



ulm university universität
uulm

**Fakultät für
Ingenieurwissenschaften,
Informatik und
Psychologie**

Institut für Software-
technik und Program-
miersprachen

Softwaregrundprojekt Meilenstein 1

Softwaregrundprojekt an der Universität Ulm

Vorgelegt von:

Gruppe 10

Dozent:

Florian Ege

Betreuer:

Stefanos Mytilineos

2018

Inhaltsverzeichnis

1	Kontextanalyse	3
2	Fachwissen	3
3	Domänenmodell	9
4	Anforderungsdefinition	10
4.1	Akteure	10
4.2	Allgemeine funktionale Anforderungen	13
4.3	Server spezifische funktionale Anforderungen	33
4.4	Client spezifische Funktionale Anforderungen	33
4.5	Quidditchteam-Editor spezifische funktionale Anforderungen	37
4.6	Nicht funktionale Anforderungen	38

1 Kontextanalyse

2 Fachwissen

Begriff	Spieler
Beschreibung	Ein Nutzer, der das Computerspiel "Fatastic Feasts" spielt.
Ist-ein	Nutzer
Kann-sein	-
Aspekt	Zur Beschreibung des Programmverlaufs
Bemerkung	-
Beispiel	-

Begriff	Nutzer
Beschreibung	Ein Mensch, der einen Rechner bedient und entweder den Client zum Spielen des Spiels oder zur Beobachtung einer Partie benutzt, oder den Team-Editor bedient. Jeder Benutzer hat einen Nutzernamen, mittels dem er von anderen Nutzern erkannt werden kann.
Ist-ein	-
Kann-sein	Spieler, Gast
Aspekt	Zur Beschreibung des Programmverlaufs
Bemerkung	-
Beispiel	JägerMaister69

Begriff	Gast
Beschreibung	Ein Nutzer, der eine laufende Partie beobachtet
Ist-ein	Nutzer
Kann-sein	-
Aspekt	Zur Beschreibung des Programmverlaufs
Bemerkung	-
Beispiel	-

Begriff	Client
Beschreibung	Das Computerprogramm, das mit einer grafischen Oberfläche ausgestattet ist und einem Nutzer erlaubt, eine Verbindung mit einem Server herzustellen und damit zu Kommunizieren
Ist-ein	-
Kann-sein	-
Aspekt	Zum spielen des Spiels "Fantastic Feasts"
Bemerkung	Der Begriff bezieht sich nicht auf den Menschen, der das Programm bedient
Beispiel	-

Begriff	Server
Beschreibung	Die zentrale Komponente, in dem die Spiellogik implementiert ist und die Programmbefehle abwickelt und mit dem sich Clients verbinden können, um eine Partie zu spielen oder zu beobachten. Die Kommunikation erfolgt mit JSON
Ist-ein	-
Kann-sein	-
Aspekt	Ist für die Kommunikation von Clients, für das Verwalten des Spielgeschehens, Ressourcenverwaltung und die Spiellogik verantwortlich.
Bemerkung	-
Beispiel	-

Begriff	Team-Editor
Beschreibung	Ermöglicht einem Nutzer mit einer grafischen Oberfläche, ein eigenes Team zu erstellen und zu bearbeiten. Die Einstellungen werden danach als JSON-Datei gespeichert.
Ist-ein	-
Kann-sein	-
Aspekt	Zur Erstellung von Nutzereigenen Teams.
Bemerkung	-
Beispiel	-

Begriff	KI-Client
Beschreibung	Meldet sich beim Server wie ein normaler Client an und simuliert mit einer KI einen menschlichen Spieler. Hat keine grafische Oberfläche. Meldet sich mit dem Nutzernamen "Klëin.
Ist-ein	-
Kann-sein	-
Aspekt	Zum Spielen gegen einen Computergegner
Bemerkung	-
Beispiel	-

Begriff	KI
Beschreibung	Definiert die Regeln, nach denen der KI-Client auf die durch den Server vermittelten Geschehen im Spiel reagiert.
Ist-ein	-
Kann-sein	-
Aspekt	Zum Spielen gegen einen Computergegner
Bemerkung	Die KI ist die Logik, nach der der Computer das Spiel spielt und kein Programm
Beispiel	-

Begriff	Spielfeld
Beschreibung	Ein grafisch darstellbares Raster, auf dem sich die Spielfiguren bewegen
Ist-ein	-
Kann-sein	-
Aspekt	Dient als virtuelles Spielbrett mit klar definierten Abgrenzungen
Bemerkung	Wird nicht Spielumgebung genannt um Verwechslung mit dem Client zu vermeiden
Beispiel	-

Begriff	Zelle
Beschreibung	Die kleinste Einheit des Spielfeldes, also ein Quadrat davon
Ist-ein	-
Kann-sein	Zentrumszelle, Zielzelle, kritische Zelle
Aspekt	Mögliche Standorte der Spielfiguren
Bemerkung	Wird nicht Feld genannt, da das ein eher vager Begriff ist
Beispiel	-

Begriff	Zentrum
Beschreibung	Der 3x3 Zellen große Abschnitt in der Mitte des Spielfeldes
Ist-ein	-
Kann-sein	-
Aspekt	Summe aller Zentrumszellen
Bemerkung	Ist das Mittelfeld im Lastenheft
Beispiel	-

Begriff	kritischer Bereich
Beschreibung	Die Bereiche am linken und rechten Rand des Spielfeldes, in dem sich die Zielzellen befinden
Ist-ein	-
Kann-sein	-
Aspekt	Summe aller kritischen Zellen und Zielzellen
Bemerkung	Hüterzone im Lastenheft
Beispiel	-

Begriff	Zielzelle
Beschreibung	Die Zellen in die beide Teams die Payload bewegen wollen
Ist-ein	Zelle
Kann-sein	-
Aspekt	Hauptquelle von Punkten
Bemerkung	Torring im Lastenheft
Beispiel	

Begriff	Zentrumszelle
Beschreibung	Eine Zelle im Zentrum des Spielfeldes (siehe Zentrum)
Ist-ein	Zelle
Kann-sein	-
Aspekt	Startpunkt für Payload und Quälgeister
Bemerkung	-
Beispiel	-

Begriff	kritische Zelle
Beschreibung	Eine Zelle in einem kritischen Bereich des Spielfeldes
Ist-ein	Zelle
Kann-sein	-
Aspekt	limitierendes Element für das Abliefern der Payload
Bemerkung	-
Beispiel	-

Begriff	Figur
Beschreibung	Jedes Objekt, das sich auf dem Spielfeld befindet und darauf bewegt werden kann
Ist-ein	-
Kann-sein	Neutrale Figur, Spielfigur
Aspekt	Spielobjekte
Bemerkung	Nicht zu verwechseln mit Spielfigur
Beispiel	-

Begriff	Neutrale Figur
Beschreibung	Eine Figur, die nicht direkt, nur indirekt von einem Spieler beeinflusst werden kann
Ist-ein	Figur
Kann-sein	Payload, Quälgeist, Kostbarkeit
Aspekt	Festpunkte zur Steuerung des Spielverlaufs
Bemerkung	Bälle im Lastenheft
Beispiel	-

Begriff	Spielfigur
Beschreibung	Eine Figur, die von einem Spieler direkt gesteuert wird. Jede Spielfigur hat einen Namen, einen Rang und eine Rolle
Ist-ein	Figur
Kann-sein	Wächter, Sucher, Jäger, Kämpfer
Aspekt	Mitglieder eines Teams
Bemerkung	Spieler im Lastenheft
Beispiel	Luke Skywalker, Wächter, Rang 5

Begriff	arg1
Beschreibung	arg2
Ist-ein	arg3
Kann-sein	arg4
Aspekt	arg5
Bemerkung	arg6
Beispiel	arg7

Begriff	Spielfigur
Beschreibung	Jedes Objekt, das sich auf dem Spielfeld befindet
Ist-ein	arg3
Kann-sein	arg4
Aspekt	arg5
Bemerkung	arg6
Beispiel	arg7



4 Anforderungsdefinition

4.1 Akteure

ID:	AKT1
Titel:	Nutzer
Beschreibung:	Menschlicher Nutzer, der eine Anwendungen bedient.
Rolle:	Ein Mensch, welcher entweder als Spieler Aktiv an einem Spiel teilnimmt oder als Gas passiv einem Spiel zusieht oder den Team-Konfigurator benutzt um ein Team zu erstellen oder an zu passen.

ID:	AKT2
Titel:	Spieler
Beschreibung:	Nimmt aktiv Einfluss auf das Spielgeschehen.
Rolle:	Entweder Nutzer oder KI. Nimmt aktiv Einfluss auf das Spielgeschehen indem er Züge vorgibt, wenn er an der Reihe ist. Muss vor Spielbeginn eine Konfiguration für sein Team angeben.

ID:	AKT3
Titel:	Gast
Beschreibung:	Beobachtet eine Partie als Außenstehender.
Rolle:	Nutzer, der mit der Client-Anwendung ein laufendes Spiel beobachtet, jedoch keinen Einfluss auf das Spielgeschehen nehmen kann.

ID:	AKT4
Titel:	Systemadministrator
Beschreibung:	Anwendungsnutzer mit Zusatzqualifikation um Server zu verwalten.
Rolle:	Der Systemadministrator hat zugriff auf die Server-Anwendung. Er ist dafür verantwortlich, eine Instanz der Server Anwendung zu starten und zu betreuen. Zudem hat er Zugriff auf die Partie-Konfiguration und kann diese bei Bedarf verändern.

ID:	AKT5
Titel:	KI
Beschreibung:	Vom Computer gesteuerter Spieler.
Rolle:	Spieler, dessen Entscheidungen und Züge auf einem Computer von der KI-Client Software getroffen werden.

ID:	AKT6
Titel:	Kunde
Beschreibung:	SoPra-Tutor
Rolle:	Der Kunde kann Anforderungen und zusätzliche Wünsche an das Produkt stellen. Er entscheidet schlussendlich auch darüber, ob das Endprodukt den Anforderungen genügt.

ID:	AKT7
Titel:	Client
Beschreibung:	Software eines Nutzers
Rolle:	Mit Hilfe der Client Software kann der Nutzer entweder als Spieler oder als Gast einem von einem Server zur Verfügung gestellten Spiel beitreten.

ID:	AKT8
Titel:	KI-Client
Beschreibung:	Anwendung, welche die KI steuert.
Rolle:	Der KI-Client ist die Anwendung, welche eine KI steuert. Der Nutzer kann von Außen keinen Einfluss auf die Züge und Entscheidungen des KI-Clients nehmen.

ID:	AKT9
Titel:	Server
Beschreibung:	Zentraler Computer mit spezieller Software zu dem alle am beteiligten Clients ein Verbindung aufbauen.
Rolle:	Auf dem Server läuft die eigentliche Spiellogik. Er fungiert dabei als Bindeglied zwischen den am Spiel beteiligten Clients und stellt für diese alle benötigten Informationen, wie etwas die Spielfeldkonfiguration oder die Züge des Gegners bereit.

ID:	AKT10
Titel:	Quidditchteam-Editor
Beschreibung:	Editor für die Team-Konfiguration.
Rolle:	Mit dieser Anwendung kann ein Nutzer sein für eine Partie gewünschtes Quidditchteam im Rahmen bestimmter Grenzen beliebig konfigurieren.

ID:	AKT11
Titel:	Team
Beschreibung:	Einheit aus Spielfiguren die einem spielenden Nutzer zugeordnet ist.
Rolle:	Im Spiel ist jedem Spieler ein Team aus 7 Spielfiguren zugeordnet. Die Zusammensetzung des Team lässt sich mit der Quidditchteam-Editor den persönlichen Wünschen entsprechend anpassen.

ID:	AKT12
Titel:	Schiedsrichter
Beschreibung:	Ahndet Fouls.
Rolle:	Der Schiedsrichter wird durch eine Software Repräsentiert die mit Hilfe eines Zufallsgenerator entscheidet, ob ein Foul geahndet wird oder nicht.

ID:	AKT13
Titel:	Spiel
Beschreibung:	Eine Partie Fantastic Feasts.
Rolle:	Ein Spiel besteht aus genau 2 Teams, Fans, dem Schiedsrichter und einem Spielfeld. Das Steuern der Spiellogik übernimmt der Server. Die Spieler der beiden Teams spielen gegeneinander um den Sieg.

4.2 Allgemeine funktionale Anforderungen

ID:	FA1
Titel:	Quidditch-Spielfeld
Beschreibung:	Das Quidditch-Spielfeld ist eine Ovale Form, welche in ein Raster von 17x13 quadratischen Feldern eingepasst ist. Auf diesem Feld finden alle Spielhandlungen statt, welche während dem Spiel getätigt werden können. Auf dem Spielfeld gibt es noch ein Mittelkreis und an den jeweils gegenüberliegenden Enden noch Hüterzonen.
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	++
Akteure:	ALLE?

ID:	FA2
Titel:	Mittelkreis
Beschreibung:	Der Mittelkreis ist ein Bereich auf dem Quidditch-Spielfeld, welcher in der Mitte angeordnet ist und aus 3x3 quadratischen Kacheln besteht. In dem Mittelkreis befindet sich das Mittelfeld, welches die mittlere Kachel des Mittelkreises ist und durch ein <i>M</i> gekennzeichnet ist.
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	FA1
Priorität:	++
Akteure:	<Akteure>

ID:	FA3
Titel:	Mittelfeld
Beschreibung:	Das Mittelfeld stellt den mittleren Punkt des Mittelkreises dar, welcher im Zentrum des Quidditch-Spielfeldes ist.
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	FA2
Priorität:	++
Akteure:	<Akteure>

ID:	FA4
Titel:	Hüterzone
Beschreibung:	Die Hüterzonen sind jeweils an den jeweils gegenüberliegenden Seiten des Quidditch-Spielfeldes. Die Hüterzonen beinhalten jeweils 3 Torringe und sind somit die Zonen in den die Teams Punkten können. Die Hüterzonen sind in einer Ovalen Form, welche 11 quadratische Felder hoch ist und 5 quadratische Felder breit ist an den jeweiligen Maximalen Punkten.
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor. Außerdem beinhaltet diese Zone die Torringe, durch welche die Teams Punkte machen können.
Abhängigkeiten:	FA1
Priorität:	++
Akteure:	<Akteure>

ID:	FA5
Titel:	Torring
Beschreibung:	Die Torringe sind in der Hüterzone angebracht und dazu da, dass die Teams jeweils Punkten können. Die 3 Torringe werden jeweils von einem Torhüter bewacht, welcher verhindern kann, dass das gegnerische Team ein Tor schießen kann.
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	FA4
Priorität:	++
Akteure:	<Akteure>

ID:	FA6
Titel:	Schussvektor
Beschreibung:	Als Schussvektor wird ein möglicher Wurf bezeichnet, um einen Torring zu treffen. Der Schussvektor geht dabei vom Mittelpunkt des Rasterfeldes, von dem geworfen wird zu dem Feld, zu dem geworfen wird gerade. Alle Rasterfelder, die dabei geschnitten werden, sind so genannte überstrichene Felder. Dies gilt aber nur für Felder, die wirklich geschnitten wurden und nicht nur an einer Ecke tangiert wurden. Um einen möglichen Torschuss zu erzielen, muss der Torschussvektor mit einem Winkel unter 90 Grad auf das Feld eintreffen, andernfalls ist der Torschuss nicht möglich.
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	FA4
Priorität:	++
Akteure:	<Akteure>

ID:	FA7
Titel:	Entfernung
Beschreibung:	Die Entfernung ist die kleinst mögliche Anzahl an Zügen, die man braucht, um Feld A zu Feld B zu kommen, wobei man sich in alle Richtungen bewegen darf, also Vertikal, Horizontal und Diagonal. Dies entspricht allerdings nicht immer den überstrichenen Feldern -1!.
Begründung:	Die Entfernung ist wichtig für das Spiel, um angeben zu können, welcher Wurf am besten geeignet ist und welcher der ganzen möglichen Schussvektoren am idealsten ist von der Entfernung.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	++
Akteure:	<Akteure>

ID:	FA8
Titel:	Bälle
Beschreibung:	Es gibt 3 verschiedene Arten von Bällen. So existieren der Quaffel, der Klatscher und der Schnatz. Jeder Ball hat seine eigenen spezifischen Fähigkeiten und eigene Funktion, die er für das Spiel erfordert. In einem Quidditch Spiel kommen 4 Bälle vor. Davon ist einer ein Quaffel, einer ein Schnatz und zwei sind Klatscher.
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	FA1
Priorität:	++
Akteure:	<Akteure>

ID:	FA9
Titel:	Quaffel [Ball]
Beschreibung:	Der Quaffel ist ein roter Lederball, welcher dazu dient, dass das TEam punkten kann. Dies geschieht in dem der Quaffel von den Jägern durch einen der 3 Torringe geschmießen wird. Dies gibt dann 10 Punkte für das Team. Da der Quaffel selber keine Flugfähigkeiten oder andere magische Fähigkeiten hat, muss er immer von einem Jäger mitgetragen werden oder geworfen werden oder von einem Hüter gefangen werden. ein Jäger oder Hüter kann den Quaffel aufnehmen, in dem der Ball er auf das Feld zieht, auf dem der Quaffel liegt, oder wenn der Quaffel auf einem Feld zum liegen kommt, auf dem ein Jäger oder Hüter steht. Zieht ein Treiber oder Sucher auf das Feld des Quaffels oder konmmt der Quaffel auf einem der Felder zum liegen, so hopst der Quaffel auf ein zufälliges freies Nachbarfeld.
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor. Der Quaffel wird benötigt, dass das Team Punkte machen kann.
Abhängigkeiten:	FA8
Priorität:	++
Akteure:	<Akteure>

ID:	FA10
Titel:	Klatscher [Ball]
Beschreibung:	Bei dem Spiel Quidditch gibt es 2 Klatscher, dies sind kleine schwarze Bälle, welcher magische Eigenschaften haben, was in dem Fall bedeutet, dass sie selber fliegen können. Das Ziel von den Klatschern ist es, die Spieler beider Teams von den Besen zu schmeißen. Dies geschieht, in dem der Klatscher sich jedes mal wenn er in der Rundenphase dran ist, einen zufälligen Spieler aussucht, welcher kein Treiber ist. Dann bewegt sich der Klatscher ein Feld auf den Spieler zu. Schafft der Klatscher es auf das selbe Feld wie der Spieler zu kommen, so wird dieser Spieler mit einer gewissen Wahrscheinlichkeit ausgenockt.
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor. Die Klatscher sind dazu da um dem Spiel einen Witz zu geben, dass manche Spieler zeitweise nicht verfügbar sind.
Abhängigkeiten:	FA8
Priorität:	++
Akteure:	<Akteure>

ID:	FA11
Titel:	Goldener Schnatz [Ball]
Beschreibung:	Der goldene Schnatz ist ein kleiner goldener Ball, welcher von dem Sucher gesucht werden muss. Der goldene Schnatz ist ebenfalls ein Ball, welcher eigenständig herumfliegt, wodurch er versucht zu vermeiden von den Suchern gefangen zu werden. Wird der Schnatz jedoch von einem der beiden Suchern gefangen, so gibt dieser 30 Punkte für das Team des Suchers, welcher den Schnatz gefangen hat und das Spiel ist beendet. In jeder Rundenphase sucht der Schnatz sich ein neues Feld aus, auf dass er sich dann eine Position weiter bewegt. Dies geschieht, in dem er den Sucher, der die geringste Entfernung hat nur berücksichtigt und dann unter allen Feldern, die eine größere Entfernung als seine aktuelle Position haben, ein zufälliges Feld weitergeht, so dass er sich von dem Sucher entfernt.
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor. Der Schnatz verleiht dem Spiel eine Besonderheit, so dass ein Team durch das fangen des Schnatzes die Chance hat, evtl doch noch zu gewinnen. Außerdem ist somit das Spiel zu ende.
Abhängigkeiten:	FA8
Priorität:	++
Akteure:	<Akteure>

ID:	FA12
Titel:	Besen
Beschreibung:	Die Besen sind essentiell für das Spiel, da alle 7 Spieler auf den Besen fliegen müssen. Es gibt im Fall von <i>Fantastic Feasts</i> verschiedene Besenmodelle, welche jeweils unterschiedliche Eigenschaften haben. Dies zeichnet sich dadurch aus, dass die Besen unterschiedliche Wahrscheinlichkeiten haben, mit denen der Spieler des Besens noch ein zweites Feld sich weiter bewegen darf. Dies bedeutet, je besser der Besen ist, desto wahrscheinlicher ist es, dass der Spieler ein zweites Feld weiter darf. Zusätzlich gilt die Besen-Repräsentanz-Regel, welche besagt, dass in jedem Team jeder Typ eines Besens einmal vorkommen muss.
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor. Die Besen sind wichtig, da die Spieler sich damit fort bewegen und die Besen auch noch Eigenschaften mitbringen, welche für das Spiel wichtig sind.
Abhängigkeiten:	FA1
Priorität:	++
Akteure:	<Akteure>

ID:	FA13
Titel:	Zauberfauch [Besen]
Beschreibung:	Der Zauberfauch ist der schlechteste Besen den ein Spieler auswählen kann. Die Wahrscheinlichkeit, dass der Spieler ein zweites Feld in einem Zug vorrücken kann beträgt bei dem Besen 10%.
Begründung:	Der Besen ist ein wichtiges Feature, welches Einfluss auf das Spiel hat, da je nach Besen der Spieler einen zweiten Zug machen darf.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA14
Titel:	Sauberwisch 11 [Besen]
Beschreibung:	Der Sauberwisch 11 ist der zweit schlechteste Besen den ein Spieler auswählen kann. Die Wahrscheinlichkeit, dass der Spieler ein zweites Feld in einem Zug vorrücken kann beträgt bei dem Besen 30%.
Begründung:	Der Besen ist ein wichtiges Feature, welches Einfluss auf das Spiel hat, da je nach Besen der Spieler einen zweiten Zug machen darf.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA15
Titel:	Komet 2-60 [Besen]
Beschreibung:	Der Komet 2-60 ist der mittel beste Besen den ein Spieler auswählen kann. Die Wahrscheinlichkeit, dass der Spieler ein zweites Feld in einem Zug vorrücken kann beträgt bei dem Besen 50%.
Begründung:	Der Besen ist ein wichtiges Feature, welches Einfluss auf das Spiel hat, da je nach Besen der Spieler einen zweiten Zug machen darf.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA16
Titel:	Nimbus 2001 [Besen]
Beschreibung:	Der Nimbus 2001 ist der zweit beste Besen den ein Spieler auswählen kann. Die Wahrscheinlichkeit, dass der Spieler ein zweites Feld in einem Zug vorrücken kann beträgt bei dem Besen 70%.
Begründung:	Der Besen ist ein wichtiges Feature, welches Einfluss auf das Spiel hat, da je nach Besen der Spieler einen zweiten Zug machen darf.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA17
Titel:	Feuerblitz [Besen]
Beschreibung:	Der Feuerblitz ist der beste Besen den ein Spieler auswählen kann. Die Wahrscheinlichkeit, dass der Spieler ein zweites Feld in einem Zug vorrücken kann beträgt bei dem Besen 90%.
Begründung:	Der Besen ist ein wichtiges Feature, welches Einfluss auf das Spiel hat, da je nach Besen der Spieler einen zweiten Zug machen darf.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA18
Titel:	Teams
Beschreibung:	An einem Spiel sind 2 Teams beteiligt, die gegeneinander spielen. Diese beiden Teams haben einen Namen, ein Team-Motto, ein Logo und eine Hauptfarbe für das Trickot, aber auch eine Ersatzfarbe, falls das andere Team eine ähnliche Farbe hat, gegen das gespielt wird. Jedes Team besteht zudem aus 7 Spieler, wovon einer ein Sucher ist, einer ein Hüter, 2 sind sogenannte Treiber und die restlichen 3 sind Jäger.
Begründung:	Die Teams sind notwendig, da sie der Hauptbestandteil sind, in dem sich alles vereint.
Abhängigkeiten:	FA1, FA19
Priorität:	++
Akteure:	<Akteure>

ID:	FA19
Titel:	Spieler
Beschreibung:	Es gibt 4 Arten von Spielern, welche in einem Quidditch Spiel unterschiedliche Aufgaben haben. Ein Team besteht insgesamt aus 7 Spielern, wovon es einen Sucher gibt, einen Hüter, 2 Treiber und 3 Jäger. Jeder dieser Spieler besitzt einen Namen und ein Geschlecht. Jedes Team muss dabei ausgeglichen sein, was das Geschlecht angeht, womit immer nur 4 Spieler dem selben Geschlecht angehören dürfen. In der Rundenphase, machen die Spieler zuerst ihre Bewegung, und dann ihre Aktion. Dies bedeutet, dass sie sich erst ein Feld weiterbewegen, und anschließend je nach Wahrscheinlichkeit des Besens ein zweites Feld weiterbewegen, und danach dann erst ihre Aktion wie Quaffel werfen oder Klatscher klopfen durchführen.
Begründung:	Spieler sind absolut notwendig für das Spiel, da sie die Aktionen durchführen, welche von dem Menschen gesteuert werden.
Abhängigkeiten:	FA1
Priorität:	++
Akteure:	<Akteure>

ID:	FA20
Titel:	Jäger [Spielertyp]
Beschreibung:	Die Jäger sind die Spieler auf dem Spielfeld, welche versuchen die Punkte für das Team mithilfe des Quaffels zu erzielen. Dabei darf jedoch immer nur ein angreifender Jäger in der gegnerischen Hüterzone sein. Trifft der Jäger auf ein Feld auf dem der Quaffel ist, hebt er ihn auf. Um zu werfen, bewegt sich der Jäger auf das gewünschte Feld und nach seiner Bewegung wirft er den Ball auf einen der 3 Torringe.
Begründung:	Jäger sind wichtig für das Spiel, da diese die Punkte machen, die das Team sammelt.
Abhängigkeiten:	Spielfeld, Quaffel, Torring, Quaffel Wurf
Priorität:	++
Akteure:	<Akteure>

ID:	FA21
Titel:	Treiber [Spielertyp]
Beschreibung:	Die Treiber sind ausgestattet mit Kloppern, welcher dazu dient, die Klatscher von den eigenen Teamkameraden fern zu halten und sie auf die gegnerischen Spieler zu lenken. Kommt ein Treiber auf das Feld eines Klatschers oder falls er bereits schon dort ist, so kann der Treiber in dieser Rundenphase den Klatscher kloppen. Er macht also einen Wurf mit dem Klatscher auf ein Zielfeld.
Begründung:	Treiber haben die Funktion die eigenen Spieler vor den Klatschern zu beschützen, so dass nicht alle Spieler des Teams vom Besen fliegen.
Abhängigkeiten:	Klatscher, Spielfeld, Spieler, Kloppen
Priorität:	++
Akteure:	<Akteure>

ID:	FA22
Titel:	Hüter [Spielertyp]
Beschreibung:	Hüter sind die Beschützer der jeweils eigenen Torringe. Hüter können allerdings nur einen Quaffel aufnehmen und dann wieder werfen, zum Beispiel zu einem eigenen Jäger, die Hüter können aber keine Tore erzielen, dies ist nur durch Jäger möglich.
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	Spielfeld, Torringe, Hüterzone, Quaffel, Quaffel werfen, Spieler
Priorität:	++
Akteure:	<Akteure>

ID:	FA23
Titel:	Sucher [Spielertyp]
Beschreibung:	Der Sucher hat in dem gesamten Spiel nur eine einzige Aufgabe. Seine Aufgabe ist es, den goldenen Schnatz zu suchen und ihn genau dann zu fangen, wenn sein Team durch die zusätzlichen 30 Punkte das Spiel gewinnt.
Begründung:	Der Sucher ist wichtig für das Spiel, da er die Möglichkeit hat, dass Spiel am ende noch zu drehen und somit dem Spiel einen zusätzlichen Kick zu verleihen, außerdem ist das Spiel sofort zu Ende, sobald der Schnatz gefangen wurde.
Abhängigkeiten:	Spiefeld, Spieler, goldener Schnatz
Priorität:	++
Akteure:	<Akteure>

ID:	FA24
Titel:	Fans]
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA25
Titel:	Elfen [Fantyp]
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA26
Titel:	Kobolde [Fantyp]
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA27
Titel:	Trolle [Fantyp]
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA28
Titel:	Niffler [Fantyp]
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA29
Titel:	Foul
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA30
Titel:	Flacken [Foul]
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA31
Titel:	Nachtarocken [Foul]
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA32
Titel:	Stutschen [Foul]
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA33
Titel:	Keilen [Foul]
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA34
Titel:	Schnaltzeln [Foul]
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA35
Titel:	Wurf mit dem Quaffel
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA36
Titel:	Quaffel Abfangen
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA37
Titel:	Torschuss
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA38
Titel:	Klatscher kloppen
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA39
Titel:	Partie-Konfiguration
Beschreibung:	<Beschreibung>
Begründung:	<Begründung>
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA40
Titel:	Quidditchteam-Konfiguration
Beschreibung:	<Beschreibung>
Begründung:	<Begründung>
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA41
Titel:	Netzwerkinterface
Beschreibung:	<Beschreibung>
Begründung:	<Begründung>
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA42
Titel:	Runde
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA43
Titel:	Ballphase
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA44
Titel:	Spielerphase
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA45
Titel:	Fanphase
Beschreibung:	<Beschreibung>
Begründung:	Diese Anforderung geht aus den im Lastenheft zu Verfügung gestellten Spielregeln für das Spiel <i>Fantastic Feasts</i> hervor.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA46
Titel:	Zufallsgenerator
Beschreibung:	<Beschreibung>
Begründung:	<Begründung>
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA47
Titel:	Spielende
Beschreibung:	<Beschreibung>
Begründung:	<Begründung>
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA48
Titel:	Überlängenbehnadlung
Beschreibung:	<Beschreibung>
Begründung:	<Begründung>
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA49
Titel:	Spiellogik
Beschreibung:	<Beschreibung>
Begründung:	<Begründung>
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA50
Titel:	Log-Datei
Beschreibung:	<Beschreibung>
Begründung:	<Begründung>
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA51
Titel:	<Titel>
Beschreibung:	<Beschreibung>
Begründung:	<Begründung>
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA52
Titel:	<Titel>
Beschreibung:	<Beschreibung>
Begründung:	<Begründung>
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA53
Titel:	<Titel>
Beschreibung:	<Beschreibung>
Begründung:	<Begründung>
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA54
Titel:	<Titel>
Beschreibung:	<Beschreibung>
Begründung:	<Begründung>
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA55
Titel:	<Titel>
Beschreibung:	<Beschreibung>
Begründung:	<Begründung>
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

4.3 Server spezifische funktionale Anforderungen

4.4 Client spezifische Funktionale Anforderungen

ID:	FA56
Titel:	Hauptmenü [Client-View]
Beschreibung:	Erste grafische Oberfläche die dem Nutzer angezeigt wird, wenn die Anwendung gestartet wurde.
Begründung:	Das Hauptmenü soll den Zentralen Punkt darstellen von dem aus alle Funktionen der Software zu erreichen.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA57
Titel:	Connect to Game [Client-View]
Beschreibung:	Grafische Oberfläche um sich mit einem Server auf dem ein Spiel bereit gestellt wird zu verbinden.
Begründung:	Der Nutzer muss die Möglichkeit haben sich komfortabel mit einem Server verbinden zu können.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA58
Titel:	End of Game [Client-View]
Beschreibung:	Grafische Oberfläche die, die Spieler sehen nachdem eine Partie zu ende ist.
Begründung:	Nach dem Ende einer Partie muss dem Nutzer mitgeteilt werden ob er gewonnen hat oder nicht. Optional ist hier auch platz für etwaige Statistiken über den Spielverlauf.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA59
Titel:	Import Team Config [Client-View]
Beschreibung:	Grafische Oberfläche zum importieren bzw. öffnen einer Team-Konfiguration für ein Spiel.
Begründung:	Es muss für den Benutzer eine einfachen Weg geben eine Team Konfiguration im Dateisystem zu suchen und an die Anwendung zu übergeben.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA60
Titel:	Game Play [Client-View]
Beschreibung:	Grafische Oberfläche die der Spieler sieht.
Begründung:	
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA61
Titel:	Hilfe [Client-View]
Beschreibung:	Grafische Oberfläche in der zum einen das Spielprinzip erklärt wird und zum anderen wie genau man die Clientsoftware bedient.
Begründung:	Um unerfahren Benutzer die Bedienung der Software zu erleichtern.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA62
Titel:	Beobachter [Client-View]
Beschreibung:	Grafische Oberfläche die ein Gast sieht.
Begründung:	<Begründung>
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA63
Titel:	Game Rendering Engine
Beschreibung:	Einheit, welche die Ansicht des Spielfelds grafisch aufbereitet.
Begründung:	
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA64
Titel:	Input Handler [Client]
Beschreibung:	Einheit die für Benutzereingaben verantwortlich ist.
Begründung:	Jede Benutzereingabe muss ausgewertet werden und na der Validierung eine Entscheidung getroffen werden was als Reaktion auf diese Eingaben passieren muss.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA65
Titel:	Input Validierung [Client]
Beschreibung:	Einheit die Benutzereingaben auf Korrektheit prüft.
Begründung:	Um etwaige falsche Benutzereingaben zu erkennen und den Nutzer darauf hin weisen zu können ist es von Nöten, dass alle Benutzereingaben geprüft werden.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA66
Titel:	Hotkey [Client]
Beschreibung:	Oft benötigte Funktionen auf Bestimmte (besondere) Testen (-Kombinationen) mappen.
Begründung:	Hotkeys sind optionale Features, die im Lastenheft aufgeführt sind und zu einer einfacheren Spielsteuerung betragen können.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	–
Akteure:	<Akteure>

ID:	FA67
Titel:	Pausieren [Client]
Beschreibung:	Das aktuelle Spiel pausieren.
Begründung:	Pausieren ist ein optionales Feature, das im Lastenheft aufgeführt ist und einem menschliche Spieler im Client zur Verfügung stehen sollte.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	–
Akteure:	<Akteure>

4.5 Quidditchteam-Editor spezifische funktionale Anforderungen

ID:	FA68
Titel:	Edit Team View
Beschreibung:	Grafische Oberfläche in der der Nutzer sein Team nach seinen Wünschen entsprechend anpassen kann.
Begründung:	Für das Bearbeiten des Teams soll dem Nutzer eine grafische Oberfläche bereit gestellt werden.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA69
Titel:	Open Team View
Beschreibung:	'Datei Öffnen' Dialog um die Team-Konfiguration von einem beliebigen Ort im Dateisystem zu öffnen
Begründung:	
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	FA70
Titel:	Save Team View
Beschreibung:	'Datei Speichern' Dialog um die Team-Konfiguration an einem beliebigen Ort im Dateisystem abzulegen.
Begründung:	
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	<Prio>
Akteure:	<Akteure>

4.6 Nicht funktionale Anforderungen

ID:	QA1
Titel:	Plattformunabhängigkeit
Beschreibung:	Der Client und der Team-Konfigurator soll auf mindestens einer gängigen Computerbetriebssystem-Plattform (z.B. Linux, Windows) uneingeschränkt benutzbar sein. Des weiteren soll die Serveranwendung und der KI-Client auf mindestens zwei gängigen Computerbetriebssystem-Plattformen (z.B. Linux, Windows) uneingeschränkt benutzbar sein.
Begründung:	Die Plattformunabhängigkeit ist im Lastenheft gefordert.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	++
Akteure:	<Akteure>

ID:	QA2
Titel:	Version-Controlling
Beschreibung:	Beim Verwalten des Quellcodes soll ein Git basiertes Version-Controlling Werkzeug (<i>GitHub</i> / <i>GitLab</i>) verwendet werden.
Begründung:	Durch das Verwenden eines Versionierungswerkzeuges wird das zusammenarbeiten unterschiedlicher Entwickler erleichtert, da das zusammenführen des Codes größtenteils automatisiert abläuft.
Abhängigkeiten:	-
Priorität:	++
Akteure:	-

ID:	QA3
Titel:	Continuous Integration
Beschreibung:	In die Version-Controlling Lösung mit Hilfe einer CI automatisch jeder gepushte Commit Unit-Tests und der Statischen Codeanalyse unterzogen werden. Zudem soll eine automatisierte Code Dokumentation angestoßen werden. Bei erfolgreichem Abschließen alle Test soll zum Schluss der aktuelle Stand deployed werden.
Begründung:	Die CI nimmt den Entwicklern Arbeit ab und kann dazu beitragen, dass Fehler frühzeitig erkannt und behoben werden können.
Abhängigkeiten:	-
Priorität:	-+
Akteure:	<Akteure>

ID:	QA4
Titel:	Statische Codeanalyse
Beschreibung:	Mit Hilfe des Tools 'SonarQube' bzw. 'SonarCloud' soll der gesamte Quellcode einer statischen Analyse unterzogen werden. Dabei darf die technische Codequalität von diesem Tool nicht schlechter als 'B' bewertet werden.
Begründung:	Quellcode mit einer hohen Codequalität ist weniger problematisch.
Abhängigkeiten:	-
Priorität:	+
Akteure:	<Akteure>

ID:	QA5
Titel:	Automatisierte Unit-Tests
Beschreibung:	50 Prozent des Quellcodes der einzelnen Komponenten soll durch automatische Unit-Test auf Korrektheit und Funktion geprüft werden
Begründung:	Da alle Komponenten möglichst fehlerfrei funktionieren müssen ist es unerlässlich die einzelnen Teil der Software ständig auf ihre Funktionalität zu prüfen, die ist nur effizient möglich wenn automatisiert Test durchgeführt werden, damit Fehler frühzeitig erkannt werden.
Abhängigkeiten:	-
Priorität:	-+
Akteure:	<Akteure>

ID:	QA6
Titel:	Docker Container
Beschreibung:	Um die Plattformunabhängigkeit zu gewährleisten soll sowohl die Server Komponente, als auch die KI-Komponenten mit Hilfe eines Docker Container veröffentlicht werden.
Begründung:	Docker Container bieten den Vorteil, dass die Software nicht auf jedem Zielsystem neu kompiliert werden muss sondern, sobald sie auf einem System in einem Docker-Container lauffähig gemacht wurde lässt sich dieser Container in der Regel auf diversen anderen Zielsystemen ausführen.
Abhängigkeiten:	-
Priorität:	+
Akteure:	<Akteure>

ID:	QA7
Titel:	Dokumentation
Beschreibung:	Alle Klassen und Methoden der Software müssen dokumentiert werden. Dabei sollen mindestens alle Übergabeparameter und Rückgabewerte genau spezifiziert werden. Zudem ist sind komplexe Algorithmen detailliert zu dokumentieren.
Begründung:	Gut dokumentierte Software vereinfacht die Fehlersuche, die Wartung und das hinzufügen von neuen Features.
Abhängigkeiten:	-
Priorität:	+
Akteure:	<Akteure>

ID:	QA8
Titel:	Benutzerhandbuch
Beschreibung:	Zu jeder Komponente des Projektes muss eine Benutzerhandbuch existieren, in welchem alle Features unmissverständlich erklärt sind, sodass ein neuer Benutzer auf Basis des Benutzerhandbuches die Software bedienen kann.
Begründung:	Das Benutzerhandbuch wird im Lastenheft gefordert.
Abhängigkeiten:	-
Priorität:	-+
Akteure:	<Akteure>

ID:	QA9
Titel:	Anwendungssprache
Beschreibung:	Das User-Interface der Anwendungen soll in deutscher Sprache gestaltet werden.
Begründung:	Die Anwendungssprache ist im Lastenheft vorgegeben.
Abhängigkeiten:	-
Priorität:	-+
Akteure:	<Akteure>

ID:	QA10
Titel:	Implementierungssprache
Beschreibung:	Die Implementierung der Anwendungen soll in englischer Sprache gehalten sein.
Begründung:	Die Implementierungssprache ist im Lastenheft vorgegeben.
Abhängigkeiten:	-
Priorität:	-+
Akteure:	<Akteure>

ID:	QA11
Titel:	Dokumentationssprache
Beschreibung:	Die Dokumentation der Software kann in englischer oder deutscher Sprache gestaltet sein.
Begründung:	Die Dokumentationssprache ist im Lastenheft vorgegeben.
Abhängigkeiten:	-
Priorität:	++
Akteure:	<Akteure>

ID:	QA12
Titel:	Programmiersprache
Beschreibung:	Die Software soll in einer der folgenden Programmiersprachen geschrieben sein: Java, C++, C# Die endgültig verwendete Sprache kann jedoch von Komponente zu Komponente variieren, muss aber mit dem Kunden abgesprochen werden.
Begründung:	Es soll eine Programmiersprache mit großem Funktionsumfang verwendet werden, welche von allen Teammitgliedern beherrscht wird.
Abhängigkeiten:	-
Priorität:	++
Akteure:	<Akteure>

ID:	QA13
Titel:	Format für Konfigurationsdateien
Beschreibung:	Alle Konfigurationsdateien müssen dem <i>JSON</i> Standard genügen. Des Weiteren sind alle vom Komitee festgelegten weiteren Standards einzuhalten.
Begründung:	Das Konfigurationsformat ist im Lastenheft vorgegeben.
Abhängigkeiten:	-
Priorität:	+
Akteure:	<Akteure>

ID:	QA14
Titel:	Netzwerkcommunication
Beschreibung:	Die Netzwerkcommunication zwischen Client und Server soll über sogenannte <i>Web-Socket-Sessions</i> realisiert werden, sodass Client und Server ortsunabhängig von einander betrieben werden können.
Begründung:	Die Netzwerkcommunication muss gewissen Standards genügen, damit Client- und Serveranwendungen von unterschiedlichen Entwicklerteams mit einander kompatibel sind und Client und Server ortsunabhängig von einander betrieben werden können.
Abhängigkeiten:	-
Priorität:	<Prio>
Akteure:	<Akteure>

ID:	QA15
Titel:	Log-Dateien
Beschreibung:	Log-Dateien anlegen um unter anderem den Spielverlauf zu Speichern und eventuelle Fehlfunktionen der Software fest zu halten.
Begründung:	Log-Dateien können unter anderen die Wartung der Software erleichtern und für zusätzliche Features, wie eine Statistik über den Spielverlauf verwendet werden.
Abhängigkeiten:	-
Priorität:	-
Akteure:	<Akteure>

ID:	QA16
Titel:	Funktionalität
Beschreibung:	Die Anwendungen müssen alle im Lastenheft als Minimalanforderungen aufgeführten Anforderungen erfüllen.
Begründung:	Um die Abnahmen zu bestehen müssen die Minimalanforderungen erfüllt werden.
Abhängigkeiten:	-
Priorität:	++
Akteure:	<Akteure>

ID:	QA17
Titel:	Zuverlässigkeit
Beschreibung:	Nur in maximal einer von 50 Sessions, darf sich eine Anwendung komplett aufhängen.
Begründung:	Durch zu häufiges Abstürzen der Software ist das Benutzererlebnis massiv beeinträchtigt.
Abhängigkeiten:	-
Priorität:	+
Akteure:	<Akteure>

ID:	QA18
Titel:	Robustheit
Beschreibung:	Die Anwendungen dürfen nicht aufgrund einer falschen Benutzereingabe abstürzen, sondern müssen den Benutzer auf seinen Fehler hinweisen.
Begründung:	Um das Benutzererlebnis nicht zu beeinträchtigen und keine Sicherheitslücken zu verursachen ist es notwendig, dass die Funktion der Software nicht durch fehlerhafte Benutzereingaben beeinträchtigt wird.
Abhängigkeiten:	-
Priorität:	++
Akteure:	<Akteure>

ID:	QA19
Titel:	Benutzbarkeit
Beschreibung:	Dem Endnutzer muss es möglich sein alle Komponenten des Projektes nur auf Basis des mitgelieferten Benutzerhandbuches und den Hilfeseiten die Software ohne Einschränkungen bedienen zu können.
Begründung:	Wenn es für die Endnutzer der Software zu kompliziert ist die Software zu benutzen, dann ist das Benutzererlebnis erheblich gestört und die Software wird nicht benutzt werden, da die Endbenutzer unzufrieden sind.
Abhängigkeiten:	-
Priorität:	+
Akteure:	<Akteure>

ID:	QA20
Titel:	Wartbarkeit
Beschreibung:	Die Software muss so aufgebaut sein, dass einzelne Teilstücke bei Bedarf ohne Umbauten der übrigen Software ersetzbar sind.
Begründung:	Im Falle einer Fehlfunktion in einem Teilstück der Software muss diese einfach austauschbar sein um den Fehler schnellst möglich beheben zu können. Zudem sollte das Hinzufügen weiterer Features möglich sein um das Produkt stetig weiter entwickeln zu können.
Abhängigkeiten:	-
Priorität:	-+
Akteure:	<Akteure>

ID:	QA21
Titel:	Effizienz
Beschreibung:	Die Software sollte Ressourcen schonend arbeiten.
Begründung:	Eine ressourcenschonende Anwendung ist auch auf älteren Zielsystemen problemlos nutzbar.
Abhängigkeiten:	-
Priorität:	-
Akteure:	<Akteure>

ID:	QA22
Titel:	Kurze Ladezeiten
Beschreibung:	Systembedingte Ladezeiten der Software dürfen auf einem aktuellen Mittelklasse Computer fünf Sekunden pro geladenem Teil überschreiten.
Begründung:	Bei längeren Ladezeiten ist das Benutzererlebnis massiv beeinträchtigt.
Abhängigkeiten:	-
Priorität:	-+
Akteure:	-

ID:	QA23
Titel:	<Titel>
Beschreibung:	<Beschreibung>
Begründung:	<Begründung>
Abhängigkeiten:	-
Priorität:	<Prio>
Akteure:	<Akteure>