Brettfarben, responsive Anordnung und visuelle Effekte. JavaScript übernimmt Spiellogik, Aufgabensteuerung, Filterung, Authentifizierung, Fortschrittspeicherung und Rendering.

Die Anwendung ist auf eine statische Auslieferung vorbereitet. Für den vollständigen Betrieb müssen zusätzlich die Aufgabenliste puzzles.json, das Logo Trainer-Logo.png und die erreichbaren Backend-Schnittstellen für Login und Fortschritt vorhanden sein.



### 3.2 Datenformat der Aufgaben

Die Aufgaben werden aus einer JSON-Liste geladen. Jede Aufgabe besitzt mindestens eine FEN-Stellung und eine Zugfolge. Weitere Felder dienen der Anzeige und Filterung.

Feld	Bedeutung
id	Eindeutige Kennung der Aufgabe.
fen	Startstellung im FEN-Format.
moves	Lösungsfolge in UCI-Notation, zum Beispiel e2e4.
rating	Ungefähre Schwierigkeit der Aufgabe.
themes	Themenliste, zum Beispiel mate, fork, pin oder endgame.
popularity	Popularitätswert zur Filterung.

Wenn puzzles.json nicht geladen werden kann, verwendet die Anwendung eine eingebaute Fallback-Aufgabe. Dadurch bleibt die Oberfläche testbar, auch wenn die externe Aufgabenliste fehlt.

### 3.3 Initialisierung einer Aufgabe

Beim Einrichten einer Aufgabe wird die FEN-Stellung geladen und die Lösungslinie kopiert. Falls eine Zugfolge vorhanden ist, wird der erste Zug automatisch auf dem internen Brett ausgeführt und aus der Lösungslinie entfernt. Danach richtet sich die Brettorientierung nach der Seite, die nun am Zug ist.

21. Aufgabe auswählen.
22. FEN in die Game-Struktur laden.
23. Lösungslinie aus dem moves-Feld kopieren.
24. Ersten UCI-Zug automatisch ausführen.
25. Status, Metadaten, Badges und Brettansicht aktualisieren.

### 3.4 Interne Schachlogik

Die Game-Struktur verwaltet das Brett als 8x8-Array. Großbuchstaben stehen für weiße Figuren, Kleinbuchstaben für schwarze Figuren und ein Punkt für ein leeres Feld. Zusätzlich werden Zugrecht, en-passant-Feld, Rochaderechte, Halbzug- und Vollzugzähler gespeichert.

Zeichen	Figur
P / p	weißer / schwarzer Bauer
R / r	weißer / schwarzer Turm
N / n	weißer / schwarzer Springer
B / b	weißer / schwarzer Läufer
Q / q	weiße / schwarze Dame
K / k	weißer / schwarzer König
.	leeres Feld

### 3.5 Zugerzeugung und Legalitätsprüfung

Für die Seite am Zug werden pseudo-legale Züge erzeugt und anschließend simuliert. Ein Zug wird nur akzeptiert, wenn der eigene König nach dem Zug nicht im Schach steht. Die Angriffserkennung prüft Bauern, Springer, Läufer- und Damenlinien, Turm- und Damenlinien sowie Königskontaktfelder.

- Bauernzüge einschließlich Doppelschritt, Schlagzug, en passant und Umwandlung.
- Springerzüge.
- Linienzüge von Läufer, Turm und Dame.
- Königszüge auf angrenzende Felder.
- Prüfung auf Schach, Matt und Patt.

#### Technischer Hinweis



Die vorliegende Trainer-Version liest Rochaderechte aus der FEN ein, erzeugt in der Klicklogik jedoch keine Rochadezüge. Für typische Taktikaufgaben ist das meist unkritisch, sollte bei Aufgaben mit Rochadebedarf aber beachtet werden.

### 3.6 Zugvergleich mit der Lösung

Der vom Nutzer gespielte Zug wird intern in UCI-Notation umgewandelt. Danach vergleicht der Trainer diesen UCI-Zug mit dem nächsten erwarteten Eintrag der Lösungslinie. Stimmt der Zug nicht überein, wird der Versuch als falsch markiert und die Stellung bleibt erhalten.

Stimmt der Zug überein, wird er ausgeführt. Falls danach noch ein automatischer Antwortzug in der Lösungslinie vorhanden ist, wird dieser animiert abgespielt. Anschließend ist wieder der Nutzer am Zug. Ist keine Lösungslinie mehr übrig, gilt die Aufgabe als gelöst.

### 3.7 SAN-Notation für die Lösungsanzeige

Die Lösung wird für die Anzeige nicht als reine UCI-Liste ausgegeben. Stattdessen wandelt eine Hilfsfunktion UCI-Züge in SAN-Notation um. Dafür wird jeder Zug gegen die legalen Züge der jeweiligen Stellung geprüft, ausgeführt und anschließend mit Schlagzeichen, Umwandlung, Schach oder Matt ergänzt.

Der erste automatisch ausgeführte Zug wird bei der Lösungsanzeige übersprungen, damit nur die vom Nutzer zu findende Fortsetzung erscheint.

### 3.8 Rendering und responsives Layout

Das Brett wird dynamisch aus HTML-Elementen aufgebaut. Jedes Feld erhält Koordinaten, Feldfarbe und gegebenenfalls ein Figurenbild. Die Brettgröße verwendet CSS-Variablen und passt sich an Desktop- und Mobilansichten an. Auf kleineren Bildschirmen werden Brettbereich und Seitenleiste untereinander dargestellt.

Die Figurenbilder werden über Wikimedia-Commons-URLs eingebunden. Für einen vollständig unabhängigen Offline-Betrieb müssten diese SVG-Dateien lokal bereitgestellt und im Code entsprechend referenziert werden.

### 3.9 Lokaler Browser-Speicher

Der Trainer nutzt localStorage für mehrere dauerhafte Einstellungen. Dazu gehören das gewählte Brettfarbschema sowie bei Login der Sitzungstoken und der Benutzername. Beim Logout werden Token und Benutzername wieder entfernt.

LocalStorage-Nutzung	
Brettfarbe	Speichert das gewählte Farbschema unter einem lokalen Schlüssel.
trainerToken	Speichert das Sitzungstoken nach erfolgreichem Login.
trainerUser	Speichert den Benutzernamen für den automatischen Session-Check.

### 3.10 Backend-Schnittstellen

Für Login und Fortschritt nutzt die Anwendung HTTP-Anfragen an eine Backend-Schnittstelle. Die wichtigsten Abläufe sind Sessionprüfung, Login, Registrierung, Laden des Fortschritts und Speichern eines Trainingsergebnisses.

Funktion	Technische Rolle
auth/status	Prüft ein vorhandenes Sitzungstoken.
auth/login	Meldet einen Nutzer mit Benutzername und Passwort an.
auth/register	Registriert einen neuen Nutzer.
progress/load	Lädt das gespeicherte Taktik-Rating eines Nutzers.



Funktion	Technische Rolle
progress/save	Speichert Erfolg oder Fehlversuch und aktualisiert das Rating.

Die Rating-Speicherung ist an Bedingungen geknüpft: Sie findet nicht im Wiederholen-Modus statt, nicht nach Anzeige der Lösung und nicht ohne angemeldeten Nutzer.

### 3.11 Grenzen und Betriebshinweise

- Ohne puzzles.json steht nur die eingebaute Fallback-Aufgabe zur Verfügung.
- Ohne erreichbare Backend-Schnittstelle funktionieren Login, Registrierung und Rating-Speicherung nicht.
- Ohne Internetzugriff können extern eingebundene Figurenbilder fehlen, sofern sie nicht lokal bereitgestellt werden.
- Die Lösungslinie ist verbindlich; alternative gleichwertige taktische Züge werden nur akzeptiert, wenn sie in der Aufgabenlösung hinterlegt sind.
- Rochadezüge werden in der aktuellen Klicklogik nicht erzeugt, obwohl FEN-Rochaderechte eingelesen werden.

## 4. Resümee

Der Hammerschach-Trainer ist ein kompaktes Werkzeug für strukturiertes Taktiktraining im Browser. Die Anwendung kombiniert ein interaktives Schachbrett, filterbare Aufgaben, thematische Einordnung, Lösungsanzeige und optionales persönliches Rating.

Für Anwender ist der Trainer besonders einfach: Aufgabe wählen, Stellung berechnen, Zug ausführen und Rückmeldung beachten. Für Trainer und Vereine ist vor allem die Filterung nach Schwierigkeit, Thema, Zuglänge und Popularität hilfreich.

Technisch ist der Trainer übersichtlich aufgebaut. Eine statische HTML-Anwendung lädt Aufgaben aus JSON, verwaltet die Schachlogik intern und nutzt bei Bedarf Backend-Schnittstellen für Benutzerkonto und Fortschritt. Für einen zuverlässigen Produktivbetrieb sollten puzzles.json, Logo, Figurenbilder und Backend-Endpunkte stabil erreichbar sein.

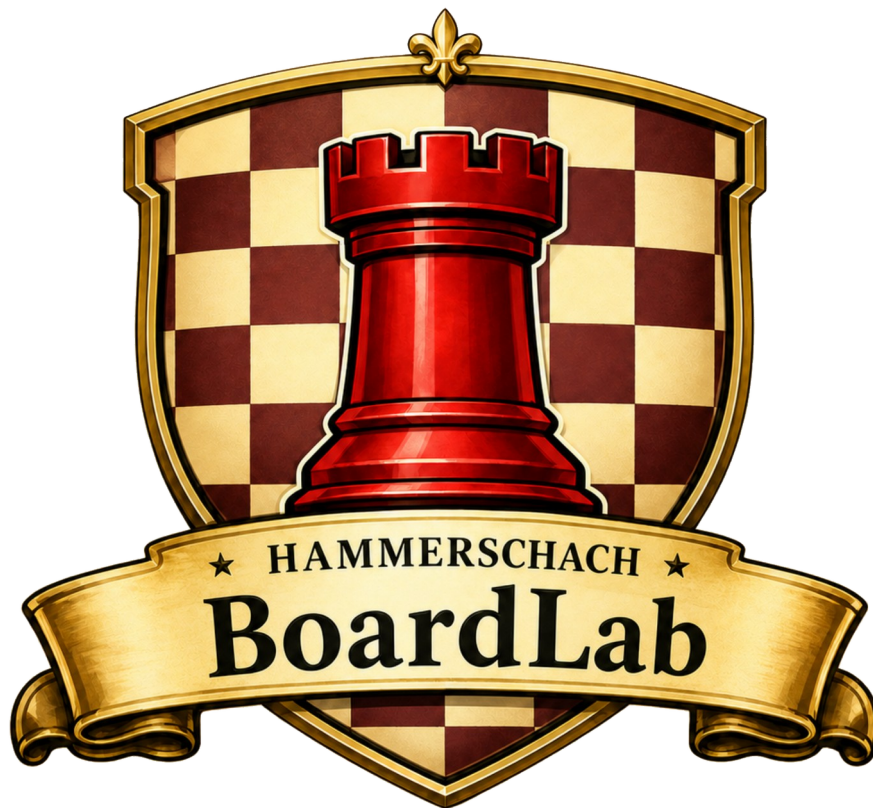
### Empfehlung

Der größte Trainingseffekt entsteht, wenn Aufgaben zuerst ohne Lösungsanzeige berechnet und erst danach mit der Lösung kontrolliert werden.

# Handbuch

## Hammerschach-BoardLab

*Stellungsaufbau, Play & Analyze und Engine-Training*





# Inhaltsverzeichnis

1. Einleitung
2. Bedienung
3. Technik
4. Resümee

## Dokumentaufbau

Dieses Handbuch beschreibt Hammerschach-BoardLab aus Anwendersicht und ergänzt die Bedienung um technische Hinweise zur HTML-, CSS- und JavaScript-Umsetzung. Das Layout ist auf DIN A4 Hochformat optimiert.

## 1. Einleitung

Hammerschach-BoardLab ist ein browserbasiertes Werkzeug zum Aufbau, Prüfen und Analysieren einzelner Schachstellungen. Im Mittelpunkt stehen das freie Setzen von Figuren, die Umwandlung einer Stellung in eine spielbare Partie und die anschließende Engine-Analyse direkt im Browser.

Das BoardLab ergänzt einen Werkzeugkasten aus Analyse- und Trainingsprogrammen: Während ein PGN-Analyser vollständige Partien untersucht und ein Taktiktrainer konkrete Aufgaben stellt, dient BoardLab als Labor für einzelne Stellungen, Varianten und Trainingssituationen.

1. **Stellungsaufbau:** Figuren werden über eine Palette auf das Brett gesetzt oder wieder entfernt.
2. **Play & Analyze:** Die aufgebaute Stellung wird gesperrt und als spielbare Ausgangslage verwendet.
3. **Engine-Arbeit:** Stockfish analysiert die Stellung, zeigt Bewertung, Bestzug, WDL und Varianten an.
4. **Training gegen Engine:** Die Engine kann eine Seite übernehmen, während die Lösung verdeckt bleibt.

### 1.1 Zielgruppe

- Trainer, die Unterrichtsstellungen schnell aufbauen und mit Engine-Hinweisen prüfen möchten.
- Spieler, die kritische Momente aus eigenen Partien nachstellen und Varianten testen wollen.
- Vereine, die ein kompaktes Stellungs- und Analysewerkzeug in eine Webseite einbinden möchten.
- Fortgeschrittene Nutzer, die mit FEN, Rochaderechten, Zugnummern und Engine-Parametern arbeiten.

### 1.2 Grundidee des Arbeitsablaufs

Der typische Ablauf besteht aus drei Phasen: Zuerst wird die gewünschte Stellung aufgebaut. Danach wird sie mit Play & Analyze in den Spielmodus übernommen. Anschließend können Züge eingegeben, Varianten vor- und zurückgespielt oder Engine-Empfehlungen ausgewertet werden.

#### Merksatz

Aufbauen, verriegeln, analysieren: BoardLab trennt den freien Aufbaumodus bewusst vom spielbaren Analysemodus.

## 2. Bedienung

### 2.1 Aufbau der Oberfläche

Die Oberfläche ist in eine obere Aufbauleiste, den Brettbereich und eine Analyse-Seitenleiste gegliedert. Auf breiten Bildschirmen steht die Analyse rechts neben dem Brett; auf kleineren Bildschirmen wird die Ansicht responsiv untereinander angeordnet.



Hauptbereiche der BoardLab-Oberfläche	
<b>Aufbauleiste</b>	Logo, Startstellung, Alles löschen, Play & Analyze, Figurenpalette, Zugrecht, En-passant, Halbzug, Zugnummer und Rochaderechte.
<b>Brett</b>	Interaktives 8x8-Schachbrett mit Koordinaten, Figuren, Markierungen und illegaler Zugrückmeldung.
<b>Navigationsleiste</b>	Anfang, zurück, vor, Ende, Rückgängig und Flip.
<b>Statusfeld</b>	Zeigt die Seite am Zug sowie Hinweise zu illegalen Eingaben oder gesperrtem Aufbaumodus.
<b>Engine-Box</b>	Auto-Analyse, manueller Analysestart, Suchmodus, Tiefe, Zeit, MultiPV, Bewertung, WDL, Tiefe, NPS, Bestzug und Varianten.
<b>Zugliste</b>	Zeigt die gespielten Züge der aktuellen Linie in einer kompakten Notationsansicht.

## 2.2 Aufbaumodus verwenden

Im Aufbaumodus ist die Stellung frei editierbar. Die Figurenpalette ist in weiße und schwarze Figuren aufgeteilt; zusätzlich steht ein Leere-Feld-Werkzeug zur Verfügung. Ein Klick wählt eine Figur aus, anschließend wird diese Figur auf das gewünschte Feld gesetzt. Per Drag & Drop kann ebenfalls gearbeitet werden, sofern der Browser dies unterstützt.

- Bei Bedarf mit **Alles löschen** ein leeres Brett erzeugen oder mit **Startstellung** die normale Ausgangsstellung laden.
- Eine Figur in der Palette auswählen; die Auswahl wird in der Aufbauleiste angezeigt.
- Auf ein Feld klicken, um die Figur zu platzieren, oder das Leere-Feld-Werkzeug nutzen, um ein Feld zu räumen.
- Zugrecht, En-passant-Feld, Halbzugzähler, Zugnummer und Rochaderechte setzen.
- Mit **Play & Analyze** die Stellung in den spielbaren Analysemodus übernehmen.

### Wichtig

Nach Play & Analyze ist der Aufbau gesperrt. Änderungen an Figuren und Metadaten sind dann nicht mehr direkt im Aufbaumodus möglich, damit die laufende Analyseposition eindeutig bleibt.

## 2.3 Stellungseigenschaften richtig setzen

Die Metadaten bestimmen, wie die Stellung regeltechnisch behandelt wird. Sie sind besonders wichtig, wenn Spezialrechte oder Zählwerte einer Partie erhalten bleiben sollen.

Feld	Bedeutung	Praxis-Hinweis
Am Zug	Legt fest, ob Weiß oder Schwarz zieht.	Muss zur Stellung passen, sonst bewertet und spielt die Engine aus falscher Perspektive.
En-passant	Speichert ein mögliches En-passant-Zielfeld.	Meistens ist "-" korrekt; nur nach einem Doppelschritt relevant.
Halbzug	Halbzugzähler für die 50-Züge-Regel.	Für reine Analyse oft 0, für exakte Rekonstruktion anpassen.
Zugnr.	Vollzugnummer der Partie.	Für Notation und Variantenanzeige sinnvoll.
Rochaderecht	Separat für Weiß kurz/lang und Schwarz kurz/lang.	Nur aktivieren, wenn König und Turm noch rochieren dürfen.



## 2.4 Play & Analyze

Mit Play & Analyze wird die aktuelle Aufbauposition zur Ausgangsstellung der Analyse. Ab diesem Moment können Züge legal gespielt, in der Zugliste verfolgt und mit der Engine bewertet werden. Die Schaltfläche ist damit der Übergang vom freien Bretttaufbau zum kontrollierten Schachmodus.

- Nur legale Züge der Seite am Zug werden akzeptiert.
- Eine ausgewählte Figur und legale Zielfelder werden visuell hervorgehoben.
- Illegale Eingaben erzeugen eine optische Rückmeldung und eine Statusmeldung.
- Bei Bauernumwandlung öffnet sich ein Dialog für Dame, Turm, Läufer oder Springer.

## 2.5 Zugnavigation und Brettsteuerung

Die Navigationsleiste arbeitet mit der aktuellen Zughistorie. Dadurch können Nutzer Varianten schrittweise durchgehen, an den Anfang springen oder die letzte Eingabe zurücknehmen.

Navigations- und Bretttasten	
«	Springt zur Ausgangsstellung der aktuellen Linie.
<	Geht einen Halbzug zurück.
>	Geht einen Halbzug vor.
»	Springt ans Ende der gespielten Linie.
Rückgängig	Entfernt den letzten gespielten Halbzug aus der Linie.
Flip	Dreht das Brett zwischen weißer und schwarzer Perspektive.

## 2.6 Engine-Analyse

Die Engine-Box kann automatisch oder manuell analysieren. Auto-Analyse startet nach relevanten Stellungsänderungen von selbst. Mit Analysieren wird die aktuelle Stellung gezielt ausgewertet; Stop beendet eine laufende Suche.

Option	Funktion
Auto-Analyse	Aktualisiert die Analyse automatisch nach Zug- oder Stellungsänderungen.
Suchmodus Tiefe	Analysiert bis zu einer eingestellten Suchtiefe.
Suchmodus Zeit	Analysiert für eine vorgegebene Zeit in Millisekunden.
MultiPV	Zeigt eine oder mehrere Hauptvarianten, maximal vier.
Bewertung	Numerische Einschätzung aus Sicht der Seite am Zug oder Mattangabe.
WDL	Win-Draw-Loss-Einschätzung, sofern von der Engine geliefert.
Bestzug	Von der Engine als stärkster Zug erkannter Zug.
Variantenliste	Anzeige der Hauptvariante(n) in lesbarer SAN-Notation.

### Interpretation

Eine Engine-Bewertung ist kein Ersatz für schachliches Verstehen. Sie zeigt Prioritäten und taktische Ressourcen, erklärt aber nicht automatisch den Plan hinter einer Stellung.

## 2.7 Training gegen die Engine

Der Trainingsmodus wird über Training gegen Engine aktiviert. Anschließend kann gewählt werden, ob die Engine Weiß oder Schwarz spielt. Wenn der Mensch am Zug ist, bleibt die Engine-Lösung verdeckt; nach einem eigenen Zug kann die Lösung bei Bedarf eingeblendet werden.



- Die Engine-Seite zieht automatisch, sobald sie am Zug ist.
- Während des Trainings werden Bewertung, Bestzug und Varianten verdeckt, bis Lösung zeigen gedrückt wird.
- Die Rückmeldung vergleicht den gespielten Zug mit den Engine-Vorschlägen, sofern eine Analyse der Ausgangsstellung vorliegt.
- Mit Wieder verdecken kann die Trainingsansicht wieder auf verdeckte Analyse zurückgestellt werden.

## 2.8 Typische Arbeitsbeispiele

Ziel	Vorgehen
Taktikstellung prüfen	Stellung aufbauen, Play & Analyze drücken, Auto-Analyse laufen lassen und Bestzug/Varianten vergleichen.
Endspielstellung trainieren	Material und Zugrecht setzen, Rochaderechte deaktivieren, Engine-Seite wählen und verdeckt gegen die Engine spielen.
Partiemoment rekonstruieren	Figuren setzen, Zugnummer und Seite am Zug eintragen, anschließend die kritische Variante durchspielen.
Unterrichtsvorbereitung	Mehrere Kandidatenzüge am Brett testen und die Engine-PV als Kontrollhinweis verwenden.

## 3. Technik

### 3.1 Architektur

BoardLab ist als HTML-Anwendung umgesetzt. Struktur, Gestaltung und Programmlogik liegen im Web-Umfeld des Browsers. Die Oberfläche nutzt CSS-Variablen für Größen, Akzentfarben und responsive Bereiche; die Logik wird in JavaScript ausgeführt.

Technische Bausteine	
HTML	Definiert Aufbauleiste, Brett, Bedienfelder, Engine-Panel, Zugliste und Dialoge.
CSS	Steuert DIN-ähnlich klare Proportionen im Browser, Farben, Schatten, responsive Layouts und Brettmarkierungen.
JavaScript	Verwaltet Stellung, Legalität, Historie, Rendering, Engine-Kommunikation und Trainingsmodus.
Bilddateien	Logo und Figurenbilder werden als externe Assets geladen.
Engine-Dateien	stockfish.js und stockfish.wasm müssen erreichbar sein, damit die Engine startet.

### 3.2 Stellung, FEN und Zuglogik

Intern wird aus Brettbelegung, Zugrecht, Rochaderechten, En-passant-Feld, Halbzugzähler und Zugnummer eine FEN-Stellung gebildet. Diese FEN dient als stabile Schnittstelle zwischen Brettdarstellung, Zuglogik und Engine-Analyse.

- Die Brettkoordinaten werden aus Sicht der aktuellen Orientierung gerendert.
- Legalität wird vor der Zugausführung geprüft; nur gültige Züge verändern die Historie.
- Die Zugliste entsteht aus der gespielten Linie und kann über die Navigation angesprungen werden.
- Engine-Markierungen werden nur angezeigt, wenn der aktuelle FEN-Schlüssel zur zuletzt berechneten Engine-Ausgabe passt.

### 3.3 Stockfish-Anbindung

Die Engine wird als Web Worker geladen. Vor dem Start prüft BoardLab, ob stockfish.js und stockfish.wasm erreichbar sind. Danach erfolgt ein UCI-Handshake; erst nach uciok und readyok gilt die Engine als bereit.



UCI-/Engine-Aspekt	Umsetzung im BoardLab
Worker	Stockfish läuft getrennt von der Oberfläche, damit die Bedienung reaktionsfähig bleibt.
UCI_ShowWDL	Aktiviert die Ausgabe der WDL-Informationen, wenn die Engine sie unterstützt.
MultiPV	Wird anhand der Auswahl auf einen Wert zwischen 1 und 4 gesetzt.
Threads	Orientiert sich an der vom Browser gemeldeten HardwareConcurrency, begrenzt auf maximal 8.
Hash	Wird auf 64 MB gesetzt.
Suche	Je nach Modus wird mit Tiefe oder Zeitlimit gearbeitet.

### 3.4 Analyseausgabe und Trainingslogik

Engine-Informationen werden aus UCI-info-Zeilen gelesen. Dazu gehören Tiefe, Suchtiefe, MultiPV-Nummer, Nodes, NPS, Hashfüllung, Zeit, Score, Mattangaben, WDL und PV-Züge. Die PV wird in lesbare Züge umgerechnet, damit die Variantenliste nicht nur UCI-Koordinaten zeigt.

Im Trainingsmodus werden dieselben Engine-Daten bewusst verborgen. Erst wenn der Nutzer die Lösung anzeigen lässt oder nach einem eigenen Zug eine Bewertung verfügbar ist, wird die Engine-Information genutzt, um den Zug einzuordnen.

### 3.5 Browser- und Bereitstellungshinweise

- Die Anwendung benötigt einen modernen Browser mit Unterstützung für Web Worker und WebAssembly.
- Die Dateien stockfish.js und stockfish.wasm müssen im gleichen Veröffentlichungsumfeld oder unter korrekten Pfaden erreichbar sein.
- Bei lokalen Dateifreigaben können Browser-Sicherheitsregeln das Laden von Worker- oder WASM-Dateien verhindern; eine Bereitstellung über einen Webserver ist zuverlässiger.
- Figurenbilder und Logo sollten gemeinsam mit der HTML-Datei ausgeliefert werden, damit die Oberfläche vollständig dargestellt wird.
- Die Engine-Leistung hängt von Gerät, Browser, verfügbarer CPU-Leistung und eingestellter Tiefe beziehungsweise Suchzeit ab.

### 3.6 Fehlerbilder und schnelle Diagnose

Meldung oder Verhalten	Mögliche Ursache	Maßnahme
Engine wird geladen bleibt stehen	Engine-Dateien nicht erreichbar oder Worker blockiert.	Pfade zu stockfish.js und stockfish.wasm prüfen, über Webserver testen.
Dateien nicht erreichbar	Fetch auf Engine-Dateien schlägt fehl.	Dateinamen, Groß-/Kleinschreibung und Veröffentlichungsort prüfen.
Keine Antwort vom Worker	Worker startet, beantwortet den UCI-Handshake aber nicht.	Browser-Konsole öffnen und Engine-Kompatibilität prüfen.
Zug wird nicht angenommen	Falsche Seite am Zug, illegaler Zug oder Aufbaumodus nicht verriegelt.	Zugrecht prüfen, Play & Analyze ausführen und legalen Zug versuchen.
Rochade nicht möglich	Rochaderecht nicht gesetzt oder Figurenstellung nicht regelgerecht.	König, Turm, Zwischenfelder und Checkboxes prüfen.

## 4. Resümee

Hammerschach-BoardLab ist ein spezialisiertes Schachlabor für Stellungen. Es konzentriert sich auf das schnelle Aufbauen, regelgerechte Durchspielen und enginegestützte Auswerten einzelner Positionen. Dadurch schließt es die Lücke zwischen vollständiger Partienanalyse und klassischem Taktiktraining.



Die klare Trennung zwischen Aufbaumodus und Play-&-Analyse-Modus ist der wichtigste Bediengedanke. Sie sorgt dafür, dass Nutzer frei konstruieren können, die eigentliche Analyse anschließend aber auf einer stabilen, reproduzierbaren Ausgangsstellung basiert.

Besonders stark ist BoardLab, wenn es als Werkzeug für Unterricht, Vereinsabende und Eigenanalyse eingesetzt wird: Eine Stellung wird aufgebaut, Kandidatenzüge werden diskutiert, die Engine liefert Kontrollvarianten und der Trainingsmodus ermöglicht verdecktes Üben gegen die Maschine.

### **Kurzfasit**

BoardLab macht aus einer einzelnen Schachstellung ein interaktives Analyse- und Trainingslabor - kompakt, webbasiert und gut in das Hammerschach-Toolset integrierbar.

## **4.1 Empfohlene Nutzung im Sammelwerk**

Im Sammelwerk der Hammerschach-Tools ergänzt dieses Handbuch die vorhandenen Handbücher sinnvoll: Der Analyzer behandelt gespeicherte Partien, der Trainer behandelt Aufgaben und BoardLab behandelt frei aufgebaute Stellungen. Zusammen decken die Handbücher damit Analyse, Training und Stellungsarbeit ab.

# Handbuch

## Hammerschach-Open-Theme

*Eröffnungsthemen, Ideen, Nachspielen und Engine-Analyse*





## Inhaltsverzeichnis

1. Einleitung
2. Bedienung
3. Technik
4. Resümee

### Dokumentaufbau

Dieses Handbuch beschreibt Hammerschach-Open-Theme aus Anwendersicht und ergänzt die Bedienung um technische Hinweise zur HTML-, CSS- und JavaScript-Umsetzung. Das Layout ist auf DIN A4 Hochformat optimiert.

## 1. Einleitung

Hammerschach-Open-Theme ist ein browserbasiertes Werkzeug zum strukturierten Lernen, Nachspielen und praktischen Ausspielen von Schacheröffnungen. Die Anwendung lädt vorbereitete Eröffnungsthemen aus einer Datei PGN.txt, stellt die vorgegebene Zugfolge am Brett dar und erlaubt anschließend eigene Fortsetzungen mit Engine-Unterstützung.

Im Unterschied zu einem allgemeinen PGN-Analyzer steht nicht die freie Verwaltung beliebiger Partien im Vordergrund, sondern ein kuratierter Eröffnungsbestand mit erklärenden Lernfeldern. Damit eignet sich Open-Theme besonders für Vereinsabende, Eröffnungstraining, Hausaufgaben, Thematic-Partien, Thematic-Turniere und persönliche Repertoirearbeit.

1. **Themen auswählen:** Eine Eröffnung wird aus der Eröffnungsliste gewählt oder zufällig geladen.
2. **Ideen verstehen:** Der Ideendialog zeigt Lerntexte zu Sinn, ersten Zügen, Plänen, Motiven und Warnungen.
3. **Linien nachspielen:** Die vorgegebene Eröffnungsfolge wird am Brett gezeigt; danach können eigene Vereinszüge ergänzt werden.
4. **Thematic-Partien spielen:** Die ausgeloste oder gewählte Eröffnung wird auf dem Brett aufgebaut; ab dieser Stellung beginnt die eigentliche Partie.
5. **Turniere organisieren:** Für mehrere Bretter kann dieselbe oder eine zufällig gewählte Startstellung vorgegeben werden.
6. **Stellungen prüfen:** Die Engine-Analyse bewertet die aktuelle Stellung und zeigt Bestzug sowie Hauptvarianten.

### 1.1 Zielgruppe

- Spieler, die ein Eröffnungsrepertoire nicht nur auswendig lernen, sondern mit Plänen und Motiven verbinden möchten.
- Trainer, die vorbereitete Themen in einer einheitlichen Oberfläche präsentieren wollen.
- Turnierleiter, die Thematic-Partien oder Thematic-Turniere mit vorgegebenen Eröffnungsstellungen durchführen möchten.
- Vereine, die eine gemeinsame Sammlung aus Eröffnungen und Erklärtexten pflegen möchten.
- Fortgeschrittene Nutzer, die Varianten nach der Startfolge selbst weiterführen und mit Stockfish kontrollieren möchten.

### 1.2 Grundidee des Workflows

Der typische Ablauf lautet: PGN.txt bereitstellen, Eröffnung wählen, Idee lesen, Zugfolge nachspielen, eigene Fortsetzung eingeben, Engine-Auswertung prüfen und den eigenen Verlauf als PGN speichern. Der wichtigste Bediengedanke ist die Trennung zwischen vorgegebener Thema-Linie und anschließend selbst gespielten Zügen.



### 1.3 Einsatz als Thematic-Partie oder Turnier

Open-Theme kann nicht nur zum Lernen, sondern auch als Startgenerator für Thematic-Partien eingesetzt werden. Dabei wird eine Eröffnung gezielt oder zufällig ausgewählt, die vorgegebenen Züge werden auf dem Brett aufgebaut und die Partie beginnt erst ab dieser Themenstellung. Die Spieler erhalten dadurch eine identische oder bewusst geloste Ausgangslage und müssen die entstehende Stellung praktisch behandeln.

Für ein Thematic-Turnier kann die Turnierleitung zum Beispiel pro Runde eine Eröffnung auslösen, diese Stellung auf allen Brettern aufbauen lassen und die Partien ab genau dieser Stellung starten. Nach der Partie können die gespielten Fortsetzungen wieder in Open-Theme geladen, nachgespielt und mit der Engine analysiert werden.

Praktischer Nutzen im Spielbetrieb	
<b>Thematic-Partie</b>	Eine einzelne Partie beginnt nicht aus der Grundstellung, sondern nach der vorgegebenen Eröffnungsfolge.
<b>Thematic-Turnier</b>	Mehrere Bretter starten mit derselben ausgelosten oder vorgegebenen Eröffnungsstellung.
<b>Zufallswahl</b>	Die Funktion Zufällige Eröffnung erzeugt eine abwechslungsreiche Auswahl aus der geladenen PGN.txt.
<b>Nachbereitung</b>	Die anschließend gespielte Partie kann als PGN gespeichert, nachgespielt und mit der Engine ausgewertet werden.

#### Merksatz

Open-Theme verbindet Eröffnungswissen mit aktivem Spielbetrieb: Thema auslösen, Stellung aufbauen, Partie spielen, anschließend nachspielen und analysieren.

## 2. Bedienung

### 2.1 Aufbau der Oberfläche

Die Oberfläche ist in Kopfdatenbereich, Brettbereich, Steuerungsleiste, Eröffnungsliste, Ideendialog, Engine-Box und Zugliste gegliedert. Auf kleineren Bildschirmen werden Brett und Seitenleiste responsiv untereinander angeordnet.

Hauptbereiche der Open-Theme-Oberfläche	
<b>Kopfdatenbereich</b>	Logo sowie Felder wie Event, Site, Date, Round, White, Black, Result, ECO, OpeningTheme, ThemeStartPly und PlyCount.
<b>Brettbereich</b>	Grafisches Schachbrett mit Koordinaten, Figuren, Markierungen, legalen Zielfeldern und illegaler Zugrückmeldung.
<b>Themensteuerung</b>	Eröffnungsliste, Zufällige Eröffnung und Idee als zentrale Einstiegspunkte.
<b>Navigation</b>	Anfang, zurück, vor, Ende, Rückgängig, Flip und PGN speichern.
<b>Engine-Analyse</b>	Auto-Analyse, manuelle Analyse, Stop, Suchmodus, Tiefe, Zeit, MultiPV, Bewertung, WDL, Bestzug und Varianten.
<b>Zugliste</b>	SAN-Notation der vorgegebenen Eröffnung und der nachgespielten Fortsetzung mit Sprungfunktion.



## 2.2 Eröffnungsliste verwenden

Beim Start versucht Open-Theme, eine Datei PGN.txt zu laden. Aus den darin enthaltenen PGN-Blöcken wird die Eröffnungsliste aufgebaut. Über Eröffnungsliste öffnet sich ein Dialog mit Suchfeld. Gesucht werden kann nach Eröffnungsname, ECO, Kopfdaten oder Zugfolge.

7. PGN.txt im gleichen Veröffentlichungsumfeld wie die HTML-Datei bereitstellen.
8. Open-Theme im Browser öffnen und warten, bis die Eröffnungsliste geladen ist.
9. Auf **Eröffnungsliste** klicken.
10. Optional Suchbegriff eingeben, zum Beispiel Eröffnungsname, ECO-Code oder ein Zugfragment.
11. Gewünschte Zeile anklicken; die Eröffnung wird geladen und am Ende der Thema-Linie angezeigt.

### Hinweis

Wenn PGN.txt nicht erreichbar ist oder keine gültigen Eröffnungsthemen enthält, zeigt Open-Theme eine Meldung an und setzt das Feld OpeningTheme auf einen Ladehinweis.

## 2.3 Zufällige Eröffnung und Thematic-Start

Mit Zufällige Eröffnung wählt die Anwendung ein Thema aus der geladenen Sammlung. Ist mehr als ein Thema vorhanden, wird möglichst nicht erneut dieselbe Auswahl geladen. Diese Funktion ist nützlich für Wiederholungstraining, spontane Vereinsübungen, Thematic-Partien und Thematic-Turniere.

Nach der Auswahl wird die Eröffnungsfolge automatisch als aktuelle Zughistorie übernommen. Das Brett steht am Ende der vorgegebenen Zugfolge; genau dort kann die eigentliche Partie beginnen. Für den Spielbetrieb bedeutet das: Die Eröffnung wird nicht mehr am Brett diskutiert oder manuell rekonstruiert, sondern als fertige Startstellung bereitgestellt.

12. Auf **Zufällige Eröffnung** klicken oder gezielt eine Eröffnung aus der Liste wählen.
13. Die angezeigte Zugfolge kurz kontrollieren und die Stellung auf dem realen Brett nachbauen oder digital übernehmen.
14. Spieler starten die Partie ab dieser Stellung; die Seite am Zug ergibt sich aus der Länge der Thema-Linie.
15. Nach Partieende die gespielte Fortsetzung in Open-Theme nachspielen oder als PGN speichern.
16. Mit Engine-Analyse und MultiPV die kritischen Entscheidungen nach der Themenstellung auswerten.

## 2.4 Ideendialog

Der Button Idee öffnet einen erklärenden Dialog zur aktuellen Eröffnung. Zuerst wird die vorgegebene Zugfolge angezeigt. Danach erscheinen die Lernabschnitte, sofern sie in den PGN-Kopfdaten vorhanden sind.

PGN-Feld	Anzeige im Ideendialog	Zweck
ThemeIdea	Sinn der Eröffnung	Erklärt den grundlegenden Gedanken des Systems.
ThemeMoves	Idee der ersten Züge	Beschreibt, warum die Startzüge gespielt werden.
WhitePlan	Plan für Weiß	Zeigt typische Aufbau- und Angriffspläne für Weiß.
BlackPlan	Plan für Schwarz	Zeigt typische Gegenpläne und Entwicklungsideen für Schwarz.
TypicalMotifs	Typische Motive	Sammelt taktische oder strategische Muster.
Warning	Worauf man achten muss	Weist auf Fallen, Fehlgriffe oder kritische Abweichungen hin.
ThemeSources	Quellen / Grundlage	Dokumentiert die Herkunft oder Grundlage des Themas.



## 2.5 Kopfdaten verstehen

Die Kopfdaten werden oben in der Oberfläche angezeigt. Einige Felder sind frei nutzbar, andere werden aus dem gewählten Eröffnungsthema abgeleitet. White, Black und Result werden für das eigene Nachspielen standardmäßig neutral geführt.

Feld	Bedeutung	Hinweis
ECO	Eröffnungscode aus dem PGN-Thema.	Wird als Orientierung in Liste und Kopfbereich angezeigt.
OpeningTheme	Name des aktuellen Eröffnungsthemas.	Kann aus OpeningTheme, Variation, Opening oder Event stammen.
ThemeStartPly	Anzahl der vorgegebenen Halbzüge.	Markiert den Übergang von Thema-Linie zu eigenen Zügen.
PlyCount	Gesamte Halbzüge der aktuellen Linie.	Wird nach eigenen Fortsetzungen aktualisiert.
White / Black	Spielernamen im exportierten PGN.	In Open-Theme standardmäßig neutral, da der Fokus auf der Eröffnung liegt.

## 2.6 Züge am Brett eingeben

Nach dem Laden einer Eröffnung steht das Brett am Ende der vorgegebenen Thema-Linie. Ab dort können eigene Züge am Brett eingegeben werden. Eine Figur wird angeklickt, danach das Zielfeld. Nur legale Züge der Seite am Zug werden angenommen.

- Ausgewählte Figuren und legale Zielfelder werden markiert.
- Illegale Züge werden nicht ausgeführt und lösen eine kurze visuelle Rückmeldung aus.
- Rochade, en passant, Schach, Matt, Patt und Bauernumwandlung werden regeltechnisch behandelt.
- Wenn an einer früheren Stellung ein neuer Zug eingegeben wird, wird die bisherige Fortsetzung ab dieser Stelle ersetzt.

## 2.7 Zugliste und Markierungen

Die Zugliste unterscheidet zwischen der vorgegebenen Eröffnung und der eigenen Fortsetzung. Vorgegebene Eröffnungszüge sind optisch anders markiert; der letzte Halbzug der Thema-Linie bildet die Grenze. Danach zählt die Anwendung die nachgespielten Vereinszüge.

Lesart der Zugliste	
Vorgegebene Eröffnung	Die Halbzüge bis ThemeStartPly stammen aus dem gewählten PGN-Thema.
Thema-Ende	Der letzte vorgegebene Halbzug ist besonders markiert und zeigt den Startpunkt für eigenes Training.
Vereinszüge	Alle danach gespielten Halbzüge sind die eigene Fortsetzung.
Sprungfunktion	Ein Klick auf einen Zug springt zur Stellung nach diesem Halbzug.

## 2.8 Navigation, Rückgängig und Flip

Steuerung am Brett	
<<	Zur Startstellung der aktuellen Linie springen.
<	Einen Halbzug zurückgehen.
>	Einen Halbzug vorgehen.
>>	Ans Ende der aktuellen Linie springen.
Rückgängig	Letzten Zug entfernen oder eine Fortsetzung abschneiden.
Flip	Brett zwischen weißer und schwarzer Perspektive drehen.



<b>Steuerung am Brett</b>	
PGN speichern	Aktuelle nachgespielte Partie als PGN-Datei herunterladen.

## 2.9 Engine-Analyse

Die Engine analysiert die aktuell angezeigte Stellung. Bei aktiver Auto-Analyse wird nach Stellungswechseln automatisch eine neue Suche angefordert. Bei deaktivierter Auto-Analyse startet Analysieren die Suche manuell; Stop beendet sie.

Option	Funktion
Auto-Analyse	Aktualisiert die Bewertung nach Stellungswechseln automatisch.
Suchmodus Tiefe	Rechnet bis zur gewählten Tiefe.
Suchmodus Zeit	Rechnet für eine vorgegebene Zeit in Millisekunden.
MultiPV	Zeigt bis zu vier Hauptvarianten.
Bewertung	Einschätzung der Stellung, einschließlich Mattangaben, sofern vorhanden.
WDL	Win-Draw-Loss-Werte, wenn die Engine diese liefert.
Bestzug	Beste Engine-Empfehlung in SAN und UCI, soweit umsetzbar.

### Praxis-Tipp

Erst die Idee lesen, dann ohne Engine nachspielen und erst anschließend die Bewertung einschalten. So bleibt der Trainingseffekt größer.

## 2.10 Speichern

PGN speichern lädt die aktuell nachgespielte Linie als Datei herunter. Dabei werden die Kopfdaten des gewählten Themas und die eigene Fortsetzung in eine PGN-Struktur geschrieben. Für eine dauerhafte Sammlung sollte die erzeugte Datei anschließend sauber abgelegt oder in die zentrale PGN.txt übernommen werden.

## 2.11 Empfohlener Trainingsablauf

17. Eröffnung aus der Liste oder zufällig laden.
18. Ideendialog öffnen und die wichtigsten Pläne lesen.
19. Startzüge am Brett kurz zurück- und vorspielen.
20. Am Thema-Ende ohne Engine eine eigene Fortsetzung eingeben.
21. Engine-Analyse aktivieren und Bestzug sowie Varianten vergleichen.
22. Bei Bedarf Zugfolge korrigieren oder eine zweite Fortsetzung testen.
23. Die finale Fortsetzung als PGN speichern.

## 2.12 Ablauf für Thematic-Partien und Turniere

Im Spielbetrieb steht nicht die Erklärung, sondern die einheitliche Startstellung im Vordergrund. Open-Theme liefert dafür die vorgegebene Eröffnung und markiert mit ThemeStartPly, wo die Theorie endet und die eigentliche Partie beginnt.

Schritt	Durchführung	Ziel
1. Thema bestimmen	Eröffnung gezielt auswählen oder per Zufall laden.	Gleiche oder bewusst geloste Ausgangsbasis schaffen.
2. Stellung aufbauen	Züge bis ThemeStartPly auf dem Brett nachspielen oder die digitale Stellung anzeigen.	Alle Bretter starten identisch.



Schritt	Durchführung	Ziel
3. Partie spielen	Ab der angezeigten Stellung normal weiterspielen.	Praktische Behandlung der Eröffnungsidee testen.
4. Partie erfassen	Fortsetzung in Open-Theme nachspielen oder als PGN speichern.	Nachbereitung ermöglichen.
5. Analyse	Engine, Bestzug und Varianten nutzen.	Kritische Momente nach dem Thematic-Start verstehen.

### Turnier-Hinweis

Für ein Thematic-Turnier sollte die ausgewählte Eröffnung vor Rundenbeginn eindeutig bekanntgegeben werden, zum Beispiel mit Eröffnungsname, Zugfolge und Seite am Zug. So vermeiden alle Bretter unterschiedliche Startstellungen.

## 3. Technik

### 3.1 Architektur

Open-Theme ist als HTML-, CSS- und JavaScript-Anwendung aufgebaut. Die HTML-Struktur enthält Kopfdaten, Brett, Schaltflächen, Dialoge, Engine-Anzeige und Zugliste. CSS steuert Layout, Farben, Responsivität und Markierungen. JavaScript übernimmt PGN-Verarbeitung, Eröffnungsauswahl, Schachregeln, Rendering, Export und Engine-Kommunikation.

Technische Bausteine	
HTML	Definiert Kopfbereich, Brett, Seitenleiste, Modale, Engine-Box und Zugliste.
CSS	Setzt Times New Roman, Akzentfarben, Brettlayout, responsive Darstellung, Modale und Markierungen.
JavaScript	Verarbeitet PGN.txt, erzeugt Eröffnungsthemen, prüft Züge, rendert Stellungen und steuert Stockfish.
PGN.txt	Externe Datendatei mit den vorbereiteten Eröffnungsthemen und Lernfeldern.
stockfish.js / stockfish.wasm	Engine-Dateien, die für die Analyse als Web Worker und WebAssembly erreichbar sein müssen.

### 3.2 PGN.txt und Themenmodell

Die Datei PGN.txt wird beim Start mit no-store-Cache geladen. Anschließend trennt die Anwendung die Inhalte anhand von Event-Blöcken, liest PGN-Kopfdaten und Zugtext und wandelt jede gültige Partie in ein Eröffnungsthema um. Ein Thema ist nur gültig, wenn daraus mindestens ein Zug gelesen werden kann.

Element	Technische Rolle
openingThemes	Liste aller geladenen Eröffnungsthemen.
selectedOpeningIndex	Index des aktuell gewählten Themas.
masterHistory	Gesamte Zugfolge aus Thema-Linie und eigener Fortsetzung.
themeStartPly	Grenze zwischen vorgegebenem Thema und eigenen Zügen.
headers	PGN-Kopfdaten einschließlich Lernfeldern und Anzeigeeinformationen.
viewIndex	Aktuell angezeigte Stellung innerhalb der Zughistorie.

### Datenprinzip



Open-Theme speichert die eigentliche Trainingsbibliothek nicht im Browser, sondern erwartet eine vorbereitete PGN.txt. Änderungen an der Sammlung erfolgen daher außerhalb der Anwendung oder durch spätere Übernahme exportierter PGNs.

### 3.3 PGN-Felder für Lerninhalte

Neben klassischen PGN-Feldern nutzt Open-Theme zusätzliche Kopfdaten für den Eröffnungsunterricht. Diese Felder erscheinen nicht zwingend in jeder PGN-Datei, werden aber vom Ideendialog ausgewertet, sobald sie vorhanden sind.

- OpeningTheme benennt das Thema; als Ersatz können Variation, Opening oder Event verwendet werden.
- ThemeStartPly wird aus der Länge der vorgegebenen Zugfolge gesetzt.
- ThemeIdea, ThemeMoves, WhitePlan, BlackPlan, TypicalMotifs, Warning und ThemeSources füllen den Ideendialog.
- PlyCount wird im Kopfbereich aus der aktuellen Gesamtlänge der Linie aktualisiert.

### 3.4 Schachlogik und Notation

Die Anwendung arbeitet intern mit einer Game-Struktur. Das Brett wird als 8x8-Struktur geführt, Züge werden legal erzeugt und im Verlauf gespeichert. Für die Anzeige wird SAN-Notation gebildet; für die Engine wird aus der aktuellen Stellung FEN erzeugt.

Technik	Nutzen
Legalitätsprüfung	Nur regelkonforme Züge können die Linie verändern.
SAN-Erzeugung	Züge werden lesbar als e4, Nf3, O-O oder Qh5+ angezeigt.
FEN-Erzeugung	Die Engine erhält die aktuelle Stellung in standardisierter Form.
Historie	Navigation und Rückgängig arbeiten über die gespeicherte Halbzugliste.
Orientierung	Flip verändert nur die Darstellung, nicht die Stellung.

### 3.5 Stockfish-Anbindung

Die Engine startet als Web Worker über stockfish.js. Vor dem Start prüft die Anwendung, ob stockfish.js und stockfish.wasm erreichbar sind. Danach läuft der UCI-Handshake. Sobald uciok und readyok gemeldet wurden, kann die aktuelle FEN analysiert werden.

Engine-Aspekt	Umsetzung in Open-Theme
Worker	Die Engine läuft getrennt von der Oberfläche, damit die Bedienung reaktionsfähig bleibt.
UCI_ShowWDL	Wird aktiviert, damit WDL-Werte angezeigt werden können.
MultiPV	Kann auf 1 bis 4 Varianten gesetzt werden.
Threads	Orientiert sich an HardwareConcurrency und wird begrenzt.
Hash	Der Hash-Speicher wird auf einen festen Wert gesetzt.
Suche	Je nach Modus mit Tiefenlimit oder Zeitlimit.

### 3.6 Technisches Prinzip des Thematic-Starts

Beim Laden eines Eröffnungsthemas erzeugt Open-Theme intern ein neues Spielobjekt aus der vorgegebenen Zugfolge. Die Länge dieser Zugfolge wird als ThemeStartPly gespeichert. Alle danach eingegebenen Züge sind die praktische Fortsetzung der Thematic-Partie. Genau dadurch kann dieselbe Oberfläche sowohl für Training als auch für echten Spielbetrieb genutzt werden.



Funktion	Bedeutung für Thematic-Partien
createGameFromOpeningTheme	Erzeugt eine neue Partie aus der gewählten Eröffnungsfolge.
selectOpeningTheme	Lädt das gewählte Thema und setzt die Anzeige auf die Startstellung nach der Theorie.
selectRandomOpening	Wählt ein Thema zufällig aus der geladenen Sammlung.
themeStartPly	Speichert, ab welchem Halbzug die gespielte Partie beginnt.
masterHistory	Enthält zunächst die Thema-Züge und später die real gespielte Fortsetzung.
buildPGNForGame	Exportiert die komplette Linie einschließlich Startthema und gespielter Fortsetzung.

### 3.7 Bereitstellung und Fehlerdiagnose

- PGN.txt, stockfish.js und stockfish.wasm müssen aus Sicht der veröffentlichten HTML-Datei erreichbar sein.
- Bei lokalen Dateiöffnungen können Browser-Sicherheitsregeln das Laden von PGN.txt, Worker oder WASM verhindern; ein Webserver ist zuverlässiger.
- Wenn die Eröffnungsliste leer bleibt, zuerst Pfad, Dateiname und PGN-Struktur der PGN.txt prüfen.
- Wenn die Engine nicht startet, Browser-Konsole, Dateipfade und WASM-Unterstützung prüfen.
- Die Engine-Leistung hängt von Gerät, Browser, CPU-Leistung, Tiefe, Suchzeit und MultiPV-Einstellung ab.

Meldung oder Verhalten	Mögliche Ursache	Maßnahme
PGN.txt nicht geladen	Datei fehlt, ist falsch benannt oder nicht erreichbar.	PGN.txt in denselben Ordner legen und Pfade prüfen.
Keine gültigen Eröffnungsthemen	PGN enthält keine lesbaren Züge oder keine passenden Event-Blöcke.	PGN-Struktur und Zugnotation prüfen.
Engine-Dateien nicht erreichbar	stockfish.js oder stockfish.wasm kann nicht geladen werden.	Dateinamen, Groß-/Kleinschreibung und Serverpfad prüfen.
Keine Antwort vom Worker	Worker startet nicht korrekt oder UCI-Handshake scheitert.	Konsole öffnen und Browser-/WASM-Kompatibilität prüfen.
Zug wird nicht angenommen	Falsche Farbe am Zug oder illegaler Zug.	Statusanzeige prüfen und legalen Zug wählen.

## 4. Resümee

Hammerschach-Open-Theme ist ein spezialisiertes Lern- und Spielwerkzeug für Eröffnungen. Es kombiniert eine vorbereitete PGN-Bibliothek, erklärende Lernfelder, interaktives Nachspielen, Thematic-Partien, Thematic-Turniere und enginegestützte Kontrolle in einer einheitlichen Browseroberfläche.

Der größte Nutzen liegt in der Verbindung von Theorie und Praxis: Nutzer sehen nicht nur eine Zugfolge, sondern erhalten Ideen, Pläne, Motive und Warnungen. Danach können sie eigene Fortsetzungen spielen oder eine echte Partie ab der Themenstellung beginnen. Die Qualität der Entscheidungen kann anschließend mit der Engine überprüft werden.

Im Sammelwerk der Hammerschach-Tools ergänzt Open-Theme die bisherigen Handbücher sinnvoll: Der Analyser behandelt vollständige Partien, der Trainer konkrete Aufgaben, BoardLab frei aufgebaute Stellungen und Open-Theme die strukturierte Eröffnungsarbeit.

#### Kurzfasit

Open-Theme macht Eröffnungstraining und Thematic-Spielbetrieb interaktiv: Thema laden oder auslösen, Stellung aufbauen, Partie spielen, anschließend nachspielen, analysieren und als PGN sichern.

# Handbuch

## Hammerschach-Freestyle-Analyzer

*Freestyle-Startstellungen, Turnierbetrieb, PGN und Engine-Analyse*





## Inhaltsverzeichnis

1. Einleitung
2. Bedienung
3. Technik
4. Resümee

### Dokumentaufbau

Dieses Handbuch beschreibt den Hammerschach-Freestyle-Analyzer aus Anwendersicht und ergänzt die Bedienung um technische Hinweise zur Freestyle-Startstellung, PGN-Verarbeitung, Rochade-Logik und Stockfish-Anbindung. Das Layout ist auf DIN A4 Hochformat optimiert.

## 1. Einleitung

Der Hammerschach-Freestyle-Analyzer ist ein browserbasiertes Werkzeug zum Spielen, Nachspielen und Analysieren von Freestyle- beziehungsweise Chess960-Partien. Im Mittelpunkt steht nicht die klassische Grundstellung, sondern eine von 960 möglichen Startreihen, die per Nummer aufgerufen oder zufällig erzeugt werden kann.

Damit eignet sich das Tool sowohl für spontanes Training als auch für echten Spielbetrieb. Mit dem Zufallsgenerator kann eine Freestyle-Startstellung bestimmt, auf einem realen Brett aufgebaut und anschließend als Partie gespielt werden. Nach Partieende kann die gespielte Partie im Analyzer nachgespielt, als PGN gespeichert oder mit der Engine ausgewertet werden.

- 1. Startstellung erzeugen:** Eine Freestyle-Nummer von 0 bis 959 eingeben oder per Zufall auswählen.
- 2. Reales Brett vorbereiten:** Die erzeugte Grundstellung auf dem physischen Brett aufbauen.
- 3. Freestyle-Partie spielen:** Ab dieser Stellung wird normal gespielt; die Rochaderegeln werden vom Tool technisch berücksichtigt.
- 4. Freestyle-Turniere durchführen:** Für mehrere Bretter kann dieselbe geloste Startstellung vorgegeben werden.
- 5. Partie nachspielen:** Züge im Analyzer eingeben, importieren oder per PGN wiederherstellen.
- 6. Engine prüfen:** Bewertung, Bestzug, WDL und Hauptvarianten für die aktuelle Stellung abrufen.

### 1.1 Zielgruppe

- Spieler, die Freestyle-Partien ohne Vorbereitungsvorteil austragen möchten.
- Vereine, die an Themenabenden oder Turnieren zufällig geloste Startstellungen verwenden wollen.
- Trainer, die Kreativität, Figurenkoordination und Orientierung in ungewohnten Startaufstellungen fördern möchten.
- Partieanalysten, die Chess960-PGNs mit FEN-Startstellung importieren und kontrollieren möchten.
- Fortgeschrittene Nutzer, die Stockfish-Auswertungen für ungewöhnliche Ausgangsstellungen benötigen.

### 1.2 Grundidee des Workflows

Der typische Ablauf lautet: Freestyle-Nummer wählen oder zufällig erzeugen, Stellung am Bildschirm und auf dem echten Brett kontrollieren, Partie spielen, gespielte Züge im Analyzer nachspielen, PGN speichern und kritische Momente mit der Engine untersuchen. Der Analyzer verbindet damit Startstellungs-Generator, Partieeditor und Analyse Brett in einer Oberfläche.

### 1.3 Einsatz als Freestyle-Partie oder Turnier

Besonders wichtig ist der Einsatz im praktischen Spielbetrieb. Der Zufallsgenerator wählt eine der 960 Startstellungen, die anschließend auf allen Brettern aufgebaut werden kann. Die Partie beginnt aus genau dieser Stellung. Dadurch entstehen Freestyle-Partien oder ganze Freestyle-Turniere, bei denen Eröffnungswissen weniger wichtig ist als Orientierung, Kreativität und saubere Analyse der ungewohnten Figurenstellung.



Nach der Runde können die Partien im Freestyle-Analyser rekonstruiert werden. Durch den PGN-Export mit Variant-, SetUp- und FEN-Angaben bleibt die konkrete Startstellung erhalten. So können die Partien später wieder geladen, nachgespielt und mit Stockfish analysiert werden.

Praktischer Nutzen im Freestyle-Spielbetrieb	
Freestyle-Partie	Eine Partie beginnt mit einer gelosten oder festgelegten Chess960-Startstellung statt mit der klassischen Grundstellung.
Freestyle-Turnier	Mehrere Bretter starten mit derselben vorgegebenen Zufallsstellung oder mit rundenweise neu ausgelosten Stellungen.
Zufallsgenerator	Der Button Zufall erzeugt eine Startnummer zwischen 0 und 959 und setzt das Brett entsprechend auf.
Nachbereitung	Die gespielten Züge werden nachgetragen, als PGN gesichert und anschließend mit Engine-Varianten ausgewertet.

### Merksatz

Freestyle-Analyser bedeutet: Stellung auslosen, reales Brett aufbauen, Partie spielen, danach im selben Tool nachspielen und analysieren.

## 2. Bedienung

### 2.1 Aufbau der Oberfläche

Die Oberfläche besteht aus Kopfleiste mit Logo und Freestyle-Nummer, Schachbrett, Steuerungsleiste, Statusanzeige, Engine-Box, Zugliste sowie Dialogen zum PGN-Import, PGN-Export und zur Bauernumwandlung. Die wichtigsten Arbeitsbereiche sind unmittelbar sichtbar, damit die Partie ohne zusätzliche Menüs geführt werden kann.

Hauptbereiche der Oberfläche	
Freestyle-Nummer	Eingabefeld für eine Zahl von 0 bis 959 sowie die Schaltflächen Anzeigen und Zufall.
Brettbereich	Grafisches Schachbrett mit Figuren, Koordinaten, Auswahlmarkierungen und legalen Zielfeldern.
Steuerungsleiste	PGN laden, Anfang, zurück, vor, Ende, Rückgängig, Flip und PGN erstellen.
Statuszeile	Zeigt Seite am Zug, Freestyle-Index und Partieende-Informationen.
Engine-Analyse	Auto-Analyse, manuelle Analyse, Stop, Suchmodus, Tiefe, Zeit, MultiPV, Bewertung, WDL und Bestzug.
Zugliste	SAN-Züge der gespielten Partie mit Sprungfunktion.

### 2.2 Freestyle-Nummer anzeigen

Über das Eingabefeld Freestyle-Nummer kann eine konkrete Startstellung gewählt werden. Zulässig sind Werte von 0 bis 959. Nach Klick auf Anzeigen wird die entsprechende Startreihe erzeugt, das Brett neu aufgebaut und die bisherige Zughistorie geleert.

1. Zahl zwischen 0 und 959 in das Feld Freestyle-Nummer eintragen.
2. Auf **Anzeigen** klicken oder im Eingabefeld die Eingabetaste verwenden.
3. Startaufstellung am Bildschirm kontrollieren.
4. Bei Nutzung am echten Brett die Figuren entsprechend aufbauen.
5. Partie ab dieser Stellung beginnen oder die Stellung als Trainingsaufgabe verwenden.



### Wichtig

Beim Wechsel der Freestyle-Nummer wird eine neue Startstellung gesetzt. Bereits eingegebene Züge gehören dann nicht mehr zu dieser Stellung und werden zurückgesetzt.

## 2.3 Zufallsgenerator für Spiel und Turnier

Der Button Zufall wählt automatisch eine Freestyle-Nummer zwischen 0 und 959. Diese Funktion ist für den Spielbetrieb besonders hilfreich, weil die Startstellung neutral, schnell und ohne lange Vorbereitung bestimmt wird.

Einsatz	Ablauf	Nutzen
Einzelpartie	Zufall drücken, Stellung aufbauen, Partie starten.	Schneller Start ohne Eröffnungsvorbereitung.
Vereinsabend	Eine Startnummer auslosen und auf allen Brettern verwenden.	Alle Spieler arbeiten an derselben ungewohnten Stellung.
Turnier	Pro Runde eine neue Startnummer bestimmen und dokumentieren.	Rundenweise neue Freestyle-Herausforderung.
Training	Mehrere Zufallsstellungen nacheinander testen.	Verbessert Orientierung und Figurenkoordination.

Für ein Freestyle-Turnier sollte die ausgeloste Nummer vor Rundenbeginn notiert und allen Brettern klar mitgeteilt werden. Nach der Partie kann der Analyzer wieder genutzt werden, um die entstandenen Partien nachzuspielen, in PGN-Form zu sichern und mit Engine-Varianten zu prüfen.

## 2.4 Züge am Brett eingeben

Nach dem Aufbau der Startstellung können Züge direkt am Brett ausgeführt werden. Eine Figur wird angeklickt, anschließend das Zielfeld. Der Analyzer akzeptiert nur legale Züge der Seite am Zug. Bauernumwandlungen öffnen einen eigenen Auswahldialog.

- Ausgewählte Figuren und legale Zielfelder werden visuell markiert.
- Rochaden werden gemäß der aktuellen Freestyle-Startstellung behandelt.
- En passant, Schach, Matt, Patt und Umwandlung werden regeltechnisch berücksichtigt.
- Neue Züge werden am Ende der aktuellen Linie gespeichert; nach Navigation an eine frühere Stelle ersetzt ein neuer Zug die bisherige Fortsetzung.

## 2.5 Navigation, Rückgängig und Flip

Steuerung am Brett	
PGN laden	Öffnet den Importdialog für PGN-Text inklusive optionaler FEN-Startstellung.
<<	Springt an den Anfang der aktuellen Partie.
<	Geht einen Halbzug zurück.
>	Geht einen Halbzug vor.
>>	Springt ans Ende der aktuellen Partie.
Rückgängig	Entfernt den letzten Zug oder schneidet die Fortsetzung ab.
Flip	Dreht die Brettansicht, ohne die Stellung zu verändern.
PGN erstellen	Öffnet den Dialog für Kopfdaten und PGN-Export.



## 2.6 PGN laden

Über PGN laden kann eine gespeicherte Partie wieder in den Analyzer übernommen werden. Der Importdialog erwartet PGN-Text mit Kopfzeilen und Zugtext. Für Freestyle-Partien ist eine FEN-Angabe besonders wichtig, weil nur so die konkrete Startstellung zuverlässig wiederhergestellt werden kann.

- PGN-Kopfdaten werden in die Eingabefelder des Exportdialogs übernommen, soweit sie unterstützt werden.
- Wenn SetUp = 1 und FEN vorhanden sind, wird die Startstellung aus der FEN-Zeile aufgebaut.
- Alternativ wird FEN auch in Verbindung mit Variant = Chess960 ausgewertet.
- SAN-Züge werden anschließend nacheinander legal auf der Startstellung ausgeführt.

### Praxis-Tipp

Bei selbst gespeicherten Freestyle-Partien immer die vom Analyzer erzeugten PGN-Tags Variant, SetUp und FEN beibehalten. Dadurch kann die Partie später korrekt rekonstruiert werden.

## 2.7 PGN erstellen und speichern

Der Exportdialog enthält klassische PGN-Felder wie Event, Site, Date, Round, White, Black, Result, ECO, WhiteElo, BlackElo und PlyCount. Beim Speichern ergänzt der Analyzer zusätzlich Freestyle-spezifische Informationen und lädt die PGN-Datei herunter.

Feld oder Tag	Bedeutung
PlyCount	Anzahl der gespielten Halbzüge; wird automatisch aus der Zughistorie gesetzt.
Variant "Chess960"	Kennzeichnet die Partie als Freestyle-/Chess960-Partie.
SetUp "1"	Zeigt an, dass die Partie aus einer gesonderten Startstellung beginnt.
FEN	Speichert die exakte Anfangsstellung einschließlich Seite am Zug, Rochaderechten und Zählern.
Result	Wird aus dem Exportdialog übernommen und ans Ende des Zugtexts gesetzt.

## 2.8 Zugliste und Status

Die Zugliste zeigt die Partie in SAN-Notation. Ein Klick auf einen Zug springt zur Stellung nach diesem Halbzug. Die Statuszeile zeigt die Seite am Zug und den aktuellen Freestyle-Index. Bei Matt, Patt oder anderen Spielenden wird der Status entsprechend aktualisiert.

## 2.9 Engine-Analyse

Die Engine bewertet die aktuell angezeigte Stellung. Bei aktiver Auto-Analyse startet die Suche nach Stellungswechseln automatisch. Alternativ kann die Analyse manuell gestartet und gestoppt werden. Die Ausgabe umfasst Bewertung, WDL, Bestzug und Hauptvarianten.

Option	Funktion
Auto-Analyse	Aktualisiert die Engine-Auswertung nach Änderungen der Stellung automatisch.
Suchmodus Tiefe	Analysiert bis zur gewählten Suchtiefe.
Suchmodus Zeit	Analysiert für die vorgegebene Zeit in Millisekunden.
MultiPV 1-4	Zeigt eine bis vier Hauptvarianten.
Bewertung	Gibt die Einschätzung aus Sicht der Seite am Zug aus.
WDL	Zeigt Win-Draw-Loss-Werte, sofern die Engine sie liefert.
Bestzug	Zeigt die beste gefundene Fortsetzung in lesbarer Form.



### Analyse-Hinweis

Für den Trainingseffekt empfiehlt sich: Partie zuerst ohne Engine spielen oder nachspielen, danach kritische Stellungen mit MultiPV vergleichen.

## 2.10 Empfohlener Ablauf im Verein

1. Freestyle-Nummer per Zufall erzeugen und öffentlich notieren.
2. Startstellung auf allen realen Brettern aufbauen lassen.
3. Rochaderegeln vor Rundenbeginn kurz erklären oder anhand des Tools überprüfen.
4. Partie normal spielen und Ergebnis festhalten.
5. Partie anschließend im Freestyle-Analyzer nachspielen oder per PGN importieren.
6. PGN mit FEN speichern, damit die Partie später reproduzierbar bleibt.
7. Kritische Stellungen mit Engine, Bestzug und Hauptvarianten analysieren.

## 3. Technik

### 3.1 Architektur

Der Freestyle-Analyzer ist als HTML-, CSS- und JavaScript-Anwendung umgesetzt. Die HTML-Struktur bildet Brett, Steuerung, Dialoge, Status, Zugliste und Engine-Anzeige ab. CSS steuert Layout, Schachbrett, Markierungen und responsive Darstellung. JavaScript erzeugt die Startstellungen, verwaltet die Partie, prüft Züge, importiert und exportiert PGN und kommuniziert mit Stockfish.

Technische Bausteine	
HTML	Definiert Oberfläche, Dialoge, Eingabefelder und Anzeigeelemente.
CSS	Setzt Times New Roman, Logo-/Brettlayout, Buttons, Modale und Markierungen.
JavaScript	Erzeugt Startstellungen, verwaltet Legalität, Historie, PGN und Engine-Kommunikation.
Stockfish	Läuft als Web Worker und analysiert aktuelle FEN-Stellungen.
FEN/PGN	Sichert die Freestyle-Startstellung und die gespielte Zugfolge.

### 3.2 Startstellungs-Generator

Die Funktion zur Startstellungserzeugung verwendet einen Index von 0 bis 959. Daraus wird eine Grundreihe gebildet, in der die Läufer auf verschiedenfarbigen Feldern stehen, die Dame und Springer verteilt werden und die verbleibenden drei Felder als Turm-König-Turm angeordnet sind. Aus dieser Startreihe baut das Tool anschließend das vollständige Brett für Weiß und Schwarz auf.

Element	Technische Rolle
generateStartRankFromIndex	Wandelt die Freestyle-Nummer in eine Startreihe aus acht Figuren um.
buildBoardFromStartRank	Erzeugt aus der Startreihe ein vollständiges 8x8-Brett.
currentIndex960	Speichert die aktuell verwendete Freestyle-Nummer.
initialBoard	Hält die Startstellung, aus der neue Game-Objekte aufgebaut werden.
applyNewIndex	Setzt Index, Startreihe, Brett, Historie und Ansicht zurück.



### 3.3 Rochade-Logik in Freestyle-Stellungen

Da König und Türme in Freestyle-Partien nicht zwingend auf den klassischen Ausgangsfeldern stehen, leitet die Anwendung die Rochadeinformationen aus dem Startbrett ab. Für jede Farbe werden König und beide Türme erkannt. Die Legalitätsprüfung berücksichtigt anschließend freie Wege, angegriffene Felder und die Zielanordnung der Rochade.

Technische Behandlung der Freestyle-Rochade	
<code>computeCastlingInfo</code>	Ermittelt König sowie linken und rechten Turm aus der Startstellung.
<code>canCastle</code>	Prüft, ob eine kurze oder lange Rochade in der aktuellen Stellung legal ist.
Zielstellung	Bei der kurzen Rochade landen König und Turm auf g- und f-Linie; bei der langen auf c- und d-Linie.
Castling Rights	Rochaderechte werden beim Ziehen von König oder Turm angepasst.

### 3.4 Partiehistorie und Notation

Die Anwendung führt die Partie in einer `masterHistory`-Liste aus Halbzügen. Für jede angezeigte Stellung wird aus Startstellung und Historie ein Game-Objekt rekonstruiert. Die Zugliste wird als SAN-Notation gerendert; für die Engine wird die aktuelle Stellung als FEN erzeugt.

Technik	Nutzen
<code>masterHistory</code>	Speichert alle gespielten Halbzüge.
<code>viewIndex</code>	Legt fest, bis zu welchem Halbzug die Stellung angezeigt wird.
<code>buildGameFromHistory</code>	Rekonstruiert die Stellung für Navigation, Anzeige und Analyse.
<code>moveToSAN</code>	Wandelt interne Züge in lesbare SAN-Notation um.
<code>gameToFEN</code>	Erstellt die Stellungsbeschreibung für Engine und PGN.

### 3.5 PGN-Import und Export

Beim Import werden PGN-Kopfzeilen, Ergebnis und SAN-Züge gelesen. Enthält die Partie eine FEN-Startstellung, wird diese als Anfangsposition gesetzt. Beim Export werden die unterstützten Kopfdaten ausgegeben und um Variant "Chess960", SetUp "1" sowie die FEN der Anfangsstellung ergänzt.

- `parsePGN` liest Header und Zugtext aus dem eingefügten PGN.
- `parseFEN` interpretiert Brett, aktive Farbe, Rochaderechte, en-passant-Feld und Zugzähler.
- `sanToMove` sucht den passenden legalen Zug zur SAN-Notation.
- `buildPGNForCurrentGame` schreibt Header, Chess960-Tags, FEN und Zugtext.
- `exportCurrentPGN` erzeugt eine herunterladbare Datei mit Startstellungsbezug.

### 3.6 Stockfish-Anbindung

Die Engine wird als Web Worker gestartet. Vor dem Start prüft die Anwendung die Erreichbarkeit der Engine-Dateien. Die Kommunikation erfolgt über UCI-Kommandos. Je nach Einstellung startet die Suche mit Suchtiefe oder Zeitlimit; MultiPV bestimmt die Anzahl der Varianten.

Engine-Aspekt	Umsetzung
Worker	Engine läuft getrennt von der Benutzeroberfläche.
UCI	Handshake und Kommandos steuern Bereitschaft, Optionen und Suche.
UCI_ShowWDL	Aktiviert WDL-Ausgaben, wenn verfügbar.
MultiPV	Wird auf Werte zwischen 1 und 4 begrenzt.



Engine-Aspekt	Umsetzung
go depth / go movetime	Startet Analyse nach Tiefe oder Zeit.

### 3.7 Bereitstellung und Fehlerdiagnose

- Die HTML-Datei sollte zusammen mit den Engine-Dateien bereitgestellt werden, sofern die Engine lokal eingebunden ist.
- Bei lokalen Dateiöffnungen können Browser-Sicherheitsregeln Worker oder WASM-Dateien blockieren; ein kleiner Webserver ist zuverlässiger.
- Wenn eine PGN nicht importiert wird, zuerst FEN, SetUp, Variant und die SAN-Zugfolge prüfen.
- Wenn ein Zug nicht angenommen wird, ist meist die falsche Seite am Zug, die Startstellung abweichend oder der Zug illegal.
- Wenn die Engine nicht startet, Browser-Konsole, Dateipfade und WASM-Unterstützung prüfen.

Problem	Mögliche Ursache	Maßnahme
Ungültige Startnummer	Eingabe liegt nicht zwischen 0 und 959.	Zahl korrigieren und erneut Anzeigen drücken.
PGN beginnt falsch	FEN fehlt oder passt nicht zur Zugfolge.	PGN mit Variant, SetUp und FEN exportieren oder korrigieren.
Rochade unklar	Startstellung weicht vom gespeicherten Index oder FEN ab.	Ausgangsstellung im Analyzer prüfen und auf dem Brett vergleichen.
Engine ohne Ausgabe	Worker oder WASM konnte nicht geladen werden.	Pfad, Serverbereitstellung und Browser-Konsole prüfen.
Keine Züge in der Liste	Neue Startstellung wurde gesetzt oder PGN enthält keinen lesbaren Zugtext.	Zugfolge eingeben oder PGN-Import prüfen.

## 4. Resümee

Der Hammerschach-Freestyle-Analyzer ist Generator, Partiebrett, PGN-Werkzeug und Engine-Analyse in einem. Er macht Freestyle- beziehungsweise Chess960-Partien praktisch nutzbar, weil Startstellung, Rochade, Zugliste, PGN-FEN und Analyse in einer Oberfläche zusammengeführt werden.

Der größte Nutzen liegt im Übergang vom Bildschirm zum echten Brett und zurück: Eine Zufallsstellung wird erzeugt, im Verein oder Turnier aufgebaut, als Partie gespielt und anschließend im Analyzer nachgespielt und ausgewertet. Dadurch eignet sich das Tool besonders für kreative Trainingsformen und abwechslungsreiche Turnierformate.

Im Sammelwerk der Hammerschach-Tools ergänzt der Freestyle-Analyzer die bisherigen Handbücher sinnvoll: Analyzer für klassische Partien, Trainer für Aufgaben, BoardLab für freie Brettarbeit, Open-Theme für Eröffnungsstellungen und Freestyle-Analyzer für zufällige Chess960-Startstellungen mit anschließender Partieanalyse.

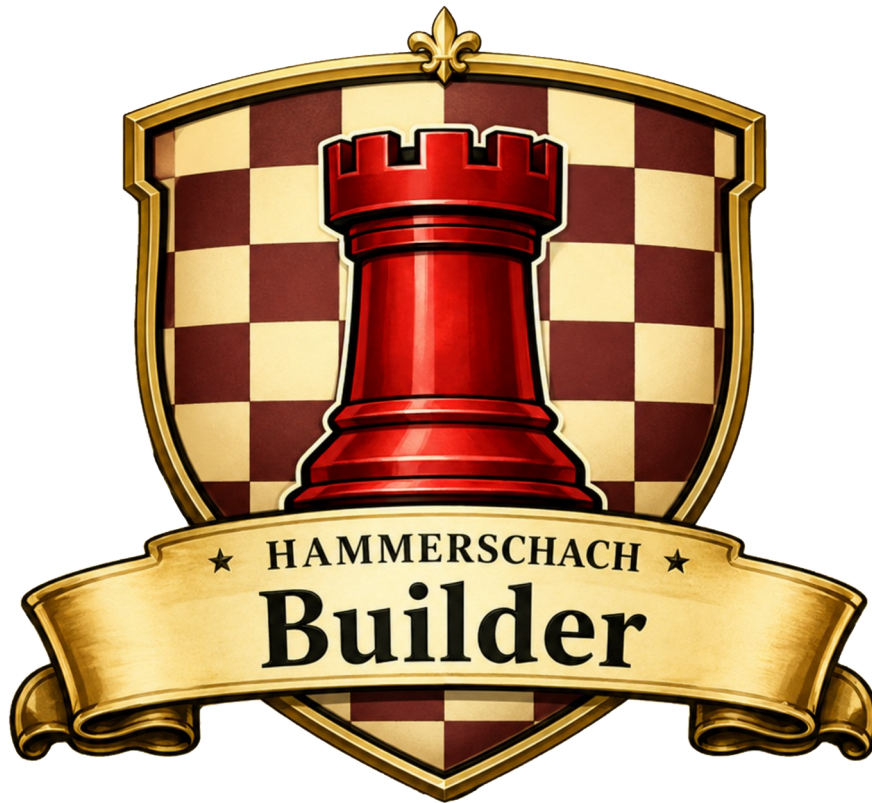
### Kurzfasit

Der Freestyle-Analyzer verbindet Zufallsstart, reales Brett, Turnierpraxis, PGN-Sicherung und Engine-Analyse zu einem kompakten Werkzeug für kreative Schachpartien.

# Handbuch

## Hammerschach-Builder

*Schachstellungen frei aufbauen, gestalten und als PNG-Bild exportieren*





# Inhaltsverzeichnis

1. Einleitung
2. Bedienung
3. Technik
4. Resümee

## Dokumentaufbau

Dieses Handbuch beschreibt den Hammerschach-Builder aus Anwendersicht und ergänzt die Bedienung um technische Hinweise zu Brettzustand, Figuren-Palette, Farbprofilen und PNG-Export. Das Layout ist auf DIN A4 Hochformat optimiert.

## 1. Einleitung

Der Hammerschach-Builder ist ein browserbasiertes Werkzeug zum freien Aufbau von Schachstellungen. Anders als ein Partie-Analyzer prüft er nicht primär Zugfolgen oder Regeln, sondern stellt ein flexibles Arbeitsbrett bereit: Figuren können gesetzt, verschoben, entfernt, farblich präsentiert und als Bild gespeichert werden.

Damit ist der Builder ein Werkzeug für Schachdiagramme, Trainingsmaterial, Vereinsunterlagen, Turnierausschreibungen, Social-Media-Beiträge und jede Situation, in der eine bestimmte Stellung schnell sichtbar gemacht werden soll. Die fertige Stellung kann als PNG-Bild exportiert und anschließend in Dokumente, Webseiten, Präsentationen oder Nachrichten eingefügt werden.

### 1.1 Zielgruppe

- Trainer, die Stellungsbilder für Aufgaben, Lösungsblätter oder Präsentationen benötigen.
- Vereine, die Diagramme für Themenabende, Turniere, Aushänge oder Webseiten erstellen möchten.
- Autoren, die Schachstellungen in Handbüchern, Artikeln oder digitalen Unterlagen abbilden wollen.
- Spieler, die eine Stellung aus einer Partie, Analyse oder Trainingsidee schnell nachbauen möchten.
- Organisatoren, die Startstellungen oder Demonstrationsstellungen visuell einheitlich ausgeben möchten.

### 1.2 Rolle im Hammerschach-Sammelwerk

Im Zusammenspiel der Hammerschach-Tools übernimmt der Builder die Aufgabe des visuellen Stellungsaufbaus. Während Analyzer und Trainer mit Partien, Aufgaben und Auswertungen arbeiten, konzentriert sich der Builder auf das sichtbare Brettbild. Er ist damit besonders gut geeignet, um Stellungen aus anderen Tools für Dokumentation, Unterricht oder Veröffentlichung aufzubereiten.

Abgrenzung zu den anderen Hammerschach-Tools	
<b>Builder</b>	Freies Setzen und Gestalten von Figuren; Ausgabe als PNG-Bild.
<b>Analyzer</b>	Nachspielen und Analysieren von Partien.
<b>Trainer</b>	Aufgaben, Lernphasen und Trainingsabläufe.
<b>BoardLab</b>	Experimentelles Brett für Varianten, Ideen und Stellungsarbeit.
<b>Open-Theme / Freestyle</b>	Themenstellungen, Zufallsstellungen und praktische Spiel- oder Turnierformate.

## Merksatz

Der Builder erzeugt kein Partieprotokoll, sondern ein sauberes Stellungsbild: Stellung aufbauen, Darstellung wählen, PNG speichern.



## 2. Bedienung

### 2.1 Aufbau der Oberfläche

Die Oberfläche besteht aus einem zentralen Schachbrett und einer rechten Seitenleiste. Die Seitenleiste enthält Statusanzeige, Grundfunktionen, Brettfarben, Figuren-Palette und PNG-Export. Dadurch bleibt der Arbeitsablauf kurz: Figur wählen, Feld anklicken, Stellung kontrollieren und Bild speichern.

Hauptbereiche der Oberfläche	
Statuszeile	Zeigt den aktuellen Arbeitsmodus oder eine Aktion wie Auswahl, Farbwechsel oder PNG-Speicherung.
Alles löschen	Leert das gesamte Brett.
Startstellung	Setzt die klassische Schachgrundstellung.
Flip	Dreht die Brettansicht, ohne die Stellung selbst zu verändern.
Brettfarben	Wechselt zwischen mehreren Farbprofilen.
Figuren-Palette	Stellt weiße und schwarze Figuren sowie ein Löschwerkzeug bereit.
PNG speichern	Exportiert das Brett als Bilddatei.

### 2.2 Figuren setzen

In der Figuren-Palette werden weiße und schwarze Figuren angeboten. Nach Auswahl einer Figur kann sie durch Klick auf ein Brettfeld gesetzt werden. Wird auf ein bereits belegtes Feld geklickt, wird die Figur dort ersetzt. So lassen sich Stellungen schnell aufbauen, korrigieren oder für Diagrammzwecke bewusst frei gestalten.

1. Gewünschte Figur in der Figuren-Palette anklicken.
2. Auf das Zielfeld des Brettes klicken.
3. Bei Bedarf weitere Felder anklicken, um dieselbe Figur mehrfach zu setzen.
4. Mit der Lösch-/Neutral-Auswahl oder der Escape-Taste die aktuelle Figurenauswahl verlassen.
5. Stellung visuell kontrollieren und anschließend speichern oder weiterbearbeiten.

#### Hinweis

Der Builder erlaubt freies Gestalten. Er verhindert deshalb nicht automatisch illegale Stellungen, doppelte Könige, fehlende Könige oder ungewöhnliche Materialverteilungen. Genau das macht ihn flexibel für Diagramme und Demonstrationen.

### 2.3 Figuren verschieben und entfernen

Figuren auf dem Brett können direkt verschoben werden. Ohne aktive Palettenfigur wird eine vorhandene Figur durch Klick aufgenommen und durch Klick auf ein Zielquadrat abgelegt. Zusätzlich unterstützt die Oberfläche Drag-and-Drop: Figuren können mit der Maus vom Brett auf ein neues Feld gezogen werden.

- **Klickbewegung:** Figur anklicken, Zielfeld anklicken.
- **Drag-and-Drop:** Figur mit gedrückter Maustaste auf ein anderes Feld ziehen.
- **Rechtsklick:** Figur auf dem angeklickten Feld entfernen.
- **Alles löschen:** Das gesamte Brett leeren und neu beginnen.
- **Startstellung:** Klassische Grundstellung wiederherstellen.

### 2.4 Brett drehen

Die Schaltfläche Flip dreht die Darstellung zwischen weißer und schwarzer Perspektive. Die tatsächlichen Brettdateien bleiben dabei unverändert. Die Koordinaten werden passend zur Ausrichtung angezeigt, damit ein Diagramm sowohl aus Sicht von Weiß als auch aus Sicht von Schwarz vorbereitet werden kann.



Aktion	Ergebnis	Typischer Nutzen
Flip	Ansicht wird gedreht.	Diagramm aus Sicht des Spielers darstellen.
Startstellung	Brett wird auf die Standardanfangsstellung gesetzt.	Ausgangspunkt für bekannte Stellungen.
Alles löschen	Alle Felder werden geleert.	Neue freie Komposition beginnen.

## 2.5 Brettfarben wählen

Der Builder bietet mehrere Farbprofile für helle und dunkle Felder. Die Presets verändern nur die Darstellung, nicht den Brettinhalt. Dadurch kann dasselbe Diagramm optisch an Handbuch, Präsentation, Webseite oder Druckausgabe angepasst werden.

Farbprofil	Charakter	Geeigneter Einsatz
Basis	Gold-/Rotton im Hammerschach-Stil.	Standardlayout der Tool-Familie.
Klassisch Braun	Traditionelle Holz- und Turnieranmutung.	Artikel, Vereinsmaterial, klassische Diagramme.
Grau / Neutral	Zurückhaltende, kontrastarme Darstellung.	Sachliche Unterlagen oder dezente Abbildungen.
Grün / Turnier	Bekanntes Turnierbrett-Farbschema.	Praxisnahe Demonstrationen und Vereinsbetrieb.

## 2.6 PNG-Export

Über den Exportbereich kann das aktuelle Brett als PNG-Datei gespeichert werden. Vor dem Speichern wird ein Dateiname eingetragen. Der Builder bereinigt ungeeignete Zeichen, ergänzt die Dateieindung und erzeugt das Bild aus dem sichtbaren Brett.

1. Stellung vollständig aufbauen und gewünschte Brettfarben wählen.
2. Optional die Ansicht mit Flip in die passende Perspektive bringen.
3. Dateinamen im Eingabefeld eintragen.
4. Exportmodus wählen: **Ohne Rand** oder **Brett mit weißem Rand**.
5. Auf **PNG speichern** klicken und die erzeugte Datei sichern.

Exportoptionen	
<b>Ohne Rand</b>	Speichert das Brett direkt als Bildfläche. Geeignet für Webseiten oder Einbettung mit eigenem Layout.
<b>Brett mit weißem Rand</b>	Fügt um das Brett einen weißen Rahmen hinzu. Geeignet für Präsentationen, Ausdrücke und Dokumente.
<b>Dateiname</b>	Leerzeichen werden vereinheitlicht und problematische Zeichen werden ersetzt.
<b>Markierungen</b>	Auswahlmarkierungen werden vor dem Screenshot entfernt, damit das Bild sauber bleibt.

## 2.7 Empfohlene Arbeitsabläufe

Ziel	Ablauf
Trainingsdiagramm	Startstellung setzen oder Brett leeren, Stellung aufbauen, Farbprofil wählen, PNG exportieren.
Aufgabe aus Partie	Kritische Stellung im Analyzer bestimmen, im Builder nachbauen, als Lösungsdiagramm speichern.
Vereinsunterlage	Brettfarben im Hammerschach-Stil wählen, Diagramm exportieren, in Aushang oder PDF einfügen.
Präsentation	Brett mit weißem Rand exportieren, damit das Diagramm auf



Ziel	Ablauf
	Folien klar abgegrenzt ist.
Webgrafik	Ohne Rand exportieren und die Bildgröße anschließend im Weblayout festlegen.

**Praxis-Tipp**

Für eine Serie von Diagrammen empfiehlt es sich, immer dasselbe Farbprofil, dieselbe Brettausrichtung und eine einheitliche Dateibenennung zu verwenden. Dadurch wirken Trainingsblätter und Handbücher geschlossen.

### 3. Technik

#### 3.1 Architektur

Der Hammerschach-Builder ist als HTML-, CSS- und JavaScript-Anwendung aufgebaut. HTML definiert Brett, Seitenleiste, Buttons und Exportbereich. CSS steuert die visuelle Gestaltung, JavaScript verwaltet den Brettzustand, Rendervorgang, Eingaben, Farbprofile und den PNG-Export.

Technische Bausteine	
HTML	Strukturiert Brett, Palette, Status, Buttons, Farbprofile und Exportbereich.
CSS	Gestaltet Layout, Schachbrett, Koordinaten, Figuren, Buttons und responsive Darstellung.
JavaScript	Speichert Brettdateien, verarbeitet Klicks, Drag-and-Drop, Rechtsklicks, Flip und Screenshot.
html2canvas	Erzeugt aus dem sichtbaren Brett eine Canvas-Grafik für den PNG-Download.
Figuren-SVGs	Die Figuren werden über Bildquellen geladen und im Brett sowie in der Palette angezeigt.

#### 3.2 Brettmodell

Intern wird das Brett als 8x8-Matrix geführt. Jedes Feld enthält entweder einen Punkt für ein leeres Feld oder ein Figurenkürzel. Großbuchstaben stehen für weiße Figuren, Kleinbuchstaben für schwarze Figuren. Dieses einfache Datenmodell macht Setzen, Verschieben und Löschen sehr schnell.

Kürzel	Figur
K / k	Weißer beziehungsweise schwarzer König
Q / q	Dame
R / r	Turm
B / b	Läufer
N / n	Springer
P / p	Bauer
.	Leeres Feld

Die Funktion zur Anzeige baut das Brett bei jeder Änderung neu auf. Dabei werden Felder, Figurenbilder, Koordinaten und Ereignisse für Klick, Drag-and-Drop und Rechtsklick erzeugt. Die Ausrichtung wird über einen Orientierungswert gesteuert, der bei Flip umgeschaltet wird.



### 3.3 Interaktion

Interaktionslogik	
Palettenauswahl	selectedPalettePiece speichert die aktuell gewählte Figur.
Feldauswahl	selectedBoardSquare merkt sich eine aufgenommene Figur beim Klick-und-Ablegen.
Drag-and-Drop	Datenpakete unterscheiden zwischen Figur aus Palette und Figur vom Brett.
Rechtsklick	Setzt das Feld sofort auf leer.
Escape	Hebt die aktuelle Figurenauswahl auf.

Die Anwendung ist bewusst direkt gehalten: Es gibt keine mehrstufigen Dialoge zum Figurenaufbau. Jede Änderung wird sofort im Brettzustand gespeichert und neu gerendert. Dadurch eignet sich der Builder besonders für schnelles Skizzieren und Korrigieren von Diagrammen.

### 3.4 Farbprofile

Die Brettfarben werden über vordefinierte Presets verwaltet. Jedes Preset enthält einen Namen sowie eine helle und eine dunkle Feldfarbe. Beim Wechsel werden CSS-Variablen gesetzt, sodass das komplette Brett sofort in den neuen Farben erscheint, ohne den Brettzustand zu verändern.

Technik	Bedeutung
boardColorPresets	Liste der verfügbaren Farbschemata.
activeColorPreset	Speichert das aktuell aktive Schema.
CSS-Variablen	Übergeben helle und dunkle Feldfarbe an die Darstellung.
renderColorPresets	Erzeugt die Auswahlbuttons für die Farbprofile.

### 3.5 PNG-Erzeugung

Beim PNG-Export wird das Brett zuerst visuell bereinigt: Markierungen werden entfernt und eine aktive Figurenauswahl wird aufgehoben. Anschließend wird das Brett mit html2canvas in eine Canvas-Grafik gerendert. Je nach Einstellung wird diese Grafik direkt verwendet oder in eine größere weiße Fläche eingefasst. Danach wird ein Blob erzeugt und als PNG-Datei heruntergeladen.

1. Dateiname bereinigen und Endung .png vorbereiten.
2. Auswahlmarkierungen entfernen.
3. Brett mit html2canvas in hoher Auflösung rendern.
4. Optional weißen Rand in einem neuen Canvas ergänzen.
5. PNG-Blob erzeugen und über einen temporären Download-Link speichern.
6. Statuszeile mit dem gespeicherten Dateinamen aktualisieren.

#### Technischer Hinweis

Wenn html2canvas oder externe Figurenbilder im Browser nicht geladen werden, kann der PNG-Export scheitern. Für zuverlässigen Offline-Betrieb sollten diese Ressourcen lokal bereitgestellt oder direkt in die Anwendung eingebettet werden.

### 3.6 Bereitstellung und Fehlerdiagnose

Problem	Mögliche Ursache	Maßnahme
Figur erscheint nicht	Bildquelle konnte nicht geladen werden.	Internetverbindung, lokale Ressourcen oder Browser-Konsole prüfen.
PNG wird nicht erzeugt	html2canvas wurde nicht geladen oder der Browser blockiert Ressourcen.	Seite neu laden, Netzwerk prüfen oder Bibliothek lokal einbinden.



Problem	Mögliche Ursache	Maßnahme
Falsche Ansicht	Brett wurde gedreht.	Flip erneut drücken, um zur gewünschten Perspektive zurückzukehren.
Figur mehrfach gesetzt	Palettenfigur bleibt aktiv.	Escape drücken oder leere Auswahl wählen.
Diagramm enthält falsches Material	Builder prüft keine Schachlegalität.	Stellung manuell kontrollieren oder in einem Analyzer gegenprüfen.

- Für Druck- und PDF-Ausgaben empfiehlt sich der Export mit weißem Rand.
- Für transparente oder flexible Weblayouts ist der Export ohne Rand meist besser geeignet.
- Bei Serien von Diagrammen sollte ein Namensschema verwendet werden, zum Beispiel thema\_01.png, thema\_02.png und thema\_03.png.
- Nach dem Export kann die PNG-Datei in Word, Pages, PDF, Präsentationen oder Webseiten weiterverwendet werden.

## 4. Resümee

Der Hammerschach-Builder ergänzt die Tool-Sammlung als kompaktes Werkzeug für die visuelle Stellungsarbeit. Er ist kein Ersatz für eine Engine-Analyse und kein vollständiger Parteeditor, sondern ein bewusst freies Brett zum schnellen Aufbau und Export von Schachdiagrammen.

Sein praktischer Wert liegt in der Geschwindigkeit: Figuren setzen, Farben wählen, Darstellung drehen, PNG speichern. Damit können Trainingsblätter, Handbuchseiten, Vereinsgrafiken und Präsentationen einheitlich und ohne zusätzliche Grafiksoftware vorbereitet werden.

Im Sammelwerk schließt der Builder die Lücke zwischen Analyse und Veröffentlichung. Was in den anderen Hammerschach-Tools gespielt, trainiert oder untersucht wird, kann mit dem Builder als klares Stellungsbild ausgegeben und weiterverwendet werden.

### Kurzfasit

Der Builder ist das Diagramm-Werkzeug der Hammerschach-Familie: freie Stellung, saubere Darstellung, schneller PNG-Export.

# Handbuch

## Hammerschach-Player

*Spiel gegen Engine, Human-Modus, Bedenkzeit und Partie-Nachspiel*



# Inhaltsverzeichnis

1. Einleitung
2. Bedienung
3. Technik
4. Resümee

## *Hinweis zum Sammelwerk*

Dieses Kapitel ergänzt die bestehende Handbuch-Kollektion um den Hammerschach-Player. Die Gestaltung folgt dem bisherigen A4-Hochformat mit Times-New-Roman-kompatibler Serifenschrift und eigener Titelseite.

## 1. Einleitung

Der Hammerschach-Player ist eine browserbasierte Spieloberfläche für Partien gegen eine integrierte Schach-Engine. Er richtet sich an Nutzer, die ohne zusätzliche Schachsoftware direkt im Browser spielen, verschiedene Spielstärken ausprobieren und die entstandene Partie anschließend über die Zugliste nachspielen möchten.

Im Mittelpunkt stehen vier Aufgaben:

- Partien spielen: Der Nutzer wählt seine Farbe, beginnt mit Weiß selbst oder lässt bei Schwarz die Engine den ersten Zug ausführen.
- Spielstärke dosieren: Mehrere DWZ-nahe Profile steuern Suchzeit, Skill-Level, Kandidatenwahl und Zufallselemente.
- Bedenkzeit trainieren: Vorgegebene Zeitmodi mit Inkrement simulieren typische Schnell-, Rapid- und längere Trainingspartien.
- Partien nachvollziehen: Die gespielten Züge werden in einer kompakten Zugliste angezeigt und können schrittweise vor- und zurückgespielt werden.

### 1.1 Zielgruppe

- Einsteiger und Schulschachspieler, die gegen eine bewusst reduzierte Engine spielen möchten.
- Vereinsspieler, die kurze Trainingspartien mit realistischen Zeitvorgaben absolvieren möchten.
- Trainer, die unterschiedliche Gegnerprofile demonstrieren und Partien direkt nachbesprechen möchten.
- Hammerschach-Nutzer, die eine einfache Spielkomponente innerhalb der Tool-Familie verwenden möchten.

### 1.2 Grundidee des Workflows

Der typische Ablauf lautet: Spielerfarbe wählen, gewünschte Spielstärke oder Human-Modus festlegen, optional eine Bedenkzeit aktivieren, neue Partie starten, Züge am Brett ausführen und die Partie anschließend über die Zugliste nachspielen. Die Anwendung ist bewusst auf das Spielen und Nachvollziehen der aktuellen Partie ausgerichtet. Ein separater PGN-Export ist in der vorliegenden Player-Oberfläche nicht als eigener Button vorgesehen.

## *Merksatz*

Farbe wählen, Niveau einstellen, Partie spielen, Zugliste prüfen und die entscheidenden Momente direkt am Brett noch einmal nachvollziehen.

## 2. Bedienung

### 2.1 Aufbau der Oberfläche

Die Oberfläche besteht aus einer Kopfleiste mit Logo und Einstellungen, einem großen Schachbrett, einem Zeitbereich unter dem Brett sowie einer rechten Zuglisten-Spalte. Statusmeldungen unterhalb des Bretts zeigen an, welche Seite am Zug ist oder ob Schach, Matt, Patt oder Zeitverlust eingetreten ist.

Bereich	Funktion
Spiel-Einstellungen	Auswahl von Spielerfarbe, Engine-Level und Human-Modus. Zusätzlich wird der Engine-Status angezeigt.
Brettbereich	Grafisches 8x8-Brett mit Figuren, Koordinaten, legalen Zielfeldern, illegalem Zugfeedback und Engine-Zuganimation.
Bedenkzeit	Auswahl typischer Zeitmodi wie 3+2, 5+3, 10+5, 15+10 und 60+30.
Zugliste	SAN-ähnliche Notation der gespielten Züge mit Navigationsbuttons für Anfang, Rückwärts, Vorwärts und Ende.
Darstellung	Brettfarben, Sound an/aus und automatische Brettausrichtung abhängig von der gewählten Spielerfarbe.

## 2.2 Spielerfarbe wählen

Über die Auswahl Spielerfarbe wird festgelegt, ob der Nutzer Weiß oder Schwarz spielt. Bei Weiß beginnt der Nutzer selbst. Bei Schwarz wird das Brett gedreht und die Engine startet nach dem Laden automatisch mit dem ersten weißen Zug.

Auswahl	Auswirkung
Weiß	Der Nutzer beginnt aus der normalen Anfangsstellung. Die Brettorientierung zeigt Weiß unten.
Schwarz	Die Engine beginnt die Partie. Das Brett wird aus schwarzer Sicht angezeigt.

## 2.3 Spielstärke einstellen

Der Player bietet mehrere Profile, die im Auswahlfeld Schwierigkeitsstufe sichtbar sind. Die DWZ-Bereiche sind als grobe Zielbereiche zu verstehen. Technisch werden Suchzeit, Skill-Level, Kandidatenvarianten und eine gewichtete Zugauswahl kombiniert, damit die Engine auf niedrigen Stufen nicht dauerhaft den objektiv stärksten Zug spielt.

Profil	Zielbereich	Charakter
Schulschach	DWZ 400-700	Sehr kurze Suchzeit, viele Kandidaten, deutliches Zufallselement.
Anfänger	DWZ 700-1000	Leichte Gegenwehr, aber weiterhin fehlertolerant.
Hobbyspieler	DWZ 1000-1300	Stabilere Züge, noch mit reduzierter Präzision.
Clubspieler	DWZ 1300-1600	Deutlich zielgerichteter, UCI-Elo-nahe Begrenzung im höheren Bereich.
Experte	DWZ 1700-2000	Stärkstes Profil der Oberfläche mit längerer Suchzeit.

## 2.4 Human-Modus

Der Human-Modus ist kein Zwei-Spieler-Modus. Er ersetzt die normale Engine-Level-Auswahl durch Profile, die menschlicher wirken sollen. Sobald ein Human-Profil aktiv ist, wird die normale Schwierigkeitsstufe gesperrt. Zur Verfügung stehen Human-Anfänger, Human-Hobbyspieler und Human-Clubspieler.

### *Praxis-Tipp*

Der Human-Modus eignet sich besonders für Trainingspartien, in denen der Gegner nicht wie eine perfekte Analysemaschine wirken soll. Die Partie bleibt aber weiterhin eine Partie gegen die Engine.

## 2.5 Neue Partie starten

Mit Neue Partie wird der aktuelle Spielverlauf zurückgesetzt. Farbe, Human-Modus, Spielstärke, Brettfarbe und aktive Bedenkzeit bleiben als gewählte Einstellungen erhalten. Wenn der Nutzer Schwarz gewählt hat, startet die Engine anschließend automatisch.

Empfohlene Reihenfolge:

- Spielerfarbe festlegen.
- Schwierigkeitsstufe oder Human-Modus auswählen.
- Optional Brettfarbe und Ton einstellen.

- Optional eine Bedenkzeit aktivieren.
- Neue Partie drücken und die Partie am Brett spielen.

## 2.6 Züge am Brett ausführen

Die Eingabe erfolgt per Klick: zuerst die eigene Figur, anschließend das Zielfeld. Nach der Auswahl markiert der Player mögliche legale Zielfelder. Ein Zug wird nur ausgeführt, wenn er nach den implementierten Schachregeln legal ist und die richtige Seite am Zug ist.

Aktion	Ergebnis
Eigene Figur anklicken	Die Figur wird ausgewählt; legale Zielfelder werden hervorgehoben.
Zielfeld anklicken	Der Zug wird ausgeführt, in die Zugliste eingetragen und anschließend ist die Gegenseite am Zug.
Fremde Figur anklicken	Der Zug wird nicht vorbereitet, wenn diese Farbe nicht am Zug ist.
Illegaler Zug	Die Eingabe wird verworfen und das Brett zeigt ein kurzes Fehlersignal.
Bauernumwandlung	Bei Erreichen der letzten Reihe erscheint eine Auswahl für Dame, Turm, Läufer oder Springer.

## 2.7 Sonderregeln und Partieende

Der Player berücksichtigt die üblichen Sonderregeln. Rochaderechte werden nach Königszug, Turmzug oder geschlagenem Turm aktualisiert. En-passant-Züge werden über das En-passant-Zielfeld erkannt. Bei Matt oder Patt endet die Partie regeltechnisch; die Statuszeile gibt den Abschluss aus.

- Rochade: nur möglich, wenn die Felder frei, die Rechte vorhanden und die durchquerten Felder nicht angegriffen sind.
- En passant: nur unmittelbar nach einem passenden gegnerischen Doppelzug.
- Umwandlung: Auswahl zwischen Dame, Turm, Läufer und Springer.
- Matt/Patt: Die Stellung wird anhand der legalen Züge und der Schachsituation erkannt.

## 2.8 Engine-Züge

Sobald die Engine am Zug ist, wird aus der aktuellen Stellung eine FEN erzeugt und an Stockfish übergeben. Der ausgewählte Kandidatenzug wird animiert. Das Zielfeld des Engine-Zugs wird anschließend kurz hervorgehoben, damit der Nutzer den letzten Zug sofort erkennt. Während die Engine rechnet, sollte die Stellung nicht über die Navigation verändert werden.

## 2.9 Bedenkzeit verwenden

Unter dem Brett stehen feste Zeitmodi bereit. Die Anzeige zeigt die eigene verbleibende Bedenkzeit. Nach einem eigenen Zug wird das eingestellte Inkrement addiert. Wenn dieselbe Zeitoption erneut gewählt wird, kann die Zeitsteuerung deaktiviert werden.

Button	Bedeutung	Typischer Einsatz
3 + 2	3 Minuten Grundzeit, 2 Sekunden Inkrement	Blitznahe Trainingspartie.
5 + 3	5 Minuten Grundzeit, 3 Sekunden Inkrement	Kurze Trainingspartie.
10 + 5	10 Minuten Grundzeit, 5 Sekunden Inkrement	Schnellschachtraining.
15 + 10	15 Minuten Grundzeit, 10 Sekunden Inkrement	Längere Analysepartie.
60 + 30	60 Minuten Grundzeit, 30 Sekunden Inkrement	Langpartie-Simulation.

### *Wichtig*

Die Bedenkzeitanzeige ist auf den Nutzer ausgerichtet. Sie dient im Player vor allem dazu, die eigene Zeiteinteilung zu trainieren.

## 2.10 Zugliste und Nachspielen

Die Zugliste dokumentiert die gespielten Züge in kompakter Notationsform. Über die Navigationsbuttons kann die Partie bis zur Anfangsstellung zurückgestellt oder wieder bis zur aktuellen Endstellung vorgespielt werden.

Taste	Funktion
«	Zur Anfangsstellung der aktuellen Partie springen.
<	Einen Halbzug zurückgehen.
>	Einen Halbzug vorgehen.
»	Zur letzten Stellung springen.

Während des Nachspiels zeigt der Zähler an, welcher Halbzug gerade betrachtet wird. Wird wieder bis zum Ende navigiert, kann die Partie normal fortgesetzt werden, sofern sie nicht bereits beendet ist.

## 2.11 Brettfarbe und Ton

Über Brettfarbe können verschiedene Farbschemata gewählt werden: Basis, Klassisch Braun, Grau/Neutral und Grün/Turnier. Die Auswahl wird im Browser gespeichert. Der Ton-Schalter aktiviert oder deaktiviert Zuggeräusche; auch diese Einstellung wird lokal gespeichert.

Einstellung	Wirkung
Brettfarbe	Ändert helle und dunkle Felder, ohne die Stellung zu verändern.
Ton	Schaltet Zug-, Schlag-, Schach- und Engine-Geräusche ein oder aus.
Browser-Speicherung	Brettfarbe und Ton werden über localStorage für spätere Besuche gemerkt.

## 2.12 Trainingsszenarien

Der Player kann auf verschiedene Weise eingesetzt werden. Besonders nützlich ist die Kombination aus reduzierter Spielstärke, Bedenkzeit und anschließender Nachbesprechung über die Zugliste.

Ziel	Empfohlene Einstellung	Ablauf
Einsteigertraining	Schulschach oder Anfänger, ohne Zeitdruck	Grundlegende Motive erkennen und nach Fehlern direkt zurücknavigieren.
Taktische Praxis	Hobbyspieler oder Clubspieler, 10+5	Partie unter moderatem Zeitdruck spielen und kritische Momente nachspielen.
Schwarztraining	Spielerfarbe Schwarz	Engine beginnt; Nutzer übt Verteidigung und Reaktion aus schwarzer Sicht.
Menschlicher Gegner	Human-Modus	Engine spielt weniger maschinenhaft und eignet sich für realistischere Trainingspartien.
Langpartie-Simulation	15+10 oder 60+30	Eigene Zeiteinteilung und ruhiges Rechnen trainieren.

## 2.13 Grenzen der Bedienung

Der Player ist keine vollständige Schachdatenbank. Er konzentriert sich auf das Spielen einer aktuellen Partie, nicht auf langfristige Partieverwaltung. In der Oberfläche sind keine separaten Funktionen für PGN-Import, PGN-Export oder Partielistenverwaltung sichtbar. Wer eine Partie dauerhaft archivieren möchte, sollte die Züge extern notieren oder sie in einem anderen Hammerschach-Werkzeug nachbauen und dort weiterverarbeiten.

### Abgrenzung

Player = spielen und nachspielen. Analyzer = importieren, verwalten und tiefer analysieren. Builder = Diagramme frei aufbauen und exportieren.

## 3. Technik

### 3.1 Architektur

Der Hammerschach-Player ist als HTML-, CSS- und JavaScript-Anwendung aufgebaut. Die Oberfläche wird im Browser gerendert. Die Schachlogik liegt im JavaScript-Code, während die Engine über einen Web Worker gestartet wird.

Baustein	Aufgabe
HTML	Definiert Kopfbereich, Einstellungen, Brett, Zeitbox, Status und Zugliste.
CSS	Gestaltet Layout, Brettfarben, Figurenanzeige, Animationen, Popups und responsives Verhalten.
JavaScript	Verwaltet Brettzustand, Zuglegalität, Historie, Navigation, Engine-Anfragen und Zeitmodus.
Stockfish.js	Wird als Web Worker geladen und berechnet Engine-Züge über das UCI-Protokoll.
localStorage	Speichert Brettfarbe und Soundeinstellung lokal im Browser.

## 3.2 Brettmodell und Zuglogik

Intern wird das Brett als 8x8-Matrix geführt. Ein Punkt steht für ein leeres Feld. Großbuchstaben stehen für weiße Figuren, Kleinbuchstaben für schwarze Figuren. Die Methode zur legalen Zugerzeugung prüft alle Pseudozüge und verwirft anschließend Züge, die den eigenen König im Schach lassen würden.

Kürzel	Bedeutung
K/k	Weißer beziehungsweise schwarzer König.
Q/q	Dame.
R/r	Turm.
B/b	Läufer.
N/n	Springer.
P/p	Bauer.
.	Leeres Feld.

Die Historie der Partie wird als Folge von Halbzügen gespeichert. Für die Anzeige einer früheren Stellung wird die Partie aus der Anfangsstellung bis zum gewünschten Halbzug rekonstruiert. Dadurch kann die Zugliste vor- und zurückblättern, ohne die eigentliche Historie zu verändern.

## 3.3 FEN, UCI und Engine-Steuerung

Vor jedem Enginezug wird die aktuelle Stellung in FEN umgewandelt. Danach sendet der Player Befehle wie `position fen` und `go movetime` an Stockfish. Je nach Profil werden `Skill Level`, `MultiPV` und - falls vorhanden - eine `UCI-Elo`-Begrenzung gesetzt. Die Engine liefert Kandidatenzüge und Bewertungsinformationen zurück.

Technik	Rolle im Player
<code>gameToFEN</code>	Erzeugt Brett, Zugrecht, Rochaderechte, En-passant-Feld, Halbzug- und Vollzugzähler.
<code>MultiPV</code>	Ermöglicht mehrere Kandidatenzüge, aus denen leichtere Profile gewichtet auswählen.
<code>Skill Level</code>	Reduziert die Engine-Stärke innerhalb der Stockfish-Möglichkeiten.
<code>UCI_Elo</code>	Wird bei höheren Profilen verwendet, sofern die Engine diese Option unterstützt.
<code>go movetime</code>	Begrenzt die Rechenzeit pro Enginezug entsprechend dem gewählten Profil.

## 3.4 Bewertungsanzeige

Während Stockfish rechnet, kann die Oberfläche Bewertungsinformationen anzeigen. Centipawn-Werte werden auf Weiß-Sicht umgerechnet. Mattangaben werden als `Mattangriff` oder `Mattdrohung` angezeigt. Diese Bewertung dient nur der Information und stoppt die Partie nicht automatisch.

## 3.5 Zeitmodus

Der Zeitmodus verwaltet Grundzeit, Inkrement, verbleibende Nutzerzeit und den laufenden Countdown. Nach einem eigenen Zug wird das Inkrement addiert. Bei Ablauf wird die Partie im Player als Zeitverlust markiert und der Status entsprechend aktualisiert.

Wichtige Variablen und Konzepte sind `baseTimeMs`, `incrementMs`, `playerTimeMs`, `clockInterval`, `firstMoveDone` und `timeLost`. Der Player trennt damit die eigentliche Schachlogik von der Bedienlogik der Uhr.

## 3.6 Animationen, Markierungen und Sound

Die Oberfläche arbeitet mit visuellen Rückmeldungen. Ausgewählte Figuren, legale Zielfelder und Engine-Zielfelder erhalten eigene CSS-Klassen. Engine-Züge werden über ein temporäres Geisterbild animiert. Für Zugeräusche werden mehrere MP3-Dateien eingebunden, zum Beispiel pickup, drop, capture, check und engine.

Rückmeldung	Bedeutung
Auswahlmarkierung	Zeigt die aktuell gewählte Figur.
Legale Zielfelder	Hilft beim Erkennen gültiger Züge.
Illegal-Flash	Kurze visuelle Warnung bei ungültiger Eingabe.
Engine-Animation	Zeigt den gegnerischen Zug sichtbar auf dem Brett.
Sound-Schalter	Erlaubt lautloses Training.

### 3.7 Bereitstellung und Fehlerdiagnose

Problem	Mögliche Ursache	Maßnahme
Engine startet nicht	stockfish.js fehlt oder darf nicht als Worker geladen werden.	Datei im gleichen Veröffentlichungsumfeld bereitstellen und Browser-Konsole prüfen.
Figuren erscheinen nicht	Externe SVG-Quellen sind nicht erreichbar.	Figuren lokal einbinden oder Netzwerkzugriff prüfen.
Kein Ton	Sound ist deaktiviert oder MP3-Dateien fehlen.	Ton-Schalter prüfen und sounds-Ordner bereitstellen.
Brettfarbe bleibt nicht erhalten	localStorage ist blockiert.	Browser-Einstellungen oder privaten Modus prüfen.
Engine wirkt zu stark oder zu schwach	Profil passt nicht zum Trainingsziel.	Anderes Engine-Level oder Human-Modus wählen.

## 4. Resümee

Der Hammerschach-Player ergänzt die Tool-Sammlung als direktes Spielwerkzeug. Während Analyzer und andere Werkzeuge stärker auf Import, Auswertung oder Stellungsaufbau ausgerichtet sind, bietet der Player eine klare Oberfläche für die praktische Partie gegen eine Engine.

Sein Nutzen liegt in der schnellen Vorbereitung: Farbe wählen, Spielstärke festlegen, gegebenenfalls Bedenkzeit aktivieren und losspielen. Die Zugliste macht die Partie anschließend nachvollziehbar, sodass Fehler, Wendepunkte und verpasste Chancen unmittelbar besprochen werden können.

#### *Kurzfasit*

Der Player ist das Spielbrett der Hammerschach-Familie: unkompliziert starten, gegen abgestufte Engine-Profile spielen und die Partie direkt im Anschluss nachverfolgen.