

Implementation des Brettspiel-Klassikers Schach als Videospiel

Pflichtenheft

Inhaltsverzeichnis

1	Einleitung	3
2	Rahmenbedingungen.....	4
2.1	Anforderungen an die Zielmaschine	4
2.2	Entwicklungsumgebung	4
2.3	Anwendungsbereiche	4
3	Funktionale Anforderungen.....	5
3.1	Muss-Kriterien.....	5
3.3	Kann-Kriterien	5
3.4	Abgrenzungskriterien.....	5
4	Funktionalität	6
4.1	Grafische Benutzeroberfläche	6
4.2	Abnahmekriterien	7
5	Qualitätsziele	8

1 Einleitung

Das Brettspiel Schach ist in der europäischen Kultur seit über 700 Jahren fest verwurzelt und erfreut sich auch im Zeitalter der Computertechnik immer noch großer Beliebtheit. Das Spielprinzip ist einfach zu lernen, und schwer zu meistern: Zwei Spieler starten mit einem Set festgelegter Spielfiguren auf festgelegten Positionen auf einem karierten Brett. Alle Figurenarten haben vorgegebene Bewegungsmöglichkeiten und das Ziel ist es, den gegnerischen König zu "schmeißen", und so ein "Schach Matt" zu erzielen.

Im Zuge des Softwarepraktikums der Universität Würzburg soll nun der analoge Spieleklassiker ins digitale Zeitalter gehoben werden. Dazu soll der Nutzer Schachfiguren auf einem digitalen Schachbrett bewegen, und gegen andere Spieler im Hotseat-Modus oder den Computer antreten können. Wir hoffen allen Spielern mit unserem Spiel viele angenehme Spielstunden zu bescheren. Und auch nach einer verlorenen Partie nicht vergessen: "Dem wahren Schachspieler zählt eine schöne Partie mehr als ein Sieg"! (Fernando Arrabal)

2 Rahmenbedingungen

2.1 Anforderungen an die Zielmaschine

- Auf dem Rechner muss Java 11 und JavaFX installiert und lauffähig sein.
- Tastatur, Maus, Bildschirm
- 64-bit CPU Architektur
- 1920x1080 Mindestauflösung

2.2 Entwicklungsumgebung

- IntelliJ IDEA
- Java 11 mit JavaFX
- Windows / GNU + Linux

2.3 Anwendungsbereiche

- Unterhaltung

3 Funktionale Anforderungen

3.1 Muss-Kriterien

- Es muss ein Menü vorhanden sein.
- Es müssen nur regelkonforme Spielzüge möglich sein.
- Es muss mindestens eine Animation erstellt werden.
- Es muss das Spielen gegen eine KI möglich sein.
- Es muss das Spielen gegen einen anderen Spieler möglich sein.
- Im Laufe der Entwicklung wird es eine unerwartete Requirement-Änderung geben.
- Es muss eine strikte Model-View-Controller Architektur implementiert werden.
- Für das GUI:
 - Es muss das Spielfeld mit den Spielfiguren angezeigt werden.
 - Es muss erkennbar sein, welcher Spieler momentan am Zug ist.
 - Es muss erkennbar sein, welche Züge möglich sind.

3.3 Kann-Kriterien

- Der Spielstand kann gespeichert und geladen werden.
- Zug-Geschichte

3.4 Abgrenzungskriterien

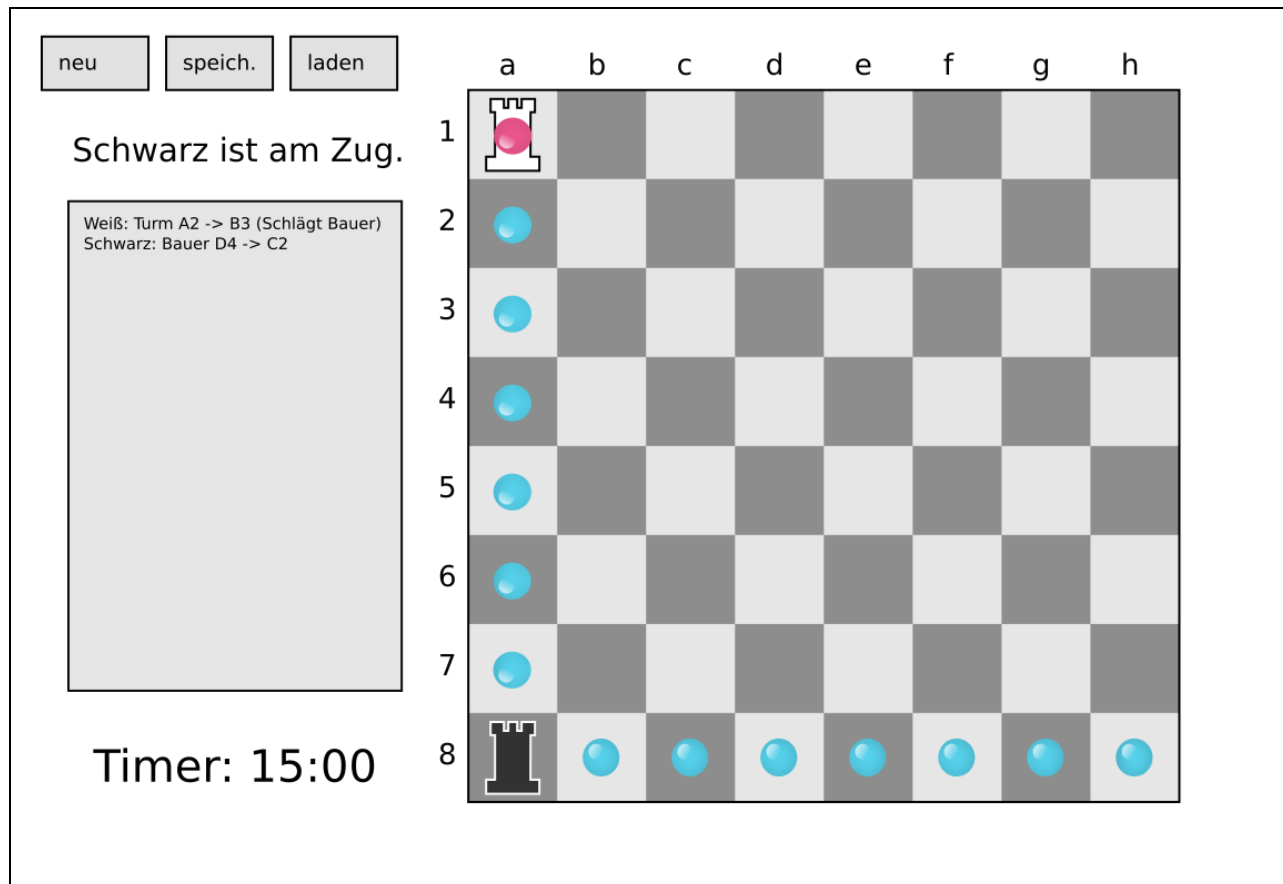
- Es muss kein Online-Spiel möglich sein.
- Es sollen keine Graphiken erstellt werden.
- Es muss kein eigener Sound designed werden.

4 Funktionalität

4.1 Grafische Benutzeroberfläche

Die folgenden Mockups sollen die grafische Benutzeroberfläche des Spiels zeigen:

Für das Hauptfenster:



Das graue Rechteck zur linken Seite des Fensters stellt die Zuggeschichte dar. Die blauen bzw. roten Kreise auf dem Spielfeld zeigen die möglichen Spielzüge für die vom schwarzen Spieler ausgewählte Spielfigur dar. Links unten befindet sich ein optionaler Timer, der beim Start des Spiels ggf. aktiviert werden kann.

Hinweis: Sowohl die Zuggeschichte und der Timer, als auch die Buttons zum Speichern und Laden des Spielstandes gehören zu den Kann-Kriterien.

Für das „neues Spiel“-Fenster:

Rollen:
Weiß:
Schwarz:
Timer: ☒ kein Timer
☐ pro Spiel
☐ pro Zug

4.2 Abnahmekriterien

Bei Abnahme des Projekts durch den Betreuer Simon Eismann soll mindestens der folgende Ablauf des Spiels ohne kritische Fehler möglich sein:

- Programmstart
- Einstellen der Spieleinstellungen („Neues Spiel“-Fenster)
- Spielbeginn
- Auswählen der zu bewegenden Spielfigur durch den Spieler
- Anzeigen der erreichbaren Felder
- Bewegen der Spielfigur
- Bewegen der Spielfigur durch den Gegner (ggf. computergesteuert)
- Schmeißen von gegnerischen Spielfiguren
- Regelkonformes Beenden des Spiels
- „Game Over“-Anzeige

5 Qualitätsziele

- Das Spiel soll stabil und spielbar sein.
- Die Codebase soll wartbar sein.
- Es soll ein komplettes Benutzerhandbuch erstellt werden, in dem die Bedienung des Spiels dokumentiert wird.
- Es soll ein Developerhandbuch erstellt werden, in dem eventuell für das Spiel entwickelte Dateiformate oder Ähnliches dokumentiert werden.