



INTERNATIONALE
HOCHSCHULE



Anforderungsdokument – DLMCSPSE01_D Projekt: Software Engineering

Erstellt von:	Dustin Lucht
Matrikelnummer:	92203340
Studiengang:	Master of Science Informatik
Tutor:	Markus Kleffmann
Datum:	17.12.2023

1 Management Summary

Dieses Projekt fokussiert sich auf die Entwicklung eines innovativen interaktiven Schachspiels, welches das klassische Schach durch die Integration von Power-Ups bereichert. Diese neuen Elemente ermöglichen es Spielern, unkonventionelle strategische Züge zu tätigen und verleihen dem Spiel eine einzigartige Dynamik.

Das Hauptziel ist es, ein ansprechendes und benutzerfreundliches Schachspiel zu schaffen, das sowohl traditionelle Schachliebhaber als auch Gelegenheitsspieler anspricht. Die Power-Ups sollen eine frische Herausforderung und Unterhaltung in das traditionelle Schachspiel bringen.

Nutzen und Vorteile

- Innovatives Spielerlebnis: Durch die Power-Ups wird eine neue Ebene der Taktik und Spannung eingeführt.
- Anpassbare Schwierigkeitsstufen: Die KI-Anpassung ermöglicht es Spielern aller Erfahrungsstufen, das Spiel zu genießen.
- Zwei Spielmodi: Ein Modus gegen den Computer (Stockfish) und ein Modus in welchen man gegen einen Freund spielen kann.

Zielgruppe

Das Spiel richtet sich an eine breite Zielgruppe – von Schachenthusiasten bis hin zu Gelegenheitsspielern, die nach einer neuen Herausforderung suchen.

Projektumfang

Das Projekt umfasst die Entwicklung der Spiellogik, die Integration der Power-Ups, das Design der Benutzeroberfläche und die Anpassung der KI.

Zeitplan und Meilensteine

- Anfangsphase (Woche 1-2): Konzeptentwicklung und Aufbau der Projektstruktur.
- Mittlere Phase (Woche 3-5): Implementierung der Spiellogik und Integration der Power-Ups.
- Endphase (Woche 6): Optimierung, Tests und Finalisierung.

Potenzielle Risiken und Herausforderungen

- Komplexität der Power-Up-Implementierung: Planung und regelmäßige Code-Reviews zur Risikominderung.
- Rechenleistung: Performance-Tests zur Optimierung.

- Balancing und Benutzererfahrung: Usability-Tests und Feedback-Einholung für ein ausgewogenes Spielerlebnis

2 Systemumfang und Kontext

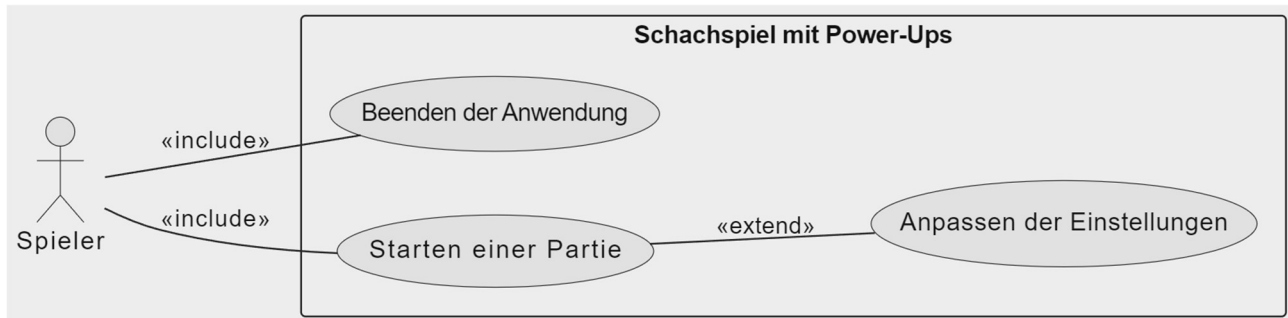


Abbildung 1: UML-Anwendungsfalldiagramm

3 Funktionale Anforderungen

Kennung:	1
Titel:	Neue Partie starten
Beschreibung:	Der Spieler möchte eine neue Partie gegen den Computer beginnen.
Aktoren:	Spieler
Vorbedingungen:	<ul style="list-style-type: none"> • Die Anwendung ist gestartet und die Benutzeroberfläche ist sichtbar.
Nachbedingungen:	<ul style="list-style-type: none"> • Eine neue Partie gegen den Computer hat begonnen, und der Spieler ist am Zug.
Ablauf:	<ol style="list-style-type: none"> 1. Der Spieler wählt die Option "Start". 2. Der Spieler gelangt ins „Pre-Game“ Menü. 3. Der Spieler wählt „Spiele gegen Stockfish“ aus und drückt auf „Partie starten“. 4. Das System generiert ein neues Schachbrett und platziert die Figuren gemäß den Schachregeln. 5. Der Computer wird als Gegner ausgewählt und eine KI wird aktiviert. 6. Der Spieler beginnt die Partie.

Tabelle 1: Use Case 1

Kennung:	2
Titel:	Neue Partie starten
Beschreibung:	Die beiden Spieler möchten eine neue Partie Gegeneinander beginnen.
Aktoren:	2 Spieler
Vorbedingungen:	<ul style="list-style-type: none"> Die Anwendung ist gestartet und die Benutzeroberfläche ist sichtbar.
Nachbedingungen:	<ul style="list-style-type: none"> Eine neue Partie für zwei Spieler hat begonnen, und der erste Spieler ist am Zug.
Ablauf:	<ol style="list-style-type: none"> Der erste Spieler wählt die Option "Neue Partie starten". Der erste Spieler gelangt ins „Pre-Game“ Menü. Der erste Spieler wählt „Spiel gegen einen Freund“ aus und drückt auf „Partie starten“. Das System generiert ein neues Schachbrett und platziert die Figuren gemäß den Schachregeln. Der erste Spieler beginnt die Partie.

Tabelle 2: Use Case 2

Kennung:	3
Titel:	Figuren bewegen und mögliche Züge anzeigen
Beschreibung:	Der Spieler möchte eine Schachfigur bewegen und benötigt eine Anzeige der möglichen Züge.
Aktoren:	Spieler
Vorbedingungen:	<ul style="list-style-type: none"> Der Spieler ist am Zug, und es wurden noch keine Züge getätigt.
Nachbedingungen:	<ul style="list-style-type: none"> Die ausgewählte Figur wurde bewegt, und die möglichen Züge wurden dem Spieler vor der Bewegung angezeigt.
Ablauf:	<ol style="list-style-type: none"> Der Spieler wählt eine Figur auf dem Schachbrett aus. Das System hebt alle möglichen Züge für die ausgewählte Figur hervor. Der Spieler wählt einen der möglichen Züge aus.

	4. Das System bewegt die Figur auf das entsprechende Feld und aktualisiert die Stellung auf dem Schachbrett.
--	--------------------------------------------------------------------------------------------------------------

Tabelle 3: Use Case 3

Kennung:	4
Titel:	Power-Up einsetzen
Beschreibung:	Der Spieler möchte ein Power-Up während des Spiels einsetzen, um einen strategischen Vorteil zu erlangen.
Aktoren:	Spieler
Vorbedingungen:	<ul style="list-style-type: none"> • Der Spieler ist am Zug. • Der Spieler hat mindestens ein Power-Up zur Verfügung.
Nachbedingungen:	<ul style="list-style-type: none"> • Das Power-Up wurde eingesetzt, und die Partie wird fortgesetzt.
Ablauf:	<ol style="list-style-type: none"> 1. Der Spieler wählt die Option "Power-Up einsetzen". 2. Das System zeigt dem Spieler die verfügbaren Power-Ups an. 3. Der Spieler wählt das gewünschte Power-Up aus. 4. Das System führt die entsprechende Aktion gemäß dem ausgewählten Power-Up aus.

Tabelle 4: Use Case 4

Kennung:	5
Titel:	Schachmatt erkennen
Beschreibung:	Das Spiel soll erkennen, ob eine der Könige schachmatt gesetzt wurde und die Partei somit vorbei ist.
Aktoren:	Spieler
Vorbedingungen:	<ul style="list-style-type: none"> • Es ist der Zug des Spielers. • Der Spieler hat einen möglichen Zug durchgeführt.
Nachbedingungen:	<ul style="list-style-type: none"> • Die Partie endet, und der entsprechende Spieler gewinnt.
Ablauf:	<ol style="list-style-type: none"> 1. Das Spiel erkennt, dass ein Spieler im Schachmatt steht und beendet das Spiel. 2. Es erscheint eine entsprechende Nachricht auf dem Bildschirm mit der Möglichkeit ins Hauptmenü zurückzukehren, oder die Partei erneut zu starten.

Tabelle 5: Use Case 5

Kennung:	6
Titel:	Unentschieden erkennen
Beschreibung:	Das Spiel soll erkennen, ob ein Unentschieden vorliegt.
Aktoren:	Spieler
Vorbedingungen:	<ul style="list-style-type: none"> • Es ist der Zug des Spielers. • Der Spieler hat einen möglichen Zug durchgeführt.
Nachbedingungen:	<ul style="list-style-type: none"> • Die Partie endet unentschieden.
Ablauf:	<ol style="list-style-type: none"> 1. Das Spiel erkennt, dass ein Unentschieden vorliegt (z.B. durch dreifache Stellungswiederholung oder 50-Züge-Regel). 2. Es erscheint eine entsprechende Nachricht auf dem Bildschirm mit der Möglichkeit ins Hauptmenü zurückzukehren, oder die Partei erneut zu starten.

Tabelle 6: Use Case 6

Kennung:	7
Titel:	Einstellungen ändern
Beschreibung:	Der Spieler möchte die Einstellungen des Spiels anpassen, um seine Präferenzen zu berücksichtigen.
Aktoren:	Spieler, Einstellungen
Vorbedingungen:	<ul style="list-style-type: none"> • Die Anwendung ist gestartet und die Benutzeroberfläche ist sichtbar.
Nachbedingungen:	<ul style="list-style-type: none"> • Die ausgewählten Einstellungen werden gespeichert und auf das laufende Spiel angewendet.
Ablauf:	<ol style="list-style-type: none"> 1. Der Spieler navigiert zu den Einstellungen vor den Beginn eines Spiels. 2. Das System zeigt dem Spieler verschiedene Optionen an, die angepasst werden können, wie z.B. die Schwierigkeit des Computers oder mit welcher Farbe man beginnen möchte. 3. Der Spieler wählt die gewünschten Einstellungen aus und kann eine Partie starten.

Tabelle 7: Use Case 7

Kennung:	8
-----------------	---

Titel:	Spiel beenden
Beschreibung:	Der Spieler soll das Spiel während eines Spiels und aus dem Hauptmenü heraus beenden können.
Aktoren:	Spieler
Vorbedingungen:	<ul style="list-style-type: none"> Die Anwendung ist gestartet, und entweder ist eine Partie aktiv, oder der Spieler befindet sich im Hauptmenü.
Nachbedingungen:	<ul style="list-style-type: none"> Das Spiel ist beendet, und die Anwendung ist geschlossen oder der Spieler ist zurück im Hauptmenü.
Ablauf:	<ol style="list-style-type: none"> Der Spieler wählt die Option „Spiel beenden“ aus dem Hauptmenü oder während eines laufenden Spiels aus. Wenn ein Spiel aktiv ist, fragt das System, ob der Spieler den aktuellen Fortschritt speichern möchte, bevor das Spiel beendet wird. Der Spieler bestätigt, dass das Spiel beendet und die Anwendung geschlossen werden soll, oder er wählt aus, zum Hauptmenü zurückzukehren. Das System führt die gewählte Aktion aus, beendet das Spiel und schließt die Anwendung oder kehrt zum Hauptmenü zurück.

Tabelle 8: Use Case 8

4 Nicht-funktionale Anforderungen

- Benutzeroberfläche
 - Die Benutzeroberfläche soll intuitiv und benutzerfreundlich gestaltet sein.
 - Die Anwendung soll auf verschiedenen Bildschirmgrößen und -auflösungen gut lesbar und bedienbar sein.
- Leistung
 - Das Spiel soll auf Windows-Plattformen flüssig und ohne spürbare Verzögerungen laufen (Ausnahme bildet ein hoch gesetzter Schwierigkeitsgrad des Computers).
 - Die Ladezeiten sollen minimal sein.
- Grafische Darstellung
 - Trotz fehlenden UI-Artists soll die Oberfläche ansprechend aussehen.
 - Die Figuren sollen eindeutig erkennbar und differenzierbar sein.
- Plattformkompatibilität
 - Die Anwendung soll auf den Windows-Versionen 10 und 11 reibungslos funktionieren.

5 Glossar

Begriff	Erklärung
Schach (Im Schach stehen)	Schach tritt ein, wenn der König bedroht ist, aber noch mindestens eine legale Fluchtmöglichkeit hat.
Schachmatt	Schachmatt tritt ein, wenn der König bedroht ist und keine legale Fluchtmöglichkeit hat, das Spiel ist dann für den bedrohten Spieler verloren.
Power-Ups	Power-Ups sind spezielle Fähigkeiten oder Gegenstände in einem Spiel, die dem Spieler vorübergehende Vorteile oder besondere Aktionen verleihen. Sie werden eingesetzt, um den Spielverlauf zu beeinflussen.

Tabelle 9: Glossar