





## Programm starten (Konsole)

Als Benutzer:in möchte ich das Programm ohne GUI in der Konsole starten.

---

Ein Start des Programms mit dem Argument `--no-gui` führt zum Start des konsolenbasierten Programms und einem Menü zur Spielmodus-Wahl zwischen einem Netzwerk-Spiel und einem lokalen Spiel wird angezeigt.

Vorbedingung: keine

Nachbedingung: Das Programm ist gestartet und das Auswahlmenü wird dargestellt.

Eingangsbedingung: Der/Die Benutzer:in startet das Programm mit dem Argument `--no-gui`.

## Programm starten (GUI)

Als Benutzer:in möchte ich das Programm in der GUI starten.

---

Ein Aufruf des Programms in der Konsole ohne Argument führt zum Start des GUI-basierten Programms und das Hauptmenü zur Wahl zwischen **New Game/Neues Spiel**, **Network Game/Netzwerk-Spiel**, **Load Game/Lade Spiel**, **Language/Sprache** wird angezeigt.

Vorbedingung: keine

Nachbedingung: Das Programm ist gestartet und das Auswahlmenü wird in der GUI dargestellt.

Eingangsbedingung: Der/Die Benutzer:in startet das Programm ohne Argument.

## Netzwerkspiel wählen

Als Benutzer:in möchte ich ein Netzwerkspiel im selben Netzwerk spielen.

---

Das Argument `network` führt zur Eingabe der IP-Adresse.

Variation GUI: Der Klick auf den Button `Network Game/Netzwerk-Spiel` führt zu Öffnen eines Fensters, in dem die IP-Adresse eingegeben werden kann.

Vorbedingung: Der/Die Benutzer:in startet das Programm mit dem Argument `--no-gui`.

Variation GUI: Der/Die Benutzer:in startet das Programm ohne Argument.

Nachbedingung: Die Eingabe einer IP-Adresse ist möglich.

Eingangsbedingung: Der/Die Benutzer:in wählt ein Netzwerk-Spiel.

## Lokales Spiel wählen

Als Benutzer:in möchte ich ein lokales Spiel spielen.

---

Das Argument `local` führt zur Gegnermodus-Wahl.

Variation GUI: Der Klick auf den Button **New Game/Neues Spiel** führt zum Öffnen eines Fensters, in dem der Gegnermodus gewählt werden kann.

Vorbedingung: Der/Die Benutzer:in startet das Programm mit dem Argument `--no-gui`.

Variation GUI: Der/Die Benutzer:in startet das Programm ohne Argument.

Nachbedingung: Die Wahl eines Gegnermodus ist möglich.

Eingangsbedingung: Der/Die Benutzer:in wählt ein lokales Spiel.

Variation GUI: Der/Die Benutzer:in klickt auf den Button **New Game /Neues Spiel**.

## Lokales Spiel gegen Person starten

Als Benutzer:in möchte ich das lokale Spiel gegen eine andere Person starten.

---

Das Argument `person` führt zum Starten eines neuen Spiels gegen eine andere Person.

Variation GUI: Der Klick auf den Button `Person/Mensch` führt zu einem Spiel gegen eine andere Person.

Vorbedingung: Der/Die Benutzer:in hat ein lokales Spiel gewählt.

Variation GUI: Der Button `Neues Spiel/New Game` wurde angeklickt.

Nachbedingung: Ein neues Schachspiel gegen eine andere Person ist gestartet.

Eingangsbedingung: Der/Die Benutzer:in wählt eine andere Person als Gegner.

## Lokales Spiel gegen KI starten (Konsole)

Als Benutzer:in möchte ich das lokale Spiel gegen die KI starten.

---

Die Auswahl ai führt zum Starten eines neuen Spiels gegen die KI.

Vorbedingung: Als Gegner wurde die KI ausgewählt.

Nachbedingung: Ein neues Schachspiel gegen die KI ist gestartet. Der/die Benutzer:in hat die weißen Figuren, die KI die schwarzen Figuren.

Eingangsbedingung: Der/Die Benutzer:in wählt die KI als Gegner.



## Lokales Spiel gegen KI starten (GUI)

Als Benutzer:in möchte ich das lokale Spiel gegen die KI in der GUI starten.

---

Die Auswahl einer Farbe führt zu einem Spiel gegen die KI.

- Vorbedingung: Als Gegner wurde die KI ausgewählt.
- Nachbedingung: Ein neues Schachspiel gegen die KI ist gestartet. Der Klick auf den Button **White/Weiß** startet ein Spiel mit der Farbe Weiß für den/die Benutzer:in und der Farbe Schwarz für die KI, mit einem Klick auf den Button **Black/Schwarz** startet ein Spiel mit der Farbe Schwarz für den/die Benutzer:in und der Farbe Weiß für die KI.
- Eingangsbedingung: Der/Die Benutzer:in wählt die KI als Gegner.

## Zug durchführen (Konsole)

Als Benutzer:in möchte ich ein Zug durchführen.

---

Das Argument [letter][number]-[letter][number] mit letter = {a,b,c,d,e,f,g,h} und number = {1,2,3,4,5,6,7,8,} muss einen gültigen Zug für die zu bewegende Figur beschreiben. Eine ungültige Eingabe führt zu der Konsolenausgabe !Invalid move, ein ungültiger Zug führt zu der Konsolenausgabe !Move not allowed.

Vorbedingung:	Ein aktives Schachspiel.
Nachbedingung:	Der Zug ist ausgeführt oder es erscheint eine entsprechende Fehlermeldung in der Konsole.
Eingangsbedingung:	Der/Die Benutzer:in gibt in die Konsole ein Argument [letter][number]-[letter][number] ein.

## Zug durchführen (GUI)

Als Benutzer:in möchte ich ein Zug durchführen.

---

Ein Klick auf die Spielfigur und anschließend auf ein Feld führt zu einem Zug. Ein ungültiger Zug oder ein Klick außerhalb des Spielfeldes wird ignoriert.

- |                    |   |
|--------------------|---|
| Vorbedingung:      | Ein aktives Schachspiel.  |
| Nachbedingung:     | Der Zug ist ausgeführt oder es passiert nichts.   |
| Eingangsbedingung: | Der/Die Benutzer:in klickt mit der Maus auf die zu bewegende Figur und anschließend auf ein Schachfeld. |

## Bauernumwandlung (Konsole)

Als Benutzer:in möchte ich eine Bauernumwandlung durchführen.

---

Beim Zug des Bauern auf ein Feld am Ende des Schachbretts wird eine Umwandlung in gewählte Figur (Königin-Q, Turm-R, Läufer-B, Springer-N) vollzogen.

- |                    |   |
|--------------------|---|
| Vorbedingung:      | Ein aktives Schachspiel, Bauer kann auf ein Feld am Ende des Schachbretts ziehen und entsprechender Buchstabe ist an die Zug-Eingabe angehängt. |
| Nachbedingung:     | Der Bauer wurde in entsprechende Figur umgewandelt.   |
| Eingangsbedingung: | Der/Die Benutzer:in zieht mit einem Bauern auf ein Feld am Ende des Schachbretts und gibt dabei einen Buchstaben an.                            |

## Bauernumwandlung (GUI)

Als Benutzer:in möchte ich eine Bauernumwandlung durchführen.

---

Beim Zug des Bauern auf ein Feld am Ende des Schachbretts wird eine Umwandlung in gewählte Figur (Königin, Turm, Läufer, Springer) vollzogen.

- |                    |  |
|--------------------|--|
| Vorbedingung:      | Ein aktives Schachspiel und Bauer kann auf ein Feld am Ende des Schachbretts ziehen.   |
| Nachbedingung:     | Der Bauer wurde in entsprechende Figur umgewandelt.  |
| Eingangsbedingung: | Der/Die Benutzer:in zieht mit einem Bauern auf ein Feld am Ende des Schachbretts und klickt im Fenster auf den Button mit dem Namen der gewünschten Figur. |

## Rochade

Als Benutzer:in möchte ich eine Rochade mit meinem König und einem Turm durchführen.

---

Der König wird in zwei Felder in Richtung des entsprechenden Turms gesetzt.

**Vorbedingung:** Ein aktives Schachspiel, es ist der erste Zug des Königs und des entsprechenden Turms, es steht keine Figur zwischen König und Turm und König wird auf keinem Feld des Zugs angegriffen.

**Nachbedingung:** Der König steht zwei Felder in Richtung des entsprechenden Turms und der Turm ist über den König gesprungen und steht neben diesem.

**Eingangsbedingung:** Der/Die Benutzer:in gibt einen Zug ein, der den König um zwei Felder in Richtung des entsprechenden Turms bewegt.

Variation GUI: Der/Die Benutzer:in auf den König und anschließend auf ein Schachfeld zwei Felder vom König entfernt in Richtung des entsprechenden Turms.

## Geschlagene Figuren anzeigen

Als Benutzer:in möchte ich die schon geschlagenen Figuren angezeigt bekommen.

---

Das Argument `beaten` listet die geschlagenen Figuren als Konsolenausgabe auf.

Variation GUI: Die Figuren erscheinen am Rand des Spielfeldes, nachdem sie geschlagen wurden.

Vorbedingung: Ein aktives Schachspiel und es wurde schon eine Schachfigur geschlagen.

Nachbedingung: Die geschlagenen Figuren sind auf der Konsole ausgegeben.

Variation GUI: Die geschlagenen Figuren sind unter dem Spielfeld zu sehen.

Eingangsbedingung: Der/Die Benutzer:in gibt `beaten` in die Konsole ein.

Variation GUI: keine

## Neues Spiel starten

Als Benutzer:in möchte ich ein neues Spiel starten.

---

Das Argument newGame zeigt das Menü zur Wahl des Spielmodus (Netzwerk/Lokal).

Variation GUI: Ein Klick auf den Button **New Game** im **Chess/Schach**-Menü beendet das Spiel und zeigt das Hauptmenü zur Wahl zwischen **New Game/Neues Spiel**, **Network Game/Netzwerk-Spiel**, **Load Game/Lade Spiel**, **Language/Sprache** an.

**Vorbedingung:** Das Programm ist gestartet.

Variation GUI: Ein aktives Spiel.

**Nachbedingung:** Es erscheint das Auswahlmenü zur Wahl eines Spielmodus (Netzwerk/Lokal).

Variation GUI: Das aktive Schachspiel wurde beendet und das Hauptmenü zur Wahl zwischen **New Game/Neues Spiel**, **Network Game/Netzwerk-Spiel**, **Load Game/Lade Spiel**, **Language/Sprache** wird angezeigt.

**Eingangsbedingung:** Der/Die Benutzer:in gibt newGame in die Konsole ein.

Variation GUI: Der/Die Benutzer:in klickt auf den Button **New Game** im **Chess/Schach**-Menü.



## Spiel speichern

Als Benutzer:in möchte ich den Spielstand speichern.

---

Das Argument `save` speichert den Spielstand im Verzeichnis des Skripts.

Variation GUI: Ein Klick auf den Button **Save Game** im Menü **Chess/Schach** speichert den Spielstand im Verzeichnis des Skripts.

Vorbedingung: Ein aktives Schachspiel.

Nachbedingung: Das Spiel ist gespeichert und weiter spielbar.

Eingangsbedingung: Der/Die Benutzer:in gibt das Argument `save` in die Konsole ein.

Variation GUI: Der Benutzer klickt auf den Button **Save Game** im Menü **Chess/Schach**.

## Spiel laden

Als Benutzer:in möchte ich ein gespeichertes Spiel laden, um es weiterzuspielen.

---

Das Argument `load` gibt eine Liste aller Dateien, die im Verzeichnis liegen, auf der Konsole aus.

Variation GUI: Ein Klick auf den Button **Load Game** im Hauptmenü oder im Menü **Chess/Schach** zeigt ein Dialogfenster mit den gespeicherten Spielen an.

**Vorbedingung:** Das Programm ist gestartet und es wurde schon ein Spiel gespeichert.

**Nachbedingung:** Ein Auswahl der gespeicherten Spiele wird angezeigt.

**Eingangsbedingung:** Der/Die Benutzer:in gibt das Argument `load` in die Konsole ein.

Variation GUI: Der Benutzer klickt auf den Button **Load Game**.

## Spiel aufgeben

Als Benutzer:in möchte ich aufgeben.

---

Das Argument `giveUp` führt zur Aufgabe des Spiels.

Variation GUI: Ein Klick auf den Button **Give up/Aufgeben** im Menü **Chess/Schach** führt zur Aufgabe des Spiels.

Vorbedingung: Es läuft ein aktives Schachspiel.

Nachbedingung: Das Spiel wurde aufgegeben und es wird ein Menü angezeigt, ob ein neues Spiel gestartet werden soll.

Variation GUI: Das Spiel wurde aufgegeben und es wird das Hauptmenü angezeigt.

Eingangsbedingung: Der/Die Benutzer:in gibt das Argument `giveUp` in die Konsole ein.

Variation GUI: Der Benutzer klickt auf den Button **Give up/Aufgeben** im Menü **Chess/Schach**.

## Sprache zu Deutsch ändern

Als Benutzer:in möchte ich die Sprache zu Deutsch ändern.

---

Das Argument `german` ändert die Sprache zu Deutsch.

Variation GUI: Ein Klick auf den Button `German` im Menü `Language` ändert die Sprache zu Deutsch.

Vorbedingung: Das Programm ist gestartet oder ein aktives Spiel läuft und die aktuelle Sprache ist Englisch.

Nachbedingung: Die Sprache ist geändert.

Eingangsbedingung: Der/Die Benutzer:in gibt das Argument `german` in die Konsole ein.

Variation GUI: Der Benutzer klickt auf den Button `German` im `Chess/Schach`-Menü im Unter-Menü `Language`.

## Sprache zu Englisch ändern

Als Benutzer:in möchte ich die Sprache zu Englisch ändern.

---

Das Argument `english` ändert die Sprache zu Deutsch.

Variation GUI: Ein Klick auf den Button `Englisch` im Menü `Language` ändert die Sprache zu Englisch.

Vorbedingung: Das Programm ist gestartet oder ein aktives Spiel läuft und die aktuelle Sprache ist Deutsch.

Nachbedingung: Die Sprache ist geändert.

Eingangsbedingung: Der/Die Benutzer:in gibt das Argument `english` in die Konsole ein.

Variation GUI: Der Benutzer klickt auf den Button `Englisch` im `Chess/Schach`-Menü im Unter-Menü `Language`.

## Bedienungsanleitung öffnen

Als Benutzer:in möchte ich mir die Bedienungsanleitung anschauen.

---

Das Argument `help` öffnet die Bedienungsanleitung in einem PDF-Viewer des Computers.

Variation GUI: Ein Klick auf den **User Guide/Bedienungsanleitung**-Button im **Help/Hilfe**-Menü öffnet die Bedienungsanleitung in einem PDF-Viewer des Computers.

**Vorbedingung:** Es läuft ein aktives Spiel und es befindet sich ein PDF-Viewer auf dem Computer.

**Nachbedingung:** Die Bedienungsanleitung wurde im PDF-Viewer geöffnet.

**Eingangsbedingung:** Der/Die Benutzer:in gibt das Argument `help` in die Konsole ein.

Variation GUI: Der Benutzer klickt auf den **User Guide/Bedienungsanleitung**-Button im Menü **Help/Hilfe**.

## Zug-Historie anzeigen (GUI)

Als Benutzer:in möchte ich die Zug-Historie des aktuellen Spiels ansehen.

---

Der Klick auf den Button **Move History/Zug-Historie** im **Chess/Schach**-Menü zeigt ein Fenster mit der Zug-Historie des aktuellen Spiels an.

Vorbedingung: Es läuft ein aktives Spiel.

Nachbedingung: Ein Fenster mit der aktuellen Zug-Historie wird angezeigt.

Eingangsbedingung: Der/Die Benutzer:in klickt auf den Button **Move History/Zug-Historie** im **Chess/Schach**-Menü.

## Schachbrett-Rotation an-/ausschalten (GUI)

Als Benutzer:in möchte ich die Rotation des Schachbretts an- oder ausschalten.

---

Ein Klick auf den Button **Board Rotation/Drehung des Spielbretts** im **Settings/Einstellungen**-Menü schaltet die Rotation an oder aus.

**Vorbedingung:** Es läuft ein aktives Spiel gegen eine andere Person.

**Nachbedingung:** Die Rotation des Schachbretts ist an- oder ausgeschaltet.

**Eingangsbedingung:** Der/Die Benutzer:in klickt auf den Button **Board Rotation/Drehung des Spielbretts** im **Settings/Einstellungen**-Menü.



## Züge hervorheben an-/ausschalten (GUI)

Als Benutzer:in möchte ich die Option, dass mögliche Züge hervorgehoben werden an- oder ausschalten.

---

Der Klick auf den Button **Highlight Moves/Züge hervorheben** im **Settings/Einstellungen**-Menü schaltet die Option Züge hervorzuheben an oder aus.

Vorbedingung: Es läuft ein aktives Spiel.

Nachbedingung: Die möglichen Züge einer Spielfigur werden (nicht) angezeigt.

Eingangsbedingung: Der/Die Benutzer:in klickt auf den Button „**Highlight Moves/Züge hervorheben** im **Settings/Einstellungen**-Menü.

## Gewählte Figur ändern an-/ausschalten (GUI)

Als Benutzer:in möchte ich die Option, dass ich nach der Auswahl einer Figur eine andere Figur auswählen darf, an- oder ausschalten.

---

Der Klick auf den Button **Change Selected Piece/Gewählte Figur ändern** im **Settings/Einstellungen**-Menü schaltet die Option eine gewählte Figur zu ändern an oder aus.

- Vorbedingung:** Es läuft ein aktives Spiel und es ist momentan keine Figur für einen Zug ausgewählt.
- Nachbedingung:** Nach der Auswahl einer Figur darf (k)eine andere Figur ausgewählt werden.
- Eingangsbedingung:** Der/Die Benutzer:in klickt auf den Button **Change Selected Piece/Gewählte Figur ändern** im **Settings/Einstellungen**-Menü.

## Hinweis für Schach an-/ausschalten (GUI)

Als Benutzer:in möchte ich die Option, dass ein Hinweis angezeigt wird, wenn jemand im Schach steht, an- oder ausschalten.

---

Der Klick auf den Button **Hint: In Check/Hinweis: Schach** im **Settings/Einstellungen**-Menü schaltet die Option ob ein Hinweis bei Schach angezeigt wird an oder aus.

Vorbedingung: Es läuft ein aktives Spiel.

Nachbedingung: Es wird (k)ein Hinweis angezeigt, wenn jemand im Schach steht.

Eingangsbedingung: Der/Die Benutzer:in klickt auf den Button **Hint: In Check/Hinweis: Schach** im **Settings/Einstellungen**-Menü.

## Schachbrett-Design ändern (GUI)

Als Benutzer:in möchte ich das Design des Schachbretts ändern.

---

Ein Klick auf den Button **Classic/Klassik** bzw. **Black'n'White/Schwarz-Weiß** im **Themes/Stil**-Menü ändert das Design des Schachbretts.

Vorbedingung:	Es läuft ein aktives Spiel.
Nachbedingung:	Das Design des Schachbretts wurde geändert.
Eingangsbedingung:	Der/Die Benutzer:in klickt auf den Button <b>Classic/Klassik</b> bzw. <b>Black'n'White/Schwarz-Weiß</b> im <b>Themes/Stil</b> -Menü.

## Programm beenden

Als Benutzer:in möchte ich das Programm beenden.

---

Das Argument quit beendet das Programm.

Variation GUI: Ein Klick auf den **Exit/Beenden**-Button im **Chess/Schach**-Menü oder den **Close Window**-Button beendet das Programm.

Vorbedingung: Das Programm wurde gestartet.

Variation GUI: Es läuft ein aktives Spiel.

Nachbedingung: Das Programm ist beendet.

Eingangsbedingung: Der/Die Benutzer:in gibt das Argument quit in die Konsole ein.

Variation GUI: Der Benutzer klickt auf den **Exit**-Button im **Chess/Schach**-Menü.