

HOCHSCHULE RHEIN MAIN

SOFTWARETECHNIK PROJEKT

MonsterSchach

VERSION 1.0

Julian Andrlé, Sadia Miah, David Rossel

Contents

1	Projektgrundlagen	2
1.1	Einführung System	2
1.2	Technische Information	2
1.3	Spiefeld	2
1.4	Spielfiguren	2
1.5	Wettstreit	3
1.6	Items (Gegenstände)	3
1.6.1	Buffs/Debuffs	3
1.6.2	Falle	3
1.7	Spielziel	4
2	Abläufe und Funktionen	4
2.1	Anwendungsszenarios	4
2.1.1	Wettkampf + Item benutzen	4
2.1.2	Ende	4
2.1.3	Anfang des Spiels	4
2.2	Anwendungsfälle	4
2.2.1	Anwendungsfall 1	4
2.2.2	Anwendungsfall 2	6
2.2.3	Anwendungsfall 3	7
2.2.4	Anwendungsfall 4	8
2.2.5	Anwendungsfall 5	9
2.2.6	Anwendungsfall 6	9
3	Daten / Domänenmodell	10
4	Glossar	10

1 Projektgrundlagen

1.1 Einführung System

Es wird ein Spiel entworfen, das 3 verschiedenen Typen an Figuren besitzt, welche jeweilige Stärken und Schwächen haben.

1.2 Technische Information

Es werden Java 17 für das Model sowie den Controller und JavaFX für die grafische Benutzeroberfläche verwendet.

1.3 Spielfeld

Das Spielfeld ist in 8x8 Feldern, insgesamt 64, aufgeteilt. Jedes Feld ist quadratisch. Die Horizontale ist mit I bis H und die Vertikale mit 1 bis 8 beschriftet.

1.4 Spielfiguren

Jeder Spieler besitzt vier Spielfiguren. Davon sind zwei sogenannte "Bauern", eine "Königin" und eine "Turmfigur".

- Er kann sich diagonal, horizontal und vertikal bewegen.
- Er darf nur um 1 Feld weiterziehen.
- Wenn er die gegnerische Anfangsreihe erreicht, wird er zum "Turm".
- Die Anfangspositionen der Rang 1 Figuren sind 4 und 5 jeweils in B und G

Die Regeln des "Turms" (Rang 2 Figur) sind folgende:

- Er kann sich nur horizontal und vertikal bewegen.
- Er darf um beliebig viele Felder weiterziehen.
- Die Anfangsposition der Rang 2 Figuren sind 5 auf A und 4 auf H

Die Regeln der "Königin" (Rang 3 Figur) sind folgende:

- Er kann sich diagonal, horizontal und vertikal bewegen.
- Sie darf um beliebig viele Felder weiterziehen.
- Die Anfangsposition der Rang 3 Figuren sind 4 auf A und 5 auf H

Die Spielfiguren müssen stets die oben beschriebenen Regeln einhalten.

1.5 Wettstreit

Wenn zwei Figuren auf dasselbe Feld ziehen, wird ein Wettstreit ausgelöst. Der Angreifende darf das Spiel au

- Das Rechenspiel
 - Das Ziel des Rechenspiels ist es, in der schnelleren Zeit die Rechenaufgabe zu lösen.
- Klick die Kreise-Event
 - Das Ziel dieses Wettstreits ist es, die meisten verschwundenen Kreise in einer bestimmten Zeit angeklickt zu haben.

1.6 Items (Gegenstände)

Die Items sind auf zufälligen Feldern am Anfang des Spiels verteilt und werden bei geteilten Feldern mit Spiel

1.6.1 Buffs/Debuffs

- Schild
 - Es kann bei einem verlorenem Wettstreit einmalig eingesetzt werden und verhindert dann, dass die Spielfigur vom Feld genommen wird. Daraufhin wird ein weiterer Wettstreit gestartet.
 - Item kann nicht abgelegt werden
- Spielverschwerer
 - Wird beim Aufnehmen ins Inventar gelegt und beim nächsten Wettstreit konsumiert.
 - Bei dem "Rechenspiel" werden hier zur Endzeit 10 Prozent hinzugezählt.
 - Bei dem "Klick die Kreise-Event" werden 10 Prozent von der Gesamtzeit abgezogen.
 - Item kann nicht abgelegt werden
- Rang Upgrade/Downgrade
 - Wird beim Aufnehmen direkt konsumiert.
 - Wenn der Rang einer Figur 1 oder 2 ist, wird der Rang um 1 erhöht und bei der Rang 3 Figur wird der Rang um 1 verringert.

1.6.2 Falle

Wird beim ersten Aufnehmen in das Inventar gelegt, kann danach in einem beliebigen Zug auf einem benachb

- Bewegungsfalle
 - Die aktivierte Falle macht die Figur für 3 Züge bewegungsunfähig.

- Teleportierfalle
 - Die aktivierte Falle lässt die Figur auf die rangniedrigste Figur des Gegners teleportieren und einen Wettstreit mit jener auslösen.
 - Der Fallenauslöser fungiert hier jedoch als Verteidiger und darf somit das Spiel nicht aussuchen.

1.7 Spielziel

Das Spiel ist gewonnen, wenn alle Spielfiguren des Gegners vom Feld genommen wurden.

2 Abläufe und Funktionen

2.1 Anwendungsszenarios

2.1.1 Wettkampf + Item benutzen

Der erste Spieler zieht mit seinem Turm von E4 auf E8, fährt dabei über ein Item und wird somit auf E6 gest

2.1.2 Ende

Spieler 1 bewegt seinen Bauer zum Feld A5. Spieler 2 bewegt seine Königin zum Feld A5. Ein Wettstreit wird

2.1.3 Anfang des Spiels

Einer der beiden Spieler Klickt auf das Feld "Neues Spiel Starten". Anschliesend kommt ein Popupfenster mi

2.2 Anwendungsfälle

2.2.1 Anwendungsfall 1

- Titel: Wettkampf spielen
- Akteure: Spieler
- Fachlicher Auslöser: Zwei Spielfiguren befinden sich auf dem gleichen Spielfeld
- Standardablauf:
 - 1. System: System fragt den Angreifer welches der beiden Spiele er spielen möchte
 - 2. Spieler: Wählt eins der beiden verfügbaren Spiele aus
 - 3. System: lädt das Spiel, das ausgewählt wurde und gibt eine zufällige Aufgabe aus
 - 4. Spieler: Beendet das Spiel in der, für ihn, schnellstmöglichen Zeit

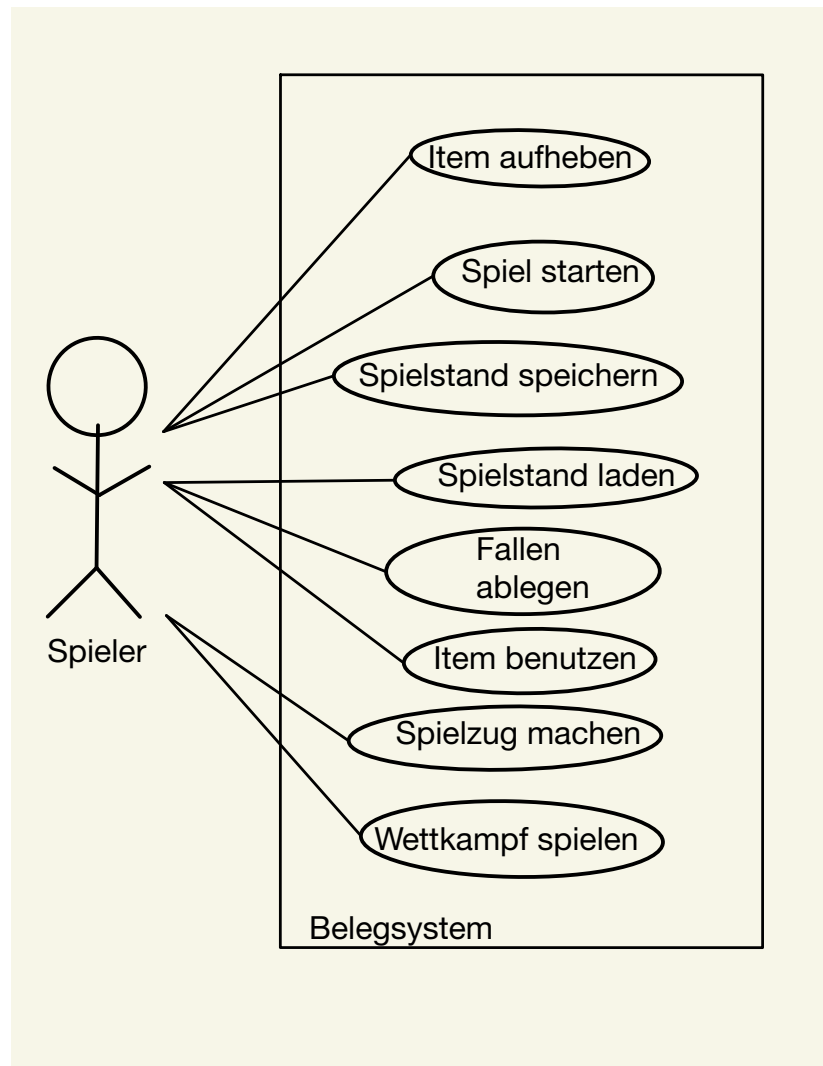


Figure 1: Anwendungsdiagramm

Der Spieler interagiert mit dem System, befindet sich jedoch außerhalb des Systems. Die Linien verbinden den

- 5. System: speichert die Zeit in der das Spiel beendet wurde und wiederholt den Ablauf ab 3. für den anderen Spieler
- 6. System: vergleicht beide Ergebnisse und gibt den besseren als Gewinner aus
- Alternative Abläufe / Fehlersituationen / Sonderfälle:
 - 4a Spieler schafft es nicht die Aufgabe zu lösen
 - * 4a1 System: gibt den anderen Spieler dieselbe Aufgabe
 - * 4a2 Spieler: löst die Aufgabe knapp
 - * 4a3 der zweite Spieler wird als Gewinner deklariert
- Nachbedingung/Ergebnis: System geht in Zustand "Figur entfernen" über
- Nicht-funktionale Anforderungen: Reaktionszeit weniger als 4 sek.
- Parametrisierbarkeit / Flexibilität:
- Nutzungshäufigkeit / Mengengerüst: mindestens 3x pro Spiel und höchstens so oft bis alle Figuren eines Spielers geworfen wurden

2.2.2 Anwendungsfall 2

- Titel: Spielstand laden
- Akteure: Spieler
- Fachlicher Auslöser: Bei starten des Programms wird Spiel laden ausgewählt
- Standardablauf:
 - 1. Spieler: Auswahl des Spielstandes, der geladen werden soll
 - 2. System: Den Spielstand aus einer Datenbank laden
 - 3. System: Die Darstellung des Spielstandes
- Alternative Abläufe / Fehlersituationen / Sonderfälle:
 - 2a System: findet den Spielstand in der Datenbank nicht
 - * 2a1 System: Zeigt error an, dass der Spielstand entweder nicht existiert oder korumpiert ist
 - * 2a2 System: Navigiert zurück zu der Startseite
 - * 2a3 weiter bei 1
- Nachbedingung/Ergebnis: System geht in Zustand "Spiel starten"
- Nicht-funktionale Anforderungen: Reaktionszeit weniger als 10 sek.
- Parametrisierbarkeit / Flexibilität:
- Nutzungshäufigkeit / Mengengerüst: höchstens 1x pro Spielaufruf

2.2.3 Anwendungsfall 3

- Titel: Einsammeln von Items
- Akteure: Spieler
- Fachlicher Auslöser: Der Spieler möchte die verfügbaren Items im Spiel einsammeln
- Vorbedingungen: Das Spiel läuft bereits und die Items sind auf dem Spielfeld verteilt
- Standardablauf:
 1. Der Spieler bewegt eine Figur auf ein Feld, auf dem sich ein Item befindet
 2. Das Item wird automatisch eingesammelt und ins Inventar des Spielers gelegt
- Alternative Abläufe / Fehlersituationen / Sonderfälle:
 - 1a Der Spieler bewegt sich auf ein Feld, wo kein Item ist
 - 1b Wenn das Feld mit dem Item bereits von einer anderen Figur besetzt ist, kann das Item nicht eingesammelt werden
 - 2a Wenn eine Figur bereits ein Item besitzt, kann sie kein weiteres Item einsammeln
 - 2b Wenn ein Item direkt aktiviert wird, wird es nicht ins Inventar des Spielers gelegt
- Nachbedingung/Ergebnis: Die Figur hat das Item eingesammelt und es steht dem Spieler zur Verfügung
- Nicht-funktionale Anforderungen:
 - Das Spiel sollte reibungslos und ohne Verzögerungen laufen, um ein gutes Spielerlebnis zu gewährleisten
 - Die Items sollten klar gekennzeichnet und leicht verständlich sein, damit die Spieler sie effektiv einsetzen können
 - Das Spiel muss die verschiedenen Item-Arten korrekt erkennen und verwalten können
 - Die Item-Aktionen müssen korrekt ausgeführt werden und dürfen das Spiel nicht zum Absturz bringen
- Parametrisierbarkeit / Flexibilität: Die Anzahl Items können angepasst werden, um das Spiel zu verändern
- Nutzungshäufigkeit / Mengengerüst:

- Die Häufigkeit, mit der Items auf dem Spielfeld verteilt werden, sollte angemessen sein, um den Spielverlauf interessant und abwechslungsreich zu gestalten
- Die Anzahl der verfügbaren Items sollte nicht zu groß sein, um den Spielverlauf nicht zu kompliziert zu machen

2.2.4 Anwendungsfall 4

- Titel: Speichern des Spielstands
- Akteure: Spieler 1, Spieler 2
- Fachlicher Auslöser: Der Spieler möchte den aktuellen Spielstand speichern, um das Spiel später fortsetzen zu können
- Vorbedingungen: Das Spiel ist im Gange und es gibt noch mindestens eine Spielfigur auf dem Spielfeld vom Gegner
- Standardablauf:
 1. Der Spieler wählt die Option "Spielstand speichern"
 2. Das Spiel speichert den aktuellen Spielstand
 3. Das Spiel gibt dem Spieler ein Feedback, dass der Spielstand erfolgreich gespeichert wurde
- Alternative Abläufe / Fehlersituationen / Sonderfälle:
 - 1a Wenn es während des Speicherns des Spielstands zu einem unerwarteten Fehler kommt, wird dem Benutzer eine Fehlermeldung angezeigt und das Spiel wird fortgesetzt
 - 2a Wenn eine Verbindung zum Server verloren geht, kann der Spielstand zu jedem Zeitpunkt gespeichert und geladen werden
 - 2b Wenn beide Spieler das Spiel verlassen, bevor das Spiel beendet ist, kann ebenso der Spielstand zu jedem Zeitpunkt gespeichert und geladen werden
- Nachbedingung/Ergebnis: Der Spielstand wurde erfolgreich gespeichert
- Nicht-funktionale Anforderungen:
 - Das Speichern des Spielstands muss schnell und zuverlässig sein
 - Die grafische Benutzeroberfläche sollte dem Benutzer ein klares Feedback geben, dass der Spielstand erfolgreich gespeichert wurde
- Nutzungshäufigkeit / Mengengerüst: Das Speichern des Spielstands kann jederzeit während eines Spiels auftreten und kann somit in jeder Runde vorkommen

2.2.5 Anwendungsfall 5

- Titel: Spiel Starten
- Akteure: Spieler 1, Spieler 2
- Fachlicher Auslöser: Einer der beiden Spieler Klickt auf das Feld "Neues Spiel Starten" um ein neues Spiel zu initiieren.
- Vorbedingungen: Das Spiel zeigt den Startbildschirm an.
- Standardablauf:
 1. Einer der Spieler wählt die Option "Neues Spiel".
 2. Das Spiel fragt Spieler eins nach seinem Namen.
 3. Spieler eins bestätigt seinen Namen.
 4. Der Vorgang (Punkt 2 und 3) wird Analog für Spieler zwei wiederholt.
- Alternative Abläufe / Fehlersituationen / Sonderfälle:
 1. Einer der Beiden Spieler Bestätigt nicht die Eingabe seines Namens.
 - (a) Das Bestätigungspopup schließt sich.
 - (b) Das Feld für die Eingabe des Namens wird wieder angezeigt und ist leer.
 - (c) Der Spieler gibt seinen Namen in das Feld ein und bestätigt seine Eingabe.
- Nachbedingung/Ergebnis: Das Spiel wurde erfolgreich gestartet und beide Spieler können sich über ihre eingegeben Namen identifizieren.
- Nicht-funktionale Anforderungen: Die Namen von Spieler 1 und 2 dürfen nicht die gleich sein.
- Parametrisierbarkeit / Flexibilität: /
- Nutzungshäufigkeit / Mengengerüst: Einmal pro Spielaufruf.

2.2.6 Anwendungsfall 6

- Titel: Spielzug machen
- Akteure: Spieler
- Fachlicher Auslöser: Ein Spieler Klickt auf das Feld der Figur die Sie bewegen will. Darf sich die Figur bewegen so klickt der Spieler anschließend auf das Feld auf das die Figur laufen soll.
- Vorbedingungen: Spieler verfügt über eine Figur die sich bewegen darf.
- Standardablauf:

1. Spieler wählt die Figur aus die er bewegen will.
 2. Darf sich die Figur bewegen so kann der Spieler das Feld aussuchen auf das sich die Figur bewegen soll.
 3. Die Figur bewegt sich auf das ausgewählte Feld.
 4. Ist auf dem Feld eine gegnerische Spielfigur wird ein Wettstreit ausgelöst. Ist auf dem Spielfeld ein Item oder eine Falle werden die Aktionen für Items ausgelöst. Sind auf dem Feld weder gegnerische Spielfiguren noch Items oder Fallen ist der Spielzug beendet.
- Alternative Abläufe / Fehlersituationen / Sonderfälle:
 1. Der Spieler verfügt nur über eine Spielfigur und diese ist in einer Bewegungsfalle.
 - (a) Dem Spieler wird angezeigt das er keine Figur hat die sich bewegen kann und beendet den Spielzug
 - Nachbedingung/Ergebnis: Der Spieler hat eine Spielfigur bewegt. (Optional: Der Spieler hat eine Aktion ausgelöst. (Falle, Wettstreit, Item))
 - Nicht-funktionale Anforderungen: /
 - Parametrisierbarkeit / Flexibilität: /
 - Nutzungshäufigkeit / Mengengerüst: Mehrmals pro Spiel. Eine genaue Angabe ist nicht möglich, da der Spielverlauf nicht genau absehbar ist.

3 Daten / Domänenmodell

4 Glossar

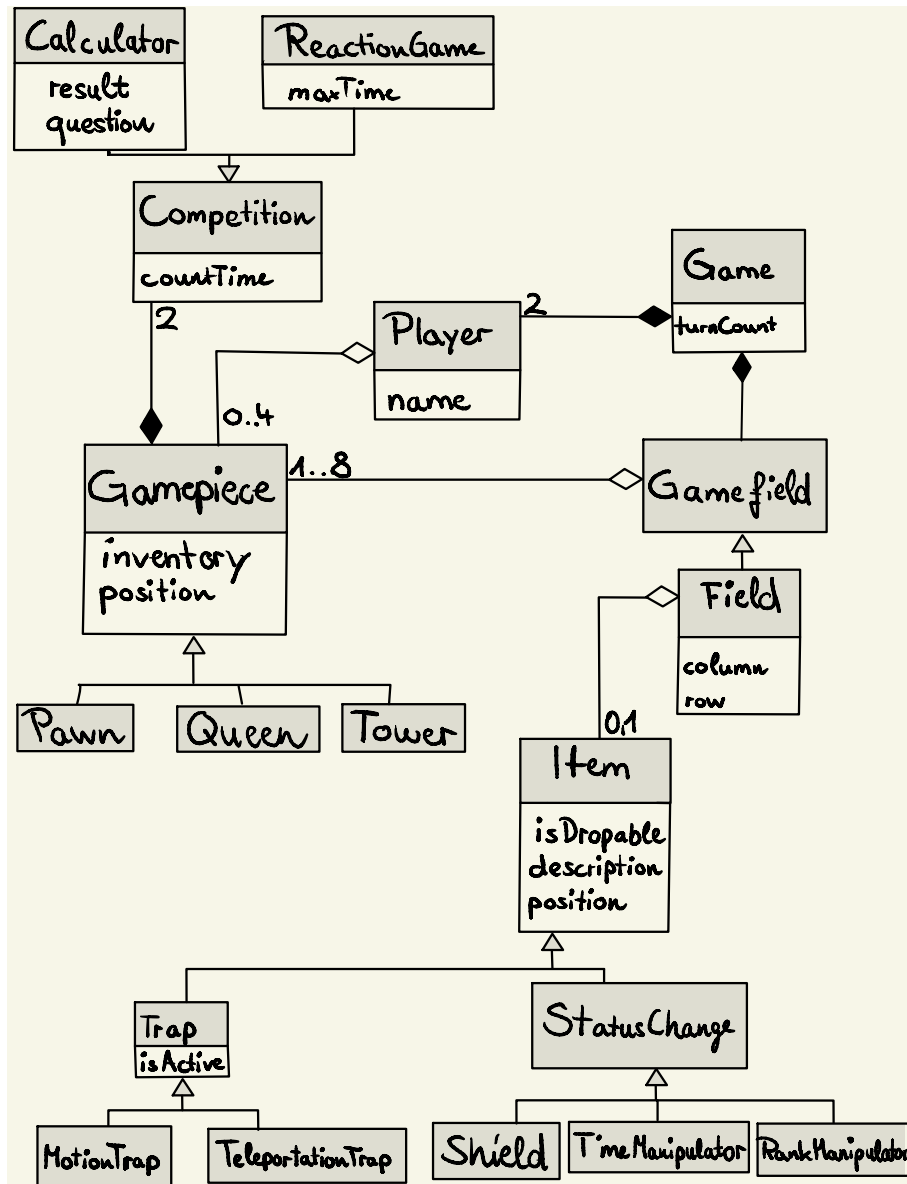


Figure 2: Klassendiagramm