



## **- Team 16 -**

Cornelius Huber

Enrique Rempp

Philipp Letschka

René Müller

Pflichtenheft v21.4.28

# Inhaltsverzeichnis:

<b>Pflichtenheft v21.4.28</b>	<b>1</b>
<b>Inhaltsverzeichnis:</b>	<b>2</b>
<b>Meilenstein 1</b>	<b>4</b>
Kontextanalyse	4
Fachwissen	5
Domänenmodell	12
Anforderungsdefinitionen	13
Akteure:	13
Funktionale Anforderungen:	14
Nichtfunktionale Anforderungen:	32
<b>Meilenstein 2</b>	<b>36</b>
Anwendungsfälle:	36
Main Menu:	36
Spiel beitreten:	38
Spielablauf (Serverside):	39
Spieler am Zug:	42
Editor:	43
Abläufe im System:	44
Zustandsdiagramme:	44
Player am Zug:	44
Hauptmenü:	46
Sequenzdiagramme:	47
<b>Meilenstein 3</b>	<b>51</b>
Dialogstruktur	51
Graphische Gestaltung und Nutzungskonzept	53
Hauptmenü:	53
Join Game:	55
Create Game:	55
Game:	57
Editor:	58
Character Editor:	58
Map Editor:	59
Szenario Editor:	60
Dialoge:	61
Alert Dialog:	61
Info Dialog:	62
Rückmeldung:	63
Server und KI Client:	64
<b>Meilenstein 5:</b>	<b>65</b>
Komponentendiagramm:	65
Artefakte:	68

## Kontextanalyse

In diesem Softwaregrundprojekt geht es um die Realisierung eines rundenbasiertes Taktik-Spiels welches sich an dem Marvel Universum orientiert. Das Spiel trägt den Namen "Marvelous Mashup" und lässt den Spieler in eine Rolle eines Marvel Helden steigen und gegen andere Spieler oder eine Künstliche Intelligenz spielen, die von Zuschauern im Nachhinein verfolgt werden kann.

Die Versionierung der Projektdateien wird durch Git gestützt, die Architektur basiert auf der Client-Server-Architektur, das bedeutet, dass die Spieler Clients nicht untereinander kommunizieren, sondern lediglich über den Server.

Es werden von verschiedenen Parteien die essentiellen Module der Spiele Software entwickelt (KI-Client, Server, Benutzer-Client, Editor) die nach Fertigstellung zusammengesetzt werden können. Durch diese Modularität können verschiedene Modulare Realisierungen ausgetauscht werden.

Das fertige System umfasst eine spielbare Version von Marvelous Mashup, bei dem der Spieler das Ziel hat, die sechs Infinity Stones anzueignen und das Universum zu retten - und das bevor der Gegner dies getan hat.

Nach der Umsetzung des hiesigen Projekts lassen sich zahlreiche weitere Zusatzfunktionen realisieren, wie das erweitern der Heldenauswahl, der Ausreifung der KI durch weitere Lernalgorithmen oder dem Hinzufügen weiterer Modi wie beispielsweise ein Kompetitives Ranking-System, dass derzeitigen Esport-Standards gerecht wird.

Durch die Freiheit eines modularen System lassen sich dementsprechend auch Lastenheft Fremde Erweiterungen einfach implementieren, wie beispielsweise ein Chat oder Ähnliches, die das System ausschmücken.

## Fachwissen

<b>Begriff</b>	Charakter
<b>Beschreibung</b>	Eine Figur aus allseits beliebten Comic-Heften, die durch eine Einheit auf dem Raster der Spielfelder repräsentiert wird.
<b>Ist-ein</b>	Bewegliche Einheit
<b>Kann-sein</b>	Player Character (PC), Non-Player Character (NPC)
<b>Aspekt</b>	Name, zugehöriger Spieler oder NPC, Eigenschaften, Health Points, Inventar, Position auf dem Spielfeld
<b>Bemerkung</b>	Der Begriff "Charakter" bezeichnet einen NPC der auf dem Spielfeld herumläuft
<b>Beispiele</b>	Wade Winston Wilson aka. Wade "Wichslurch" Wilson aka. Deadpool Gehört zur Heldengruppe von Benutzer Billy  Healthpoints: 80 Movement Points: 2 Action Points: 2 Inventar: empty Position {x: 3, y: 4} etc.

<b>Begriff</b>	Player Character
<b>Beschreibung</b>	Ein Charakter der von einem Menschen gesteuert wird
<b>Ist-ein</b>	Charakter
<b>Kann-sein</b>	
<b>Aspekt</b>	Der Player Charakter bringt das Spielgeschehen durch Aktionen voran
<b>Bemerkung</b>	
<b>Beispiele</b>	Deadpool, Thanos

<b>Begriff</b>	Non Player Character
<b>Beschreibung</b>	Ein Programmgesteuerter Charakter. Er wird nicht von einem Spieler gesteuert, daher kann sein handeln nicht gesteuert werden.
<b>Ist-ein</b>	Programmgesteuerter Verhaltens Algorithmus
<b>Kann-sein</b>	
<b>Aspekt</b>	Non Player Characters (NPCs) beeinflussen das Spielgeschehen auf Basis eines Algorithmus
<b>Bemerkung</b>	Diese Charaktere können nicht manuell vom Spieler gesteuert werden
<b>Beispiele</b>	Stan Lee, Goose

<b>Begriff</b>	Spieler
<b>Beschreibung</b>	Eine Person oder künstliche Intelligenz der oder die Aktionen von den Charakteren (wie Spielzüge) ausführt.
<b>Ist-ein</b>	Aktiver Teilnehmer des Spiels
<b>Kann-sein</b>	Mensch, künstliche Intelligenz
<b>Aspekt</b>	Ein Spieler besitzt 6 Charaktere, welche in der Spielvorbereitung gewählt werden.
<b>Bemerkung</b>	Es sind immer 2 Spieler pro Spiel notwendig
<b>Beispiele</b>	

<b>Begriff</b>	Künstliche Intelligenz / KI
<b>Beschreibung</b>	Ersetzt einen oder mehrere Spieler, um das Spiel auch alleine spielen zu können oder falls Soll an Haben der Mitspieler nicht erfüllt ist
<b>Ist-ein</b>	Ersatz für einen Spieler
<b>Kann-sein</b>	Unterschiedlicher Schweregrad
<b>Aspekt</b>	Schwierigkeit, IP und Port des Servers
<b>Bemerkung</b>	Kann aktiviert werden falls es nicht genug Spieler gibt.
<b>Beispiele</b>	

<b>Begriff</b>	Wahlphase
<b>Beschreibung</b>	Zu Beginn muss der Spieler eine Heldengruppe aus 6 Charakteren zusammenstellen. Die Herausforderung hier ist, die Charaktere aus dem Vorrat der in der Charakter-Konfiguration definierten Figuren so zu wählen, dass eine möglichst gute Kombination entsteht.
<b>Ist-ein</b>	Spielvorbereitung
<b>Kann-sein</b>	
<b>Aspekt</b>	
<b>Bemerkung</b>	Der Ablauf der Wahlphase aus Sicht eines Spielers ist so: 1. Der Server nimmt aus der Liste aller Charaktere der geladenen Charakter-Konfiguration 12 zufällige Charaktere, die dem Spieler angezeigt werden. 2. Der Spieler wählt 6 davon aus.
<b>Beispiele</b>	Spieler A wählt Thor, Hulk, Mantis, Rocket, Spider Man und Captain Marvel

<b>Begriff</b>	Highscore
<b>Beschreibung</b>	Zeigt den höchsten erreichten Punktestand an
<b>Ist-ein</b>	Vergleichswert den man übertreffen möchte
<b>Kann-sein</b>	Orientierung eines Höchstwertes, den es zu knacken gilt (Motivation)
<b>Aspekt</b>	Punktzahl
<b>Bemerkung</b>	
<b>Beispiele</b>	Spieler A hat einen Punktestand von 420.000 und damit einen Highscore, da es der Höchstwert ist

<b>Begriff</b>	Marvel
<b>Beschreibung</b>	Marvel ist eine US-Amerikanische Comic und Film Produktionsfirma. Sie hat u.a. Das marvel Cinematic Universe erschaffen, zusammen mit einer großen Anzahl an Superhelden.
<b>Ist-ein</b>	US-amerikanische Produktionsfirma für Film- und Fernsehen
<b>Kann-sein</b>	
<b>Aspekt</b>	Hat die Charaktere erfunden welche in unserem Spiel spielbar sein werden.
<b>Bemerkung</b>	
<b>Beispiele</b>	

<b>Begriff</b>	Infinity Stones
<b>Beschreibung</b>	Steine verschiedener Farbe mit unterschiedlichen Effekten, 6 Stück
<b>Ist-ein</b>	Item auf dem Spielfeld oder getragen von einem Charakter
<b>Kann-sein</b>	Verstärker eines Charakters wenn dieser ihn trägt indem es ihm einen zusätzlichen Effekt oder eine Fähigkeit gibt
<b>Aspekt</b>	Effekt (individuell)
<b>Bemerkung</b>	Jede Farbe hat einen bestimmten Effekt, ein Charakter kann mehrere Steine besitzen und so mehrere Effekte nutzen, Ziel des Spiel ist es alle 6 Steine auf einem Charakter zu haben.
<b>Beispiele</b>	Space Stone (blau): Der Träger kann sich mit einer Aktion auf ein beliebiges freies feld teleportieren.

<b>Begriff</b>	Effekte
<b>Beschreibung</b>	Jeder Infinity Stone hat einen eigenständigen
<b>Ist-ein</b>	Ein Effekt ist eine weitere Aktion die vom Spieler ausgeführt werden kann
<b>Kann-sein</b>	Eine Verstärkung für den Charakter
<b>Aspekt</b>	Cooldown Zeit (individuell)
<b>Bemerkung</b>	Jeder Effekt hat eine eigene Farbe
<b>Beispiele</b>	Space Stone (blau): Der Träger kann sich mit einer Aktion auf ein beliebiges freies feld teleportieren.

<b>Begriff</b>	Spielzug
<b>Beschreibung</b>	Die Charaktere kommen in einer zufälligen Reihenfolge hintereinander dran. Wenn ein Charakter an der Reihe ist kann dieser seine Movement Points und Action Points nutzen.
<b>Ist-ein</b>	Moment in dem ein Charakter eine oder mehrere Handlungen ausführen kann.
<b>Kann-sein</b>	
<b>Aspekt</b>	
<b>Bemerkung</b>	Alle Action und Movement Points werden zu Beginn des Spielzugs von einem Charakter zurückgesetzt.
<b>Beispiele</b>	Rocket Raccoon ist am Zug. Er hat z.B. laut der Charakter-Konfiguration 2 MP und 2 AP. Als erstes benutzt der Spieler einen AP um Rocket einen Fernkampf-Angriff auf Nebula machen zu lassen, die in Reichweite und Sichtlinie steht. Dann benutzt er einen MP, um den Space Stone auf einem Nachbarfeld aufzusammeln. Mit dem zweiten AP benutzt Rocket den Space Stone um sich neben den ausgeknockten Starlord zu teleportieren, damit er ihn im nächsten Zug (wenn er wieder APs hat) mit dem Soul Stone in seinem Inventar wiederbeleben kann.

<b>Begriff</b>	Movement-Points (MP)
<b>Beschreibung</b>	Diese Punkte geben die Anzahl an möglichen Bewegungen an, die ein Spieler oder die KI während ihrer Runde tätigen können
<b>Ist-ein</b>	Indikator der die Anzahl an Bewegungen einschränkt
<b>Kann-sein</b>	Eine Meta die Taktik implementiert und zum Überlegen anreizt, da die Bewegungen limitiert werden
<b>Aspekt</b>	Anzahl der freien Bewegungen pro Runde
<b>Bemerkung</b>	Jeder Spieler / Jede KI hat zu ihrer Rundenzeit eine verfügbare MP pro Charakter
<b>Beispiele</b>	Thanos hat 5 MP in der Runde, somit kann er 5 Bewegungen durchführen und sich auf dem Spielfeld bewegen

<b>Begriff</b>	Health-Points (HP)
<b>Beschreibung</b>	Diese Punkte indizieren die Lebenspunkte die ein Charakter zur Verfügung hat. Fällt dieser Wert auf 0, ist der Charakter außer Gefecht gesetzt und steht dem Spieler / der KI nicht mehr zur Verfügung (außer ein bestimmter Effekt wird im weiter Spielverlauf aktiviert).
<b>Ist-ein</b>	Indikator für die Lebenspunkte eines Charakters
<b>Kann-sein</b>	Motivation, den Charakter taktisch klug einzusetzen, damit er zum Sieg verhilft
<b>Aspekt</b>	Anzahl der Lebenspunkte
<b>Bemerkung</b>	Jeder Charakter besitzt Lebenspunkte, die im Verlauf des Spiels zu oder abnehmen
<b>Beispiele</b>	

<b>Begriff</b>	Meta
<b>Beschreibung</b>	Die Meta ist im Esports die "Most effective strategy available". Durch sorgfältiges balancieren von Werten und Fähigkeiten kann diese angepasst werden.
<b>Ist-ein</b>	Spielregeln, die das Spiel interessanter / kompetitiver gestalten
<b>Kann-sein</b>	Ansporn nachzudenken, da diese Spielregeln das Spiel meist taktisch schwieriger gestalten
<b>Aspekt</b>	
<b>Bemerkung</b>	Die Meta ist nicht essentiell für ein Spiel notwendig, ein Spiel kann theoretisch auch ohne Meta Elemente gespielt werden, allerdings gibt die Meta dem Spiel meist erst einen Spielwert, da sie das Projekt interessanter gestaltet für den Spieler, da dieser sich mit dieser vertraut machen muss und somit durch überlegen das Spiel gewinnen kann. Hiermit wird also auch die geistige Fähigkeit der Spieler unter Beweis gestellt.
<b>Beispiele</b>	Movement-Points, Health-Points

<b>Begriff</b>	Aktion
<b>Beschreibung</b>	Aktionen werden von Charakteren ausgeführt und bewirken verschiedene Effekte und Zustände, sowie Inkrementierung oder Dekrementierung von HP sowie MP und sind ein essentieller Bestandteil von <i>Marvelous Mashup</i>
<b>Ist-ein</b>	Essentielles Spielelement, dass es den Spielern / der KI erlaubt den Charakter bestimmte Dinge ausführen zu lassen
<b>Kann-sein</b>	
<b>Aspekt</b>	Art der Aktion, Aktion und Reaktion auf vorhergehende Einflüsse
<b>Bemerkung</b>	Durch Aktionen wird der Spielverlauf beeinflusst
<b>Beispiele</b>	Ein Charakter möchte sich bewegen- Dazu muss er die Bewegungsaktion anwenden um sich über das Spielfeld zu bewegen; Dies kostet in MP, also wird hierbei pro Bewegungsaktion ein MP dekrementiert



<b>Begriff</b>	Savanne von Wakanda
<b>Beschreibung</b>	Die Savanne von Wakanda ist ein fiktiver Ort auf dem Kontinent Afrika südlich der Sahara, welcher von Marvel erschaffen wurde.
<b>Ist-ein</b>	Sie dient als Ort des Spiels und wird damit auch zum Ort des Spielfelds.
<b>Kann-sein</b>	
<b>Aspekt</b>	Beinhaltet das Spielfeld
<b>Bemerkung</b>	Der Hintergrund im Spiel sollte an die Savanne von Wakanda erinnern.
<b>Beispiele</b>	

<b>Begriff</b>	Spielfeld
<b>Beschreibung</b>	Ein x auf y großes rechteckiges Raster aus Feldern
<b>Ist-ein</b>	
<b>Kann-sein</b>	Benutzerdefiniert angepasst
<b>Aspekt</b>	Beinhaltet alle Felder
<b>Bemerkung</b>	
<b>Beispiele</b>	Ein 10 * 10 großes Feld mit einer Mischung aus verschiedenen Feldern

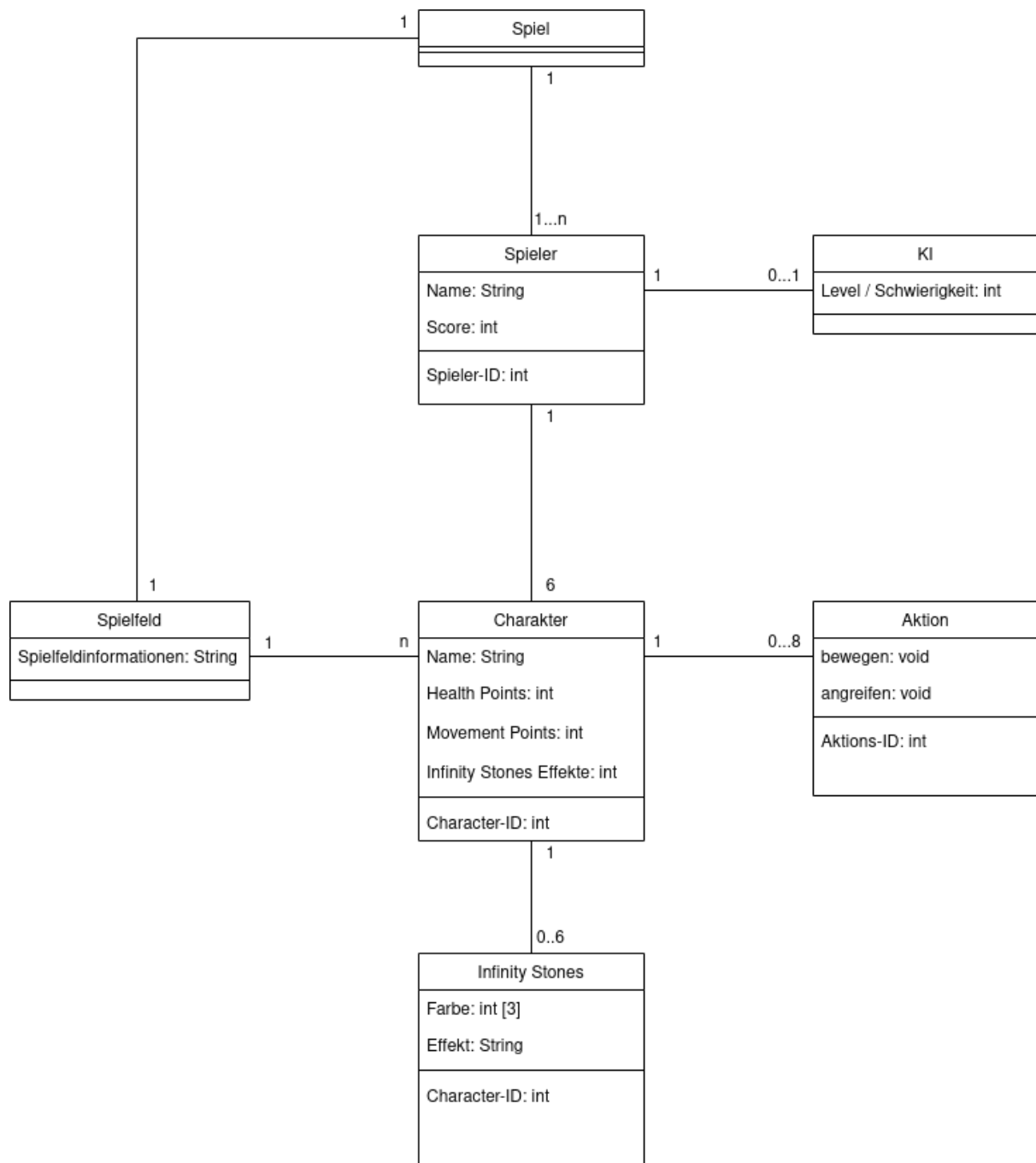
<b>Begriff</b>	Feld
<b>Beschreibung</b>	Felder des Spielbretts können freier Raum sein, oder durch Hindernisse oder Objekte besetzt sein. Sie können betretbar sein oder nicht, und die Sichtlinie durchlassen, oder blockieren.
<b>Ist-ein</b>	Ort auf dem ein Charakter stehen kann
<b>Kann-sein</b>	Gras, Gras mit Fels, Gras mit Charakter, Gras mit Infinity Stone.
<b>Aspekt</b>	Position auf dem Spielfeld
<b>Bemerkung</b>	
<b>Beispiele</b>	Das Feld an der rechten unteren Ecke ist ein Grasfeld ohne weitere besetzung.

<b>Begriff</b>	Goose
<b>Beschreibung</b>	Goose ist eine (besondere) Katze (Flerken) aus dem Marvel Cinematic Universe. Sie hat alle Infinity Stones verschluckt und verteilt nun mit der Zeit alle Steine auf dem Spielfeld.
<b>Ist-ein</b>	Flerken
<b>Kann-sein</b>	
<b>Aspekt</b>	
<b>Bemerkung</b>	Zu Beginn der Runden 1 bis 6 (also bevor die Charaktere ihre Züge machen) erscheint Goose auf einem zufälligen freien Feld, kotzt einen zufälligen der sich noch in ihrem Magen befindenden Infinity Stones auf diesem Feld aus, und verschwindet dann wieder.
<b>Beispiele</b>	Runde 1: Goose hat noch alle Steine, daher wird zufällig der Space Stone gewählt und auf Feld {3,4} platziert.

<b>Begriff</b>	Stan Lee
<b>Beschreibung</b>	Tritt vor Runde 7 auf und erscheint neben den schwächsten Charakter. Er heilt daraufhin alle die sich in seiner Sichtweite befinden.
<b>Ist-ein</b>	Erfinder der (meisten) Superhelden von Marvel und Comicautor der Marvel Comics. NPC
<b>Kann-sein</b>	
<b>Aspekt</b>	
<b>Bemerkung</b>	Stan ruft Excelsior um alle wiederzubeleben.
<b>Beispiele</b>	

<b>Begriff</b>	User-Client/Benutzer-Client
<b>Beschreibung</b>	Software, die im Gegensatz zum Server auf dem Rechner des Anwenders läuft.
<b>Ist-ein</b>	Die graphische Anwendung des Spiels.
<b>Kann-sein</b>	
<b>Aspekt</b>	
<b>Bemerkung</b>	
<b>Beispiele</b>	

## Domänenmodell



## Anforderungsdefinitionen

**Akteure:**

**Entwickler:**

Die Entwickler sind wir. Wir erstellen das Spiel komplett. Dabei können wir in unterschiedlichen Bereichen arbeiten.

**Server:**

Alle Aktionen die von dem Server ausgeführt werden.

**Spieler:**

Die Person, die das Spiel an der Client App spielt.

**KI:**

KI steht für künstliche Intelligenz. Dies ist Code, der einen Spieler im Spiel ersetzen soll, indem dieser sich wie ein Mensch im Spiel verhält und Entscheidungen trifft.

**NPC:**

NPC steht für Non-Player-Character. Ein NPC stellt einen Charakter im Spiel dar, der nicht von Menschen gespielt werden kann.

**Thanos:**

Thanos ist ein NPC. Dieser NPC ist der Antagonist im Spiel.

**Stan Lee:**

Stan Lee ist ein NPC. Er ist der Erfinder bzw. Autor von Marvel.

**Goose:**

Goose ist ein NPC. Goose erscheint vor jeder Runde auf einem zufälligen und freien Feld und legt einen Infinity Stone ab.

## Funktionale Anforderungen:

<b>ID:</b>	<b>FA1</b>
<b>Titel:</b>	Architektur
<b>Beschreibung:</b>	Das zu entwickelnde Produkt ist eine verteilte Anwendung mit Client-Server-Architektur. Die Clients kommunizieren dabei nur mit dem Server, aber nicht untereinander.
<b>Begründung:</b>	Das Spiel ist ein Online Spiel das von mehreren Spielern gleichzeitig gespielt wird und von einem Server abhängig ist.
<b>Abhängigkeiten:</b>	FA2 - FA8, FA33, FA34
<b>Akteure:</b>	Entwickler

<b>ID:</b>	<b>FA2</b>
<b>Titel:</b>	Server
<b>Beschreibung:</b>	Der Server beinhaltet die Anwendungslogik zur Verwaltung von Clients und Partien sowie die Umsetzung der Spielregeln. Spiele können von beliebig vielen Zuschauern passiv mitverfolgt werden.
<b>Begründung:</b>	Es ist eine zentrale Software benötigt, die (möglichst nicht manipulierbar) das Spielgeschehen steuert.
<b>Abhängigkeiten:</b>	FA1, FA33, FA34
<b>Akteure:</b>	Server, Entwickler

<b>ID:</b>	<b>FA3</b>
<b>Titel:</b>	User-Client
<b>Beschreibung:</b>	Ein Benutzer-Client mit grafischer Oberfläche, der es einem einzelnen menschlichen Benutzer ermöglicht, entweder aktiv als Mitspieler an einer Partie teilnehmen, oder als passiver Zuschauer im Zuschauermodus eine Partie zwischen zwei anderen Clients zu verfolgen.
<b>Begründung:</b>	Das Spiel muss auch gespielt werden können
<b>Abhängigkeiten:</b>	FA2, FA8, FA33, FA34
<b>Akteure:</b>	Entwickler, Spieler

<b>ID:</b>	<b>FA4</b>
<b>Titel:</b>	KI-Client
<b>Beschreibung:</b>	KI-Clients werden autonom von einer Künstlichen Intelligenz gesteuert und können als Mitspieler an einer Partie teilnehmen.
<b>Begründung:</b>	Damit können auch Personen ohne Freunde gegen jemanden spielen.
<b>Abhängigkeiten:</b>	FA2, FA3, FA33, FA34
<b>Akteure:</b>	KI, Entwickler

<b>ID:</b>	<b>FA5</b>
<b>Titel:</b>	Editor
<b>Beschreibung:</b>	Ein Editor, mit dem Content und Konfigurationen für das Spiel erstellt werden können, wie Szenarios, Partie Konfigurationen und Charakter-Konfigurationen.
<b>Begründung:</b>	Somit können die Spiele nach eigenem Ermessen beliebig gestaltet werden.
<b>Abhängigkeiten:</b>	FA35
<b>Akteure:</b>	Entwickler

<b>ID:</b>	<b>FA8</b>
<b>Titel:</b>	Kommunikation möglich
<b>Beschreibung:</b>	Über die WebSocket-Verbindung werden Textstrings im UTF-8 Encoding ausgetauscht, die im Format JSON repräsentierte Nachrichten enthalten.
<b>Begründung:</b>	Damit ist die Nachricht für sowohl Client-Anwendungen oder Server lesbar und interpretierbar.
<b>Abhängigkeiten:</b>	FA2, FA3, FA4, FA5, FA7, FA33, FA34
<b>Akteure:</b>	Entwickler, Server

<b>ID:</b>	<b>FA9</b>
<b>Titel:</b>	Charakterinfo anzeigen
<b>Beschreibung:</b>	Der Spieler kann nach belieben einsehen welche Werte alle Charaktere haben.
<b>Begründung:</b>	Das Spiel ist ein Spiel mit vollständiger Information.
<b>Abhängigkeiten:</b>	FA3, FA11, FA21
<b>Akteure:</b>	Spieler

<b>ID:</b>	<b>FA10</b>
<b>Titel:</b>	Spielfeld anzeigen
<b>Beschreibung:</b>	Das Spielfeld ist ein rechteckiges kartesisches Raster aus $x * y$ Feldern.
<b>Begründung:</b>	Der Spieler muss den momentanen Zustand des Spiels sehen.
<b>Abhängigkeiten:</b>	FA3, FA17, FA20, FA21, FA22, FA23, FA24
<b>Akteure:</b>	Spieler



ID:	FA11
Titel:	Charakter Info verwalten
Beschreibung:	<p>Charaktere haben bestimmte Werte, die in der Charakter-Konfiguration definiert sind:</p> <ul style="list-style-type: none"><li>• Name: Superhelden haben einen coolen Superhelden-Namen.</li><li>• Movement Points (MP) und Action Points (AP): Jeder Charakter bekommt pro Zug eine bestimmte Anzahl an Punkten, die er jeweils für Bewegungsschritte und Aktionen nutzen kann.</li><li>• Health Points (HP): Stellen den Gesundheitszustand des Charakters dar. Wenn die HPs auf 0 sinken, ist der Charakter ausgeknockt, und liegt einfach passiv am Boden. Er kann keine Züge mehr machen, bis er ggf. wiederbelebt wird.</li><li>• Inventar: Ein Charakter kann Infinity Stones in seinem Inventar tragen.</li><li>• Position: Die Position auf dem Spielfeld</li></ul> <p>Diese Informationen müssen richtig gespeichert, verwaltet, und übertragen werden.</p>
Begründung:	Damit die Charaktere korrekt gesteuert werden können müssen die Werte von ihnen richtig gespeichert und verwaltet werden.
Abhängigkeiten:	FA17
Akteure:	Spieler, Server

ID:	FA14
Titel:	Infinity Stones
Beschreibung:	<p>Es gibt sechs Infinity Stones. Jeder Stein hat eine Farbe und erlaubt dem Charakter, der ihn im Inventar hat, eine spezielle Aktion auszuführen. Nachdem diese Aktion ausgeführt wurde, kann der jeweilige Stein n Runden lang nicht mehr benutzt werden, um diese Aktion auszuführen. Auch wenn der Stein von jemand anderem übernommen wird, bleibt sie wirksam.</p> <p>Steine:</p> <ul style="list-style-type: none"> <li>• Space Stone (blau): Der Träger kann sich mit einer Aktion auf ein beliebiges freies Feld teleportieren.</li> <li>• Mind Stone (gelb): Der Träger kann seine mentale Energie in einen Strahl fokussieren, und einen Fernkampf Angriff auf jedes Ziel in Sichtlinie ausführen.</li> <li>• Reality Stone (rot): Der Träger kann mit einer Aktion auf einem benachbarten Feld einen Felsen verschwinden lassen, oder auf einem benachbarten freien Feld einen Felsen erscheinen lassen.</li> <li>• Power Stone (violett): Der Träger kann mit einer Aktion einen Nahkampfangriff ausführen, der den doppelten Schaden seines normalen Nahkampf-Angriffs macht. Die HPs des Trägers bei der Aktion um 10% seiner maximalen HPs reduziert.</li> <li>• Time Stone (grün): Der Träger kann mit einer Aktion die "Zeit zurückspulen", und seine MPs und APs auf den Anfangswert zurücksetzen.</li> <li>• Soul Stone (orange): Der Träger kann mit einer Aktion einen ausgeknockten Charakter auf einem benachbarten Feld "wiederbeleben".</li> </ul>
Begründung:	Zusätzliche Fähigkeiten und Gegenstand der Begierde.
Abhängigkeiten:	FA15
Akteure:	Goose, Spieler, KI

<b>ID:</b>	<b>FA15</b>
<b>Titel:</b>	Felder
<b>Beschreibung:</b>	Jedes einzelne Feld auf dem Spielbrett hat 3 Wahrheitswerte, ob es frei ist, ob es betretbar ist und ob es die Sicht blockiert, diese hängen von dem ab was sich auf dem Feld befindet
<b>Begründung:</b>	
<b>Abhängigkeiten:</b>	FA2, FA10, FA14
<b>Akteure:</b>	Spieler

<b>ID:</b>	<b>FA16</b>
<b>Titel:</b>	Spielstart
<b>Beschreibung:</b>	Vor der ersten Runde platziert der Server alle Charaktere auf zufälligen freien Felder auf dem Spielbrett.
<b>Begründung:</b>	Die Charaktere brauchen einen Ort an dem sie sich am Anfang befinden.
<b>Abhängigkeiten:</b>	FA2, FA15
<b>Akteure:</b>	Server, Spieler, KI

<b>ID:</b>	<b>FA17</b>
<b>Titel:</b>	Aktuellen Charakter anzeigen
<b>Beschreibung:</b>	Der Charakter, welcher gerade an der Reihe ist, wird gehighlightet.
<b>Begründung:</b>	Damit wird dem Spieler gezeigt, welcher Charakter am Zug ist.
<b>Abhängigkeiten:</b>	FA2, FA9, FA15
<b>Akteure:</b>	Server, Spieler

<b>ID:</b>	<b>FA18</b>
<b>Titel:</b>	Thanos
<b>Beschreibung:</b>	Nach einer zuvor definierten Anzahl an Runden erscheint Thanos auf einem zufälligem freiem Feld. Er bewegt sich auf das nächstgelegene Feld mit Infinity Stones zu und verwandelt auf dem Weg dazu alle Hindernisse zu Gras. Thanos erhält jede Runde einen zusätzlichen MP und ist immun gegen Angriffe.
<b>Begründung:</b>	Allzulange Partien sollen durch eine spieleigene Mechanik verhindert werden.
<b>Abhängigkeiten:</b>	FA21
<b>Akteure:</b>	Server, Thanos, NPC

<b>ID:</b>	<b>FA19</b>
<b>Titel:</b>	Sieges-Bildschirm anzeigen
<b>Beschreibung:</b>	Sobald ein Charakter am Ende einer Zugphase alle sechs Infinity Stones in seinem Inventar hat, endet die Partie sofort, und der Spieler dieses Superhelden hat gewonnen.
<b>Begründung:</b>	Ein Spieler hat die Gewinnbedingung erreicht und gewinnt das Spiel.
<b>Abhängigkeiten:</b>	FA2
<b>Akteure:</b>	Spieler

<b>ID:</b>	<b>FA20</b>
<b>Titel:</b>	Hintergrund anzeigen
<b>Beschreibung:</b>	Das Spiel ist thematisch in der Savanne Wakandas angesiedelt. Entsprechend muss der Hintergrund texturiert sein.
<b>Begründung:</b>	Der Hintergrund gibt einen Kontext für das Spiel
<b>Abhängigkeiten:</b>	FA10, FA3
<b>Akteure:</b>	Spieler

<b>ID:</b>	<b>FA21</b>
Titel:	Figuren anzeigen
Beschreibung:	Spieler und NPCs haben Charaktere auf dem Feld, diese werden angezeigt.
Begründung:	Um einen Überblick über das Spielgeschehen zu gewährleisten, muss die UI die Positionen der Charaktere anzeigen.
Abhängigkeiten:	FA9, FA11, FA17
Akteure:	NPC, Spieler

<b>ID:</b>	<b>FA22</b>
Titel:	Hindernisse anzeigen
Beschreibung:	Hindernisse können über das Spielfeld verteilt sein. Über Hindernisse kann nicht gegangen werden.
Begründung:	Für eine Routenplanung muss die Topologie der Karte bekannt sein.
Abhängigkeiten:	FA10
Akteure:	Spieler

<b>ID:</b>	<b>FA23</b>
Titel:	HP der Felsen anzeigen
Beschreibung:	Felsen können zerstört werden. Am Anfang haben sie 100 HP, diese können durch angriffe reduziert werden. Bei 0 werden sie zerstört.
Begründung:	Diese Anzeige ist für taktische Entscheidungen wichtig.
Abhängigkeiten:	FA10, FA22
Akteure:	Spieler

<b>ID:</b>	<b>FA24</b>
Titel:	Animationen
Beschreibung:	Möglichst alle Handlungen die auf dem Spielfeld geschehen sollen animiert werden.
Begründung:	Damit kann der Spieler eine Veränderung wahrnehmen (und es sieht gut aus)
Abhängigkeiten:	FA3
Akteure:	Spieler

<b>ID:</b>	<b>FA25</b>
Titel:	Helden auswählen
Beschreibung:	Der Spieler wählt seine 6 Charaktere aus 12 vom Server vorgegebenen.
Begründung:	Der Spieler soll seine Charaktere selber bestimmen können.
Abhängigkeiten:	FA2, FA3
Akteure:	Spieler, Server

<b>ID:</b>	<b>FA26</b>
Titel:	Reihenfolge anzeigen
Beschreibung:	Die Reihenfolge der nächsten Charaktere wird als UI angezeigt.
Begründung:	Damit kann der Spieler im Voraus planen welche Züge er machen will.
Abhängigkeiten:	FA3, FA17
Akteure:	Spieler

<b>ID:</b>	<b>FA27</b>
<b>Titel:</b>	Movement und Action Points anzeigen
<b>Beschreibung:</b>	Ein Charakter hat bei seinem Zug eine bestimmte Anzahl an Movement und Action Points. Ihm soll nun angezeigt werden wie viele ihm zur Verfügung stehen
<b>Begründung:</b>	Damit kann der Spieler besser planen was er noch tun kann.
<b>Abhängigkeiten:</b>	FA3, FA11, FA28, FA29
<b>Akteure:</b>	Spieler

<b>ID:</b>	<b>FA28</b>
<b>Titel:</b>	Movement und Action Points benutzen
<b>Beschreibung:</b>	Der Spieler steuert über ein UI wie er Action und Movement Points einsetzt und wie sich der Charakter dementsprechend verhält.
<b>Begründung:</b>	Der Spieler muss die Möglichkeit haben seine Movement und Action Points einzusetzen.
<b>Abhängigkeiten:</b>	FA3, FA11, FA27, FA29
<b>Akteure:</b>	Spieler

<b>ID:</b>	<b>FA29</b>
<b>Titel:</b>	Möglichkeiten einschränken
<b>Beschreibung:</b>	Die Benutzung von Action und Movement Points wird eingeschränkt sobald keine mehr vorhanden sind oder er nicht an der Reihe ist.
<b>Begründung:</b>	Da der Spieler nicht unbegrenzt Action und Movement Points besitzt und nicht beliebig einsetzen darf, muss seine Steuerung eingeschränkt werden.
<b>Abhängigkeiten:</b>	FA3, FA11, FA27, FA28
<b>Akteure:</b>	Spieler

<b>ID:</b>	<b>FA30</b>
<b>Titel:</b>	Goose
<b>Beschreibung:</b>	Zu Beginn von Runden 1 bis 6 übergibt sich dieser Charakter auf ein zufälliges freies Feld und erbricht einen der Infinity Stones, der noch nicht bereits im Spiel ist.
<b>Begründung:</b>	Die Infinity Stones sollten nicht direkt, sondern mit der Zeit auf dem Feld platziert werden.
<b>Abhängigkeiten:</b>	FA9, FA10, FA14, FA15, FA21
<b>Akteure:</b>	Server, Goose, NPC

<b>ID:</b>	<b>FA31</b>
<b>Titel:</b>	Stan Lee
<b>Beschreibung:</b>	Stan Lee ist der Kopf hinter Marvel. Er erscheint zu Beginn von Runde sieben neben einem zufälligen ausgeknockten Charakter und setzt die Anzahl seiner Leben wieder auf die maximale Anzahl zurück. Falls es keine ausgeknockten Charakter gibt, dann wird der mit den geringsten HP geheilt. Anschließend ruft Stan "Excelsior" und verschwindet.
<b>Begründung:</b>	Forderung des Lastenhefts.
<b>Abhängigkeiten:</b>	FA10, FA15, FA21
<b>Akteure:</b>	Server, Stan Lee, NPC



<b>ID:</b>	<b>FA32</b>
<b>Titel:</b>	Spiel Informationen anzeigen
<b>Beschreibung:</b>	Generelle Informationen werden als UI angezeigt. Zb: <ul style="list-style-type: none"><li>• Rundennummer</li><li>• Spielernamen</li><li>• Charakternamen</li></ul>
<b>Begründung:</b>	Damit bekommt der Spieler weitere Informationen die er nutzen kann um seine Züge zu planen.
<b>Abhängigkeiten:</b>	FA3, FA9, FA11
<b>Akteure:</b>	Spieler

<b>ID:</b>	<b>FA33</b>
<b>Titel:</b>	Nachrichten lesen
<b>Beschreibung:</b>	Ankommende Nachrichten vom Server werden richtig gelesen und interpretiert.
<b>Begründung:</b>	Der Server sendet Nachrichten, welche für die Spielbarkeit des Spiels notwendig sind. Dazu zählen unter anderem Handlungen des anderen Spielers oder Handlungen von NPCs.
<b>Abhängigkeiten:</b>	FA2, FA3, FA4
<b>Akteure:</b>	Server

<b>ID:</b>	<b>FA34</b>
<b>Titel:</b>	Nachrichten an den Server senden.
<b>Beschreibung:</b>	Jede Handlung des Spielers wird als Nachricht an den Server übertragen.
<b>Begründung:</b>	Da die Kommunikation der beiden Spieler über den Server läuft, muss jede Handlung des Spielers als Nachricht an den Server übertragen werden.
<b>Abhängigkeiten:</b>	FA2, FA3, FA4
<b>Akteure:</b>	Spieler, Server

<b>ID:</b>	<b>FA35</b>
<b>Titel:</b>	Hauptmenu UI
<b>Beschreibung:</b>	Das Spiel soll ein Hauptmenü mit allen notwendigen Funktionen des Spiels haben. Funktionen sind u.a: <ul style="list-style-type: none"><li>• Spiel beitreten</li><li>• Spiel erstellen</li><li>• Editor öffnen</li><li>• Anwendung beenden</li></ul>
<b>Begründung:</b>	Damit kann der Spieler entscheiden was er tun will.
<b>Abhängigkeiten:</b>	FA3
<b>Akteure:</b>	Spieler

<b>ID:</b>	<b>FA36</b>
<b>Titel:</b>	Spiel erstellen
<b>Beschreibung:</b>	Der Spieler kann ein benutzerdefiniertes Spiel erstellen.
<b>Begründung:</b>	Möglichkeit ein Spiel zu starten.
<b>Abhängigkeiten:</b>	FA3, FA35
<b>Akteure:</b>	Spieler

<b>ID:</b>	<b>FA37</b>
<b>Titel:</b>	Spiel beitreten
<b>Beschreibung:</b>	Ein Spieler kann einem Spiel aus einer Liste von verfügbaren Spielen beitreten.
<b>Begründung:</b>	Damit finden sich 2 Spieler für ein Spiel
<b>Abhängigkeiten:</b>	FA3, FA35
<b>Akteure:</b>	Spieler, Server

<b>ID:</b>	<b>FA38</b>
<b>Titel:</b>	Anwendung beenden
<b>Beschreibung:</b>	Eine saubere Möglichkeit die Anwendung zu beenden
<b>Begründung:</b>	Damit ist es für den Spieler einfach die Anwendung auch wieder zu beenden.
<b>Abhängigkeiten:</b>	FA3, FA35
<b>Akteure:</b>	Spieler

<b>ID:</b>	<b>FA39</b>
<b>Titel:</b>	Spiel pausieren
<b>Beschreibung:</b>	Falls ein Spieler eine Pausierung der Partie wünscht, bis ein Spieler anzeigt, dass er weiterspielen will. KI-Clients können keine Pause einlegen.
<b>Begründung:</b>	Verbindliche Anforderung im Lastenheft
<b>Abhängigkeiten:</b>	FA43, FA3
<b>Akteure:</b>	Spieler

<b>ID:</b>	<b>FA40</b>
<b>Titel:</b>	Spiel personalisieren
<b>Beschreibung:</b>	Der Spieler kann bei der Erstellung eines Spiels entscheiden wie das Spiel aussehen soll. Dabei kann er eine im Editor erstellte Konfiguration auswählen.
<b>Begründung:</b>	Bringt mehr Abwechslung
<b>Abhängigkeiten:</b>	FA3, FA5, FA35
<b>Akteure:</b>	Spieler

<b>ID:</b>	<b>FA41</b>
<b>Titel:</b>	Spieler Timeout
<b>Beschreibung:</b>	Wenn einer der Spieler wegen eines Verbindungsabbruchs nicht mehr mitspielen kann wird erstmal gewartet ob der Spieler wieder zurückkehrt.
<b>Begründung:</b>	Dadurch wird ein Spiel nicht direkt abgebrochen.
<b>Abhängigkeiten:</b>	FA39, FA42
<b>Akteure:</b>	Spieler

<b>ID:</b>	<b>FA42</b>
<b>Titel:</b>	Spieler reconnect
<b>Beschreibung:</b>	Wenn ein Spieler die Verbindung verloren hat, während er noch in einem Spiel war kann er dem Spiel wieder beitreten.
<b>Begründung:</b>	Dadurch wird ein Spiel nicht direkt abgebrochen.
<b>Abhängigkeiten:</b>	FA39, FA41
<b>Akteure:</b>	Spieler, Server

<b>ID:</b>	<b>FA43</b>
<b>Titel:</b>	Pause Button
<b>Beschreibung:</b>	Falls ein Spieler eine Pausierung der Partie wünscht, bis ein Spieler anzeigt, dass er weiterspielen will. Hier wird das passende UI Element geliefert.
<b>Begründung:</b>	Verbindliche Anforderung im Lastenheft
<b>Abhängigkeiten:</b>	FA39
<b>Akteure:</b>	Spieler

<b>ID:</b>	<b>FA44</b>
<b>Titel:</b>	Spiel auswählen
<b>Beschreibung:</b>	Auswahl eines Spiels in der Aufbauphase aus einer Liste aller erstellten noch nicht begonnenen Spiele.
<b>Begründung:</b>	Somit können Spieler gezielt Spielen beitreten.
<b>Abhängigkeiten:</b>	FA36
<b>Akteure:</b>	Spieler

ID:	FA45
Titel:	Spiel zusehen
Beschreibung:	Zuschauer sollten dem Spiel beitreten können
Begründung:	Wenn das Spiel bereits zu voll ist, kann zugesehen werden.
Abhängigkeiten:	FA10, FA36
Akteure:	Spieler

## Nichtfunktionale Anforderungen:

<b>ID:</b>	<b>QA1</b>
Titel:	Berechnungszeit
Beschreibung:	Spieler sollten nicht länger als eine Sekunde auf das Resultat ihrer Handlung warten müssen, daher sollte der Server möglichst effizient implementiert sein.
Begründung:	Übermäßige Wartezeiten verwirren und desorientieren Spieler.
Abhängigkeiten:	
Akteure:	
Priorität:	5

<b>ID:</b>	<b>QA2</b>
Titel:	Crashes
Beschreibung:	Nur jedes 100. Spiel sollte crashen.
Begründung:	Durch dieses QA wird sichergestellt, dass Spieler nicht an Fehlern die Lust am Spielen verlieren.
Akteure:	
Priorität:	3

<b>ID:</b>	<b>QA3</b>
Titel:	Zuschauer
Beschreibung:	Mindestens 10 Zuschauer sollten zusehen können.
Begründung:	Dies ist nötig, um eine Skalierbarkeit gewährt zu leisten.
Akteure:	
Priorität:	2

<b>ID:</b>	<b>QA4</b>
<b>Titel:</b>	Kommentar Sprache
<b>Beschreibung:</b>	Als Kommentar Sprache sollte Englisch verwendet werden.
<b>Begründung:</b>	Die Kommentare im Code sollten von jedem Informatiker verstanden werden.
<b>Abhängigkeiten:</b>	
<b>Akteure:</b>	Entwickler
<b>Priorität:</b>	4

<b>ID:</b>	<b>QA5</b>
<b>Titel:</b>	Dokumentensprache
<b>Beschreibung:</b>	Dokumente können in Deutsch oder in Englisch geschrieben werden.
<b>Begründung:</b>	Da die Dokumente nur von einer begrenzten Anzahl an Personen gelesen werden. Außerdem ist von diesen Personen zu erwarten das sie deutsch sprechen können.
<b>Abhängigkeiten:</b>	
<b>Akteure:</b>	Entwickler
<b>Priorität:</b>	7

<b>ID:</b>	<b>QA6</b>
<b>Titel:</b>	UI Sprache
<b>Beschreibung:</b>	Die von der UI verwendeten Sprache kann auch Deutsch oder Englisch sein.
<b>Begründung:</b>	Da das Spiel auch nur von einer begrenzten Anzahl an Personen gespielt werden wird, kann die Sprache auch deutsch sein. Außerdem ist von diesen Personen zu erwarten das sie deutsch sprechen können.
<b>Abhängigkeiten:</b>	
<b>Akteure:</b>	Entwickler
<b>Priorität:</b>	7



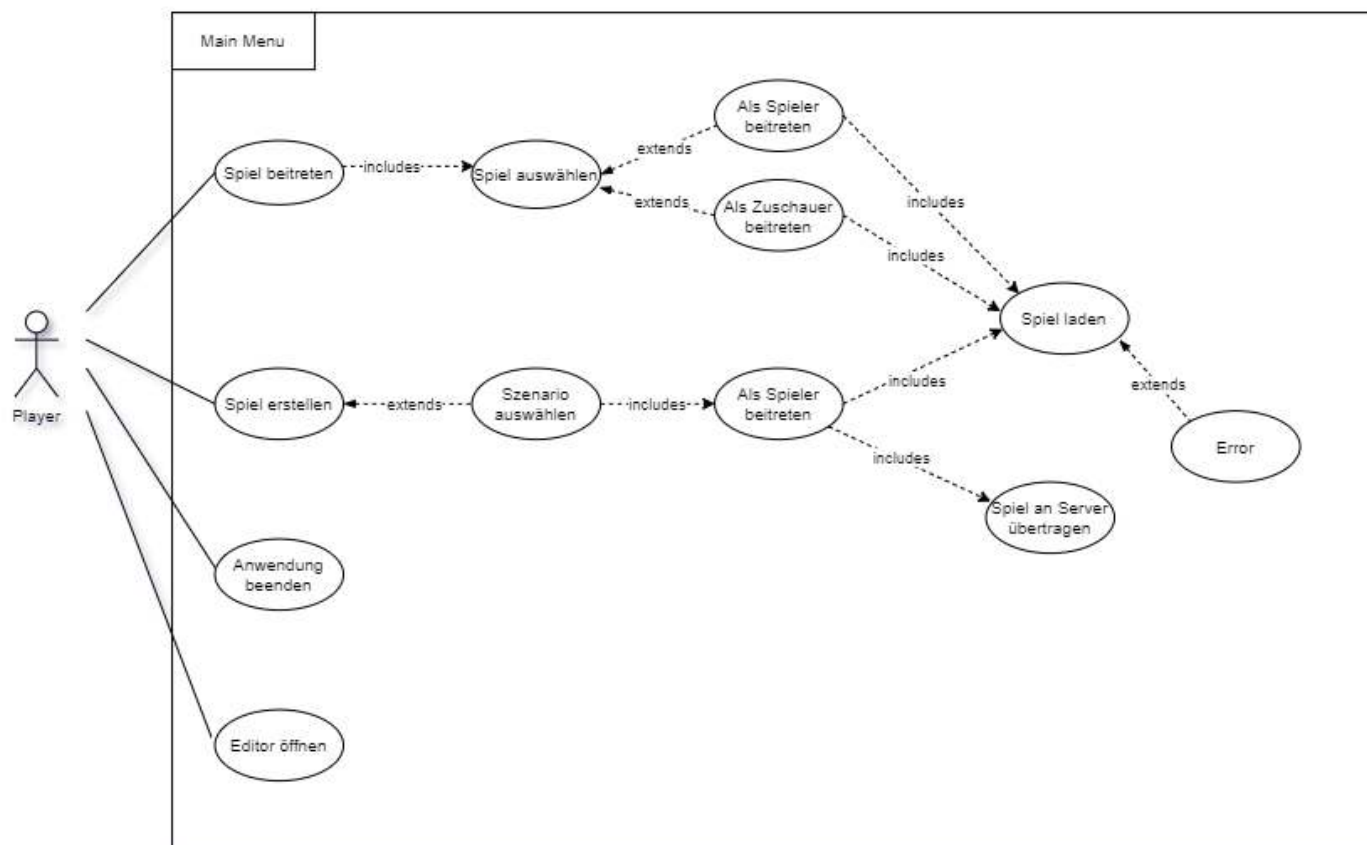
<b>ID:</b>	<b>QA7</b>
<b>Titel:</b>	Plattform
<b>Beschreibung:</b>	<p>Die Komponenten Benutzer-Client und Editor müssen jeweils auf mindestens einer der folgenden Plattformen laufen:</p> <ul style="list-style-type: none"> <li>• Eine der verbreiteten, aktuellen Linux-Distributionen</li> <li>• Eine aktuelle Version von Microsoft Windows</li> <li>• Basierend auf Webtechnologien, in einem aktuellen, standardkonformen Browser</li> </ul> <p>Die Komponenten KI-Client und Server müssen als Docker-Container lauffähig sein.</p>
<b>Begründung:</b>	Dadurch kann der Server-Code auf dem passenden Server laufen. Außerdem hat ein großer Teil der Personen, welche das Spiel spielen wollen die Möglichkeit dazu
<b>Abhängigkeiten:</b>	
<b>Akteure:</b>	Entwickler, Spieler
<b>Priorität:</b>	7

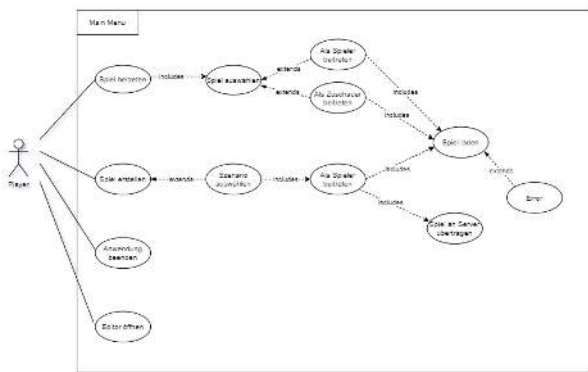
<b>ID:</b>	<b>QA8</b>
<b>Titel:</b>	Sicherheit / Datenschutz
<b>Beschreibung:</b>	Der Austausch von Informationen soll gegen mögliche Angriffe geschützt sein.
<b>Begründung:</b>	Dadurch könnten die Aktionen verändert und ein unfaires Spiel erzeugt werden.
<b>Abhängigkeiten:</b>	
<b>Akteure:</b>	Entwickler
<b>Priorität:</b>	7

ID:	QA9
Titel:	Wartbarkeit / Portabilität / Änderbarkeit / Testbarkeit / Verständlichkeit
Beschreibung:	Der Code soll so geschrieben sein, das einfach Änderungen an ihm vorgenommen werden können. Dafür muss er verständlich geschrieben sein. Tests sollen möglich sein und daraus folgend sollte es gut möglich sein den Code anzupassen.
Begründung:	Der Code muss einfach zu verändern sein um ein Gelingen des Projekts sicher zu stellen.
Abhängigkeiten:	
Akteure:	Entwickler
Priorität:	1

## Anwendungsfälle:

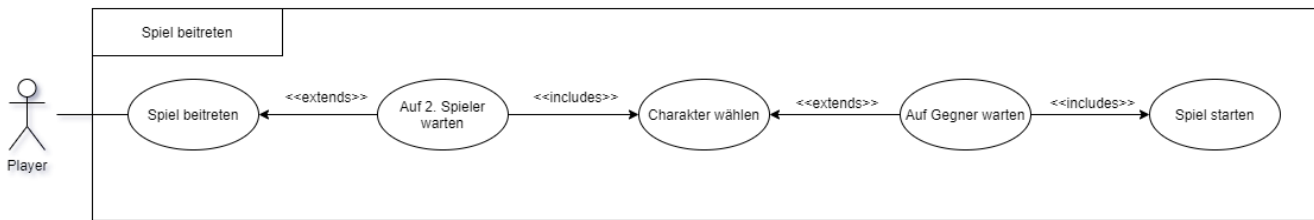
### Main Menu:





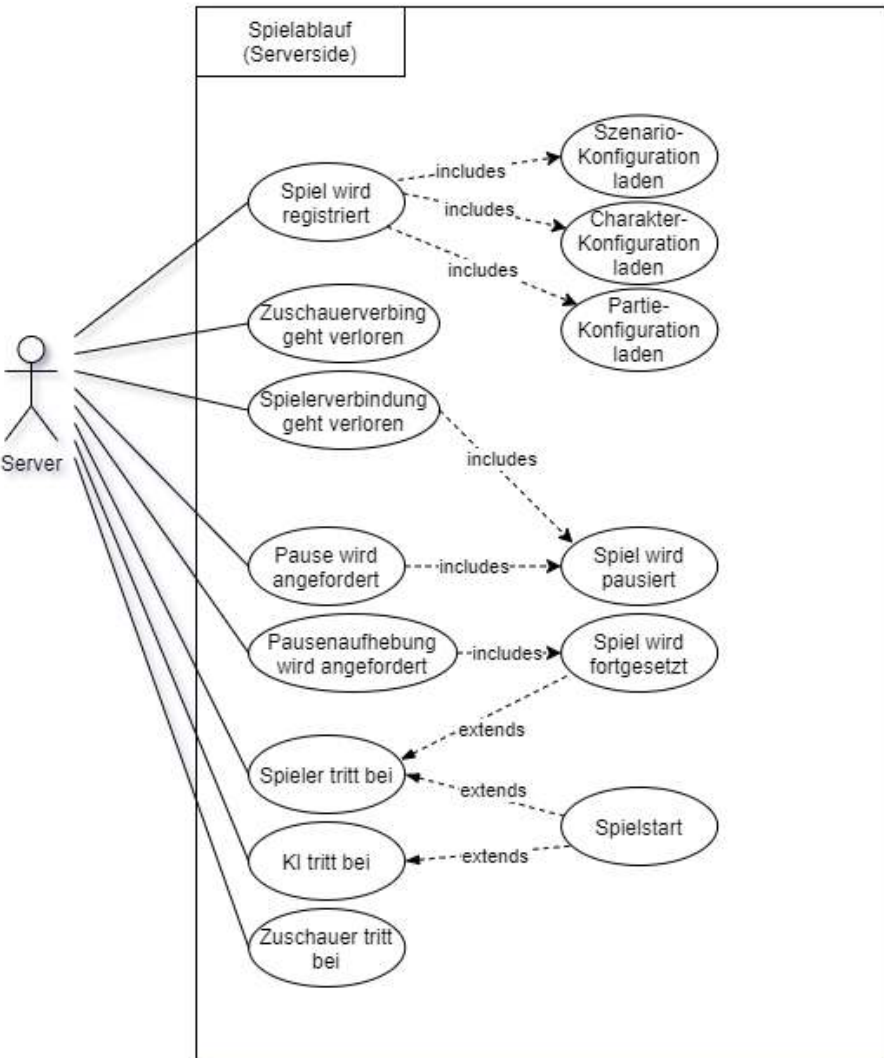
Anwendungsfall	Beschreibung	FAs
<b>Spiel beitreten</b>	Der Spieler/Zuschauer kann einem bereits erstellten noch nicht laufendem Spiel beitreten, ohne jedoch Eistellungen zu dem Spiel tätigen zu können (Anzahl der Spieler, Schwierigkeit etc)	FA37
<b>Spiel auswählen</b>	Auswahl eines Spiels in der Aufbauphase aus einer Liste aller erstellten noch nicht begonnenen Spiele	FA44
<b>Als Spieler beitreten</b>	Beitritt zu einem Spiel als aktiver Akteur und Spieler, wobei man berechtigt ist Züge und Aktionen auf dem Spielfeld mit Helden auszuführen	FA37, FA44
<b>Als Zuschauer beitreten</b>	Beitritt zu einem Spiel als passiver Akteur und Zuschauer, wobei man keine Berechtigung hat ins Spielgeschehen einzugreifen, dieses aber mitzuverfolgen	FA1, FA2, FA3
<b>Spiel laden</b>	Lädt den aktuellen Fortschritt eines Spiels, zu Beginn verwendet um ein neues Spiel zu laden	FA16
<b>Error</b>	Fehlermeldung aufgrund eines Fehlers im Spiel und/oder eine nicht zugelassene Aktion	FA2, FA33, FA34
<b>Spiel erstellen</b>	Der Spieler erstellt ein neues Spiel, wobei nur er allein die Rechte hat gewisse Werte des Spiels festzulegen, wie Anzahl der Spieler oder der Schwierigkeitsgrad	FA36
<b>Szenario auswählen</b>	Auswahl zwischen mehreren Spielmöglichkeiten	FA40
<b>Spiel an Server übertragen</b>	Überträgt den aktuellen Spielfortschritt an den Server	FA2, FA34
<b>Anwendung beenden</b>	Beendet das Spiel und schließt das Fenster	FA38
<b>Editor öffnen</b>	Öffnet den Editor Screen	FA5

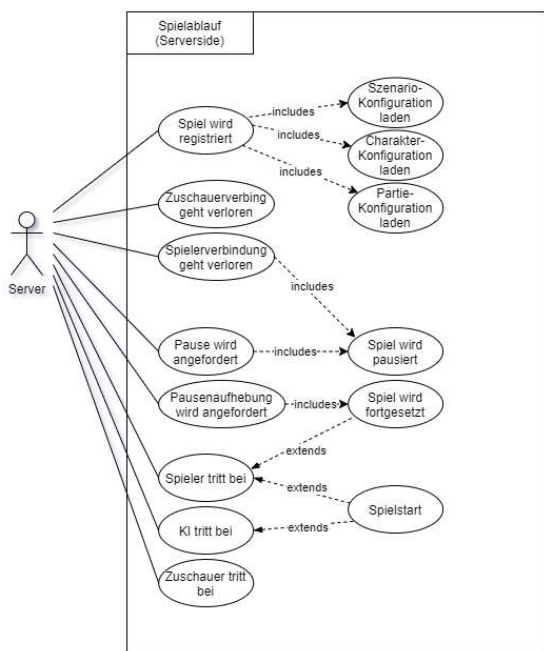
## Spiel beitreten:



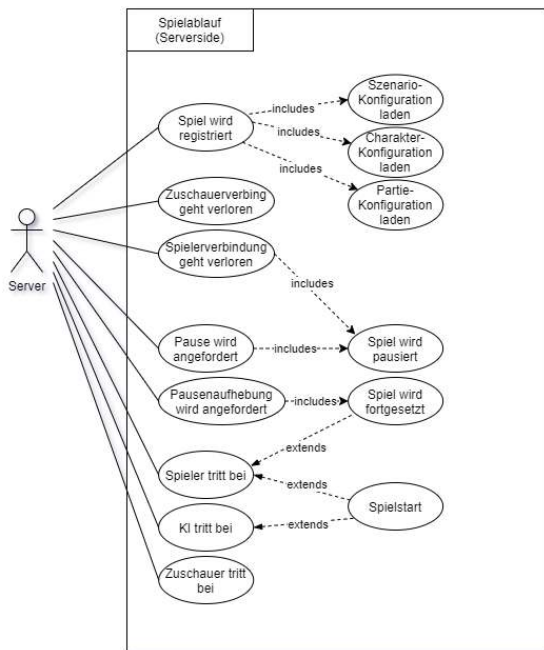
Anwendungsfall	Beschreibung	FAs
<b>Auf 2. Spieler warten</b>	Falls der Spieler bis jetzt der einzige spielende Spieler (Zuschauer ausgeschlossen) ist, wird auf einen weiteren Spieler gewartet. Der Spieler hat hier zusätzlich die Möglichkeit eine KI als Gegner hinzuzufügen.	FA36, FA37
<b>Charakter wählen</b>	Auswahl von 6 Helden (gilt universell, da jeder Held nur einmal existiert und nicht mehrmals in einem Team oder in mehreren Teams gleichzeitig kämpfen kann). Der Spieler hat hierbei außerdem nur die Wahl aus 12 Helden.	FA25
<b>Auf Gegner warten</b>	Wenn die 6 Helden fertig gewählt wurden muss evtl noch auf den anderen Spieler gewartet werden.	FA36, FA37
<b>Spiel starten</b>	Das Spiel kann gestartet werden.	FA1, FA10, FA16, FA20, FA21, FA22, FA23, FA25, FA26, FA32

## Spielablauf (Serverside):





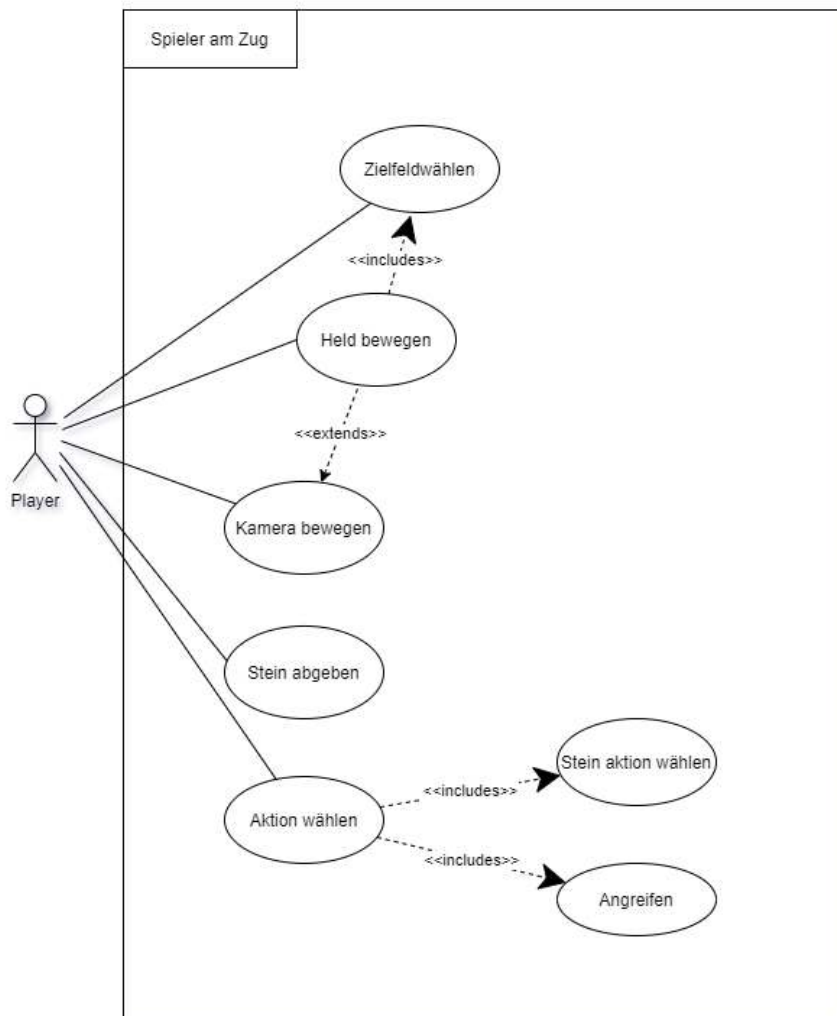
Anwendungsfall	Beschreibung	FAs
<b>Spiel wird registriert</b>	Ein neues Spiel wird erstellt. Die Konfig Dateien werden empfangen und werden nun interpretiert. Das Spiel wird nun zur Verfügung gestellt.	FA1, FA2, FA4, FA5, FA14, FA16
<b>Szenario-Konfiguration laden</b>	Hierbei werden generelle Einstellungen geladen.	FA1, FA2, FA4, FA5, FA14, FA16
<b>Charakter-Konfiguration laden</b>	Die Helden werden mit ihren Werten geladen.	FA11
<b>Partie-Konfiguration laden</b>	Die Karte wird geladen.	FA5
<b>Zuschauer-Verbindung geht verloren</b>	Ein Zuschauer beendet die Verbindung (beabsichtigt oder wegen einem Verbindungsfehler)	
<b>Spieler-Verbindung geht verloren</b>	Ein Spieler verliert die Verbindung. Hierbei wird das Spiel pausiert.	FA41
<b>Pause wird angefordert</b>	Einer der beiden Spieler pausiert das Spiel.	FA39, FA43
<b>Pausen Aufhebung wird angefordert</b>	Einer der beiden Spieler kann nun die Pause wieder aufheben. Das muss hierbei nicht der Spieler sein der pausiert hat.	FA39, FA43



Anwendungsfall	Beschreibung	FAs
<b>Spiel wird pausiert</b>	Der Spielablauf wird pausiert. Das wird mit einem Dialog angezeigt, dies ist nur Benutzer-Clients möglich, im Gegensatz zu KI-Clients.	FA41
<b>Spiel wird fortgesetzt</b>	Das Spiel wird fortgesetzt.	FA42
<b>Spieler tritt bei</b>	Ein Spieler tritt dem Spiel bei. Hierbei wird nun entweder auf einen 2. Spieler oder Bot gewartet oder das Spiel gestartet.	FA3
<b>KI tritt bei</b>	Ein Bot (KI) tritt dem Spiel bei. Hierbei wird nun entweder auf einen 2. Spieler oder Bot gewartet oder das Spiel gestartet.	FA4
<b>Zuschauer tritt bei</b>	Ein Zuschauer tritt bei. Das kann zu jedem Zeitpunkt im Spiel passieren.	FA37
<b>Spielstart</b>	Wenn 2 Spieler oder Bots beigetreten sind wird das Spiel gestartet.	FA1, FA10, FA16, FA20, FA21, FA22, FA23, FA25, FA26, FA32

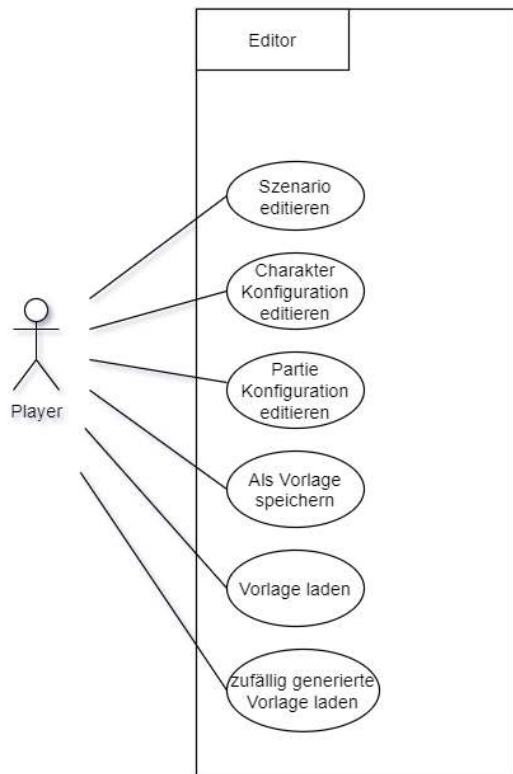


## Spieler am Zug:



Anwendungsfall	Beschreibung	FAs
<b>Held bewegen</b>	Der Held nutzt einen oder mehrere Movement Points um sich auf einem anderen Feld zu platzieren.	FA27, FA28, FA29
<b>Zielfeld wählen</b>	Das Feld auf das sich der Held bewegen soll wird gewählt.	FA15
<b>Kamera bewegen</b>	Die Kamera wird bewegt, um einen bestimmten Teil des Spielfelds anzuschauen oder zu fokussieren.	FA24
<b>Stein abgeben</b>	Ein Stein der in dem Besitz des aktuellen Charakters ist wird an einen anderen Charakter abgegeben.	FA14, FA25
<b>Aktion wählen</b>	Der Spieler wählt was er mit einem Action Point machen will.	FA28
<b>Stein Aktion wählen</b>	Wenn der Charakter Infinity Steine besitzt kann er wählen ob und welchen er benutzen möchte.	FA14
<b>Angreifen</b>	Er kann einen Nah- oder Fernkampf Angriff machen, der Angriff und der am Ziel zugefügte Schaden wird visualisiert, sodass sichtbar ist, welches Ziel angegriffen wurde.	FA28

Editor:

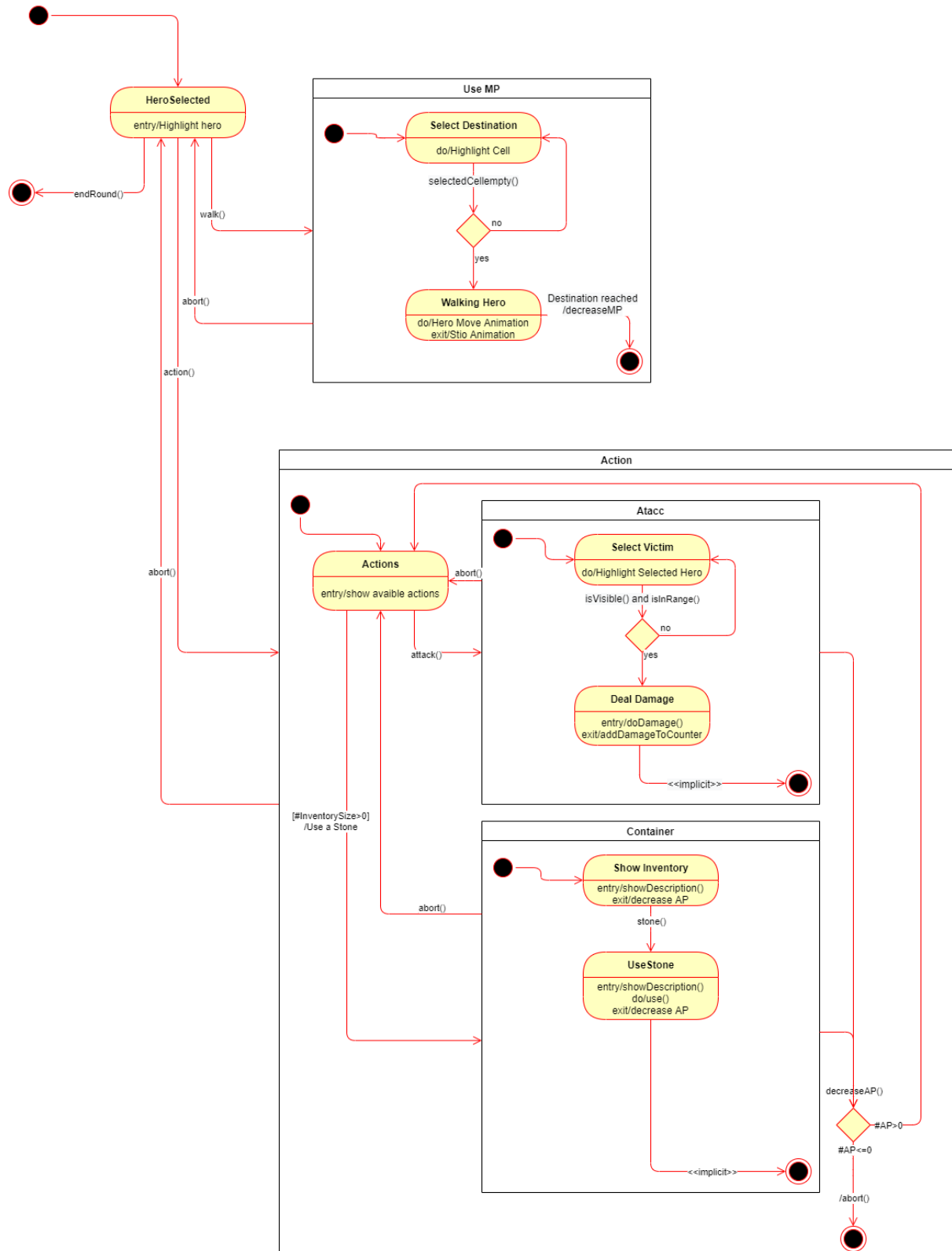


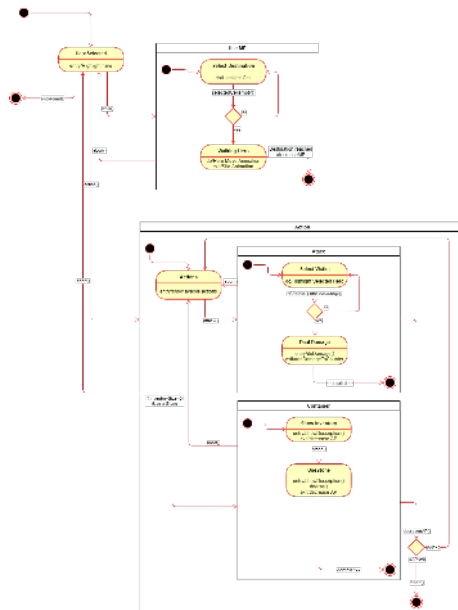
Anwendungsfall	Beschreibung	FAs
<b>Szenario editieren</b>	Die Map kann hier angepasst werden. Es kann sowohl Größe als auch aussehen eingestellt werden.	FA5
<b>Charakter Konfiguration editieren</b>	Hier können die Werte der einzelnen Charakter angepasst werden.	FA5, FA11
<b>Partie Konfiguration editieren</b>	Die generellen Spieleinstellungen werden hier eingestellt.	FA5
<b>Als Vorlage speichern</b>	Hier kann eine Konfiguration gespeichert werden. Sie kann ab sofort bei der Erstellung eines neuen Spiels verwendet werden.	FA2, FA34
<b>Vorlage laden</b>	Eine Konfiguration wird geladen und kann nun weiter bearbeitet werden.	FA2, FA33
<b>zufällig generierte Vorlage laden</b>	Eine zufällige Konfiguration wird gesetzt und kann nun noch bearbeitet werden.	FA2, FA33

## Abläufe im System:

### Zustandsdiagramme:

#### Player am Zug:

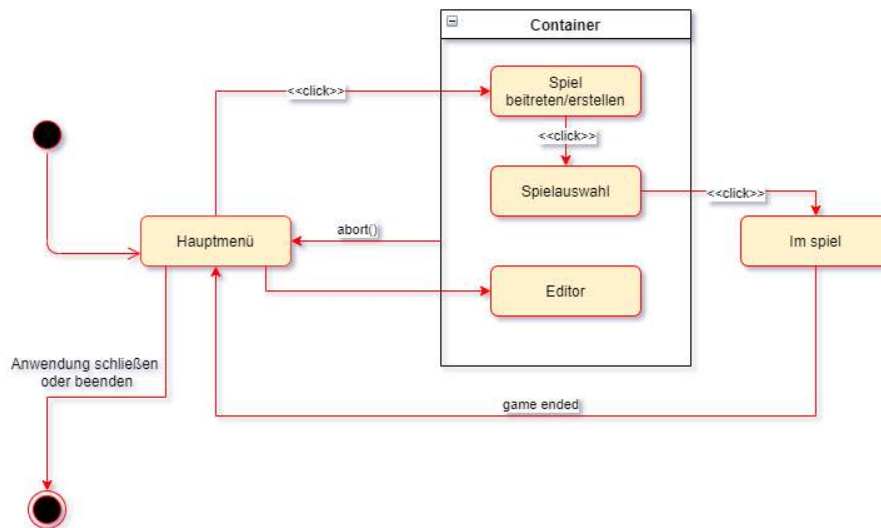




Der Spieler hat nach der Auswahl eines Charakters die Wahl, sich zu bewegen, oder eine Aktion durchzuführen. Wenn der Spieler sich bewegen will, wird überprüft, ob das angegebene Spielfeld leer ist. Ist dies der Fall, bewegt sich der entsprechende Charakter zum angegebenen Spielfeld. Entscheidet der Spieler sich zu einer Aktion, kann er wählen zwischen dem Attackieren eines anderen Charakters oder dem Benutzen eines Infinity Stones. Im Falle dessen, dass der Spieler erstes wählt, kann ein Opfer gewählt werden, dass wenn sichtbar und in Reichweite Schaden bekommt. Entscheidet der Spieler sich hingegen für zweiteres, so muss er in einem Dialog einen der Infinity stones in seinem Besitz auswählen und benutzen. Egal, was passiert, am Ende werden die Action Points reduziert.

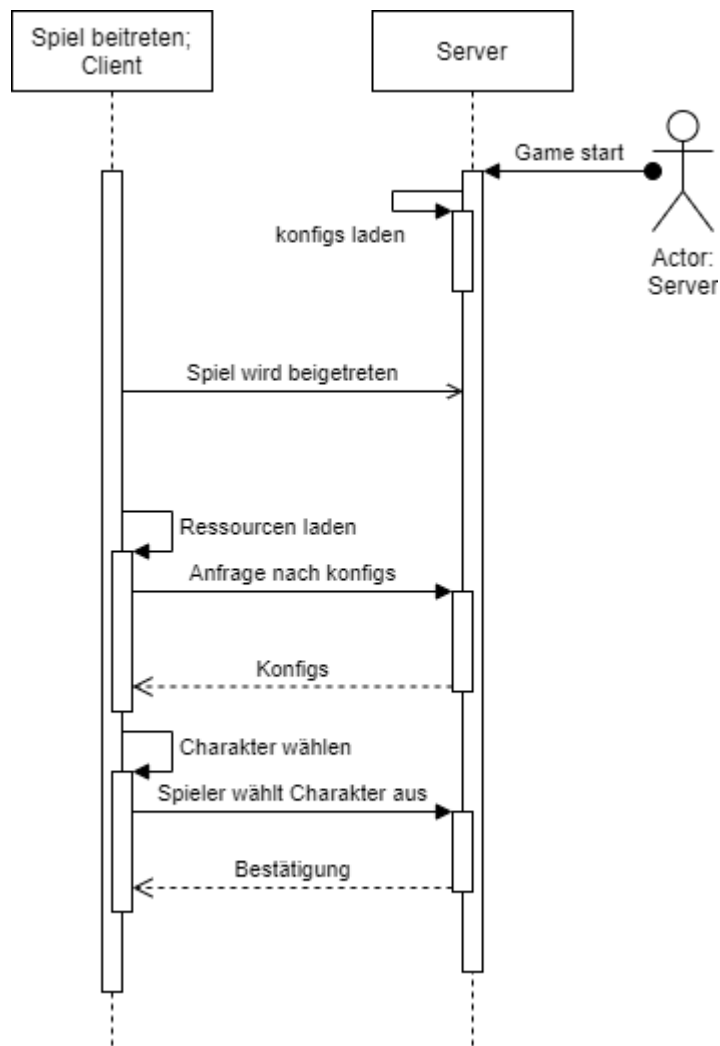
Nach der Ausführung von Sowohl eines Movement Points als auch eines Action Points hat der Spieler die Möglichkeit weitere Action und Movement Points zu verwenden.

## Hauptmenü:



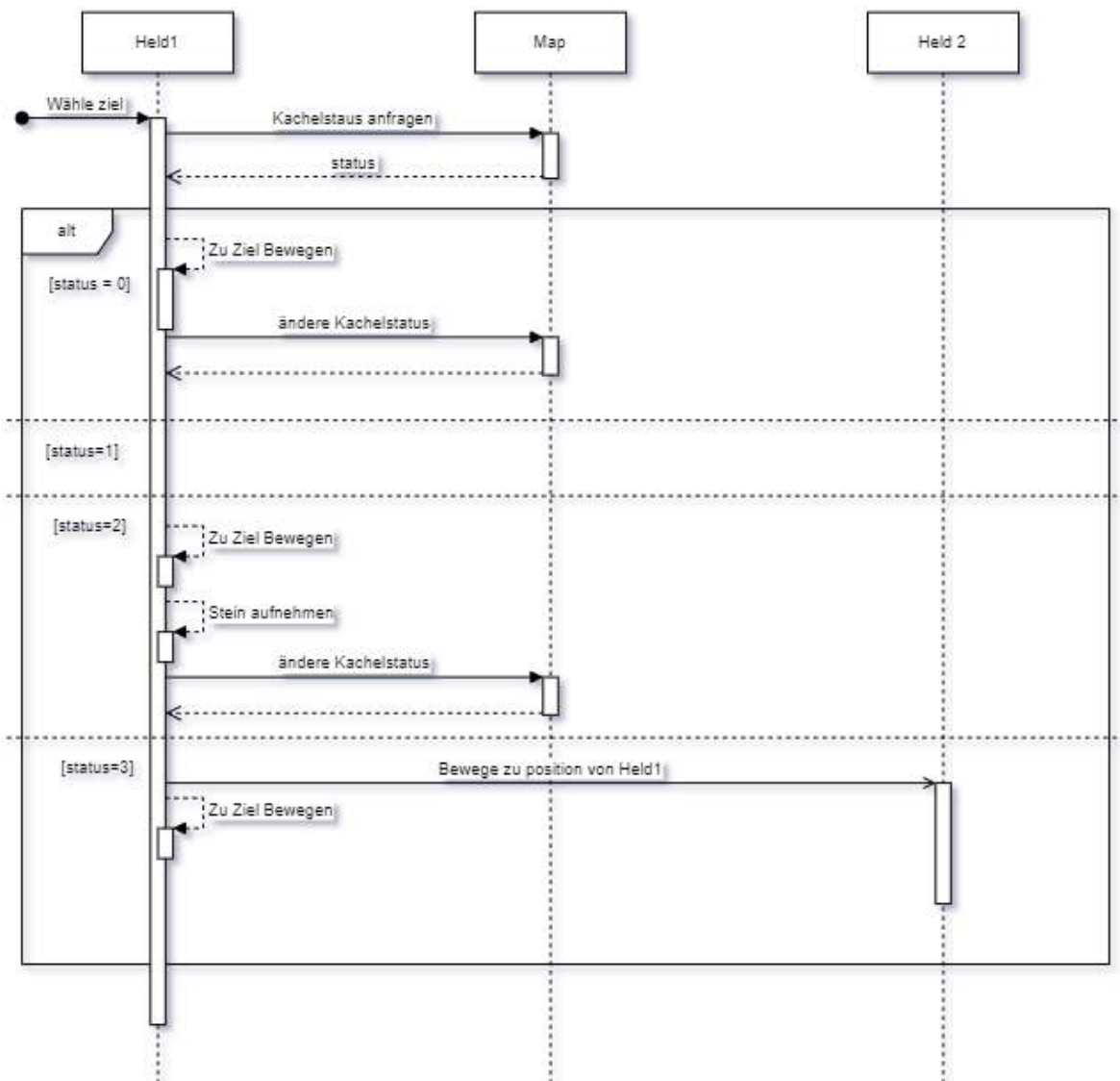
Im Hauptmenü hat der Spieler durch klicken der linken Maustaste die Möglichkeit auszuwählen zwischen den Optionen ein Spiel zu erstellen oder zu erstellen, ein Spiel auszuwählen, oder in den Editor zu wechseln.

## Sequenzdiagramme:



Der Spieler tritt dem Spiel bei und überträgt seinen Namen in diesem Zug an den Server. Wenn ein Spieler einem bereits laufenden Spiel beitrifft, sind alle Konfigurationsdateien bereits geladen. Der Client empfängt diese dann über das Netzwerk vom Server und lädt sie selbst. Anschließend kann vom Spieler ein Charakter ausgewählt werden, diese Auswahl wird vom Client an den Server weitergegeben, welcher dann die Auswahl bestätigt, sollte sie valide sein.

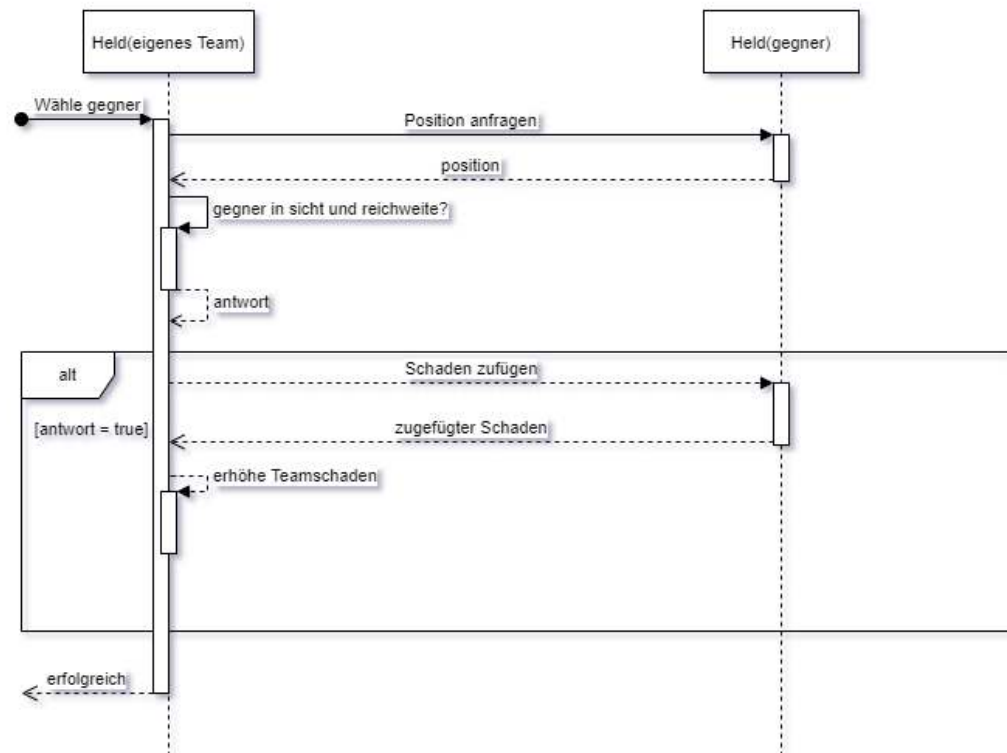
Player bewegt sich



<b>status</b>	Status ist ein ganzzahliger wert der eindeutig beschreibt was sich auf dem Feld befindet:	0 = nichts 1 = Stein 2 = Infinity Stone 3 = Held
---------------	---	---

Ein Spieler wählt eine Kachel aus und es wird nun bestimmt was sich in dieser Kachel (oder auch Feld genannt) befindet und dementsprechend wird entschieden was nun passiert.

### Player attacks enemy

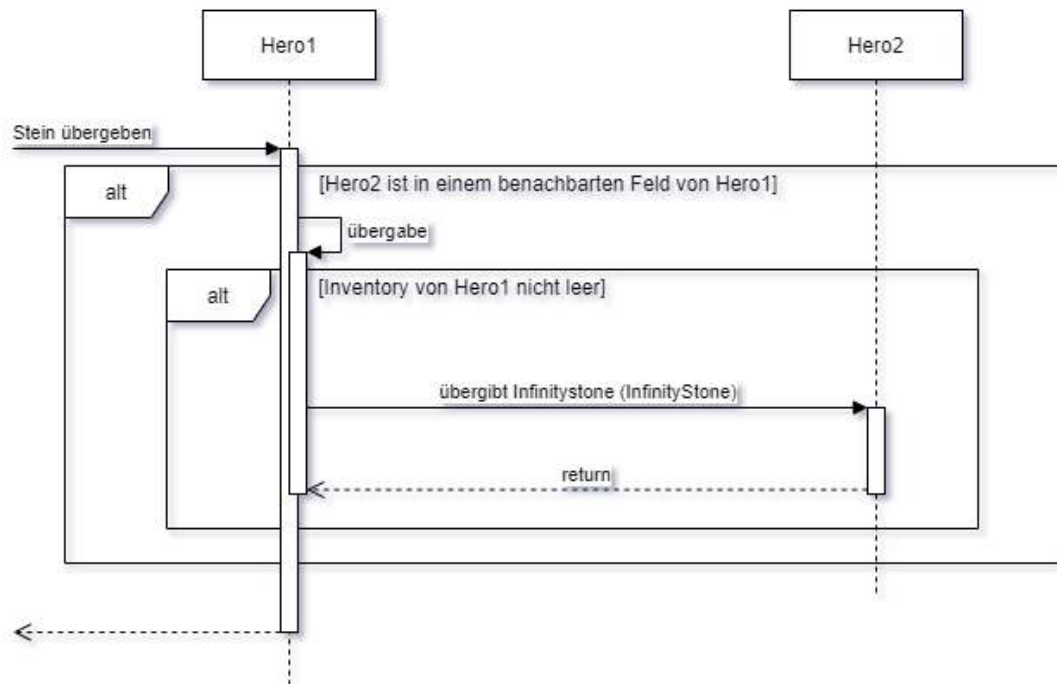


<b>Teamschaden</b>	Es wird gespeichert wieviel Schade der Spieler am ende der Partie insgesamt zugefügt hat, dieser wert wird auch als tiebreaker verwendet	
<b>return</b>	Der Wert ob der Angriff erfolgreich war wird returned	true = Angriff war erfolgreich false = Angriff ist fehlgeschlagen

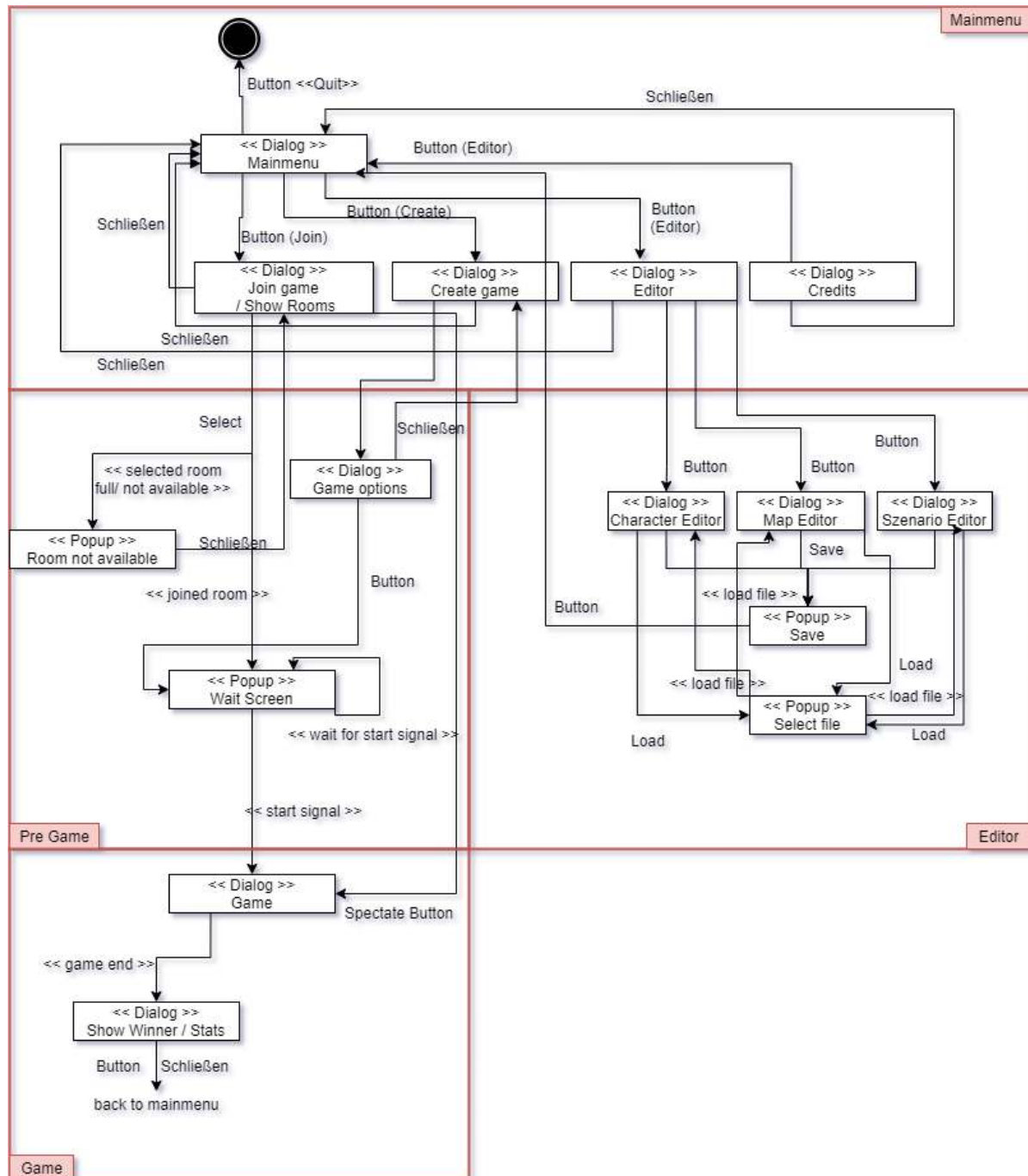
Ein Player wählt mit einem Charakter einen Gegner aus. Wenn dieser in Reichweite ist, wird ihm Schaden zugefügt und der Teamschaden wird erhöht.



Held übergibt Infinitystone



## Dialogstruktur



### **Mainmenu**

Das Hauptmenü navigiert den Nutzer durch die verschiedenen Menüeinträge. Durch die Einträge "Join Game", "Create Game" und "Editor" gelangt man in eine tiefere Auswahlebene des ausgewählten Eintrags. Credits zeigt eine Auflistung der Personen an, die an dem Projekt mitgewirkt haben.

### **Pregame**

Der Pregame Abschnitt beschreibt das Dialogverhalten bis zum Beitritt in das Spiel. Dorthin gelangt man durch das Erstellen eines neuen Raumes der zuerst konfiguriert werden muss, oder durch das Auswählen eines existierenden Raumes. Vom Raum Auswahldialog kann auch die "Spectate" Taste gedrückt werden, wodurch man ein laufendes Spiel beobachten kann. Erst wenn genügend Spieler (2 Spieler, oder 1 Spieler, 1 KI) verbunden sind und das Startsignal erteilt wurde, gelangt man zum Spielszenario, bei welchem das eigentliche Spiel angezeigt wird und gespielt werden kann.

### **Editor**

Im Editor Abschnitt hat man die Möglichkeit Character- Map und Szenario Editoren zu öffnen in welchen die jeweiligen Konfigurationsmöglichkeiten angezeigt werden, mit denen der Nutzer die Aspekte anpassen kann. Die Schaltflächen "Load" und "Save" ermöglichen ein abspeichern und bearbeiten von Konfigurationen.

### **Game**

Im Spielszenario kann der Nutzer mit dem Hauptspiel interagieren und seinem Charakter Anweisungen geben. Am Ende des Spiel wird der Gewinner angezeigt.

## Graphische Gestaltung und Nutzungskonzept

Hauptmenü:



Farben:

- |                      |         |
|----------------------|---------|
| • Rot (Marvel Logo): | #F0131E |
| • Gelb:              | #FECB02 |
| • Schwarz:           | #FFFFFF |
| • Weiß:              | #000000 |



Das **Hauptmenü** navigiert den Nutzer über die verschiedenen Menüeinträge in die verschiedenen Bereiche des Spiels.

Die Einträge "Join Game", "Create Game" und "Editor" sind die zentralen Bereiche und sollen priorisiert werden gegenüber den anderen Einträgen. Daher sind diese Buttons größer als die restlichen Buttons.

Durch die Einträge "Join Game", "Create Game" gelangt man zu weiteren Darstellungen, welche es ermöglichen ein Spiel zu spielen.

Der Eintrag "Editor" öffnet den Editor, welcher auch ein sehr zentraler Aspekt des Spiels ist.

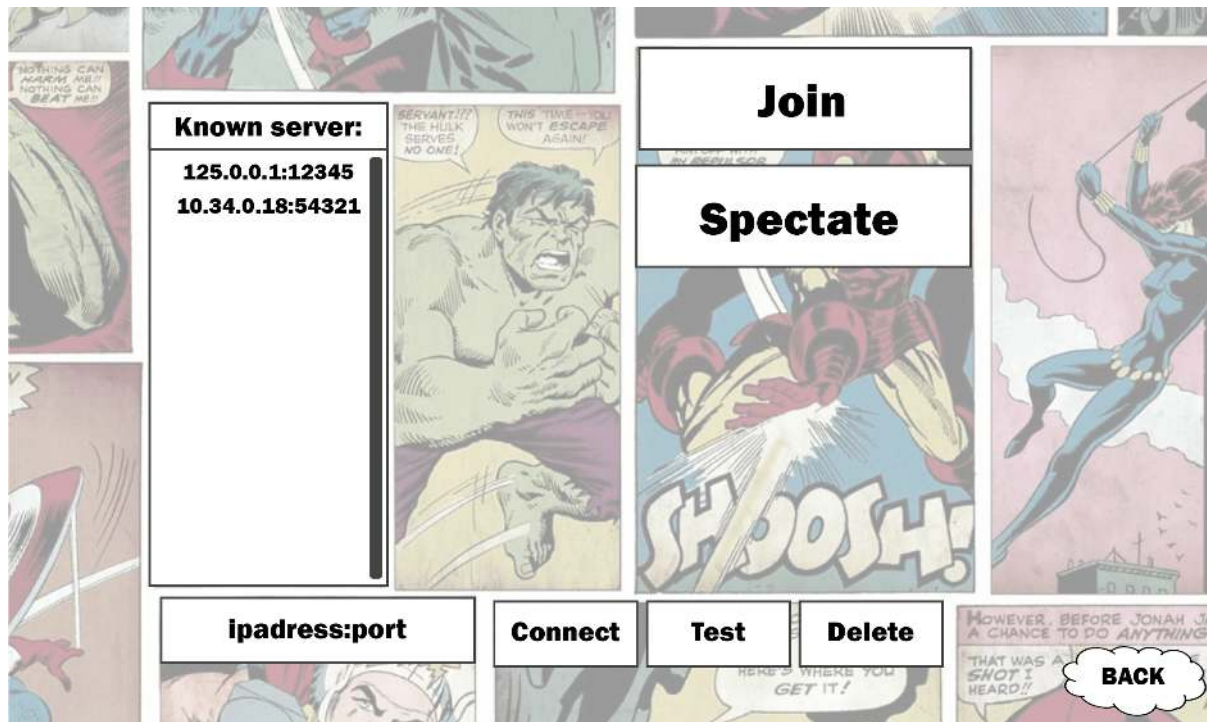
Über "Settings" kommt man in die Einstellungen des Spiels. Es gibt hier noch kein Konzept, welche Einstellungen angepasst werden können. "Credits" zeigt eine Auflistung der Personen an, die an dem Projekt mitgewirkt haben. "Quit" beendet die Anwendung.

Das Design ist sehr einfach gehalten um den Nutzer nicht zu überfordern. Oben in der Mitte ist das Logo zu sehen. Das Logo ist im Stil einer Action bubble aus Comics übernommen. Das Rot der Schrift ist die Farbe des Marvel Logos.

Die Auswahlknöpfe sind auch in diesem Rot.

Der Hintergrund zeigt schwach ein Comic an. Das macht das Hauptmenü weniger langweilig.

Join Game:



Im **Join Game** Menü kann der Nutzer einem Spiel beitreten. Hier kann eine Kombination aus IP-Adresse und Portnummer entweder manuell eingegeben werden, oder aus der Liste der bekannten Server eingefügt werden.

Der Nutzer kann sich nun zu dieser IP über den Button "Connect" verbinden. Er kann auch die Verbindung überprüfen mit dem Button "Test". Dieser verbindet den Client mit dem Server und baut daraufhin die Verbindung wieder ab. Mit dem Button "Delete" wird die IP aus der Liste gelöscht.

Der Button "Back" navigiert den Nutzer wieder zurück ins Hauptmenü.

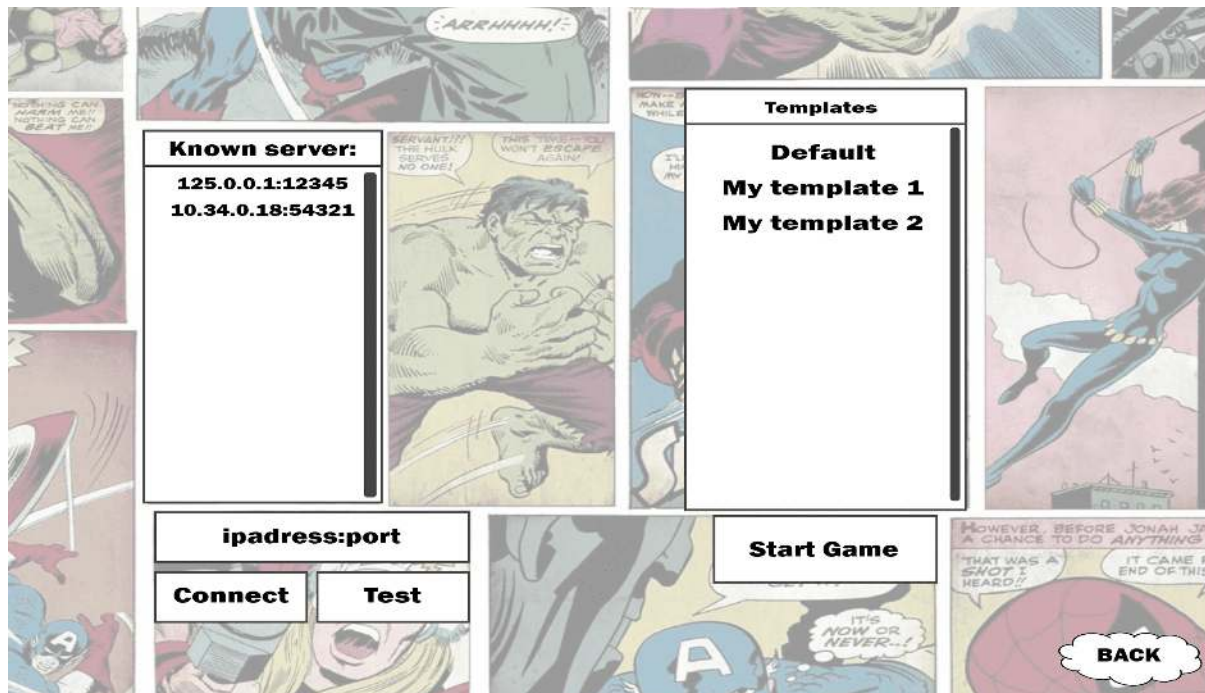
Sobald der Client mit einem Server verbunden ist, kann der Nutzer nun über die beiden verbleibenden Buttons "Join" und "Spectate" dem Spiel beitreten oder zuschauen.

Im Hintergrund befindet sich weiterhin ein Auszug aus einem Marvel Comic.

Die einzelnen Buttons, Rechtecke und Fenster sind an die Konturen des Comics angepasst um sich in den Hintergrund einzufügen. Der Ausschnitt des Comic soll nicht der selbe sein wie der im Hauptmenü.



## Create Game:



Im **Create Game** Menü kann der Nutzer ein Spiel erstellen. Hier kann eine Kombination aus IP-Adresse und Portnummer entweder manuell eingegeben werden, oder aus der Liste der bekannten Server eingefügt werden um sich mit dem Server, welcher das Spiel hosten soll zu verbinden.

Der Nutzer kann sich nun zu dieser IP über den Button "Connect" verbinden. Er kann auch die Verbindung überprüfen mit dem Button "Test". Dieser verbindet den Client mit dem Server und baut daraufhin die Verbindung wieder ab.

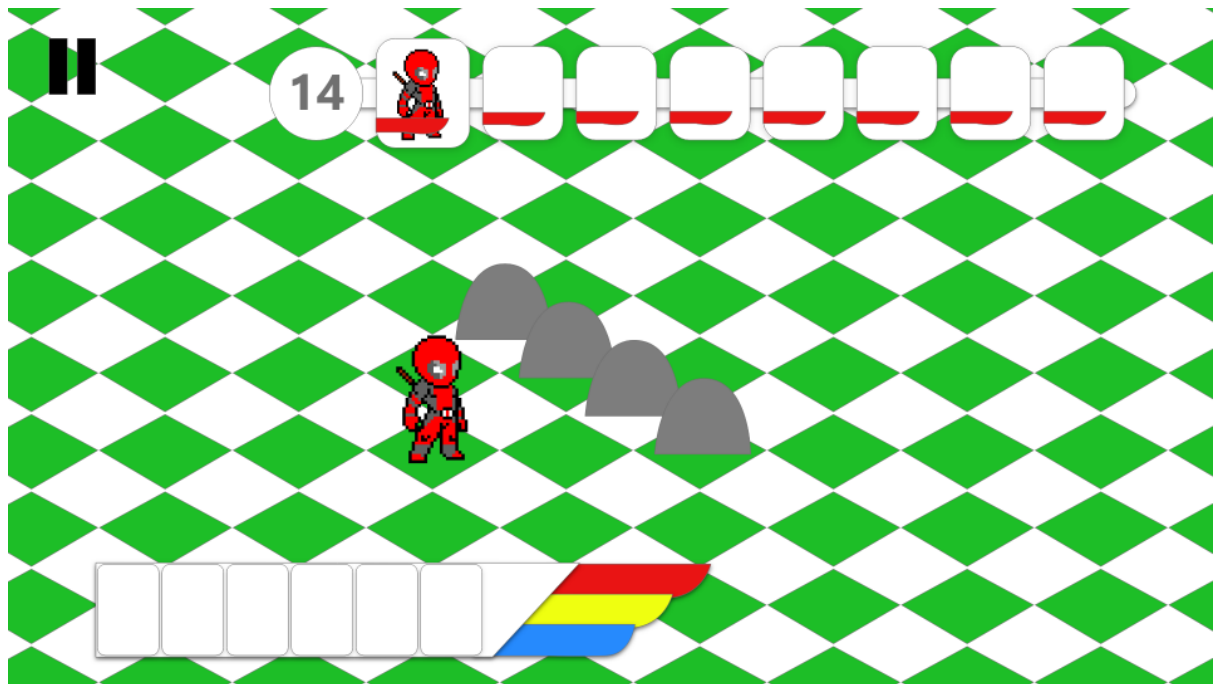
Der Button "Back" navigiert den Nutzer wieder zurück ins Hauptmenü.

Sobald der Client mit einem Server verbunden ist, kann der Nutzer nun über die die Liste an Vorlagen eine Konfiguration für das Spiel auswählen. Hit dem "Start Game" Button wird die Konfiguration bestätigt und das Spiel wird im Server gestartet.

Im Hintergrund befindet sich weiterhin ein Auszug aus einem Marvel Comic.

Die einzelnen Buttons, Rechtecke und Fenster sind an die Konturen des Comics angepasst um sich in den Hintergrund einzufügen. Der Ausschnitt des Comic soll nicht der selbe sein wie der im Hauptmenü.

Game:

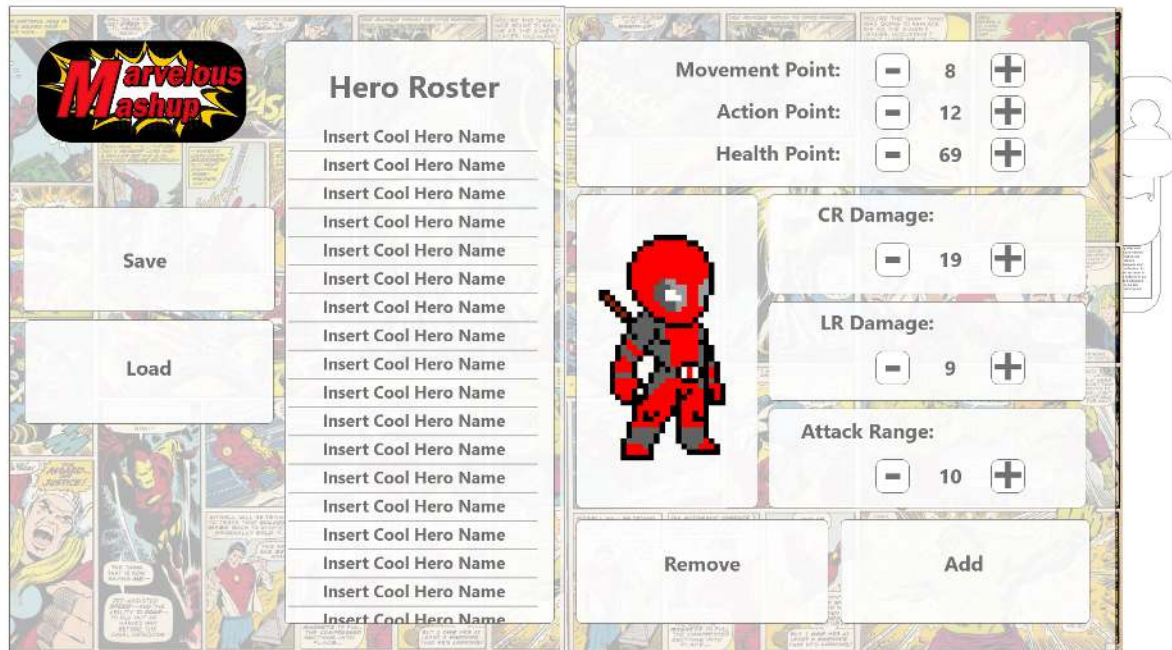


Im **Spiefenster** werden dem Spieler alle nötigen Informationen gezeigt, um das Spiel zu spielen. Der Bildschirm wird von der Spielfläche, der vom Szenario bestimmten Map, ausgefüllt, auf der die Charaktere sich durch simples Klicken bewegen können. Am oberen Rand wird dem Spieler präsentiert, wieviele HP der momentan aktive Charakter hat und welche Charaktere als nächstes ihren Zug haben und wieviele HP diese haben. Links an der "Zeitleiste" ist die Anzahl des momentan Zugs. Unten links wird dem Spieler das Inventar des momentan aktiven Charakters angezeigt, sowie 3 Balken: der rote Balken stellt die HP dar, der gelbe die AP und der blaue die MP dieses Charakters für diese Runde. Oben links ist noch ein Menüknopf.



Editor:

Character Editor:

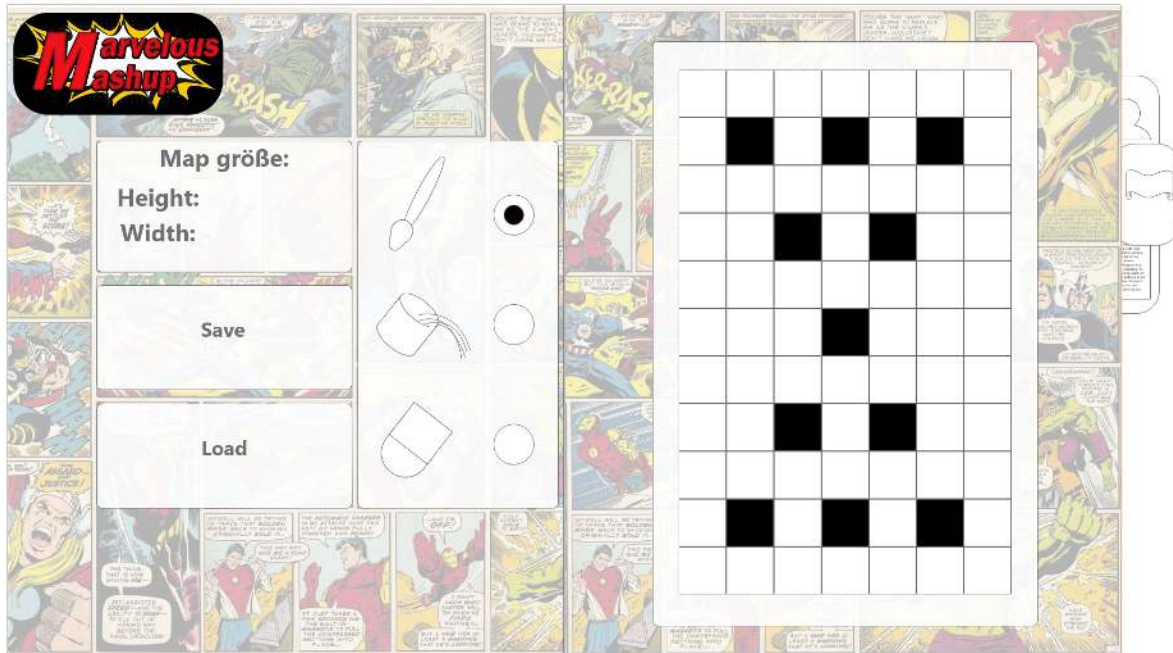


Im **Character Editor** kann der Nutzer eine ganze Reihe an Charakteren erstellen und editieren welche später spielbar sein werden. Der Nutzer kann auch einzelne Charaktere, welche zuvor hinterlegt wurden, in die Liste an, in diesem Szenario verfügbaren, Helden/Charakteren laden.

Das Design soll an ein Comic Buch erinnern. Die einzelnen Fenster der UI sind an die Rechtecke des Comics angelehnt. Die linke Seite listet alle möglichen Charaktere auf, die auf der rechten Seite nach eigenen Wünschen angepasst werden können.

Außerdem ist an der Seite mit einer Markierung die Seiten markiert die zu den anderen Bestandteilen des Editors führen. Diese können mit einem Klick ausgewählt werden.

Map Editor:

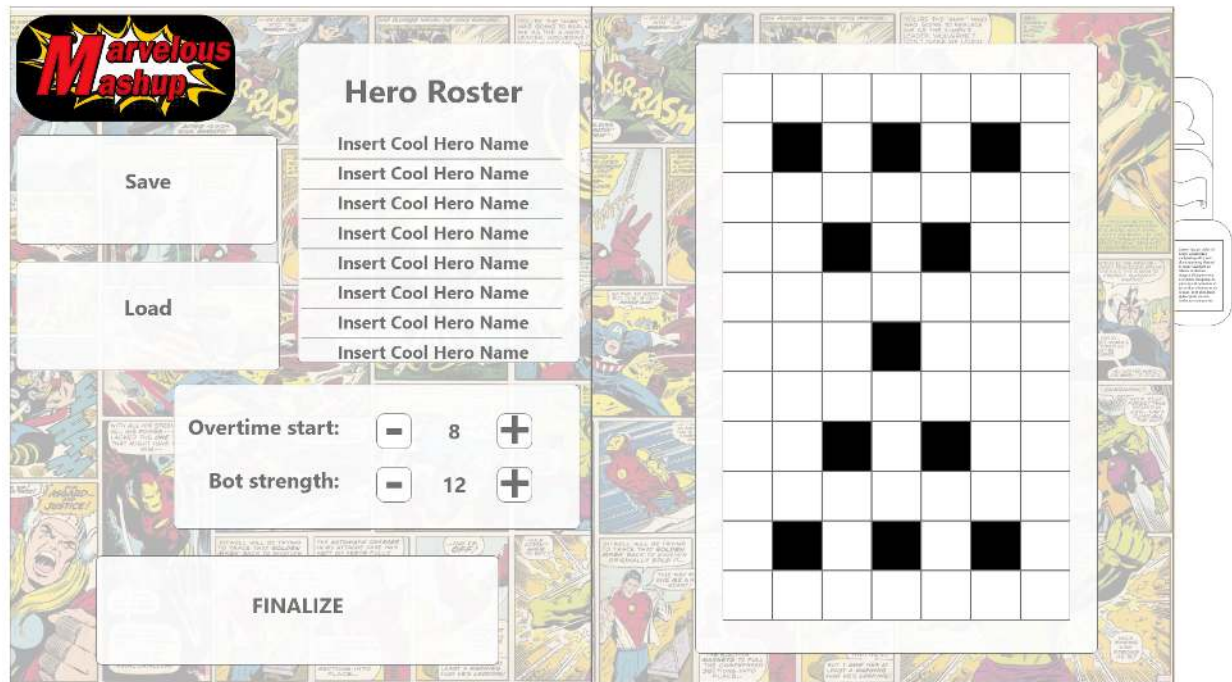


Im **Map Editor** kann der Nutzer Felsen auf das Spielfeld zeichnen und entfernen und somit eine eigene Map erstellen auf welcher dann im Szenario gespielt wird. Es wird dem Nutzer auch freigestellt vorherig gespeicherte Maps zu laden oder die neu von ihm erstellte zu hinterlegen.

Das Design soll an ein Comic Buch erinnern. Die einzelnen Fenster der UI sind an die Rechtecke des Comics angelehnt. Die Rechte Seite dient hier als Vorschau für das Spielfeld und die linke Seite ist für die Interaktion gedacht.

Außerdem ist an der Seite mit einer Markierung die Seiten markiert die zu den anderen Bestandteilen des Editors führen. Diese können mit einem Klick ausgewählt werden.

Szenario Editor:



Im **Szenario Editor** kann der Nutzer sich die Liste an Verfügbaren helden und die Map nochmal ansehen sowie die Overtime der runden und die Stärke der eventuellen KI einstellen. Dem Nutzer steht es auch frei Komplettes Szenario zu laden das zuvor Hinterlegt wurde und kann das nun von ihm erstellte auch hinterlegen.

Das Design soll an ein Comic Buch erinnern. Die einzelnen Fenster der UI sind an die Rechtecke des Comics angelehnt. Die Rechte Seite dient hier als Vorschau für das Spielfeld und die linke Seite lässt generelle Spielinformationen einstellen.

Außerdem ist an der Seite mit einer Markierung die Seiten markiert die zu den anderen Bestandteilen des Editors führen. Diese können mit einem Klick ausgewählt werden.

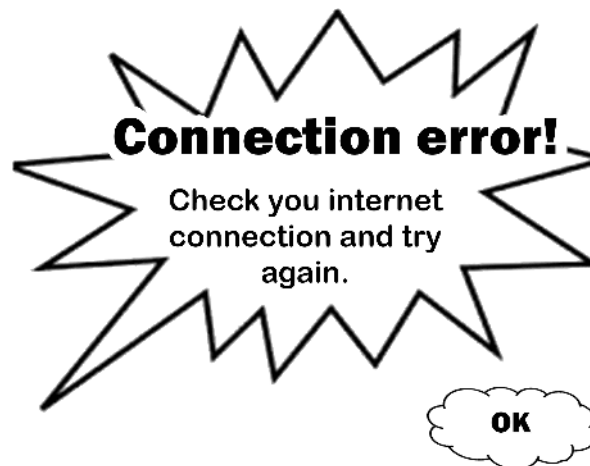
## Dialoge:

Die **Dialoge** teilen sich in 2 verschiedene Arten von Dialogen auf.

Es gibt Alert Dialoge die den Nutzer auf etwas hinweisen. Diese Dialoge haben nur einen Knopf um den Dialog wieder zu schließen. Diese Dialoge sind meistens Error Dialoge. Das bedeutet, der Spieler wird auf einen Fehler aufmerksam gemacht.

Die andere Art sind Auswahldialoge. Bei diesen Dialogen hat der Nutzer mehrere Optionen aus denen er auswählen kann.

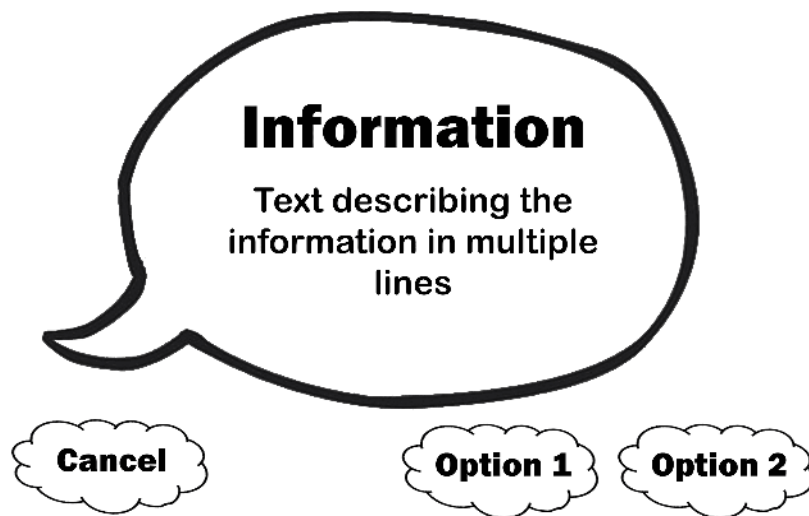
### Alert Dialog:



Der **Alert Dialog** besteht aus einer Action Bubble mit einem Titel und einer Beschreibung. Titel und beschreibung befinden sich innerhalb der Action Bubble und nennen den Grund für den Dialog. Diese dürfen auch über die Bubble hinausgehen.

Der Button um den Dialog wieder zu schliessen befindet sich unterhalb der Action Bubble und besteht aus einer eigenen Thinking Cloud mit dem Text "OK"

Info Dialog:



Der **Info Dialog** lässt dem Nutzer eine Auswahl wie er fortfahren möchte.

Er besteht aus einer Speech Bubble die den Text beinhaltet. Der Text hat auch eine Überschrift und eine Beschreibung. Beide dürfen nicht über die Speech Bubble hinausgehen.

Unter der Bubble befinden sich weitere Thinking Clouds über die der Nutzer auswählen kann wie er fortfahren will. Standardmäßig gibt es eine zum abrechnen und 2 für weitere Optionen.

## Rückmeldung:

Um dem Nutzer **Rückmeldung** zu geben, wenn auf etwas gewartet wird, gibt es unten rechts im Eck eine Anzeige dafür. Dadurch ist der Nutzer nicht verwirrt wenn eine Aktion ungewöhnlich lange dauert.

Es gibt hier 2 verschieden Arten von Wartezeiten. Wenn der Client auf eine Nachricht von einem Server wartet, und wenn eine Berechnung ungewöhnlich lange dauert.

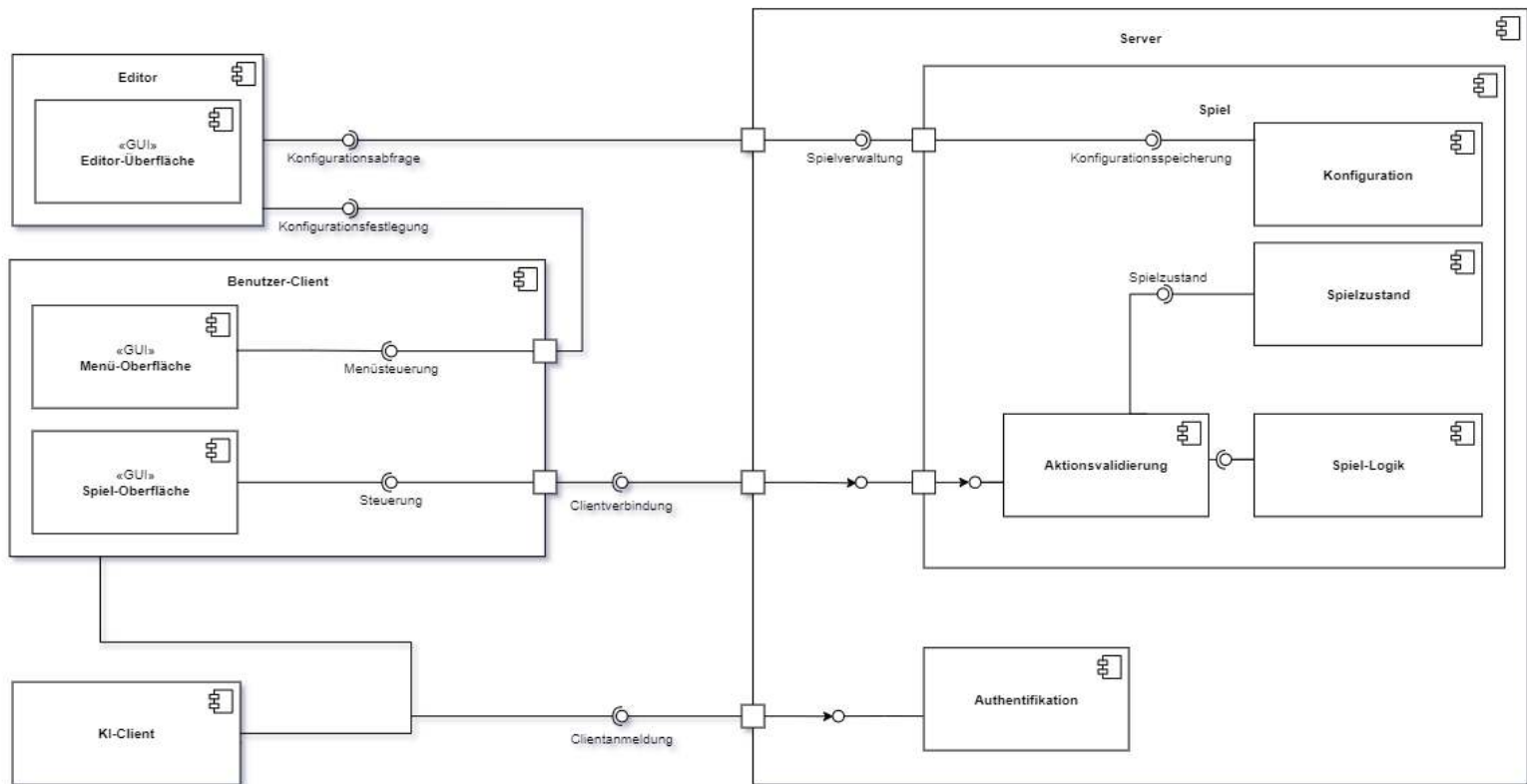


## Server und KI Client:

Im Gegensatz zum Client bekommt der Nutzer selbst das Serverprogramm und den KI-Client nicht zu Gesicht. Um hier Ressourcen bei der Entwicklung zu sparen, wird der KI-Client keine graphische Oberfläche haben. Aus demselben Grund, und der Tatsache, dass Docker keine graphische Oberfläche in diesem Sinne unterstützt, wird auch der Server über die Kommandozeile gesteuert werden.

Meilenstein 5:

## Komponentendiagramm:



### Editor:

Die Komponente ist essentiell um individuelle Konfigurationen zu erstellen, die vom Spiel interpretiert werden. Diese können vom Server angefordert werden.

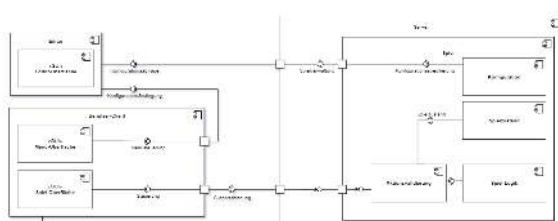
FAs: FA5

### Editor-Oberfläche:

Die Oberfläche ist der Interaktionsbereich des Nutzers um den Editor zu steuern. Er hat eine eigene Komponente, da die Daten vom Nutzer eingegeben werden.

Die an der Oberfläche getätigten Eingaben werden vom Editor verarbeitet und bei einer Konfigurationsabfrage an den Server geschickt, wo die Dateien weiter interpretiert werden.

FAs: FA5





### Benutzer-Client:

An dieser Komponente spielt sich die Eingabe des Spielers ab, heisst der Nutzer steuert was gemacht werden soll. Bei einer Front- & Backend Architektur handelt es sich hierbei um das Frontend.

FAs: FA1, FA3, FA11, FA34

### Menü-Oberfläche:

Hier steuert der Nutzer über einzelne Menüpunkte. Er kann sich durch Menüs bewegen, den Editor starten und Menüpunkte auswählen.

FAs: FA25, FA35, FA36, FA37, FA38, FA40, FA44, FA45

### Spiel-Oberfläche:

Der Spieler steuert hier verschiedene Aktionen des Spiels. Die Anweisungen werden verarbeitet und dann als Aktionen an den Server geschickt.

FAs: FA9, FA10, FA15, FA17, FA19, FA20, FA21, FA22, FA23, FA24, FA 26, FA27, FA28, FA29, FA32, FA33, FA39, FA43

### KI-Client:

Client Anwendung die über einen Docker-Container läuft und Computergesteuert auch Anweisungen an den Server schickt.

FAs: FA4

### Server:

Der Server hostet ein Spiel. Er authentifiziert dabei Clients. Außerdem kann er danach auch über Verbindungen Nachrichten entgegennehmen und an das Spiel weiterleiten. Hier werden die Verbindungen auch verwaltet.

FAs: FA1, FA2, FA8, FA16, FA18, FA30, FA31, FA32, FA41, FA42

### Spiel:

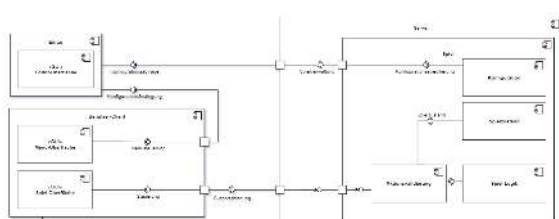
Das Spiel stellt die zentrale Spielsteuerung da. Hierbei werden alle Informationen verwaltet, die mit dem Spiel selber zu tun haben. Client Nachrichten und beliebige Konfigurationsdateien werden entgegengenommen und entsprechend verarbeitet.

FAs: FA14, FA15

### Authentifikation:

Hier verwaltet der server anmeldungen der Spieler und KI clients. Angemeldete Spieler werden mit mit einer bestimmten ID versehen um sie eindeutig unterscheiden zu können.

FAs: FA37, FA41, FA42, FA45



**Konfiguration:**

Umfasst die 3 Jasons, welche den Spielverlauf und die Helden definieren. Diese werden im Editor erstellt und bearbeitet. Sie werden vom Server geladen und können danach beliebig gelesen werden.

FAs: FA15

**Spielzustand:**

Der Spielzustand ist eine Speicherung des aktuellen Spiels und allen zugehörigen Werten. Er kann erst nach einer Authentifizierung der Nachricht verändert werden.

FAs: FA15

**Aktionsvalidierung:**

Jede einkommende Nachricht wird hier auf Korrektheit überprüft. Dadurch kann keine unmögliche, invalide Aktion ausgeführt werden. Wenn eine Nachricht als OK eingestuft wird, wird der Spielzustand dementsprechend verändert und eine Nachricht an alle Clients geschickt.

FAs: FA32

**Spiel-Logik:**

Die Logik umfasst alle im Spiel vorhandenen Vorgänge wie zum Beispiel den Helden bewegen oder Schaden zufügen. Jede von Clients Aktion wird von der Spiellogik erfasst und ausgeführt.

FAs: FA32

## Artefakte:

### Charakter Konfiguration:

characterConfig.json

### Partie Konfiguration:

partie-konfig.json

### Szenario Konfiguration:

*szenario-name*.scenario, "*szenario-name*" muss ersetzt werden