ulm university universität
uulm

Fakultät für Ingenieurwissenschaften, Informatik und Psychologie
Institut für Softwaretechnik und Programmiersprachen

Pflichtenheft Marvelous Mashup

Softwaregrundprojekt 2020/2021

Team 15:
Jannis Dommer
Julia Drozd
Daniel Klier
Valentin Kolb
Lars Licha
Michel Lutz

Tutor:
Jakob Meyer-Hilberg

Version: 1.1 Stand: 17. Februar 2021

Inhaltsverzeichnis

1 Überblick **1**

1.1 Einleitung 1
1.2 Motivation 2
1.3 Vision 2
1.4 Projektkontext 3

2 Anforderungsanalyse **4**

2.1 Fachwissen 4
2.2 Anwendungskontext 13
Akteure 13
2.3 Funktionale Systemanforderungen 15
Server 16
Game-Engine 18
Benutzer-Client 20
KI-Client 26

3 Softwarespezifikationen **29**

3.1 Funktionen 29
Spielablauf Übersicht 29
Spielablauf Benutzer-Client 31
Spielablauf Server 32
3.2 Server 33
Schnittstelle 33
Nutzungskonzept 33
3.3 Benutzer-Client 34
Schnittstelle 34
Nutzungskonzept 41

3.4 Editor 43
Schnittstelle 43
Nutzungskonzept 47
3.5 KI-Client 48
3.6 Datenmodell 49

4 Randbedingungen **50**
4.1 Qualität 50
4.2 Betriebskonzept 55
Systemumgebung 55
Abhängigkeiten von Produkten Dritter 55
Schulungskonzept 56
4.3 Entwicklungsvorgaben 57
4.4 Abnahmekriterien 58

1 Überblick

1.1 Einleitung

Das Ziel des Projektes Marvelous Mashup ist die Entwicklung und Implementierung des gleichnamigen Spiels. Marvelous Mashup ist ein Rundenbasierten-Taktik-Online-Multiplayer Spiel (RBTOMG), für den Computer, bei dem der Spieler sowohl allein, gegen den Computer, als auch gegen einen anderen Spieler über das Internet spielen kann. Das Projekt wurde von der Universität Ulm in Auftrag gegeben und richtet sich primär an Studierende der Studiengänge Informatik, Software-Engineering und Medien-Informatik. Allerdings soll Marvelous Mashup auch weiteren begeisterten Spielern auf der ganzen Welt Freude bereiten. Das Projekt gliedert sich in die Komponenten Server, Game-Client, KI-Client und Editor, die eigenständig entwickelt und implementiert werden. Das hier vorliegende Dokument definiert alle Anforderungen, Softwarespezifikationen und alle Lasten und Pflichten, die mit der Entstehung von Marvelous Mashup assoziiert sind. Hingewiesen soll an dieser Stelle auf das Standardisierungsdokument, das weitere Netzwerk- und Spieldetails bereitstellt. Wir wünschen allen Lesern bei der Erkundung der heldenhaften Welt von Marvelous Mashup viel Spaß.

1.2 Motivation

> "Denn, um es endlich einmal herauszusagen, der Mensch spielt nur, wo er in voller Bedeutung des Wortes Mensch ist, und er ist nur da ganz Mensch, wo er spielt."
>
> *Friedrich Schiller*

Marvelous Mashup soll ein unterhaltsamer, spaßiger Zeitvertreib für Groß und Klein sein, der den Spielern unvergessliche Stunden in der phantastischen Welt der Marvel-Comics beschert und das nicht nur allein, sondern mit Freunden, Corona-konform von Zuhause aus. Dabei erreicht der Auftragsgeber, die Universität Ulm, durch das Projekt, dass ihre Studierenden den Umgang, Planung, Entwicklung und Implementierung größerer Softwareprojekte erlernen. Dadurch soll sich ihr Verständnis der Software-Entwicklung vertiefen und sie auf ein späteres Berufsleben als Software-Entwickler vorbereiten. Neben diesem edukativen Aspekt, konkurrieren mehrere voneinander unabhängige Entwicklerteams darum das beste Spiel anzufertigen. Durch diesen Wettstreit der KIs gegeneinander bekommt das Projekt einen zusätzlichen Wettbewerbscharakter.

1.3 Vision

Das Entwicklerteam und die Auftraggeber haben bei Marvelous Mashup ein Spiel im Sinne, das leichte und doch taktisch tiefgehende Unterhaltung für einen oder zwei Spieler bietet. Eine Marvelous Mashup Partie findet auf einem zwei dimensionalen Spielfeld statt, auf dem zwei Teams mit je sechs Superhelden gegeneinander antreten. Ziel des Spiel ist es dabei die sechs Infinity-Steine einzusammeln und somit die Partie zu gewinnen bevor Thanos, der verrückte Titan mit dem großen Kinn, auftaucht und damit beginnt ebenfalls die Steine zu sammeln und damit die Helden, und 50% der Bevölkerung des Universums, zu vernichten. Das fertige Spiel gliedert sich in vier Module. Diese sind der Server, der Editor, der Spielerclient und der KI-Client. Der Server verwaltet das laufende Spiel und kommuniziert mit den Spielcli-

ents. Dabei gibt es zwei aktive Spieler, die sowohl menschlich als auch artifiziell sein können. Des Weiteren kann eine beliebige Anzahl an Clients einem laufenden Spiel als Zuschauer beiwohnen. Diese erhalten dann vom Server alle nötigen Informationen über die laufende Partie um diese, ähnlich den Spielern, darzustellen. Der Server verwaltet nicht nur die Spieler und Zuschauer, sondern überwacht und verwirklicht die Spiellogik. Die Game-Clients stellen das Fenster der Spieler zur Welt von Marvelous Mashup dar. Er erlaubt es sowohl als aktiver Spieler eine Partie zu spielen, als auch als passiver Zuschauer einer bereits laufenden Partie beizuwohnen. Der Client stellt das Spiel in einer ansprechenden und aufregenden graphischen Benutzer-Oberfläche dar und nimmt die Befehle des Spielers entgegen, überprüft sie auf ihre Gültigkeit und übermittelt sie an den Server. KI-Clients sind vom Computer gesteuerte Gegner, die Spielern eine Partie auch ohne einen menschlichen Mitspieler ermöglichen. Die KI soll dem Spieler dabei ein ebenso aufregendes und anspruchsvolles Spielerlebnis wie ein menschlicher Gegner bieten. Der Editor, dient zur Konfiguration des Spiels, durch die Erzeugung von Spielkarten und Superhelden, ganz nach dem Geschmack der Spieler.

1.4 Projektkontext

Das Projekt wird vom Institut für Softwaretechnik und Programmiersprachen der Universität Ulm organisiert. Neben unserem Entwicklerteam beteiligen sich weitere Gruppen an der Entwicklung, wobei diese eigenständige Entwürfe vorlegen. Das Institut kommt dabei seinem edukativen Auftrag nach, den Studierenden einen praktischen Exkurs in die Entwicklung und Implementation eines größeren Softwareprojekts zu ermöglichen. Selbstredend kann und soll das Spiel jedoch auch im Nachgang von weiteren Begeisterten gespielt werden. Die Projektaufsicht führt dabei Florian Ege vom Institut für Softwaretechnik und Programmiersprachen. Das Projekt könnte in Zukunft derart erweitert werden, dass ein Ranking-System die Leistung der Spieler bewertet und transparent zur Verfügung stellt. Ein internes Belohnungssystem könnte den Spielern, beim Aufstieg innerhalb der Ranglisten besondere Erfolge, Titel oder Skins bereitstellen. Ein solches Ranking System würde den sportlichen Aspekt des Spiels hervorheben und es somit auch langfristig unterhaltsam gestalten.

2 Anforderungsanalyse

2.1 Fachwissen

Die folgenden Begriffe werden hier geklärt, um später Missverständnisse zu vermeiden.

Begriff	Held (Hero)
Beschreibung	Eine Figur aus allseits beliebten Comic-Heften, die durch eine Einheit auf dem Raster der Spielfelder repräsentiert wird
Ist-ein	bewegliche Einheit
Kann-sein	Player Character (PC), Non-Player Character (NPC)
Aspekt	Name, zugehöriger Spieler oder NPC, Eigenschaften,Health Points (HP), Inventar, Position auf dem Spielfeld, ...
Bemerkung	Der Begriff “Held” bezeichnet einen Superhelden, der auf dem Spielfeld herumläuft.
Beispiel	Tony Stark a.k.a. The Invincible Iron Man, gehört zur Heldengruppe von Benutzer Alice, Healthpoints: 100 Movement Points: 3 Action Points: 2 Inventar: Space Stone Position: x: 7, y: 18 etc.

Begriff	Team (Team)
Beschreibung	Wir verstehen darunter unser Sopra-Team
Ist-ein	Ansammlung von Menschen
Kann-sein	Chaotisch, Menschen
Aspekt	—
Bemerkung	—
Beispiel	Jannis, Julia, Lars

Begriff	Benutzer (User)
Beschreibung	Der Benutzer verwendet einen Benutzer-Client um das Spielgeschehen zu erfassen
Ist-ein	Mensch
Kann-sein	Zuschauer, Spieler
Aspekt	Verwendet einen Benutzer-Client
Bemerkung	—
Beispiel	—

Begriff	Zuschauer (Spectator)
Beschreibung	Kann das Spiel verfolgen, aber keine Züge ausführen oder in irgendeiner Weise das Spielgeschehen beeinflussen.
Ist-ein	Benutzer
Kann-sein	—
Aspekt	—
Bemerkung	Zuschauer sehen einer laufenden Partie nur zu. Sie können das Geschehen nicht beeinflussen.
Beispiel	—

Begriff	Spieler (Player)
Beschreibung	Nimmt aktiv an einer Partie teil und kann Spielzüge durchführen
Ist-ein	—
Kann-sein	Benutzer, KI
Aspekt	—
Bemerkung	Ein Spieler spielt eine Partie
Beispiel	—

Begriff	KI (AI)
Beschreibung	Es gibt eine Künstliche Intelligenz, die am Spiel teilnehmen kann und sich nach außen wie ein menschlicher Spieler verhält.
Ist-ein	Spieler
Kann-sein	—
Aspekt	—
Bemerkung	KI spielt eine Partie gegen eine KI oder einen menschlichen Spieler. Die KI zeichnet sich durch ihre globale Strategie aus.
Beispiel	Skynet, HAL 9000, Deep Thought, Earth

Begriff	System (System)
Beschreibung	Benennt die Gesamtheit der Softwaremodule, also den Verbund aus Client, Server und Editor
Ist-ein	—
Kann-sein	—
Aspekt	—
Bemerkung	Ist aus Vollständigkeitsgründen definiert.
Beispiel	—

Begriff	Client (Client)
Beschreibung	Verbindet sich mit dem Server und kann zum Zuschauen oder Spielen verwendet werden.
Ist-ein	Eine Komponente des Systems, das entweder aktiv als Spieler an einer Partie teilnimmt oder als Zuschauer eine Partie beobachtet
Kann-sein	Benutzer-Client, KI-Client
Aspekt	—
Bemerkung	Der KI-Client kann natürlich nicht als Zuschauer agieren.
Beispiel	—

Begriff	Server (Server)
Beschreibung	Der Server ist nicht steuerbar. Er erhält Szenario- / Partie- / Charakter-Config und baut daraus das Spielfeld auf. Zudem verwaltet er alle Clients, die mit ihm verbunden sind und sorgt sich um die korrekte Umsetzung der Spielregeln. Er nutzt hierfür die Game-Engine und kümmert sich selbst primär um das Netzwerk.
Ist-ein	Komponente des Systems
Kann-sein	—
Aspekt	—
Bemerkung	—
Beispiel	—

Begriff	Benutzer-Client (User Client)
Beschreibung	Client, der von einem Benutzer verwendet wird.
Ist-ein	Client
Kann-sein	—
Aspekt	—
Bemerkung	Wird von einem Mensch gesteuert und kann somit in der Funktion als Zuschauer und menschlicher Spieler auftreten.
Beispiel	—

Begriff	KI-Client (AI Client)
Beschreibung	Ist ein spezieller Client, der nur von einer KI gesteuert werden kann.
Ist-ein	Client
Kann-sein	Spieler
Aspekt	globale und Zugstrategie
Bemerkung	Die KI kann gegen einen Spieler eine Partie spielen. Wird vom KI-Administrator gestartet
Beispiel	—

Begriff	Editor (Editor)
Beschreibung	Kann genutzt werden um Partie-/Charakter-/SzenarioConfig zu erstellen
Ist-ein	Ist eine Komponente des Systems
Kann-sein	—
Aspekt	—
Bemerkung	—
Beispiel	—

Begriff	Heldengruppe (Hero Group)
Beschreibung	Eine Heldengruppe setzt sich zusammen aus sechs Helden. Der Spieler stellt sich diese zusammen, wobei alle sechs Helden verschieden voneinander sind.
Ist-ein	Menge von Helden
Kann-sein	—
Aspekt	Menge aus sechs verschiedenen Helden
Bemerkung	—
Beispiel	Heldengruppe: Charakter Dr.Strange Charakter Hulk Charakter Iron-Man Charakter Spider-Man Charakter Thor Charakter Loki

Begriff	Partie (Match)
Beschreibung	Eine Partie ist definiert durch zwei Spieler, die gegeneinander spielen. Eine Partie enthält mehrere Runden und wird durch den Sieg eines Spielers oder durch frühzeitiges abbrechen beendet.
Ist-ein	—
Kann-sein	—
Aspekt	Spielfeld, Spielkonfiguration
Bemerkung	—
Beispiel	—

Begriff	Runde (Round)
Beschreibung	Eine Runde enthält Spielerzüge, die von Spielern oder KI durchgeführt werden, und Serverzügen.
Ist-ein	—
Kann-sein	—
Aspekt	—
Bemerkung	In einer Runde handen alle Helden der Spieler und die NPC Spielfiguren, wie Stan Lee, Goose und Thanos
Beispiel	—

Begriff	Serverzug (Server Move)
Beschreibung	Sind Handlungen die der Server ausführt pro Runde. Dabei steuert er NPCs.
Ist-ein	—
Kann-sein	—
Aspekt	Stan Lee, Goose, Thanos
Bemerkung	Serverzüge sind keine zwangsläufig auftretenden Events pro Runde, nur wenn den Spielregeln nach eine spezielle Aktion erforderlich ist
Beispiel	Goose kotzt Steine in Runde 1-6 Thanos kommt zurück und hat miese Laune

Begriff	Spielerzug (Player Move)
Beschreibung	Ein Zug enthält mehrere Zugphasen. Der Zug ist beendet, sofern der Spieler dies wünscht oder er alle mögliche Zugphasen ausgekostet hat.
Ist-ein	Teil einer Spielrunde
Kann-sein	—
Aspekt	Bewegung, Angriff (Nah- und Fernkampf), Aufnahme eines Infinity Stones, Übergeben eines Infinity Stones, Verwenden eines Infinity Stones
Bemerkung	—
Beispiel	—

Begriff	Zugphase (Move Phase)
Beschreibung	Jede Interaktion mit einem Helden wird als Zugphase bezeichnet.
Ist-ein	—
Kann-sein	Bewegung, Angriff (Nah- und Fernkampf), Aufnahme eines Infinity Stones, Übergabe eines Infinity Stones, Verwenden eines Infinity Stones
Aspekt	—
Bemerkung	Die Anzahl der Zugphasen wird limitiert durch die verfügbaren Action Points (AP) und Movement Points (MP) des Helden. Ein Spieler muss nicht alle mögliche Zugphasen ausnutzen, sondern kann auch früher seinen Spielerzug beenden.
Beispiel	Held Dr.Strange bewegt sich von Feld (1,1) auf Feld (1,2)

Begriff	Spielbrett (Game Board)
Beschreibung	Das Spielbrett enthält alle Felder, auf ihm findet dann das gesamte Spielgeschehen statt und die Züge werden auf dem Spielbrett durchgeführt.
Ist-ein	—
Kann-sein	—
Aspekt	Felder, Gras, Felsen, Helden, NPC
Bemerkung	Das Spiel findet auf dem Spiebrett statt.
Beispiel	—

Begriff	NPC (NPC)
Beschreibung	Non-Player Character: Spezielle Spielfiguren, die nicht von Spielern gesteuert werden können.
Ist-ein	Spielfigur
Kann-sein	Goose, Thanos, Stan Lee, Felsen
Aspekt	—
Bemerkung	—
Beispiel	Goose, Thanos, Felsen.

Begriff	Spielfigur (Meeple)
Beschreibung	Eine Spielfigur ist ein Held oder ein NPC.
Ist-ein	—
Kann-sein	Held, NPC
Aspekt	—
Bemerkung	Eine Spielfigur wird auf dem Spielbrett angezeigt und belegt dort ein Feld.
Beispiel	(Held) Dr.Strange

Begriff	Feld (Field)
Beschreibung	Ein Feld ist ein Teil des Spielbretts, auf dem sich maximal ein Objekt oder Held befinden kann.
Ist-ein	—
Kann-sein	Gras
Aspekt	Ist durch Koordinaten eindeutig bestimmt
Bemerkung	Feld ist die kleinste Einheit eines Spielbretts
Beispiel	Leeres Feld = Gras

Begriff	Attribute
Beschreibung	Ein Attribut ist eine Eigenschaft eines Helden.
Ist-ein	Attribut
Kann-sein	AP, MP, HP
Aspekt	—
Bemerkung	—
Beispiel	Action Points

Begriff	Action Points (AP)
Beschreibung	Jeder Held hat eine in der Heldenkonfiguration festgelegte Anzahl dieser Punkte. Im Laufe des Spieles kann in jeder Runde durch Aktionen der Helden diese Punkte aufgebraucht werden. Nach jeder Runde werden die Punkte wieder aufgefüllt auf die in der Konfiguration angegebenen Zahl.
Ist-ein	Attribut
Kann-sein	—
Aspekt	—
Bemerkung	Die Wortbedeutung wird damit äquivalent zum Lastenheft übernommen.
Beispiel	4 (drei)

Begriff	Health Points (HP)
Beschreibung	Jeder Held hat eine in der Heldenkonfiguration festgelegte Anzahl dieser Punkte. Sie repräsentieren den Gesundheitszustand des Helden. Im Laufe des Spieles kann sich dieser durch Aktionen der Helden verändern. Wenn die Health Points auf null sinken, ist der Held ausgeknockt.
Ist-ein	Attribut
Kann-sein	—
Aspekt	—
Bemerkung	Die Wortbedeutung wird damit äquivalent zum Lastenheft übernommen.
Beispiel	4 (drei)

Begriff	Movement Points (MP)
Beschreibung	Jeder Held hat eine in der Heldenkonfiguration festgelegte Anzahl dieser Punkte. Im Laufe des Spieles kann in jeder Runde durch Aktionen der Helden diese Punkte aufgebraucht werden. Nach jeder Runde werden die Punkte wieder aufgefüllt auf die in der Konfiguration angegebenen Zahl.
Ist-ein	Attribut
Kann-sein	—
Aspekt	—
Bemerkung	Die Wortbedeutung wird damit äquivalent zum Lastenheft übernommen.
Beispiel	4 (drei)

2.2 Anwendungskontext

Akteure

Akteur	Benutzer
Rollen	Spieler, Zuschauer
Aufgaben	Benutzt den Benutzer-Client
Handlungen	Benutzer verwenden den Benutzer-Client, um ein Spiel zu spielen oder um ein Spiel, das andere Spieler führen zu beobachten.
Kooperationen	Benutzer kooperieren mit anderen Benutzern indem Sie ihnen beim Spielen zuschauen oder selber mit ihnen Spielen.

Akteur	KI
Rollen	Spieler
Aufgaben	Spielt das Spiel autonom, d.h. ohne menschlíche Hilfe.
Handlungen	—
Kooperationen	Wird vom KI-Administrator gestartet.

Akteur	Entwickler
Rollen	Teammitglied
Aufgaben	Das System gemäß den Anforderungen entwickeln.
Handlungen	Entwickler entwickeln die Komponenten des Systems.
Kooperationen	Die Entwickler nehmen Verbesserungen anhand des Feedbacks des Tutors vor.

Akteur	Tutor
Rollen	Product Owner
Aufgaben	Kontrolliert die Arbeit der Entwickler
Handlungen	—
Kooperationen	Der Tutor gibt Feedback, anhand dessen die Entwickler Verbesserungen vornehmen.

Akteur	Server
Rollen	Schiedsrichter, Thanos, Goose, Stan
Aufgaben	• Spiel basierend auf Konfigurationsdateien erstellen • Clients die Verbindung ermöglichen, davon sollen 2 Clients das Spiel (gegeneinander) spielen können • NPCs steuern • Einhaltung der Regeln prüfen
Handlungen	Startet die Partie auf Grundlage von Spielkonfigurationen, verbindet sich mit den Clients und verwaltet die laufende Partie
Kooperationen	Nutzt die vom Editor erstellte Konfigurationen und verbindet sich mit Clients

Akteur	KI-Administrator
Rollen	Administrator
Aufgaben	Startet den KI-Client
Handlungen	Der KI-Client, startet den Client und stellt die Details für die Verbindung mit dem Server bereit. Danach handelt die KI autonom, d.h. ohne zutun des KI-Administrators.
Kooperationen	Der KI-Administrator verbindet den KI-Client nach dem Start mit einem Server. Dafür kooperiert er mit dem Serveradministrator um die Details für den Verbindungsaufbau zu erfahren.

Akteur	Serveradministrator
Rollen	Administrator
Aufgaben	Startet den Server
Handlungen	Der Serveradministrator startet den Server, dabei wählt er eine Spielkonfiguration, das Spielfeld, und die verwendbaren Helden aus. Des weiteren überwacht er den Server und gewährleistet dessen Funktionsfähigkeit.
Kooperationen	Der Serveradministrator kooperiert mit den Benutzern, insofern sich diese mit dem Server verbinden. Dafür muss er die notwendigen Details für den Verbindungsaufbau bereitstellen.

2.3 Funktionale Systemanforderungen

Die folgenden Anforderungen sind nach Komponente benannt und erhalten eine entsprechende Kennzeichnung. Die Unterteilungen sind Server (S-1 bis S-8), Game-Engine (GE-1 bis GE-7), Benutzer-Client (BC-1 bis BC-10) und KI-Client (KI-1 bis KI-10).

Server

AnforderungsNR	S-1
Titel	Verbindungen zu Clients
Beschreibung	Der Server soll Verbindungsanfragen von Clients akzeptieren und über den Zeitraum einer aktiven Session halten. Abschließend soll die Verbindung geschlossen werden.
Begründung	Mit dieser Verbindung können Nachrichten ausgetauscht werden.
Priorität	Muss
Abhängigkeiten	S-2

AnforderungsNR	S-2
Titel	Konfigurationsdatei
Beschreibung	Beim Start des Servers soll dieser eine Datei laden, die wichtige Konfigurationsdaten enthält.
Begründung	In dieser Datei können Daten, die der Serverbetreiber anpassen kann, wie der Port etc angegeben werden
Priorität	Muss
Abhängigkeiten	—

AnforderungsNR	S-3
Titel	Unterscheiden von Spielern und Zuschauer
Beschreibung	Der Server soll unterschieden ob ein Client als Spieler bei einer Partie teilnimmt oder diese als Zuschauer betrachtet.
Begründung	Der Server darf nur Spielzüge von den beiden Spieler Clients annehmen, muss aber an alle Clients in einer aktiven Session Partie Updates senden.
Priorität	Muss
Abhängigkeiten	S-1

AnforderungsNR	S-4
Titel	Verarbeiten von Spielzügen
Beschreibung	Der Server soll übers Netzwerk gesendete Spielzüge der beiden Spieler an die Game-Engine weitergeben.
Begründung	Die Game-Engine verarbeitet die Spielzüge.
Priorität	Muss
Abhängigkeiten	S-1, S-3

AnforderungsNR	S-5
Titel	Session Timeout
Beschreibung	Wenn ein verbundener Client sind nicht innerhalb eines vorgegebenen Timeouts meldet, wird die Verbindung zu diesem Client abgebrochen. Falls der Client einer der beiden Spieler ist, muss die Game-Engine über den Verbindungsabbruch informiert werden.
Begründung	Durch einen Timeout können geschlossene Verbindungen, bei denen der Client den Server nicht informiert hat gefunden und geschlossen werden.
Priorität	Muss
Abhängigkeiten	S-1, S-2

AnforderungsNR	S-6
Titel	Behandlung von Verbindungsabbrüchen
Beschreibung	Wenn einer der beiden Spieler die Verbindung zum Server verliert, kann er sich innerhalb einer gewissen Zeitspanne wieder mit dem Server verbinden und die Partie wieder aufnehmen.
Begründung	Ohne die Möglichkeit der Verbindungswiederaufnahme würde bei jedem Verbindungsproblem die Partie beendet werden.
Priorität	Muss
Abhängigkeiten	S-1, S-5, S-2

AnforderungsNR	S-7
Titel	Behandlung von Protokollverletzungen
Beschreibung	Wenn ein Client eine Nachricht sendet, die von dem Kommunikationsprotokoll abweicht, sendet der Server dem Client eine entsprechende Fehlermeldung und beendet die Verbindung. Dies geschieht ebenfalls, wenn die Game-Engine den server über einen Regelbruch benachrichtigt.
Begründung	Nachrichten die sich nicht an das offizielle Format halten können nicht verstanden werden.
Priorität	Muss
Abhängigkeiten	S-1, S-2

AnforderungsNR	S-8
Titel	Log-Datei
Beschreibung	Der Server soll alle relevanten Informationen in eine Log Datei schreiben.
Begründung	Anhand dieser Datei können Fehler und unerwartete Ereignisse nachvollzogen werden.
Priorität	Muss
Abhängigkeiten	S-2

Game-Engine

AnforderungsNR	GE-1
Titel	Laden einer Partie Konfiguration
Beschreibung	Beim Start lädt die Game-Engine eine Partie Konfiguration.
Begründung	Verbindliche Anforderung (Lastenheft)
Priorität	Muss
Abhängigkeiten	—

AnforderungsNR	GE-2
Titel	Laden der Szenario Konfiguration
Beschreibung	Beim Start lädt die Game-Engine eine Szenario Konfiguration.
Begründung	Verbindliche Anforderung (Lastenheft)
Priorität	Muss
Abhängigkeiten	—

AnforderungsNR	GE-3
Titel	Laden einer Charakter-Konfiguration
Beschreibung	Beim Start lädt die Game-Engine eine Charakter-Konfiguration.
Begründung	Verbindliche Anforderung (Lastenheft)
Priorität	Muss
Abhängigkeiten	—

AnforderungsNR	GE-4
Titel	Überprüfen von Spielzügen
Beschreibung	Die Game-Engine muss überprüfen, ob ein Spielzug generell korrekt ist, bevor dieser weiter verarbeitet wird. Damit ist gemeint, dass er unabhängig von der momentanen Spielsituation das korrekte Schema hat.
Begründung	Wenn ein Spielzug sich nicht an das korrekte Protokoll hält kann er nicht verstanden werden.
Priorität	Muss
Abhängigkeiten	—

AnforderungsNR	GE-5
Titel	Verarbeiten von Spielzügen
Beschreibung	Die Game-Engine muss Spielzüge, gemäß dem im Lastenheft vorgegebenen Regelwerk verarbeiten. Anschließend muss der veränderte Partie Zustand zurückgegeben werden.
Begründung	Das Verändern des momentanen Partie Zustandes durch Spielzüge stellt den Inbegriff des Spiels dar.
Priorität	Muss
Abhängigkeiten	—

AnforderungsNR	GE-6
Titel	Überprüfen von Siegbedingungen
Beschreibung	Die Game-Engine prüft nach jeder Zugphase, ob eine Siegbedingung eingetreten ist. Anschließend wird der Sieger bekannt gegeben.
Begründung	Ohne das Überprüfen der Siegbedingungen endet das Spiel nicht.
Priorität	Muss
Abhängigkeiten	—

Benutzer-Client

AnforderungsNR	BC-1
Titel	Benutzer wird Spieler
Beschreibung	Der Benutzer-Client kann sich über das Netzwerk bei einem Server für eine dort angebotene Partie als Spieler registrieren.
Begründung	Verbindliche Anforderung (Lastenheft)
Priorität	Muss
Abhängigkeiten	

AnforderungsNR	BC-2
Titel	Benutzer wird Zuschauer
Beschreibung	Der Benutzer-Client hat einen Zuschauermodus, bei dem er sich als passiver Zuschauer bei einem Server für eine Partie registrieren, oder einer bereits laufenden beitreten kann.
Begründung	Verbindliche Anforderung (Lastenheft)
Priorität	Muss
Abhängigkeiten	BC-3

AnforderungsNR	BC-3
Titel	Als menschlicher Spieler ausweisen
Beschreibung	Der Benutzer-Client teilt dem Server mit, dass er ein von einem Menschen gesteuerter Client ist.
Begründung	Verbindliche Anforderung (Lastenheft)
Priorität	Muss
Abhängigkeiten	

AnforderungsNR	BC-4
Titel	Namen mitteilen
Beschreibung	Benutzer-Clients teilen dem Server beim Verbinden einen Namen mit.
Begründung	Verbindliche Anforderung (Lastenheft). So kann man Clients (sowohl Spieler als auch Zuschauer) auf der Benutzeroberfläche klar identifizieren.
Priorität	Muss
Abhängigkeiten	

AnforderungsNR	BC-5
Titel	Graphische Oberfläche
Beschreibung	Der Benutzer-Client visualisiert das Spielgeschehen mittels einer graphischen Oberfläche.
Begründung	Verbindliche Anforderung (Lastenheft)
Priorität	Muss
Abhängigkeiten	

AnforderungsNR	BC-6
Titel	Unterscheidbarkeit der Spielfiguren und Spieler
Beschreibung	Die Spielfiguren werden auf der Karte unterscheidbar dargestellt (etwa durch individuelle Avatare, oder eingeblendete Namen). Die Zugehörigkeit zu einem Spieler wird mit einer eigener Farbe je nach Spieler dargestellt.
Begründung	Verbindliche Anforderung (Lastenheft). So kann man als Zuschauer erkennen, wer die Charaktere sind und zu welchem Spieler sie gehören.
Priorität	Muss
Abhängigkeiten	BC-5

AnforderungsNR	BC-7
Titel	Werte & Zustand der Spielfiguren anzeigen
Beschreibung	Auf der grafischen Oberfläche werden für die Spielfiguren Health Bars, Icons für Steine im Inventar, sowie ein Panel mit Werten für Movement Points, Action Points und den Schadenswerten für Angriffe angezeigt.
Begründung	Verbindliche Anforderung (Lastenheft)
Priorität	Muss
Abhängigkeiten	BC-5

AnforderungsNR	BC-8
Titel	Visualisierung der Angriffe
Beschreibung	Nahkampf- und Fernkampf-Angriffe werden so visualisiert, dass man erkennen kann, wer wen angegriffen hat. Evtl. sollte auch noch der gemachte Schaden erkennbar gemacht werden, etwa durch eine Änderung der Health Bar oder durch Anzeige einer Zahl über dem Angegriffenen.
Begründung	Verbindliche Anforderung (Lastenheft)
Priorität	Muss
Abhängigkeiten	BC-5, BC-7

AnforderungsNR	BC-9
Titel	Aktionen innerhalb einer Phase vornehmen
Beschreibung	Der Spieler kann über die graphische Oberfläche während einer Partie die möglichen Aktionen vornehmen.
Begründung	Verbindliche Anforderung (Lastenheft)
Priorität	Muss
Abhängigkeiten	BC-5

AnforderungsNR	BC-10
Titel	Animation der Spielphasen
Beschreibung	Die Abwicklung der einzelnen Rundenphasen und Zugphasen werden vom Benutzer-Client animiert dargestellt. Dabei wird darauf geachtet, dass die Dauer der Animationen an die für die jeweilige Phase in der Partie-Konfiguration gewährte Zeit angepasst ist.
Begründung	Verbindliche Anforderung (Lastenheft)
Priorität	Muss
Abhängigkeiten	BC-5

AnforderungsNR	BC-11
Titel	Pop-Up Fenster
Beschreibung	Es soll Pop-Ups geben, die immer dann angezeigt werden, wenn der Spieler über eine ungewöhnliche Situation informiert werden muss. Darunter fallen Fehlermeldungen als auch Statusinformationen, des Spielgeschehens wie beispielweise das Erreichen der maximalen Rundenanzahl
Begründung	Dient einer besseren Bedienbarkeit und erleichtert dem Benutzer das Verständnis, was die Anwendung aktuell tut.
Priorität	Soll
Abhängigkeiten	

AnforderungsNR	BC-12
Titel	Credits
Beschreibung	Der Benutzer-Client soll Credits anzeigen können, sodass ersichtlich ist, wer Urheber der verwendeten Inhalte ist.
Begründung	Je nachdem was für Resourcen verwendet werden, gerade im Bereich der visuellen Gestaltung, müssen eventuell die Urheber genannt werden um konform mit den Lizenzbedingungen zu sein.
Priorität	Soll
Abhängigkeiten	BC-16

AnforderungsNR	BC-13
Titel	Rundenzähler
Beschreibung	Auf dem Spielbildschrim soll die Nummer der aktuellen Runde angezeigt werden.
Begründung	Spieler können daran messen, wie weit sie aktuell im Spiel vorangeschritten sind und können, sofern ihnen die maximale Rundenanzahl R_{Max} bekannt ist, sehen wie lange es noch dauert bis Thanos erscheint.
Priorität	Soll
Abhängigkeiten	

AnforderungsNR	BC-14
Titel	Aktivitätsanimation
Beschreibung	Auf dem Spielbildschirm soll eine Animation angezeigt werden, die durchläuft.
Begründung	Wenn ein Spieler darauf warten muss, dass der Andere seinen Zug durchführt und bestätigt, so soll ihm die Animation symbolisieren, dass die Anwendung noch aktiv und nicht abgestürzt ist oder ein anderer Fehler vorliegt.
Priorität	Soll
Abhängigkeiten	

AnforderungsNR	BC-15
Titel	Heroselection
Beschreibung	Bei Beginn eines Spieles muss der Benutzer aus 6 Helden sein Team zusammenstellen. Der Benutzerclient soll zur Visualisierung 12 (Vorschau-)Bilder anzeigen, die der Benutzer über anklicken auswählen kann.
Begründung	Einfache und intuitive Form der Visualisierung. Ist koherent mit den im Mock-Up erarbeiteten Ansicht.
Priorität	Kann
Abhängigkeiten	

AnforderungsNR	BC-16
Titel	Hauptmenü
Beschreibung	Auf dem Hauptmenü sollen nur Buttons zur Interaktion zur Verfügung stehen, die dann an entsprechende Screens weiterleiten. Vorhanden sein sollten auf jeden Fall Spiel starten, Spiel zuschauen, Credits und Beenden.
Begründung	Hauptmenü soll schlicht gehalten werden. Es dient hauptsächlich der Navigation und sollte daher übersichtlich sein um neue Benutzer nicht abzuschrecken.
Priorität	Kann
Abhängigkeiten	BC-12

AnforderungsNR	BC-17
Titel	Reihenfolge der Helden
Beschreibung	Auf dem Spielbildschirm wird die Reihenfolge der Helden die in der aktuellen Runde spielen werden angezeigt (Sofern der Server diese bekannt gibt).
Begründung	Zusätzliche taktische Komponente des Spiels. Hilft einem seine Runden zu strukturieren und bessere Zugfolgen zu erzeugen.
Priorität	Kann
Abhängigkeiten	

KI-Client

AnforderungsNR	KI-1
Titel	Autonomie der KI
Beschreibung	Die KI handelt nach der Verbindung mit dem Server und dem Beitritt zu einer Partie völlig autonom. Das heißt bis zum Spielende nimmt der KI-Client keine Befehle mehr von Benutzern entgegen. Der KI-Client wird vom KI-Administrator gestartet
Begründung	Eine Manipulation der KI durch Benutzer soll während einer laufenden Partie nicht möglich sein. Außerdem ist eine vollständig autonome KI viel beeindruckender und nützlicher, als eine nur “semi-selbstständige” KI.
Priorität	Muss
Abhängigkeiten	—

AnforderungsNR	KI-2
Titel	Kommunikation mit dem Server
Beschreibung	Der KI-Client kommuniziert mit dem Server wie vom Standardisierungskomitee vorgegeben.
Begründung	Reibungslose Kommunikation mit dem Server, insbesondere auch für Server, die von anderen Teams entwickelt wurden.
Priorität	Muss
Abhängigkeiten	—

AnforderungsNR	KI-3
Titel	Umgang mit Pausen
Beschreibung	Der KI-Client beantragt keine Spielpause vom Server und akzeptiert jede beliebig lange Pause von Spielern.
Begründung	Die KI soll menschlichen Spielern ein angenehmes Spielerlebnis bieten.
Priorität	Muss
Abhängigkeiten	—

AnforderungsNR	KI-4
Titel	Regelkonformität
Beschreibung	Die KI führt nur regelkonforme Spielzüge aus. Sie hat damit genau die selben Handlungsoptionen, wie menschliche Spieler.
Begründung	Die KI hat keine regeltechnischen Vorteile gegenüber menschlichen Spielern.
Priorität	Muss
Abhängigkeiten	—

AnforderungsNR	KI-5
Titel	Planung der KI-Handlung
Beschreibung	In der ersten Stufe hat die KI zunächst nur eine Strategie für die Spielzüge einzelner Helden. In zweiter Stufe hat die KI eine Strategie für eine ganze Runde (also alle Züge der Helden innerhalb einer Runde). Und in dritter Stufe berücksichtigt die KI eine globale Strategie über die gesamte Partie hinweg.
Begründung	Durch diese Entwicklungsstrategie wird die KI sukzessive optimiert, lässt sich aber bereits in früheren Stadien testen und verwenden.
Priorität	Muss
Abhängigkeiten	—

AnforderungsNR	KI-6
Titel	Strategierepräsentation
Beschreibung	Die KI repräsentiert ihre Strategie in einer speziellen Datenstruktur anhand von Gewichten.
Begründung	Die Strategie der KI soll ohne eine Änderung der Implementierung angepasst werden können.
Priorität	Muss
Abhängigkeiten	KI-5

AnforderungsNR	KI-7
Titel	Autonome Strategieauswahl
Beschreibung	Die KI soll die Möglichkeit besitzen, die globale Strategie autonom aus verschiedenen Möglichkeiten auszuwählen.
Begründung	Die KI soll die Möglichkeit besitzen, die globale Strategie autonom aus verschiedenen Möglichkeiten auszuwählen.
Priorität	Muss
Abhängigkeiten	KI-6, KI-8

AnforderungsNR	KI-8
Titel	Heldengruppen Auswahl
Beschreibung	Die KI wählt eigenständig eine Heldengruppe aus sechs Helden aus.
Begründung	Die KI wählt genau wie ein menschlicher Spieler eine Heldengruppe aus.
Priorität	Muss
Abhängigkeiten	—

AnforderungsNR	KI-9
Titel	Heldengruppen Auswahl Strategie
Beschreibung	Die KI besitzt eine Strategie bei der Heldenauswahl. Diese kann von der globalen Strategie abhängen.
Begründung	Die KI wählt eine Heldengruppe nach strategischen Gesichtspunkten und nicht willkürlich aus.
Priorität	Muss
Abhängigkeiten	KI-8

AnforderungsNR	KI-10
Titel	Namen mitteilen
Beschreibung	KI-Clients teilen dem Server beim Verbinden einen Namen mit.
Begründung	Verbindliche Anforderung (Lastenheft). So kann man Clients (sowohl Spieler als auch Zuschauer) auf der Benutzeroberfläche klar identifizieren.
Priorität	Muss
Abhängigkeiten	—

3 Softwarespezifikationen

3.1 Funktionen

Spielablauf Übersicht

Das Sequenzdiagramm (siehe nächste Seite) stellt den gesamten Spielablauf von Marvelous Mashup dar. Zu Beginn verbinden sich zwei Spieler und gegebenenfalls noch Zuschauer. Die verbundenen Spieler führen ihre Heldenauswahl durch, bei der sie jeweils sechs von 12 vom Server zur Verfügung gestellten Helden auswählen dürfen. Danach beginnt das eigentliche Spiel. In jeder Runde führen die Spieler Züge mit ihren Helden aus, wobei die Reihenfolge der Helden vom Server bestimmt wird. Zudem kommen in bestimmten Runden zusätzliche NPCs dazu, die sich an dem Spielgeschehen beteiligen. So erscheint in den ersten sechs Runden Goose, die pro Runde je einen Stein zufällig auf dem Spielfeld platziert. In Runde sieben erscheint zudem Stan Lee, der Spieler wiederbeleben kann und ist die maximale Rundenzahl R_{Max} (spezifiziert in der Partie-Konfiguration) erreicht, so führt nach jeder Runde Thanos noch einen Zug aus. Über das Netzwerk werden dabei von den Benutzer-Clients der Spieler Steuerungsbefehle übertragen, die ihre Spielzüge übertragen, zudem erhalten die Spieler als auch alle Zuschauer entsprechende Aktualisierungen über Spielfeld und aktuelle Situation.

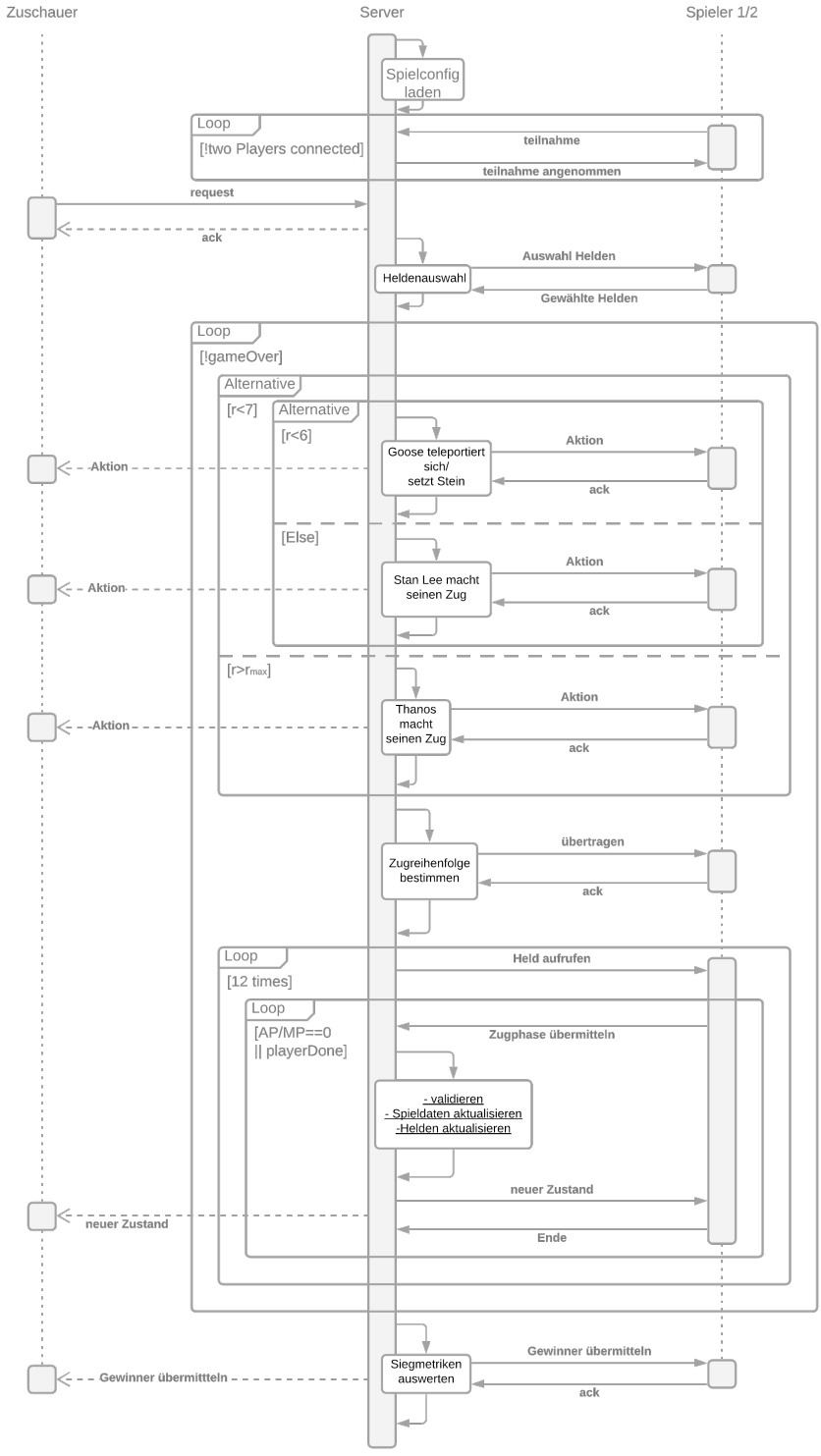

Abbildung 3.1: Spielablauf

Spielablauf Benutzer-Client

Im Folgenden ist der Spielablauf in einem Zustandsdiagramm dargestellt, wie er von Benutzer-Clients durchlaufen wird. Die Game Engine realisiert die hierfür notwendige Logik.

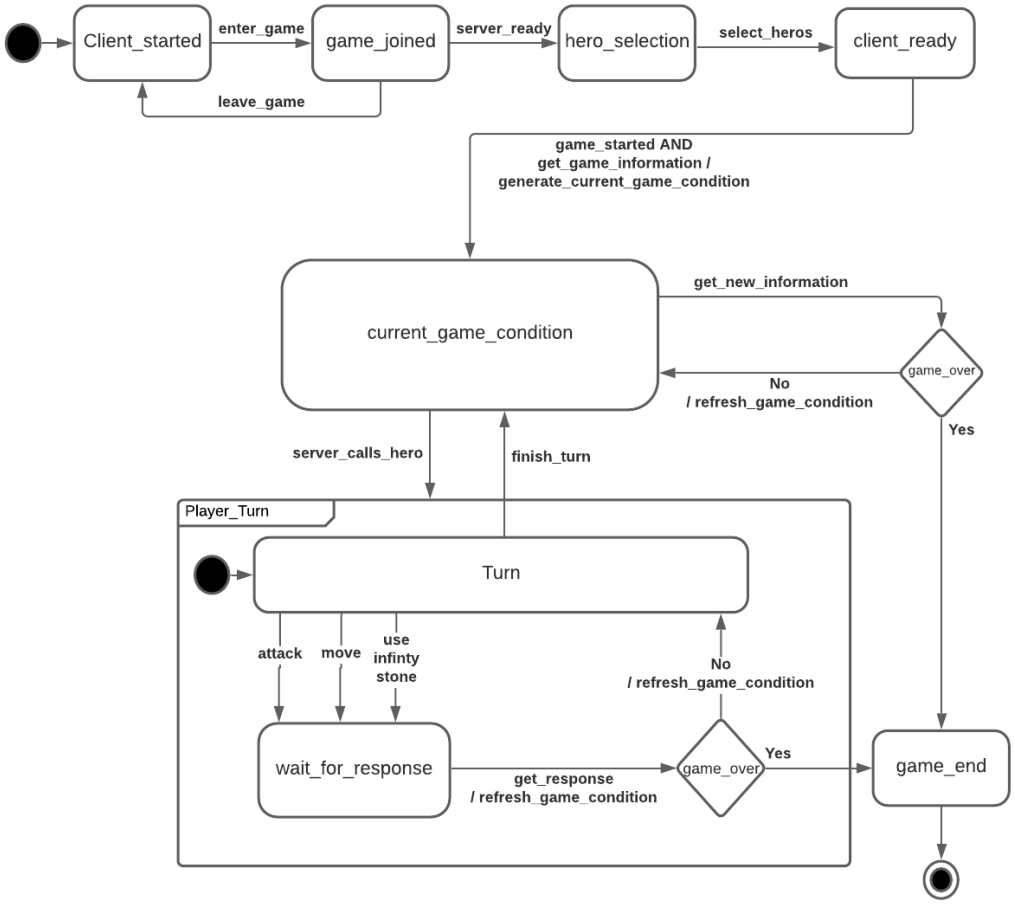

Abbildung 3.2: Benutzer-Client Spielablauf Zustandsdiagram

Der Benutzer-Client durchläuft zunächst die Serverconnection und das Auswahlfenster für die Helden. Danach startet das eigentliche Spiel. Ist der Spieler an der Reihe, weil vom Server der Aufruf eines zugehörigen Heldes erfolgt, so kann dieser nun einen Spielzug deklarieren. Andernfalls muss der Spieler warten, bis der Gegner seinen Zug vollendet hat. Sollte es dazu kommen, dass ein Zug des Gegners oder des Spieler das Spiel beendet, so verlässt der Benutzer-Client diese Schleife.

Spielablauf Server

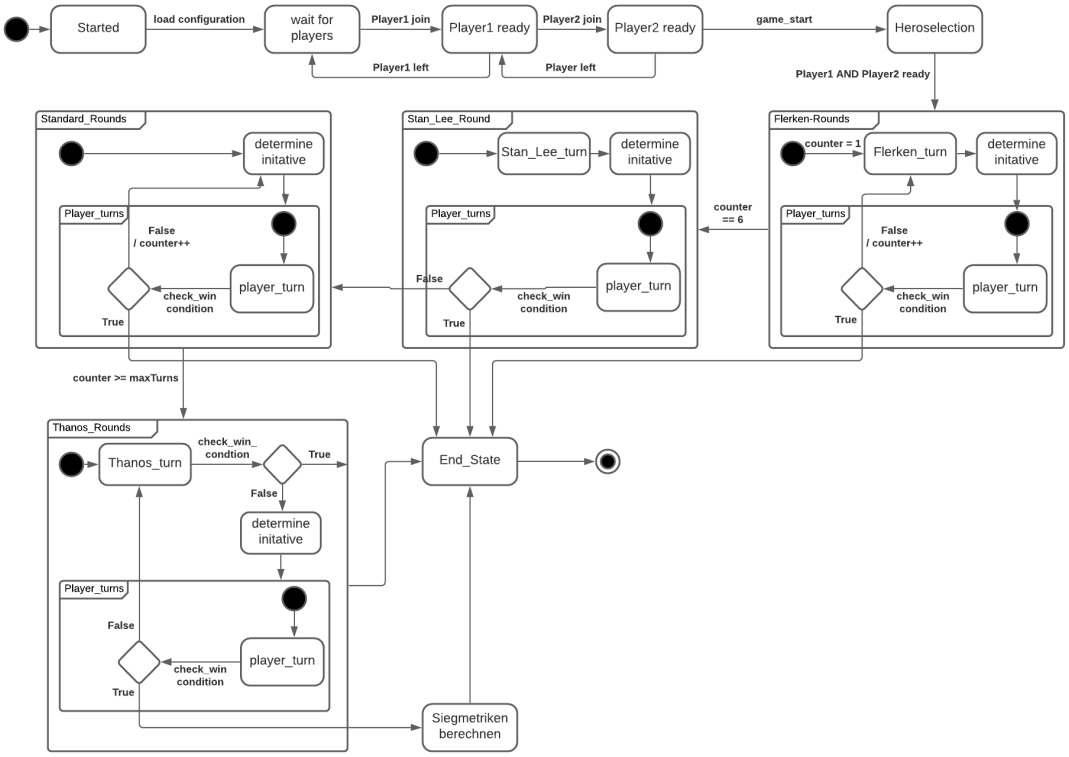

Abbildung 3.3: Server Zustandsdiagramm

Der Server wird mit einer entsprechenden Spielkonfiguration gestartet. Danach wartet er auf die Spieler 1 und 2. Haben sich zwei Spieler mit dem Server verbunden, wählt der Server zwölf Helden für jeden Spieler aus und wartet darauf, dass die Spieler ihre Helden auswählen. Ist dies erfolgt, wird das Spiel gestartet. Dabei gibt es verscheidene Spielphasen. Zunächst die Züg in denen der Flerken handelt. Dann taucht Stan Lee auf und zuletzt erfolgen die Thanos Züge. Diese Sonderfiguren müssen vom Server gesteuert werden.

3.2 Server

Schnittstelle

Der Server verwendet keine GUI sondern ein Komandozeileninterface, da er nur vom Serveradministrator gestartet wird. Der Administrator benötigt keine GUI, da davon ausgegangen werden kann, dass er mit der Anwendung und der Komandozeile vertraut ist. Des Weiteren greift der Administrator unter Umständen mittels `ssh` auf den Server zu, darum wird ein Komandozeileninterface sowieso benötigt.

Die Netzwerkschnittstelle des Servers ist im Standardisierungsdokument nachzulesen. Da zum jetztigen Stand das Standardisierungsdokument noch nicht veröffentlicht wurde, wurden die Spezifikationen nicht in das Pflichtenheft übernommen.

Nutzungskonzept

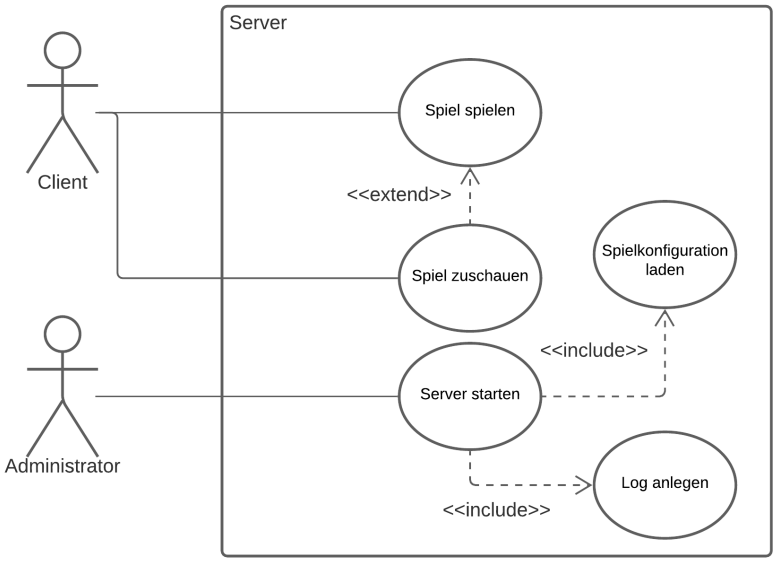

Abbildung 3.4: Anwendungsfalldiagramm Server

Erklärungen:

- «include» zwischen “Server starten” und “Spielkonfiguration laden”: Immer wenn der Server gestartet wird, muss er die Spielkonfiguration laden.
- «include» zwischen “Server starten“ und “Log anlegen“: Es wird ein Log-File angelegt, wenn der Server gestartet wird. In diesem Log-File legt der Server dann Einträge zu allen Ereignissen ab.
- «extend» zwischen “Spiel spielen” und “Spiel zuschauen”: Es ist nur möglich, einem Spiel zuzuschauen, wenn auch jemand spielt, es muss aber niemand zuschauen.

Anforderungen:

- Spiel spielen: S-1; S-3; S-4; S-5; S-6; S-7; GE-4; GE-5; GE-6; NF-3
- Spiel zuschauen: S-1; S-3; S-5; S-7; NF-3
- Server starten: NF-1
- Spielkonfiguration laden: S-2; GE-1; GE-2; GE-3; NF-4
- Log anlegen: S-8

3.3 Benutzer-Client

Schnittstelle

Main-Menu-Screen

Das Mainmenu ist die erste, dem Benutzer angezeigte Oberfläche. Sie stellt drei Buttons bereit, um die vom Benutzer-Client angebotenen Funktionen, ein Spiel spielen, einem Spiel als Zuschauer beiwohnen und den Credit Screen anzeigen, zu nutzen. Die Auswahl erfolgt über einen linken Mausklick auf die jeweiligen Buttons. Der Credit Screen zeigt eine statische Auflistung aller würdigungswürdigen Personen und Institutionen. Der Play-Game-Button und der Watch-Game-Button wechseln jeweils in den Connect-to-Server-Screen.

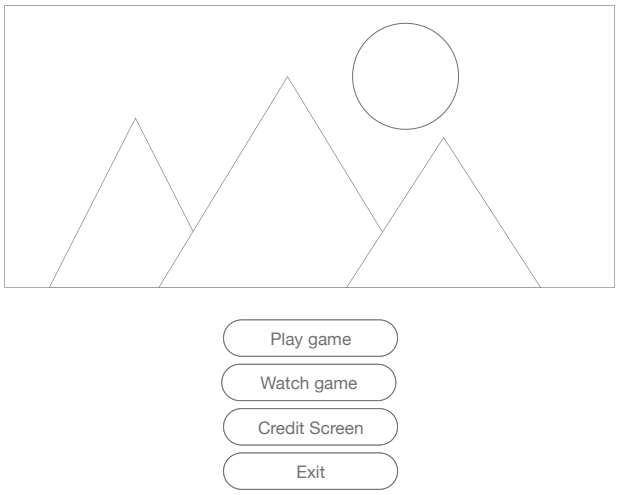

Abbildung 3.5: Main Menu Mock-Up

Connect-to-Server-Screen

In diesem Screen können Benutzer eine Verbindung zu einem Server ihrer Wahl aufbauen. Hierfür muss ein gültiger Spielername und die Adresse des Servers, in die entsprechenden Textfelder mittels Tastatureingabe eingegeben werden. Die Verbindung wird dann durch einen Mausklick auf den Connect-Button bestätigt. Sollte die angegebene Adresse ungültig sein, öffnet sich ein Pop-Up-Fenster. Dieses weist den Benutzer mittels eines entsprechenden Hinweistextes auf das Problem hin. Das Pop-Up-Fenster wird durch einen linken Mausklick auf den O.K. Button geschlossen und der Benutzer wird zum Connect-to-Server-Screen zurückgebracht. Der Back Button des Screens wechselt zurück zum Main-Menu-Screen. Kommt es zu einer gültigen Serververbindung, wechselt der Benutzer-Client, abhängig davon, ob der Benutzer als Zuschauer oder Spieler mit dem Server Verbindung aufnimmt - entweder im ersten Fall zum Game-Screen oder im zweiten Fall zum Hero-Select-Screen.

Abbildung 3.6: Connect to Server Mock-up

Hero-Select-Screen

Der Hero-Select-Screen zeigt die Namen und die Portrait-Bilder der 12 vom Server für den Spieler bereitgestellten Helden. Darunter findet man die Namen und die entsprechenden Statuswerte des Helden. Bereits gewählte Helden werden durch einen leuchtenden Rahmen hervorgehoben. Durch erneuten Mausklick auf einen ausgewählten Helden, wird dieser wieder aus der Auswahl entfernt. Wenn der Spieler 6 Helden ausgewählt hat, wird der bis dato ausgegraute Continue-Button farbig hervorgehoben und kann betätigt werden. Dies geschieht mittels Linksklick. Dadurch wird dem Server signalisiert, dass der Spieler bereit ist. Im Fall, dass der andere Spieler die Heldenauswahl noch nicht getroffen hat, wird ein Pop-Up-Fenster eingeblendet. Dieses zeigt durch eine Zahnrad-Animation an, dass auf den anderen Spieler und somit auf den Spielstart gewartet wird. Der Spieler hat dabei die Möglichkeit mittels eines Back-Buttons in die Heldenauswahl zurückzukehren. Betätigt ein Spieler den back-Button im Hero-Select-Screen wird die Verbindung zum Server getrennt.

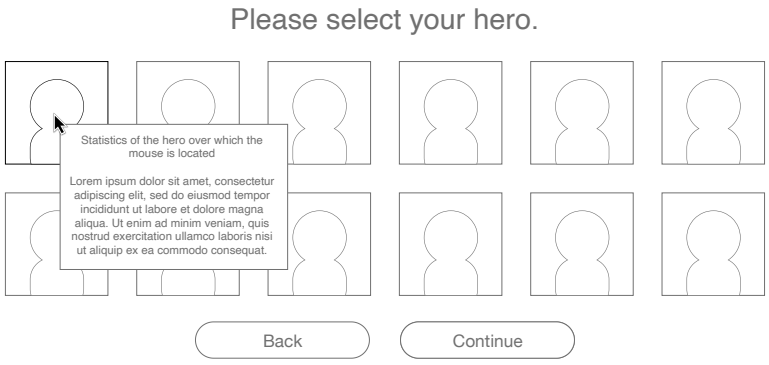

Abbildung 3.7: Hero Select Mock-Up

Game-Screen

Der Game-Screen zeigt im gesamten Fenster eine Repräsentation des Spielfelds. An den Fensterrändern finden sich die Kontroll-Fenster und Menüs. Ist ein Benutzer nur als Zuschauer in den Game-Screen gewechselt, sind für ihn alle Aktionen die Einfluss auf das Spielgeschehen ausüben ausgegraut, ansonsten hat er den selben Nutzenumfang wie ein Spieler. Alle Menu- und Auswahl-Optionen werden durch einen Linksklick der Maus aktiviert.
Im Oberen Linken Eck findet sich der Menu Button. Wird dieser betätigt wird bei Spielern das Spiel pausiert, bei Zuschauern passiert das selbstredend nicht. Es öffnet sich jedoch für bei eine Menu-Bar, die erlaubt das Menu zu schließen oder das Spiel zu verlassen. Im „Whose turn“ Fenster wird immer der Name des Spielers angezeigt der Momentan an der Reihe ist. Handelt der Server, das heißt Thanos, Stan Lee usw., so stehen deren Namen im Fenster. Das „Round-Status-Fenster“ zeigt die aktuelle Rundenzahl. Im Fenster Additional Information werden weitere Informationen angezeigt. Hier findet sich zum Beispiel die Anzahl an Runden bis Thanos erscheint. Des Weiteren kann hier durch Klick auf einen Button eine Liste der Namen der Zuschauer eingeblendet werden. Am linken Seitenrand werden die Namen und Portraits der Helden in Reihenfolge, von oben nach unten, entspre-

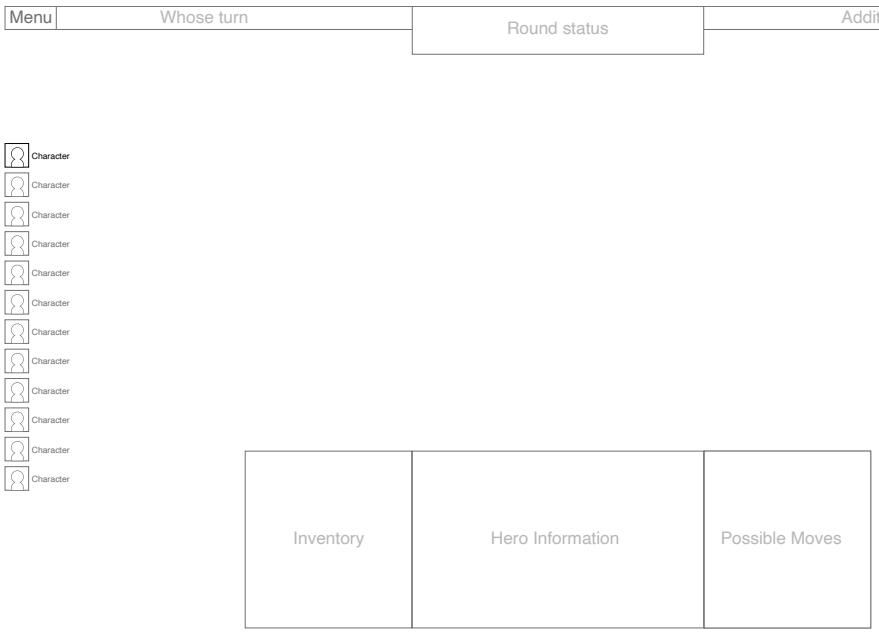

Abbildung 3.8: Game Screen Mock-Up

chend ihrer Zugreihenfolge dargestellt. Durch farbige Rahmen wird die Zugehörigkeit zum entsprechenden Spieler gezeigt. Der Held, der momentan an der Reihe ist wird durch einen leuchtenden Rahmen hervorgehoben. Am unteren Fensterrand findet sich die Helden-Kontroll-Leiste. Diese zeigt den Helden an, der momentan im Fokus des Spielers oder Benutzers steht. Ein Held wird in den Fokus genommen, in dem entweder die Avatar Figur auf dem Spielfeld oder das Portrait-Bild des Helden am linken Bildrand ausgewählt wird. Alle Aktionen, die die Leiste bietet, sind für Zuschauer ausgegraut und können von diesen nicht verwendet werden. Ebenso für den Spieler der nicht an der Reihe ist, oder wenn dieser einen Held im Fokus hat, den er nicht kontrolliert oder der nicht an der Reihe ist. Die Leiste ist dreigeteilt. Ganz links findet sich das Inventar. Dieses zeigt die Infinity Stones, die der entsprechende Held besitzt. Durch Klick auf einen Stein, kann dessen Fähigkeit eingesetzt werden. Benutzte Steine werden während ihrer Abklingzeit ausgeraut und eine Zahl zeigt die verbleibende Abklingzeitdauer in Runden an. Im Hero-Information-Fenster befinden sich die aktuellen Statuswerte des Helden und ein Portrait-Bild. Das rechte Fenster wiederum bietet eine Auswahl an möglichen Aktionen, die der Held durchführen kann. Diese sind Bewegen, Nah- und Fernkampfangriffe und einen Infinity Stein übergeben. Des Weiteren gibt es einen „keine-Aktion-Button“. Durch Klick auf

einen solchen Button wird die entsprechende Aktion „aktiv“ das heißt der nächste Klick auf zum Beispiel ein Feld auf dem Spielbrett bewirkt bei ausgewählter Bewegung die entsprechende Bewegungsaktion. Es kann dabei immer nur eine Aktion gleichzeitig ausgewählt werden. Soll keine besondere Aktion ausgeführt werden, um beispielsweise einen anderen Helden in den Fokus zu nehmen, kann der keine-Aktion-Button ausgewählt werden.

Status Pop-Up

Der Status Pop-Up zeigt dem Benutzer eine Status- oder Fehlermeldung. Das ist beispielsweise der Fall, wenn der Benutzer sich mit einem Server verbinden möchte und der Verbindungsaufbau fehlschlägt.

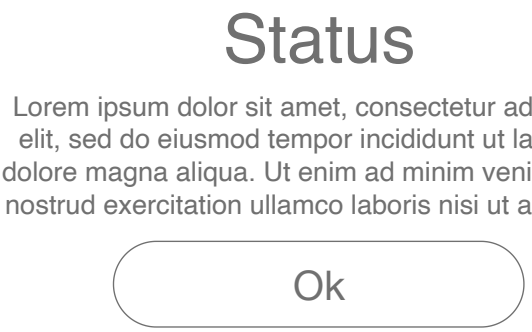

Abbildung 3.9: Status Pop-Up Mock-Up

Waiting for other player Pop-Up

Der Waiting for other player Pop-Up wird dem Benutzer angezeigt, wenn er auf eine Aktion des anderen Spielers warten muss. Das ist beispielsweise der Fall, wenn der Benutzer sich mit einem Server verbunden hat, allerdings noch kein anderer Spieler verbunden ist.

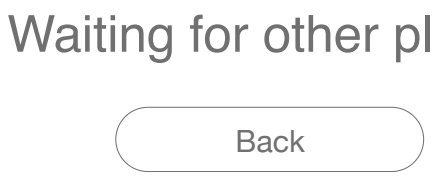

Abbildung 3.10: Waiting for other player Pop-Up Mock-Up

Nutzungskonzept

Dialogstrukturdiagramm

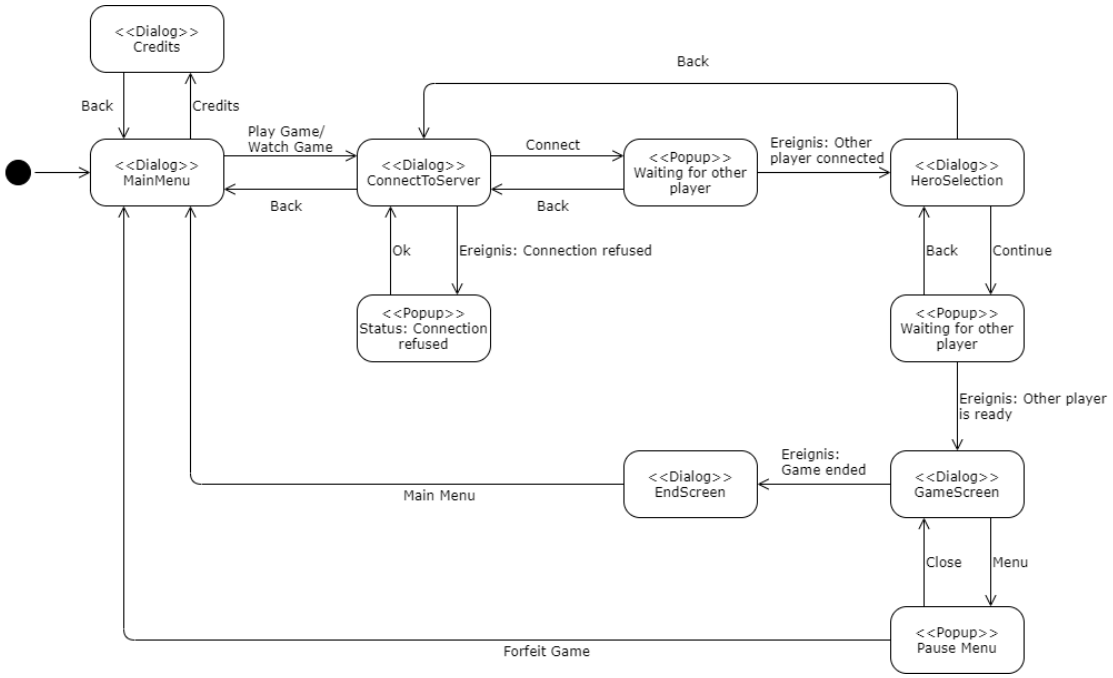

Abbildung 3.11: Dialogstrukturdiagramm Benutzer-Client

Das Dialogstrukturdiagramm für den Benutzer-Client zeigt, wie der Benutzer durch die verschiedenen Dialoge des Benutzer-Clients navigieren kann und an welchen Stellen Pop-Ups auftauchen.

Anwendungsfalldiagramm

Erklärungen:

- «extend» zwischen “Client beenden” und “Client starten”: Es ist nur möglich, den Client zu beenden, wenn er zuvor gestartet wurde, er muss aber (theoretisch) nicht beendet werden
- «include» zwischen “Spiel spielen” und “Spiel zuschauen”: Wenn man das Spiel spielt, schaut man zwingend auch zu. Zuschauen bedeutet in diesem

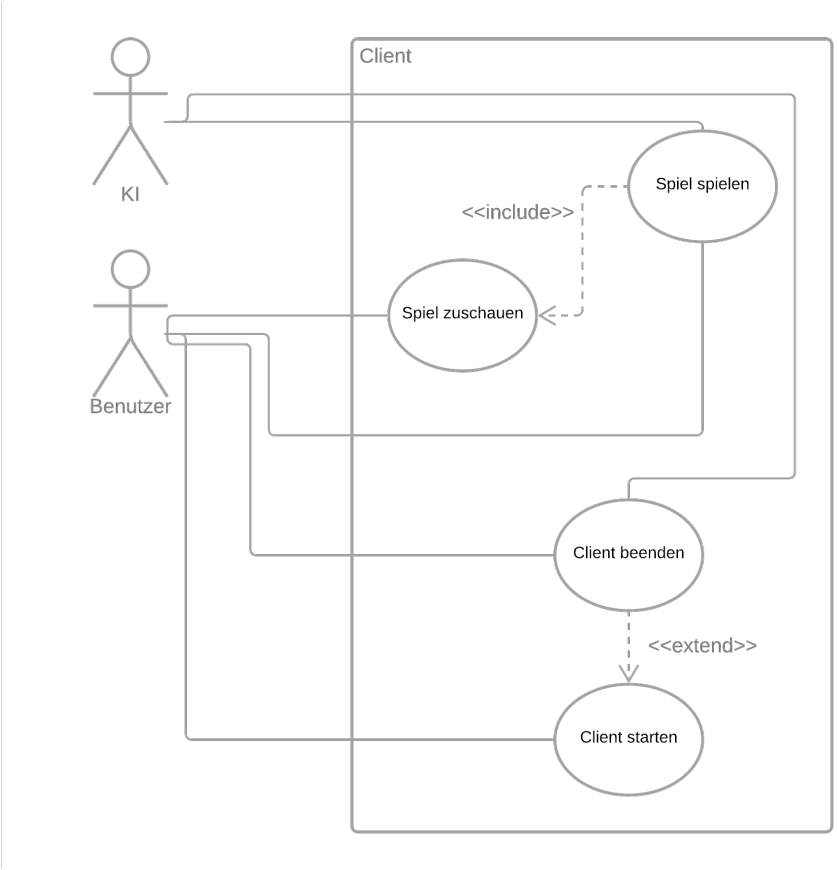

Abbildung 3.12: Anwendungsfalldiagramm Benutzer-Client

Kontext nicht zwingend, dass das Spiel grafisch dargestellt wird, sondern nur, dass man Informationen über den aktuellen Spielzustand erhält.

- Anmerkung: Die KI kann den Client nicht starten, viel mehr wird sie mit dem KI-Client zusammen vom Benutzer gestartet.

Anforderungen:

- Spiel spielen: BC-1; BC-3; BC-4; BC-9; KI-1; KI-2; KI-3; KI-4; KI-5; KI-6; KI-7; KI-8; KI-9; NF-3
- Spiel zuschauen: BC-2; BC-4; BC-5; BC-6; BC-7; BC-8; BC-10; NF-3
- Client beenden: —
- Client starten: KI-10; NF-1

3.4 Editor

Schnittstelle

Der Editor soll eine Graphische Benutzeroberfläche (GUI) erhalten. Dies soll sowohl Übersichtlichkeit als auch Benutzerfreundlichkeit dienen. Da in dieser Anwendung eine sehr hohe Anzahl an Einstellung und Optionen verwaltet wird, wäre eine Kommandozeile ungeeignet, da die Anzahl und Komplexität der Befehle unangebracht hoch wäre. Wir haben uns für eine simple Darstellung mit Tabelle zur Anpassung der Werte und einem konsistenten Button-Layout, wie es im Folgenden näher beschrieben wird, entschieden. Allerdings ist an dieser anzumerken, dass der Editor keine von Team 15 erstellte Komponente sein wird, sondern diese von einem anderen Entwicklerteam gekauft werden wird. Daher wird das finale Produkt vermutlich Abweichungen zu unserem aktuellen Entwurf aufweisen.

Hauptmenü

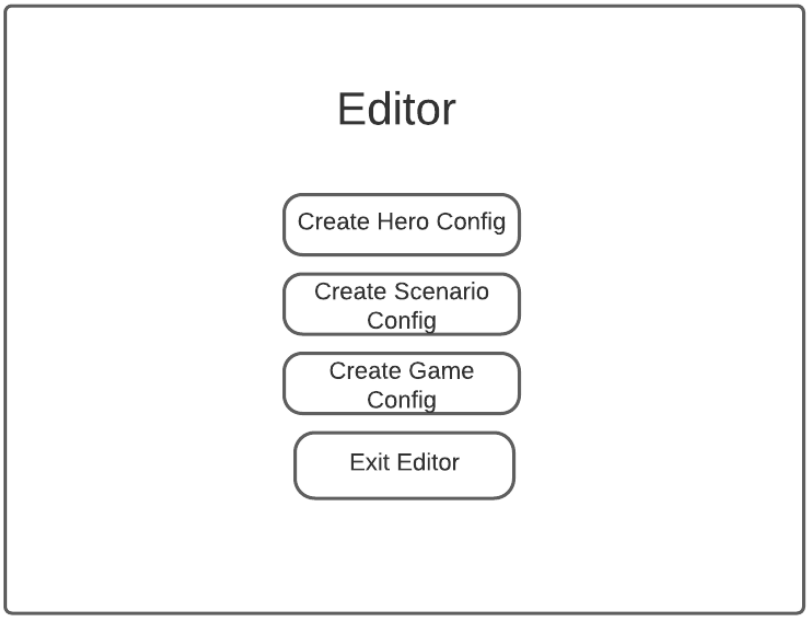

Abbildung 3.13: Hauptmenü

Beim Starten des Editors wird sich ein Hauptmenü öffnen, über das man auf alle grundlegenden Funktionen zugreifen kann. Dazu zählen das Erstellen von Szenario-/Partie- und Heldenkonfiguration. Die jeweiligen Konfigurationen erstellt/bearbeitet man auf verschiedenen Screens, die über die entsprechenden Buttons erreichbar sind. Die Screens werden auf den folgenden Seiten weiter beschrieben. Zusätzlich soll man die Möglichkeit haben den Editor über den “Exit”-Button wieder ordnungsgemäß beenden zu können.

Helden-Konfiguration

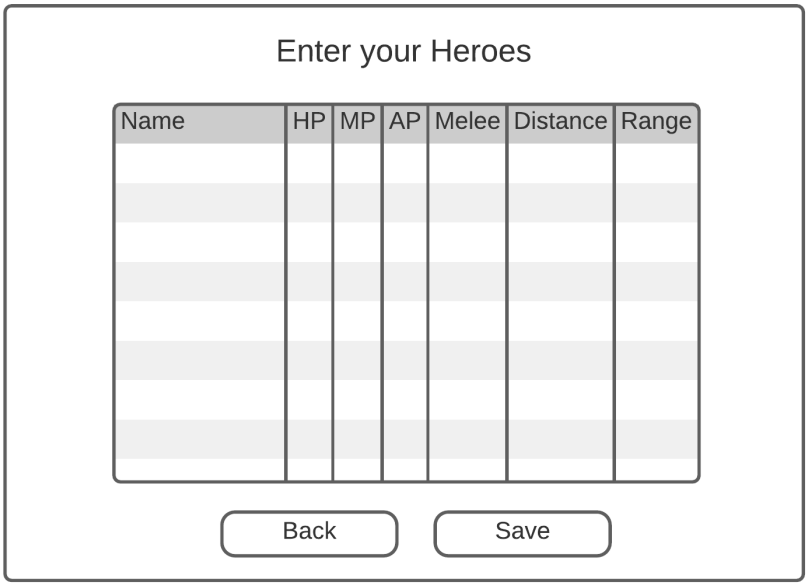

Abbildung 3.14: Helden-Konfiguration

Zur Erstellung und Bearbeitung von Heldenkofigurationen wird eine Tabelle verwendet. In dieser können alle Attribute und Werte bequem über Tasteneingabe gegeben werden. Über den ’Save’-Button kann die Konfiguration gespeichert werden, sofern diese korrekt ist, und wird im Standardverzeichnis des Editors abgelegt mit einem vom Editor generierten Namen. Über ein Pop-Up wird dem Benutzer dann angegeben, ob die Aktion erfolgreich war, oder ob es zu einem Fehler kam. Dazu wird ein der entsprechende Fehler ausgegeben und wenn möglich ein Verbesserungs-/Korrekturvorschlag gegeben, sodass der Benutzer das Problem beheben kann. Bei

betätigen des ’Back’-Buttons werden die aktuellen Änderungen verworfen und man wird wieder zurück zum Hauptmenü geführt. Damit es dabei nicht ungewollt zum Verlust von Änderungen kommt, soll auch hier ein Pop-Up erscheinen, in dem der Benutzer die Aktion nochmals bestätigen muss.

Szenario-Konfiguration

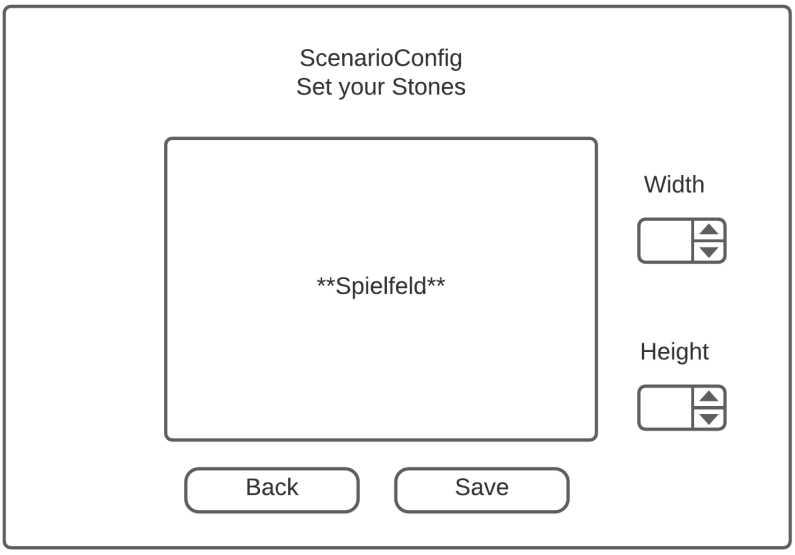

Abbildung 3.15: Szenario-Konfiguration

Bei der Szenariokonfiguration wird das Spielfeld entworfen. Daher erhält der Benutzer ein Spielfeld, wie es ihm aus dem Benutzer-Client bekannt ist. Durch einen Mausklick auf ein einzelnes Feld kann dabei ein Stein gesetzt werden. Über die beiden Eingabefelder links des Spielfeldes lassen sich Höhe und Breite einstellen, je nach Belieben kann dabei die Zahl direkt eingegeben werden oder über die Stepper die entsprechende Zahl eingestellt werden. Back- und Save-Button funktionieren dabei identisch zur Heldenkonfiguration. Über den Save-Button werden alle Änderungen gesichert und eine neue Datei angelegt, sofern der Benutzer keine ungültigen Werte eingetragen hat. Über den Back-Button gelangt man wieder zurück ins Hauptmenü, sofern man dies im folgenden Pop-up bestätigt.

Partie-Konfiguration

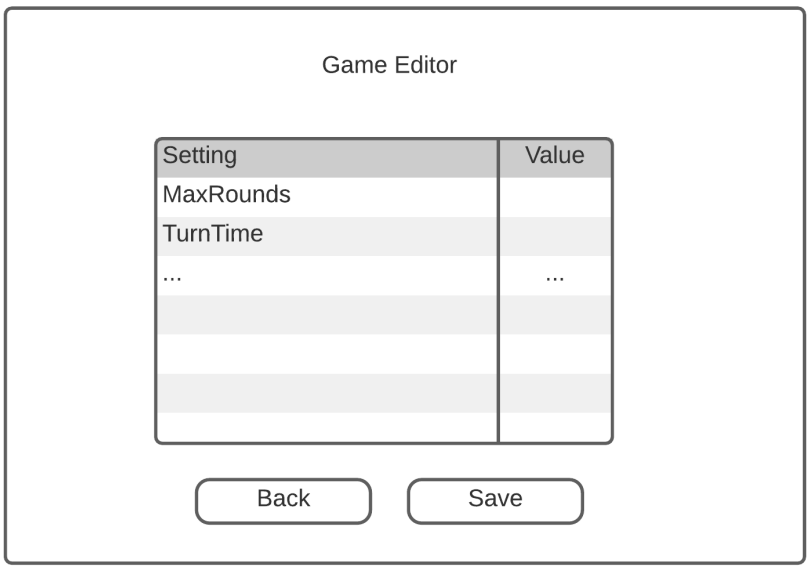

Abbildung 3.16: Partie-Konfiguration

Ähnlich wie bei der Heldenkonfiguration erhält man zum Editieren von Partie-Konfigurationen eine Tabelle. In diesem Fall enthält diese aber nur einfach Schlüssel-Wert Paare, die beim Speichern leicht geprüft werden können. Zudem sol die Tabelle in jeder Zeile nur eine bestimmte Eingabe zulassen, sodass zu gewissem Maß eingeschränkt werden kann, dass ungültige Eingabe getätigt werden, beispielsweise durch die Verwendung eines unzulässigen Datentyps. Save- und Back-Button werden hier äquivalent, wie auch auf den anderen Screens verwendet.

Nutzungskonzept

Dialogstrukturdiagramm

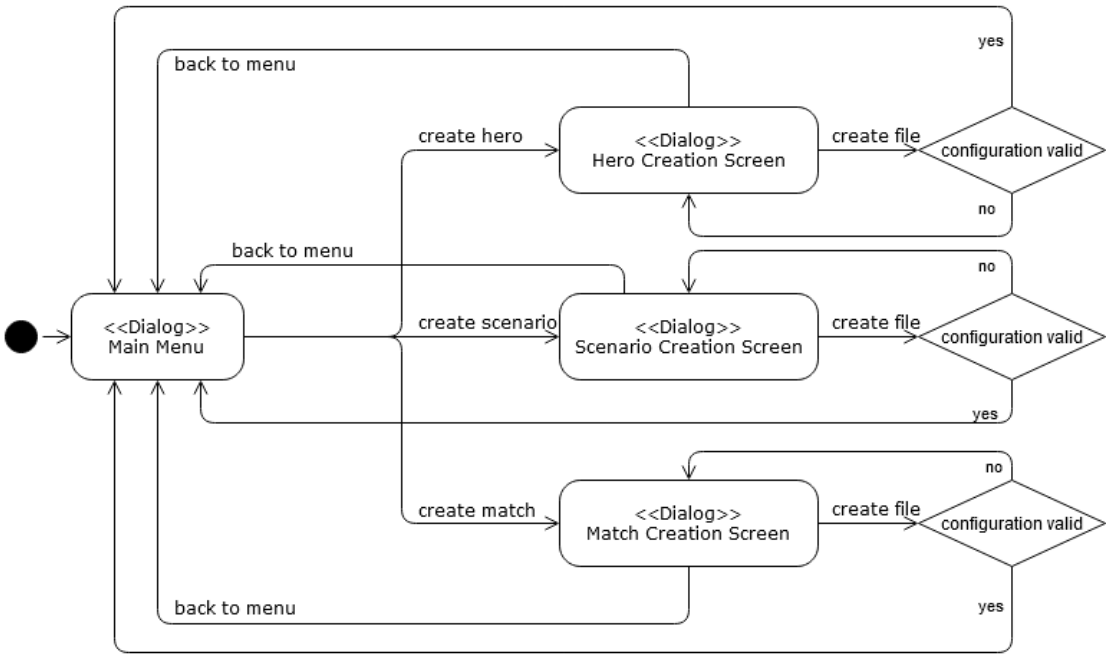

Abbildung 3.17: Dialogstrukturdiagramm Editor

Dieses Dialogstrukturdiagramm zeigt, wie man vom Hauptmenü in die entsprechenden Screens zur Konfiguartion von Held, Szenario oder Partie gelangt. Wenn man die Konfiguration speichert, um eine Datei zu erstellen, wird kontrolliert, ob die Konfiguration valide ist. Ist sie gültig gelangt man wieder ins Hauptmenü, wenn nicht, ist man wieder im entsprechenden Konfigurationsscreen.

Anwendungsfalldiagramm

Erklärungen:

- «include» zwischen “Spielkonfiguration erstellen” und “Konfiguration überprüfen”: Wann immer eine Spielkonfiguration erstellt wird, soll der Editor automatisch die Konfiguration auf Gültigkeit prüfen.

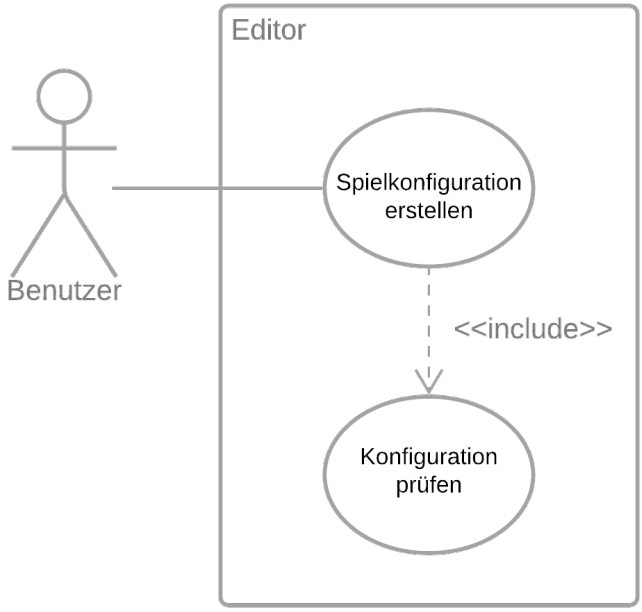

Abbildung 3.18: Anwendungsfalldiagramm Editor

Anforderungen:

- Spielkonfiguration erstellen: —
- Konfiguration prüfen: NF-4

3.5 KI-Client

Der KI-Client verwendet keine GUI sondern ein Komandozeileninterface, da er nur vom KI-Administrator gestartet wird. Der Administrator benötigt keine GUI, da davon ausgegangen werden kann, dass er mit der Anwendung, sowie der Kommandozeile vertraut ist.

3.6 Datenmodell

Im Folgenden werden die Kern Klassen, die für das Spiel benötigt werden, mit ihren Attributen, Methoden und Beziehungen dargestellt. Es handelt sich hierbei um kein vollstäniges Klassendiagramm. Somit sind die hier gezeigten Klassen nicht als Implementierungsklassen zu verstehen, sondern als Konzepte der Andwenungsdomäne.

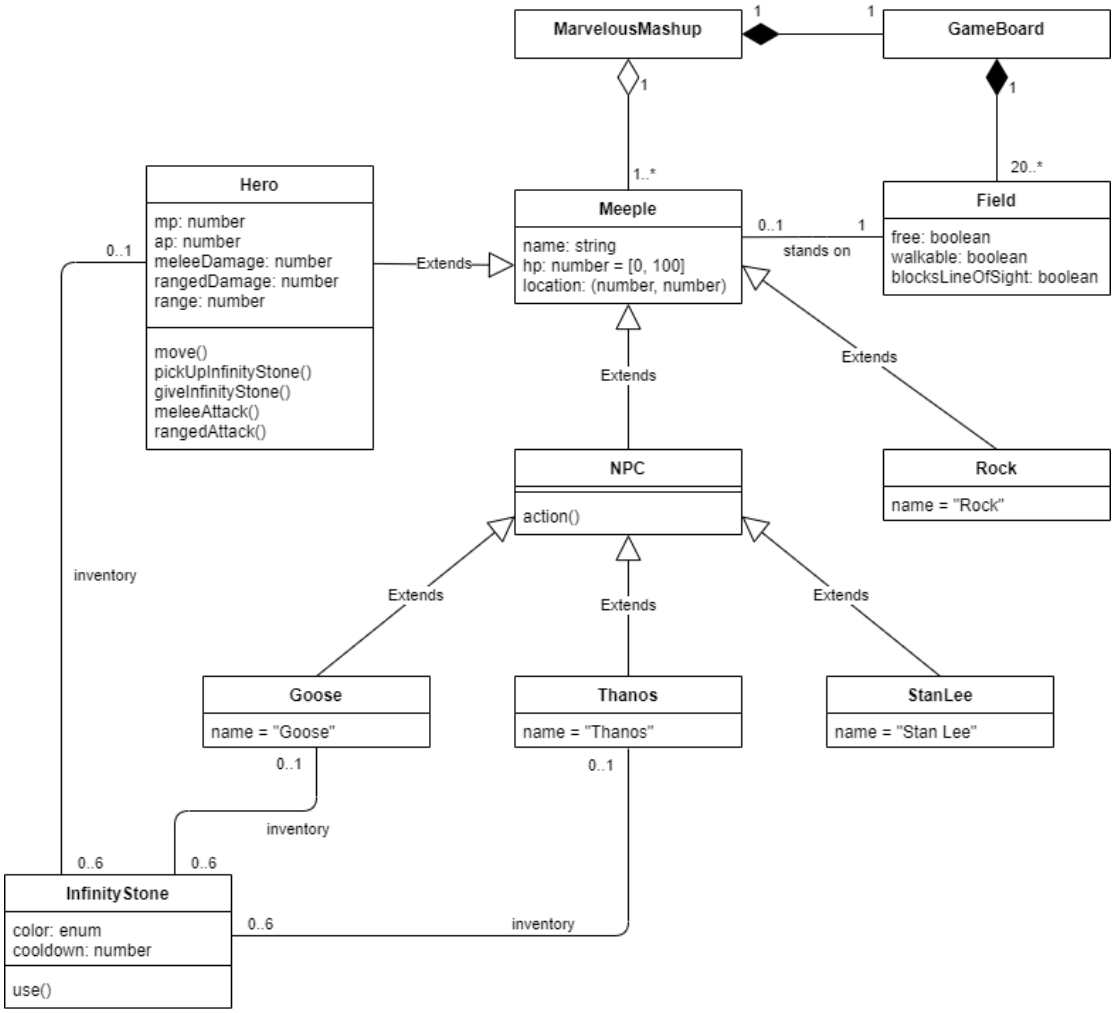

Abbildung 3.19: Domänenmodell

4 Randbedingungen

4.1 Qualität

AnforderungsNR	NF-1
Titel	Programmiersprachen
Beschreibung	Für das gesamte Projekt dürfen nur die Sprachen Java und Python verwendet werden. Java wird hierbei bevorzugt.
Begründung	Insgesamt sollten nur Sprachen verwendet werden, mit denen alle im Team vertraut sind. In manchen Fällen könnte das im Team nicht ganz so verbreitete Python aber vorteilhaft sein.
Überprüfung	Betrachten des Codes: Falls andere Programmiersprachen als Java und Python verwendet wurden, so ist diese Anforderung nicht erfüllt. Die Teile des Codes, die in Python verfasst wurden benötigen zusätzlich eine sinnvolle Begründung, warum Python bevorzugt wurde.
Maßnahmen zur Erfüllung	Es werden im gesamten Entwicklungsprozess nur Java und Python eingesetzt, für Python wird vorher im Team diskutiert, ob das an der fraglichen Stelle sinnvoll ist.
Priorität	Muss
Abhängigkeiten	

AnforderungsNR	NF-2
Titel	Betriebssystem
Beschreibung	Der Benutzer-Client und Editor sollen mindestens unter Windows oder Linux lauffähig sein. Der KI-Client und der Server müssen mithilfe von Docker unabhängig vom Betriebssystem lauffähig sein.
Begründung	Für den Benutzer-Client und den Editor genügt es, ein Betriebssystem zu unterstützen, auf das die meisten Menschen Zugriff haben. KI-Client und Server sollen auch für Tuniere, etc. auf "ungewöhnlichen" Betriebssystemen laufen können, hier kann aber dafür davon ausgegangen werden, dass die Person, die die Installation vornimmt, technisch versiert genug ist, um mit Docker umzugehen.
Überprüfung	Benutzer-Client und Editor werden sowohl unter Windows, als auch unter Linux getestet, bei Server und KI-Client genügt es, zu überprüfen, dass sie in einem Docker Container laufen.
Maßnahmen zur Erfüllung	Schon während dem Entwicklungsprozess werden Benutzer-Client und Editor (sobald letzterer eingekauft ist, wobei beim Kauf auch auf die Kompatibilität geachtet werden muss) regelmäßig auf beiden Betriebssystemen getestet. KI-Client und Server werden regelmäßig in Docker Containern getestet, um sicherzustellen, dass sie in Docker funktionieren.
Priorität	Muss
Abhängigkeiten	

AnforderungsNR	NF-3
Titel	Netzwerkkommunikation
Beschreibung	Der vom Standardisierungskomitee ausgearbeitete Netzwerkstandard, sowie das WebSocket-Protokoll werden eingehalten.
Begründung	Dies ist nötig, damit der Server und Clients mit denen von anderen Teams kombiniert werden können.
Überprüfung	Die Tests (siehe Maßnahmen zur Erfüllung) müssen erfolgreich durchlaufen und der Server und die Clients müssen jeweils mit den Komponenten mindestens 2 anderer Teams verwendet werden können.
Maßnahmen zur Erfüllung	Es werden Tests geschrieben und regelmäßig ausgeführt, die den Netzwerkstandard abprüfen. Falls diese Tests nicht erfolgreich durchlaufen, wird der Netzwerkcode entsprechend korrigiert. Bezüglich des WebSocket-Protokolls wird eine Library eingesetzt, die dieses für uns implementiert.
Priorität	Muss
Abhängigkeiten	

AnforderungsNR	NF-4
Titel	Standard für Konfigurationsdateien
Beschreibung	Die vom Standardisierungskomitee ausgearbeiteten Schemata für Konfigurationsdateien werden eingehalten.
Begründung	Dies ist essentiell, damit der Server und der zugekaufte Editor kompatibel zueinander sind.
Überprüfung	Server und zugekaufter Editor müssen kompatibel sein, die Tests (siehe Maßnahmen zur Erfüllung) müssen erfolgreich durchlaufen werden.
Maßnahmen zur Erfüllung	Es werden Tests geschrieben und regelmäßig ausgeführt, die beim Server das Auslesen von standardgemäßen Konfigurationsdateien abprüfen. Falls diese Tests nicht erfolgreich abgeschlossen werden, wird der Parser entsprechend korrigiert. Beim Kauf des Editors wird darauf geachtet, dass dieser die Einhaltung des Standards bei mit ihm erzeugten Konfigurationsdateien sicherstellt.
Priorität	Muss
Abhängigkeiten	

AnforderungsNR	NF-5
Titel	Implementierungssprache
Beschreibung	Die Implementierungssprache ist Englisch.
Begründung	Verbindliche Forderung des Lastenhefts. Desweiteren ist eine englische Implementierungsprache internationaler Standard.
Überprüfung	Nachlesen im Code.
Maßnahmen zur Erfüllung	Alle Entwickler achten während dem gesamten Entwicklungsprozess darauf, keine andere Sprache in Kommentaren oder Bezeichnern im Code zu verwenden.
Priorität	Muss
Abhängigkeiten	

AnforderungsNR	NF-6
Titel	Versionierung
Beschreibung	Der Code wird über git in einem Repository unter `https://gitlab.informatik.uni-ulm.de` versioniert.
Begründung	Eine solche Versionierung ist sinnvoll und das Repository ist vom Kunden vorgegeben.
Überprüfung	Der Code sollte sich im gitlab befinden und an der Historie sollte erkennbar sein, dass mehrere Branches zur Entwicklung verwendet wurden.
Maßnahmen zur Erfüllung	Git wird während dem gesamten Entwicklungsprozess (unter Verwendung des oben genannten Repositories) eingesetzt.
Priorität	Muss
Abhängigkeiten	

AnforderungsNR	NF-7
Titel	Qualitätssicherung
Beschreibung	Es wird das Tool SonarQube verwendet, um vor jedem Push in das Repository die Qualität der Software zu überprüfen und bei Bedarf nachzubessern.
Begründung	Der Kunde wünscht dies so, außerdem ist eine hohe Qualität für uns wichtig, da Code mit niedriger Qualität den Entwicklungsprozess erschwert.
Überprüfung	SonarQube muss zu jedem Zeitpunkt für die im Repository vorhandene Version des Codes `passed` zurückgeben. Temporäre Außnahmen für Branches, auf denen aktiv gearbeitet wird sind nach Absprache möglich.
Maßnahmen zur Erfüllung	(siehe Beschreibung)
Priorität	Muss
Abhängigkeiten	

AnforderungsNR	NF-8
Titel	Entwicklungsprozess
Beschreibung	In der Implementierungsphase wird ein an Scrum orientierter agiler Prozess durchgeführt.
Begründung	Derartige Entwicklungsprozesse sind in der Softwareentwicklung aus verschiedenen Gründen üblich. Darüber hinaus wird dies vom Kunden so gewünscht.
Überprüfung	Anhand der Zeiterfassung und über die Scrum-Meetings (Kontrolle durch den Tutor).
Maßnahmen zur Erfüllung	Die Einhaltung des Scrum Prozesses wird regelmäßig überprüft und bei Bedarf nachgesteuert.
Priorität	Muss
Abhängigkeiten	

4.2 Betriebskonzept

Systemumgebung

Das System muss mindestens auf aktueller, für den Consumer-Markt bestimmter Hardware flüssig lauffähig sein, d.h. ohne grafische “Ruckler” (bei Komponenten mit GUI) oder übermäßig lange Wartezeiten. Bezüglich der Betriebssysteme, auf denen die einzelnen Komponenten lauffähig sein müssen siehe NF-1 (Kapitel 4.1, Seite 50).

Abhängigkeiten von Produkten Dritter

Zum derzeitigen Stand werden die folgenden von Dritten entwickelten Produkte eingesetzt:

- Editor: von einem anderen Team zugekauft

Bibliotheken

Die folgenden Programmbibliotheken werden zum derzeitigen Stand eingesetzt:

- libgdx: `https://libgdx.com/`, Lizenz: Apache 2.0
- gson: `https://github.com/google/gson`, Lizenz: Apache 2.0
- junit: `https://junit.org/junit5/`, Lizenz: Eclipse Public License 2.0

Entwicklungstools

Die folgenden Entwicklungstools werden zum derzeitigen Stand eingesetzt:

- Docker: `https://www.docker.com/`
- SonarQube: `https://www.sonarqube.org/`
- Git: `https://git-scm.com/`
- GitLab (Instanz des Kunden): `https://gitlab.informatik.uni-ulm.de/`
- Gradle: `https://gradle.org/`
- IntelliJ IDEA: `https://www.jetbrains.com/idea/`
- PyCharm: `https://www.jetbrains.com/pycharm/`

Schulungskonzept

Der Benutzer-Client, der KI-Client und der Server werden mit einer Anleitung bereitgestellt. Diese enthält notwendige und nützliche Informationen, die den Benutzern und Administratoren das Nutzen der Komponenten ermöglichen. Sollten Benutzern im Rahmen dieses Selbststudiums weitere Fragen aufkommen oder sollten sie bestimmte Aspekte der Software nicht verstehen, können sie diese Fragen an die in den Anleitungen bereitgestellten Support-E-Mail-Adressen stellen und werden eine individualisierte Schulung erhalten.

4.3 Entwicklungsvorgaben

Um die Entwicklung des Projekts einheitlich zu gestalten, werden an dieser Stelle einige Vorgaben gemacht.

Projekt

Der Entwicklungsprozess entspricht dem Scrum Modell. Dieses agile Vorgehen ermöglicht selbst bei flukturierenden Anforderungen ein Produkt zu erzeugen, welches höchsten Qualitätsansprüchen genügt. In regelmäßigen Abständen finden Telefonkonferenzen mit dem Kunden statt. Hierbei wird dem Kunden der aktuelle Stand des Projekts präsentiert, sowie das weitere Vorgehen abgestimmt.

Repository

Der Quellcode wird in einem Git Repository, welches über die GitLab Instanz des Kunden bereits zur Verfügung gestellt wurde, verwaltet. Um die Lesbarkeit der History zu gewährleisten werden Commit Messages nach der Conventional Commits Spezifikation verfasst.

Java

Das Projekt wird hauptsächlich in der Programmiersprache Java implementiert. Als Buildtool kommt Gradle zum Einsatz. Entwickelt wird der Quellcode nach dem Google Style Guide for Java in der IDE IntelliJ IDEA. Hierbei werden Klassen und Methoden stets mit JavaDoc kommentiert. Um eine hohe Code- und Produktqualität sicherzustellen, wird das Analyseprogramm SonarQube eingesetzt. Außerdem werden für alle außer den GUI Klassen Unit Tests erstellt. Hierzu wird das JUnit Framework verwendet.

Python

Für ausgewählte Teile des Projekts kann auch die Programmiersprache Python verwendet werden. In welchen Situationen dies konkret der Fall ist, wird das Entwick-

lerteam gegebenenfalls abwägen. Für Python Code gelten, sofern anwendbar, die gleichen Vorgaben wie für Java Code. Der Quellcode folgt dem Google Python Style Guide. Als IDE wird PyCharm verwendet.

4.4 Abnahmekriterien

Die Abnahme des Produkts erfolgt gemeinsam mit dem Kunden. Hierbei werden alle Anforderungen auf Einhaltung geprüft. Hierbei ist ein besonderes Augemerk darauf zu setzen, dass das Spiel Marvelous Mashup korrekt umgesetzt wurde und den Spielern ein tolles Spielerlebnis geboten wird.

Ein weiteres kritisches Abnahmekriterium ist die korrekte Umsetzung des vom Standardkomitee herausgegebenen Standards. Konkret bedeutet dies, dass der entwickelte Server mit dem Konfigurationsdateiformat des Standards kompatibel sein muss. Außerdem muss der Server mit beliebigen, dem Standard entsprechenden Clients kompatibel sein. Für den Benutzer-Client und den KI-Client gilt, dass diese mit beliebigen, dem Standard entsprechenden Servern kompatibel sein müssen.