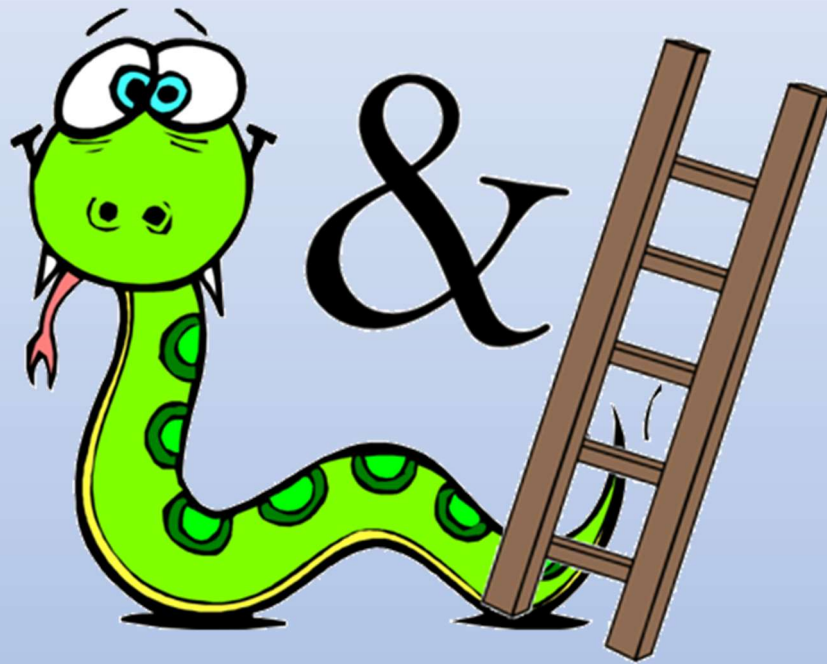


PSIT3 – Snakes and Ladders

Analyse-Resultate



Namen:	Milan Natkeeran Severin Roost Jan Sonderegger Dominik Steiner Philipp Wetzel
Klasse:	IT16a_WIN
Gruppe:	3
Schule:	ZHAW
Abgabetermin:	24.10.2017

Inhaltsverzeichnis

Versionierung	3
1. Projektmanagement	4
1.1. Grobplanung	4
1.2. Risikoliste	4
1.3. Vergangene Iterationen	5
1.3.1. Iterationsplan #2	5
1.3.2. Iterationsplan #3	5
1.4. Nächste Iterationen	5
1.4.1. Iterationsplan #4	5
2. Anwendungsfälle	6
2.1. Use Case 1: Spiel Umgebung starten	6
2.2. Use Case 2: Spiel beitreten	8
2.3. Use Case 3: Frageliste einlesen	10
2.4. Use Case 4: Spiel spielen	11
2.4.1. Use Case 4.1: Spiel spielen - Felder	13
3. Anwendungsfalldiagramm	15
4. Domänenmodell	15
5. Eine erste Architektur	16
6. Zusätzliche Spezifikationen	17
6.1. Funktionale Anforderungen:	17
6.2. Nicht funktionale Anforderungen	17
6.3. Spielregeln	18
7. System-Sequenzdiagramm	19
7.1. Use Case 1: Spiel Umgebung starten	19
7.2. Use Case 2: Spiel beitreten	19
7.3. Use Case 3: Frageliste einlesen	20
7.4. Use Case 4: Spiel spielen	20
7.4.1. Use Case 4.1: Spiel spielen - Felder	21
8. Glossar	22

Versionierung

Version	Datum	Bearbeitet von	Änderung / Grund
1	03.10.2017	Milan	Analyse-Resultate Dokument erstellt
1.01	05.10.2017	Philipp	Kapitel gemäss PSIT3-Resultate eingefügt
1.01	06.10.2017	Philipp	Struktur für Anwendungsfälle vorbereitet.
1.02	11.10.2017	Philipp	Anwendungsfalldiagramm + System-Sequenzdiagramm für UC 4 eingefügt
1.03	12.10.2017	Alle	Glossar begonnen
1.04	15.10.2017	Milan	Anwendungsfalldiagramm + System-Sequenzdiagramm für UC 1 eingefügt
1.05	15.10.2017	Severin	Anwendungsfalldiagramm + System-Sequenzdiagramm für UC 4.1 eingefügt
1.06	18.10.2017	Dominik	Anwendungsfalldiagramm + System-Sequenzdiagramm für UC 3 eingefügt
1.07	21.10.2017	Jan	Anwendungsfalldiagramm + System-Sequenzdiagramm für UC2 eingefügt
1.08	21.10.2017	Philipp	UML Packet Diagramm erstellt
1.09	21.10.2017	Severin	Anforderungen gemäss FURPS analysiert und umschrieben
1.10	22.10.2017	Milan	Spielregeln definiert
1.11	23.10.2017	Philipp	Einfügen der Projektmanagement-Informationen
1.12	23.10.2017	Philipp / Milan	Korrektur der Struktur und Rechtschreibung

Tabelle 1: Versionierung

1. Projektmanagement

1.1. Grobplanung

Iteration	Semester	Woche	Start	Ende	Anz Tage	Milestone	Ziele, Release	Aufwand geplant [h]	Aufwand aus Iter # [h]	Aufwand effektiv [h]	geplant vs effektiv
			25.09.2017	02.10.2017	7	1	Inception Phase	20	22.5	24.3	21.25%
1	2		25.09.2017	02.10.2017	7	1	- Projektthema, Idee, Funktionalität, Umfang im Projektteam definieren sodass allen die Vision klar ist.	20	22.5	24.3	21.25%
			03.10.2017	23.10.2017	20	2	Elaboration Phase	110	105.0	106.8	-2.95%
2	3,4		03.10.2017	13.10.2017	10	2	Absprache und klare definition des Ablaufs des Spiels (Anwendungsfälle)	55	50.0	55.8	1.36%
3	4,5		13.10.2017	23.10.2017	10	2	Kommunikation zwischen Client und Server herstellen Spielregeln definieren	55	55.0	51.0	-7.27%
			23.10.2017	20.11.2017	28	3	Construction Phase	160	94.0	0.0	-100.00%
4	6,7		23.10.2017	06.11.2017	14	3	- Klassendiagramme für die Logik, Status, den Befehlen und des GUI erstellen - Erstellen der Klassen und Strukturen gemäss Klassendiagrammen	80	80.0	0.0	-100.00%
5	8,9		06.11.2017	20.11.2017	14	3	- Implementierung der Logik, Befehlen und fertigstellung des GUI in den vorbereiteten Klassen.	80	14.0	0.0	-100.00%
			21.11.2017	18.12.2017	27	4	Transiton Phase	155	26.0	0.0	-100.00%
6	10, 11		21.11.2017	03.12.2017	12	4	- Tests mit einer Testgruppe durchführen - Erkenntnisse aus Testgruppe einfließen lassen - BETA-Version erstellen	70	13.0	0.0	-100.00%
7	12,13		03.12.2017	18.12.2017	15	4	- Schlussbericht verfassen (Projektidee, Analyse, Design, Implementation, Resultat und Anhang) - Abschluss Projekt	85	13.0	0.0	-100.00%
(Pro Kopf 18h - 2 Tage) Reserve:								90			
Total:								535	247.5	131.0	-75.51%
Vorgaben:								500-600h			

1.2. Risikoliste

Nr.	Name	Bschreibung	Wahrsch.	Schaden	Stufe	Prio	Massnahmen	Verantwortlich
1	Firewall	Server-Client-Kommunikation funktioniert nicht via Firewall	40%	Spiel ist nicht funktionsfähig und nicht brauchbar	M		1 Berechtigung im Netzwerk für das Spiel vergeben	Entwicklungsteam
2	Effizienz	Geforderte Antwortzeiten können nicht erreicht werden.	30%	Spiel ist nur bedingt spielbar	M		1 Mögliche Verbesserung der Netzwerkkommunikation in betracht ziehen --> Prototyp entwickeln	Entwicklungsteam
3	Konkurrenz	Es erscheint ein Produkt auf dem Markt welches unser Produkt obsolet aussehen lässt.	30%	Spiel lässt sich nicht mehr oder nur schlecht verkaufen	H		2 Projektskizze überarbeiten und Massgebende unterschiede definieren.	PL
4	Personell	Es fällt ein Mitarbeiter des Teams aus.	10%	Wissen über das Projekt und Ideen gehen verloren.	L		3 Verteilung der Aufgaben auf andere Entwickler, Reduktion des Projektumfanges.	PL

1.3. Vergangene Iterationen

1.3.1. Iterationsplan #2

Task	Arbeitspaket	Aufwand geplant [h]	Aufwand effektiv [h]	Verantwortlich	Erledigt ?
1	Projektmanagement	4	5	Philipp	x
2	Anwendungsfälle umschreiben	5	6	alle	x
3	Anwendungsfalldiagramm zeichnen	2.5	2	Philipp	x
4	Balsamiq Mockups Entwurf erstellen	1.5	3	Severin / Philipp	x
5	System-Sequenzdiagramm	6	8.5	alle	x
6	Glossar	1	0.25	alle	x
7	Domänenmodell	5	4.5	Philipp	x
8	Meeting / Aussprache / Diskussion	25	26.5	alle	x
9					
Total:		50	55.75		

1.3.2. Iterationsplan #3

Task	Arbeitspaket	Aufwand geplant [h]	Aufwand effektiv [h]	Verantwortlich	Erledigt ?
1	Projektmanagement	7	8	Philipp	x
2	Zusätzliche Spezifikationen / Spielregeln	3	3.5	Severin / Milan	x
3	Umsetzung der ersten Architektur	4	2.5	Jan	x
4	Glossar ergänzen	2	1.5	alle	x
5	Erste Connection zwischen Server und Client implementieren	10	6	Jan	x
6	UC Frageliste einlesen	6	4	Dominik	x
7	Präsentation vorbereiten	4	2	Philipp / Dominik	x
8	Rechtschreibkorrektur & Struktur prüfen im Dokument	2	4	Philipp / Milan	x
9	Eine erste Architektur entwerfen	2	3	alle	x
10	Meeting / Aussprache / Diskussion	15	16.5	alle	x
11					
Total:		55	51		

1.4. Nächste Iterationen

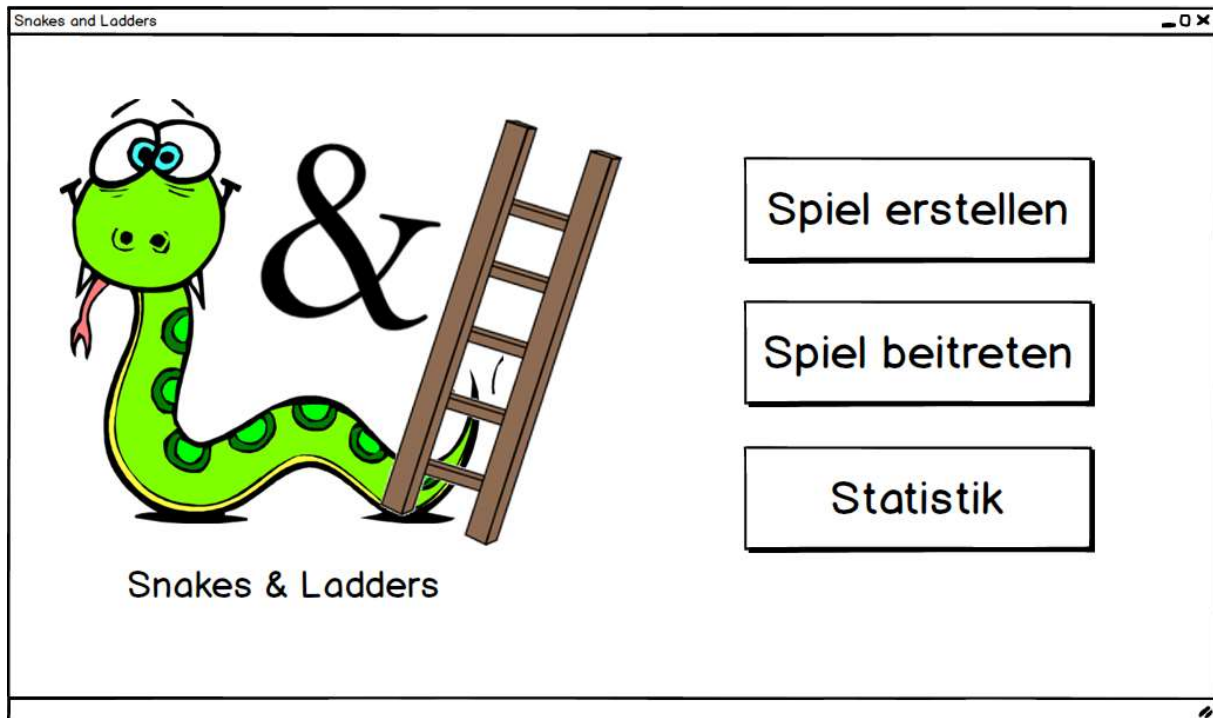
1.4.1. Iterationsplan #4

Task	Arbeitspaket	Aufwand geplant [h]	Aufwand effektiv [h]	Verantwortlich	Erledigt ?
1	Architektur	4		Milan	
2	Design-Klassendiagramm	4		Philipp	
3	Klassenverantwortlichkeiten	4		Jan	
4	Interaktionsdiagramme	4		Severin	
5	Glossar weiter ausbauen	3		alle	
6	Klassendiagramm für Logik-Aktionen erstellen	6		Jan / Dominik	
7	Klassendiagramm für Status erstellen	6		Severin / Milan	
8	Klassendiagramm für Befehle erstellen	6		Dominik / Severin	
9	Klassendiagramm für GUI erstellen	6		Philipp / Milan	
10	Klassen gemäss Klassendiagramm implementieren	25		alle	
11	Projektmanagement	4		Philipp	
12	Meeting / Aussprache / Diskussion	8		alle	
13					
Total:		80	0		

2. Anwendungsfälle

2.1. Use Case 1: Spiel Umgebung starten

Umfang:	Snakes and Ladders	
Ebene:	Anwenderziel	
Primärakteur:	Administrator	
Stakeholder und Interessen:	<p>Spieler: Möchten das Spiel so unkompliziert wie möglich starten.</p> <p>Administrator: Das Spiel den Voraussetzungen entsprechend einrichten und das Spiel bereitstellen zum Verbinden.</p>	
Vorbedingungen:	<p>Entscheidung zwischen klassischem oder erweitertem Modus.</p> <p>Beim Erweiterten Modus können eigene Fragen für das Quiz verwendet werden. Die Fragenliste muss vorher vorbereitet werden. Eine Vorlage für die Fragenliste kann von der Applikation entnommen werden, um diese dann zu bearbeiten und auch hochzuladen.</p>	
Nachbedingungen:	Applikation nach dem Spielen beenden (Verbindung sicher trennen).	
Standardablauf:	<p>Klassisches Spiel-Modus:</p> <ol style="list-style-type: none"> 1. Administrator gibt für das Spiel einen Namen ein. 2. Administrator gibt für das Spiel ein Passwort ein (Passwort ist optional). 3. Administrator tippt die Spieleranzahl ein. 4. Administrator wählt den klassischen Spiel-Modus aus (Standard Spiel-Modus). 5. Administrator stellt den Server für die Verbindung bereit bzw. erstellt die Spiel Umgebung. 	<p>Erweitertes Spiel-Modus:</p> <ol style="list-style-type: none"> 1. Administrator gibt für das Spiel einen Namen ein. 2. Administrator gibt für das Spiel ein Passwort ein (Passwort ist optional). 3. Administrator tippt die Spieleranzahl ein. 4. Administrator wählt den erweiterten Spiel-Modus aus. 5. Administrator liest die vorbereiteten Fragen ein. 6. Administrator stellt den Server für die Verbindung bereit bzw. erstellt die Spiel Umgebung.
Erweiterungen:	a) Statt im erweiterten Spiel-Modus die Frageliste selber zu vorbereiten können bereits zur Verfügung gestellte Fragelisten ausgewählt werden.	
Spezielle Anforderungen:	<p>Die Fragenliste muss der Vorlage entsprechen, damit das Spiel erfolgreich gestartet werden kann.</p> <p>Netzwerkverbindung muss gewährleistet sein.</p>	
Liste der Technik- und Datenvariationen:	a) Computer wird für die Server-Applikation benötigt.	
Häufigkeit des Auftretens:	Einmalig (bevor das Spiel gestartet wird)	
Verschiedenes (z.B. Offene Fragen):	Keine	



The screenshot shows a web application window titled "Snakes and Ladders - Server". The main heading is "Spiel erstellen". Below the heading, there are several input fields and controls:

- Name:** A text input field.
- Passwort:** A text input field, followed by the text "(optional)".
- Spieleranzahl:** A text input field.
- Erweitertes Spiel-Modus:** A toggle switch, currently turned on (green).
- Frageliste:** A dropdown menu showing "Frageliste 1" and a button labeled "Eigene Frageliste importieren".
- A button labeled "Vorlage herunterladen".
- A large button labeled "Erstellen" at the bottom.

2.2. Use Case 2: Spiel beitreten

Umfang:	Snakes & Ladders Software
Ebene:	Benutzer-Funktion
Primärakteur:	Spieler
Stakeholder und Interessen:	Ein Spieler möchte an einem Spiel teilnehmen und sich auf dem Server registrieren. Der Administrator hat den Server aufgesetzt und wartet auf Spieler, welche dem Server beitreten möchten.
Vorbedingungen:	Spiel-Server ist konfiguriert und erreichbar
Nachbedingungen:	Spieler nimmt an einem Spiel teil, sieht Spielfeld und Gegenspieler. Das Spiel kann gestartet werden.
Standardablauf:	<ul style="list-style-type: none"> - Spieler öffnet GUI und gelangt zum Menü - Spieler wählt Knopf «Server beitreten» - Spieler kann aus einer Liste den gewünschten Server auswählen - Spieler schickt eine Anfrage an den Server-Administrator - Server-Administrator nimmt den Spieler an - Spieler ist auf dem Server registriert und kann am Spiel nun teilnehmen
Erweiterungen:	<p>a)</p> <ul style="list-style-type: none"> - der Admin nimmt den Spieler nicht auf dem Spiel-Server an - dem Spieler wird eine entsprechende Meldung angezeigt - der Spieler kann weitere Anfragen starten <p>b)</p> <ul style="list-style-type: none"> - Der Server ist nicht erreichbar - Es erscheint eine Fehlermeldung - der Spieler kann weitere Anfragen starten
Spezielle Anforderungen:	Ein Spiel-Server ist gestartet Es muss eine Internet-Verbindung bestehen
Liste der Technik- und Datenvariationen:	
Häufigkeit des Auftretens:	Pro Spieler einmal, sofern die Verbindung korrekt hergestellt wird. Im Falle eines Fehlers pro Spieler mehrfach.
Verschiedenes (z.B. Offene Fragen):	<ul style="list-style-type: none"> - Muss ein Passwort eingegeben werden um einem Server beizutreten? - Wie wird die Liste der verfügbaren Server erstellt bzw. wie werden die Server identifiziert?

Snakes and Ladders - Client

Spiel beitreten

Spielname	Spieleranzahl
Spiel1	2
Spiel2	1
Spiel3	4

Suchen nach Spielname

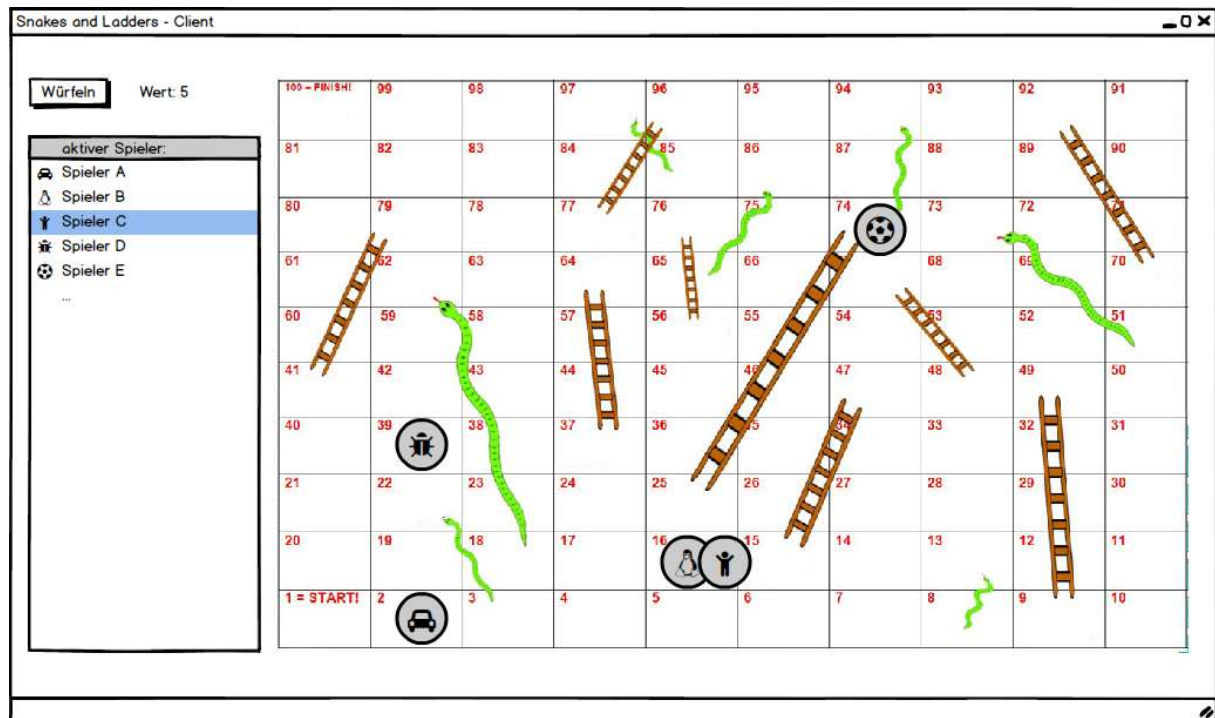
Spiel beitreten

2.3. Use Case 3: Frageliste einlesen

Umfang:	Server Applikation
Ebene:	Anwenderziel
Primärakteur:	Administrator
Stakeholder und Interessen:	Administrator: Will schnell und ohne Probleme seine Frageliste in das System laden Spieler: Möchte das Spiel spielen
Vorbedingungen:	Das Spiel ist noch nicht gestartet und der Server ist noch in der Konfigurationsphase.
Nachbedingungen:	Die eingelesenen Fragen und Antworten sind im Server für das nächste Spiel zwischengespeichert.
Standardablauf:	<ol style="list-style-type: none"> 1. Der Administrator begibt sich im Server zur Ansicht, bei dem er die Frageliste einlesen kann. 2. Administrator wählt die Option um eine neue Frageliste hochzuladen. 3. Administrator wählt im eigenem Dateisystem die Datei mit der Frageliste aus und lädt sie hoch 4. Das System speichert die Frageliste für das nächste Spiel 5. Administrator bestätigt die neuen Änderungen
Erweiterungen:	<p>*a. Administrator kann auswählen, welche Fragen nicht im Spiel sein sollten.</p> <p>*b. Administrator kann das Format für die Frageliste-Datei selber definieren.</p> <p>*c. In den Fragen kann man auch Bilder anzeigen lassen.</p> <p>3a. Ungültiges Format bei der Frageliste</p> <ol style="list-style-type: none"> 1. System signalisiert den Fehler und ignoriert die Frageliste 2. Der Benutzer hat die Möglichkeit, nochmals eine Frageliste hochzuladen, falls er das möchte. <p>3b. Frageliste wird doppelt hochgeladen</p> <ol style="list-style-type: none"> 1. System signalisiert, dass die hochgeladene Frageliste schon im System ist. 2. Der Benutzer hat die Möglichkeit eine andere Frageliste hochzuladen, falls er das möchte.
Spezielle Anforderungen:	<ul style="list-style-type: none"> • auch grosse Fragelisten sollten mühelos benutzbar sein. • Die Frageliste soll auch spezielle Symbole (z.B. mathematische) unterstützen
Liste der Technik- und Datenvariationen:	2a. Eingabe der Fragen und Antworten per Tastatur direkt im System.
Häufigkeit des Auftretens:	oft – maximal ein Mal pro Spiel
Verschiedenes (z.B. Offene Fragen):	Was passiert wenn das System des Administrators während der Konfiguration abstürzt?

2.4. Use Case 4: Spiel spielen

Umfang:	Snakes and Ladders	
Ebene:	Anwenderziel	
Primärakteur:	Spieler	
Stakeholder und Interessen:	Spieler: Will das Spiel spielen und als erstes das letzte Feld erreichen.	
Vorbedingungen:	Spieler konnte einem Spiel beitreten. Das Spiel wurde gestartet.	
Nachbedingungen:	Beantwortete Fragen werden im Spielstand des Spielers gespeichert.	
Standardablauf:	1. Spieler würfelt (virtuell) (widerholend)	
		2. Applikation fährt mit der Spielerfigur die gewürfelte Anzahl Felder weiter.
	Bei einem Fragefeld:	
		1. Die Applikation wählt eine Frage aus und stellt diese dem Spieler.
	2. Spieler beantwortet Frage	
		3. Die Applikation prüft die Antwort und gibt Feedback dem Spieler
	4. Spieler sieht Antwort	
		5. Bei Richtiger Antwort: 5.1. Spielfigur bleibt auf dem Feld 6. Bei falscher Antwort: 6.1. Spielfigur wird zurückgesetzt
	Bei einer Leiter: Siehe «Use Case 4.1 Spiel spielen – Felder»	
	Bei einer Schlange: Siehe «Use Case 4.1 Spiel spielen – Felder»	
	Beim letzten Feld: Siehe «Use Case 4.1 Spiel spielen – Felder»	
Erweiterungen:	*A. Spieler gibt Würfelzahl per Tastatur ein. *B. Jederzeit, wenn das System des Spielers ausfällt: 1. Spieler verbindet sich neu 2. Standardablauf	
Spezielle Anforderungen:	Synchronisation der Spielstände an die Spieler muss in 0.5s erfolgen.	
Liste der Technik- und Datenvariationen:	*A. Die Eingabe des Wertes des Würfels erfolgt über die eingeblendete oder angeschlossene Tastatur.	
Häufigkeit des Auftretens:	Beinahe laufend.	
Verschiedenes (z.B. Offene Fragen):	Wie ist der Ablauf für den Spieler, wenn das System auf den Administratoren Seite ausfällt?	



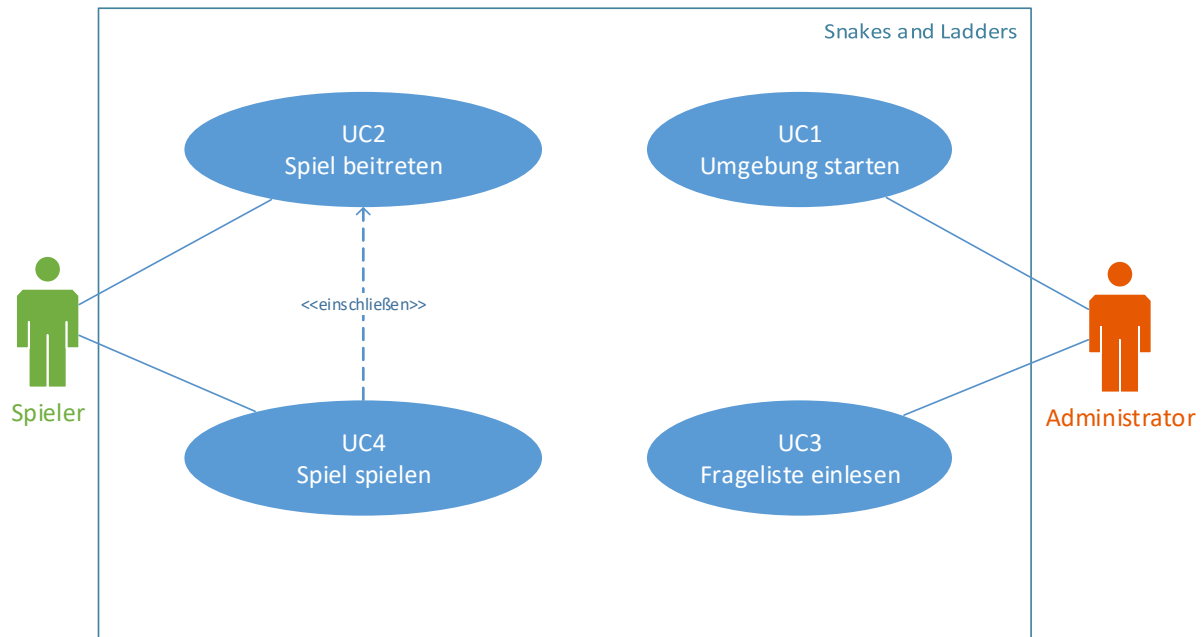
2.4.1. Use Case 4.1: Spiel spielen - Felder

Umfang:	Snakes & Ladders	
Ebene:	Subfunktion	
Primärakteur:	Spieler, Serverapplikation	
Stakeholder und Interessen:	Spieler und Administrator wollen, dass das Spiel den definierten Regeln entsprechend abläuft.	
Vorbedingungen:	Spiel läuft. Spieler zieht auf ein Feld. (vgl. UC4 Spiel spielen)	
Nachbedingungen:	Neue Position des Spielers wurde gespeichert und wird ihm korrekt auf dem Spielfeld visuell dargestellt.	
Standardablauf:	<i>Bei einem normalen Feld</i>	
	1. Spieler sieht neue Position seiner Figur	
	<i>Bei einer Leiter</i>	
		1. Applikation wählt eine Frage aus und stellt sie dem Spieler
	2. Spieler gibt Antwort auf die Frage	
		3. Applikation wertet Antwort aus und zeigt sie dem Spieler
		4.1 Bei richtiger Antwort bewegt sich die Spielfigur des Spielers ans Ende der Leiter. 4.2 Bei falscher Antwort bleibt die Spielfigur stehen.
	5. Spieler sieht Antwort zur Frage und neue Position seiner Figur.	
	<i>Bei einer Schlange</i>	
		1. Applikation wählt eine Frage aus und stellt sie dem Spieler
	2. Spieler gibt Antwort auf die Frage	
		3. Applikation wertet Antwort aus und zeigt sie dem Spieler
		4.1 Bei richtiger Antwort bleibt die Spielfigur stehen. 4.2 Bei falscher Antwort bewegt sich die Spielfigur ans Ende der Schlange.
	5. Spieler sieht Antwort zur Frage und neue Position seiner Figur.	

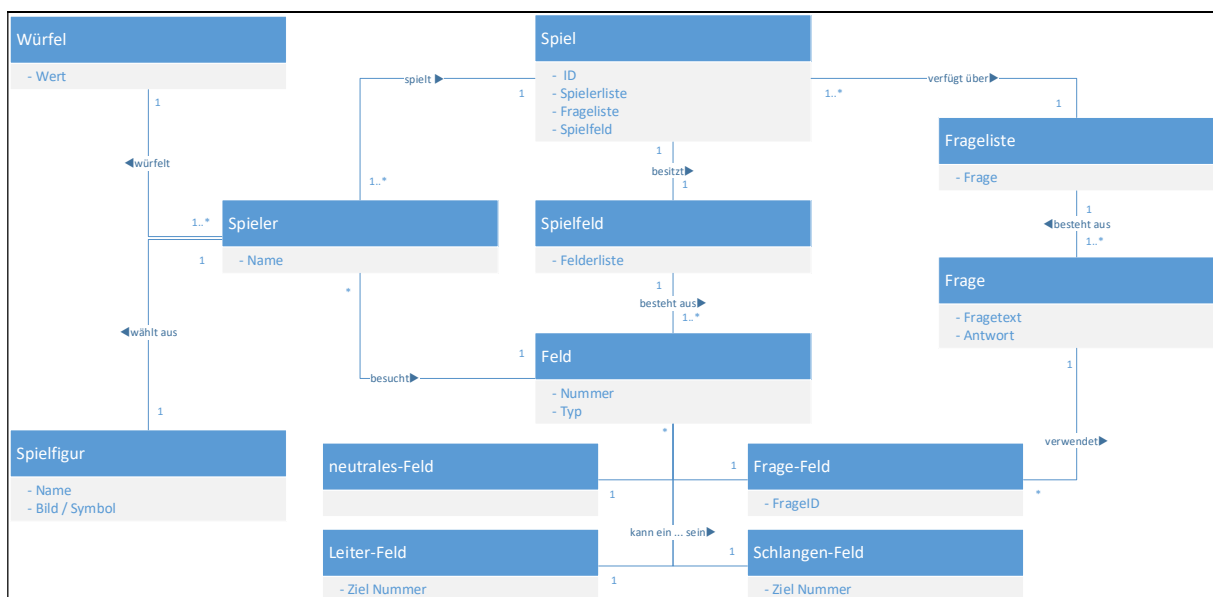
	<i>Beim letzten Feld</i>	
		1. Applikation wählt eine Frage aus und stellt sie dem Spieler.
	2. Spieler gibt Antwort auf die Frage.	
		3. Applikation wertet Antwort aus und zeigt sie dem Spieler.
		4.1 Bei richtiger Antwort bleibt die Spielfigur stehen und das Spiel ist beendet für besagten Spieler. 4.2 Bei falscher Antwort wird die Spielfigur zurückgesetzt.
	5. Spieler sieht Antwort zur Frage, neue Position seiner Figur sowie eine Siegesnachricht (falls Frage richtig beantwortet wurde).	
Erweiterungen:	Alternativ (falls Server entsprechend konfiguriert) kann auch ohne Fragen gespielt werden. Indem Fall werden die Schritte 1. - 3. übersprungen und stattdessen wird bei 4.1 (Leiter / letztes Feld) bzw. 4.2 (Schlange) fortgesetzt.	
Spezielle Anforderungen:	Kommunikation/Synchronisation von Server und Clients soll nicht länger als 0.5 Sekunden dauern.	
Liste der Technik- und Datenvariationen:	-	
Häufigkeit des Auftretens:	Mehrmals während eines Spiels	
Verschiedenes (z.B. Offene Fragen):	Beim letzten Feld eine höhere Zahl gewürfelt wird?	



3. Anwendungsfalldiagramm



4. Domänenmodell

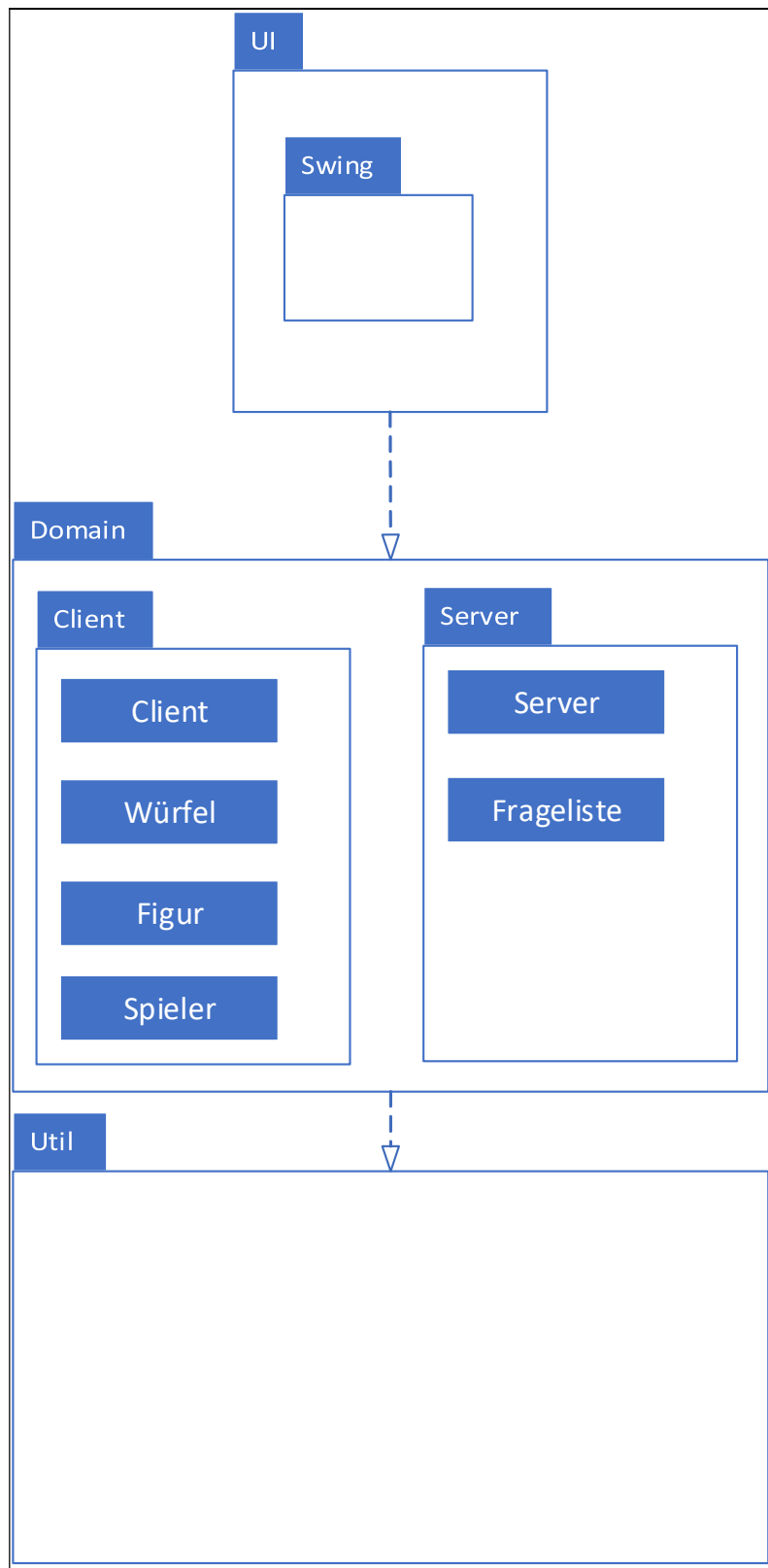


Ein Spieler kann eine Spielfigur auswählen und einen Würfel würfeln, die von allen Spielern geteilt wird, d.h. alle Spieler haben zusammen für das Spiel ein Würfel.

Im Spiel gibt es ein Spielfeld, das eine bestimmte Anzahl Felder hat. Diese Felder können vier verschiedene Typen haben. Entweder ist das Feld ein neutrales-Feld, Frage-Feld, Leiter-Feld oder ein Schlangen-Feld. Bei den zwei Felder Leiter und Schlangen-Feld gibt es ein Attribut mit Ziel Nummer, da die Spielfigur von diesem Feld zu einem anderen Feld (Ziel Feld) positioniert wird.

Das Frage-Feld verwendet die Fragen von der Frageliste, welche das Spiel zur Verfügung hat.

5. Eine erste Architektur



6. Zusätzliche Spezifikationen

6.1. Funktionale Anforderungen:

- Sämtliche Antworten der Fragen müssen aufgezeichnet werden.

6.2. Nicht funktionale Anforderungen

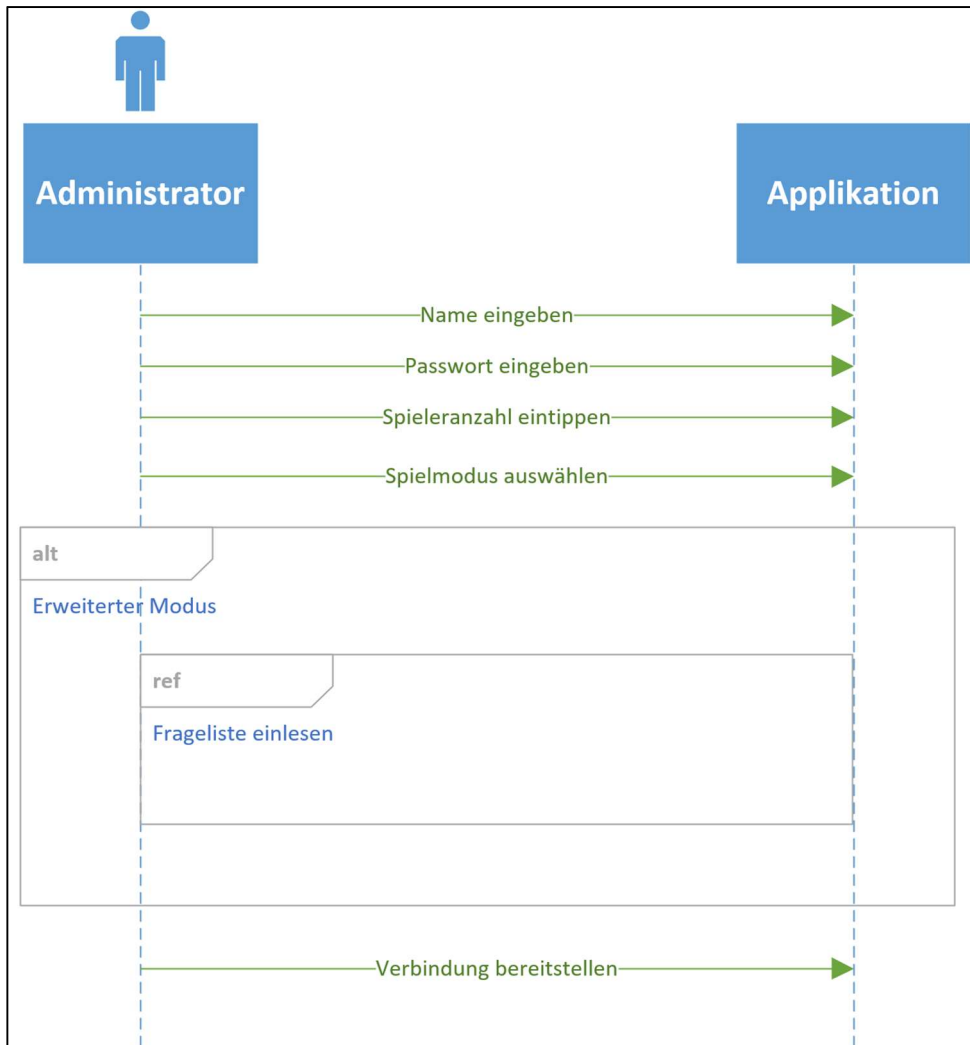
Functionality:	<ul style="list-style-type: none">- Das Spielfeld muss so designt werden, dass pro Spiel eine gewisse mindeste Anzahl an Fragen gestellt werden, damit dessen Funktion als Lernspiel garantiert werden kann.- Antwort auf Fragen werden auf Inhalt, nicht Rechtschreibung geprüft. (case-insensitive)- Beim Beantworten der Fragen gibt eine Möglichkeit, einen Hinweis auf die Antwort anzuzeigen.
Usability:	<ul style="list-style-type: none">- Einfache Bedienung da diese unter «Beobachtung» erfolgt.- Neue Benutzer müssen mit maximal 30 Minuten Aufwand rechnen, um alle Funktionen der Applikation verstehen und nutzen zu können.
Reliability:	<ul style="list-style-type: none">- Die Applikation soll jederzeit funktionsfähig und einsatzbereit sein.
Performance:	<ul style="list-style-type: none">- Die Netzwerkverzögerung während einem Multiplayer-Spiel darf nicht mehr als 0.5 Sekunden betragen.- Am Ende eines Spieles wird der Lernfortschritt jedes einzelnen Spielers in 99.95% der Fälle gespeichert.
Supportability:	<ul style="list-style-type: none">- Excel Import Format kann ändern- Teile des Programms sollten auch bei geänderten Abläufen genutzt werden können (Schnittstellen der Kommunikation zwischen Client und Server)

6.3. Spielregeln

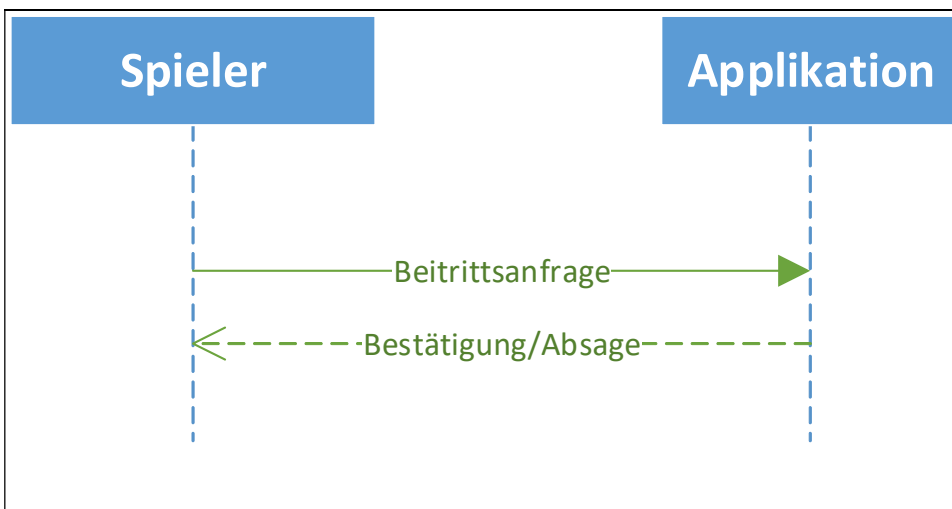
- Es wird zufällig gewählt welcher Spieler beginnt.
- Ist ein Spieler am Zug, kann er die Aktion wählen, einen (virtuellen) Würfel zu werfen. Anschliessend bewegt sich seine Spielfigur entsprechend der gewürfelten Zahl vorwärts.
- Bleibt die Figur auf einem Quiz-Feld stehen, wird eine Frage aus der hinterlegten Frageliste gewählt, auf welche der Spieler (per Tastatureingabe) eine Antwort eingeben kann.
- Bei einer falschen Antwort wird die Figur der Anzahl Felder, die er nach vorne bewegt hat zurückbewegt.
- Bleibt die Figur auf einem Leiter- oder Schlangen-Feld stehen, wird die Spielfigur zum Zielfeld der Leiter-oder Schlangen-Feld bewegt.
- Das Spielfeld, die Figuren und wer an der Reihe ist, können die Spieler jederzeit der visuellen Darstellung entnehmen.
- Der Spielvorgang wird wiederholt und es wird angezeigt, welcher Spieler zurzeit am Zug ist.
- Das Spiel endet im Normalfall, sobald ein Spieler das Zielfeld erreicht hat, jedoch können je nachdem wie viele Spieler spielen auch weitergespielt werden.

7. System-Sequenzdiagramm

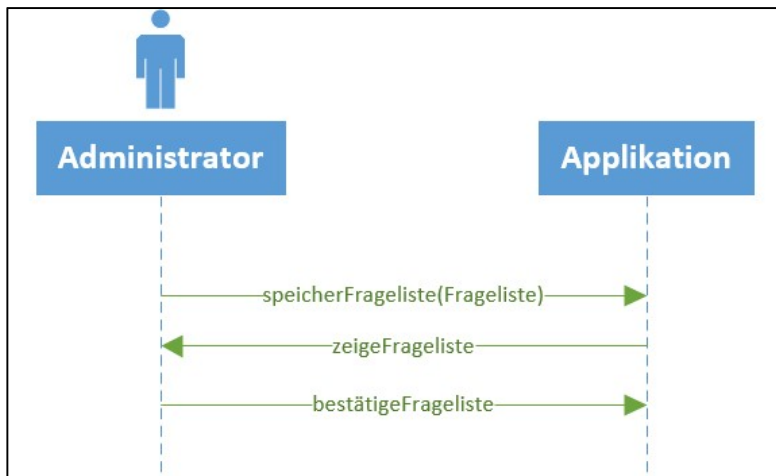
7.1. Use Case 1: Spiel Umgebung starten



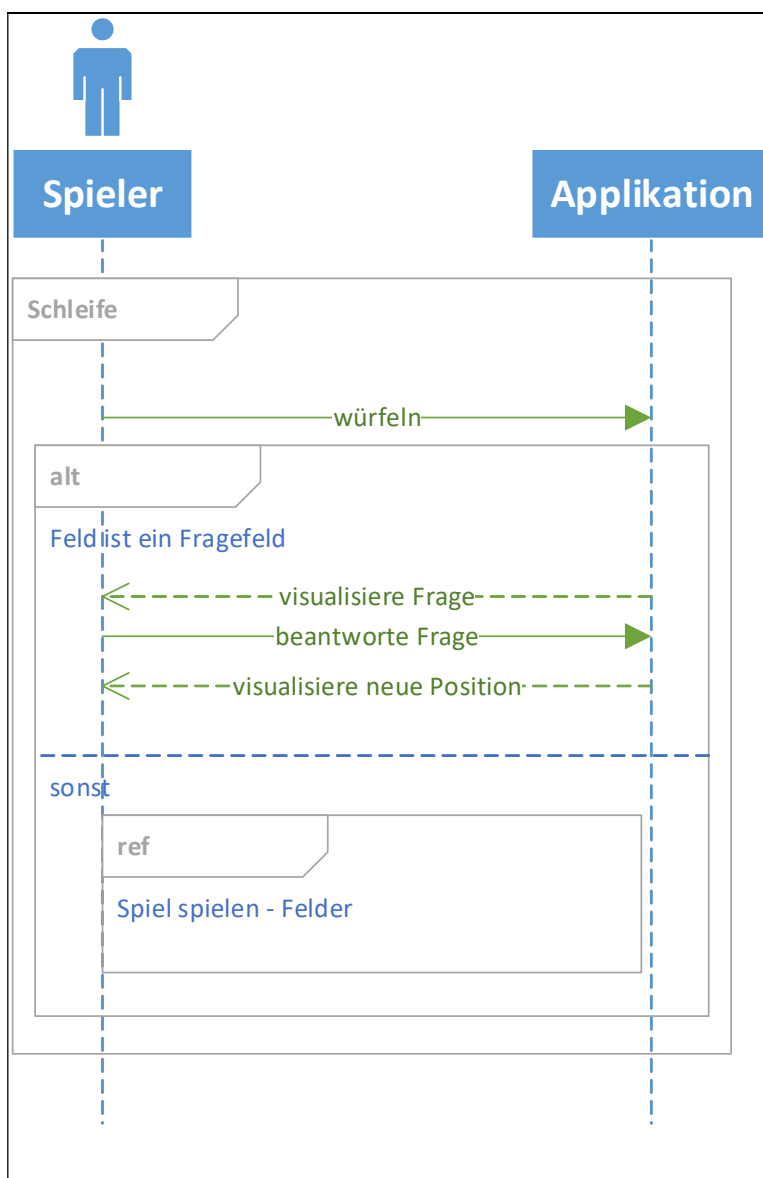
7.2. Use Case 2: Spiel beitreten

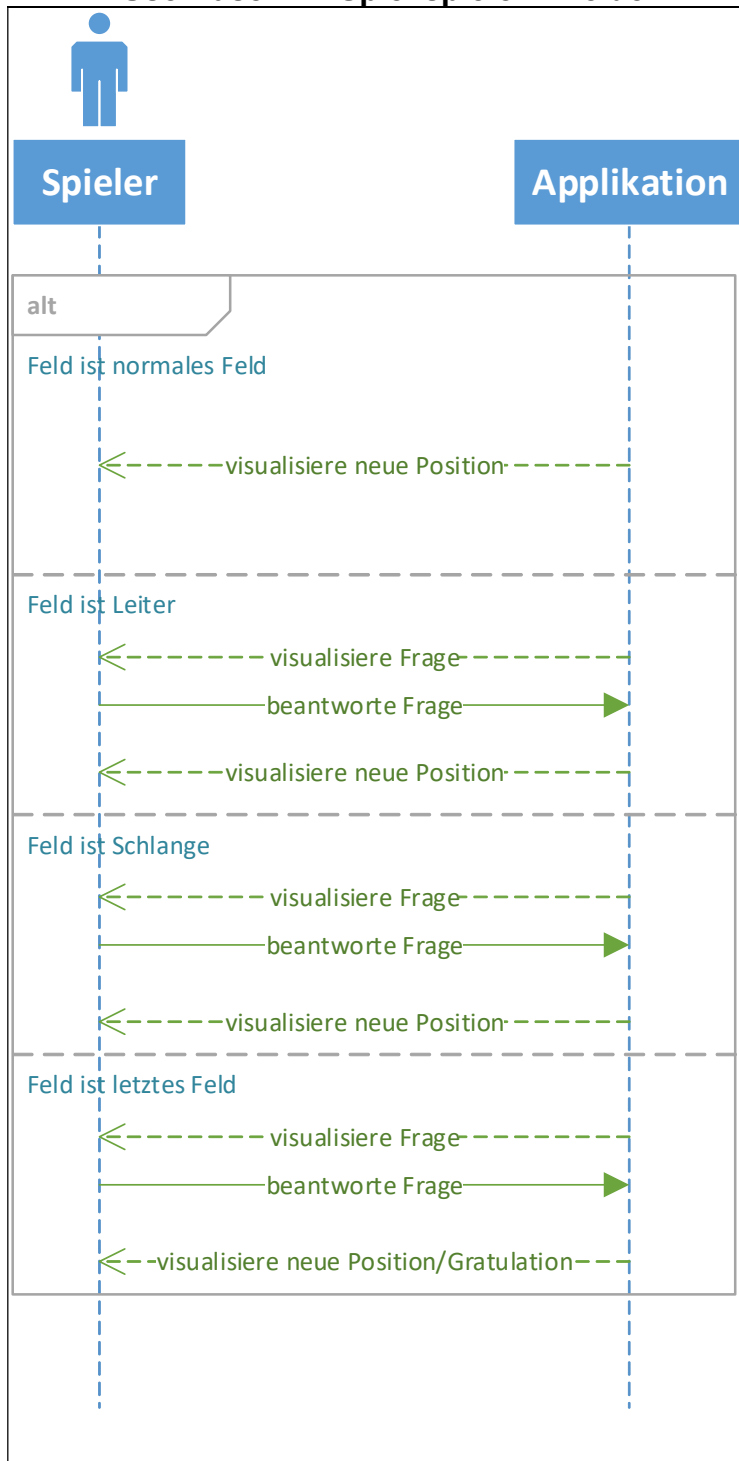


7.3. Use Case 3: Frageliste einlesen



7.4. Use Case 4: Spiel spielen



7.4.1. Use Case 4.1: Spiel spielen - Felder

8. Glossar

Begriff	Definition und Information	Weitere Erklärungen	Aliase, auch in anderen Sprachen	Beziehungen zu anderen Elementen
Administrator	Der Benutzer der Server-Applikation welche das Spiel konfigurieren kann.	Ist von einem Administrator die Rede so kann dieser sich nur auf der Server-Applikation befinden.	Admin, Host	Spieler
Spieler	Der Benutzer der Client-Applikation welche das Spiel spielt.	Ist von einem Spieler die Rede so kann dieser sich nur auf der Client-Applikation befinden.	Player, User	Administrator
Server-Applikation	Ist ein Programm, welches ein «Snakes and Ladder» Spiel verarbeitet und laufend den einzelnen Spielern ihre nächsten Schritte anzeigt.			