



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
1.1	Einleitung	3
1.2	Motivation	3
1.3	Vision	3
1.4	Projektkontext	4
2	Fachwissen	5
3	Domänenmodell	21
4	Anforderungsdefinition	22
4.1	Akteure	22
4.2	Allgemeine funktionale Anforderungen	25
4.3	Server spezifische funktionale Anforderungen	51
4.4	Client spezifische Funktionale Anforderungen	53
4.5	Quidditchteam-Editor spezifische funktionale Anforderungen	57
4.6	Nicht funktionale Anforderungen	59

1 Kontextanalyse

1.1 Einleitung

Bei dem Projekt handelt es sich um die Konzeption und Implementierung eines online Multiplayer-Spiels aus der Welt von Harry Potter – genauer: *Fantastic Feasts*. Es ist eine rundenbasierte Form des bekannten Spiels Quidditch.

Im Mittelpunkt des Projekts steht das Erlernen von Fähigkeiten im Umgang mit einem größeren Softwareprojekt. Es werden keine kommerziellen Ziele verfolgt.

Der Auftraggeber – im weitesten Sinne die Universität Ulm – verfolgt das Ziel, den Studenten Fähigkeiten zu vermitteln und sie anschließend nach genau definierten Maßstäben zu bewerten. Zu diesen Fähigkeiten gehört folgendes:

Zunächst einmal stehen Planen, Formulieren von Anforderungen und Modellierung von Software an. Es folgt die Auseinandersetzung mit verschiedenen Plattformen und Technologien auf die für die Implementierung zurückgegriffen werden soll. Gleichzeitig wird das Ziel verfolgt, übergeordnete Fähigkeiten zur Qualitätssicherung, zur Versionenverwaltung oder zu agilen Entwicklungsprozessen im Team zu erwerben. Erst dann kommen praktische Programmierfähigkeiten zum tragen. Auch hier ist es das Ziel, diese auszubauen.

Die Studierenden – in diesem Fall 6 Studenten der Informationssystemtechnik – verfolgen das Ziel, das Projekt nach den Anforderungen im Lastenheft erfolgreich umzusetzen und die Abnahmeprüfung zu bestehen.

1.2 Motivation

Die Motivation für das Projekt lässt sich – wie bei den Zielen – in die der Universität und die der Studenten aufteilen.

Die Universität gibt die Inhalte vor. Diese sollen von den Studenten bestmöglich erlernt werden, da es Teil ihrer Ausbildung darstellt. Die Universität will somit ihrem Auftrag der Lehre gerecht werden.

Für die Studenten ist die Motivation der Erwerb und Ausbau der oben genannten Fähigkeiten und im weitesten Sinne eine erfolgreiche Ausbildung in ihrem Fach. Zusätzlich soll ein Spiel entwickelt werden, das funktioniert und Spaß macht.

1.3 Vision

Das fertige System soll folgendermaßen aufgebaut sein: Einer Client-Server-Architektur folgend kommunizieren ein oder mehrere Clients mit dem Server, auf dem die Spiellogik

läuft. Die Spieler haben client-seitig eine ansprechende und lebendige GUI, über die sie *Fantastic Feasts* spielen, eine Partie als Zuschauer verfolgen, Charaktere und Ausrüstung zu Teams mitsamt Farben und Logo zusammen stellen können und die Möglichkeit haben, Partien zu konfigurieren. Begleitet wird die visuelle Darstellung von Soundeffekten und einer thematisch Ansprechenden Spielmusik.

Im durch und durch taktischen Spiel mit zwei sich gegenüberstehenden Quidditch-Teams können die Spieler Runde für Runde Spielfiguren auf Besen über das Spielfeld jagen lassen, Punkte erzielen, den Gegner sabotieren und Publikumseffekte zu ihrem Vorteil einsetzen. Doch selbst bei noch so guter Taktik kann ihnen der Zufall einen Strich durch die Rechnung machen, da nicht immer alles so eintritt, wie es sich die einzelnen Spieler vielleicht erhofft haben.

Was die Spielmodi betrifft, ist das Kern-Szenario das Multiplayer-Spiel. Hier entsteht durch den Wettstreit zweier Spieler die größte Spannung. Doch um das Spiel auch alleine spielbar zu machen, existiert ein Singleplayer-Modus. Eine ausgefeilte KI mit voraussichtlich mehreren Schwierigkeitsstufen stellt für Einzelspieler eine spannende Herausforderung dar. Wer sich noch auf den großen Wettkampf vorbereitet, die Taktik andere Spieler erlernen will oder einfach Spaß am Zuschauen hat, kann sich im Zuschauer-Modus in andere Multiplayer-Partien einklinken. Damit bleibt einer breiten Zielgruppe an Spielern kaum etwas zu wünschen übrig.

1.4 Projektkontext

Auftraggeber des Projektes ist die Servicegruppe Informatik der Universität Ulm, die das Modul Softwaregrundprojekt veranstaltet. Der Tutor Stefanos Mytilineos vertritt den Auftraggeber während der Projektlaufzeit, in höherer Instanz ist Florian Ege verantwortlich. Als weiterer Stakeholder tritt das Team auf, das das Projekt letztendlich entwickelt. Es besteht aus sechs Studenten der Informationssystemtechnik: Tarik Enderes, Tim Luchterhand, Jonas Merkle, Paul Nykiel, Björn Petersen und Michael von Hohnhorst. Diese bearbeiten das Projekt nicht in Vollzeit, da sie parallel den weiteren Verlauf ihres Studiums verfolgen.

Indirekt am Entwicklungsprozess beteiligt ist das Standardisierungskomitee, in das auch aus diesem Team ein Vertreter geschickt wird. Dort werden alle nötigen Protokolle und Schnittstellen definiert, die bei der Entwicklung von *Fantastic Feasts* benötigt werden. Weitere Stakeholder, die jedoch erst später in Erscheinung treten, sind andere Teams, die gegebenenfalls auf einen Komponenten dieses Projektes angewiesen sind. Sie werden auf einer bevorstehenden Messe zu potenziellen Kunden. In ihrem Interesse liegt eine saubere Implementierung bei gleichzeitig guter Dokumentation des Komponenten. Zu guter letzt muss dieses Projekt sowie die Einzelleistung eines jeden Team-Mitgliedes die Prüfer in der Abnahmeprüfung überzeugen – sie stellen so gesehen die wichtigsten

Kunden dar.

Da bei der Implementierung auf den agilen Entwicklungsprozess Scrum zurückgegriffen wird, sollen auch hier die Rollen kurz benannt werden. In dieser Phase übernimmt der oben genannte Tutor die Rolle des Product Owners. Der ScrumMaster wird innerhalb des oben genannten Teams ernannt.

Folgeprojekte von *Fantastic Feasts* sind derzeit nicht vorgesehen. Denkbar wäre jedoch ein Publishing des Spiels mit ständig laufendem Server und beliebig vielen Server-Instanzen – so könnte *Fantastic Feasts* weltweit von zahlreichen Spielern gespielt werden.

2 Fachwissen

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 Nutzer 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	Spieler
Beschreibung	Ein Nutzer, der das Computerspiel „Fantastic Feasts“ spielt.
Ist-ein	Nutzer
Kann-sein	-
Aspekt	Zur Beschreibung des Programmverlaufs
Bemerkung	-
Beispiel	-

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, 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 „KI“ ein.
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, Toring, Hüterzonenzelle
Aspekt	Mögliche Standorte der Spielobjekte
Bemerkung	Wird nicht Feld genannt, um Verwechslungen mit dem Spielfeld zu vermeiden.
Beispiel	-

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

Begriff	Hüterzone
Beschreibung	Die Bereiche am linken und rechten Rand des Spielfeldes, in denen sich die Torringe befinden.
Ist-ein	-
Kann-sein	-
Aspekt	Summe aller Hüterzonenzellen und Torringe
Bemerkung	-
Beispiel	-

Begriff	Torring
Beschreibung	Die Zellen, in die beide Teams den Quaffel bewegen wollen. Es wird zwischen eigenen und gegnerischen Torringen unterschieden.
Ist-ein	Zelle
Kann-sein	Eigener Torring, Gegnerischer Torring
Aspekt	Hauptquelle von Punkten
Bemerkung	-
Beispiel	-

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

Begriff	Hüterzonenzelle
Beschreibung	Eine Zelle in einer Hüterzone.
Ist-ein	Zelle
Kann-sein	-
Aspekt	Limitierendes Element für das Abliefern der Quaffel
Bemerkung	-
Beispiel	-

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

Begriff	Ball
Beschreibung	Ein Spielobjekt, das nicht direkt, nur indirekt von einem Spieler beeinflusst werden kann.
Ist-ein	Spielobjekt
Kann-sein	Quaffel, Klatscher, Schnatz
Aspekt	Festpunkte zur Steuerung des Spielverlaufs
Bemerkung	-
Beispiel	-

Begriff	Spielfigur
Beschreibung	Ein Spielobjekt, das von einem Spieler direkt gesteuert wird. Jede Spielfigur hat einen Namen, einen Besenrang, ein Geschlecht, ein Team und eine Rolle. Man unterscheidet außerdem zwischen eigenen und gegnerischen Spielfiguren.
Ist-ein	Spielobjekt
Kann-sein	Hüter, Sucher, Jäger, Treiber
Aspekt	Mitglieder eines Teams
Bemerkung	Spieler im Lastenheft
Beispiel	Name: Luke Skywalker, Rolle: Hüter des Teams „Jedi“, Geschlecht: männlich, Besenrang: 5

Begriff	Quaffel
Beschreibung	Passives Objekt, mit dem Jäger und Hüter interagieren können und von ihnen nach Möglichkeit in einen gegnerischen Tarring befördert werden soll.
Ist-ein	Ball
Kann-sein	-
Aspekt	Zentrales Spielobjekt
Bemerkung	-
Beispiel	-

Begriff	Klatscher
Beschreibung	Ball, der sich von selbst auf Spielfiguren zubewegt, die keine Treiber sind und diese betäuben kann. Kann von Treibern geschlagen werden, was technisch gesehen einem Wurf entspricht.
Ist-ein	Ball
Kann-sein	-
Aspekt	Zusätzliches taktisches Spielelement
Bemerkung	-
Beispiel	-

Begriff	Schnatz
Beschreibung	Beschreibung: Ball, der sich selbstständig bewegt und von den Suchern gejagt wird. Wird er von einem Sucher gefangen, bekommt dessen Team 30 Punkte und die Partie ist zu Ende.
Ist-ein	Ball
Kann-sein	-
Aspekt	Definiert Spielende
Bemerkung	-
Beispiel	-

Begriff	Partie
Beschreibung	Ein einzelnes Spiel. Beginnt beim Platzieren der Figuren und endet mit dem Bestimmen des Gewinners.
Ist-ein	-
Kann-sein	-
Aspekt	Beschreibung des Spielablaufs
Bemerkung	-
Beispiel	Spieler VodkaVodka98 spielt gegen Spieler LongEiland

Begriff	Hüter
Beschreibung	Spielfigur, deren Aufgabe es ist, den Quaffel von den eigenen Ziel-feldern fernzuhalten.
Ist-ein	Spielfigur
Kann-sein	Eigener Hüter, Gegnerischer Hüter
Aspekt	Letzte Verteidigungslinie
Bemerkung	-
Beispiel	Siehe „Spielfigur“

Begriff	Sucher
Beschreibung	Spielfigur, die den Schnatz jagt.
Ist-ein	Spielfigur
Kann-sein	Eigener Sucher, Gegnerischer Sucher
Aspekt	Beendet die Partie
Bemerkung	-
Beispiel	Darth Vader, gegnerischer Sucher, Besenrang 2

Begriff	Jäger
Beschreibung	Spielfigur, die den Quaffel in ein einen gegnerischen Tarring beför- dern soll.
Ist-ein	Spielfigur
Kann-sein	Eigener Jäger, Gegnerischer Jäger
Aspekt	Holt Punkte für das eigene Team
Bemerkung	-
Beispiel	Han Solo, eigener Jäger, Besenrang 3

Begriff	Treiber
Beschreibung	Spielfigur, mit der der Spieler eigene Spielfiguren vor Klatschern schützt und gegnerische damit abschießen kann.
Ist-ein	Spielfigur
Kann-sein	Eigener Treiber, Gegnerischer Treiber
Aspekt	Interagiert mit Klatschern
Bemerkung	-
Beispiel	Boba Fett, gegnerischer Treiber, Besenrang 4

Begriff	Geschlecht
Beschreibung	Jede Spielfigur ist entweder männlich oder weiblich.
Ist-ein	-
Kann-sein	-
Aspekt	Team-Editierung
Bemerkung	-
Beispiel	-

Begriff	Team
Beschreibung	Die Menge aller Spielfiguren auf dem Spielfeld, die von einem einzigen Spieler kontrolliert wird. Ein Team hat einen Namen, ein Motto, ein Logo, eine Hauptfarbe und eine Ersatzfarbe.
Ist-ein	eigenes Team, gegnerisches Team
Kann-sein	-
Aspekt	Beschreibung einer Partie
Bemerkung	-
Beispiel	Galaktisches Imperium, Motto: „Unbegrenzte MAAACHT!“, [Todesstern als Logo], Hauptfarbe: Schwarz, Ersatzfarbe: Rot

Begriff	Punkte
Beschreibung	Der Spieler mit mehr Punkten am Ende der Partie gewinnt. Werden durch das Platzieren des Quaffel in einem gegnerischen Torring oder das Finden des Schnatzes erhalten.
Ist-ein	-
Kann-sein	-
Aspekt	Bestimmung des Gewinners
Bemerkung	-
Beispiel	SchnapsNase hat 20 Punkte.

Begriff	Besetzen
Beschreibung	Eine Spielfigur besetzt das Feld, auf dem sie sich befindet.
Ist-ein	-
Kann-sein	-
Aspekt	Beschreibung des Spielgeschehens
Bemerkung	Zwei Spielfiguren können sich nicht auf derselben Zelle befinden.
Beispiel	Chewbacca besetzt Zelle 5:3.

Begriff	Besenrang
Beschreibung	Jede Spielfigur hat einen Besenrang von 1 bis 5, der die Wahrscheinlichkeit bestimmt, dass sie nach einer Bewegung erneut eine Bewegung um ein Feld ausführen darf. Besenrang 1 ist der beste.
Ist-ein	-
Kann-sein	-
Aspekt	Unterscheidet Qualität der Spielfiguren.
Bemerkung	Ersetzt die verschiedenen „Besen“ aus dem Lastenheft mit einer Skala von 1 bis 5 zur besseren Übersicht.
Beispiel	Yoda hat Besenrang 1.

Begriff	Aktion
Beschreibung	Jede durch einen Spieler hervorgerufene Änderung der Spielsituation.
Ist-ein	-
Kann-sein	Ziehen, Schießen, Schlagen, Einmischung, Übernahme
Aspekt	Weiterführung der Partie
Bemerkung	-
Beispiel	Obi-Wan Kenobi zieht von Zelle 8:7 auf Zelle 9:7.

Begriff	Ziehen
Beschreibung	Die Bewegung einer Spielfigur von einer Zelle auf eine andere durch direkten Befehl des Spielers.
Ist-ein	Aktion
Kann-sein	-
Aspekt	Beschreibung des Spielverlaufs
Bemerkung	Bezieht sich nicht auf erzwungene Bewegungen einer Spielfigur.
Beispiel	Obiwan Kenobi zieht von Zelle 8:7 auf Zelle 9:7.

Begriff	Befördern
Beschreibung	Bewegen des Quaffel mittels einer Spielfigur.
Ist-ein	-
Kann-sein	-
Aspekt	Bewegen des Quaffel, allgemeiner Begriff
Bemerkung	Keine Aktion, da eventuell eine passive Folge, z.B. durch Ziehen
Beispiel	-

Begriff	Schießen
Beschreibung	Die Bewegung des Quaffel durch einen Hüter oder Jäger auf eine andere, entfernte Zelle ohne Bewegung der Spielfigur.
Ist-ein	Aktion
Kann-sein	-
Aspekt	Bewegung des Quaffel um mehrere Felder
Bemerkung	„Werfen“ im Lastenheft. Analog zum Schussvektor benannt.
Beispiel	Mace Windu schießt den Quaffel auf Zelle 10:4.

Begriff	Schlagen
Beschreibung	Die erzwungene Bewegung eines Klatschers durch einen Treiber.
Ist-ein	Aktion
Kann-sein	-
Aspekt	Interaktion mit Klatschern
Bemerkung	„Kloppen“ im Lastenheft
Beispiel	R2-D2 schlägt einen Klatscher auf Zelle 5:10.

Begriff	Einmischung
Beschreibung	Hilfsfähigkeiten, die nicht von Spielobjekten ausgehen. Werden von einem Spieler gesteuert. Bei jeder Benutzung besteht eine Chance, dass die verwendete Einmischung bis zum Ende der Partie für den jeweiligen Spieler vom Schiedsrichter deaktiviert werden.
Ist-ein	Aktion
Kann-sein	Teleportation, Fernangriff, Impuls, Schnatzjagd
Aspekt	Zusätzliche taktische Element
Bemerkung	Ersetzt die „Fans“ aus dem Lastenheft.
Beispiel	Lando Calrissian wird auf Zelle 6:6 teleportiert.

Begriff	Teleportation
Beschreibung	Einmischung, die eine Spielfigur auf eine zufällige Zelle teleportiert.
Ist-ein	Einmischung
Kann-sein	-
Aspekt	-
Bemerkung	Ersetzt „Elfen“ aus Lastenheft
Beispiel	Siehe „Einmischungen“

Begriff	Fernangriff
Beschreibung	Trifft eine gegnerische Spielfigur. Ziel verliert gegebenenfalls den Quaffel und wird auf eine zufällige benachbarte, freie Zelle bewegt.
Ist-ein	Einmischung
Kann-sein	-
Aspekt	-
Bemerkung	Statt „Kobolde“ im Lastenheft
Beispiel	Jango Fett wird von Fernangriff auf Zelle 5:6 gestoßen

Begriff	Impuls
Beschreibung	Wenn eine Spielfigur den Quaffel hält, wird sie bei Benutzung verloren.
Ist-ein	Einmischung
Kann-sein	-
Aspekt	-
Bemerkung	Statt „Trolle“ im Lastenheft
Beispiel	C-3PO verliert wegen eines Impuls den Quaffel.

Begriff	Schnatzstoß
Beschreibung	Bewegt den Schnatz in eine zufällige Richtung um ein Feld
Ist-ein	Einmischung
Kann-sein	-
Aspekt	-
Bemerkung	„Schnatzschnappen“ im Lastenheft
Beispiel	Ein Schnatzstoß treibt den Schnatz auf Zelle 4:12.

Begriff	Entfernung
Beschreibung	Eine Entfernung zwischen zwei Zellen ist die minimale Anzahl von Zügen, in denen eine Spielfigur von der einen auf die andere ziehen kann.
Ist-ein	-
Kann-sein	-
Aspekt	Spielfeldgeometrie
Bemerkung	-
Beispiel	-

Begriff	Schussvektor
Beschreibung	Pfeil vom Mittelpunkt einer Zelle zum Mittelpunkt einer anderen.
Ist-ein	-
Kann-sein	Torschussvektor
Aspekt	Spielfeldgeometrie
Bemerkung	-
Beispiel	-

Begriff	Torschussvektor
Beschreibung	Schussvektor zu einem Schuss, der möglicherweise in einem Torschuss resultiert. (Ein Schussvektor, der die linke oder rechte Seite eines Torrings schneidet.)
Ist-ein	Schussvektor
Kann-sein	-
Aspekt	Punkte sammeln
Bemerkung	-
Beispiel	-

Begriff	Torschuss
Beschreibung	Ein Jäger schießt den Quaffel in einen Torring und holt damit Punkte für sei Team.
Ist-ein	-
Kann-sein	-
Aspekt	Punkte sammeln
Bemerkung	Nur erfolgreiche Schüsse auf das Tor werden als Torschüsse bezeichnet.
Beispiel	Darth Sidious schießt den Quaffel in ein eigenes Tor.

Begriff	Zugphase
Beschreibung	Phase, in der eine Spielfigur Aktionen durchführt. Beginnt, sobald der Spieler die Möglichkeit hat, die jeweilige Spielfigur zu steuern und endet, sobald er ihr den letzten Befehl für diesen Zug gegeben hat. Ein Zug enthält mehrere Zugphasen.
Ist-ein	-
Kann-sein	-
Aspekt	Zeiteinteilung
Bemerkung	
Beispiel	Leia Organa ist dran.

Begriff	Zug
Beschreibung	Von der ersten Aktion eines Spielers bis zur ersten Aktion des Gegners.
Ist-ein	-
Kann-sein	-
Aspekt	Zeiteinteilung
Bemerkung	Nicht die Zugphase einer Spielfigur
Beispiel	Bierdurst69 ist am Zug.

Begriff	Endphase
Beschreibung	Letzter Teil eines Zuges. Der Spieler kann darin Einmischungen vornehmen.
Ist-ein	-
Kann-sein	-
Aspekt	Zeiteinteilung
Bemerkung	-
Beispiel	-

Begriff	Verlieren
Beschreibung	Der Quaffel wird auf eine zufällige Zelle bewegt, die an die Zelle angrenzt, auf der sich die Spielfigur, die bis jetzt in Ballbesitz war, befindet.
Ist-ein	-
Kann-sein	-
Aspekt	Spielablauf
Bemerkung	„Vertändeln“ im Lastenheft
Beispiel	Jar Jar verliert den Quaffel.

Begriff	Halten
Beschreibung	Ein Jäger oder Hüter kann den Quaffel halten. Ist das der Fall, bewegt sich der Quaffel auf die Zelle, auf die die Spielfigur zieht.
Ist-ein	-
Kann-sein	-
Aspekt	Beschreibung des Spielgeschehens
Bemerkung	-
Beispiel	-

Begriff	Übernahme
Beschreibung	Ein Jäger neben einer gegnerischen Spielfigur, die den Quaffel hält, kann diesen mit einer bestimmten Wahrscheinlichkeit übernehmen und hält ihn anschließend selbst.
Ist-ein	Aktion
Kann-sein	-
Aspekt	Aggressives Spielmanöver
Bemerkung	-
Beispiel	Darth Vader übernimmt den Quaffel von Anakin Skywalker.

Begriff	Betäubt
Beschreibung	Eine betäubte Spielfigur kann in seiner nächsten Rundenphase keine Aktion durchführen.
Ist-ein	-
Kann-sein	-
Aspekt	Wirkung der Klatscher
Bemerkung	„Ausgeknockt“ im Lastenheft
Beispiel	Jango Fett ist betäubt.

Begriff	Foul
Beschreibung	Handlung, wegen der eine Spielfigur vorübergehend vom Spielfeld entfernt werden kann.
Ist-ein	-
Kann-sein	Torring Blockieren, Stürmen, Großoffensive, Rammen, Schnatz Blockieren
Aspekt	Taktische Elemente
Bemerkung	-
Beispiel	Qui-Gon Jinn blockiert den Schnatz.

Begriff	Torring Blockieren
Beschreibung	Eine eigene Spielfigur besetzt einen eigenen Torring, was eine Torsschuss verhindert.
Ist-ein	Foul
Kann-sein	-
Aspekt	Taktik
Bemerkung	„Flackern“ im Pflichtenheft
Beispiel	-

Begriff	Stürmen
Beschreibung	Ein Jäger, der den Quaffel hält, zieht auf einen gegnerischen Topping, was das Abliefern garantiert.
Ist-ein	Foul
Kann-sein	-
Aspekt	Taktik
Bemerkung	„Nachtarocken“ im Lastenheft
Beispiel	Han Solo stürmt mittleren gegnerischen Topping.

Begriff	Großoffensive
Beschreibung	Ein eigener Jäger betritt eine gegnerische Hüterzonenzelle während ein anderer eigener Jäger sich auf einer anderen befindet.
Ist-ein	Foul
Kann-sein	-
Aspekt	Taktik
Bemerkung	„Stutschen“ im Lastenheft
Beispiel	Lando Calrissia schließt sich Chewbacca in einer Großoffensive an.

Begriff	Rammen
Beschreibung	Eine eigene Spielfigur zieht auf eine Zelle, die von einer gegnerischen Spielfigur besetzt wird. Dadurch wird die gegnerische Spielfigur auf eine benachbarte Zelle bewegt und verliert den Quaffel
Ist-ein	Foul
Kann-sein	-
Aspekt	Taktik
Bemerkung	„Keilen“ im Lastenheft
Beispiel	Boba Fett rammt Jar Jar.

Begriff	Schnatz blockieren
Beschreibung	Eine Spielfigur, die kein Sucher ist, besetzt die Zelle, auf der sich der Schnatz befindet.
Ist-ein	Foul
Kann-sein	-
Aspekt	Taktik
Bemerkung	„Schnatzeln“ im Lastenheft
Beispiel	Darth Maul blockiert den Schnatz.

Begriff	Schiedsrichter
Beschreibung	Entfernt mit bestimmter Wahrscheinlichkeit eine Spielfigur, die ein Foul ausführt vom Spielfeld bis ein Torschuss erfolgt und deaktiviert permanent eine Einmischung für den Rest der Partie.
Ist-ein	-
Kann-sein	-
Aspekt	Taktik
Bemerkung	„Schiedsrichter“ im Lastenheft
Beispiel	Sheev Palpatine wurde vom Schiedsrichter vom Spielfeld entfernt.

Begriff	Disqualifikation
Beschreibung	Tritt ein wenn drei oder mehr Spielfiguren eines Spielers in der selben Rundenphase durch den Schiedsrichter aus dem Spiel entfernt wurden. Führt zur Niederlage des Spielers.
Ist-ein	-
Kann-sein	-
Aspekt	Erhöhtes Risiko
Bemerkung	-
Beispiel	CubaLibre wurde disqualifiziert. CaptainCola gewinnt die Partie.



4 Anforderungsdefinition

4.1 Akteure

ID:	AKT1
Titel:	Nutzer
Beschreibung:	Menschlicher Nutzer, der eine Anwendungen bedient.
Rolle:	Ein Mensch, der entweder als Spieler aktiv an einem Spiel teilnimmt, als Gast passiv einem Spiel zusieht oder den Quidditchteam-Editor benutzt, um ein Team zu erstellen oder anzupassen.

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 erweiterten Rechten, 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:	Entwickler
Beschreibung:	Person, die an der Entwicklung der Anwendung beteiligt ist.
Rolle:	Diese Software wird von einem Team aus 6 Entwicklern entwickelt und umgesetzt.

ID:	AKT6
Titel:	KI
Beschreibung:	Vom Computer gesteuerter Spieler.
Rolle:	Spieler, dessen Entscheidungen und Züge auf einem Computer von der KI-Client Software getroffen werden.

ID:	AKT7
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:	AKT8
Titel:	Client
Beschreibung:	Stellt die grafische Oberfläche zur Verfügung, über die ein Nutzer das Spiel „Fantastic Feasts“ spielen kann.
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:	AKT9
Titel:	KI-Client
Beschreibung:	Anwendung, die die KI steuert.
Rolle:	Der KI-Client ist die Anwendung, die eine KI steuert. Der Nutzer kann von Außen keinen Einfluss auf die Züge und Entscheidungen des KI-Clients nehmen.

ID:	AKT10
Titel:	Server
Beschreibung:	Zentraler Computer mit spezieller Software, zu dem alle am Spiel beteiligten Clients eine 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 etwa die Spielfeldkonfiguration oder die Züge des Gegners, bereit.

ID:	AKT11
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 konfigurieren.

ID:	AKT12
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:	AKT13
Titel:	Schiedsrichter
Beschreibung:	Ahndet Fouls.
Rolle:	Der Schiedsrichter wird durch eine Software repräsentiert, die mit Hilfe eines Zufallsgenerators entscheidet, ob ein Foul geahndet wird oder nicht.

ID:	AKT14
Titel:	Spielfigur
Beschreibung:	Spielfigur auf dem Spielfeld
Rolle:	Eine Spielfigur ist ein Teil eines Teams und kann einen der folgenden Typen besitzen: Sucher, Hüter, Treiber oder Jäger.

ID:	AKT15
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 hat eine Ovale Form, die in ein Raster von 17x13 quadratischen Feldern eingepasst ist. Auf diesem Feld finden alle Spielhandlungen statt, die während dem Spiel getätigt werden können. Einen Teil des Spielfeldes bilden der Mittelkreis im Zentrum und die Hüterzonen, jeweils an den gegenüberliegenden Enden.
Begründung:	Das Spielfeld ist die zentrale Komponente des Spiels, da sich hier während einer Partie sämtliche Abläufe abspielen.
Abhängigkeiten:	-
Priorität:	++
Akteure:	Team, Spiel, Spieler, KI, Gast, Schiedsrichter

ID:	FA2
Titel:	Mittelkreis
Beschreibung:	Der Mittelkreis ist ein Bereich auf dem Quidditch-Spielfeld, der in der Mitte angeordnet ist und aus 3x3 quadratischen Kacheln besteht. Im Mittelkreis befindet sich das Mittelfeld, die zentrale Kachel des Spielfeldes, die durch ein <i>M</i> gekennzeichnet ist.
Begründung:	Der Mittelkreis markiert das Zentrum des Spielfeldes, in dem das Spiel gestartet wird. Zu Beginn dürfen sich hier keine Spielfiguren befinden.
Abhängigkeiten:	Spielfeld
Priorität:	++
Akteure:	Spiel

ID:	FA3
Titel:	Mittelfeld
Beschreibung:	Das Mittelfeld stellt den mittleren Punkt des Mittelkreises dar, der sich im Zentrum des Quidditch-Spielfeldes befindet.
Begründung:	Auf dem Mittelfeld werden initial die Bälle des Spiels platziert.
Abhängigkeiten:	Spielfeld, Mittelkreis
Priorität:	++
Akteure:	Spiel

ID:	FA4
Titel:	Hüterzone
Beschreibung:	Die Hüterzonen sind an den jeweils gegenüberliegenden Seiten des Quidditch-Spielfeldes. Sie beinhalten jeweils 3 Torringe und sind somit die Zonen, in denen die Teams punkten können. Die Hüterzonen sind ovalförmig und bestehen aus 11x5 Kacheln. Außerdem beinhaltet diese Zone die Torringe, durch welche die Teams Punkte machen können.
Begründung:	Hier finden die wichtigsten Aktionen des Spiels statt: Jäger versuchen, den Quaffel durch die gegnerischen Torringe zu werfen, um Punkte zu erzielen, während der Hüter genau das zu verhindern sucht.
Abhängigkeiten:	Spielfeld
Priorität:	++
Akteure:	Spiel

ID:	FA5
Titel:	Zelle
Beschreibung:	Die Zelle ist die kleinste Einheit des Spielfeldes, auf ihr darf sich immer nur maximal ein Spieler befinden. Bälle können sich jedoch eine Zelle mit einem anderen Ball und / oder einer Spielfigur teilen.
Begründung:	Das gesamte Spielfeld ist aus Zellen aufgebaut. Sie bestimmen, wie sich Spielobjekte bewegen können.
Abhängigkeiten:	Spielfeld
Priorität:	++
Akteure:	Spiel

ID:	FA6
Titel:	Torring
Beschreibung:	Jeweils 3 Torringe sind in einer Hüterzone angebracht. Die Teams können Punkte erzielen, indem sie den Quaffel durch einen generischen Torring werfen.
Begründung:	Die Torringe dienen den Teams als Hauptquelle von Punkten und stellen daher ein zentrales Spielelement dar.
Abhängigkeiten:	Hüterzone
Priorität:	++
Akteure:	Spiel

ID:	FA7
Titel:	Schussvektor
Beschreibung:	Ein Schussvektor beschreibt, wie ein Ball von einer Spielfigur über das Spielfeld bewegt werden kann. Der Schussvektor zeigt dabei vom Mittelpunkt des Rasterfeldes, von dem geworfen wird, auf das Feld, auf das geworfen wird. 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.
Begründung:	Der Schussvektor ist wichtig, da über den Vektor entschieden wird, ob ein Wurf erfolgreich ist oder nicht und auch je nachdem wie der Vektor ausfällt kann ein Tor erzielt werden oder nicht.
Abhängigkeiten:	Hüterzone
Priorität:	++
Akteure:	Spiel

ID:	FA8
Titel:	Torschuss
Beschreibung:	Für einen Torschuss muss ein Spieler einen Schussvektor auf ein Toringfeld auswählen. Das Toringfeld hat zwei rot markierte Kanten, welche jeweils die Toröffnungen sind. geht der Vektor durch eine dieser beiden Öffnungen, so erzielt der Spieler ein Tor. geht der Vektor durch eine der beiden anderen Kanten, so wird kein Tor erzielt. In beiden Fällen liegt der Quaffel am Ende auf dem Toringfeld, was zur Folge hat, dass nach der Rundenphase der Quaffel in den Besitz des abwehrenden Hüters geht.
Begründung:	Diese Anforderung ist wichtig, da die Teams Punkte erzielen sollen und dazu müssen die einen Torschuss absolvieren.
Abhängigkeiten:	Spiel, Quaffel, Quaffel-Werfen, Jäger, Hüter
Priorität:	++
Akteure:	Spiel

ID:	FA9
Titel:	Punkte
Beschreibung:	In dem Spiel <i>Fantastic Feasts</i> gibt es zwei verschiedene Möglichkeiten an Punkte zu kommen. Die eine Möglichkeit ist, dass die Jäger und der Hüter versuchen den Quaffel in einen der drei gegnerischen Torringe zu werfen, bzw. verhindern dass das gegnerische Team Tore erzielt. Schafft es jedoch ein Team den Quaffel durch eines der Tore zu werfen, so erhält das Team 10 Punkte. Die andere Möglichkeit ist es den Schnatz zu fangen durch den Sucher. Dies bringt den Team des Suchers, der den Schnatz gefangen hat, 30 Punkte, allerdings ist das Spiel direkt beendet anschließend, wodurch dies nur zu empfehlen ist, wenn man dadurch mehr Punkte als das gegnerische Team bekommt.
Begründung:	Die punkte sind ein wichtiger Bestandteil des Spiels, da durch sie ein Gewinner und ein Verlierer ermittelt wird.
Abhängigkeiten:	Spielfeld, Torschuss, Schnatz fangen
Priorität:	++
Akteure:	Spiel

ID:	FA10
Titel:	Entfernung
Beschreibung:	Die Entfernung ist die kleinst mögliche Anzahl an Zügen, die man braucht, um Feld A zu Feld B zu kommen. Dabei darf man sich in alle Richtungen bewegen, 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:	Spielfeld, Schussvektor
Priorität:	++
Akteure:	Spieler

ID:	FA11
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 in dem Spiel. In einem Quidditch Spiel kommen 4 Bälle vor. Davon ist einer ein Quaffel, einer ein Schnatz und zwei sind Klatscher.
Begründung:	Die Bälle sind essentieller Bestandteil des Spiels, da sie dem Spiel den Sinn verleihen, dass man Punkte machen kann oder einen Spieler ausnocken kann.
Abhängigkeiten:	Spielfeld
Priorität:	++
Akteure:	Spiel

ID:	FA12
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 kommt der Quaffel auf einem der Felder zum liegen, so springt der Quaffel auf ein zufälliges freies Nachbarfeld.
Begründung:	Der Quaffel wird benötigt, dass das Team Punkte machen kann.
Abhängigkeiten:	Bälle
Priorität:	++
Akteure:	Spiel

ID:	FA13
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:	Die Klatscher sind dazu da um dem Spiel einen Witz zu geben, dass manche Spieler zeitweise nicht verfügbar sind.
Abhängigkeiten:	Bälle
Priorität:	++
Akteure:	Spiel

ID:	FA14
Titel:	Goldener Schnatz [Ball]
Beschreibung:	
Begründung:	Der goldene Schnatz ist ein kleiner goldener Ball, welcher von dem Sucher gesucht werden muss. Der goldene Schnatz fliegt ebenfalls eigenständig über das Spielfeld, wodurch er versucht zu vermeiden von den Suchern gefangen zu werden. Wird der Schnatz jedoch von einem der beiden Suchern gefangen, so gibt dies 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.
Abhängigkeiten:	Der Schnatz verleiht dem Spiel eine Besonderheit, so dass ein Team durch das fangen des Schnatzes die Chance hat, eventuell doch noch zu gewinnen. Außerdem ist somit das Spiel zu ende.
Priorität:	Bälle
Akteure:	++

Spiel

ID:	FA15
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:	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:	Spielfeld
Priorität:	++
Akteure:	Spielfigur

ID:	FA16
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:	Besen
Priorität:	-+
Akteure:	-

ID:	FA17
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:	Besen
Priorität:	-+
Akteure:	-

ID:	FA18
Titel:	Komet 2-60 [Besen]
Beschreibung:	Der Komet 2-60 ist der Besen mittlerer Güte 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:	Besen
Priorität:	-+
Akteure:	-

ID:	FA19
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:	Besen
Priorität:	-+
Akteure:	-

ID:	FA20
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:	Besen
Priorität:	-+
Akteure:	-

ID:	FA21
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 Trikot, 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. Zusätzlich darf jedes Team noch 7 Fans einladen, die das Team durch Zaubersprüche unterstützen können.
Begründung:	Die Teams sind notwendig, da sie der Hauptbestandteil sind, in dem sich alles vereint.
Abhängigkeiten:	Quidditch-Spielfeld, Spielfigur, Besen, Fans
Priorität:	++
Akteure:	Spieler

ID:	FA22
Titel:	Spielfiguren
Beschreibung:	Es gibt 4 Arten von Spielfiguren, 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:	Quidditch-Spielfeld
Priorität:	++
Akteure:	Team

ID:	FA23
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-Werfen
Priorität:	++
Akteure:	Team, Spieler, Spielfigur, Spiel

ID:	FA24
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:	Team, Spieler, Spielfigur, Spiel

ID:	FA25
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:	Der Hüter ist wichtig für das Spiel, da er verhindern soll, dass das gegnerische Team es schafft Punkte zu machen in dem sie den Quaffel durch einen der 3 Torringe werfen.
Abhängigkeiten:	Spielfeld, Torringe, Hüterzone, Quaffel, Quaffel-Werfen, Spieler
Priorität:	++
Akteure:	Team, Spieler, Spielfigur, Spiel

ID:	FA26
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:	Spielefeld, Spieler, goldener Schnatz
Priorität:	++
Akteure:	Team, Spieler, Spielfigur, Spiel

ID:	FA27
Titel:	Quaffel-Werfen
Beschreibung:	<p>Der Quaffel kann nur von Jägern und Hütern geworfen werden. Die beiden Typen von Spielern können den Quaffel aufheben, in dem sie einfach auf das Feld des Quaffels sich bewegen und ihn dadurch aufheben. Möchte ein Spieler den Quaffel werfen, so wählt dieser ein Zielfeld aus, und dadurch wird der Schussvektor erstellt. Die von dem Schussvektor überstrichenen Felder werden nun der Reihe nach betrachtet. Jeder Gegner, der auf einem der überstrichenen Felder steht, hat eine bestimmte Wahrscheinlichkeit den Quaffel abzufangen. Gelingt dies einem gegnerischen Hüter oder Jäger, so geht der Ball in seinen Besitz über. Passiert dies aber bei einem Sucher oder Treiber, so springt der Quaffel auf ein zufälliges freies Nachbarfeld des Spielers. Falls der Quaffel jedoch die überstrichenen Felder durchquert, ohne abgefangen zu werden, so wird der eigentliche Wurf abgehandelt. Der Erfolg des Wurfes ist ebenfalls an eine Wahrscheinlichkeit P^d gebunden, dass dieser Erfolgreich ist, wobei P eine elementare Wurferfolgswahrscheinlichkeit ist, und d die Entfernung zum Zielfeld. Handelt es sich um einen erfolgreichen Wurf, so landet der Quaffel auf dem Zielfeld, ist dies jedoch nicht der Fall, so wird der Quaffel zufällig verteilt auf einem $n \times n$ großen Feld um das Zielfeld, wobei $n = \lceil \frac{d}{7} \rceil$. Toringfelder gelten hierbei auch als freie Felder, wenn kein Spieler drauf steht und im Zuge dessen können bei einem missglückten Wurf auch Eigentore passieren, wenn der Quaffel dadurch auf einem Toringfeld kommt. Für einen erfolgreichen Torschuss muss der Spieler der den Quaffel wirft einen Torschussvektor haben. Dieser ist allerdings nur dann gültig, wenn der Vektor durch eine der Rot markierten Seiten eines Torfeldes läuft, wobei dies auch diagonal erlaubt ist. Der Wurf zählt nicht als Tor, wenn der Vektor über eine der anderen Kanten in das Tor geht.</p>
Begründung:	Diese Anforderung ist sehr wichtig, da sie essentiell wichtig ist, dass das Team punkten kann. Ohne den Quaffel werfen zu können, kann das Team auch keine Punkte erzielen.
Abhängigkeiten:	Spielfeld, Jäger, Hüter, Quaffel
Priorität:	++
Akteure:	Spiel, Spieler

ID:	FA28
Titel:	Quaffel Abfangen
Beschreibung:	Wird ein Quaffel von einem Spieler geworfen, so besteht eine Wahrscheinlichkeit für jedes überstrichene Feld, auf dem ein gegnerischer Spieler steht, dass er den Quaffel abfängt. Tritt dieser Fall ein, so erhält der gegnerische Spieler den Quaffel und der Wurf war nicht erfolgreich.
Begründung:	Dies ist wichtig, da nicht jeder Wurf immer erfolgreich sein kann, da sonst der Ballbesitz sich nie ändern würde.
Abhängigkeiten:	Spiel, Jäger, Hüter, Quaffel, Quaffel-Werfen
Priorität:	++
Akteure:	Spiel

ID:	FA29
Titel:	Quaffel vertändeln
Beschreibung:	Ein Spieler kann den Quaffel vertändeln, was bedeutet, dass er aus irgendeinem Grund die Kontrolle über den Quaffel verliert, welcher dann auf einem zufälligem freien Nachbarfeld landet.
Begründung:	Diese Anforderung ist wichtig, da der Quaffel den Besitz wechseln muss, dass beide Teams punkten können und dies damit möglich wird ohne ein Foul zu begehen.
Abhängigkeiten:	Spiel, Jäger, Hüter, Quaffel
Priorität:	++
Akteure:	Spiel

ID:	FA30
Titel:	Quaffel entreißen
Beschreibung:	Ein Jäger darf einem anderen Jäger, sofern dieser auf einem seiner Nachbarfelder ist, diesen entreißen. Dies gelingt allerdings nur mit einer bestimmten Wahrscheinlichkeit. Jäger dürfen auch dem gegnerischen Hüter den Quaffel entreißen, jedoch dies geht nur, wenn sich der Hüter nicht in seiner eigenen Hüterzone befindet.
Begründung:	Diese Anforderung ist wichtig, da der Quaffel den Besitz wechseln muss, dass beide Teams punkten können und dies damit möglich wird ohne ein Foul zu begehen.
Abhängigkeiten:	Spiel, Jäger, Hüter, Quaffel
Priorität:	++
Akteure:	Spiel, Spieler

ID:	FA31
Titel:	Klatscher Kloppen
Beschreibung:	Ein Treiber kann einen Klatscher kloppen, wenn dieser auf dem Feld wie der Klatscher ist. Der Treiber wählt um den Klatscher zu kloppen ein Zielfeld aus, welches Maximal 3 Felder zu ihm entfernt ist. Zusätzlich müssen aber auch alle überstrichenen Felder frei sein, dass der Wurf gültig ist. Ist dies beides der Fall, so landet der Klatscher auf dem Zielfeld, wo mit einer bestimmten Wahrscheinlichkeit der darauf stehende gegnerische Spieler (Jäger, Sucher, Hüter) ausgenockt wird.
Begründung:	Diese Anforderung bringt eine Schwierigkeit für das Spiel mit, da Spieler dadurch ausgenockt werden können, was das Spiel somit interessanter macht.
Abhängigkeiten:	Spielfeld, Treiber, Klatscher
Priorität:	++
Akteure:	Spiel, Spieler

ID:	FA32
Titel:	Spieler ausgenockt
Beschreibung:	Wird ein Spieler von einem Klatscher ausgenockt, so verändelt dieser Spieler, sofern er den Quaffel hat diesen, und muss das nächste mal wenn er in einer Rundenphase drankommt, entweder noch in dieser oder in der nächsten, aussetzen. Dies bedeutet, dass der Spieler seine Position beibehält und auch auf dieser bleibt, aber keine Aktion mehr ausführen darf für die Rundenphase, also er darf sich nicht bewegen, keine Würfe machen, keine Bälle abfangen und auch keinen Schnatz fangen.
Begründung:	Die Anforderung ist wichtig, damit klar ist, was bei einem Treffer durch einen Klatscher passiert.
Abhängigkeiten:	Klatscher
Priorität:	+
Akteure:	Spiel

ID:	FA33
Titel:	Schnatz fangen
Beschreibung:	Der Schnatz kann nur von einem der Sucher aus jedem Team gefangen werden. Dies geschieht, in dem der Sucher aus einem der beiden Teams es schafft auf das selbe Feld wie der Schnatz zu kommen. In dem Moment gilt der Schnatz als gefangen und das Spiel ist beendet.
Begründung:	Durch das fangen des Schnatzes kriegt das Spiel ein Ende und man hat als Team die Chance eventuell durch die 30 Punkte noch den Sieg zu erreichen.
Abhängigkeiten:	goldener Schnatz, Punkte
Priorität:	++
Akteure:	Spiel, Spieler

ID:	FA34
Titel:	Fans
Beschreibung:	Als Zuschauer sind in <i>Fantastic Feasts</i> magische Wesen vorgesehen, was zur Folge hat, dass diese Einfluss auf das Spielgeschehen haben können. Fans können während dem Spiel Zaubersprüche sprechen, welche dem Team helfen können oder auch schaden können. Es gibt 5 Arten von Fans, die alle eigene spezifische Zaubersprüche haben mit unterschiedlichen Wirkungen.
Begründung:	Diese Anforderung gibt dem Spiel <i>Fantastic Feasts</i> einen witzigen Aspekt und einen gewissen Schwierigkeitsgrad, da auch von den Fans unerwartete Aktionen passieren können.
Abhängigkeiten:	Spielfeld
Priorität:	+
Akteure:	Spiel

ID:	FA35
Titel:	Elfen [Fantyp]
Beschreibung:	Elfen haben die Eigenschaft einen Spieler von der eigenen oder auch von der gegnerischen Mannschaft auf ein zufälliges freies Spielfeld zu Teleportieren.
Begründung:	Diese Anforderung gibt dem Spiel ein zufälliges geschehen, was dem eigenen Team helfen kann oder dem gegnerischen Team schaden kann.
Abhängigkeiten:	Fans
Priorität:	+
Akteure:	Spiel

ID:	FA36
Titel:	Kobolde [Fantyp]
Beschreibung:	Kobolde haben die Fähigkeit einen Schockzauber zu sprechen, welcher bewirkt, dass der Spieler den Quaffel fallen lässt, sofern er diesen hat und anschließend auf ein zufälliges freies Nachbarfeld gestoßen wird.
Begründung:	Diese Anforderung gibt dem Spiel ein zufälliges geschehen, was dem eigenen Team helfen kann oder dem gegnerischen Team schaden kann.
Abhängigkeiten:	Fans
Priorität:	+
Akteure:	Spiel

ID:	FA37
Titel:	Trolle [Fantyp]
Beschreibung:	Da Trolle keine magischen Wesen sind, können sie keinen Zauberspruch ausüben. Dafür sind Trolle sehr Laute Geschöpfe, wodurch sie so laut brüllen können, dass der Spieler, welcher den Quaffel hat, diesen vor Schreck fallen lässt.
Begründung:	Diese Anforderung gibt dem Spiel ein zufälliges geschehen, was dem eigenen Team helfen kann oder dem gegnerischen Team schaden kann.
Abhängigkeiten:	Fans
Priorität:	+
Akteure:	Spiel

ID:	FA38
Titel:	Niffler [Fantyp]
Beschreibung:	Niffler lieben alles was glänzt, wodurch sie nach dem Schnatz schnappen können, wodurch dieser eine Ausweichbewegung macht auf ein zufälliges freies Nachbarfeld von ihm.
Begründung:	Diese Anforderung gibt dem Spiel ein zufälliges geschehen, was dem eigenen Team helfen kann oder dem gegnerischen Team schaden kann.
Abhängigkeiten:	Fans
Priorität:	+
Akteure:	Spiel

ID:	FA39
Titel:	Foul
Beschreibung:	Wie in jeder Sportart ist es in Quidditch möglich sich gegenseitig zu foulern. Es gibt dabei spezifische Fouls die nur von manchen bestimmten Spieler ausgeübt werden können und Fouls die von allen ausgeübt werden können. Nachdem eines der Fouls ausgeübt wurde, wird dieses Foul mit einer jeweils eigenen Wahrscheinlichkeit vom Schiedsrichter entdeckt, woraufhin der ausübende Spieler entsprechend bestraft wird.
Begründung:	Diese Anforderung gehört zu einer Sportart einfach dazu, da Fouls bei jedem Sport vorkommen und das Spiel so spannender machen.
Abhängigkeiten:	Spiel
Priorität:	+
Akteure:	Spiel, Schiedsrichter

ID:	FA40
Titel:	Flacken [Foul]
Beschreibung:	Diese Aktion kann von jedem Spieler ausgeführt werden. Flacken bedeutet, dass ein Spieler sich direkt auf ein Toringfeld stellt, wodurch die Trefferwahrscheinlichkeit für den Schützen auf 0% sinkt. Falls der Quaffel durch den Wurf auf das Toringfeld gekommen wäre, ohne vorher abgefangen zu werden, so springt der Quaffel auf ein zufälliges freies Nachbarfeld.
Begründung:	Dieses Foul ermöglicht es dem Team Tore zu verhindern aber auf eine nicht regelkonforme Möglichkeit.
Abhängigkeiten:	Foul
Priorität:	+
Akteure:	Spiel, Schiedsrichter

ID:	FA41
Titel:	Nachtarocken [Foul]
Beschreibung:	Diese Aktion kann nur von Jägern ausgeübt werden. Führt ein Jäger diese Aktion aus, so erzielt er zu 100% ein Tor, in dem er den Quaffel hält und damit auf ein Toringfeld fliegt, und diesen nicht, wie es die Regeln verlangen würden, von einem Feld außerhalb der Torringe wirft.
Begründung:	Dieses Foul ermöglicht es dem Team Tore mit 100% zu machen, aber man darf sich nicht erwischen lassen. Durch dieses Foul ist eventuell ein schnelleres Punkte holen ermöglicht und man kann eine Niederlage noch aufholen.
Abhängigkeiten:	Foul
Priorität:	+
Akteure:	Spiel, Schiedsrichter

ID:	FA42
Titel:	Stutschen [Foul]
Beschreibung:	Diese Aktion kann nur von Jägern ausgeübt werden. Bei diesem Foul fliegt ein zweiter Jäger in die gegnerische Hüterzone um seinem Team zu helfen Punkte zu machen, obwohl ein Team immer nur einen angreifenden Jäger in der gegnerischen Hüterzone haben darf.
Begründung:	Dieses Foul hilft dem Team leichter einen Punkt zu machen, ist aber gegen die Regeln.
Abhängigkeiten:	Foul
Priorität:	+
Akteure:	Spiel, Schiedsrichter

ID:	FA43
Titel:	Keilen [Foul]
Beschreibung:	Dieses Foul können alle Spieler ausführen. Hierbei zieht ein Spieler auf das selbe Spielfeld wie ein gegnerischer Spieler, woraufhin dieser den Quaffel, sofern er diesen hat vertändelt und anschließend wird der gegnerische Spieler auf ein zufälliges freies Nachbarfeld verdrängt.
Begründung:	Dieses Foul bietet die Möglichkeit den Ballbesitz wieder zu ändern, was aber auch nicht regelkonform ist eigentlich.
Abhängigkeiten:	Foul
Priorität:	+
Akteure:	Spiel, Schiedsrichter

ID:	FA44
Titel:	Schnaltzeln [Foul]
Beschreibung:	Dieses Foul können alle Spieler machen außer der Sucher. Dazu bewegt sich ein Spieler auf das Feld des Schnatzes, obwohl er kein Sucher ist, wodurch er verhindert, dass der Sucher den Schnatz fangen kann, da er das Feld blockiert und somit der Schnatz nicht gefangen werden kann.
Begründung:	Diese Anforderung soll es dem Sucher schwerer machen den Schnatz zu fangen, da das gegnerische Team den Schnatz so blockieren kann.
Abhängigkeiten:	Foul
Priorität:	+
Akteure:	Spiel, Schiedsrichter

ID:	FA45
Titel:	Setzen auf freies Nachbarfeld
Beschreibung:	Soll ein Spieler oder Ball auf ein zufälliges freies Nachbarfeld von seiner Position gesetzt werden, aber es sind alle 8 umliegenden Felder bereits voll, oder können durch den Spielfeldrand nicht ausgewählt werden, so wird rekursiv von einem zufällig besetzten Nachbarfeld weiter gesucht, bis sich schließlich ein freies Feld findet. Dies bedeutet, dass der Spieler oder Ball nicht unbedingt auf einem Nachbarfeld seiner Position landet, sondern auch weiter entfernt gesetzt werden kann und so durch andere besetzte Felder sich hindurchtunnelt.
Begründung:	Diese Anforderung ist notwendig, da man den Fall abdecken muss, falls kein freies Nachbarfeld mehr frei ist.
Abhängigkeiten:	Spielfeld
Priorität:	++
Akteure:	Spiel

ID:	FA46
Titel:	Partie-Konfiguration
Beschreibung:	JSON Datei mit Informationen über eine Spiel. Zu den Informationen zählen die maximale Rundenzahl bevor die Überlängenbedingung eintritt, die Zeit, welche die Spieler haben um den nächsten gewünschten Zug zu übermitteln und welche Wahrscheinlichkeiten den einzelnen Ereignissen zugeordnet sind.
Begründung:	Um das Spiel abwechslungsreicher zu gestalten ist es von Nöten, dass verschiedenen Abläufe in einer Partie angepasst werden können. Um Kompatibilität zwischen verschiedenen Komponenten zu gewährleisten ist es dabei zudem wichtig, dass diese Dateien einheitlich Formatiert sind.
Abhängigkeiten:	Spiellogik
Priorität:	-+
Akteure:	Server, Spiel, Systemadministrator

ID:	FA47
Titel:	Quidditchteam-Konfiguration
Beschreibung:	Lokal hinterlegte Datei im <i>JSON</i> Format in der die Zusammensetzung des Teams eines Spielers hinterlegt ist.
Begründung:	Die Konfiguration eines Teams muss zwischen dem Team-Editor und dem Client ausgetauscht werden können und muss daher in einer einheitlichen Datei lokal hinterlegt werden.
Abhängigkeiten:	Team, Spielfiguren
Priorität:	+
Akteure:	Nutzer, Client, KI-Client, Quidditchteam-Editor, Server

ID:	FA48
Titel:	Netzwerkschnittstelle
Beschreibung:	Zwei Clients sollen über eine Netzwerkschnittstelle miteinander kommunizieren können.
Begründung:	Bei <i>Fantastic Feasts</i> handelt sich es um ein Online Multiplayer Spiel. Es ist also notwendig, dass einzelne Komponenten miteinander Kommunizieren können.
Abhängigkeiten:	-
Priorität:	++
Akteure:	Systemadministrator, Entwickler, Client, KI-Client, Server

ID:	FA49
Titel:	Runde
Beschreibung:	Das Spiel läuft in Runden ab. Jede Runde ist dabei in die drei Rundenphasen Ballphase, Spielerphase und Fanphase unterteilt.
Begründung:	Bei <i>Fantastic Feasts</i> handelt es sich laut den Spielregeln um ein Rundenbasiertes Spiel.
Abhängigkeiten:	Ballphase, Spielerphase, Fanphase
Priorität:	+
Akteure:	Spiel, Spieler, Team, Spielfigur

ID:	FA50
Titel:	Ballphase
Beschreibung:	In dieser Phase eines Spiels bewegen sich die Bälle über das Spielfeld. Dabei machen die beiden Klatscher ihre Bewegungen in zufälliger Reihenfolge.
Begründung:	
Abhängigkeiten:	Runde
Priorität:	+
Akteure:	Spiel

ID:	FA51
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:	Runde
Priorität:	+
Akteure:	Spiel

ID:	FA52
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:	Runde
Priorität:	+
Akteure:	Spiel

ID:	FA53
Titel:	Zufallsgenerator
Beschreibung:	Einheit die per Zufall entscheidet, ob bestimmt Ereignisse eintreten oder nicht.
Begründung:	An mehreren Stellen in der Spiellogik ist es notwendig, dass via Zufall Entschieden wird ob ein Ereignis eintritt oder nicht. Dies ist z.B. der Fall wen entschieden werden muss ob der Schiedsrichter ein Foul ahndet oder wenn entschieden werden muss ob ein geworfener Ball geblockt wird.
Abhängigkeiten:	Spiellogik
Priorität:	-+
Akteure:	Server, Schiedsrichter

ID:	FA54
Titel:	Spielende
Beschreibung:	Die Partie ist vorüber und es gibt einen Gewinner. Dies ist der Fall wenn entweder ein Spieler Disqualifiziert wird oder ein Sucher in der Rundenphase den Goldenen Schnatz findet. Gewonnen hat dann das Team, das entweder nicht disqualifiziert wurde oder die meisten Punkte erzielt hat. Bei Punktegleichstand gewinnt das Team, welches den Goldenen Schnatz gefunden hat.
Begründung:	Das Spielende muss erkannt und geprüft werden, damit ein Spieler der Partie zum Sieger gekrönt werden kann.
Abhängigkeiten:	Spiellogik, Runde, Goldener Schnatz, Sucher
Priorität:	+
Akteure:	Spiel, Spieler, Schiedsrichter

ID:	FA55
Titel:	Überlängenbehandlung
Beschreibung:	Reaktionen, welche getroffen werden, falls eine Partie zu lange läuft. Zieht sich ein Spiel über mehr Runden hin, als in der Partie-Konfiguration über einen Höchstwert fest, ohne dass ein Sieger ermittelt werden kann, so wird das Verhalten des Goldenen Schnatz angepasst. Die Wahrscheinlichkeit, dass ein Sucher den Goldenen Schnatz findet wird auf 100% gesetzt. Falls dann nach drei Runden das Spiel immer noch läuft bewegt sich der Goldenen Schnatz ohne Suchern auszuweichen in die Mitte des Spielfelds und verharrt dort. Wird der dort nach weiteren drei Runden immer noch nicht gewunden, so fliegt er in der nächsten Rundenphase dem nächsten Sucher ins Gesicht.
Begründung:	Dadurch wird sichergestellt das ein Spiel spannend bleibt und die Spieler nicht die Lust verlieren, wenn eine Partie zu zäh läuft.
Abhängigkeiten:	Spielende, Spiellogik, Runde, Goldener Schnatz, Sucher
Priorität:	-+
Akteure:	Spiel, Spieler

ID:	FA56
Titel:	Spiellogik
Beschreibung:	Einheit, in der die Züge der Teams verarbeitet werden und auf Basis der Spielregeln im Lastenheft Entscheidungen getroffen werden. Zudem wird von der Spiellogik bestimmt welcher Spieler gerade am Zug ist.
Begründung:	Ohne eine Spiellogik wären alle Anwendungen nutzlos.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	++
Akteure:	Server, Spiel

ID:	FA57
Titel:	Log-Datei
Beschreibung:	Datei zum Speichern bestimmter Ereignisse. Diese Datei wird lokal auf dem Endgerät hinterlegt.
Begründung:	Um die während der Nutzung der Software aufgetretenen Aktionen im Nachhinein nachvollziehen zu können und daraus Informationen für Statistiken und Wartung ziehen zu können ist es wichtig das diese Aktionen lokal hinterlegt werden.
Abhängigkeiten:	Spiellogik
Priorität:	-
Akteure:	Client, KI-Client, Server

4.3 Server spezifische funktionale Anforderungen

ID:	FA58
Titel:	Spiellogik
Beschreibung:	Einheit, in der die Züge der Teams verarbeitet werden und auf Basis der Spielregeln im Lastenheft Entscheidungen getroffen werden. Zudem wird von der Spiellogik bestimmt welcher Spieler gerade am Zug ist.
Begründung:	Ohne eine Spiellogik wären alle Anwendungen nutzlos, da die Spiellogik die Einheit bildet, welche alles für das Spiel relevanten Informationen verarbeitet.
Abhängigkeiten:	Spiel, Benutzereingaben Validierung
Priorität:	++
Akteure:	Server, Spiel

ID:	FA59
Titel:	Partie-Konfiguration
Beschreibung:	JSON Datei mit Informationen über eine Spiel. Zu den Informationen zählen die maximale Rundenzahl bevor die Überlängenbedingung eintritt, die Zeit, welche die Spieler haben um den nächsten gewünschten Zug zu übermitteln und welche Wahrscheinlichkeiten den einzelnen Ereignissen zugeordnet sind.
Begründung:	Um das Spiel abwechslungsreicher zu gestalten ist es von Nöten, dass verschiedenen Abläufe in einer Partie angepasst werden können. Um Kompatibilität zwischen verschiedenen Komponenten zu gewährleisten ist es dabei zudem wichtig, dass diese Dateien einheitlich Formatiert sind.
Abhängigkeiten:	Spiellogik
Priorität:	-+
Akteure:	Server, Spiel, Systemadministrator

ID:	FA60
Titel:	Überlängenbehandlung
Beschreibung:	Reaktionen, welche getroffen werden, falls eine Partie zu lange läuft. Zieht sich ein Spiel über mehr Runden hin, als in der Partie-Konfiguration über einen Höchstwert fest, ohne dass ein Sieger ermittelt werden kann, so wird das Verhalten des Goldenen Schnatz angepasst. Die Wahrscheinlichkeit, dass ein Sucher den Goldenen Schnatz findet wird auf 100% gesetzt. Falls dann nach drei Runden das Spiel immer noch läuft bewegt sich der Goldenen Schnatz ohne Suchern auszuweichen in die Mitte des Spielfelds und verharrt dort. Wird der dort nach weiteren drei Runden immer noch nicht gewunden, so fliegt er in der nächsten Rundenphase dem nächsten Sucher ins Gesicht.
Begründung:	Dadurch wird sichergestellt das ein Spiel spannend bleibt und die Spieler nicht die Lust verlieren, wenn eine Partie zu zäh läuft.
Abhängigkeiten:	Spielende, Spiellogik, Runde, Goldener Schnatz, Sucher
Priorität:	-+
Akteure:	Spiel, Spieler

ID:	FA61
Titel:	Zufallsgenerator
Beschreibung:	Einheit die per Zufall entscheidet, ob bestimmte Ereignisse eintreten oder nicht.
Begründung:	An mehreren Stellen in der Spiellogik ist es notwendig, dass via Zufall Entschieden wird ob ein Ereignis eintritt oder nicht. Dies ist z.B. der Fall wen entschieden werden muss ob der Schiedsrichter ein Foul ahndet oder wenn entschieden werden muss ob ein geworfener Ball geblockt wird.
Abhängigkeiten:	Spiellogik
Priorität:	-+
Akteure:	Server, Schiedsrichter

4.4 Client spezifische Funktionale Anforderungen

ID:	FA62
Titel:	Hauptmenü [Ansicht]
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. Es soll also unter anderem ein Spiel gestartet werden können, die Hilfe aufgerufen werden können, die Einstellungen der Anwendung angepasst werden können und eventuell vorhandene Statistiken aufgerufen werden können.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	+
Akteure:	Nutzer

ID:	FA63
Titel:	Spiel beitreten [Ansicht]
Beschreibung:	Grafische Oberfläche um sich mit einem Server auf dem ein Spiel bereit gestellt wird zu verbinden. Dabei soll man außerdem die Möglichkeit haben seine Team-Konfiguration an zu geben, die man für das neue Spiel verwenden möchte.
Begründung:	Der Nutzer muss die Möglichkeit haben sich komfortabel mit einem Server verbinden zu können.
Abhängigkeiten:	Abb.
Priorität:	+
Akteure:	Spieler, Gast

ID:	FA64
Titel:	Spiel Ende [Ansicht]
Beschreibung:	Grafische Oberfläche die, die Spieler sehen nachdem eine Partie zu Ende ist. Der Nutzer sollte hier auch die Möglichkeit haben die Anwendung zu verlassen oder wieder ins Hauptmenü zurück kehren. Optional ist hier auch Platz für etwaige Statistiken über den Spielverlauf.
Begründung:	Nach dem Ende einer Partie muss dem Nutzer mitgeteilt werden ob er gewonnen hat oder nicht und wie es von da an weiter geht.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	-+
Akteure:	Spieler

ID:	FA65
Titel:	Team-Konfiguration importieren [Ansicht]
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:	-+
Akteure:	Spieler

ID:	FA66
Titel:	Spiel [Ansicht]
Beschreibung:	Grafische Oberfläche die der Spieler sieht.
Begründung:	Es handelt sich um eine Anwendung mit grafischer Benutzeroberfläche. Es ist also zwingend von Nöten, dass auch das aktuelle Spielgeschehen angezeigt werden kann.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	++
Akteure:	Spieler

ID:	FA67
Titel:	Hilfe [Ansicht]
Beschreibung:	Grafische Oberfläche, in der zum einen das Spielprinzip erklärt wird und zum anderen gezeigt wird wie genau man die Client-Software bedient wird.
Begründung:	Um unerfahren Benutzer die Bedienung der Software zu erleichtern.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	-+
Akteure:	Nutzer

ID:	FA68
Titel:	Beobachter [Ansicht]
Beschreibung:	Grafische Oberfläche, die ein Gast sieht.
Begründung:	Wenn ein Nutzer einem Spiel nur als Gast zuschaut muss ihm das Spiel trotzdem in eine grafische Oberfläche aufbereitet werden.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	+
Akteure:	Gast

ID:	FA69
Titel:	Spiel Rendering Engine
Beschreibung:	Die Spiel Rendering Engine bereitet die Ansicht des Spielfelds grafisch auf.
Begründung:	Die Benutzeroberfläche muss während des Spiels mit Inhalt gefüllt werden. Dieser Inhalt muss je nach Spielgeschehen automatisch generiert werden.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	++
Akteure:	Client, Spieler

ID:	FA70
Titel:	Eingabenverarbeitung
Beschreibung:	Diese Einheit ist für die Verarbeitung von Benutzereingaben verantwortlich.
Begründung:	Jede Benutzereingabe muss ausgewertet werden. Nach der Validierung muss dann eine Entscheidung getroffen werden, was als Reaktion auf diese Eingaben passieren muss.
Abhängigkeiten:	Spiellogik
Priorität:	++
Akteure:	Spieler, KI

ID:	FA71
Titel:	Eingaben Validierung
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:	Spiellogik
Priorität:	<Prio>
Akteure:	Spieler, KI

ID:	FA72
Titel:	Hotkey
Beschreibung:	Oft benötigte Funktionen auf bestimmte (besondere) Testen (-Kombinationen) abzubilden.
Begründung:	Hotkeys sind optionale Features, die im Lastenheft aufgeführt sind und zu einer einfacheren Spielsteuerung und höherem Spielkomfort betragen können.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	–
Akteure:	Spieler

ID:	FA73
Titel:	Pausieren
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 um den Spielkomfort zu erhöhen.
Abhängigkeiten:	<Abhängigkeiten>
Priorität:	–
Akteure:	Spieler

4.5 Quidditchteam-Editor spezifische funktionale Anforderungen

ID:	FA74
Titel:	Team Konfiguration bearbeiten
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:	Quidditchteam-Konfiguration
Priorität:	-
Akteure:	Nutzer

ID:	FA75
Titel:	Team Konfiguration öffnen
Beschreibung:	'Datei Öffnen' Dialog um die im <i>JSON</i> Format gespeicherte Team-Konfiguration von einem beliebigen Ort im Dateisystem zu öffnen.
Begründung:	Um eine hohe Benutzerfreundlichkeit zu garantieren muss es den Nutzer möglich sein einfach eine bereits erstellt Konfiguration für ein Team einzulesen und die zu bearbeiten.
Abhängigkeiten:	Team Konfiguration bearbeiten, Quidditchteam-Konfiguration
Priorität:	-
Akteure:	Nutzer

ID:	FA76
Titel:	Team Konfiguration speichern
Beschreibung:	'Datei Speichern' Dialog um die Team-Konfiguration an einem beliebigen Ort im Dateisystem im <i>JSON</i> Format abzulegen.
Begründung:	Ohne die Möglichkeit die Team-Konfiguration lokal permanent zu speichern wäre das Benutzererlebnis und die Bedienbarkeit der Software massiv beeinträchtigt.
Abhängigkeiten:	Team Konfiguration bearbeiten, Quidditchteam-Konfiguration
Priorität:	-
Akteure:	Nutzer

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 von großer Bedeutung, da die Anwendungen auf möglichst vielen Zielsystemen funktionieren sollen um die Menge an Endnutzer so wenig wie möglich einzuschränken.
Abhängigkeiten:	Programmiersprache
Priorität:	++
Akteure:	Nutzer, Entwickler

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:	Entwickler

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:	Entwickler

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 diesen Tool nicht schlechter als 'B' bewertet werden.
Begründung:	Quellcode mit einer hohen Codequalität ist weniger anfällig für Fehler und Probleme.
Abhängigkeiten:	-
Priorität:	+
Akteure:	Entwickler

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:	Entwickler

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:	Plattformunabhängigkeit
Priorität:	+
Akteure:	Entwickler

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. Die gesamte Dokumente ist dabei mit dem Tool Doxygen zu erstellen.
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:	Dokumentation
Priorität:	-+
Akteure:	Nutzer

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:	Nutzer

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:	Entwickler

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:	Entwickler

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 verwendet werden, welche von allen Teammitgliedern beherrscht wird.
Abhängigkeiten:	-
Priorität:	++
Akteure:	Entwickler

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:	Durch Einheitliche Formate der Konfigurationsdateien lässt sich sicherstellen, dass einzelne Komponenten zwischen den Entwicklungsteams ausgetauscht werden können und diese miteinander kompatibel sind.
Abhängigkeiten:	-
Priorität:	+
Akteure:	Entwickler, Nutzer

ID:	QA14
Titel:	Netzwerkcommunication
Beschreibung:	Die Netzwerkcommunication zwischen Client und Server soll über sogenannte <i>Web-Socket-Sessions</i> realisiert werden, so dass 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:	++
Akteure:	Spiel

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:	Spiel

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:	Kunde, Nutzer, Entwickler

ID:	QA17
Titel:	Zuverlässigkeit
Beschreibung:	Die Anwendungen dürfen niemals komplett abstürzen.
Begründung:	Durch zu häufiges Abstürzen der Software ist das Benutzererlebnis massiv beeinträchtigt.
Abhängigkeiten:	Robustheit
Priorität:	+
Akteure:	Nutzer

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:	Nutzer

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:	Nutzer

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:	Entwickler

ID:	QA21
Titel:	Effizienz
Beschreibung:	Die Software sollte Ressourcen schonend arbeiten. Keine Komponente darf mehr als ein Gigabyte Arbeitsspeicher benötigen und Keine Komponente darf mehr als 50% der auf dem System zur Verfügung stehenden Prozessorleistung benötigen. Zudem darf im Durchschnitt während einer Partie nicht mehr als 1MBit/s an Netzwerkbandbreite benötigt werden um das Spiel ohne Einschränkungen nutzen zu können.
Begründung:	Eine ressourcenschonende Anwendung ist auch auf älteren Ziel-systemen problemlos nutzbar.
Abhängigkeiten:	-
Priorität:	-
Akteure:	Nutzer, Entwickler

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