Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Implementation des Brettspiel-Klassikers Schach als Videospiel

Pflichtenheft

von: Adrian Samoticha, Simon Trapp, Mohamad Halloway

Inhaltsverzeichnis

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

Die Rahmenbedingung für dieses Projekt lassen sich in folgende Punkte untergliedern:

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

- Die Anwendung dient ausschließlich der Unterhaltung.

3 Funktionale Anforderungen

3.1 Muss-Kriterien

- Es muss ein Menü vorhanden sein.
- Es sollen 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.
- Es muss eine strikte Model-View-Controller Architektur implementiert werden.
- Beim Bewegen einer Spielfigur soll ein Soundeffekt abgespielt 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, beispielsweise indem beim Anklicken einer Spielfigur alle Zielfelder aufleuchten.
- Es sollen folgende Schachregeln gelten:
    - Es gelten für alle Spielfiguren die allbekannten Bewegungsmöglichkeiten. Weiß soll immer den ersten Zug machen dürfen.
    - Bauern sollen sich nur in Richtung des gegnerischen Spielfeldes bewegen können. Sie sollen sich jeweils nur um ein Feld bewegen können, ausgenommen beim ersten Zug, bei dem auch ein Sprung von zwei Feldern möglich ist. Erreicht ein Bauer das Ende des gegnerischen Feldes, kann der Spieler den Bauern wahlweise durch eine Dame, ein Pferd, einen Turm oder einen Bischoff eintauschen.
    - Rochade: Ein Turm soll den König „überspringen“ können. Dies soll nur dann möglich sein, wenn der König soeben seinen ersten Zug machte (und dieser in Richtung des Turmes ging) und der Turm zuvor keinen Zug machte. Der König soll sich währenddessen nicht in Schach befinden dürfen, oder in Schach geraten.
    - En Passant: Ein Bauer kann einen anderen Bauern seitlich (nicht diagonal) schlagen, wenn dieser soeben um zwei Felder gesprungen ist.
    - Bei 50 aufeinanderfolgenden Zügen, bei denen weder ein Bauer bewegt, noch eine Spielfigur geschmissen wird, soll unentschieden gelten.
    - Im Falle eines Schachs, soll der angegriffene Spieler darauf automatisch hingewiesen werden. Während eines Schachs sind nur Züge möglich, die den Spieler, der momentan am Zug ist, aus dem Schach befreien.

3.3 Kann-Kriterien

- Der Spielstand kann gespeichert und geladen werden.
- Zug-Geschichte
- Beim Schmeißen soll ein anderer Soundeffekt abgespielt werden, als bei einem gewöhnlichen Zug.
- Erreicht ein Bauer das Ende des gegnerischen Spielfeldes, wird ein Soundeffekt abgespielt.
- Bei einem Sieg oder einem Verlust soll ein entsprechender Soundeffekt abgespielt werden.
- Es soll mehrere Schwierigkeitsgrade für die KI geben.

- Es soll einen optionalen Timer geben, der die Spielzeit je nach Präferenz des Spielers für jede Runde oder für das ganze Spiel begrenzt.
    - Nach Ablauf des Timers soll der Spieler verloren haben. Sollten zwei Spieler gegeneinander spielen (d.h. es gibt keine KI), so soll der Timer nur auf „pro Zug“ und nicht auf „pro Spiel“ gestellt werden können.
- Kommt die gleiche Position des Spiels dreimal im Spiel vor, so soll ein unentschieden gelten.
- Im Falle eines Unentschieden soll dieses analysiert werden und es soll angezeigt werden, wie dieses zustande kommt.

3.4 Abgrenzungskriterien

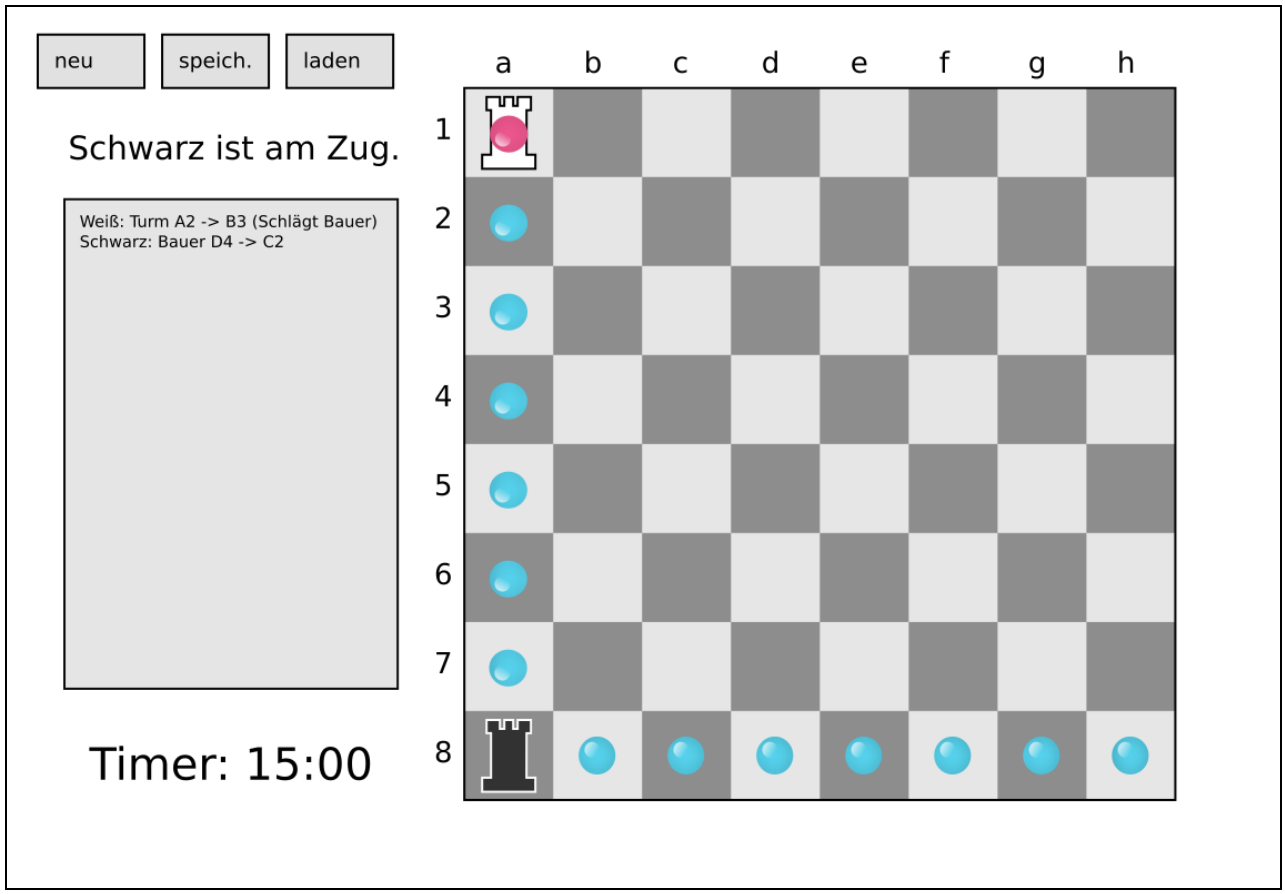
- Es muss kein Online-Spiel möglich sein.
- Es sollen keine Graphiken erstellt werden.
- Es muss kein eigener Sound designed werden.

4 Funktionalität

4.1 Grafische Benutzeroberfläche

Die folgenden Mockups sollen die grafische Benutzeroberfläche des Spiels zeigen:

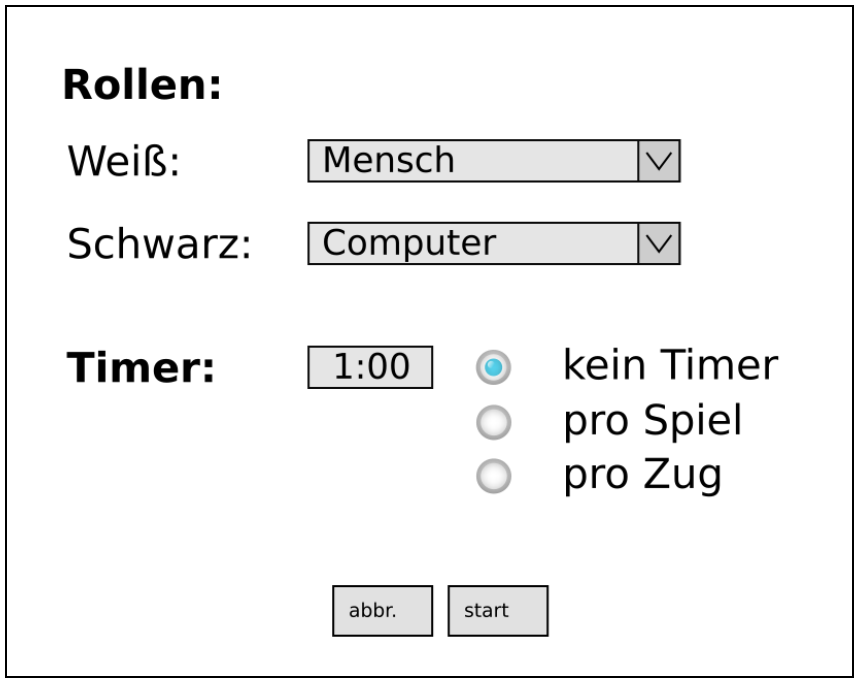
Für das Hauptfenster:



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:



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.