League of Stones

Dokumentation zur Kartenspieldatenbank

Alex Theijs, Luca Tirabassi

Exposee

[Fesseln Sie Ihre Leser mit einem ansprechenden Exposee. Normalerweise ist dies eine kurze Zusammenfassung des Dokuments.   
Wenn Sie Ihre Inhalte hinzufügen möchten, einfach hier klicken und mit der Eingabe beginnen.]

Inhaltsverzeichnis

[1 Konzept 2](#_Toc409848753)

[1.1 Anforderungsanalyse 2](#_Toc409848754)

[1.1.1 Datenbankanalyse 2](#_Toc409848755)

[1.1.2 Benutzeranforderungen 2](#_Toc409848756)

[1.1.3 Spielreglement 2](#_Toc409848757)

[1.2 Klassendiagramm 3](#_Toc409848758)

[1.2.1 Alte Klassendiagramme 3](#_Toc409848759)

[1.2.2 Neue Klassendiagramme 4](#_Toc409848760)

[1.3 Use Cases 5](#_Toc409848761)

[1.3.1 Alte Userspezifische Use Cases 5](#_Toc409848762)

[1.3.2 Alte Spielspezifische Use Cases 5](#_Toc409848763)

[1.3.3 Neue Userspezifische Use Cases 5](#_Toc409848764)

[1.3.4 Neue Spielspezifische Use Cases 5](#_Toc409848765)

# 1 Konzept

## 1.1 Anforderungsanalyse

### 1.1.1 Datenbankanalyse

Wir wollen eine Datenbank für ein Kartenspiel entwerfen, welches über ein Login für mehrere Benutzer gleichzeitig zur Verfügung steht. Dafür brauchen wir ein Securitymanagement und eine Spieldatenbank. Die Securitydatenbank enthält das Usermanagment mit sämtlichen Logins und Userdaten.

Die Spieldatenbank enthält:-

* eine Tabelle mit den enthaltenen Karten
* eine Tabelle mit den Decks die auf den Benutzer zugeschnitten sind(Jeder Benutzer kann sich eine Menge von Karten in ein Deck tuen und mit diesem dann spielen)
* Charaktertabelle welche angezeigt wird
* eine Tabelle für die Spielsession damit mehrere Spieler gleichzeitig spielen können und das Programm nicht cheatanfällig wird und diverse Zwischentabellen.

### 1.1.2 Benutzeranforderungen

Mehrere Spieler gleichzeitig sollen sich einloggen können. Die Benutzer müssen sich Spielsessions erstellen können, welche gleichzeitig von dem Serverclient bearbeitet und gehostet werden. Der Benutzer muss Extrainfos zu seinem Profil eingeben, speichern und verwalten können.

### 1.1.3 Spielreglement

Das Ziel des Spieles ist es, den Gegner in die Knie zu zwingen. Es gibt zwei Wege dies zu tun: entweder, gibt der Gegner auf, oder man bringt sein Leben auf 0.

Vor dem Spielen muss der Spieler sein Deck aussuchen. Zusätzlich zum Deck muss man die zugehörige Rasse aussuchen. Das Deck besteht aus 30 Karten. Bei dem Aufbau des Decks stehen dem Benutzer diverse Basiskarten zur Verfügung.

Jede Rasse hat seine eigenen Basiskarten. Diese Karten können nur von einer Bestimmten Rasse gewählt werden. Neben den Basiskarten gibt es neutrale Karten, welche jede Rasse davon profitieren kann.

Beim Beginn des Spieles zieht jeder Spieler 4 Karten. Der Spieler der im zweiten Zug dran ist, bekommt eine zusätzliche Karte. Man hat die Möglichkeit, die Karten wieder zurück ins Deck zu legen und gleichviele Karten ziehen.

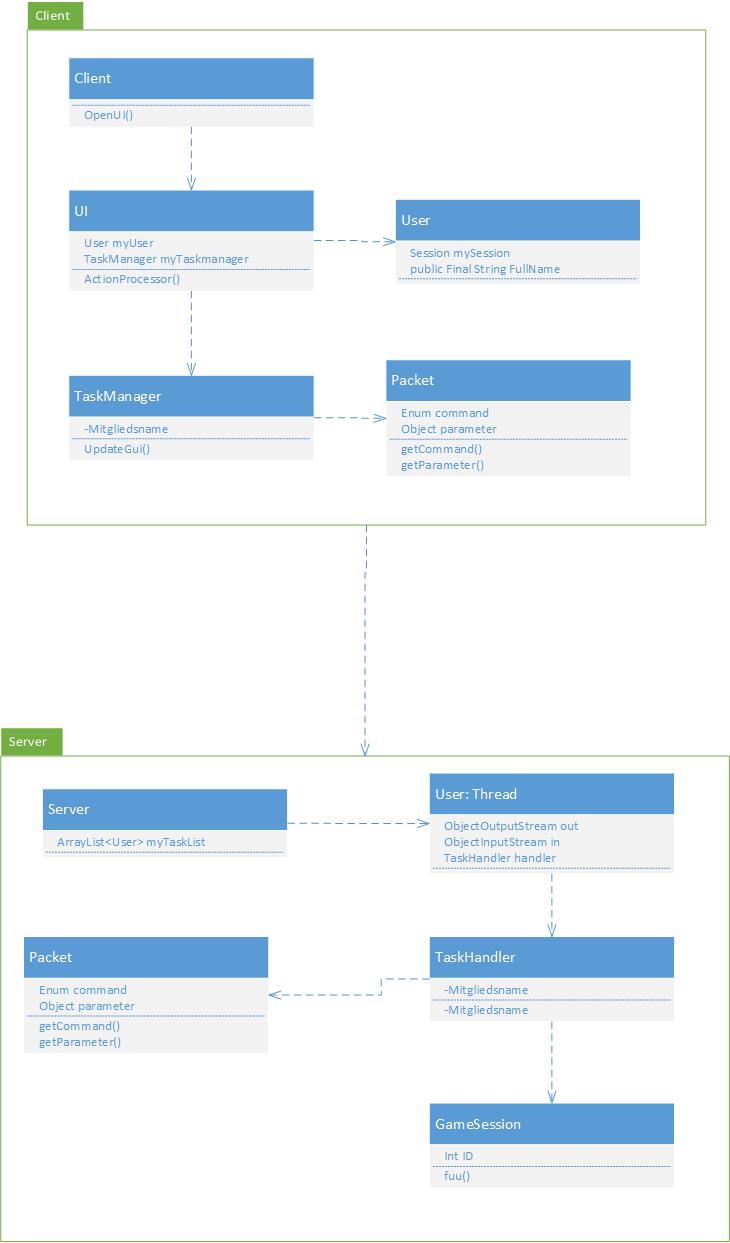
Man startet mit 0 Mana. In jedem Zug hat man 1 Mana mehr zur Verfügung.  
Das Mana wird dazu verwendet um die Karten zu spielen. Jede Karte hat seine eigenen Manakosten.  
Wenn wir zum Beispiel eine 8 Mana Monsterkarte spielt, wäre das erst im 8. Zug möglich.

Jede Rasse hat auch seine eigene Fähigkeit. Diese kostet 2 Mana und kann in jedem Zug dementsprechend verwendet werden.

Jede Karte hat seine Eigenschaften: Leben, Angriff und Passive / Aktive Fähigkeiten.  
Diese Fähigkeiten unterscheiden sich je nach Fähigkeitstyp.

## 1.2 Klassendiagramm

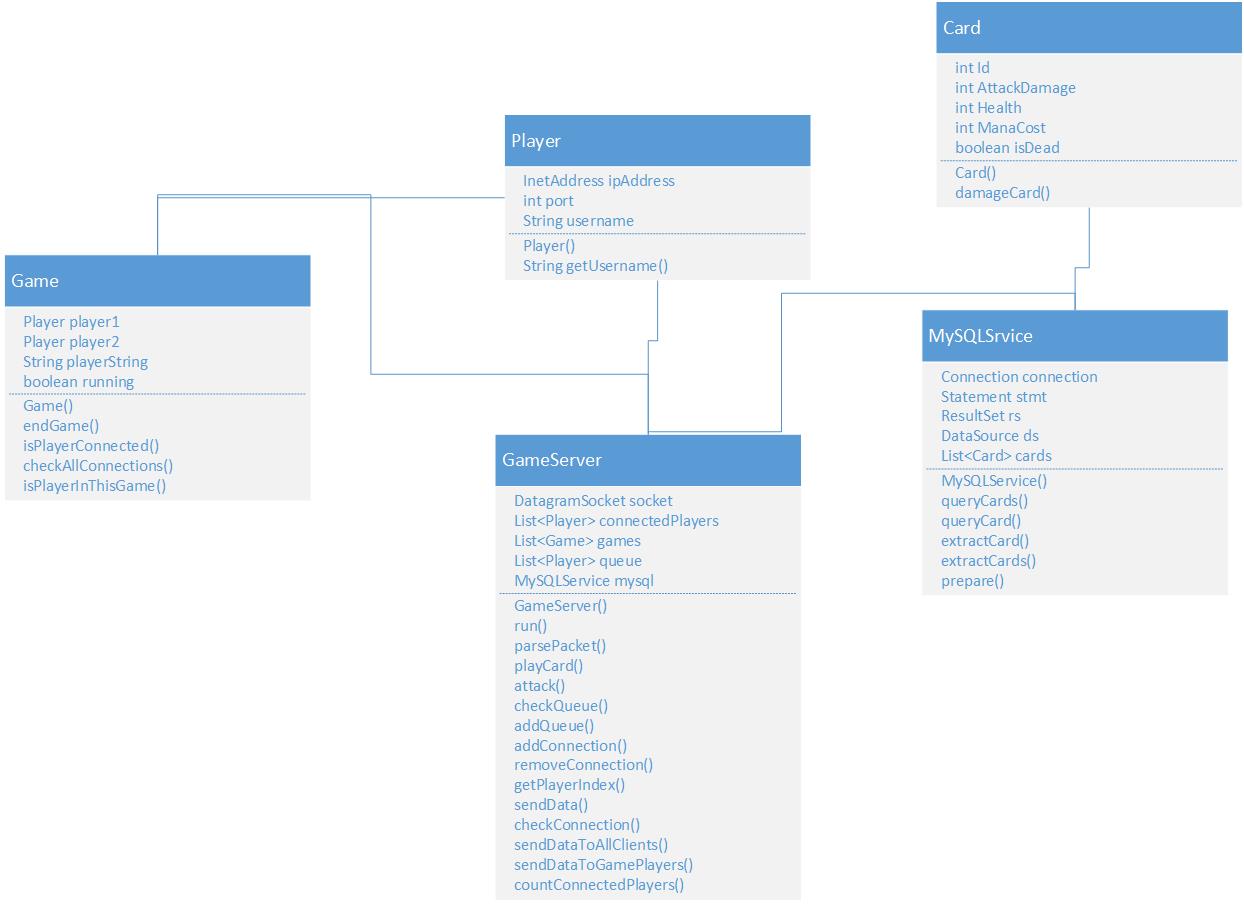
### 1.2.1 Alte Klassendiagramme



### 1.2.2 Neue Klassendiagramme



Server



## 1.3 Use Cases

### 1.3.1 Alte Userspezifische Use Cases

* User kann auf einloggen klicken 🡪 User wird eingeloggt
* User wird bei Inaktivität gekickt
* User kann Privatinformationen bearbeiten
* User kann auf ausloggen klicken –> User wird ausgeloggt
* User kann Decks erstellen / bearbeiten
* User kann Kartenbelohnung auswählen
* User kann leveln

### 1.3.2 Alte Spielspezifische Use Cases

* User kann Karten ziehen
* User kann Karten spielen
* User kann mit den Karten interagieren
* User kann aufgeben
* User kann Spiel starten

### 1.3.3 Neue Use Cases

* User gibt Nick Name ein
* User meldet sich an
* User verbindet mit Server
* User loggt sich aus
* User sucht Spiel
* User findet Gegenspieler
* User startet Spiel
* Spielfenster wird für User angezeigt
* Initiale Handkarten werden für User sauber und komplett angezeigt
* Karten sind für User anklickbar
* User betrachtet Spielfeld

Weitere Use Cases konnten wir leider nicht vollenden da uns die Zeit nicht gereicht hat und wir am Schluss einen merkwürdigen Fehler hatten mit den Libraries der uns sehr viel Zeit gekostet hat.

## 1.4 Unit Tests

Unit Tests haben wir keine eingebaut, da wir an keinem Ort es für sinnvoll gehalten haben diese einzubauen. Im Gegenteil: Wir hätten unsere Architektur weiterhin anpassen müssen, was mehr Aufwand bedeutet hätte. Des Weiteren haben wir eine Server-Client Logik, die die Anwendung von Unit-Tests nur erschwert. Wir wissen jedoch, wie man Unit-Tests verwendet und hatten diese auch schon mehrfach im Einsatz.

## 1.5 Testing

|  |  |  |
| --- | --- | --- |
| Testfall | Erwartetes Resultat | Resultat |
| User gibt Nick Name ein | User kann Text eingeben und es „zwischenspeichern“ | Funktioniert |
| User meldet sich an | User kann sich mit Nick Name anmelden | Funktioniert |
| User verbindet mit Server | User kann sich mit Server verbinden | Funktioniert |
| User loggt sich aus | User kann sich ausloggen | Funktioniert |
| User sucht spiel | User kann Spiel suchen | Funktioniert |
| User findet Gegenspieler | User kann Gegenspieler finden | Funktioniert |
| User startet Spiel | User kann Spiel starten | Funktioniert |
| Spielfenster wird für User angezeigt | Spielfenster wird für User angezeigt | Funktioniert |
| Initiale Handkarten werden für User sauber und komplett angezeigt | Initiale Handkarten werden für User sauber und komplett angezeigt | Funktioniert |
| User klickt Karten an | Karten sind für User anklickbar | Funktioniert |
| User schaut auf Spielfeld | Spielfelder sind für User sichtbar | Funktioniert |
| User schaut auf Spielfeld | Kompletter Inhalt des Fensters ist für User sichtbar | Funktioniert nicht: Die Seite für Status anzeigt geht aus dem Fenster. Man müsste die Controls kleiner skalieren. |