

# Softwaregrundprojekt

Institut für Softwaretechnik und Programmiersprachen | WiSe 2021/22

Prof. Dr. Thomas Thüm, Florian Ege, Dennis Jehle

## Meilenstein 1: Kontextanalyse und Anforderungsdefinition

### 1 Überblick

#### 1.1 Einleitung

In unserem Projekt geht es darum, ein funktionierendes Spiel gemäß Vorgaben aus einem Lastenheft zu erstellen, welches den Namen *Deserts of Dune* tragen wird. Das Spiel ist für zwei Personen gedacht, wobei mehrere Personen dem Spielgeschehen als Zuschauer folgen können. Hierbei wird alles, also sowohl die gesamte Spiellogik, als auch die graphische Gestaltung von uns selbst entwickelt und implementiert werden.

#### 1.2 Motivation

Das Projekt soll dazu beitragen, uns in dem langfristigen Arbeiten an größeren Projekten mit anderen Personen weiterzubringen und somit unser Arbeiten mit fremdem Code zu verbessern. Außerdem lernen wir dadurch, dass das Spiel auch mit Teilen der Codes anderer Gruppen funktionieren soll, eine universell anwendbare Kompatibilität zu entwickeln. Nicht zu vernachlässigen ist, dass jedes Teammitglied lernt seine Stärken einzusetzen und durch die Stärken anderer mögliche Schwächen verbessern kann. Der Auftraggeber bekommt ein funktionstüchtiges Spiel mit guter Qualität zum Zeitvertreib und kann unsere Fähigkeiten zu programmieren und als Team zu arbeiten besser einschätzen und bewerten.

#### 1.3 Vision

Die fertige Anwendung soll ein Hauptmenü haben, von dem aus der Spieler das Spiel starten kann oder sich eine Übersicht der Häuser mit ihren dazugehörigen Charakteren und deren Statuswerte anzeigen lassen kann. Startet der Spieler das Spiel, so soll ihm eine Auswahl an zwei Spielbaren Häusern zur Verfügung stehen, mit welchen er spielen kann.

Das Spiel selbst soll eine eigene Szenerie haben, die die Dünenlandschaft von Arrakis, mit Flachsand, Dünen, Gesteinsfeldern und Gebirgen darstellen soll. Die Statuswerte der Charaktere auf dem Feld sollen jederzeit für jeden Spieler einsehbar sein und außerdem soll ein Pausenmenü jederzeit im Spiel aufgerufen werden können.

#### 1.4 Projektkontext

An dem Projekt beteiligt sind der Auftraggeber, das arbeitende Team und auch die anderen Teams, da das Spiel auch mit Teilen von Spielen von anderen Teams funktionieren soll. Möglicherweise sind auch außenstehende als Spieletester beteiligt.

Denkbare Weiterentwicklungen für das Spiel sind die Erweiterung auf mehrere Spieler, die gegeneinander spielen

können. Daraus ergeben sich dann weiter taktische Möglichkeiten, wie zum Beispiel Bündnisse zwischen Häusern. Eine weitere denkbare Erweiterung wäre das hinzufügen von neuen spielbaren Charakteren oder Charakter-Typen oder gar neuen Häusern.

## 2 Anforderungsanalyse

### 2.1 Fachwissen

<b>Begriff</b>	<b>Bene Gesserit</b>
<b>Beschreibung</b>	Ist ein schneller und wendiger Charakter-Typ mit hoher Heilungsrate.
<b>Ist-ein</b>	Charakter-Typ
<b>Kann-sein</b>	
<b>Aspekt</b>	
<b>Bemerkung</b>	
<b>Beispiel</b>	HP: 150, Heilungs-HPs: 20, MP: 3, AP: 2, Angriffsschaden: 20, Inventargröße: 5

<b>Begriff</b>	<b>Benutzer-Client</b>
<b>Beschreibung</b>	Ermöglicht es einem menschlichen Benutzer über eine grafischer Oberfläche entweder aktiv, als Spieler an einer Partie teilzunehmen oder als passiver Zuschauer im Zuschauermodus eine Partie zwischen zwei anderen Clients zu verfolgen.
<b>Ist-ein</b>	Mensch
<b>Kann-sein</b>	Spielerclient, Zuschauerclient
<b>Aspekt</b>	-
<b>Bemerkung</b>	Kann sich über das Netzwerk bei einem Server für eine Partie registrieren.
<b>Beispiel</b>	Eine Person registriert sich beim Server als Spieler.

<b>Begriff</b>	<b>Charakter</b>
<b>Beschreibung</b>	Eine Figur aus dem Dune-Universum, welche durch eine Einheit auf dem Raster der Spielfelder (Spielbrett) repräsentiert wird.
<b>Ist-ein</b>	bewegliche Einheit
<b>Kann-sein</b>	Player Charakter (PC), Non-Player Character (NPC)
<b>Aspekt</b>	Name, zugehöriger Spieler oder NPC, Charakter-Typ, Health-Points, Inventar, Position auf dem Spielfeld, etc.
<b>Bemerkung</b>	
<b>Beispiel</b>	Baron Vladimir Harkonnen, gehört zu House Harkonnen,  Character-Typ: Noble HP: 100 Heilungs-HPs: 10 MP: 2 Inventargröße: 5 etc.

<b>Begriff</b>	<b>Charakter-Typ</b>
<b>Beschreibung</b>	Jeder Charakter hat einen Charakter-Typ, der seine Statistiken (HP, MP, AP, ...) bestimmt und ihm ggf. spezielle Aktionen ermöglicht.
<b>Ist-ein</b>	
<b>Kann-sein</b>	Noble, Mentat, Bene Gesserit, Fighter
<b>Aspekt</b>	Die Charaktere haben also unterschiedliche Stärken und Schwächen, in der Art des Schere-Stein-Papier Prinzips.
<b>Bemerkung</b>	
<b>Beispiel</b>	

Begriff	Charakter-Zug
Beschreibung	Züge von Charakter bestehen aus Zugphasen, in denen jeweils entweder ein Bewegungsschritt oder eine Aktion gemacht wird.
Ist-ein	
Kann-sein	Bewegungsschritt, Aktion
Aspekt	Die Abhandlung des Zuges eines Charakters findet in aufeinanderfolgenden Zugphasen statt.
Bemerkung	
Beispiel	Duke Leto ist am Zug. Er hat 2 MP und 2 AP. Zuerst benutzt der Spieler einen MP, um Leto auf ein Feld mit einem Spicekrümel zu bewegen. In der zweiten Zugphase verwendet Leto einen AP, um den Spicekrümel aufzusammeln.

Begriff	Düne
Beschreibung	Hügeliges, höher gelegenes Stück Wüste
Ist-ein	Wüstenfeld
Kann-sein	
Aspekt	Ist betretbar und das Höhenniveau ist hoch.
Bemerkung	
Beispiel	

Begriff	Dünenwanderung
Beschreibung	Dünen wandern ständig umher. Diese Dünenwanderungen werden mittels eines zellulären Automaten berechnet.
Ist-ein	
Kann-sein	
Aspekt	Wüstenfelder mit niedrigem Höhenniveau werden im Automaten als "tot" betrachtet und Wüstenfelder mit hohem Höhenniveau werden als "lebendig" betrachtet.
Bemerkung	
Beispiel	

Begriff	Editor
Beschreibung	Der Editor wird zum Erstellen von Content und Konfigurationen für ein Spiel verwendet.
Ist-ein	-
Kann-sein	-
Aspekt	Szenarios und Partie-Konfigurationen können über eine grafische Oberfläche erzeugt und bearbeitet werden.
Bemerkung	Die mit dem Editor erstellten Ressourcen werden im JSON-Format in Dateien gespeichert.
Beispiel	-

Begriff	Felsplateau
Beschreibung	Flaches, niedrig gelegenes Gelände mit massivem Untergrund.
Ist-ein	Felsenfeld
Kann-sein	
Aspekt	Das Felsenfeld ist betretbar und sein Höhenniveau ist niedrig.
Bemerkung	
Beispiel	

Begriff	Fighter
Beschreibung	Ist ein Charakter-Typ, der im Kampf ordentlich einstecken und austreten kann.
Ist-ein	Charakter-Typ
Kann-sein	
Aspekt	
Bemerkung	
Beispiel	HP: 200, Heilungs-HPs: 20, MP: 2, AP: 2, Angriffsschaden: 40, Inventargröße: 3

<b>Begriff</b>	<b>Flachsand</b>
<b>Beschreibung</b>	Flachsand ist ein flaches, niedrig gelegenes Wüstenfeld.
<b>Ist-ein</b>	Wüstenfeld
<b>Kann-sein</b>	
<b>Aspekt</b>	Flachsand ist betretbar und sein Höhenniveau ist niedrig.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Gebirge</b>
<b>Beschreibung</b>	Das Gebirge ist ein hohes, unpassierbares Felsenfeld.
<b>Ist-ein</b>	Felsenfeld
<b>Kann-sein</b>	
<b>Aspekt</b>	Das Gebirge ist nicht betretbar und sein Höhenniveau ist hoch.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Great-Houses</b>
<b>Beschreibung</b>	Great-Houses sind große Adelshäuser, die miteinander konkurrieren. Jedes dieser Häuser besteht aus einer Reihe von Charakteren, die jeweils einen Charakter-Type haben.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	Es gibt im Spiel sechs Great-Houses.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Great-House-Wahlphase</b>
<b>Beschreibung</b>	Zu Beginn jeder Partie muss der Spieler sich entscheiden, mit welchem Great House er auf Arrakis antreten möchte.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	Der Server nimmt aus der Liste der sechs Great Houses zwei zufällige heraus, die dem Spieler angezeigt werden. Der Spieler wählt eines davon aus. Dies findet nebenläufig für beide Spieler statt, also müssen die Mengen der Great House Wahl disjunkt sein.
<b>Bemerkung</b>	Durch die unterschiedlichen Zusammensetzungen der Charaktere bzw. deren Typen, sollte man das Haus wählen, das dem eigenen Spielstil am nächsten kommt.
<b>Beispiel</b>	

<b>Begriff</b>	<b>KI-Client</b>
<b>Beschreibung</b>	Werden autonom von einer Künstlichen Intelligenz gesteuert und könne als Mitspieler an einer Partie teilnehmen.
<b>Ist-ein</b>	Spieler-Client
<b>Kann-sein</b>	-
<b>Aspekt</b>	Der Spieler-Client behandelt Aktionen eines Spielers und nimmt aktiv als Spieler an einer Partie teil.
<b>Bemerkung</b>	-
<b>Beispiel</b>	-

<b>Begriff</b>	<b>Szenario-Konfiguration</b>
<b>Beschreibung</b>	Eindeutige Beschreibung eines Spielfeldes in JSON-Format.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	So kann man selbst konfigurierte Spielfelder speichern und diese immer wieder verwenden.
<b>Bemerkung</b>	
<b>Beispiel</b>	-

<b>Begriff</b>	<b>Mentat</b>
<b>Beschreibung</b>	Ist jemand, der sehr schlau ist, und deshalb besonders effizient darin, Aktionen auszuführen und Spice zu sammeln. Im Kampf hält dieser Charakter-Typ jedoch nicht sehr viel aus.
<b>Ist-ein</b>	Charakter-Typ
<b>Kann-sein</b>	
<b>Aspekt</b>	
<b>Bemerkung</b>	
<b>Beispiel</b>	HP: 75, Heilungs-HPs: 10, MP: 2, AP: 3, Angriffsschaden: 10, Inventargröße: 10

<b>Begriff</b>	<b>Movement Points (MP)</b>
<b>Beschreibung</b>	Jeder Charakter bekommt zu Beginn seines Zuges Movement Points, mit denen er sich auf ein anderes Feld bewegen kann.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	Für einen Movement Point kann ein Charakter sich auf ein betretbares Nachbarfeld bewegen. Dabei kann er sich im Raster vertikal, horizontal oder diagonal bewegen. Ist das Feld von einem anderen Charakter belegt, so wird dieser weggedrängt und die Plätze werden einfach getauscht.
<b>Bemerkung</b>	Die Movement Points werden für jeden Charakter in der Partie-Konfiguration festgelegt.
<b>Beispiel</b>	Charakter 1 bewegt sich von Feld x auf das Feld y, welches von Charakter 2 besetzt ist. Dann ist Charakter 1 danach auf Feld y und Charakter 2 auf Feld x.

<b>Begriff</b>	<b>Noble</b>
<b>Beschreibung</b>	Ist ein hochwohlgeborener Adliger, der durchschnittlich in all seinen Eigenschaften ist.
<b>Ist-ein</b>	Charakter-Typ
<b>Kann-sein</b>	
<b>Aspekt</b>	
<b>Bemerkung</b>	
<b>Beispiel</b>	HP: 100, Heilungs-HPs: 10, MP: 2, AP: 2, Angriffsschaden: 20, Inventargröße: 5

<b>Begriff</b>	<b>Normale Aktion</b>
<b>Beschreibung</b>	Zu Beginn eines Zuges bekommt jeder Charakter so viel Action Points (AP), wie in der Partie-Konfiguration für seinen Charakter-Typ festgelegt ist. Normale Aktionen stehen allen Charakteren zur Verfügung.
<b>Ist-ein</b>	Charakter-Zug
<b>Kann-sein</b>	Angriff, Spice aufsammeln, Spice übergeben
<b>Aspekt</b>	Normale Aktionen kosten 1 AP.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Partie</b>
<b>Beschreibung</b>	Ein ganzes Spiel von Deserts of Dune wird als eine Partie bezeichnet und endet, wenn einer der beiden Spieler gewonnen hat.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	Zwei Spieler spielen gegeneinander und dabei kann es Zuschauer geben, die das Spiel anschauen.
<b>Bemerkung</b>	
<b>Beispiel</b>	Spieler A spielt gegen einen KI-Client und Zuschauer B beobachtet das Spiel.

<b>Begriff</b>	<b>Partie-Beginn</b>
<b>Beschreibung</b>	Vor der ersten Runde weist der Server jedem der Häuser eine der beiden Städte zu und platziert dann alle Charaktere eines Hauses auf zufälligen freien betretbaren Felder, die mit der jeweiligen Stadt benachbart sind.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	Zu Beginn stehen alle Charaktere eines Hauses um ihre eigene Stadt herum. Sie stehen also auf den acht Nachbarfeldern der Stadt.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Partie-Ende</b>
<b>Beschreibung</b>	Wenn in der Sandwurm-Rundenphase der letzte Charakter, der ein Haus kontrolliert, durch einen Sandwurm-Angriff verschluckt wird, während das andere Haus nach Abhandlung dieser Sandwurm-Rundenphase noch Charaktere hat, gewinnt das andere Haus unmittelbar die Partie. Falls nach $r_{max}$ Runden noch immer kein Gewinner feststeht, so werden alle Felsplateaus und Gebirge zu Wüstenfelder. Zusätzlich werden fortan die Sandwurm-Runden durch Shai-Hulud Runden ersetzt.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	Beenden einer Partie.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Partie-Konfiguration</b>
<b>Beschreibung</b>	Hier sind fast alle Parameter und Werte gespeichert, welche für das Spiel wichtig sind.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	Die Partie-Konfiguration Wird zu Beginn einer Partie vom Server geladen und enthält alle wichtigen Spieleinstellungen.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Rundenphase</b>
<b>Beschreibung</b>	<i>Deserts of Dune</i> läuft in Runden ab, in denen Ereignisse passieren und die einzelnen Charaktere ihre Züge nacheinander machen.
<b>Ist-ein</b>	
<b>Kann-sein</b>	Dünenwanderungs-, Sandsturm-, Sandwurm-, Klon-, Charakterzug-Rundenphase
<b>Aspekt</b>	Die Reihenfolge der Rundenphasen ist wie folgt: 1. Dünenwanderungs-Rundenphase 2. Sandsturm-Rundenphase 3. Sandwurm-Rundenphase 4. Klon-Rundenphase 5. Charakterzug-Rundenphase
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Sandsturm</b>
<b>Beschreibung</b>	Zu Beginn jeder Partie wird vom Server ein zufälliges Zentralfeld gewählt. Auf diesem Zentralfeld und allen Nachbarfeldern, also ein 3x3 Quadrat, tobt ein gefährlicher Sandsturm. Charaktere, die sich in diesem Sturm befinden, können keine Aktionen machen und auch selbst nicht Ziel von Aktionen sein. Einzig die Aktion Atomics-Einsatz kann Charaktere beeinflussen.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	Zu Beginn jeder Runde wird in der Sandsturm-Rundenphase das Zentralfeld des Sturms auf ein zufälliges Nachbarfeld bewegt. Das 3x3 Sturmquadrat bewegt sich also Runde für Runde über das Spielfeld. Außerdem wird jedes Wüstenfeld in diesem 3x3 Bereich zufällig auf ein niedriges oder hohes Höhenniveau gesetzt.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Sandwurm</b>
<b>Beschreibung</b>	Zu Beginn jeder Runde, in der Sandwurm-Rundenphase, falls es bereits einen Sandwurm gibt, läuft dieser n Felder lang auf seine Zielperson zu. Wenn der Sandwurm auf dem Feld eines Charakters ist, wird dieser Charakter verschluckt und der Sandwurm verschwindet ebenfalls.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	Der verschluckte Charakter muss nicht der Zielcharakter sein, er kann auch unglücklich im Weg stehen. Der Sandwurm kann sich nur über Wüstenfelder bewegen.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Server</b>
<b>Beschreibung</b>	Zentrale Komponente, die eine Partie von <i>Deserts of Dune</i> verwalten und abwickeln kann.
<b>Ist-ein</b>	-
<b>Kann-sein</b>	-
<b>Aspekt</b>	Beinhaltet die Anwendungslogik zur Verwaltung zur Verwaltung von Clients und Partien, sowie der Umsetzung der Spielregeln von <i>Deserts of Dune</i> .
<b>Bemerkung</b>	-
<b>Beispiel</b>	-

<b>Begriff</b>	<b>Siegbedingungen</b>
<b>Beschreibung</b>	Es werden nacheinander Sieg-Metriken in Reihenfolge betrachtet, wie z.B. welches Haus den größten Spice-Vorrat hat.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	Am Ende einer Partie wird überprüft, welcher der beiden Spieler gewonnen hat.
<b>Bemerkung</b>	Wenn nach einer Metrik Gleichstand herrscht, wird die jeweils nächste Metrik als Tie-Breaker herangezogen.
<b>Beispiel</b>	Alle Charaktere sind Tod. Da Spieler B über mehr Spice als Spieler A verfügt, hat Spieler B gewonnen.

<b>Begriff</b>	<b>Spezielle Aktion</b>
<b>Beschreibung</b>	Zu Beginn eines Zuges bekommt jeder Charakter so viel Action Points (AP), wie in der Partie-Konfiguration für seinen Charakter-Typ festgelegt ist. Spezielle Aktionen sind nur für bestimmte Charakter-Typen verfügbar.
<b>Ist-ein</b>	Charakter-Zug
<b>Kann-sein</b>	Kanly, Family Atomics, Spice Hoarding, Voice, Sword Spin
<b>Aspekt</b>	Spezielle Aktionen kosten so viele APs, wie der Charakter pro Zug zur Verfügung hat.
<b>Bemerkung</b>	Kanly nur für Noble Family Atomics nur für Noble Spice Hoarding nur für Mentat Voice nur für Bene Gesserit Sword Spin nur für Fighter
<b>Beispiel</b>	

<b>Begriff</b>	<b>Spice</b>
<b>Beschreibung</b>	Zu Beginn jeder Runde, falls die Anzahl der auf der Karte liegenden Spicekrümel kleiner als der Spice-Schwellenwert $s$ ist, findet ein Spice-Blow statt, bei dem ein zufälliges Wüstenfeld gewählt wird und alle benachbarten Wüstenfelder werden zufällig auf Flachsand oder Düne gesetzt, und es werden Spicekrümel auf den Felder verteilt.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	Bei einem Spice-Blow werden 3-6 Spicekrümel irgendwo in der Wüste auf Feldern verteilt.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Spiceablieferung</b>
<b>Beschreibung</b>	Tritt ein Charakter auf ein Nachbarfeld der eigenen Stadt, so wird direkt nach dieser Zugphase alles Spice aus seinem Inventar in der Stadt abgeladen und zum Spice-Vorrat des Hauses hinzugezählt.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	Alle Charaktere versuchen so viel Spice wie möglich in der Wüste zu sammeln und dann zu ihrer Stadt zurückbringen, da dies die Siegmetrik ist.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Spielbrett</b>
<b>Beschreibung</b>	Ein Spielbrett besteht aus $x$ mal $y$ schachbrettartigen Feldern.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	Besteht aus verschiedenen Arten von Feldern, die von Charakteren betretbar sein können oder nicht.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Spieler</b>
<b>Beschreibung</b>	Der Spieler nimmt aktiv am Spielgeschehen teil und führt Aktionen aus.
<b>Ist-ein</b>	Teilnehmer
<b>Kann-sein</b>	Benutzer-Client, KI-Client
<b>Aspekt</b>	Nimmt aktiv am Spiel teil durch ausführen von Aktionen.
<b>Bemerkung</b>	
<b>Beispiel</b>	Der Spieler A bewegt den Charakter X auf das Wüstenfeld nebenan.



<b>Begriff</b>	<b>Spielstand</b>
<b>Beschreibung</b>	Stellt den aktuellen Spielstand dar mit Spiceguthaben, Position der Charaktere, usw.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	Aktueller Spielstand wird gespeichert, dass es bei einer Pause genau gleich wie davor weitergeht.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Stadt</b>
<b>Beschreibung</b>	Ist ein Spezialfeld. Auf Arrakis gibt es zwei größere Städte, Arrakeen und Carthag. Beiden Spielern gehört eine davon.
<b>Ist-ein</b>	Spezialfeld
<b>Kann-sein</b>	
<b>Aspekt</b>	Ist nicht betretbar und das Höhenniveau ist hoch.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Wüstenlautheit</b>
<b>Beschreibung</b>	Damit man die gefährlichen Sandwürmer nicht anlockt, muss man sich in der Wüste sehr langsam fortbewegen, da die Sandwürmer auf Geräusche reagieren. Am Anfang seines Zuges gilt jeder Charakter als "leise" für diese Runde. Wenn der Charakter während seines Zuges mehr als einmal ein Wüstenfeld betritt oder einen Spicekrümel von einem Wüstenfeld ins Inventar aufnimmt, wird er für diese Runde als "laut" markiert.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	Um beim Laufen leise zu bleiben, darf also pro Zug höchstens einmal ein Wüstenfeld betreten werden.
<b>Bemerkung</b>	
<b>Beispiel</b>	Charakter A wandert über mehrere Wüstenfelder. Aufgrund des zunehmenden Sandes in seinen Schuhen lockt er feindliche Sandwürmer an.

<b>Begriff</b>	<b>Zuschauer</b>
<b>Beschreibung</b>	Der menschliche Benutzer kann sich als Zuschauer im Zuschauermodus und das Spiel anschauen, jedoch nicht ins Spielgeschehen eingreifen.
<b>Ist-ein</b>	Mensch
<b>Kann-sein</b>	
<b>Aspekt</b>	Nimmt passiv am Spielgeschehen teil.
<b>Bemerkung</b>	Man kann von Anfang an zuschauen oder man kann auch während eine Partie schon läuft noch dazukommen.
<b>Beispiel</b>	Dennis möchte aufgrund seiner hohen Verlustrate bei einem reinen KI-Spiel zuschauen, um von diesen zu lernen.

<b>Begriff</b>	<b>Desert of Dune</b>
<b>Beschreibung</b>	Ein rundenbasiertes Taktik-Spiel für zwei Spieler es spielt auf dem Wüstenplaneten Arakis.
<b>Ist-ein</b>	Taktik-Spiel
<b>Kann-sein</b>	
<b>Aspekt</b>	Ist das Grundkonzept des Spiels.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Arrakis</b>
<b>Beschreibung</b>	Der Planet auf dem das Spiel stattfindet.
<b>Ist-ein</b>	ein Planet
<b>Kann-sein</b>	
<b>Aspekt</b>	Ist die Szenerie des Spiels Deserts of Dune.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Health Points</b>
<b>Beschreibung</b>	Health Points sind die Eigenschaft wie viele Lebenspunkte ein Charakter hat.
<b>Ist-ein</b>	Eine Eigenschaft eines Charakters.
<b>Kann-sein</b>	
<b>Aspekt</b>	ist ein Parameter eines Charakters.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Inventar</b>
<b>Beschreibung</b>	Definiert wie viele Spice krümel ein Charakter maximal tragen kann und wie viele er aktuell trägt.
<b>Ist-ein</b>	Eine Eigenschaft eines Charakters.
<b>Kann-sein</b>	
<b>Aspekt</b>	ist ein Parameter eines Charakters.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Heilungs-Health Points</b>
<b>Beschreibung</b>	Definiert um wie viele Health Points sich ein Charakter pro Spielrunde Heilen kann, wenn er keine Aktion ausführt.
<b>Ist-ein</b>	Eine Eigenschaft eines Charakters.
<b>Kann-sein</b>	
<b>Aspekt</b>	ist ein Parameter eines Charakters.
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Spice-schwellwert</b>
<b>Beschreibung</b>	Der Spice-Schwellwert definiert ist eine feste Anzahl an minimalen Spice Einheiten auf dem Spielfeld. Falls dieser Schwellwert unterschritten wird wird der Spice Blow ausgelöst.
<b>Ist-ein</b>	Ist eine Eigenschaft der Karte.
<b>Kann-sein</b>	
<b>Aspekt</b>	
<b>Bemerkung</b>	
<b>Beispiel</b>	Wenn der spezifische Spiceschwellwert unterschritten wird wird ein Spiceblow ausgelöst.

<b>Begriff</b>	<b>Nachbarfeld</b>
<b>Beschreibung</b>	Ein Nachbarfeld ist eines der acht Felder, welche ein spezifisches Feld umgeben.
<b>Ist-ein</b>	Feld auf dem Spielbrett.
<b>Kann-sein</b>	
<b>Aspekt</b>	
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Zufall</b>
<b>Beschreibung</b>	Wenn etwas zufällig gewählt werden soll, ist damit immer eine gleich verteilte zufällige Auswahl unter den entsprechenden Möglichkeiten gemeint.
<b>Ist-ein</b>	
<b>Kann-sein</b>	
<b>Aspekt</b>	
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Charakteren, die ein Haus kontrolliert</b>
<b>Beschreibung</b>	Das sind die Charaktere welche nicht von einem Sandwurm verschluckt worden.
<b>Ist-ein</b>	Charakter
<b>Kann-sein</b>	verschluckt oder nicht verschluckt.
<b>Aspekt</b>	
<b>Bemerkung</b>	
<b>Beispiel</b>	

<b>Begriff</b>	<b>Spieler</b>
<b>Beschreibung</b>	Ein Spieler ist ein Client, welcher aktiv am Spiel teilnimmt.
<b>Ist-ein</b>	Client
<b>Kann-sein</b>	User-Client oder KI-Client
<b>Aspekt</b>	
<b>Bemerkung</b>	
<b>Beispiel</b>	

Diese Tabelle enthält Abkürzungen und domänenspezifische Begriffe, die im Dokument verwendet werden.

## 2.2 Anwendungskontext

### 2.2.1 Akteure und Rollen

<b>AKTEUR:</b>	Server
<b>BESCHREIBUNG:</b>	Zentraler Rechner, der allen Clients das Spiel zur Verfügung stellt
<b>ROLLE:</b>	Der Server ist die zentrale Vermittlungsinstanz zwischen den Clients. Das bedeutet, dass der Server für den Informationsaustausch zwischen den verschiedenen Clients zuständig ist. Dabei ist der Server auch für die Verwaltung der Clients zuständig. Weiterhin verwaltet er auf eine Partie des Spiels <i>Deserts of Dune</i> und beinhaltet die Spielregeln und -logik. Das heißt, der Server simuliert das Spiel, verarbeitet die Befehle von den Clients und sendet ihnen den aktuellen Spielstand. Außerdem kann der Server mithilfe einer Partie- und Szenariokonfiguration eine Partie erstellen.

<b>AKTEUR:</b>	Benutzer-Client
<b>BESCHREIBUNG:</b>	Rechner vom Benutzer, auf dem die Endanwendung läuft
<b>ROLLE:</b>	Der Benutzer-Client dient dem menschlichen Benutzer als grafische Benutzeroberfläche, um mit dem Spiel interagieren oder beobachten zu können (z. B. aktuellen Spielstand ansehen, Zug ausführen, Spiel starten, ...). Der Client muss sich mit einem Server verbinden. Dann kann der Client dem Server durch das Senden von Nachrichten Befehle geben <sup>1</sup> oder durch den Empfang und die Verarbeitung von Nachrichten vom Server den aktuellen Spielstand anzeigen.

<sup>1</sup>Das gilt nur für Spieler und nicht für Zuschauer

AKTEUR:	menschlicher Benutzer
BESCHREIBUNG:	Mensch, der aktiv oder passiv am Spiel teilnimmt
ROLLE:	Der menschliche Benutzer ist entweder ein passiver Zuschauer oder ein aktiver Spieler. Er kann entweder aktiv oder passiv am Spiel und damit an einer Partie teilnehmen. Dafür startet er den Benutzer-Client und kann über dessen grafische Benutzeroberfläche das Spielgeschehen verfolgen und gegebenenfalls mit der Partie interagieren.

AKTEUR:	Spieler
BESCHREIBUNG:	Menschlicher Benutzer bzw. Teilnehmer mit direktem Einfluss auf das Spielgeschehen
ROLLE:	Ein Spieler ist ein aktiver Teilnehmer an einer Partie. Ein Spieler spielt das Spiel <i>Deserts of Dune</i> gegen eine andere Person und kann dabei von Zuschauern beobachtet werden. Der Spieler kann gemäß der Regeln Aktionen ausführen und Befehle an den Server senden, mit dem Ziel, das Spiel zu gewinnen. Ein Spieler hat auch die Möglichkeit über den Editor das Spiel zu konfigurieren. Außerdem sieht der Spieler den aktuellen Spielstand über den Benutzer-Client.

AKTEUR:	Zuschauer
BESCHREIBUNG:	Menschlicher Benutzer ohne Einfluss auf die Partie
ROLLE:	Ein Zuschauer kann passiv eine Partie, das heißt das Spielgeschehen, mitverfolgen. Dafür besitzt der Zuschauer auch einen (eingeschränkten) Benutzer-Client, über den er über die grafische Benutzeroberfläche die Partie angezeigt bekommt. Der Zuschauer hat im Gegensatz zu dem Spieler keine Möglichkeit, aktiv in das Spielgeschehen einzugreifen. Das bedeutet dieser Benutzer kann nicht mit dem Server kommunizieren.

AKTEUR:	KI-Client
BESCHREIBUNG:	Teilnehmer, der von einer künstlichen Intelligenz gesteuert wird
ROLLE:	Ein KI-Client kann als Mitspieler an einer Partie teilnehmen. Der KI-Client kann dabei das Spiel <i>Deserts of Dune</i> aktiv gegen einen anderen Teilnehmer spielen. Dafür muss der KI-Client eine Verbindung beim Server aufbauen und kann dann mithilfe von Nachrichten mit dem Server kommunizieren. Dabei hat er gegenüber dem Spieler nur eingeschränkte Rechte, wie zum Beispiel, dass er keine Partie pausieren kann. Der KI-Client trifft seine Entscheidungen nach vorprogrammierten oder angelernten Regeln, die statisch oder dynamisch an das Spielgeschehen angepasst werden können. Damit versucht der KI-Client das menschliche Verhalten oder eine Spielstrategie zu simulieren.

AKTEUR:	Teilnehmer
BESCHREIBUNG:	Spieler oder KI-Client
ROLLE:	Ein Teilnehmer ist eine aktive Person oder Software, die mit dem Server kommunizieren kann und dadurch aktiv in das Spielgeschehen eingreifen kann. Dafür muss der Teilnehmer eine Verbindung beim Server aufbauen und kann dann mithilfe von Nachrichten mit dem Server kommunizieren. Ein Teilnehmer kann entweder ein Spieler oder ein KI-Client sein.

AKTEUR:	Editor
BESCHREIBUNG:	Grafische Benutzeroberfläche zur Erstellung von Konfigurationen
ROLLE:	Der Editor ist ein Teil des Benutzer-Clients und ist damit eine grafische Benutzeroberfläche. Der Editor soll dem Benutzer die Möglichkeit geben, Inhalt und Konfigurationen, wie zum Beispiel die Szenarios oder die Partiekonfiguration zu erstellen.

AKTEUR:	Auftraggeber
BESCHREIBUNG:	Servicegruppe Information am Institut für Softwaretechnik und Programmiersprachen
ROLLE:	Der Auftraggeber ist die Person, die Erstellung des Produkts, das heißt die Umsetzung des Spiels <i>Deserts of Dune</i> , beauftragt hat. Der Auftraggeber stellt in dem Lastenheft Anforderung an die Entwickler und entscheidet am Ende, ob das erstellte Produkt den Anforderungen genügt. Außerdem wird sie durch den Tutor vertreten.

AKTEUR:	Vertreter des Auftraggebers
BESCHREIBUNG:	Zugewiesener Tutor: Tim Wibiral
ROLLE:	Der Vertretet des Auftragsgebers, das heißt der für das Projekt zugewiesene Tutor vertritt den Auftraggeber. Das heißt, er überwacht den Entwicklungsprozess, übernimmt während der Implementierungsphase die Rolle eines <i>Product Owners</i> . Zusätzlich veranstaltet der Tutor wöchentliche Treffen, die Tutorien oder <i>Sprint Meetings</i> und steht für Rückfragen zur Verfügung oder gibt Feedback.

AKTEUR:	Entwickler
BESCHREIBUNG:	Team 008 des Softwaregrundprojekts
ROLLE:	Das Produkt, das heißt das gesamte System, wird von den Entwicklern geplant, modelliert und anschließend implementiert, sowie dokumentiert. Dafür werden basierend auf dem Lastenheft Anforderungen, Definitionen, Analysen und Modelle erstellt, die danach implementiert werden. Am Ende sollen die Entwickler ein funktionierendes Spiel mit allen Komponenten fertigstellen.

AKTEUR:	Standardisierungskomitee
BESCHREIBUNG:	teamübergreifendes Komitee zur Standardisierung von Protokollen
ROLLE:	Das zu entwickelnde Spiel ist eine verteilte Anwendung. Da nicht alle Komponenten von einem Team entwickelt werden und die Komponenten über Protokolle kommunizieren, muss für die Kommunikation ein einheitliches Protokoll existieren. Damit dieses Protokoll teamübergreifend identisch ist und damit von allen Teams verwendet werden kann, muss es von einem Standardisierungskomitee definiert werden. Außerdem muss jedes Team einen Vertreter in dem Komitee haben.

## 2.3 Funktionale Anforderungen

Dieser Abschnitt enthält alle Anforderungen, die die grundlegenden Aktionen des Softwaresystems spezifizieren. Dabei gilt für Priorisierung, dass eine Anforderung eine Priorität von 1 (sehr gering) bis 5 (essenziell wichtig) bekommen kann.

### 2.3.1 Komponenten und Architektur

<b>ID</b>	<b>FA1</b>
TITEL:	Architektur
BESCHREIBUNG:	Das Spiel „Deserts of Dune“ ist eine verteilte Anwendung mit der Client-Server-Architektur.
BEGRÜNDUNG	Damit hat man das Spiel in Instanzen aufteilen, die das Spiel bereit stellen und organisieren und Instanzen, die es spielen. Außerdem können somit zwei Spieler auf unterschiedlichen Systemen gegeneinander spielen
ABHÄNGIGKEITEN	FA2 → p. 13
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12

<b>ID</b>	<b>FA2</b>
TITEL:	Komponenten des Spiels
BESCHREIBUNG:	Das gesamte System besteht aus vier Komponenten: Server, Benutzer-Client, KI-Client und Editor
BEGRÜNDUNG	Die Spiellogik soll in diese vier Komponenten aufgeteilt werden als Vorgabe vom Auftraggeber
ABHÄNGIGKEITEN	FA3→ p. 14, FA4→ p. 14, FA5→ p. 14, FA8→ p. 14, FA9→ p. 15
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12

<b>ID</b>	<b>FA3</b>
TITEL:	Kombinierbarkeit von Komponenten
BESCHREIBUNG:	Die Komponenten: Server, Benutzer-Client, KI-Client und Editor sollen beliebig über Teamgrenzen hinweg kombinierbar sein.
BEGRÜNDUNG	Dies ist nötig, um in einer späteren Phase des Projektes Komponenten auszutauschen.
ABHÄNGIGKEITEN	FA4→ p. 14, FA5→ p. 14, FA8→ p. 14, FA2→ p. 13
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Entwickler→ p. 13

<b>ID</b>	<b>FA4</b>
TITEL:	Server
BESCHREIBUNG:	Der Server hält die Anwendungslogik zur Verwaltung von Clients, Partien und zur Umsetzung der Spielregeln.
BEGRÜNDUNG	Der Server soll die gesamte Partie managen und abwickeln, weil diese Aufgabe zentral erledigt werden soll.
ABHÄNGIGKEITEN	
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, menschlicher Benutzer→ p. 11, Teilnehmer→ p. 12

<b>ID</b>	<b>FA5</b>
TITEL:	Benutzer-Client
BESCHREIBUNG:	Der Benutzer-Client ermöglicht es einem einzelnen menschlichen Benutzer, als Spieler oder als Zuschauer an einer Partie teilzuhaben.
BEGRÜNDUNG	Der Benutzer braucht eine Anwendung, um eine Partie zu verfolgen oder daran teilzunehmen
ABHÄNGIGKEITEN	FA101→ p. 36, FA102→ p. 36
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, menschlicher Benutzer→ p. 11

<b>ID</b>	<b>FA6</b>
TITEL:	Teilnehmeranzahl an Partie
BESCHREIBUNG:	Es nehmen genau zwei gegeneinander spielende Spieler an einer Partie teil.
BEGRÜNDUNG	Das Spiel ist darauf ausgerichtet, dass immer zwei Spieler gegeneinander spielen.
ABHÄNGIGKEITEN	
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12

<b>ID</b>	<b>FA7</b>
TITEL:	Beobachten von Partien
BESCHREIBUNG:	Das Spiel kann von beliebig vielen Benutzer-Clients beobachtet werden.
BEGRÜNDUNG	Es soll möglich sein ein beliebiges Spiel zu beobachten.
ABHÄNGIGKEITEN	FA102→ p. 36
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, menschlicher Benutzer→ p. 11

<b>ID</b>	<b>FA8</b>
TITEL:	KI-Client
BESCHREIBUNG:	Der KI-Client wird autonom durch eine Künstliche Intelligenz gesteuert und kann als Mitspieler an einer Partie teilnehmen.
BEGRÜNDUNG	Um auch einzelnen Benutzer ein spannendes Multiplayer-Erlebnis bieten zu können muss es einen KI-Client geben, welcher gegen diese Benutzer antreten kann.
ABHÄNGIGKEITEN	
PRIORITÄT	5
AKTEUR	Auftraggeber <sup>→ p. 12</sup> , menschlicher Benutzer <sup>→ p. 11</sup>
<b>ID</b>	<b>FA9</b>
TITEL:	Editor
BESCHREIBUNG:	Mit dem Editor kann der Spieler das Spiel konfigurieren
BEGRÜNDUNG	Dem Nutzer muss es möglich sein, die Konfiguration für ein Spiel anzupassen.
ABHÄNGIGKEITEN	FA125 <sup>→ p. 43</sup> , FA127 <sup>→ p. 43</sup> , FA122 <sup>→ p. 42</sup>
PRIORITÄT	5
AKTEUR	Auftraggeber <sup>→ p. 12</sup> , menschlicher Benutzer <sup>→ p. 11</sup>

### 2.3.2 Netzwerkkommunikation und Nachrichtenprotokoll

<b>ID</b>	<b>FA10</b>
TITEL:	Netzwerkprotokoll
BESCHREIBUNG:	Die Komponenten verwenden für die Kommunikation untereinander ein Netzwerkprotokoll. Das Netzwerkprotokoll ist ein WebSocket-Protokoll.
BEGRÜNDUNG	Die Komponenten müssen nach einem Standard kommunizieren und das WebSocket-Protokoll ist ein etabliertes, gut funktionierendes Protokoll
ABHÄNGIGKEITEN	FA11 <sup>→ p. 15</sup>
PRIORITÄT	5
AKTEUR	Auftraggeber <sup>→ p. 12</sup> , Entwickler <sup>→ p. 13</sup>
<b>ID</b>	<b>FA11</b>
TITEL:	Kodierung der Netzwerknachrichten
BESCHREIBUNG:	Die über die WebSocket-Verbindung ausgetauschten Textstrings sind im UTF-8 Format kodiert
BEGRÜNDUNG	Die Textstrings müssen aufgrund des WebSocket-Protokoll kodiert werden und UTF-8 ist ein Standardformat
ABHÄNGIGKEITEN	FA10 <sup>→ p. 15</sup>
PRIORITÄT	5
AKTEUR	Auftraggeber <sup>→ p. 12</sup> , Entwickler <sup>→ p. 13</sup>
<b>ID</b>	<b>FA12</b>
TITEL:	Nachrichtenformat des Spielprotokolls
BESCHREIBUNG:	Die über die WebSocket-Verbindung ausgetauschten Textstrings enthalten die Nachrichten des Spielprotokolls. Diese Nachrichten sind in dem Format JSON formatiert.
BEGRÜNDUNG	Die Textstrings müssen aufgrund des WebSocket-Protokoll kodiert werden und UTF-8 ist ein Standardformat.
ABHÄNGIGKEITEN	FA10 <sup>→ p. 15</sup> , FA13 <sup>→ p. 15</sup>
PRIORITÄT	5
AKTEUR	Auftraggeber <sup>→ p. 12</sup> , Entwickler <sup>→ p. 13</sup>

ID	FA13
TITEL:	Spielprotokoll
BESCHREIBUNG:	Das Spielprotokoll wird durch das Standardisierungskomitee definiert.
BEGRÜNDUNG	Es muss ein fest definiertes Spielprotokoll verwendet werden, um eine Standardisierung zu erhalten. Dieses Protokoll muss teamübergreifend erstellt werden, da es mehrere Teams verwenden.
ABHÄNGIGKEITEN	FA10 <sup>→ p. 15</sup> , FA12 <sup>→ p. 15</sup>
PRIORITÄT	5
AKTEUR	Auftraggeber <sup>→ p. 12</sup> , Entwickler <sup>→ p. 13</sup>

### 2.3.3 Szenarios

ID	FA14
TITEL:	Sichtbarkeit des Spielfeldes
BESCHREIBUNG:	Alle Spieler können zu jedem Zeitpunkt das gesamte Spielfeld sehen.
BEGRÜNDUNG	Die Spieler sollen zu jedem Zeitpunkt des Spiels sämtliche Informationen über den Status des Spiels haben, weil das Spiel ein Spiel mit vollständiger Information sein soll.
ABHÄNGIGKEITEN	FA16 <sup>→ p. 16</sup>
PRIORITÄT	5
AKTEUR	Auftraggeber <sup>→ p. 12</sup> , menschlicher Benutzer <sup>→ p. 11</sup> , Teilnehmer <sup>→ p. 12</sup>

ID	FA15
TITEL:	Sichtbarkeit Zustand und Charakterinventar
BESCHREIBUNG:	Alle Spieler können zu jedem Zeitpunkt alle Charaktere und deren Inventar sehen.
BEGRÜNDUNG	Die Spieler sollen zu jedem Zeitpunkt des Spiels sämtliche Informationen über den Status des Spiels haben, weil das Spiel ein Spiel mit vollständiger Information sein soll.
ABHÄNGIGKEITEN	FA36 <sup>→ p. 21</sup>
PRIORITÄT	5
AKTEUR	Auftraggeber <sup>→ p. 12</sup> , menschlicher Benutzer <sup>→ p. 11</sup> , Teilnehmer <sup>→ p. 12</sup>

ID	FA16
TITEL:	Spielbrett
BESCHREIBUNG:	Das Spielbrett ist ein rechteckiges kartesisches Raster aus $x * y$ Feldern. Dieses Spielbrett inklusive aller Felder und Charakter repräsentiert das Szenario.
BEGRÜNDUNG	Alle Spielbretter sollen rechteckig sein, weil das die Berechnung von Zügen vereinfacht, und die einzelnen Felder sollen klar unterscheidbar sein.
ABHÄNGIGKEITEN	FA18 <sup>→ p. 16</sup> , FA17 <sup>→ p. 16</sup>
PRIORITÄT	5
AKTEUR	Auftraggeber <sup>→ p. 12</sup> , menschlicher Benutzer <sup>→ p. 11</sup> , Teilnehmer <sup>→ p. 12</sup>

ID	FA17
TITEL:	Erstellung Spielbrett
BESCHREIBUNG:	Das Spielbrett wird als Szenario im Editor erstellt und kann dann vom Server geladen werden
BEGRÜNDUNG	Das Spielbrett muss irgendwie erstellt oder definiert werden können.
ABHÄNGIGKEITEN	FA16 <sup>→ p. 16</sup> , FA127 <sup>→ p. 43</sup> , FA122 <sup>→ p. 42</sup>
PRIORITÄT	5
AKTEUR	Auftraggeber <sup>→ p. 12</sup> , menschlicher Benutzer <sup>→ p. 11</sup> , Teilnehmer <sup>→ p. 12</sup>



ID	FA18
TITEL:	Feld
BESCHREIBUNG:	Ein Feld ist ein Teil auf dem Spielbrett und wird durch eine Koordinate positioniert.
BEGRÜNDUNG	Felder auf dem Spielbrett sollen genau einer Koordinate zugeordnet werden.
ABHÄNGIGKEITEN	FA16 → p. 16
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

ID	FA19
TITEL:	Eigenschaften von Feldern
BESCHREIBUNG:	Ein Feld hat die Eigenschaft Feldart, Betretbarkeit und ein Höhenniveau
BEGRÜNDUNG	Felder müssen spezifische Eigenschaften haben, um den Spielverlauf durch diese Eigenschaften zu beeinflussen.
ABHÄNGIGKEITEN	FA18 → p. 16, FA22 → p. 17, FA20 → p. 17, FA21 → p. 17
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

ID	FA20
TITEL:	Feldeigenschaft Betretbarkeit
BESCHREIBUNG:	Die Eigenschaft Betretbarkeit besagt, ob ein Charakter ein Feld betreten kann oder nicht. Wenn dieses betretbare Feld durch ein Objekt blockiert ist, dann heißt es <i>besetzt</i> , ansonsten heißt es <i>frei</i>
BEGRÜNDUNG	Es muss klar definiert sein, ob ein spezifisches Feld durch einen Charakter betreten werden kann oder nicht und ob es frei ist.
ABHÄNGIGKEITEN	FA19 → p. 17
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

ID	FA21
TITEL:	Feldeigenschaft Höhenniveau
BESCHREIBUNG:	Die Eigenschaft Höhenniveau gibt an, ob ein Feld <i>hoch</i> oder <i>niedrig</i> ist.
BEGRÜNDUNG	Es muss klar definiert sein, ob ein spezifisches Feld hoch oder niedrig ist, was bei der Bewegung und bei den Aktionen von Charakteren eine Rolle spielt
ABHÄNGIGKEITEN	FA19 → p. 17
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

ID	FA22
TITEL:	Feldarten
BESCHREIBUNG:	Es gibt die verschiedenen Feldarten: <i>Stadt</i> , <i>Flachsand</i> , <i>Düne</i> , <i>Felsplateau</i> , <i>Gebirge</i> . Für jede Feldart sind die Eigenschaften <i>Betretbarkeit</i> und <i>Höhenniveau</i> definiert.
BEGRÜNDUNG	Jedes Feld muss eine definierte Feldart und damit einhergehende Parameter haben.
ABHÄNGIGKEITEN	FA20 → p. 17, FA21 → p. 17
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

<b>ID</b>	<b>FA23</b>
TITEL:	Feldart Stadt
BESCHREIBUNG:	Die Stadt ist ein Spezialfeld. Auf dem Spielbrett gibt es zwei große Städte, Arrakeen und Carthag. Beiden Spielern gehört jeweils eine davon. Felder mit der Feldart Stadt sind nicht betretbar und haben ein hohes Höhenniveau.
BEGRÜNDUNG	Jede Feldart hat besondere Eigenschaften und die Parameter müssen wohldefiniert sein.
ABHÄNGIGKEITEN	FA22→ p. 17, FA20→ p. 17, FA21→ p. 17
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12, menschlicher Benutzer→ p. 11
<b>ID</b>	<b>FA24</b>
TITEL:	Feldart Flachsand
BESCHREIBUNG:	Flachsand ist ein Wüstenfeld, welches flach und niedrig gelegen ist. Felder der Feldart Flachsand sind betretbar und haben ein niedriges Höhenniveau.
BEGRÜNDUNG	Jede Feldart hat besondere Eigenschaften und die Parameter müssen wohldefiniert sein.
ABHÄNGIGKEITEN	FA22→ p. 17, FA20→ p. 17, FA21→ p. 17
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12, menschlicher Benutzer→ p. 11
<b>ID</b>	<b>FA25</b>
TITEL:	Feldart Düne
BESCHREIBUNG:	Die Düne ist ein Wüstenfeld, welches hügelig und höher gelegen ist. Felder der Feldart Düne sind betretbar und haben ein hohes Höhenniveau.
BEGRÜNDUNG	Jede Feldart hat besondere Eigenschaften und die Parameter müssen wohldefiniert sein.
ABHÄNGIGKEITEN	FA22→ p. 17, FA20→ p. 17, FA21→ p. 17
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12, menschlicher Benutzer→ p. 11
<b>ID</b>	<b>FA26</b>
TITEL:	Feldart Felsplateau
BESCHREIBUNG:	Das Felsplateau ist ein Felsenfeld, welches flach und niedrig gelegen ist. Felder der Feldart Felsplateau sind betretbar und haben ein niedriges Höhenniveau.
BEGRÜNDUNG	Jede Feldart hat besondere Eigenschaften und die Parameter müssen wohldefiniert sein.
ABHÄNGIGKEITEN	FA22→ p. 17, FA20→ p. 17, FA21→ p. 17
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12, menschlicher Benutzer→ p. 11
<b>ID</b>	<b>FA27</b>
TITEL:	Feldart Gebirge
BESCHREIBUNG:	Das Gebirge ist ein Felsenfeld, welches hoch gelegen und unpassierbar ist. Felder der Feldart Gebirge sind nicht betretbar und haben ein hohes Höhenniveau.
BEGRÜNDUNG	Jede Feldart hat besondere Eigenschaften und die Parameter müssen wohldefiniert sein.
ABHÄNGIGKEITEN	FA22→ p. 17, FA20→ p. 17, FA21→ p. 17
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12, menschlicher Benutzer→ p. 11

ID	FA28
TITEL:	Entfernung Felder
BESCHREIBUNG:	Die Entfernung zwischen zwei Feldern A und B ist definiert als die minimale Anzahl von aufeinanderfolgenden Schritten auf alle 8 Nachbarfelder (gemäß der Moore Neighborhood-Definition), um von A nach B zu gelangen.
BEGRÜNDUNG	Der Abstand zwischen Feldern muss klar definiert sein.
ABHÄNGIGKEITEN	FA18 → p. 16
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13

### 2.3.4 Great Houses

ID	FA29
TITEL:	Great Houses
BESCHREIBUNG:	Im Spiel gibt es sechs Great Houses, das heißt große Adelshäuser, die miteinander konkurrieren. Jedes der Häuser hat genau einen Namen und genau eine Farbe und besteht aus einer Reihe von Charakteren, die bei FA36 → p. 21 genauer beschrieben werden
BEGRÜNDUNG	Die Great Houses sollen farblich, namentlich und durch die Eigenschaften ihrer Charaktere unterscheidbar sein und dem Spieler Abwechslung bieten und verschiedene Strategien in das Spiel bringen.
ABHÄNGIGKEITEN	FA36 → p. 21
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12, menschlicher Benutzer → p. 11

ID	FA30
TITEL:	Haus Corrino
BESCHREIBUNG:	Dieses Great House hat den Namen <i>House Corrino</i> , die Farbe <i>Gold</i> und die folgenden Charaktere: Emperor Shaddam IV Corrino (Noble), Princess Irulan Corrino (Bene Gesserit), Count Hasimir Fenring (Mentat), Lady Margot Fenring (Bene Gesserit), Reverend Mother Gaius Helen Mohiam (Bene Gesserit) und Captain Aramsham (Fighter).
BEGRÜNDUNG	Die Great Houses sollen farblich, namentlich und durch die Eigenschaften ihrer Charaktere unterscheidbar sein.
ABHÄNGIGKEITEN	FA29 → p. 19, FA41 → p. 21, FA42 → p. 22, FA43 → p. 22, FA44 → p. 22
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12, menschlicher Benutzer → p. 11

ID	FA31
TITEL:	Haus Atreides
BESCHREIBUNG:	Dieses Great House hat den Namen <i>House Atreides</i> , die Farbe <i>Grün</i> und die folgenden Charaktere: Duke Leto Atreides (Noble), Paul Atreides (Noble), Lady Jessica (Bene Gesserit), Thufir Hawat (Mentat), Gurney Halleck (Fighter) und Space Pug, Duke Letos tapferer Mopshund (Fighter).
BEGRÜNDUNG	Die Great Houses sollen farblich, namentlich und durch die Eigenschaften ihrer Charaktere unterscheidbar sein.
ABHÄNGIGKEITEN	FA29 → p. 19, FA41 → p. 21, FA42 → p. 22, FA43 → p. 22, FA44 → p. 22
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12, menschlicher Benutzer → p. 11

ID	FA32
TITEL:	Haus Harkonnen
BESCHREIBUNG:	Dieses Great House hat den Namen <i>House Harkonnen</i> , die Farbe <i>Rot</i> und die folgenden Charaktere: Baron Vladimir Harkonnen (Noble), Count Glossu Beast Rabban (Fighter), Feyd-Rautha Rabban (Fighter), Piter De Vries (Mentat), Iakin Nefud (Fighter) und Pet Spider (Fighter).
BEGRÜNDUNG	Die Great Houses sollen farblich, namentlich und durch die Eigenschaften ihrer Charaktere unterscheidbar sein.
ABHÄNGIGKEITEN	FA29→ p. 19, FA41→ p. 21, FA42→ p. 22, FA43→ p. 22, FA44→ p. 22
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12, menschlicher Benutzer→ p. 11

ID	FA33
TITEL:	Haus Ordos
BESCHREIBUNG:	Dieses Great House hat den Namen <i>House Ordos</i> , die Farbe <i>Blau</i> und die folgenden Charaktere: Executrix (Noble), The Speaker (Noble), Ammon (Mentat), Edric (Mentat), Roma Atani (Mentat) und Robot (Fighter).
BEGRÜNDUNG	Die Great Houses sollen farblich, namentlich und durch die Eigenschaften ihrer Charaktere unterscheidbar sein.
ABHÄNGIGKEITEN	FA29→ p. 19, FA41→ p. 21, FA42→ p. 22, FA43→ p. 22, FA44→ p. 22
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12, menschlicher Benutzer→ p. 11

ID	FA34
TITEL:	Haus Richese
BESCHREIBUNG:	Dieses Great House hat den Namen <i>House Richese</i> , die Farbe <i>Silber</i> und die folgenden Charaktere: Count Ilban Richese (Noble), Helena Richese (Noble), Haloa Rund (Mentat), Flinto Kinnis (Mentat), Tenu Chobyn (Mentat) und Yresk (Fighter).
BEGRÜNDUNG	Die Great Houses sollen farblich, namentlich und durch die Eigenschaften ihrer Charaktere unterscheidbar sein.
ABHÄNGIGKEITEN	FA29→ p. 19, FA41→ p. 21, FA42→ p. 22, FA43→ p. 22, FA44→ p. 22
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12, menschlicher Benutzer→ p. 11

ID	FA35
TITEL:	Haus Vernius
BESCHREIBUNG:	Dieses Great House hat den Namen <i>House Vernius</i> , die Farbe <i>Violett</i> und die folgenden Charaktere: Earl Dominic Vernius (Noble), Lady Shando Vernius (Noble), Kailea Vernius (Noble), Tessia Vernius (Bene Gesserit), Rhombur Vernius (Fighter) und Bronso Vernius (Mentat).
BEGRÜNDUNG	Die Great Houses sollen farblich, namentlich und durch die Eigenschaften ihrer Charaktere unterscheidbar sein.
ABHÄNGIGKEITEN	FA29→ p. 19, FA41→ p. 21, FA42→ p. 22, FA43→ p. 22, FA44→ p. 22
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12, menschlicher Benutzer→ p. 11

### 2.3.5 Charaktere

<b>ID</b>	<b>FA36</b>
TITEL:	Charaktere
BESCHREIBUNG:	Charaktere haben bestimmte Eigenschaften, deren Werte (abhängig vom Charakter-Typ) in der Partiekonfiguration definiert sind. Sie haben einen <i>Namen</i> , ein <i>Haus</i> , welchem sie angehören, <i>Health Points</i> , <i>Movement Points</i> , <i>Angriffsschaden</i> und ein <i>Inventar</i> . Die Werte und das Inventar aller Charaktere sind für alle Spieler sichtbar.
BEGRÜNDUNG	Charaktere werden strategisch durch ihre Eigenschaften und deren Werte unterschieden.
ABHÄNGIGKEITEN	FA125 → p. 43, FA15 → p. 16
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12, menschlicher Benutzer → p. 11
<b>ID</b>	<b>FA37</b>
TITEL:	Eigenschaft Health Points
BESCHREIBUNG:	Charaktere haben sogenannte Health Points, die angeben, wie viel Schaden der Charakter noch bekommen kann, bevor er stirbt.
BEGRÜNDUNG	Charaktere brauchen Lebenspunkte, um zu entscheiden, ob sie noch leben oder nicht.
ABHÄNGIGKEITEN	FA36 → p. 21
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12, menschlicher Benutzer → p. 11
<b>ID</b>	<b>FA38</b>
TITEL:	Tod eines Charakters
BESCHREIBUNG:	Wenn der Wert der Health Points eines Charakters auf 0 sinkt, dann ist der Charakter besiegt. Anschließend wird dieser von der Karte entfernt und alle Spicekrümel nach der Spicekrümel-Logik auf dem Spielfeld verteilt.
BEGRÜNDUNG	Es muss definiert sein, was nach dem Tod eines Charakters inklusive seines Spice passiert.
ABHÄNGIGKEITEN	FA37 → p. 21, FA64 → p. 27
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12, menschlicher Benutzer → p. 11
<b>ID</b>	<b>FA39</b>
TITEL:	Heilung von Health-Points
BESCHREIBUNG:	Ein Charakter, welcher in seinem Zug keine Movement-Points ausgibt, bekommt am Ende des Zuges seine Health-Points um seine Heilungs-Health Points erhöht.
BEGRÜNDUNG	Es muss klar definiert sein, wann ein Charakter sich heilt.
ABHÄNGIGKEITEN	
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12
<b>ID</b>	<b>FA40</b>
TITEL:	Charakter-Typen
BESCHREIBUNG:	Der Charakter-Typ bestimmt die <i>Health Points</i> , <i>Heilungs-Health Points</i> , <i>Movement Points</i> , <i>Action Points</i> , <i>Inventargröße</i> und den <i>Angriffsschaden</i> des Charakters. Zudem ermöglicht der Charakter-Typ, dem Charakter eventuell spezielle Aktionen ausführen zu können.
BEGRÜNDUNG	Charaktere werden strategisch durch ihre Eigenschaften und deren Werte unterschieden.
ABHÄNGIGKEITEN	FA36 → p. 21, FA56 → p. 25
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12, menschlicher Benutzer → p. 11

<b>ID</b>	<b>FA41</b>
TITEL:	Noble
BESCHREIBUNG:	Noble ist ein hochwohlgeborener Adliger. Dafür sind seine Eigenschaften nur durchschnittlich. Die genauen Werte seiner Eigenschaften sind in der Partiekonfiguration zu konfigurieren.
BEGRÜNDUNG	Die Eigenschaften der Charakter-Typen werden in der Partiekonfiguration konfiguriert und zu einem späteren Zeitpunkt genauer festgelegt, um eine Einfluss-Balance zwischen den Charakter-Typen zu ermöglichen.
ABHÄNGIGKEITEN	FA36→ p. 21, FA40→ p. 21
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12, menschlicher Benutzer→ p. 11
<b>ID</b>	<b>FA42</b>
TITEL:	Mentat
BESCHREIBUNG:	Mentat ist ein Charakter-Typ, der sehr schlau ist, und deshalb besonders effizient darin, Aktionen auszuführen und Spice zu sammeln. Er hält aber im Kampf nicht besonders viel aus. Die genauen Werte der Eigenschaften sind in der Partiekonfiguration zu konfigurieren.
BEGRÜNDUNG	Die Eigenschaften der Charakter-Typen werden in der Partiekonfiguration konfiguriert und zu einem späteren Zeitpunkt genauer festgelegt, um eine Einfluss-Balance zwischen den Charakter-Typen zu ermöglichen.
ABHÄNGIGKEITEN	FA36→ p. 21, FA40→ p. 21
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12, menschlicher Benutzer→ p. 11
<b>ID</b>	<b>FA43</b>
TITEL:	Bene Gesserit
BESCHREIBUNG:	Bene Gesserit ist ein Charakter-Typ, der schnell und wendig ist und eine hohe Heilungsrate besitzt. Die genauen Werte der Eigenschaften sind in der Partiekonfiguration zu konfigurieren.
BEGRÜNDUNG	Die Eigenschaften der Charakter-Typen werden in der Partiekonfiguration konfiguriert und zu einem späteren Zeitpunkt genauer festgelegt, um eine Einfluss-Balance zwischen den Charakter-Typen zu ermöglichen.
ABHÄNGIGKEITEN	FA36→ p. 21, FA40→ p. 21
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12, menschlicher Benutzer→ p. 11
<b>ID</b>	<b>FA44</b>
TITEL:	Fighter
BESCHREIBUNG:	Fighter ist ein Charakter-Typ, der im Kampf viel Schaden macht und viele Health-Points hat. Die genauen Werte der Eigenschaften sind in der Partiekonfiguration zu konfigurieren.
BEGRÜNDUNG	Die Eigenschaften der Charakter-Typen werden in der Partiekonfiguration konfiguriert und zu einem späteren Zeitpunkt genauer festgelegt, um eine Einfluss-Balance zwischen den Charakter-Typen zu ermöglichen.
ABHÄNGIGKEITEN	FA36→ p. 21, FA40→ p. 21
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12, menschlicher Benutzer→ p. 11

<b>ID</b>	<b>FA45</b>
TITEL:	Stärken und Schwächen von Charakter-Typen
BESCHREIBUNG:	Charakter-Typen haben unterschiedliche Stärken und Schwächen ähnlich dem Schere-Stein-Papier Prinzip.
BEGRÜNDUNG	Kein Charakter-Typ soll alleine alles gut können, sie sollen sich gegenseitig ergänzen und effektiv gegeneinander sein.
ABHÄNGIGKEITEN	FA36→ p. 21, FA40→ p. 21
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12, menschlicher Benutzer→ p. 11

### 2.3.6 Bewegung

<b>ID</b>	<b>FA46</b>
TITEL:	Bewegung von Charakteren
BESCHREIBUNG:	Zu Beginn seines Zuges bekommt jeder Charakter so viele Movement Points (MP), wie in der Partiekonfiguration für seinen Charakter-Typ festgelegt ist. Für einen Movement Point kann sich der Charakter auf ein betretbares Nachbarfeld bewegen.
BEGRÜNDUNG	Es muss klar definiert sein, wie weit und wohin sich Charaktere jede Runde bewegen dürfen.
ABHÄNGIGKEITEN	FA36→ p. 21, FA40→ p. 21, FA18→ p. 16, FA20→ p. 17, FA47→ p. 23
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12

<b>ID</b>	<b>FA47</b>
TITEL:	Bewegungsrichtung von Charakteren
BESCHREIBUNG:	Ein Charakter kann sich in horizontaler, vertikaler oder diagonaler Richtung bewegen.
BEGRÜNDUNG	Es muss klar definiert sein, in welche Richtungen sich ein Charakter-Typ bewegen darf.
ABHÄNGIGKEITEN	FA36→ p. 21, FA40→ p. 21, FA18→ p. 16, FA20→ p. 17
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12

<b>ID</b>	<b>FA48</b>
TITEL:	Drängeln von Charakteren
BESCHREIBUNG:	Bewegt sich ein Charakter auf ein Feld, auf dem bereits ein Charakter steht, tauschen die beiden Charaktere die Felder.
BEGRÜNDUNG	Die Interaktion zwischen einem Charakter, der bereits auf einem Feld steht und einem zweiten Charakter, welcher das Feld betritt, muss eindeutig definiert sein.
ABHÄNGIGKEITEN	FA36→ p. 21, FA40→ p. 21, FA18→ p. 16, FA22→ p. 17, FA20→ p. 17
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12

<b>ID</b>	<b>FA49</b>
TITEL:	Lautheit in der Wüste
BESCHREIBUNG:	Am Anfang seines Zuges gilt ein Charakter als leise. Charaktere werden als laut markiert, wenn sie mindestens zweimal aktiv ein Wüstenfeld in einem Zug betreten.
BEGRÜNDUNG	Es muss klar definiert werden, wann Charaktere als laut markiert werden.
ABHÄNGIGKEITEN	FA36→ p. 21
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12

ID	FA50
TITEL:	Abliefern von Spice
BESCHREIBUNG:	Ist ein Charakter auf einem Nachbarfeld seiner Stadt, so wird das Spice aus seinem Inventar direkt der Stadt übergeben und dem Spice-Vorrat des Hauses hinzugezählt. Das heißt der Charakter hat genau 0 Spice im Inventar und die Stadt genau die Menge mehr, die der Charakter vorher im Inventar hatte.
BEGRÜNDUNG	Es muss klar geregelt sein, wie Charaktere Spice bei ihrer Stadt abliefern.
ABHÄNGIGKEITEN	FA36 → p. 21
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

### 2.3.7 Aktionen

ID	FA51
TITEL:	Aktionen
BESCHREIBUNG:	Aktionen sind aktive Handlungen, welche Charaktere ausführen können. Das Einsetzen von Aktionen kostet den Charakter immer Action Points, die der Charakter zu Beginn seines Zuges erhält.
BEGRÜNDUNG	Aktionen sind notwendig, um zu definieren, welche Handlungen welche Charaktere ausführen können.
ABHÄNGIGKEITEN	FA36 → p. 21, FA40 → p. 21
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

ID	FA52
TITEL:	normale Aktion
BESCHREIBUNG:	Normale Aktionen können von allen Charakteren ausgeführt werden. Dies ist unabhängig davon, welchen Charakter-Typ ein Charakter hat. Sie kosten den Charakter immer genau einen Action Point. Zu den normalen Aktionen gehören Angriff, Spice aufsammeln und Spice übergeben.
BEGRÜNDUNG	Normale Aktionen kategorisieren alle Aktionen, welche durch jeden Charakter ausgeführt werden können.
ABHÄNGIGKEITEN	FA36 → p. 21, FA40 → p. 21, FA51 → p. 24
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

ID	FA53
TITEL:	Aktion: Angriff
BESCHREIBUNG:	Mit dieser normalen Aktion kann ein beliebiger Charakter einen gegnerischen Charakter auf einem benachbarten Feld verletzen. Der gegnerische Charakter bekommt, wenn beide Charaktere auf dem gleichen Höhengniveau sind, den Angriffsschaden des Angreifers von seinen Health Points abgezogen. Ist der Angreifer auf einem hohen Feld und der gegnerische Charakter auf einem niedrigen Feld, so bekommt der gegnerische Charakter $\frac{4}{3}$ * Angriffsschaden des Angreifers von seinen Health Points abgezogen. Bei gegenteiliger Situation erhält der gegnerische Charakter $\frac{2}{3}$ * Angriffsschaden des Angreifers von seinen Health Points abgezogen.
BEGRÜNDUNG	Es muss klar definiert sein, wie die Aktion Angriff abläuft, welche Vorbedingungen gelten müssen und welche Auswirkungen sie hat.
ABHÄNGIGKEITEN	FA36 → p. 21, FA40 → p. 21, FA52 → p. 24, FA21 → p. 17
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12



<b>ID</b>	<b>FA54</b>
TITEL:	Aktion: Spice aufsammeln
BESCHREIBUNG:	Ein Charakter kann mit einer Aktion einen Spicekrümel von dem Feld, auf dem er steht, in sein Inventar aufnehmen, sofern dieses nicht voll ist.
BEGRÜNDUNG	Es muss klar definiert sein, wie die Aktion Spice aufsammeln abläuft, welche Vorbedingungen gelten müssen und welche Auswirkungen sie hat.
ABHÄNGIGKEITEN	FA36 → p. 21, FA40 → p. 21, FA52 → p. 24, FA21 → p. 17
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12
<b>ID</b>	<b>FA55</b>
TITEL:	Aktion: Spice übergeben
BESCHREIBUNG:	Ein Charakter kann mit einer Aktion eine beliebige Teilmenge der Spicekrümel in seinem Inventar auf das Inventar eines benachbarten verbündeten Charakters übertragen, sofern dieser genügend Platz im Inventar hat.
BEGRÜNDUNG	Es muss klar definiert sein, wie die Aktion Spice übergeben abläuft, welche Vorbedingungen gelten müssen und welche Auswirkungen sie hat.
ABHÄNGIGKEITEN	FA36 → p. 21, FA40 → p. 21, FA52 → p. 24
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12
<b>ID</b>	<b>FA56</b>
TITEL:	spezielle Aktion
BESCHREIBUNG:	Über diese Kategorie von Aktionen verfügen nur spezielle Charakter-Typen. Sie kosten immer die gesamten Action Points, die der Charakter in einer Runde zur Verfügung hat.
BEGRÜNDUNG	Spezielle Aktionen kategorisieren alle Aktionen, welche nur durch bestimmte Charakter-Typen ausgeführt werden können.
ABHÄNGIGKEITEN	FA36 → p. 21, FA40 → p. 21, FA51 → p. 24
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12
<b>ID</b>	<b>FA57</b>
TITEL:	Aktion: Kanly
BESCHREIBUNG:	Kanly ist eine spezielle Aktion, welche nur Charaktere vom Charakter-Type Noble ausführen können. Und die nur auf Charaktere vom Charakter-Type Nobel angewendet werden kann. Dies ist ein ritualisierter Angriff zwischen Nobel, welcher das Ziel des Angriffs mit einer Kanly-Erfolgswahrscheinlichkeit, welche in der Partie-Konfiguration festgehalten ist, besiegt. Bei Misserfolg passiert nichts.
BEGRÜNDUNG	Es muss klar definiert sein, wie die Aktion Kanly abläuft, welche Vorbedingungen gelten müssen und welche Auswirkungen sie hat.
ABHÄNGIGKEITEN	FA36 → p. 21, FA40 → p. 21, FA56 → p. 25, FA54 → p. 24
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

<b>ID</b>	<b>FA58</b>
TITEL:	Aktion: Family Atomics
BESCHREIBUNG:	Die spezielle Aktion Family Atomics kann nur durch Nobel eingesetzt werden. Dies ist für jedes Great House bis zu drei mal möglich, da der Atomic vorrat jedes Hauses drei entspricht. Hierbei wird ein beliebiges Zielfeld gewählt Auf diesem Zielfeld und auf dem gesamten 3x3 Quadrat ums Zielfeld werden Gebirge zu Felsplateaus und Dünen zu Flachsand Städten passiert nichts, alle Charaktere werden besiegt, Sandwürmer und Spicekrümel verschwinden. Wird hierbei ein gegnerischer Charakter besiegt wird und der Gegner zuvor nicht die Greatkonvention verletzt hat wird die Greatkonvention verletzt.
BEGRÜNDUNG	Es muss klar definiert sein, wie die Aktion abläuft, welche Vorbedingungen gelten müssen und welche Auswirkungen sie hat.
ABHÄNGIGKEITEN	FA36 → p. 21, FA40 → p. 21, FA56 → p. 25, FA54 → p. 24, FA59 → p. 26
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12
<b>ID</b>	<b>FA59</b>
TITEL:	Great Convention
BESCHREIBUNG:	Die Great Convention besagt, dass wenn ein Great House Atomics gegen einen Charakter eines anderen Great House einsetzt und zuvor keiner gegen die Great Convention verstoßen hat, dass dieses Great House geächtet wird. Das gegnerische Great House erhält dann von den anderen vier Great Houses zu Anfang der nächsten Runde jeweils einen zufälligen Charakter. Dieser wird auf ein zufälliges Feld auf der Karte platziert. Zudem darf das Great House, dass durch den Atomic Einsatz angegriffen wurde unbestraft auch Atomics auf das andere Great House einsetzen.
BEGRÜNDUNG	Die Great Convention muss eindeutig definiert sein.
ABHÄNGIGKEITEN	FA58 → p. 25
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12
<b>ID</b>	<b>FA60</b>
TITEL:	Aktion: Spice Hoarding
BESCHREIBUNG:	Die spezielle Aktion Spice Hoarding kann nur von einem Charakter mit dem Charakter-Typ Mentat eingesetzt werden. Hierbei werden alle Spicekrümel auf dem aktuellen Feld des Charakters und auf den Nachbarfeldern eingesammelt und in das Inventar des Charakters aufgenommen.
BEGRÜNDUNG	Es muss klar definiert sein, wie die Aktion Spice Hoarding abläuft, welche Vorbedingungen gelten müssen und welche Auswirkungen sie hat.
ABHÄNGIGKEITEN	FA36 → p. 21, FA40 → p. 21, FA56 → p. 25, FA54 → p. 24
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12
<b>ID</b>	<b>FA61</b>
TITEL:	Aktion: Voice
BESCHREIBUNG:	Die spezielle Aktion Voice kann nur von einem Charakter mit dem Charakter-Typ Bene Gesserit eingesetzt werden. Hierbei kann die Aktion auf einen benachbarten Charakter verbündet oder feindlich eingesetzt werden. Dieser Charakter übergibt dann hypnotisiert sofort das gesamte Spice in seinem Inventar an den Charakter, der die Aktion eingesetzt hat, bis dessen Inventar voll ist.
BEGRÜNDUNG	Es muss klar definiert sein, wie die Aktion Voice abläuft, welche Vorbedingungen gelten müssen und welche Auswirkungen sie hat.
ABHÄNGIGKEITEN	FA36 → p. 21, FA40 → p. 21, FA56 → p. 25, FA55 → p. 25
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

ID	FA62
TITEL:	Aktion: Sword Spin
BESCHREIBUNG:	Die spezielle Aktion Sword Spin kann nur von einem Charakter mit dem Charakter-Typ Fighter eingesetzt werden. Die Aktion Sword Spin macht allen benachbarten Charakteren Schaden wie bei einem normalen Angriff.
BEGRÜNDUNG	Es muss klar definiert sein, wie die Aktion Sword Spin abläuft, welche Vorbedingungen gelten müssen und welche Auswirkungen sie hat.
ABHÄNGIGKEITEN	FA36→ p. 21, FA40→ p. 21, FA56→ p. 25,
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12

### 2.3.8 Spice

ID	FA63
TITEL:	Spice Blow
BESCHREIBUNG:	Spice Blow findet statt, wenn zu Beginn eine Runde die Spicekrümel Anzahl auf der Karte kleiner als der Spiceschwellwert ist. Es wird ein zufälliges Wüstenfeld gewählt und eine Zufallszahl im Intervall [3,6] ermittelt. Das gewählte Wüstenfeld und die benachbarten Wüstenfelder werden zufällig auf Flachland oder Düne gesetzt. Entsprechend der ermittelten Zufallszahl werden Spicekrümel über die Felder mittels Spice Streuung verstreut.
BEGRÜNDUNG	Es muss klar definiert sein, wie der Spice Blow funktioniert.
ABHÄNGIGKEITEN	FA18→ p. 16, FA22→ p. 17, FA64→ p. 27,
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12

ID	FA64
TITEL:	Spice Streuung
BESCHREIBUNG:	Spicekrümel werden bei Tod eines Charakters oder einem Spice Blow verteilt. Es wird das aktuelle Feld mit Spice bestreut, falls kein Spice auf dem aktuellen Feld liegt. Wenn das aktuelle Feld bestreut ist, werden n-1 zufällige Nachbarmfelder bestreut, falls dieses betretbar sind.
BEGRÜNDUNG	Es muss klar definiert sein wie die Spice Streuung funktioniert.
ABHÄNGIGKEITEN	FA18→ p. 16, FA22→ p. 17
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12, menschlicher Benutzer→ p. 11

### 2.3.9 Dünenwanderung

ID	FA65
TITEL:	Dünenwanderung
BESCHREIBUNG:	Die Dünenwanderung muss mittels zellulären Automaten implementiert werden. Jedes Wüstenfeld hat ein niedriges oder hohes Höhenniveau, ist also Flachsand oder eine Düne. Wir betrachten niedrige Felder als "tot" und hohe als "lebendig" im Sinne des zellulären Automaten.
BEGRÜNDUNG	Es muss klar definiert sein wie die Dünenwanderung funktioniert.
ABHÄNGIGKEITEN	FA18→ p. 16, FA22→ p. 17, FA66→ p. 27
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12

<b>ID</b>	<b>FA66</b>
TITEL:	Dünenwanderung-Rundenphase
BESCHREIBUNG:	In der Dünenwanderungs-Rundenphase wird eine Iteration für den zellulären Automaten gemacht. Dabei wird für jedes Wüstenfeld aus dem eigenen Höhengniveau-Zustand und dem der Nachbar-Felder der Folgezustand berechnet, also ob das Feld Flachsand oder eine Dune bleiben bzw. werden soll. Städte und Gebirge gelten dabei als konstant hohe, also "lebendige", Felder, niedrige Felsplateaus entsprechend als konstant "tote" Felder.
BEGRÜNDUNG	Es muss klar definiert sein wie die Dünenwanderung-Rundenphase funktioniert.
ABHÄNGIGKEITEN	FA18 → p. 16, FA22 → p. 17, FA65 → p. 27, FA67 → p. 28
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12
<b>ID</b>	<b>FA67</b>
TITEL:	Transitionsregel des zellulären Automaten
BESCHREIBUNG:	Die Transitionsregel des zellulären Automaten beschreibt, für den zellulären Automaten, bei welcher Anzahlen an lebendigen Nachbarn eine tote Zelle lebendig wird oder eine lebendige Zelle überlebt. Die Transitionen orientieren sich an Conway's Original Game of Life. Eine spezifische zu verwendete Regel wird in der Partie-Konfiguration festgehalten. Das Format ist hierbei ein String mit dem Inhalt B <sub>i</sub> Ziffernfolge <sub>i</sub> /S <sub>j</sub> Ziffernfolge <sub>j</sub> (B für Born und S für survive).
BEGRÜNDUNG	Es muss klar definiert sein, was mit den Transitionsregeln gemeint ist, und wo diese spezifiziert werden.
ABHÄNGIGKEITEN	FA66 → p. 27, FA65 → p. 27
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

### 2.3.10 Sandsturm

<b>ID</b>	<b>FA68</b>
TITEL:	Sandsturm
BESCHREIBUNG:	Zu Beginn der Partie wird vom Server ein zufälliges Feld gewählt zentrales Feld des Sandsturms gewählt. Auf diesem Feld und den Nachbarn, also einem 3x3-Quadrat aus Feldern tobt ein gefährlicher Sandsturm. Zu Beginn jeder Runde wird die Sandsturm-Rundenphase ausgelöst. Charaktere, die sich im Sturm befinden, können keine Aktionen machen, und auch selbst nicht das Ziel von Aktionen sein. Lediglich durch einen Atomics-Einsatz können sie Ziel einer Aktion sein.
BEGRÜNDUNG	Es bedarf einer klaren Definition des Sandsturms.
ABHÄNGIGKEITEN	FA69 → p. 28
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12
<b>ID</b>	<b>FA69</b>
TITEL:	Sandsturm-Rundenphase
BESCHREIBUNG:	In der Sandsturm-Rundenphase wird das Zentralfeld des Sturms auf ein zufälliges Nachbarfeld gesetzt. Danach wird jedes Feld im 3x3 Bereich des Sturms zufällig auf ein niedriges oder hohes Höhengniveau gesetzt. Der Sandsturm bringt also kleine lokale Störungen in den zellulären Automaten ein, und verhindert damit statische Zustände des Automaten.
BEGRÜNDUNG	Es bedarf einer klaren Definition des Ablaufes der Sandsturm-Rundenphase.
ABHÄNGIGKEITEN	FA68 → p. 28, FA67 → p. 28, FA21 → p. 17
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

### 2.3.11 Sandwurm

ID	FA70
TITEL:	Sandwurm
BESCHREIBUNG:	Auf Arakis leben riesige Sandwürmer, sie können sich über Wüstenfelder bewegen, aber nicht über Felsenfelder. Der Sandwurm wird in der Sandwurm-Rundenphase aktiv. Wüstenfelder, auf denen sich ein Sandwurm befindet, gelten als nicht betretbar. Alle Wüstenfelder, von denen ein Sandwurm sich wegbewegt, werden auf Flachsand gesetzt.
BEGRÜNDUNG	Es muss klar definiert sein, was ein Sandwurm ist und welche Eigenschaften er hat.
ABHÄNGIGKEITEN	FA71→ p. 29, FA20→ p. 17, FA22→ p. 17
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12
ID	FA71
TITEL:	Sandwurm-Rundenphase
BESCHREIBUNG:	Zu Beginn jeder Runde in der Sandwurm-Rundenphase bewegt sich der Sandwurm, falls vorhanden n Felder auf eine Zielperson zu (n ist in der Partie-Konfiguration definiert). Wenn der Sandwurm sich auf das Feld eines Charakters bewegt, so wird dieser verschluckt und der Sandwurm verschwindet. Falls der Sandwurm keine nur über Wüstenfelder führende Strecke hat, um zu seiner Zielperson zu gelangen, verschwindet er und taucht direkt danach auf einem zufällig gewählten Wüstenfeld wieder auf. Seine Rundenphase ist damit beendet. Falls es keinen Sandwurm gibt, wird geprüft, ob in der vorangegangenen Rundenphase ein Charakter als Lauf markiert wurde. Falls keiner als laut markiert wurde passiert nichts. Wenn es mindestens einen lauten Charakter gibt, wird dieser oder ein zufälliger lauter Charakter vom Sandwurm ausgewählt. Der Sandwurm taucht dann auf einem zufälligen Wüsten Feld, welches kein Nachbarfeld des ausgewählten Charakters ist auf. Seine Rundenphase endet und er verfolgt später weiter den gleichen Charakter.
BEGRÜNDUNG	Die Sandwurm-Rundenphase muss wohl definiert sein.
ABHÄNGIGKEITEN	FA70→ p. 29, FA36→ p. 21, FA49→ p. 23
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12

### 2.3.12 Klonen von besiegten Charakteren

ID	FA72
TITEL:	Klon-Rundenphase
BESCHREIBUNG:	Jeder besiegte Charakter eines der beiden Great Houses wird mit einer Klon-Erfolgswahrscheinlichkeit aus der Partiekonfiguration geklont. Der geklonte Charakter wird nun auf ein zufälliges freies Nachbarfeld der Stadt platziert. Charaktere die vom Sandwurm verschluckt wurden, können nicht geklont werden.
BEGRÜNDUNG	Es muss klar definiert sein, wie die Klon-Rundenphase abläuft und von welchen Parameter sie beeinflusst wird.
ABHÄNGIGKEITEN	FA70→ p. 29, FA36→ p. 21
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12

### 2.3.13 Partievorbereitung

ID	FA73
TITEL:	Partievorbereitung
BESCHREIBUNG:	Bevor die Partie startet, wählen die Spieler eines der Great Houses für sich aus. Dies geschieht in der Great House Wahl.
BEGRÜNDUNG	Es muss klar definiert sein, was in der Partie-Vorbereitung passiert.
ABHÄNGIGKEITEN	FA74 → p. 30, FA29 → p. 19
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

ID	FA74
TITEL:	Great House Wahl
BESCHREIBUNG:	Der Server nimmt aus der Liste der sechs Great Houses zwei zufällige heraus, die dem Spieler angezeigt werden. Der Spieler wählt eines davon aus. Dies findet Nebenläufig für beide Spieler statt. Der Server muss den beiden Spielern jeweils eine Menge aus zwei Häusern zur Auswahl anbieten, diese Mengen müssen disjunkt sein.
BEGRÜNDUNG	Es muss klar definiert sein, wie die Great House Wahl abläuft.
ABHÄNGIGKEITEN	FA29 → p. 19
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

### 2.3.14 Beginn einer Partie

ID	FA75
TITEL:	Beginn einer Partie
BESCHREIBUNG:	Vor der ersten Runde weist der Server jedem der Häuser eine der beiden Städte zu, und platziert dann alle Charaktere eines Hauses auf zufälligen freien betretbaren Felder benachbart der jeweiligen Stadt.
BEGRÜNDUNG	Es muss klar definiert sein, wie die Partie beginnt.
ABHÄNGIGKEITEN	FA29 → p. 19, FA36 → p. 21, FA18 → p. 16
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

### 2.3.15 Runden und Züge

ID	FA76
TITEL:	Runden und Züge
BESCHREIBUNG:	Deserts of Dune läuft in Runden ab, in denen Ereignisse passieren, und die einzelnen Charaktere nacheinander ihre Züge machen. Zu Beginn jeder Runde handelt der Server einige Ereignisse ab. Die Reihenfolge der Rundenphasen ist: 1. Dünenwanderungs-Rundenphase 2. Sandsturm-Rundenphase 3. Sandwurm-Rundenphase 4. Klon-Rundenphase 5. Charakterzug-Rundenphase
BEGRÜNDUNG	Es muss klar definiert sein, welche Rundenphasen in welcher Reihenfolge stattfinden.
ABHÄNGIGKEITEN	FA66 → p. 27, FA69 → p. 28, FA72 → p. 29, FA77 → p. 30
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

<b>ID</b>	<b>FA77</b>
TITEL:	Charakterzug-Rundenphase
BESCHREIBUNG:	In der Charakterzug-Rundenphase werden zunächst alle Charaktere vom Server für diese Runde in eine zufällige Reihenfolge gebracht. Die Charaktere kommen nun der Reihe nach dran, und können, wenn sie zu diesem Zeitpunkt leben, ihren Zug machen, der aus mehreren Zugphasen besteht.
BEGRÜNDUNG	s muss klar definiert sein, wie die Charakterzug-Rundenphase abläuft.
ABHÄNGIGKEITEN	FA36 → p. 21
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

<b>ID</b>	<b>FA78</b>
TITEL:	Züge von Charakteren
BESCHREIBUNG:	Züge von Charakteren bestehen aus Zugphasen, in denen jeweils entweder ein Bewegungsschritt oder eine Aktion gemacht wird. Wenn ein Charakter mit seinem Zug an der Reihe ist, werden zuerst seine Movement Points und Action Points auf die seinem Charakter-Typ entsprechenden Werte aus der Partie-Konfiguration gesetzt. Der Spieler kann dann für den aktiven Charakter diese Punkte in beliebiger Reihenfolge, Zugphase für Zugphase, einen nach dem anderen ausgeben.
BEGRÜNDUNG	Es muss klar definiert sein, wie Züge von Charakteren ablaufen.
ABHÄNGIGKEITEN	
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

### 2.3.16 Ende der Partie und Bestimmung des Siegers

<b>ID</b>	<b>FA79</b>
TITEL:	Ende der Partie
BESCHREIBUNG:	Wenn in der Sandwurm-Rundenphase der letzte Charakter, den ein Haus kontrolliert, durch einen gewöhnlichen Sandwurm-Angriff verschluckt wird, während das andere Haus nach Abhandlung dieser Sandwurm-Rundenphase noch Charaktere hat, gewinnt das andere Haus unmittelbar die Partie.
BEGRÜNDUNG	Es muss klar definiert sein, wann die Partie endet und wie der Sieger ermittelt wird.
ABHÄNGIGKEITEN	FA36 → p. 21, FA70 → p. 29, FA71 → p. 29
PRIORITÄT	5
AKTEUR	Server → p. 11, Auftraggeber → p. 12

<b>ID</b>	<b>FA80</b>
TITEL:	Behandlung überlanger Partien
BESCHREIBUNG:	Wenn die Partie über mehr Runden läuft, als der in der Partie-Konfiguration festgelegter Höchstwert. Wird die Partie durch einen speziellen Überlängenmechanismus einem beschleunigten Ende zugeführt.
BEGRÜNDUNG	Es muss klar definiert sein, wann der Überlängenmechanismus Anwendung findet.
ABHÄNGIGKEITEN	FA81 → p. 31
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12

<b>ID</b>	<b>FA81</b>
TITEL:	Überlängenmechanismus
BESCHREIBUNG:	Durch ein großes Erdbeben werden alle Felsfelder zu Dünenfeldern. Die Sandwurm-Rundenphase wird durch die Shai-Hulud-Rundenphase ersetzt. Falls gerade ein gewöhnlicher Sandwurm unterwegs ist, verschwindet dieser, und taucht nicht mehr auf. Auch die Regel, dass ein Haus unmittelbar verliert, wenn alle von ihm kontrollierten Charaktere verschluckt wurden, wird außer Kraft gesetzt. Sobald in einer Shai-Hulud-Rundenphase der letzte Charakter, der noch auf der Karte stand, verschluckt wird, ist die Partie beendet.
BEGRÜNDUNG	Es muss klar definiert sein, wie der Überlängenmechanismus funktioniert.
ABHÄNGIGKEITEN	FA80 → p. 31
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12
<b>ID</b>	<b>FA82</b>
TITEL:	Shai-Hulud-Rundenphase
BESCHREIBUNG:	In der Shai-Hulud-Rundenphase wählt sich Shai-Hulud einen zufälligen Charakter, und macht unvermittelt einen Sandwurmangriff auf dessen Feld. Der Charakter wird also verschluckt, und ist damit unwiederbringlich aus dem Spiel entfernt.
BEGRÜNDUNG	Es muss klar definiert sein, was in der Shai-Hulud-Rundenphase passiert.
ABHÄNGIGKEITEN	
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12
<b>ID</b>	<b>FA83</b>
TITEL:	Sieg-Metrik überlanger Runden
BESCHREIBUNG:	Nachdem die Partie durch Shai-Hulud beendet wurde, wird der Sieger anhand der Sieg-Metriken ermittelt. Erste Metrik: Welches Haus hat den größten Spice-Vorrat? Zweite Metrik: Welches Haus hat mehr Spicekrümel aufgenommen? Dritte Metrik: Welches Haus hat mehr Charaktere des Gegners besiegt? Vierte Metrik: Bei welchem Haus wurden weniger Charaktere durch gewöhnliche Sandwürmer verschluckt? Fünfte Metrik: Welches Haus hatte den letzten Charakter auf dem Spielbrett?
BEGRÜNDUNG	Es muss klar definiert sein, wie der Sieger ermittelt wird, wenn das Spiel durch den Überlängenmechanismus beendet wird.
ABHÄNGIGKEITEN	FA81 → p. 31, FA82 → p. 32
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Server → p. 11
<b>ID</b>	<b>FA84</b>
TITEL:	Behandlung kein freies Nachbarfeld
BESCHREIBUNG:	Falls kein Nachbarfeld frei ist, wird rekursiv zufällig ein besetztes Feld gesucht und ein zufällig freies Nachbarfeld gesucht.
BEGRÜNDUNG	Es muss klar definiert sein, wie die Behandlung von besetzten Nachbarfelder bei der Suche nach einem freien Nachbarfelde funktioniert.
ABHÄNGIGKEITEN	
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12



ID	FA85
TITEL:	Behandlung von gleichwertigen Alternativen
BESCHREIBUNG:	Liegen zwei gleichwertige Alternativen vor, dann wird zufällig eine der beiden gewählt.
BEGRÜNDUNG	Es muss klar definiert werden, wie zwei gleichwertige Alternativen behandelt werden.
ABHÄNGIGKEITEN	
PRIORITÄT	5
AKTEUR	Server <sup>→ p. 11</sup> , Auftraggeber <sup>→ p. 12</sup>

### 2.3.17 Server

ID	FA86
TITEL:	Starten des Servers über Docker Container
BESCHREIBUNG:	Ein Server muss nicht-interaktiv über die Kommandozeile in Form eines Docker-Containers gestartet werden können. Dabei können in einer vom Standardisierungskomitee definierten Form die nötigen Argumente übergeben werden.
BEGRÜNDUNG	Es muss möglich sein den Server, ohne kompliziertes lokales Build verfahren, mit allen notwendigen Parametern zu starten.
ABHÄNGIGKEITEN	FA4 <sup>→ p. 14</sup>
PRIORITÄT	5
AKTEUR	Auftraggeber <sup>→ p. 12</sup> , Entwickler <sup>→ p. 13</sup> , Teilnehmer <sup>→ p. 12</sup> , Server <sup>→ p. 11</sup>

ID	FA87
TITEL:	Server lädt Partie-Konfiguration
BESCHREIBUNG:	Der Server lädt beim Start eine Partie-Konfiguration. Alle Einstellungen für eine Partie werden in der Partie-Konfiguration gespeichert. Enthaltene Konfigurationsdaten sind bspw. erlaubte Zeitspannen für Aktionen in den Rundenphasen, Rundenanzahl bis zum Eintritt der Überlängenbedingung für eine Partie, etc.
BEGRÜNDUNG	Alle Konfigurationsdaten müssen aus der Partie-Konfiguration gelesen werden.
ABHÄNGIGKEITEN	FA4 <sup>→ p. 14</sup> , FA125 <sup>→ p. 43</sup>
PRIORITÄT	5
AKTEUR	Auftraggeber <sup>→ p. 12</sup> , Entwickler <sup>→ p. 13</sup> , Teilnehmer <sup>→ p. 12</sup> , Server <sup>→ p. 11</sup>

ID	FA88
TITEL:	Server lädt Szenario-Konfiguration
BESCHREIBUNG:	Der Server lädt beim Start eine Szenario-Konfiguration, die das Spielfeld definiert, auf dem die Partie stattfindet.
BEGRÜNDUNG	Die Spielfeldkonfiguration soll aus der Szenario-Konfiguration entnommen werden.
ABHÄNGIGKEITEN	FA4 <sup>→ p. 14</sup> , FA127 <sup>→ p. 43</sup>
PRIORITÄT	5
AKTEUR	Auftraggeber <sup>→ p. 12</sup> , Entwickler <sup>→ p. 13</sup> , Teilnehmer <sup>→ p. 12</sup> , Server <sup>→ p. 11</sup> 127 <sup>→ p. 43</sup>

ID	FA89
TITEL:	Server erlaubt Clientverbindung
BESCHREIBUNG:	Der Server erlaubt genau zwei Clients, sich über das Netzwerk bei ihm für eine Partie als Mitspieler anzumelden.
BEGRÜNDUNG	Der Server muss es genau zwei Clients ermöglichen, sich, als Mitspieler anzumelden.
ABHÄNGIGKEITEN	FA4 <sup>→ p. 14</sup> , FA5 <sup>→ p. 14</sup> , FA6 <sup>→ p. 14</sup>
PRIORITÄT	5
AKTEUR	Auftraggeber <sup>→ p. 12</sup> , Entwickler <sup>→ p. 13</sup> , Teilnehmer <sup>→ p. 12</sup> , Server <sup>→ p. 11</sup>

ID	FA90
TITEL:	Server startet Partie
BESCHREIBUNG:	Sobald sich zwei mitspielende Clients beim Server registriert haben, startet der Server eine Partie und wickelt sie gemäß der Spielregeln ab.
BEGRÜNDUNG	Es muss klar definiert sein, wann der Server die Partie startet.
ABHÄNGIGKEITEN	FA4 → p. 14, FA75 → p. 30
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13, Teilnehmer → p. 12, Server → p. 11

ID	FA91
TITEL:	Server erlaubt Zuschauer
BESCHREIBUNG:	Der Server erlaubt Benutzer-Clients sich als Zuschauer für eine Partie zu registrieren. Sie bekommen dann den aktuellen Spielzustand und zukünftige Updates geschickt.
BEGRÜNDUNG	Es muss klar definiert sein, ob und wie viele Zuschauer Clients der Server zulässt.
ABHÄNGIGKEITEN	FA4 → p. 14, FA102 → p. 36
PRIORITÄT	3
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13, Teilnehmer → p. 12, Server → p. 11

ID	FA92
TITEL:	Server trennt Verbindung
BESCHREIBUNG:	Falls von einem Client erwartet wird, innerhalb einer in der Partie-Konfiguration festgelegten Zeitspanne eine Nachricht an den Server zu schicken, ein Fortgang der Partie ohne diese Nachricht nicht sinnvoll möglich ist, und der Server keine Nachricht rechtzeitig empfängt, soll der Server die Verbindung zum Client abbrechen. Im Falle eines mitspielenden Clients wird dieser disqualifiziert.
BEGRÜNDUNG	Es muss klar definiert sein, unter welchen Bedingungen der Server die Verbindung zum Client trennt.
ABHÄNGIGKEITEN	FA4 → p. 14, QA3 → p. 47
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13, Teilnehmer → p. 12, Server → p. 11

ID	FA93
TITEL:	Server übernimmt Client Input
BESCHREIBUNG:	In Situationen, in denen der Server eine nicht rechtzeitig empfangene Nachricht als Wunsch des Clients interpretieren kann, nichts zu tun, sollte er tolerant sein, und für den Clients ein Default-Verhalten wählen. Damit kann eine Partie manchmal fortgesetzt werden, wenn ein Client kurzzeitig die Verbindung verliert, oder für eine Nachricht zu lange braucht. Der Client verliert lediglich einen Zug, wird aber nicht sofort aus dem Spiel geworfen.
BEGRÜNDUNG	Der Server soll im Falle eines kurzzeitigen Verbindungsabbruchs den Client vertreten, um den Spielfluss in Gang zu halten, und dafür zu sorgen, dass der möglichst weiter spielen kann.
ABHÄNGIGKEITEN	FA4 → p. 14, QA3 → p. 47
PRIORITÄT	3
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13, Teilnehmer → p. 12, Server → p. 11

<b>ID</b>	<b>FA94</b>
TITEL:	Server Erkennung verspäteter Nachrichten
BESCHREIBUNG:	Verspätet eintreffende Nachrichten für eine Rundenphase, die in einer späteren Rundenphase beim Server eingehen, sollten entsprechend vom Server nicht als Protokollverletzung betrachtet, sondern als verspätet erkannt und einfach verworfen werden.
BEGRÜNDUNG	Der Server soll klar definiert wissen, ob eine Client-Nachricht eine Protokollverletzung darstellt oder verspätet ist.
ABHÄNGIGKEITEN	
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13, Teilnehmer → p. 12, Server → p. 11
<b>ID</b>	<b>FA95</b>
TITEL:	Server pausiert Partie
BESCHREIBUNG:	Falls ein mitspielender Benutzer-Client eine Pausierung der Partie wünscht, unterbricht der Server diese, bis irgendeiner der Mitspieler anzeigt, dass er weiterspielen möchte. KI-Clients dürfen keine Pausen verlangen, und auch keine Pausen beenden.
BEGRÜNDUNG	Es muss klar definiert sein, welche Clients eine Pausierung wünschen können und wie diese Pausierung vom Server gehandhabt wird.
ABHÄNGIGKEITEN	
PRIORITÄT	4
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13, Teilnehmer → p. 12, Server → p. 11
<b>ID</b>	<b>FA96</b>
TITEL:	Server hält Session offen
BESCHREIBUNG:	Falls ein mitspielender Client seine Connection zum Server verliert, so bleibt seine Session zunächst bestehen. Der Client kann sich, innerhalb einer gewissen Zeitspanne, erneut mit dem Server verbinden, und seine Session fortsetzen. Der Client bekommt dazu vom Server den vollständigen aktuellen Spielzustand geschickt, für den Fall, dass der Client bspw. neu gestartet werden musste.
BEGRÜNDUNG	Es muss definiert sein, wie der Server mit dem Verbindungsverlust eines Clients umgeht.
ABHÄNGIGKEITEN	
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13, Teilnehmer → p. 12, Server → p. 11
<b>ID</b>	<b>FA97</b>
TITEL:	Server Handhabung von Protokollverletzungen
BESCHREIBUNG:	Falls ein Client sich nicht an das Kommunikationsprotokoll hält, oder eine nach den Spielregeln unzulässige Aktion durchführen will, sendet der Server dem Client eine Nachricht mit einer aussagekräftigen Fehlermeldung, beendet die Verbindung zu ihm und schließt ihn damit vom weiteren Verlauf der Partie aus. Im Fall eines mitspielenden Clients gewinnt dadurch der gegnerische Mitspieler.
BEGRÜNDUNG	Es muss definiert sein, wie der Server darauf reagiert, wenn ein Client sich nicht an das Kommunikationsprotokoll hält.
ABHÄNGIGKEITEN	
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13, Teilnehmer → p. 12, Server → p. 11

<b>ID</b>	<b>FA98</b>
TITEL:	Server informiert Clients über Spielzustand
BESCHREIBUNG:	Der Server informiert alle Clients über die Aktionen der Spieler und die Ereignisse, die sich daraus ergeben haben, und den daraus resultierenden Spielzustand.
BEGRÜNDUNG	Es muss definiert sein, welche Clients wie über den Spielzustand informiert werden.
ABHÄNGIGKEITEN	
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13, Teilnehmer → p. 12, Server → p. 11

<b>ID</b>	<b>FA99</b>
TITEL:	Server überprüft Siegbedingungen
BESCHREIBUNG:	Der Server überprüft zu den relevanten Zeitpunkten, ob ein Spieler gemäß der Siegbedingungen gewonnen hat. Wenn dies der Fall ist, beendet er die Partie und benachrichtigt alle Clients entsprechend.
BEGRÜNDUNG	Der Server muss dazu in der Lage sein zu den relevanten Zeitpunkten die Spielbedingung zu prüfen.
ABHÄNGIGKEITEN	
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13, Teilnehmer → p. 12, Server → p. 11

<b>ID</b>	<b>FA100</b>
TITEL:	Server loggt Partie
BESCHREIBUNG:	Der Server schreibt alle Informationen in eine Logdatei, die zum Nachvollziehen des Partieverlaufs nötig sind. Dadurch können Clients ein Replay der Partie abspielen. Das Standardisierungskomitee legt das Format für Logs bzw. Replay-Dateien fest. Am Ende einer Partie kann das Replay auch direkt den teilnehmenden Clients zugeschickt werden.
BEGRÜNDUNG	Der Server solle alle Informationen in eine Logdatei schreiben, um den Informationsgewinn über den Verlauf der Partie zu optimieren.
ABHÄNGIGKEITEN	
PRIORITÄT	3
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13, Teilnehmer → p. 12, Server → p. 11

### 2.3.18 Benutzer-Client

<b>ID</b>	<b>FA101</b>
TITEL:	Registrierung als Spieler
BESCHREIBUNG:	Ein menschlicher Teilnehmer muss sich über den Benutzer-Client beim Server als Spieler registrieren können. Diese Registrierung gilt für genau eine Partie.
BEGRÜNDUNG	Der Server muss wissen, dass ein Spieler eine Partie spielen möchte
ABHÄNGIGKEITEN	FA103 → p. 37, FA104 → p. 37
PRIORITÄT	5
AKTEUR	Spieler → p. 12, Auftraggeber → p. 12

<b>ID</b>	<b>FA102</b>
<b>TITEL:</b>	Registrierung als Zuschauer
<b>BESCHREIBUNG:</b>	Ein menschlicher Teilnehmer muss sich über den Benutzer-Client beim Server als Zuschauer registrieren können. Diese Registrierung gilt für genau eine Partie, die entweder noch ausstehend ist oder schon läuft.
<b>BEGRÜNDUNG</b>	Der Server muss wissen, dass ein Benutzer bei einer Partie zuschauen will (auch wenn erst später einsteigen will) und ihm dann auch den Spielstand aktualisiert immer zusenden.
<b>ABHÄNGIGKEITEN</b>	FA103→ p. 37, FA104→ p. 37
<b>PRIORITÄT</b>	5
<b>AKTEUR</b>	Zuschauer→ p. 12, Auftraggeber→ p. 12

<b>ID</b>	<b>FA103</b>
<b>TITEL:</b>	Verbindungsaufbau zum Server
<b>BESCHREIBUNG:</b>	Der Benutzer-Client muss sich mit einem Server verbinden können. Dabei müssen sie dem Server die Rolle mitteilen, das heißt ob sie ein Spieler sind oder nicht, sowie ihren Namen.
<b>BEGRÜNDUNG</b>	Der Server verwaltet die Partien und Teilnehmer, deswegen muss ein Benutzer-Client sich damit verbinden, um an einer Partie teilnehmen oder sie beobachten zu können. Außerdem muss der Server wissen, ob es sich bei dem Benutzer um einen Spieler handelt, um die entsprechenden Rechte zu vergeben. Der Name ist für den Server wichtig, damit man die Clients auf der Benutzeroberfläche identifizieren kann
<b>ABHÄNGIGKEITEN</b>	FA101→ p. 36, FA102→ p. 36, FA104→ p. 37
<b>PRIORITÄT</b>	5
<b>AKTEUR</b>	Spieler→ p. 12, Auftraggeber→ p. 12, Server→ p. 11

<b>ID</b>	<b>FA104</b>
<b>TITEL:</b>	Graphische Benutzeroberfläche
<b>BESCHREIBUNG:</b>	Der Benutzer-Client hat eine graphische Benutzeroberfläche, über die dem menschlichen Benutzer alle Daten zum Spiel angezeigt werden können und über die der Spieler mit dem Spiel interagieren, das heißt seine Züge ausführen kann.
<b>BEGRÜNDUNG</b>	Der menschliche Benutzer muss das Spiel sehen können, da er im Gegensatz zu Computern nicht gut mit rohen Daten umgehen kann, sondern lieber die Daten visualisiert verwertet. Außerdem macht ein Spiel mit einer graphische Oberfläche und einem schönen Design mehr Spaß. Zudem braucht der Spieler eine Möglichkeit zur Eingabe seiner Züge.
<b>ABHÄNGIGKEITEN</b>	FA105→ p. 37, FA106→ p. 38, FA107→ p. 38, FA108→ p. 38, FA110→ p. 39, FA111→ p. 39, FA113→ p. 40, FA109→ p. 39
<b>PRIORITÄT</b>	5
<b>AKTEUR</b>	Auftraggeber→ p. 12, menschlicher Benutzer→ p. 11

<b>ID</b>	<b>FA105</b>
<b>TITEL:</b>	Visualisierung Spielgeschehen
<b>BESCHREIBUNG:</b>	Der Benutzer-Client muss das Spielgeschehen auf der graphischen Benutzeroberfläche visualisieren können. Das bedeutet, dass der Benutzer-Client die Karte beziehungsweise das Spielfeld anzeigen muss, sowie alle sich darauf befindenen Charaktere. Die Charaktere müssen so angezeigt werden, dass sie auf dem Spielfeld unterschieden werden können, durch zum Beispiel unterschiedliche Avatare oder Namen.
<b>BEGRÜNDUNG</b>	Der Benutzer muss den aktuellen Spielstand sehen können und die Charaktere auf dem Spielfeld unterscheiden können. Ohne Unterscheidung oder Anzeige des Spielstandes, kann der Benutzer den aktuellen Spielstand nicht (korrekt) wahrnehmen und damit dem Spiel nicht folgen oder Züge basierend auf dem aktuellen Stand ausführen.
<b>ABHÄNGIGKEITEN</b>	FA104 → p. 37, FA110 → p. 39
<b>PRIORITÄT</b>	5
<b>AKTEUR</b>	menschlicher Benutzer → p. 11, Auftraggeber → p. 12
<b>ID</b>	<b>FA106</b>
<b>TITEL:</b>	Visualisierung Spielstatus
<b>BESCHREIBUNG:</b>	Der Benutzer-Client muss alle spiel-relevanten Informationen, das heißt den Status des Spiels, visualisieren. Das heißt auf der graphischen Benutzeroberfläche müssen die Werte und die Zustände der Charaktere angezeigt werden. Zu den spielrelevanten Informationen zählen mindestens: <ul style="list-style-type: none"> <li>▪ Health-Bar der Charakter</li> <li>▪ das Inventar und gesammelte Spice</li> <li>▪ aktuelle Phase</li> <li>▪ Spielerinformationen, wie MP, AP oder Angriffsschaden</li> </ul>
<b>BEGRÜNDUNG</b>	Der Benutzer müssen den aktuellen Status des Spiels sehen, um zu wissen welche Aktionen möglich oder sinnvoll sind und dem Spielverlauf als Zuschauer zu folgen oder als Spieler fundierte Entscheidungen für eine Aktion treffen.
<b>ABHÄNGIGKEITEN</b>	FA104 → p. 37, FA110 → p. 39
<b>PRIORITÄT</b>	5
<b>AKTEUR</b>	menschlicher Benutzer → p. 11, Auftraggeber → p. 12
<b>ID</b>	<b>FA107</b>
<b>TITEL:</b>	Visualisierung Aktionen
<b>BESCHREIBUNG:</b>	Der Benutzer-Client muss die Aktionen des Spielers visualisieren. Das bedeutet, dass Angriffe animiert werden müssen und zwar so, dass man sieht, welcher Charakter welchen anderen angegriffen hat und wie viel Schaden er ihm zugefügt hat. Weiterhin sollen alle Veränderungen auf dem Spielfeld, wie zum Beispiel eine Family Atomic oder die Bewegung des Sandwurms animiert werden.
<b>BEGRÜNDUNG</b>	Der Benutzer muss sehen können, welche Aktion gerade ausgeführt wird und welche Veränderungen dadurch auf dem Spielfeld entstehen. Außerdem ist eine Animation von Zügen ein gutes Benutzererlebnis.
<b>ABHÄNGIGKEITEN</b>	FA104 → p. 37, FA110 → p. 39
<b>PRIORITÄT</b>	5
<b>AKTEUR</b>	menschlicher Benutzer → p. 11, Auftraggeber → p. 12

<b>ID</b>	<b>FA108</b>
<b>TITEL:</b>	Visualisierung Phasen
<b>BESCHREIBUNG:</b>	Der Benutzer-Client muss die Abwicklung der einzelnen Runden- und Zugphasen animieren. Dabei muss die Dauer der Animation an die vorgegebene Dauer für die Phase in der Partiekonfiguration angepasst werden.
<b>BEGRÜNDUNG</b>	Der Benutzer möchte sehen, in welcher Phase er sich gerade befindet. Außerdem zeigt eine Animation der Phasen den Spielverlauf besser an und stellt ein gutes Benutzererlebnis dar.
<b>ABHÄNGIGKEITEN</b>	FA104 → p. 37, FA110 → p. 39, FA125 → p. 43
<b>PRIORITÄT</b>	5
<b>AKTEUR</b>	menschlicher Benutzer → p. 11, Auftraggeber → p. 12

<b>ID</b>	<b>FA109</b>
<b>TITEL:</b>	Visualisierung Spielausgang
<b>BESCHREIBUNG:</b>	Am Ende der Partie soll der Benutzer-Client dem Benutzer den Gewinner anzeigen. Optional sollen noch interessante Statistiken angezeigt werden können, wie zum Beispiel in welcher Phase man dem Gegner wie viel Schaden zugefügt hat.
<b>BEGRÜNDUNG</b>	Der Benutzer möchte sehen, wer gewonnen hat und kann sich mit Hilfe der Statistiken eventuell verbessern.
<b>ABHÄNGIGKEITEN</b>	FA104 → p. 37, FA110 → p. 39
<b>PRIORITÄT</b>	4
<b>AKTEUR</b>	menschlicher Benutzer → p. 11, Auftraggeber → p. 12

<b>ID</b>	<b>FA110</b>
<b>TITEL:</b>	Visualisierungsdauer
<b>BESCHREIBUNG:</b>	Die Animation von bestimmten Aktionen oder Phasen darf nur eine in der Partiekonfiguration festgesetzte Zeit dauern.
<b>BEGRÜNDUNG</b>	Wenn die Animationen zu lange dauern, dann kommt das Spiel dem Benutzer ruckelig vor und stört den Spielfluss. Außerdem sehen lange Animationen nicht schön aus und nerven den Benutzer.
<b>ABHÄNGIGKEITEN</b>	FA104 → p. 37, FA125 → p. 43
<b>PRIORITÄT</b>	5
<b>AKTEUR</b>	menschlicher Benutzer → p. 11, Auftraggeber → p. 12

<b>ID</b>	<b>FA111</b>
<b>TITEL:</b>	Ermöglichung Spielinteraktion
<b>BESCHREIBUNG:</b>	Der Benutzer soll über die graphische Benutzeroberfläche des Benutzer-Clients die Möglichkeit haben, mit dem Spiel zu interagieren. Das bedeutet, dass der Spieler über die Oberfläche Aktionen vornehmen kann, die der Client an den Server sendet. Dabei muss der Benutzer-Client gewährleisten, dass nur regelkonforme Eingaben getätigt werden können.
<b>BEGRÜNDUNG</b>	Der Spieler muss irgendwie mit dem Spiel interagieren können und seine Züge dem Server mitteilen. Damit der Server nicht abstürzt dürfen nur valide Züge an ihn gesendet werden.
<b>ABHÄNGIGKEITEN</b>	FA104 → p. 37
<b>PRIORITÄT</b>	5
<b>AKTEUR</b>	Spieler → p. 12, Auftraggeber → p. 12

<b>ID</b>	<b>FA112</b>
<b>TITEL:</b>	Vereinfachung Spielinteraktion
<b>BESCHREIBUNG:</b>	Der Benutzer-Client soll die Möglichkeit haben, dem Benutzer mit Hilfe von Hotkeys die Interaktion zu erleichtern. Damit kann der Spieler bestimmte Aktionen auch über Tastenkombinationen ausführen.
<b>BEGRÜNDUNG</b>	Hotkeys können dem Spieler die Bedienung des Spiels komfortabler gestalten.
<b>ABHÄNGIGKEITEN</b>	FA104 → p. 37, FA111 → p. 39
<b>PRIORITÄT</b>	2
<b>AKTEUR</b>	Spieler → p. 12, Auftraggeber → p. 12

<b>ID</b>	<b>FA113</b>
<b>TITEL:</b>	Antrag auf Pausierung und Spielfortsetzung
<b>BESCHREIBUNG:</b>	Der Spieler muss über den Benutzer-Client dem Server einen Antrag auf Pause oder Wiederaufnahme der Partie senden können. Der Antrag auf Wiederaufnahme kann nur geschehen, wenn ein Spiel pausiert ist und ein Spiel kann nur pausiert werden, wenn nicht schon pausiert ist.
<b>BEGRÜNDUNG</b>	Der Spieler soll die Möglichkeit haben, ein Spiel anzuhalten und fortzusetzen, weil er zwischenzeitlich etwas anderes erledigen muss, das Spiel aber nicht beenden will.
<b>ABHÄNGIGKEITEN</b>	FA104 → p. 37
<b>PRIORITÄT</b>	5
<b>AKTEUR</b>	Spieler → p. 12, Auftraggeber → p. 12

<b>ID</b>	<b>FA114</b>
<b>TITEL:</b>	Abspielen eines Replay
<b>BESCHREIBUNG:</b>	Der Benutzer kann mit Hilfe einer Log-Datei vom Server ein Replay laden und somit eine gespielte Partie nochmal anschauen
<b>BEGRÜNDUNG</b>	Der Spieler möchte vielleicht alte Partien analysieren und sich so verbessern oder sehen, wie andere Spieler spielen, auch wenn er nicht live die Partie folgen konnte. Für die Entwicklern kann ein Replay beim Debuggen helfen, um so sehen, wie gut der KI-Client spielt.
<b>ABHÄNGIGKEITEN</b>	FA104 → p. 37, FA100 → p. 36
<b>PRIORITÄT</b>	2
<b>AKTEUR</b>	menschlicher Benutzer → p. 11, Auftraggeber → p. 12, Entwickler → p. 13

### 2.3.19 KI-Client

<b>ID</b>	<b>FA115</b>
<b>TITEL:</b>	Start des KI-Client
<b>BESCHREIBUNG:</b>	Der KI-Client muss über die Kommandozeile mit Hilfe eines Docker-Containers gestartet werden können. Dafür werden ihm die von dem Standardisierungskomitee nötigen Argumente übergeben.
<b>BEGRÜNDUNG</b>	Der KI-Client braucht keine graphische Oberfläche und der Start über die Kommandozeile als Docker-Container erleichtert den Build- und Startprozess
<b>ABHÄNGIGKEITEN</b>	FA116 → p. 40
<b>PRIORITÄT</b>	5
<b>AKTEUR</b>	Auftraggeber → p. 12



<b>ID</b>	<b>FA116</b>
TITEL:	Verbindungsaufbau zum Server
BESCHREIBUNG:	Mit den übergebenen Argumenten beim Start des KI-Clients, wie zum Beispiel der IP-Adresse und dem Port des Servers, baut der KI-Client eine Verbindung zum Server auf und teilt diesem mit, dass dieser Client eine KI ist und welchen Namen der KI-Client hat. Danach kann er einer Partie beitreten und nicht mehr mit dem Benutzer kommunizieren
BEGRÜNDUNG	Um an einer Partie teilnehmen, muss sich auch ein KI-Client mit dem Server verbinden. Außerdem muss der Server wissen, dass es sich um eine KI handelt, da diese andere Regeln hat, als ein Spieler (zum Beispiel keine Möglichkeit der Pausierung)
ABHÄNGIGKEITEN	FA117 → p. 41, FA115 → p. 40
PRIORITÄT	5
AKTEUR	Server → p. 11, Auftraggeber → p. 12
<b>ID</b>	<b>FA117</b>
TITEL:	Pausierung KI-Client
BESCHREIBUNG:	KI-Clients dürfen keine Pausierung der Partie bei dem Server anfragen. Jedoch muss dieser Client mit einer Pausierung, die beliebig lang sein kann, von einem Benutzer-Client umgehen können. Das heißt sie müssen den Fall behandeln, dass die Partie keinen Fortschritt machen und einen Wartezustand einnehmen
BEGRÜNDUNG	KI-Client müssen keine Pause einlegen, jedoch sollte der Spieler die Möglichkeit haben, ein Spiel anzuhalt. Damit muss dann der KI-Client umgehen können, damit das Spiel nicht abbricht.
ABHÄNGIGKEITEN	
PRIORITÄT	5
AKTEUR	Spieler → p. 12, Auftraggeber → p. 12
<b>ID</b>	<b>FA118</b>
TITEL:	Zugwahl KI-Client
BESCHREIBUNG:	In jeder Zug- und Rundenphase muss der KI-Client regelkonforme und nach Möglichkeit sinnvolle Züge beziehungsweise Aktionen auswählen. Diese Wahl wird von einem Algorithmus getroffen und darf nur eine gewisse Zeit andauern. Danach muss der KI-Client die Aktion dem Server über das definierte Protokoll mitteilen.
BEGRÜNDUNG	Der KI-Client muss auch Züge machen, die jedoch nicht zu lange dauern sollten, damit der Spieler nicht ewig warten muss. Außerdem sollten die Züge auch regelkonform sein, damit der Server die Züge auch ausführen kann und nicht abstürzt, sowie sinnvoll, damit der andere Spieler auch gegen einen Gegner spielt, bei dem Spiel Spaß macht.
ABHÄNGIGKEITEN	FA119 → p. 41, FA120 → p. 41
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Server → p. 11, Spieler → p. 12
<b>ID</b>	<b>FA119</b>
TITEL:	Zugdauer KI-Client
BESCHREIBUNG:	Der KI-Client hat nur eine in der Konfiguration festgesetzte Zeitspanne, um einen Zug auszuführen.
BEGRÜNDUNG	Das Spiel soll nicht ewig dauern und der andere Spieler will ebenso Züge machen. Über eine variable Länge kann man jedoch die Schwierigkeit der KI anpassen
ABHÄNGIGKEITEN	FA118 → p. 41, FA125 → p. 43
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Server → p. 11, Spieler → p. 12

<b>ID</b>	<b>FA120</b>
<b>TITEL:</b>	Anpassung Intelligenzstufen
<b>BESCHREIBUNG:</b>	<p>Der KI-Client soll verschiedene Intelligenzstufen haben, die in der Partiekonfiguration eingestellt werden. Dabei entspricht eine Intelligenzstufe dem Schwierigkeit der KI, der vorher von den Entwicklern individualisiert wurden. Es wird die folgenden Schwierigkeitsgrade geben:</p> <ul style="list-style-type: none"> <li>▪ Einsteiger</li> <li>▪ Amateur</li> <li>▪ Halbprofi</li> <li>▪ Profi</li> <li>▪ Weltklasse</li> </ul>
<b>BEGRÜNDUNG</b>	Durch die unterschiedlichen Schwierigkeitsgrade kann der Spieler das Spiel individualisieren. Dadurch kann man leicht in das Spiel einsteigen, kann sich steigern und hat lange eine Herausforderung, aber keine Überforderung.
<b>ABHÄNGIGKEITEN</b>	FA118 → p. 41
<b>PRIORITÄT</b>	5
<b>AKTEUR</b>	Spieler → p. 12, Auftraggeber → p. 12
<b>ID</b>	<b>FA121</b>
<b>TITEL:</b>	Einbindung in Benutzer-Client
<b>BESCHREIBUNG:</b>	Der KI-Client soll auch in den Benutzer-Client einbindbar sein, das heißt aus dem Benutzer-Client heraus gestartet werden.
<b>BEGRÜNDUNG</b>	Der Benutzer möchte vielleicht nicht selber spielen, sondern ein Spiel nur beobachten. Eine Einbindung in den Benutzer-Client ermöglicht den Start eines KI-Clients auch für menschliche Benutzer, die nicht wissen, wie man den KI-Client über die Kommandozeile startet
<b>ABHÄNGIGKEITEN</b>	FA115 → p. 40
<b>PRIORITÄT</b>	5
<b>AKTEUR</b>	menschlicher Benutzer → p. 11

### 2.3.20 Editor

<b>ID</b>	<b>FA122</b>
<b>TITEL:</b>	Konfiguration über graphische Benutzeroberfläche
<b>BESCHREIBUNG:</b>	<p>Der Benutzer kann über die graphische Benutzeroberfläche des Editors Konfigurationen vornehmen. Dabei gibt es die folgenden Konfigurationen:</p> <ul style="list-style-type: none"> <li>▪ Partiekonfiguration (siehe FA125 → p. 43)</li> <li>▪ Szenariokonfiguration (siehe FA127 → p. 43)</li> </ul> <p>Diese müssen in dem Editor erzeugt und bearbeitet werden können.</p>
<b>BEGRÜNDUNG</b>	Die Teilnehmer sollen das Spiel individualisieren können, um zum Beispiel verschiedene Schwierigkeitslevel zu erstellen. Außerdem ermöglicht eine ausgelagerte Konfiguration, dass der Entwickler spiel-relevante Daten nicht festlegt, sondern diese später leicht auch von Teilnehmern angepasst werden können.
<b>ABHÄNGIGKEITEN</b>	FA125 → p. 43, FA127 → p. 43
<b>PRIORITÄT</b>	5
<b>AKTEUR</b>	Auftraggeber → p. 12, Entwickler → p. 13, Teilnehmer → p. 12, Server → p. 11

<b>ID</b>	<b>FA123</b>
TITEL:	Format der Konfigurationen
BESCHREIBUNG:	Die Szenario- und Partiekonfigurationen müssen in einem von dem Standardisierungskomitee definierten JSON-Schema erstellt werden. Die Konfiguration werden in diesem Format in Dateien gespeichert und können aus diesen Daten zur weiteren Bearbeitung geladen werden.
BEGRÜNDUNG	Zur späteren Validierung und Verarbeitung durch den Server (Server wird unabhängig von Editor und Client entwickelt) müssen die Konfigurationen einem Schema folgen, wobei sich JSON als weitverbreitetes Format anbietet.
ABHÄNGIGKEITEN	FA122 → p. 42, FA124 → p. 43
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13, Spieler → p. 12, Server → p. 11
<b>ID</b>	<b>FA124</b>
TITEL:	Laden und Modifikation von Konfigurationen
BESCHREIBUNG:	Der Spieler soll im Editor erstellte Konfigurationen laden oder abspeichern können. Die geladenen Konfigurationen müssen validiert werden.
BEGRÜNDUNG	Der Spieler soll die Möglichkeit haben beliebige Konfigurationen wiederverwenden zu können, ohne sie immer neu erstellen zu müssen. Trotzdem soll der Server keine ungültigen Daten bekommen, weswegen die Konfigurationen validiert werden müssen
ABHÄNGIGKEITEN	FA122 → p. 42, FA123 → p. 42
PRIORITÄT	3
AKTEUR	Entwickler → p. 13, Spieler → p. 12
<b>ID</b>	<b>FA125</b>
TITEL:	Partiekonfiguration
BESCHREIBUNG:	Der Benutzer kann in dem Editor Konfigurationen zur Partie vornehmen. Damit kann er Einstellungen zu einer bestimmten Partie vornehmen.
BEGRÜNDUNG	Der Spieler soll Einstellungen für die Partie vornehmen können, um die Partien zu individualisieren und die Länge der Partie beeinflussen zu können
ABHÄNGIGKEITEN	FA122 → p. 42, FA123 → p. 42
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13, Spieler → p. 12, Server → p. 11
<b>ID</b>	<b>FA126</b>
TITEL:	Parameter der Partiekonfiguration
BESCHREIBUNG:	Der Spieler kann mindestens folgende Punkte bezüglich einer Partie einstellen: <ul style="list-style-type: none"> <li>▪ Zeitdauer der verschiedenen Phasen</li> <li>▪ maximale Rundenanzahl, bis die Bedingung für überlange Runden eintritt</li> </ul>
BEGRÜNDUNG	Diese Parameter müssen definiert werden.
ABHÄNGIGKEITEN	FA125 → p. 43, FA126 → p. 43
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13, Spieler → p. 12, Server → p. 11

<b>ID</b>	<b>FA127</b>
<b>TITEL:</b>	Szenariokonfiguration
<b>BESCHREIBUNG:</b>	Der Benutzer kann in dem Editor Konfigurationen zu den Szenarios vornehmen. Damit kann er Einstellungen zu den Szenarios vornehmen, das heißt das Spielfeld konfigurieren und weitere Kenngrößen des Spiels anpassen.
<b>BEGRÜNDUNG</b>	Der Spieler soll das Spielfeld und andere Kenngrößen des Spiels anpassen sollen, um die Partien zu individualisieren und die Schwierigkeit oder das Design des Spielfeldes selbst zu bestimmen.
<b>ABHÄNGIGKEITEN</b>	FA122 → p. 42, FA128 → p. 44, FA129 → p. 44, FA123 → p. 42
<b>PRIORITÄT</b>	5
<b>AKTEUR</b>	Auftraggeber → p. 12, Entwickler → p. 13, Spieler → p. 12, Server → p. 11

<b>ID</b>	<b>FA128</b>
<b>TITEL:</b>	Parameter der Szenariokonfiguration
<b>BESCHREIBUNG:</b>	Der Spieler kann mindestens folgende Punkte bezüglich eines Szenarios einstellen: <ul style="list-style-type: none"> <li>▪ soll ein Spielfeld automatisch generiert werden (siehe FA129 → p. 44)</li> <li>▪ Dimension des Spielfelds</li> <li>▪ Position und Größe der Städte, sowie die Landschaft, das heißt wo Gebirge oder Wüste ist.</li> <li>▪ Darstellung, das heißt bestimmte Farben oder Texturen der Charaktere</li> </ul>
<b>BEGRÜNDUNG</b>	Diese Parameter müssen definiert werden.
<b>ABHÄNGIGKEITEN</b>	FA127 → p. 43
<b>PRIORITÄT</b>	5
<b>AKTEUR</b>	Auftraggeber → p. 12, Entwickler → p. 13, Spieler → p. 12, Server → p. 11

<b>ID</b>	<b>FA129</b>
<b>TITEL:</b>	zufällige Szenariokonfiguration
<b>BESCHREIBUNG:</b>	Der Spieler soll im Editor die Möglichkeit haben, ein Szenario zufällig generieren zu lassen. Dabei soll der Spieler ebenso die Möglichkeit haben, in den Generierungsprozess einzugreifen und zum Beispiel die minimale und maximale Dimension des Spielfeldes einstellen oder im Nachhinein das Spielfeld anpassen können.
<b>BEGRÜNDUNG</b>	Die Konfiguration eines Spielfeldes ist aufwendig und erfordert Erfahrung. Eine automatische Generierung oder Konfiguration erleichtert den Spieleinstieg und spart Zeit. Jedoch soll trotzdem noch nachjustiert werden können.
<b>ABHÄNGIGKEITEN</b>	FA127 → p. 43
<b>PRIORITÄT</b>	3
<b>AKTEUR</b>	Entwickler → p. 13, Spieler → p. 12

### 3 Softwarespezifikation

#### 3.1 Domänenmodell

Dieser Abschnitt enthält das Domänenmodell, welches in fünf kleinere Modelle unterteilt wurde. Zu Beginn die Gesamtstruktur und darauf folgend die zugehörigen Teilmodelle.

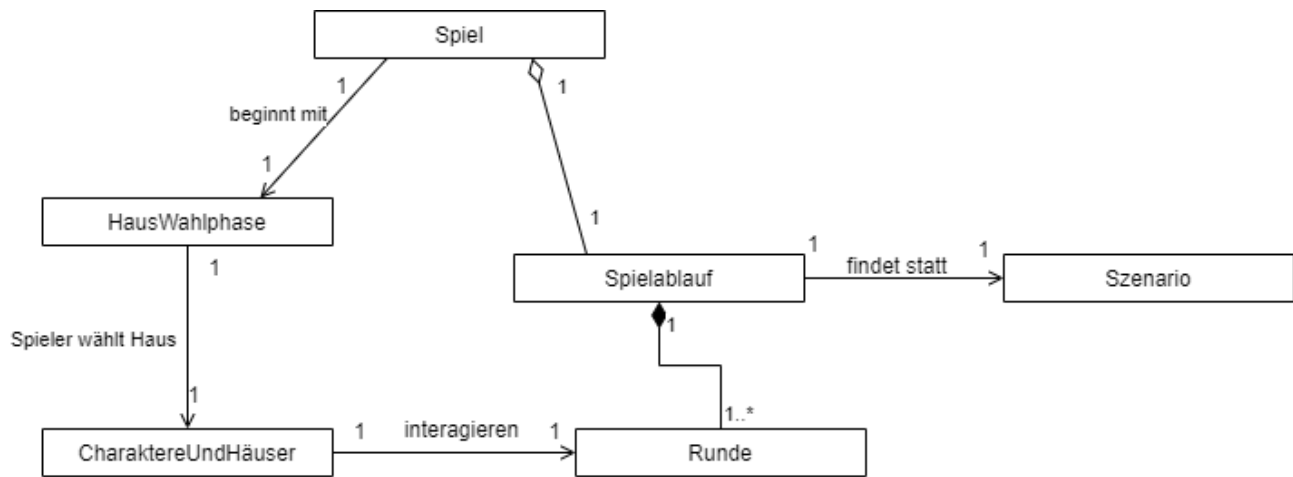


Abbildung 1: Domänenemodell: Gesamtstruktur (enthält Teilmodel: 2, 3, 4)

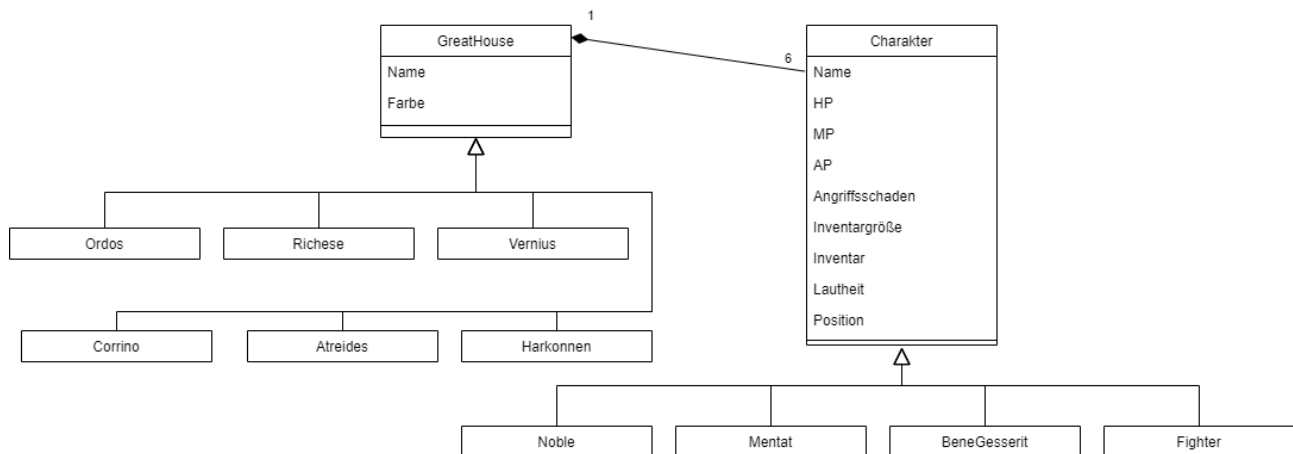


Abbildung 2: Teilmodell: Häuser und Charaktere

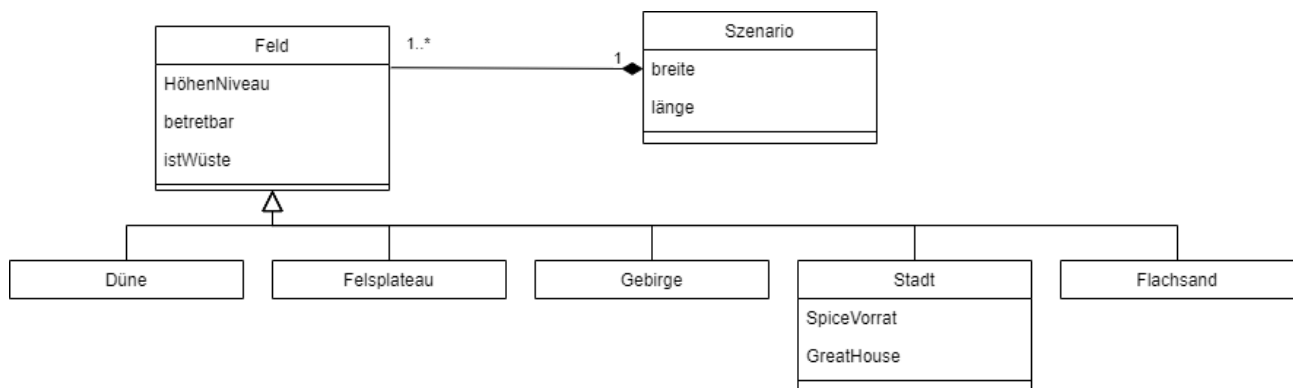
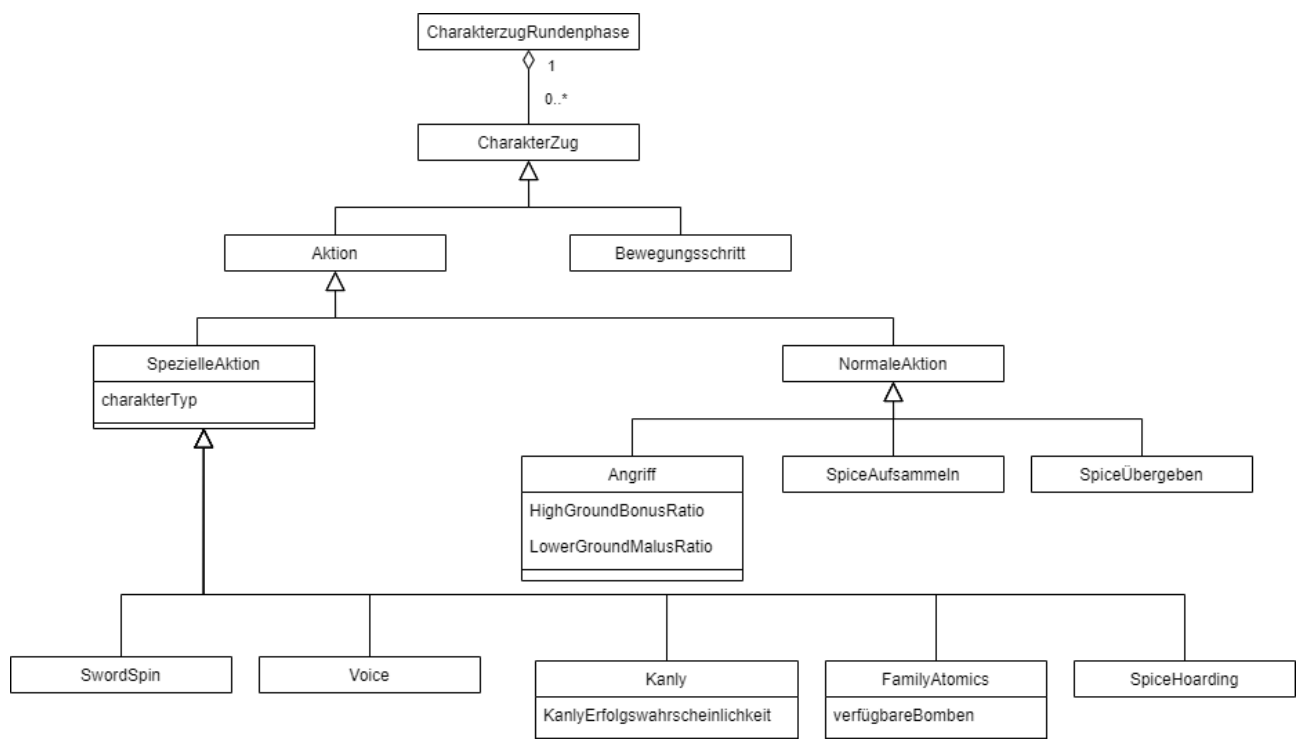
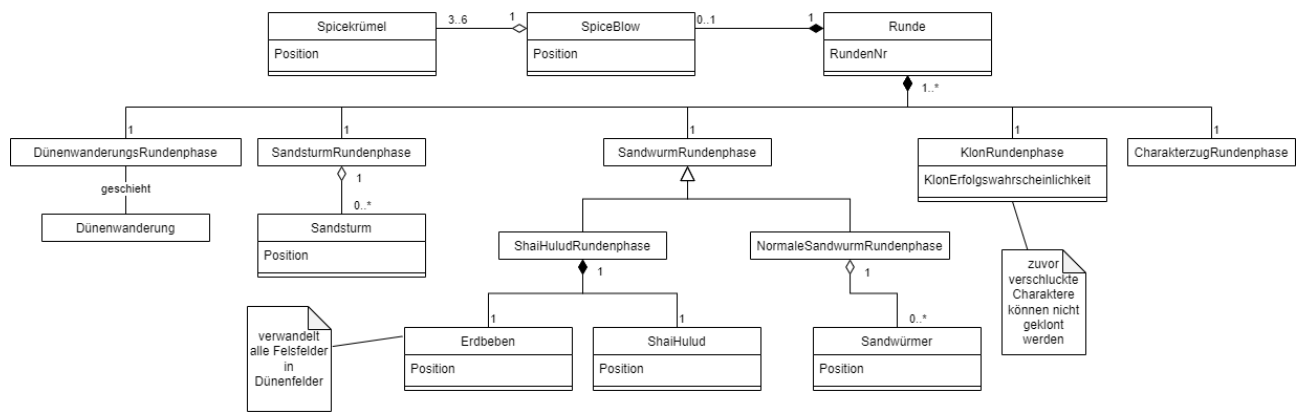


Abbildung 3: Teilmodell: Szenario



## 4 Randbedingungen

### 4.1 Qualität

Es müssen folgende nicht-funktionalen Anforderungen an das gesamte System und die Entwickler erfüllt werden:

<b>ID</b>	<b>QA1</b>
TITEL:	Robustheit
BESCHREIBUNG:	Das gesamte System darf bei 100 Partien maximal einmal abstürzen. Das gesamte System stürzt genau dann ab, wenn eine Komponente des Systems abstürzt, das heißt aufgrund eines nicht behandelten Fehlers beendet wird.
BEGRÜNDUNG	Das gesamte System soll robust sein, das heißt stabil laufen, weil der Benutzer spielen möchte und nicht ständig das Spiel neu starten oder warten will. Zudem nervt ein nicht stabiles System den Benutzer und vertreibt ihn damit möglicherweise.
ABHÄNGIGKEITEN	
PRIORITÄT	2
AKTEUR	Teilnehmer → p. 12
<b>ID</b>	<b>QA2</b>
TITEL:	Zuverlässigkeit des Servers
BESCHREIBUNG:	Der Server soll zuverlässig laufen. Das heißt, er soll eine Uptime von mindestens 95% haben und innerhalb von 1 Minute neu gestartet werden können.
BEGRÜNDUNG	Der Server ist für die Verwaltung der Spieler und der Partie zuständig und ist deshalb ein integraler Bestandteil der Anwendung. Daher sollte er zuverlässig und verfügbar sein.
ABHÄNGIGKEITEN	
PRIORITÄT	2
AKTEUR	Teilnehmer → p. 12
<b>ID</b>	<b>QA3</b>
TITEL:	Toleranz des Servers
BESCHREIBUNG:	Der Server soll bei Verbindungsabbrüchen oder zu spät gesendeten oder nicht interpretierbaren Nachrichten des Clients tolerant sein. Das bedeutet, der Server schließt die Session bei einem Verbindungsabbruch nicht sofort. Außerdem wählt der Server bei zu spät gesendeten Nachrichten oder nicht interpretierbaren Nachrichten ein Standardverhalten. Der Client soll jedoch ebenfalls versuchen, das Problem zu beheben und zum Beispiel einen erneuten Verbindungsaufbau zu initiieren.
BEGRÜNDUNG	Durch das tolerante Verhalten des Servers kann die Partie auch bei Problemen des Clients fortgesetzt werden und wird nicht jedes Mal abgebrochen, wenn zum Beispiel bei dem Client das WLAN ausfällt. Dies trägt zur Robustheit der Anwendung bei.
ABHÄNGIGKEITEN	QA1 → p. 47
PRIORITÄT	4
AKTEUR	Benutzer-Client → p. 11, Teilnehmer → p. 12

<b>ID</b>	<b>QA4</b>
TITEL:	Plattformen
BESCHREIBUNG:	Die Komponenten <i>Benutzer-Client</i> und <i>Editor</i> müssen auf einer aktuellen Version von Microsoft Windows oder einer aktuellen Debian Linux-Distribution (zum Beispiel Ubuntu) lauffähig sein. Die Komponenten <i>KI-Client</i> und <i>Server</i> müssen inklusive ihrer Abhängigkeit als Docker-Container lauffähig sein.
BEGRÜNDUNG	Der <i>Benutzer-Client</i> und der <i>Editor</i> soll auf mehreren Betriebssystemen lauffähig sein, um eine möglichst große Zielgruppe zu erreichen und keine Anforderung an den Benutzer zu stellen. <i>KI-Client</i> und <i>Server</i> sollen als Docker-Container lauffähig sein, weil der Benutzer nicht direkt damit interagiert und die Docker-Images ohne komplizierte lokale Build-Prozesse direkt gestartet werden können. Damit erreicht man eine Unabhängigkeit von der Plattform und kann somit die Komponenten einfacher zum Laufen bekommen.
ABHÄNGIGKEITEN	QA5 → p. 48
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12
<b>ID</b>	<b>QA5</b>
TITEL:	Portierbarkeit
BESCHREIBUNG:	Es muss möglich sein, die Anwendung auf andere Betriebssysteme zu portieren. Vielleicht stellt sich nach der Entwicklung heraus, dass die Anwendung auch auf anderen Plattformen laufen soll, weil eine Gruppe von Benutzern nicht die angebotenen Betriebssysteme nicht nutzen will.
BEGRÜNDUNG	Die Anwendung soll portierbar sein, weil sich nach der Entwicklung herausstellen könnte, dass die Benutzer ein anderes Betriebssystem bevorzugen und daher das Spiel auf dieser Plattform nutzen wollen.
ABHÄNGIGKEITEN	QA4 → p. 47
PRIORITÄT	1
AKTEUR	Entwickler → p. 13, Teilnehmer → p. 12
<b>ID</b>	<b>QA6</b>
TITEL:	einfache Installation
BESCHREIBUNG:	Die Anwendung sollte mithilfe eines Skripts oder einem Programm installiert werden können. Dieses Skript oder Programm nimmt dem Benutzer die typischen Aufgaben beim Installationsprozess, wie zum Beispiel das Herunterladen und Installieren von Abhängigkeiten, ab.
BEGRÜNDUNG	Der Benutzer möchte das Spiel spielen und nicht die Zeit damit verbringen, das Spiel zunächst zum Laufen zu bekommen. Ein aufwendiger Installationsprozess könnte den Benutzer frustrieren.
ABHÄNGIGKEITEN	QA4 → p. 47
PRIORITÄT	1
AKTEUR	Teilnehmer → p. 12
<b>ID</b>	<b>QA7</b>
TITEL:	Testbarkeit
BESCHREIBUNG:	Die Anwendung soll mit entsprechenden Tests, insbesondere mit Unit-Tests, geprüft werden. Diese Tests sollen zu 100% erfüllt werden und mindestens 80% des nicht automatisch generierten Source Codes abdecken. Diese Tests beziehen sich nicht auf die Benutzerschnittstelle. Zudem sollen die Tests automatisiert bei jedem Commit auf dem <i>release</i> -Branch ausgeführt werden.
BEGRÜNDUNG	Die Verwendung von Tests kann die Entwicklung des Codes vereinfachen, weil dann bei der Implementierung durch die Tests ein genaues Verständnis vorliegen muss, was der Code machen soll. Außerdem lassen sich so Fehler oder Bugs frühzeitig finden und die Qualität des Codes steigt.
ABHÄNGIGKEITEN	QA1 → p. 47, QA2 → p. 47
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13



<b>ID</b>	<b>QA8</b>
TITEL:	Bedienbarkeit
BESCHREIBUNG:	Die Anwendung soll einfach und intuitiv sein. Das bedeutet, dass die Anwendung mithilfe des Benutzerhandbuchs innerhalb von 2 Stunden vom Benutzer bedient werden kann.
BEGRÜNDUNG	Durch eine einfache Bedienbarkeit kann sich der Teilnehmer ganzheitlich auf das Spiel konzentrieren und wird nicht durch ein unübersichtliches Design vom Spiel abgehalten oder frustriert
ABHÄNGIGKEITEN	QA13 → p. 49
PRIORITÄT	3
AKTEUR	Teilnehmer → p. 12
<b>ID</b>	<b>QA9</b>
TITEL:	Programmiersprache
BESCHREIBUNG:	Für die Entwicklung der Anwendung wird hauptsächlich die Programmiersprache JAVA 11 verwendet. Die Bibliotheken sind frei wählbar.
BEGRÜNDUNG	JAVA ist die Lehrsprache an der Universität Ulm und damit den Entwicklern und dem Auftraggeber bekannt. Außerdem wird SonarQube zur Analyse des Source Codes verwendet und die Anwendung benötigt JAVA 11.
ABHÄNGIGKEITEN	QA10 → p. 49
PRIORITÄT	2
AKTEUR	Vertreter des Auftraggebers → p. 13, Entwickler → p. 13
<b>ID</b>	<b>QA10</b>
TITEL:	Dokumentation des Source Code
BESCHREIBUNG:	Der Source Code soll verständlich dokumentiert werden. Dabei wird der Javadoc-Stil verwendet. Automatisch generierter Code, wie zum Beispiel getter-Methoden sollen nicht dokumentiert werden. 90% des nicht automatisch erzeugten Codes sollen ausführlich dokumentiert werden.
BEGRÜNDUNG	Das ist eine Vorgabe des Auftraggebers. Außerdem erleichtert die Dokumentation von Code, dass andere Personen (unabhängig, ob sie Entwickler sind oder nicht) dokumentierten Code leichter verstehen und weiterentwickeln können. Ebenso kann die Dokumentation des Codes bei der Entwicklung helfen, sich bewusst zu werden, was der Code machen soll oder bei späterer Weiterentwicklung, was der Code macht.
ABHÄNGIGKEITEN	QA9 → p. 49
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13
<b>ID</b>	<b>QA11</b>
TITEL:	Implementierungs- und Dokumentationssprache
BESCHREIBUNG:	Der Source Code und die Dokumentation im Code, das heißt die Kommentare, müssen in Englisch verfasst werden.
BEGRÜNDUNG	Das ist eine Vorgabe des Auftraggebers. Außerdem ist englisch die etablierte Sprache bei der Implementierung und der Dokumentation, weil dadurch der Code international verständlich ist.
ABHÄNGIGKEITEN	QA10 → p. 49
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Entwickler → p. 13
<b>ID</b>	<b>QA12</b>
TITEL:	Sprache der Benutzerschnittstelle
BESCHREIBUNG:	Die Texte der Benutzerschnittstelle werden auf Deutsch verfasst.
BEGRÜNDUNG	Die Anwendung wird vor allem von Personen genutzt, deren Muttersprache deutsch ist und damit würde die englische Sprache wenig Vorteile bieten, jedoch eventuell Schwierigkeiten bei der Verständlichkeit hervorrufen.
ABHÄNGIGKEITEN	
PRIORITÄT	5
AKTEUR	Auftraggeber → p. 12, Teilnehmer → p. 12

<b>ID</b>	<b>QA13</b>
TITEL:	Benutzerhandbuch
BESCHREIBUNG:	Für die Anwendung soll ein Benutzerhandbuch bereitgestellt werden. Dieses Dokument wird auf Deutsch verfasst und soll dem Benutzer eine einfache Bedienung der Anwendung ermöglichen. In diesem Dokument sollen alle spiel-relevanten Informationen und Komponenten, sowie die Benutzerschnittstelle anhand von Beispielen erklärt werden.
BEGRÜNDUNG	Ein Benutzerhandbuch kann die Bedienung einer Anwendung und den Einstieg in das Spiel erleichtern. Dies vermeidet Frust bei den Benutzern, weil sie die Anwendung nicht verstehen oder Probleme bei der Bedienung haben.
ABHÄNGIGKEITEN	QA8→ p. 48, QA12→ p. 49
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Teilnehmer→ p. 12
<b>ID</b>	<b>QA14</b>
TITEL:	Dokumentation
BESCHREIBUNG:	Der gesamte Entwicklungsprozess und auch das entstandene Produkt muss dokumentiert werden. Das bedeutet, dass Tests, Reviews und weitere Maßnahmen zur Qualitätssicherung dokumentiert werden müssen. Außerdem soll ein Projekttagebuch zum Erfassen der Tätigkeiten (zum Beispiel Implementierung oder Treffen) geführt werden und ein Entwicklerhandbuch angelegt werden. Dieses Handbuch soll die Architektur und die implementierten Funktionen beschreiben, sowie die verwendeten Technologien auflisten und begründen.
BEGRÜNDUNG	Die Dokumentation ermöglicht nach und während der Entwicklung der Anwendung anderen Kunden, den Prozess zu beurteilen oder sich mit der Architektur und Umsetzung der Komponenten auseinanderzusetzen. Besonders auf der Messe, wenn man andere Komponenten einkauft, kann das die Entscheidung vereinfachen.
ABHÄNGIGKEITEN	QA7→ p. 48, QA15→ p. 50
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Entwickler→ p. 13
<b>ID</b>	<b>QA15</b>
TITEL:	Dokumentations-Sprache
BESCHREIBUNG:	Alle im Rahmen des Projekts, das heißt der Entwicklung der Anwendung, anfallenden Dokumente, wie zum Beispiel das Benutzerhandbuch werden auf Deutsch verfasst.
BEGRÜNDUNG	Die Anwendung wird vor allem von Personen genutzt, deren Muttersprache deutsch ist und damit würde die englische Sprache wenig Vorteile bieten, jedoch eventuell Schwierigkeiten bei der Verständlichkeit hervorrufen.
ABHÄNGIGKEITEN	QA14→ p. 50
PRIORITÄT	5
AKTEUR	Auftraggeber→ p. 12, Entwickler→ p. 13