ulm university universität
uulm

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

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 Funktionale Anforderungen: Spielregeln 24
4.3 Funktionale Anforderungen: Allgemein 42
4.4 Funktionale Anforderungen: Server 44
4.5 Funktionale Anforderungen: Client 45
4.6 Funktionale Anforderungen: Quidditchteam-Editor 48
4.7 Funktionale Anforderungen: KI-Client 49
4.8 Nicht funktionale Anforderungen 50

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 genüberstehenden 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, Torring, Hüterzonenzelle
Aspekt	Mögliche Standorte der Spielobjekte
Bemerkung	Wird nicht Feld genannt, um Verwechselungen 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 Torring 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 LongEiländ

Begriff	**Hüter**
Beschreibung	Spielfigur, deren Aufgabe es ist, den Quaffel von den eigenen Zielfeldern 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 Torring befördern 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 Torschuss 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 Torring, was das Abliefern garantiert.
Ist-ein	Foul
Kann-sein	-
Aspekt	Taktik
Bemerkung	„Nachtarocken“ im Lastenheft
Beispiel	Han Solo stürmt mittleren gegnerischen Torring.

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.

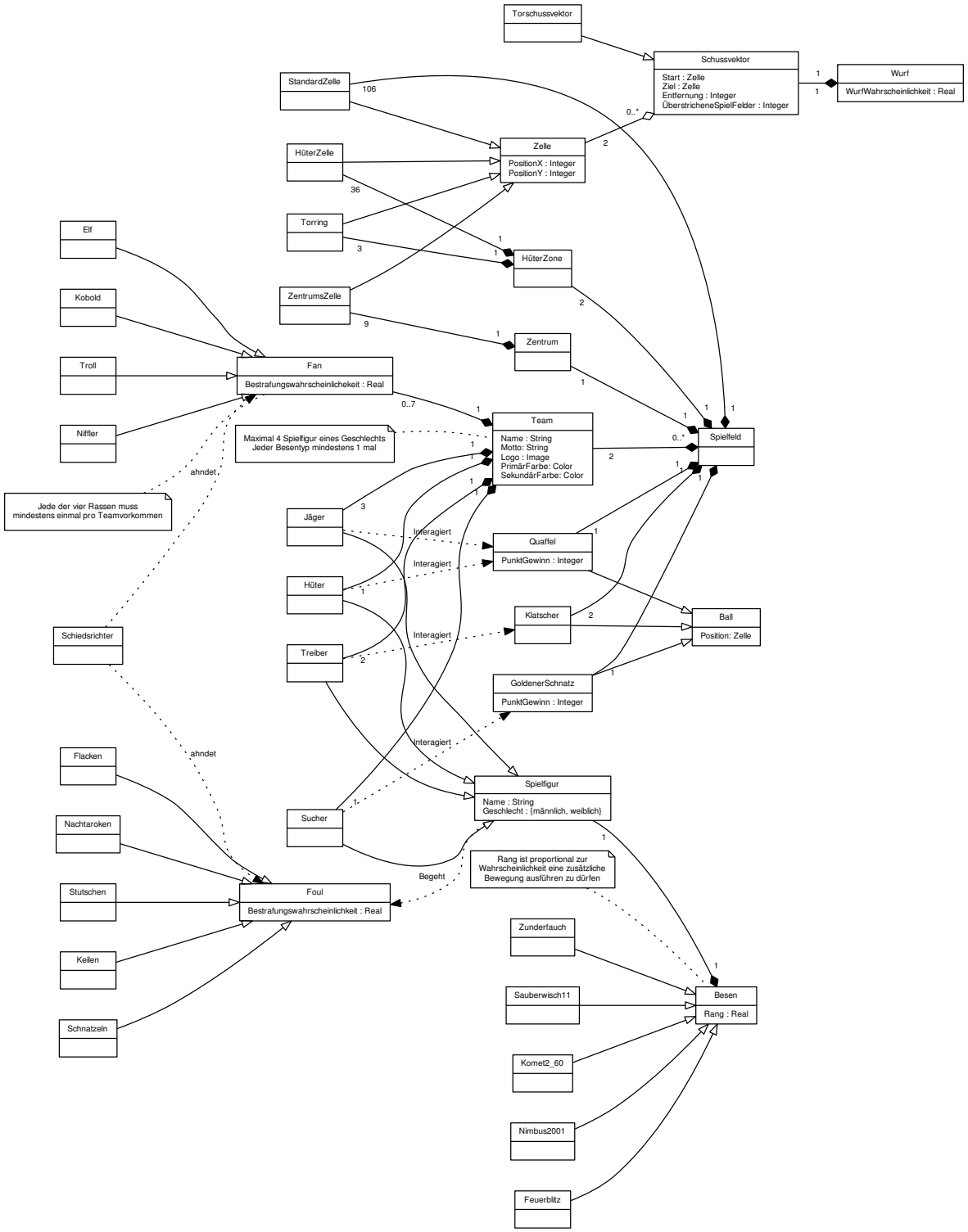
Begriff	**Rammen**
Beschreibung	Eine eigene Spielfigur zieht auf eine Zelle, die von einer gegnerischen Spielfigur besetzt wird. Dadurch wird die gegnerische Spielfigur auf eine benachbarte Zelle bewegt und verliert den Quaffel
Ist-ein	Foul
Kann-sein	-
Aspekt	Taktik
Bemerkung	„Keilen“ im Lastenheft
Beispiel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

3 Domänenmodell



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.

ID:	**AKT2**
Titel:	Spieler
Beschreibung:	Spiet das Spiel „Fantastic Feasts“.
Rolle:	Nimmt aktiv Einfluss auf das Spielgeschehen. Ist entweder Nutzer oder KI.

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

ID:	**AKT4**
Titel:	Systemadministrator
Beschreibung:	Person, die die Möglichkeit hat, die Serveranwendung des Projektes zu verwalten.
Rolle:	Der Systemadministrator ist dafür verantwortlich, eine Instanz der Serveranwendung 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:	Der Entwickler implementiert die gesamte Anwendung.

ID:	AKT6
Titel:	KI
Beschreibung:	Vom Computer gesteuerter Spieler.
Rolle:	Spieler, dessen Entscheidungen und Züge von einem Computerprogramm, dem KI-Client, getroffen werden. Es wird somit ein menschlichen Spieler.

ID:	AKT7
Titel:	Kunde
Beschreibung:	Der Kunde gibt das Projekt in Auftrag.
Rolle:	Stellt Anforderungen und Wünsche an das Entwicklerteam und nimmt das Projekt ab.

ID:	AKT8
Titel:	Client
Beschreibung:	Programm, das einem Nutzer eine grafische Oberfläche, zum Spielen oder Beobachten des Spiels, zur Verfügung stellt.
Rolle:	Der Client stellt eine Verbindung zum Server her, visualisiert die empfangenen Daten und sendet seinerseits die Eingaben des Nutzers.

ID:	AKT9
Titel:	KI-Client
Beschreibung:	Simuliert einen menschlichen Gegner.
Rolle:	Der KI-Client kommuniziert wie der normale Client mit dem Server. Allerdings werden die Entscheidungen von der KI getroffen und nicht von einem Nutzer. Er stellt keine grafische Oberfläche zur Verfügung.

ID:	AKT10
Titel:	Server
Beschreibung:	Zentrale Komponente des Projekts, die alle anderen Komponenten vernetzt.
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:	Computerprogramm, mit dem Team-Konfigurationen erstellt und bearbeitet werden können.
Rolle:	Der Quidditchteam-Editor erstellt Teamkonfigurationsdateien, die später vom Client geladen werden, um ein Spiel zu starten.

Bemerkungen zu den Abhängigkeiten der Anforderungen:
Abhängigkeiten werden aus Gründen der Übersichtlichkeit vererbt. Beispielsweise besitzt die Mittelzelle implizit alle Abhängigkeiten des Mittelkreises.

Bemerkungen zu den Prioritäten:

- \- Optionale Komponente
- 0 Geforderte Komponente
- \+ Geforderte, wichtige aber nicht zeitkritisch Komponente
- ++ Geforderte, wichtige und zeitkritische Komponente

4.2 Funktionale Anforderungen: Spielregeln

ID:	FA1
Titel:	Quidditch-Spielfeld
Beschreibung:	Das Quidditch-Spielfeld hat eine Ovale Form, die in ein Raster von 17x13 quadratischen Zellen eingepasst ist. Auf diesem Feld finden alle Spielhandlungen statt, die während dem Spiel getätigt werden können.
Begründung:	Das Spielfeld ist die zentrale Komponente des Spiels, da sich hier während einer Partie sämtliche Abläufe abspielen.
Abhängigkeiten:	-
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	**FA2**
Titel:	Mittelkreis
Beschreibung:	Der Mittelkreis ist ein Bereich auf dem Quidditch-Spielfeld, der in der Mitte angeordnet ist und aus 3x3 quadratischen Zellen besteht. Zu Beginn dürfen sich hier keine Spielfiguren befinden.
Begründung:	Der Mittelkreis markiert das Zentrum des Spielfeldes, in dem das Spiel gestartet wird.
Abhängigkeiten:	Spielfeld
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	**FA3**
Titel:	Mittelzelle
Beschreibung:	Die Mittelzelle stellt den mittleren Punkt des Mittelkreises dar, der sich im Zentrum des Quidditch-Spielfeldes befindet. Dort werden initial die Bälle des Spiels platziert.
Begründung:	Die Mittelzelle ist der Startpunkt für die Bälle beim Spielstart.
Abhängigkeiten:	Mittelkreis
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	**FA4**
Titel:	Hüterzonen
Beschreibung:	Die Hüterzonen sind an den jeweils gegenüberliegenden Seiten des Quidditch-Spielfeldes platziert. Die Hüterzonen sind ovalförmig, bestehen aus 11x5 Kacheln und beinhalten jeweils 3 Torringe.
Begründung:	In den Hüterzonen können die Teams Punkte erzielen.
Abhängigkeiten:	Spielfeld
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	**FA5**
Titel:	Zelle
Beschreibung:	Die Zelle ist die kleinste Einheit des Spielfeldes, auf ihr darf sich immer nur eine Spielfigur gleichzeitig 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:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	**FA6**
Titel:	Torring
Beschreibung:	Die Teams können Punkte erzielen, indem sie den Quaffel durch einen generischen Torring werfen.
Begründung:	Die Torringe dienen den Teams als Hauptquelle von Punkten.
Abhängigkeiten:	Hüterzone
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	**FA7**
Titel:	Schussvektorberechnung
Beschreibung:	Ein Schussvektor zeigt vom Mittelpunkt der Startzelle auf die Zielzelle des Wurfs. Alle Zellen, die von diesem Vektor geschnitten werden, sind so genannte überstrichene Felder.
Begründung:	Ein Schussvektor beschreibt, wie eine Spielfigur einen Ball über das Spielfeld bewegten kann.
Abhängigkeiten:	Zelle
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA8
Titel:	Punkte erzielen
Beschreibung:	Es gibt zwei Möglichkeiten, Punkte zu erzielen: Den Quaffel durch ein gegnerischen Torring werfen (entspricht 10 Punkten) oder den Goldenen Schnatz fangen (entspricht 30 Punkten).
Begründung:	Die Punktezahl zeigt an, welcher Spieler sich im Moment besser schlägt und dient zur Bestimmung des Gewinners am Ende der Partie.
Abhängigkeiten:	Schnatz fangen, Quaffel werfen
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA9
Titel:	Entfernungsberechnung
Beschreibung:	Die Entfernung zwischen zwei Zellen ist die kleinstmögliche Anzahl an Zügen, die man braucht, um Zelle A zu Zelle B zu kommen. Dabei darf man sich in alle Richtengen bewegen, also Vertikal, Horizontal und Diagonal.
Begründung:	Die Entfernung ist maßgeblich für den Erfolg von verschiedenen Aktionen, wie z.B. dem Werfen des Quaffels.
Abhängigkeiten:	Zelle
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA10
Titel:	Bälle
Beschreibung:	Es gibt 3 verschiedene Arten von Bällen: Der Quaffel, der Klatscher und der Goldene Schnatz.
Begründung:	Die Bälle sind zentraler Bestandteil des Spiels.
Abhängigkeiten:	-
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA11
Titel:	Quaffel [Ball]
Beschreibung:	Der Quaffel ist ein roter Lederball, mit dem die Team Punkte erzielen kann. Dazu muss der Quaffel durch einen gegnerischen Torring befördert werden.
Begründung:	Der Quaffel ist die zentrale Punktequelle.
Abhängigkeiten:	Bälle
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA12
Titel:	Klatscher [Ball]
Beschreibung:	Der Klatscher ist ein kleiner schwarzer Ball, der sich von alleine auf Spieler zubewegt (eine Zelle pro Runde), die keine Treiber sind.
Begründung:	Die Klatscher verleihen dem Spiel zusätzliche taktische Tiefe, da sie Spielfiguren für eine Runde ausschalten können.
Abhängigkeiten:	Bälle
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA13
Titel:	Goldener Schnatz [Ball]
Beschreibung:	Der Goldene Schnatz ist eine kleiner goldener Ball, sich von alleine von Suchern wegbewegt (eine Zelle pro Runde). Das bedeutet, er achtet auf den nächsten Sucher, und wählt unter allen möglichen freien Zellen, die eine größere Entfernung zu diesem haben, als seine gegenwärtige, eine zufällige aus, und bewegt sich darauf. Falls es keine solchen Zellen gibt, bewegt sich der Schnatz auf eine zufälliges freie Nachbarzelle. Der Goldene Schnatz erscheint zu Beginn der dreizehnten Runde auf einer zufällig gewählten freien Zelle, das möglichst gleich weit von beiden Suchern entfernt ist.
Begründung:	Der Goldene Schnatz dient zum Punkteerzielen und führt, wenn er gefangen wird, zum Ende des Partie.
Abhängigkeiten:	Bälle
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA14
Titel:	Besen
Beschreibung:	Jede Spielfigur besitzt einen Besen, die einen der folgenden Typen haben: Zauberfauch, Sauberwisch 11, Komet 2-60, Nimbus 2001 und Feuerblitz. Der Typ des Besens bestimmt die Wahrscheinlichkeit, mit der eine Spielfigur nach einer Bewegung um ein Feld eine weitere Bewegung ausführen darf. Diese Wahrscheinlichkeit wird in der Partiekonfiguration festgelegt, wobei die Besen in der genannten Reihenfolge aufsteigende Wahrscheinlichkeiten besitzen.
Begründung:	Die Besen geben den Spielfiguren eine unterschiedliche Qualität.
Abhängigkeiten:	-
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA15
Titel:	Teams
Beschreibung:	Ein Team besteht aus sieben Spielfiguren und sieben Fans. Außerdem hat jedes Team einen Namen, ein Motto, eine Hauptteamfarbe und eine Ersatzteamfarbe. Die sieben Spielfiguren Teilen sich wie folgt auf: ein Hüter, zwei Treiber, drei Jäger und ein Sucher. Bei den Spielfiguren darf jedes Geschlecht bis zu vier mal vertreten sein. Zudem muss jeder Besentyp einmal vertreten sein. Bei den sieben Fans muss jeder Fantyp mindestens einmal vertreten sein.
Begründung:	Quidditch ist ein Teamspiel, weshalb Teams benötigt werden.
Abhängigkeiten:	Spielfigur, Fans, Besen
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA16
Titel:	Spielfiguren
Beschreibung:	Es gibt 4 Arten von Spielfiguren: Jäger, Sucher, Hüter und Treiber. Jede Spielfigur hat dabei einen Namen und ein Geschlecht.
Begründung:	Die unterschiedlichen Typen der Spielfiguren geben dem Spiel taktische Tiefe.
Abhängigkeiten:	-
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA17
Titel:	Jäger [Spielfigur]
Beschreibung:	Jäger können den Quaffel aufnehmen und werfen und damit Punkte für ihr Team erzielen.
Begründung:	Jäger können Punkte für ihr Team erzielen.
Abhängigkeiten:	Spielfigur
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA18
Titel:	Treiber [Spielfigur]
Beschreibung:	Treiber können den Klatscher schlagen und somit zum Gegner hin und / oder von Teammitgliedern weg befördern.
Begründung:	Treiber dienen zum Schutz des eigenen Teams vor den Klatschern. Gleichzeitig können sie den Gegner aktiv sabotieren, in dem sie ihm den Klatscher zuspielen..
Abhängigkeiten:	Spielfigur
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA19
Titel:	Hüter [Spielfigur]
Beschreibung:	Hüter können den Quaffel aufnehmen und versuchen, den Gegner daran zu hindern, ein Tor zu erzielen. Landet der Quaffel auf einem Torring so geht der Quaffel am ende der Rudenphase in den Besitz des Hüters über, wenn er sich selbst in der Hüterzone befindet. Ein Hüter kann selbst keine Tore erzielen.
Begründung:	Hüter stellen die Verteidigung seines Teams dar.
Abhängigkeiten:	Spielfigur, Hüterzone
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA20
Titel:	Sucher [Spielfigur]
Beschreibung:	Sucher versuchen den Goldenen Schnatz zu finden, um Punkte zu erzielen und das Spiel zu beenden.
Begründung:	Der Sucher beendet das Spiel.
Abhängigkeiten:	Spielfigur
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA21
Titel:	Quaffel-Werfen
Beschreibung:	Hüter und Jäger können den Quaffel werfen. Der Wurf wird über ein Schussvektor angegeben. Jede gegnerische Spielfigur, die sich auf einer überstrichenen Zelle des Schussvektors befindet, kann den Quaffel abfangen. Wird der Ball von keiner Spielfigur abgefangen, so ist der Wurf mit der Wahrscheinlichkeit P^d erfolgreich, wobei P eine elementare Wurfwahrscheinlichkeit und d die Entfernung zur Zielzelle ist. Ist der Wurf erfolgreich, so landet der Quaffel auf der Zielzelle. Wenn nicht wir der Quaffel auf einem zufälligen freien Zelle in einem $n \times n$ Quadrat um die Zielzelle platziert, wobei $n = \lceil \frac{d}{7} \rceil$ ist.
Begründung:	Das Werfen des Quaffels ermöglicht Passspiel und das erzielen von Punkten.
Abhängigkeiten:	Schussvektorberechnung, Entfernungsberechnung, Jäger, Hüter, Quaffel
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA22
Titel:	Quaffel Abfangen
Beschreibung:	Jede gegnerische Spielfigur auf einer überstrichenen Zelle des Schussvektors eines Wurfes hat eine gewisse Wahrscheinlichkeit, den Quaffel abzufangen. Diese Wahrscheinlichkeit ist in der Partiekonfiguration vermerkt. Gelingt das Abfangen, so landet der Quaffel auf der Zelle der Abfangenden Spielfigur.
Begründung:	Das Abfangen bietet die Möglichkeit, Würfe des Gegners zu unterbinden.
Abhängigkeiten:	Quaffel-Werfen, Spielfiguren
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA23
Titel:	Quaffel prallt ab
Beschreibung:	Landet der Quaffel auf einer Zelle, auf der sich eine Spielfigur befindet, die weder Hüter noch Jäger ist, so wird der Quaffel auf eine zufällige freie Nachbarzelle gesetzt.
Begründung:	Nur Jäger und Hüter können direkt mit der Quaffel interagieren.
Abhängigkeiten:	Sucher, Treiber, Quaffel
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA24
Titel:	Quaffel Halten
Beschreibung:	Ein Jäger oder Hüter kann den Quaffel halten, was zur Folge hat, dass der Quaffel sich immer auf der selben Zelle befindet wie besagter Jäger bzw. Hüter. Der Quaffel bewegt sich dadurch mit dem Jäger bzw. Hüter mit.
Begründung:	Jäger und Hüter können sollen den Quaffel, ohne zu passen, über das Spielfeld transportieren können.
Abhängigkeiten:	Jäger, Hüter, Quaffel
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA25
Titel:	Quaffel Verlieren
Beschreibung:	Ein Jäger kann den Quaffel verlieren, dass bedeutet, das der Quaffel auf ein freie Nachbarzelle zufällig bewegt wird.
Begründung:	Durch das Verlieren des Quaffels werden Tore im Alleingang erschwert.
Abhängigkeiten:	Jäger, Hüter, Quaffel, Fouls, Klatscher
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA26
Titel:	Quaffel Entreißen
Beschreibung:	Ein Jäger darf einem anderen Jäger den Quaffel entreißen, sofern dieser auf einem seiner Nachbarfelder ist. 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:	Bietet zusätzliche Möglichkeiten, den Ball zu erobern.
Abhängigkeiten:	Jäger, Hüter, Quaffel
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA27
Titel:	Tor Erzielen
Beschreibung:	Ein Jäger kann ein Tor erzielen, indem er den Quaffel erfolgreich auf eine Torringzelle wirft. Dabei muss der Schussvektor des Wurfes durch die linke oder rechte Seite der Torringzelle gehen.
Begründung:	Stellt für einen Jäger die Möglichkeit dar, Punkte zu erzielen.
Abhängigkeiten:	Torring, Quaffel Werfen, Jäger
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA28
Titel:	Klatscher Schlagen
Beschreibung:	Ein Treiber kann einen Klatscher schlagen, wenn er sich auf derselben Zelle wie der Klatscher befindet. Der Treiber wählt, um den Klatscher zu schlagen, eine Zielzelle aus, die eine maximale Entfernung von drei zu ihm hat. Zusätzlich müssen auch alle überstrichenen Felder frei sein. Ist dies beides der Fall, so wird das Schlagen des Klatschen wie ein normaler Quaffel-Wurf behandelt, allerdings mit einer 100% Wahrscheinlichkeit.
Begründung:	Stellt für den Treiber die Möglichkeit dar, den Katscher zu bewegen.
Abhängigkeiten:	Spielfeld, Treiber, Klatscher
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA29
Titel:	Spieler Betäuben
Beschreibung:	Bewegt sich der Klatscher auf deine Zelle, auf der sich eine Spielfigur befindet, die kein Treiber ist, wird diese mit einer gewissen Wahrscheinlichkeit betäubt. Das hat zur Folge, dass diese Spielfigur gegebenenfalls den Quaffel verliert, keine Ball fangen und für eine Runde keine Aktion ausführen kann. Der Klatscher, der den Spiler betäubt hat, wird auf eine zufällige freie Zelle gesetzt.
Begründung:	Stellt ein zusätzliches taktisches Spielelement dar.
Abhängigkeiten:	Klatscher, Jäger, Hüter, Sucher
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA30
Titel:	Schnatz Fangen
Beschreibung:	Befindet sich ein Sucher und der Goldene Schnatz auf derselben Zelle, so fängt der Sucher den Goldenen Schnatz mit einer gewissen Wahrscheinlichkeit. Dadurch erhält sein Team 30 Punkte und das Spiel ist zu Ende.
Begründung:	Das Schnatz Fange führ das Ende der Partie herbei.
Abhängigkeiten:	Goldener Schnatz, Sucher, Punkte
Priorität:	++
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA31
Titel:	Fans
Beschreibung:	Ein Fan kann durch Zaubersprüche das eigenen Team unterstützen oder das gegnerische Team schwächen. Diese Eingriffe in das Spielgeschehen werden mit einer gewissen Wahrscheinlichkeit geahndet. Die Fans eines Teams werden jeweils vom Spieler des Teams gesteuert. Es gibt die folgenden Typen von Fans: Elfen, Kobolde, Trolle und Niffler die alle unterschiedliche Fähigkeiten beherrschen.
Begründung:	Sorgen für Abwechslung, Witz, und Überraschungen im Spiel.
Abhängigkeiten:	-
Priorität:	0
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA32
Titel:	Elfen [Fantyp]
Beschreibung:	Elfen haben die Fähigkeit, einen Spieler von der eigenen oder auch von der gegnerischen Mannschaft auf eine zufällige freie Zelle zu Teleportieren.
Begründung:	Siehe Fans
Abhängigkeiten:	Fans
Priorität:	0
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA33
Titel:	Kobolde [Fantyp]
Beschreibung:	Kobolde haben die Fähigkeit, einen Schockzauber zu sprechen, der bewirkt, dass der getroffene Spieler den Quaffel verliert, sofern er diesen hat und anschließend auf eine zufällige freie Nachbarzelle gestoßen wird.
Begründung:	Siehe Fans
Abhängigkeiten:	Fans
Priorität:	0
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA34
Titel:	Trolle [Fantyp]
Beschreibung:	Durch lautes Brüllen sorgen Trolle dafür, dass der Spieler, der den Quaffel besitzt, diesen vor Schreck fallen lässt.
Begründung:	Siehe Fans
Abhängigkeiten:	Fans
Priorität:	0
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA35
Titel:	Niffler [Fantyp]
Beschreibung:	Niffler können nach dem Goldenen Schnatz schnappen können, wodurch dieser eine Ausweichbewegung macht und sich auf eine zufällige freit Nachbarzelle bewegt.
Begründung:	Siehe Fans
Abhängigkeiten:	Fans
Priorität:	0
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA36
Titel:	Schiedsrichter
Beschreibung:	Der Schiedsrichter ahndet mit einer gewissen Wahrscheinlichkeit Fouls von Spielern oder Einmischungen von Fans. Ahndet der Schiedsrichter eine Aktion, so wird die verursachende Spielfigur bis zum nächsten Tor, bzw. der Fan komplett von der Partie ausgeschlossen.
Begründung:	Der Schiedsrichter ist die rechtschaffene Instanz und sorgt dafür, dass unfaire Aktionen nicht bedenkenlos eingesetzt werden können.
Abhängigkeiten:	Spielfigur, Fans
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA37
Titel:	Foul
Beschreibung:	Fouls sind Spielzüge, die grundsätzlich möglich, jedoch laut Regelwerk nicht zulässig sind. Es gibt die folgenden Arten von Fouls: Stürmen, Großoffensive, Rammen, Torringe blockieren und Schnatz blockieren. Der Schiedsrichter ahndet diese Fouls mit einer gewissen Wahrscheinlichkeit.
Begründung:	Fouls stellen eine zusätzliche taktische Spielkomponente dar.
Abhängigkeiten:	Spielfigur, Schiedsrichter, Spielfeld
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA38
Titel:	Torring blockieren [Foul]
Beschreibung:	Eine Spielfigur darf sich nicht direkt auf einen Torring stellen, da es sonst unmöglich wäre, durch diesen Torring ein Tor zu erzielen. Der Ball hopst im Falle eines blockierten Torrings auf eine freie Nachbarzelle des Torrings.
Begründung:	Ermöglicht einem Team zu verhindern, dass der Gegner Punkte erzielen kann.
Abhängigkeiten:	Foul
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA39
Titel:	Stürmen [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 einen Torring zieht.
Begründung:	Stellt eine unfaire Möglichkeit dar, Punkte zu erzielen.
Abhängigkeiten:	Foul
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA40
Titel:	Großoffensive [Foul]
Beschreibung:	Diese Aktion kann nur von Jägern ausgeübt werden. Es bedeutet, dass sich zwei oder mehr Jäger vom selben Team in der gegnerischen Hüterzone befinden.
Begründung:	Gibt den Angreifern einen unfairen Vorteil.
Abhängigkeiten:	Foul
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA41
Titel:	Rammen [Foul]
Beschreibung:	Dieses Foul können alle Spieler ausführen. Hierbei zieht ein Spieler auf das selbe Spielfeld wie ein gegnerischer Spieler, woraufhin dieser den Quaffel ggf. verliert und anschließend auf eine zufällige freie Nachbarzelle verdrängt wird.
Begründung:	Dieses Foul bietet die Möglichkeit, dem Gegner den Ball zu entwenden.
Abhängigkeiten:	Foul
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	**FA42**
Titel:	Schnatz blockieren [Foul]
Beschreibung:	Dieses Foul können alle Spieler ausführen mit Ausnahme der Sucher. Dazu bewegt sich ein Spieler auf das Feld des Goldenen Schnatzes, obwohl er kein Sucher ist, wodurch er verhindert, dass Sucher den Goldenen Schnatz fangen können, da sich nie zwei oder mehr Spielfiguren auf ein und der selben Zelle befinden dürfen.
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:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	**FA43**
Titel:	Setzen auf freies Nachbarfeld
Beschreibung:	Soll ein Spieler oder Ball auf ein zufälliges freies Nachbarfeld von seiner Position gesetzt werden, jedoch sind alle 8 umliegenden Felder bereits von Spielfiguren besetzt oder aus anderen Gründen nicht zulässig, so wird rekursiv von einem zufällig besetzten Nachbarfeld weiter gesucht, bis sich schließlich ein freies Feld findet. Das bedeutet, dass der Spieler oder Ball nicht unbedingt auf einem Nachbarfeld seiner Position landet, sondern auch weiter entfernt positioniert werden kann.
Begründung:	Es muss der Fall abgedeckt werden, dass kein freies Nachbarfeld mehr frei ist.
Abhängigkeiten:	Zelle, Bälle, Spielfigur
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA44
Titel:	Runde
Beschreibung:	Das Spiel läuft in Runden ab. Jede Runde ist dabei in die drei Rundenphasen Ballphase, Spielerphase und Fanphase unterteilt.
Begründung:	Bei „Fantastic Feasts“ handelt es sich laut den Spielregeln um ein Rundenbasiertes Spiel.
Abhängigkeiten:	-
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA45
Titel:	Ballphase
Beschreibung:	In dieser Phase eine Spiels bewegen sich die Bälle über das Spielfeld. Dabei machen die beiden Klatscher ihre Bewegungen in zufälliger Reihenfolge.
Begründung:	sieh Runde
Abhängigkeiten:	Runde
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA46
Titel:	Spielerphase
Beschreibung:	Jeder Spieler macht seinen Zug, wobei sich die Teams abwechseln. In jeder Runde wird neu zufällig bestimmt, welches Team dabei beginnt. Die Reihenfolge der Spieler innerhalb eines Teams wird zufällig bestimmt.
Begründung:	Siehe Runde.
Abhängigkeiten:	Runde
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA47
Titel:	Fanphase
Beschreibung:	Jeder Fan macht seinen Zug, wobei sich die Fanblöcke der Teams abwechseln. In jeder Runde wird neu zufällig bestimmt, welcher Fanblock dabei beginnt. Die Reihenfolge der Fans innerhalb eines Fanblocks wird zufällig bestimmt.
Begründung:	Siehe Runde.
Abhängigkeiten:	Runde
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA48
Titel:	Disqualifikation
Beschreibung:	Werden in der selben Runde mehr als 3 Spielfiguren eines Teams vom Schiedsrichter aus dem Spiel entfernt, gilt das Team als disqualifiziert.
Begründung:	Unfaires Spielen muss bestraft werden.
Abhängigkeiten:	Fouls, Schiedsrichter
Priorität:	0
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA49
Titel:	Spielende
Beschreibung:	Die Partie endet, wenn ein Sucher den Schnatz fängt oder ein Team disqualifiziert wird. Das Team mit den meisten Punkten gewinnt, sofern es nicht disqualifiziert ist.
Begründung:	Damit das Spiel endet.
Abhängigkeiten:	Goldener Schnatz, Sucher, Disqualifikation
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA50
Titel:	Spielfigur platzieren
Beschreibung:	Ein Spieler darf seine Spielfiguren in seiner Hälfte des Spilefeldes beliebig platzieren. Es darf jedoch nie mehr als eine Figur auf der selben Zelle platziert sein. Außerdem muss der Mittelkreis unbesetzt sein.
Begründung:	Zu Beginn des Spiels wählt jeder Spieler ine Aufstellung für sein Team.
Abhängigkeiten:	Mittelkreis, Spielbeginn
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA51
Titel:	Spielbeginn
Beschreibung:	Zu Beginn werden alle Spielfiguren von den Spielern auf dem Spielfeld aber außerhalb des Mittelkreises platziert. Die Bälle, mit Ausnahme des Schnatzes, werden auf der Mittelzelle platziert.
Begründung:	Nötige Aufstellung zu Spielbeginn.
Abhängigkeiten:	-
Priorität:	+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

ID:	FA52
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 bewegt er sich in der nächsten Rundenphase auf die nächste Zelle, auf der sich ein Sucher befindet.
Begründung:	Dadurch wird sichergestellt, dass ein Spiel spannend bleibt und die Spieler nicht die Lust verlieren, wenn eine Partie zu zäh läuft.
Abhängigkeiten:	Runde, Goldener Schnatz, Sucher
Priorität:	-+
Akteure:	Spieler, KI, Gast, Client, KI-Client, Server

4.3 Funktionale Anforderungen: Allgemein

ID:	FA53
Titel:	Partie-Konfiguration
Beschreibung:	Die Partie-Konfiguration spezifiziert sämtliche Wahrscheinlichkeiten für zufällige Ereignisse im Spiel, sowie die maximale Rundenanzahl, bevor die Überlängenbehandlung eintritt.
Begründung:	Bietet dem Nutzer die Möglichkeit, eine Partie nach persönlichen Präferenzen zu gestalten.
Abhängigkeiten:	Quidditchteam-Editor
Priorität:	+
Akteure:	Server, Quidditchteam-Editor, Client, KI-Client, Nutzer

ID:	FA54
Titel:	Quidditchteam-Konfiguration
Beschreibung:	Definiert alle Attribute eines Teams.
Begründung:	Nutzer sollen sich ihre Teams individuell zusammenstellen können.
Abhängigkeiten:	-
Priorität:	+
Akteure:	Server, Quidditchteam-Editor, Client, KI-Client, Nutzer

ID:	FA55
Titel:	Netzwerkschnittstelle
Beschreibung:	Die Clients und Server kommunizieren über eine Netzwerkschnittstelle. Die Clients kommunizieren ausschließlich mit dem Server und nicht untereinander.
Begründung:	Bei „Fantastic Feasts“ 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:	FA56
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 zu ziehen, ist es wichtig das Aktionen protokolliert werden.
Abhängigkeiten:	-
Priorität:	-
Akteure:	Client, KI-Client, Server

4.4 Funktionale Anforderungen: Server

ID:	**FA57**
Titel:	Partie-Konfiguration laden
Beschreibung:	Der Server muss eine vorgefertigte Partie-Konfiguration laden können.
Begründung:	Die Partie-Konfiguration definiert maßgeblich den Spielverlauf.
Abhängigkeiten:	Partie-Konfiguration
Priorität:	-+
Akteure:	Server, Systemadministrator

ID:	**FA58**
Titel:	Zufallsgenerator
Beschreibung:	Pseudozufallszahlengenerator, der bestimmt, ob ein Ereignis eintritt.
Begründung:	Viele Ereignisse im Spiel sind zufällig.
Abhängigkeiten:	-
Priorität:	++
Akteure:	Server

ID:	**FA59**
Titel:	Spielmechanik
Beschreibung:	Der Server interpretiert die Nachrichten der Clients nach den oben definierten Spielregeln und versendet wiederum Updates des Spielzustandes.
Begründung:	Der Server ist die Zentrale Systemkomponente, die die Partie verwaltet.
Abhängigkeiten:	Von den oben definierten Spielregeln.
Priorität:	++
Akteure:	Server

4.5 Funktionale Anforderungen: Client

ID:	**FA60**
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, die Hilfe aufgerufen werden, die Einstellungen der Anwendung angepasst werden und eventuell vorhandene Statistiken aufgerufen werden können.
Abhängigkeiten:	-
Priorität:	+
Akteure:	Nutzer, Client

ID:	**FA61**
Titel:	Spiel beitreten [Ansicht]
Beschreibung:	Grafische Oberfläche, die erscheint, wenn man versucht, sich mit einem Server zu verbinden, um eine neue Partie zu starten. Dabei soll man außerdem die Möglichkeit haben, seine Team-Konfiguration anzugeben, die man für die Partie verwenden möchte.
Begründung:	Der Nutzer muss die Möglichkeit haben, sich komfortabel mit einem Server zu verbinden.
Abhängigkeiten:	Hauptmenü [Ansicht]
Priorität:	+
Akteure:	Nutzer, Client

ID:	FA62
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:	Spiel [Ansicht]
Priorität:	0
Akteure:	Nutzer, Client

ID:	FA63
Titel:	Team-Konfiguration importieren [Ansicht]
Beschreibung:	Grafische Oberfläche zum Importieren 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:	Hauptmenü [Ansicht]
Priorität:	0
Akteure:	Spieler, Client

ID:	FA64
Titel:	Spiel [Ansicht]
Beschreibung:	Grafische Oberfläche, die das Spielgeschehen visualisiert.
Begründung:	Um dem Nutzer das Spiel zu visualisieren.
Abhängigkeiten:	Hauptmenü [Ansicht]
Priorität:	++
Akteure:	Nutzer, Client

ID:	FA65
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.
Begründung:	Um unerfahren Benutzer die Bedienung der Software zu erleichtern.
Abhängigkeiten:	Hauptmenü [Ansicht], Spiel [Ansicht], Beobachter [Ansicht]
Priorität:	0
Akteure:	Nutzer, Client

ID:	FA66
Titel:	Beobachter [Ansicht]
Beschreibung:	Wie die Spiel-Ansicht, nur ohne die Möglichkeit, in das Spiel einzugreifen.
Begründung:	Damit Zuschauer eine Partie mitverfolgen können.
Abhängigkeiten:	Hauptmenü [Ansicht]
Priorität:	0
Akteure:	Gast, Client

ID:	FA67
Titel:	Eingabeverarbeitung
Beschreibung:	Diese Einheit ist für die Verarbeitung von Benutzereingaben verantwortlich.
Begründung:	Jede Benutzereingabe muss ausgewertet werden. Valide Eingaben werden zum Steuern der Anwendung genutzt.
Abhängigkeiten:	Spiellogik
Priorität:	++
Akteure:	Spieler, KI, Client, KI-Client

ID:	FA68
Titel:	Hotkeys
Beschreibung:	Oft benötigte Funktionen werden auf bestimmte (besondere) Tasten (-Kombinationen) abgebildet.
Begründung:	Hotkeys sind optionale Features, die im Lastenheft aufgeführt sind und zu einer einfacheren Spielsteuerung und höherem Spielkomfort betragen können.
Abhängigkeiten:	Spiel [Ansicht], Hilfemenü [Ansicht]
Priorität:	-
Akteure:	Nutzer, Client

ID:	FA69
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:	Spiel [Ansicht], Hotkeys
Priorität:	–
Akteure:	Spieler, Client

4.6 Funktionale Anforderungen: Quidditchteam-Editor

ID:	FA70
Titel:	Team- / Partie-Konfiguration visualisieren
Beschreibung:	Der Editor kann eine geöffnete Team- / Partie-Konfiguration grafisch darstellen und anzeigen, damit ein Nutzer diese bearbeiten kann.
Begründung:	Konfiguration eines Teams / Einer Partie.
Abhängigkeiten:	Quidditchteam-Konfiguration
Priorität:	-
Akteure:	Nutzer, Quidditchteam-Editor

ID:	FA71
Titel:	Team- / Partie-Konfiguration erstellen und speichern
Beschreibung:	Der Editor kann eine Team- / Partie-Konfiguration erstellen und speichern.
Begründung:	Konfiguration eines Teams / Einer Partie.
Abhängigkeiten:	Quidditchteam-Konfiguration
Priorität:	-
Akteure:	Nutzer, Quidditchteam-Editor

ID:	FA72
Titel:	Team- / Partie-Konfiguration bearbeiten
Beschreibung:	Der Editor kann eine bestehende Team- / Partie-Konfiguration öffnen und bearbeiten.
Begründung:	Konfiguration eines Teams / Einer Partie.
Abhängigkeiten:	Quidditchteam-Konfiguration
Priorität:	-
Akteure:	Nutzer, Quidditchteam-Editor

4.7 Funktionale Anforderungen: KI-Client

ID:	FA73
Titel:	Schwierigkeitsgrad einstellen
Beschreibung:	In der KI-Clientanwendung hat ein Nutzer die Möglichlkeit, die Spielstärke der KI einzustellen.
Begründung:	Um dem Nutzer unterschiedlich starke KI-Gegner zur Verfügung zu stellen.
Abhängigkeiten:	-
Priorität:	-
Akteure:	Nutzer, KI

ID:	FA74
Titel:	Serverkonfiguration einstellen
Beschreibung:	In der KI-Clientanwendung kann ein Nutzer einstellen, mit welchem Server sich der KI-Client verbinden soll.
Begründung:	Damit sich der KI-Client mit dem gewünschten Server verbinden kann.
Abhängigkeiten:	-
Priorität:	-
Akteure:	Nutzer, KI

ID:	FA75
Titel:	Team-Konfiguration laden
Beschreibung:	Der Nutzer kann der KI eine gewünschte Team-Konfiguration zuweisen, indem er eine bereits erstellte Konfiguration läd.
Begründung:	Um der KI ein gewünschte Team-Konfiguration zuzuweisen.
Abhängigkeiten:	Team-Konfiguration
Priorität:	-
Akteure:	Nutzer, KI

4.8 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, Docker-Container
Priorität:	+
Akteure:	Nutzer, Entwickler

ID:	QA2
Titel:	Version-Controlling
Beschreibung:	Beim Verwalten des Quellcodes soll ein Git basiertes Version-Controlling Werkzeug (*GitHub / GitLab*) 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:	Jeder gepushte Commit soll automatisch mit Hilfe der CI Unit-Tests und der Statischen Codeanalyse unterzogen werden. Zudem soll eine automatisierte Code Dokumentation angestoßen werden. Bei erfolgreichem Abschließen aller Tests 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:	Version-Controlling
Priorität:	0
Akteure:	Entwickler

ID:	QA4
Titel:	Statische Codeanalyse
Beschreibung:	Mit Hilfe des Tools 'SonarQube' bzw. 'SonarCloud' soll der gesamt 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:	Alle definierten Unit-Tests müssen fehlerfrei bestanden werden.
Begründung:	Da alle Komponenten möglichst fehlerfrei funktionieren müssen ist es unerlässlich die einzelnen Teil der Software ständig auf ihre Funktionalität zu prüfen.
Abhängigkeiten:	-
Priorität:	0
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 compiliert 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 sind komplexe Algorithmen detailliert zu dokumentieren. Die gesamte Dokumentation 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:	Entwickler

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 erleichtert die Bedienung der Anwendung.
Abhängigkeiten:	Dokumentation
Priorität:	0
Akteure:	Nutzer, Entwickler

ID:	QA9
Titel:	Anwendungssprache
Beschreibung:	Das User-Interface der Anwendungen soll in deutscher Sprache gestaltet werden.
Begründung:	Die Zielkundschaft spricht überwiegend Deutsch.
Abhängigkeiten:	-
Priorität:	0
Akteure:	Nutzer, Entwickler

ID:	QA10
Titel:	Implementierungssprache
Beschreibung:	Die Implementierung der Anwendungen soll in englischer Sprache gehalten sein.
Begründung:	Die Implementierungssprache ist im Lastenheft vorgegeben.
Abhängigkeiten:	-
Priorität:	0
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:	0
Akteure:	Entwickler, Kunde

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 den *JSON* Standard erfüllen Des Weiteren sind alle vom Komitee festgelegten weiteren Standards einzuhalten.
Begründung:	Durch Einheitliche Formate der Konfigurationsdateien lässt sich sicherstellen, das einzelne Komponenten zwischen den Entwicklungsteams ausgetauscht werden können und diese miteinander kompatibel sind.
Abhängigkeiten:	-
Priorität:	+
Akteure:	Entwickler, Nutzer

ID:	**QA14**
Titel:	Netzwerkkommunikation
Beschreibung:	Die Netzwerkkommunikation zwischen Client und Server soll über sogenannte *Web-Socket-Sessions* realisiert werden, sodass Client und Server ortsunabhängig von einander betrieben werden können. Die Nachrichten sollen im *JSON* formatiert sein.
Begründung:	Die Netzwerkkommunikation muss gewisse Standards erfüllen, damit Client- und Serveranwendungen von unterschiedlichen Entwicklerteams mit einander kompatibel sind.
Abhängigkeiten:	-
Priorität:	++
Akteure:	Server, Client, KI-Client

ID:	**QA15**
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, Entwickler

ID:	QA16
Titel:	Zuverlässigkeit
Beschreibung:	Bei 100 Spielen darf maximal eine Partie aufgrund eines Fehlers in der Anwendung abgebrochen werden müssen.
Begründung:	Durch zu häufige Ausfälle der Software ist das Benutzererlebnis massiv beeinträchtigt.
Abhängigkeiten:	Robustheit
Priorität:	+
Akteure:	Nutzer, Entwickler

ID:	QA17
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, Entwickler

ID:	QA18
Titel:	Benutzerfreundlichkeit
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:	QA19
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:	QA20
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 Zielsystemen problemlos nutzbar.
Abhängigkeiten:	-
Priorität:	-
Akteure:	Nutzer, Entwickler

ID:	QA21
Titel:	Kurze Ladezeiten
Beschreibung:	Systembedingte Ladezeiten der Software dürfen auf einem aktuellen Computer eine Sekunde pro geladener Ansicht nicht überschreiten.
Begründung:	Bei längeren Ladezeiten ist das Benutzererlebnis massiv beeinträchtigt.
Abhängigkeiten:	-
Priorität:	-+
Akteure:	Nutzer, Entwickler