StoryCards

Programm starten (Konsole)

Vorderseite:	Als Benutzer möchte ich das Programm ohne GUI in der Konsole starten.
Rückseite:	Ein Start des Programms mit dem Argument --no-gui führt zum Start des konsolenbasierten Programms und ein Menü zur Wahl zwischen „Play new game“ / „Load saved game“ wird angezeigt.
Vorbedingung:	keine
Nachbedingung:	Das Programm ist gestartet und das Auswahlmenü wird dargestellt.
Eingangsbedingung:	Der Benutzer startet das Programm mit dem Argument --no-gui in der Konsole.

Programm starten (GUI)

Vorderseite:	Als Benutzer möchte ich das Programm in der GUI starten.
Rückseite:	Ein Aufruf des Programms in der Konsole ohne Argument führt zum Start des GUI-basierten Programms und ein Menü zur Wahl zwischen „Play new game“ / „Load saved game“ wird angezeigt.
Vorbedingung:	keine
Nachbedingung:	Das Programm ist gestartet und das Auswahlmenü wird in der GUI dargestellt.
Eingangsbedingung:	Der Benutzer startet das Programm ohne Argument in der Konsole.

Neues Spiel starten

Vorderseite:	Als Benutzer möchte ich ein neues Spiel starten.
Rückseite:	Das Argument newgame führt zum Start eines neuen Schachspiels. Variation GUI: Das Anklicken des Buttons „Play new game“ führt zum Start eines neuen Schachspiels.
Vorbedingung:	Das Programm wurde gestartet.
Nachbedingung:	Ein neues Schachspiel ist initialisiert und das Menü zur Wahl des Gegnermodus „Human“ / „AI“ wird angezeigt.
Eingangsbedingung:	Der Benutzer startet nach Aufforderung durch Konsole ein neues Spiel mit dem Argument newgame in der Konsole. Variation GUI: Der Benutzer klickt auf den Button „Play new game“.

Gegnerwahl

Vorderseite:	Als Benutzer möchte ich einen Gegnermodus auswählen.
Rückseite:	Das Argument human führt zum Spiel gegen einen anderen Menschen, es öffnet sich ein Menü zu Auswahl des Spielmodus (Netzwerk/lokal). Das Argument ai führt zum Spiel gegen eine KI, es öffnet sich ein Menü zur Auswahl der Farbe. Variation GUI: Die Gegnerauswahl erfolgt über das jeweilige Klicken des Buttons „Human“/ „AI“.
Vorbedingung:	Das neue Spiel ist gestartet.
Nachbedingung:	Der Gegnermodus ist gewählt, falls ein Mensch gewählt wurde erscheint ein Menü zur Auswahl eines Netzwerkspiels oder lokalen Spiels. Falls eine KI gewählt wurde, erscheint ein Menü zur Auswahl der Farbe.
Eingangsbedingung:	Der Benutzer wählt mit dem Argument human einen Menschen und mit dem Argument ai in der Konsole eine KI als Gegner. Variation GUI: Der Benutzer wählt den Gegner durch einen Klick auf den entsprechenden Button.

Netzwerkspiel (Konsole)

Vorderseite:	Als Benutzer möchte ich ein Netzwerkspiel starten.
Rückseite:	Das Argument nw-*IP-adress* führt zur Verbindung mit einem anderen Netzwerk, es erscheint eine Menüauswahl zur Farbwahl.
Vorbedingung:	Die Gegnerwahl beträgt Mensch.
Nachbedingung:	Eine Verbindung zu einem anderen Netzwerk ist hergestellt und ein Menü zur Farbauswahl erscheint.
Eingangsbedingung:	Der Benutzer wählt mit dem Argument nw-*IP-adress* den Modus Netzwerkspiel aus.

Netzwerkspiel (GUI)

Vorderseite:	Als Benutzer möchte ich ein Netzwerkspiel starten.
Rückseite:	Der Klick auf den Button „Network Game“ öffnet ein Feld zur Eingabe der IP-Adresse des Gegners, nach erfolgreicher Verbindung erscheint eine Menüauswahl zur Farbwahl.
Vorbedingung:	Die Gegnerwahl beträgt Mensch.
Nachbedingung:	Eine Verbindung zu einem anderen Netzwerk ist hergestellt und ein Menü zur Farbauswahl erscheint.
Eingangsbedingung:	Der Benutzer wählt mit einem Klick auf den Button „Network Game“ den Modus Netzwerkspiel aus.

Lokales Spiel

Vorderseite:	Als Benutzer möchte ich ein lokales Spiel starten.
Rückseite:	Das Argument local führt zu einem Spiel im lokalen Netzwerk, es öffnet sich ein Menü zur Farbauswahl. Variation GUI: Ein Klick auf den Button „Local Game“ führt zu einem Spiel im lokalen Netzwerk und es öffnet sich ein Menü zur Farbauswahl.
Vorbedingung:	Es wurde sich für einen Gegner entschieden.
Nachbedingung:	Es wurde ein Spiel im lokalen Netzwerk gewählt.
Eingangsbedingung:	Der Benutzer wählt mit dem Argument local ein lokales Spiel als Modus. Variation GUI: Der Benutzer klickt auf den Button „Local Game“.

Farbwahl

Vorderseite:	Als Benutzer möchte ich eine Farbwahl treffen, um das Schachspiel zu starten.
Rückseite:	Das Argument white führt zum Start des Spiels mit der Farbe Weiß für den Benutzer und Schwarz für den Gegner, das Argument black zur Farbe Schwarz für den Benutzer und Weiß für den Gegner. Weiß beginnt mit dem ersten Zug. Variation GUI: Der Klick auf den Button „White“ startet ein Spiel mit der Farbe Weiß für den Benutzer und der Farbe Schwarz für den Gegner, mit einem Klick auf den Button „Black“ startet ein Spiel mit der Farbe schwarz für den Benutzer und der Farbe Weiß für den Gegner. Weiß beginnt mit dem ersten Zug.
Vorbedingung:	Es wurde sich für ein Netzwerk entschieden.
Nachbedingung:	Eine Farbe ist ausgewählt und das Schachspiel startet.
Eingangsbedingung:	Der Benutzer wählt mit dem Argument white die Farbe Weiß und mit dem Argument black die Farbe Schwarz. Variation GUI: Der Benutzer wählt mit dem Klick auf den Button „White“ weiß als Farbe, mit einem Klick auf den Button „Black“ schwarz als Farbe.

Zug durchführen (Konsole)

Vorderseite:	Als Benutzer möchte ich ein Zug durchführen.
Rückseite:	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 Benutzer gibt in die Konsole ein Argument [letter][number]-[letter][number] ein.

Zug durchführen (GUI)

Vorderseite:	Als Benutzer möchte ich ein Zug durchführen.
Rückseite:	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 Benutzer klickt mit der Maus auf die zu bewegende Figur und anschließend auf ein Schachfeld.

Geschlagene Figuren anzeigen (Konsole)

Vorderseite:	Als Benutzer möchte ich die schon geschlagenen Figuren angezeigt bekommen.
Rückseite:	Das Argument beaten listet die geschlagenen Figuren als Konsolenausgabe auf.
Vorbedingung:	Ein aktives Schachspiel.
Nachbedingung:	Die geschlagenen Figuren sind auf der Konsole ausgegeben.
Eingangsbedingung:	Der Benutzer gibt beaten in die Konsole ein.

Geschlagene Figuren anzeigen (GUI)

Vorderseite:	Als Benutzer möchte ich die schon geschlagenen Figuren sehen können.
Rückseite:	Die Figuren erscheinen am Rand des Spielfeldes, nachdem sie geschlagen wurden.
Vorbedingung:	Eine Schachfigur wurde geschlagen.
Nachbedingung:	Die geschlagenen Figuren sind neben dem Spielfeld zu sehen.
Eingangsbedingung:	keine

Spiel beenden

Vorderseite:	Als Benutzer möchte ich das aktive Schachspiel beenden.
Rückseite:	Das Argument end beendet das Spiel und zeigt das Auswahlmenü zum Starten bzw. Laden eines Schachspiels. Variation GUI: Ein Klick auf den Button „End Game“.
Vorbedingung:	Ein aktives Schachspiel.
Nachbedingung:	Das aktive Schachspiel wurde beendet und es erscheint das Auswahlmenü zum Starten bzw. Laden eines Schachspiels.
Eingangsbedingung:	Der Benutzer gibt end in die Konsole ein. Variation GUI: Der Benutzer klickt auf den Button „End Game“.

Zug rückgängig machen

Vorderseite:	Als Benutzer möchte ich beliebig viele Züge rückgängig machen und einen rückgängig gemachten Zug wiederherstellen können.
Rückseite:	Das Argument undo macht einen Zug rückgängig. Das Argument redo stellt einen rückgängig gemachten Zug wieder her. Variation GUI: Analog ein Klick auf den Button „Undo“ bzw. „Redo“.
Vorbedingung:	Ein Zug wurde getätigt.
Nachbedingung:	Ein Zug wurde rückgängig gemacht oder ein rückgängig gemachter wieder hergestellt.
Eingangsbedingung:	Der Benutzer gibt undo bzw. redo in die Konsole ein. Variation GUI: Der Benutzer klickt auf den Button „Undo“ bzw. „Redo“.

Spiel speichern

Vorderseite:	Als Benutzer möchte ich den Spielstand speichern.
Rückseite:	Das Argument save speichert den Spielstand im Verzeichnis des Skripts. Variation GUI: Ein Klick auf den Button „Save Game“ speichert den Spielstand im Verzeichnis des Skripts.
Vorbedingung:	Ein aktives Schachspiel.
Nachbedingung:	Das Spiel ist gespeichert und wieder spielbar.
Eingangsbedingung:	Der Benutzer gibt das Argument save in die Konsole ein. Variation GUI: Der Benutzer klickt auf den Button „Save Game“.

Spiel laden

Vorderseite:	Als Benutzer möchte ich ein gespeichertes Spiel laden, um es weiterzuspielen.
Rückseite:	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“ zeigt ein Dialogfenster mit den gespeicherten Spielen an.
Vorbedingung:	Das Programm ist gestartet und es wurde ein Spiel gespeichert.
Nachbedingung:	Ein Spiel mit dem gespeicherten Spielstand und Einstellungen ist geladen und kann weitergespielt werden.
Eingangsbedingung:	Der Benutzer gibt das Argument load in die Konsole ein. Variation GUI: Der Benutzer klickt auf den Button „Load Game“.

Sprache ändern

Vorderseite:	Als Benutzer möchte ich die Sprache von Englisch auf Deutsch (und andersherum) ändern.
Rückseite:	Das Argument language ändert die jeweilige Sprache in die andere Sprache. Variation GUI: Ein Klick auf den Button „Language“ ändert die jeweilige Sprache.
Vorbedingung:	Das Programm ist gestartet.
Nachbedingung:	Die Sprache ist geändert.
Eingangsbedingung:	Der Benutzer gibt das Argument language in die Konsole ein. Variation GUI: Der Benutzer klickt auf den Button „Language“.

Programm beenden

Vorderseite:	Als Benutzer möchte ich das Programm beenden.
Rückseite:	Das Argument quit beendet das Programm. Variation GUI: Ein Klick auf den roten „Exit“-Button beendet das Programm.
Vorbedingung:	Das Programm ist gestartet.
Nachbedingung:	Das Programm ist beendet.
Eingangsbedingung:	Der Benutzer gibt das Argument quit in die Konsole ein. Variation GUI: Der Benutzer klickt auf den roten „Exit“-Button.