

# Software Engineering

WS2015/2016

TIM BIERENBREIER

Matrikel Nr.: 43235

JONAS ROTTMANN

Matrikel Nr.: 44501

JONAS WEBER

Matrikel Nr.: 43399

17. November 2015

# Inhaltsverzeichnis

<b>1</b>	<b>Analyse</b>	<b>2</b>
1.1	Use-Cases . . . . .	2
1.1.1	Use-Case Diagramm . . . . .	4
1.1.2	Erste Iteration . . . . .	4
1.2	Use-Cases, Details, Objektmodell und Schnittstellen der ersten Iteration . . . . .	5
1.2.1	Aktivitäts Diagramme . . . . .	5
1.2.2	Klassendiagramm . . . . .	7
1.2.3	Systemoperationen . . . . .	8
<b>2</b>	<b>Design</b>	<b>14</b>

# Kapitel 1

## Analyse

### 1.1 Use-Cases

*Beschreiben Sie jeden Use-Case (mindestens 5) mit eigenen Worten. Priorisieren Sie Ihre Use-Cases (essentiell, wichtig, unwichtig) und begründen Sie Ihre Entscheidung.*

#### **Spielfeld vorbereiten**

**Akteure** Spielleiter (Computer)

**Priorität** essentiell

Grundlage für den weiteren Spielverlauf.

**Beschreibung** Die Spieler bestimmen die 4 Kategorien. Jeder Spieler wählt eine Farbe. Die Wissensstreiter jedes Spielers werden auf die entsprechenden Heimatfelder gesetzt und die Wissensstandsanzeiger ggf. zurückgesetzt.

**Vorbedingungen** Ein Spiel wird von den Spielern gestartet.

#### **Beginnenden Spieler bestimmen**

**Akteure** Alle Spieler

**Priorität** unwichtig

Im Entwicklungsprozess nicht wichtig, da der beginnende Spieler leicht ohne Nebenwirkungen manuell bestimmt werden kann.

**Beschreibung** Alle Spieler würfeln einmal, die höchste Augenzahl beginnt. Falls mehr als ein Spieler die höchste Zahl würfelt, müssen diese Spieler erneut gegeneinander würfeln.

**Vorbedingungen** Es sind 2 bis 4 Spieler bekannt und es gibt einen Würfel.

#### **Zug spielen**

**Akteure** Ein Spieler

**Priorität** essentiell

Hauptbestandteil des Spiels.

**Beschreibung** Der Spieler würfelt. Wenn eine 6 fällt muss der Spieler einen Wissenstreiter auf das Spielfeld (sein Startfeld - das Feld seiner Farbe) bringen. Hat der Spieler keine 6 gewürfelt, oder sind bereits alle Wissenstreiter auf dem Feld, darf der Spieler einen seiner Wissenstreiter um die gewürfelte Augenzahl nach vorne ziehen. Hat der Spieler keine Wissenstreiter auf dem Feld, darf er maximal 3 mal würfeln bis eine 6 fällt. Wenn der Spielzug beendet wurde ist der Spieler zu seiner Rechten am Zug.

**Vorbedingungen** Ein Spieler ist am Zug.

#### **Wissenstest (extends „Zug spielen“)**

**Akteure** Ein oder zwei Spieler

**Priorität** essentiell

Hauptbestandteil des Spiels.

**Beschreibung** Spieler stellt anderem Spieler Frage aus einer der 4 Kategorien.

Frage wird korrekt beantwortet: Wissenstandsanzeiger dieser Kategorie wird inkrementiert. Wenn der Wissenstandszeiger dieser Kategorie auf höchster Stufe ist, kann eine beliebige andere Kategorie inkrementiert werden. Der Wissenstreiter des geprüften Spielers muss auf dessen Startfeld zurückgesetzt werden, ist dieses besetzt ins Heimatfeld.

Frage konnte nicht beantwortet werden: Der Wissenstandsanzeiger dieser Kategorie wird dekrementiert. Der Wissenstreiter des geprüften Spielers kommt ins Heimatfeld.

**Vorbedingungen** Spieler kommt auf ein Feld auf dem ein Wissenstreiter steht (beliebige Farbe).

#### **Anderen Spieler testen (erbt von „Wissenstest“)**

**Akteure** Zwei Spieler

**Priorität** essentiell

Hauptbestandteil des Spiels.

**Beschreibung** Zusätzlich: Beantwortet der zu testende Spieler die Frage falsch, kann der Fragesteller selbst eine Frage der entsprechenden Kategorie beantworten.

**Vorbedingungen** Feld ist von einem fremden Wissenstreiter belegt.

#### **Sich selbst testen (erbt von „Wissenstest“)**

**Akteure** Ein Spieler

**Priorität** unwichtig

Kann bei funktionierendem Wissenstest einfach nach implementiert werden.

**Beschreibung**

**Vorbedingungen** Feld ist von einem eigenen Wissensstreiter belegt.

**Sieger bestimmen**

**Akteure** Spielleiter (Computer)

**Priorität** unwichtig

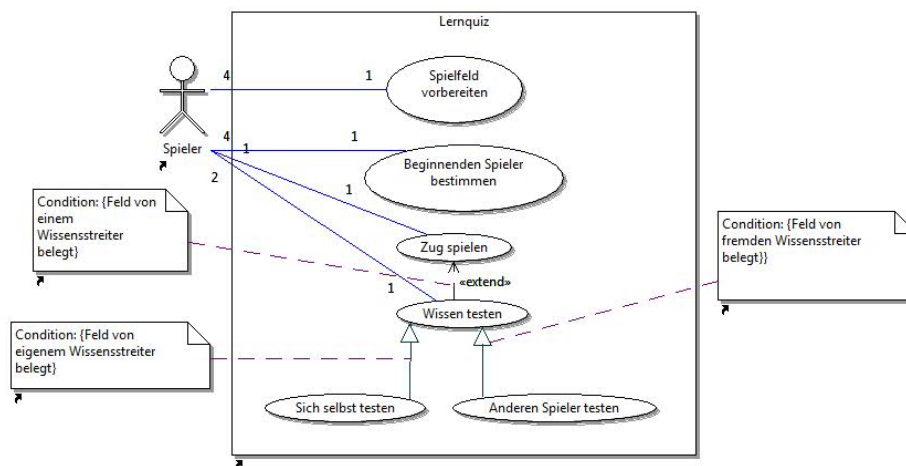
Für den Spielverlauf zuerst uninteressant.

**Beschreibung** Zeige den Gewinner an und biete an eine neue Runde zu starten.

**Vorbedingungen** Ein Spieler hat seine Wissenstandanzeige komplett gefüllt.

### 1.1.1 Use-Case Diagramm

Skizzieren Sie das Use-Case-Diagramm mit allen Akteuren und Abhängigkeiten.



### 1.1.2 Erste Iteration

Bestimmen Sie den Umfang der ersten Iteration (3 Use-Cases).

- Spielfeld vorbereiten
- Zug spielen
- Wissenstest

## 1.2 Use-Cases, Details, Objektmodell und Schnittstellen der ersten Iteration

### 1.2.1 Aktivitäts Diagramme

Erstellen Sie für die Use-Cases (obige 3) Beschreibungen in Form von Activity-Diagrammen.

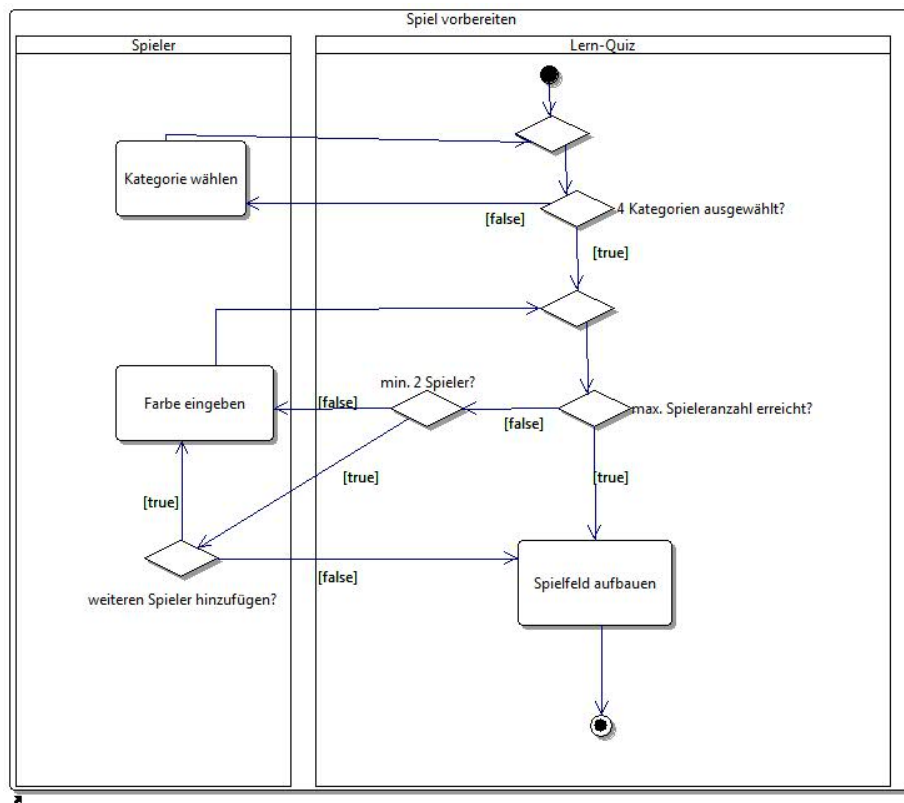


Abbildung 1.1: Spiefeld vorbereiten

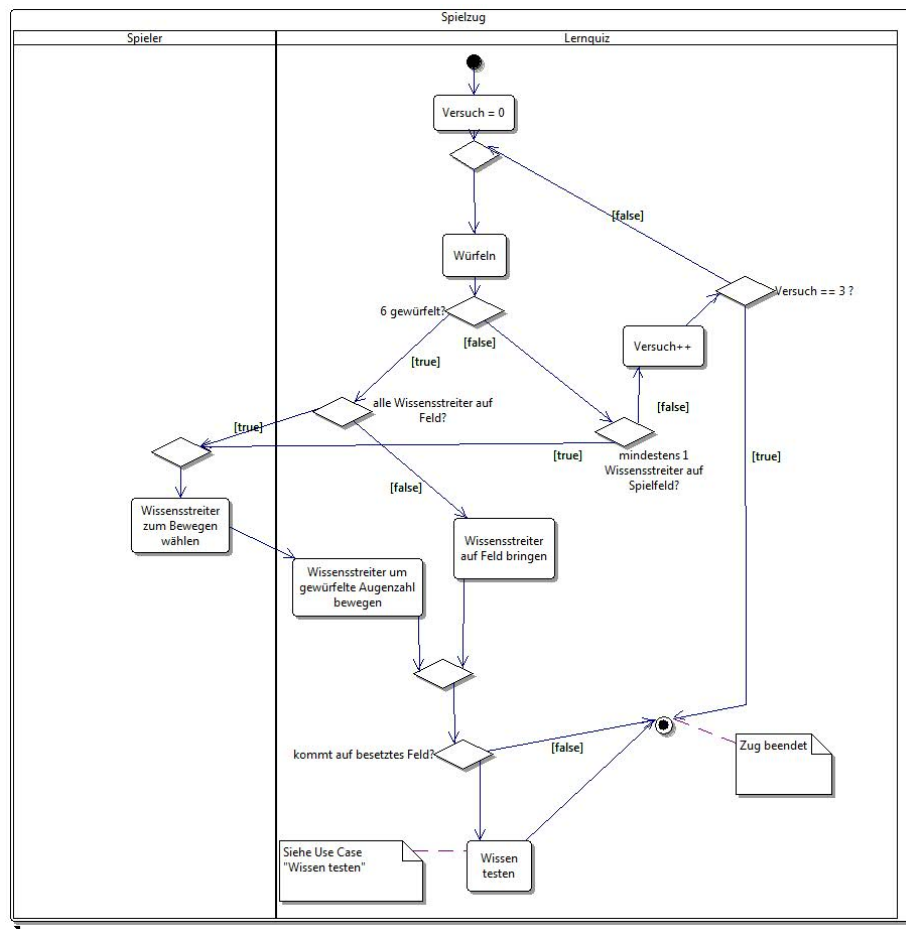


Abbildung 1.2: Zug spielen

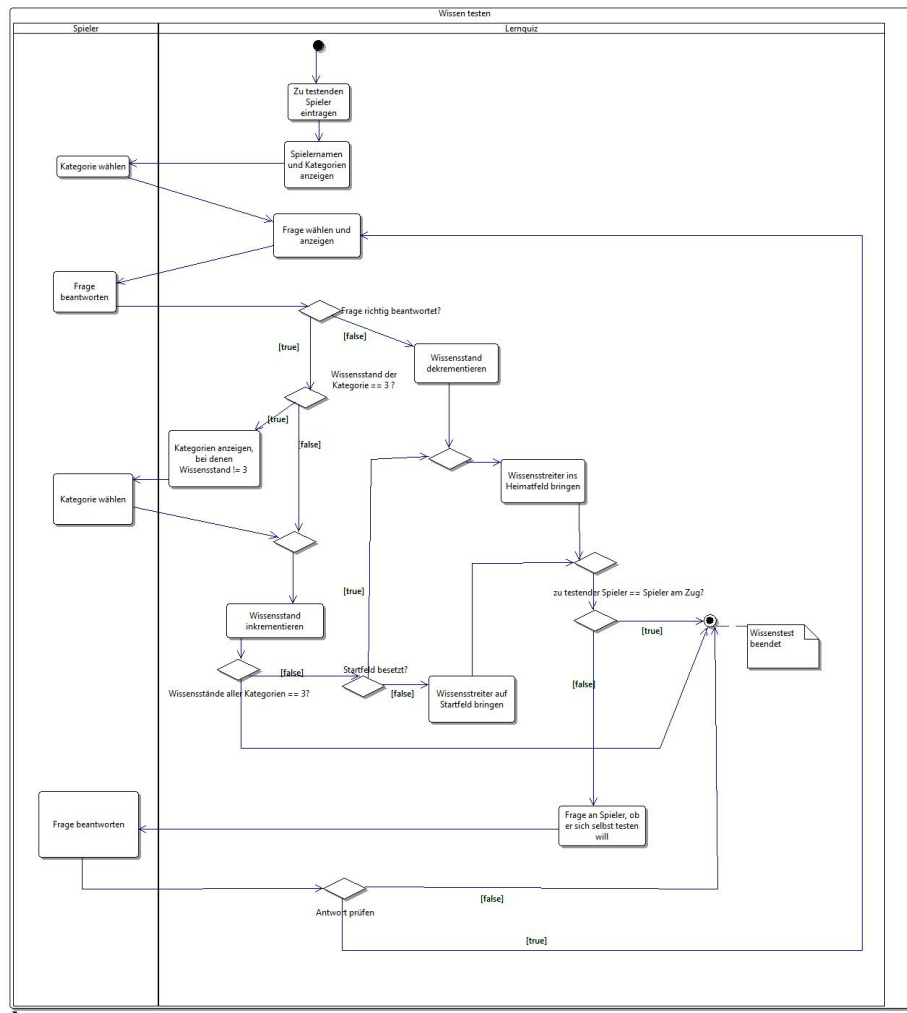


Abbildung 1.3: Wissen testen

### 1.2.2 Klassendiagramm

Extrahieren Sie aus den erstellten Diagrammen die Konzepte des Lern-Quiz-Computer-Spiels und ihre Beziehungen. Stellen Sie diese in Form eines Klassendiagramms dar (Objektmodell).



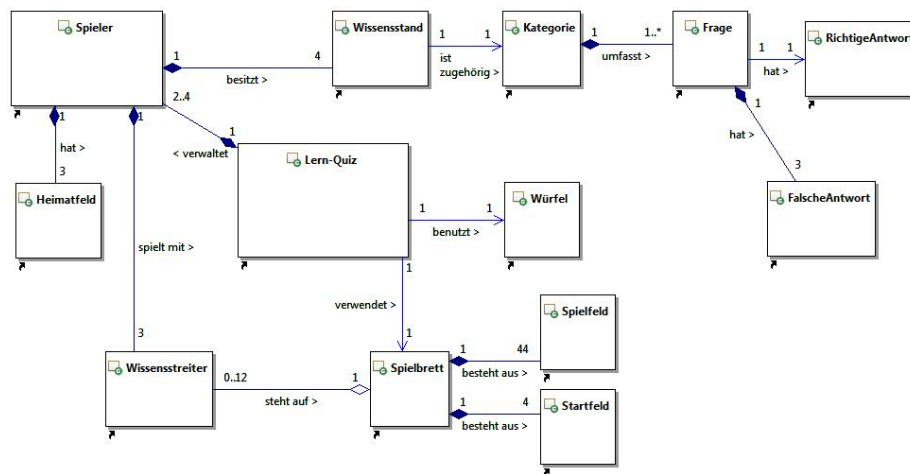
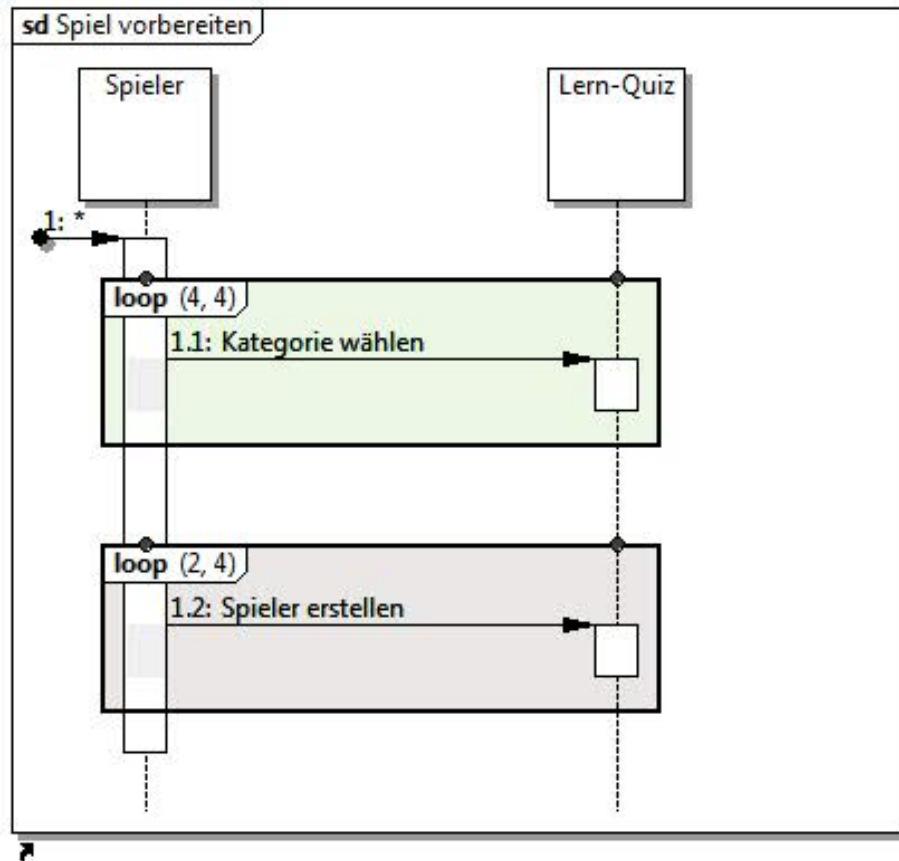


Abbildung 1.4: Klassen Diagramm

### 1.2.3 Systemoperationen

*Bestimmen Sie aus den Activity-Diagrammen die möglichen Systemoperationen. Erstellen Sie zur besseren Übersicht System-Sequenz-Diagramme und beschreiben Sie jede Operation mit eigenen Worten.*

### 1.2.3.1 Spiel vorbereiten



#### 1.2.3.1.1 Kategorie wählen

**Verantwortlichkeit** Ein Spieler wählt die Kategorien für die Fragerunden aus.

**Bemerkungen** Um Komplexität zu verhindern wählt Spieler 1 die Kategorien aus.

**Ausnahmen** Keine

**Vorbedingungen** Ein neues Spiel wurde gestartet.

**Nachbedingungen** Die Kategorien wurden im System dem neuen Spiel zugeordnet.

#### 1.2.3.1.2 Spieler erstellen

**Name** Spieler erstellen

**Verantwortlichkeit** Für jeden Spieler (2 bis 4) wird im System ein Spieler registriert.

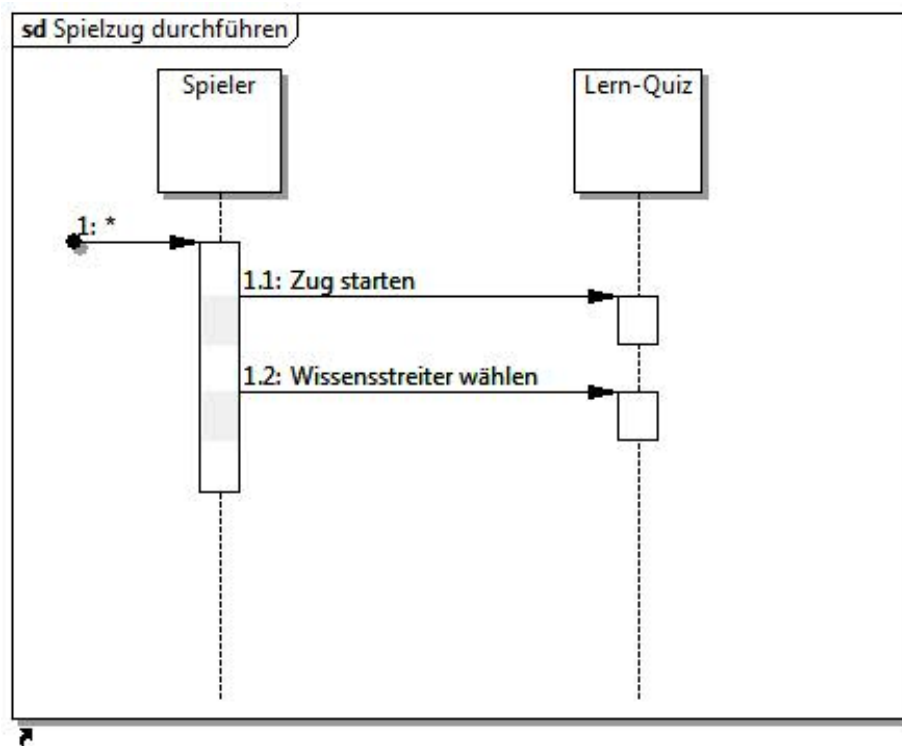
**Bemerkungen** Keine

**Ausnahmen** Keine

**Vorbedingungen** Kategorien wurden gewählt.

**Nachbedingungen** Das Spiel beginnt.

#### 1.2.3.2 Spielzug



##### 1.2.3.2.1 Zug starten

**Verantwortlichkeit** Ein Spieler beginnt seinen Zug indem er würfelt.

**Bemerkungen** Keine

**Ausnahmen** Keine

**Vorbedingungen** Der vorherige Spielzug wurde beendet und es wurde noch kein Gewinner ermittelt.

**Nachbedingungen** Ein Wissensstreiter muss bewegt werden.

#### **1.2.3.2.2 Wissensstreiter wählen**

**Verantwortlichkeit** Der Spieler wählt einen seiner Wissensstreiter aus und zieht ihn um  $n$  ( $n \approx$  gewürfelte Augenzahl) Felder nach vorne.

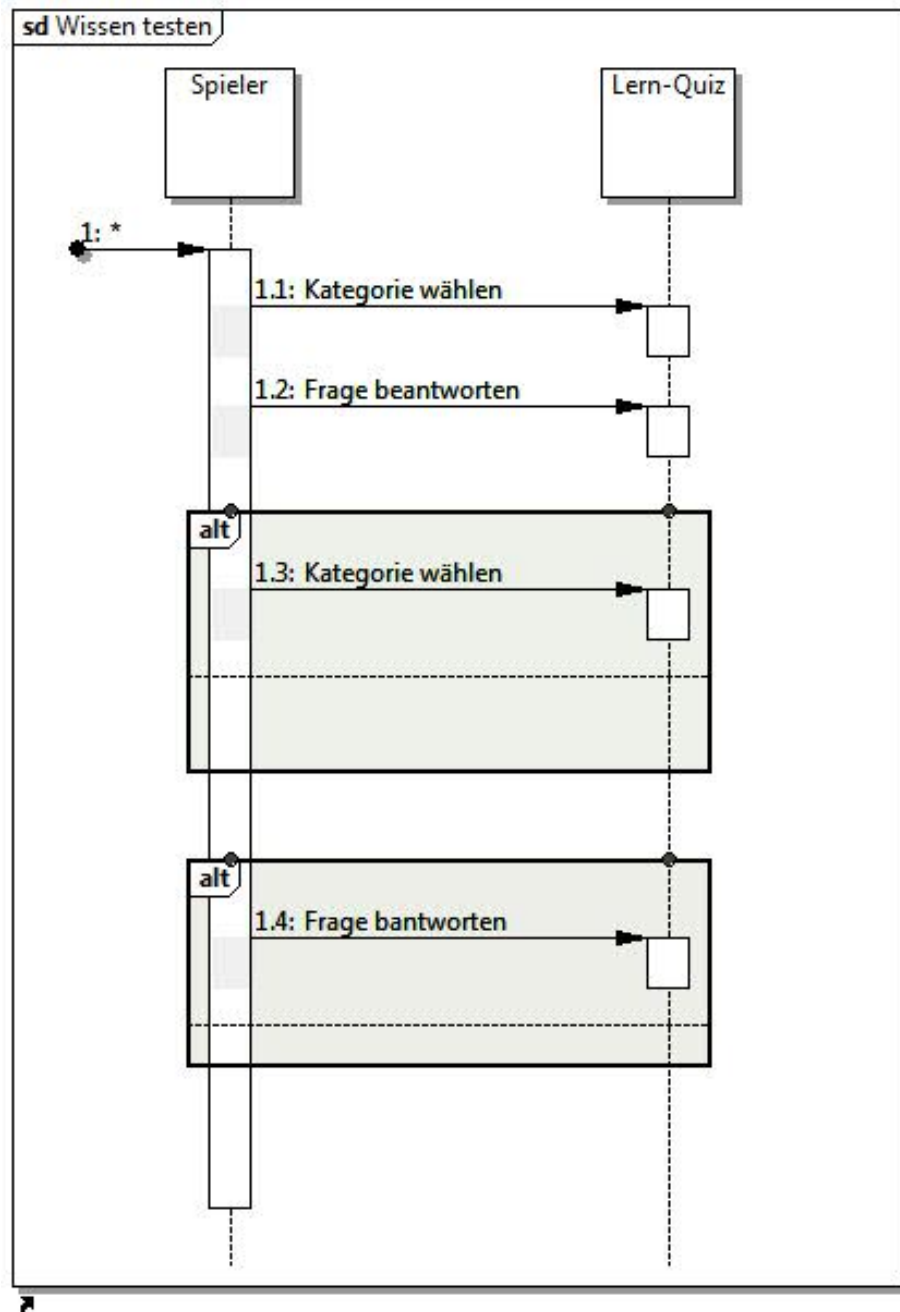
**Bemerkungen** Bei einer Augenzahl kleiner 6 darf der Spieler jeden seiner auf dem Spielfeld befindlichen Wissensstreiter wählen und bewegen. Bei einer Augenzahl von 6 muss der Spieler, falls noch nicht alle seiner Wissensstreiter auf dem Spielfeld sind einen Wissensstreiter aus seinem Heimatfeld auf sein Startfeld setzen.

**Ausnahmen** Hat der Spieler keine Wissensstreiter auf dem Feld und hat keine 6 gewürfelt, darf er maximal 3 mal würfeln bis eine 6 fällt. Dann darf ein Wissensstreiter auf das Startfeld gesetzt werden.

**Vorbedingungen** Der Spieler hat gewürfelt.

**Nachbedingungen**

### 1.2.3.3 Wissen testen



#### 1.2.3.3.1 Kategorie wählen

**Verantwortlichkeit** Der testende Spieler wählt eine der 4 Kategorien.

**Bemerkungen** Wurde die ursprüngliche Frage falsch beantwortet, erhält der Ersteller die Möglichkeit selbst eine Kategorie für eine weitere Frage zu wählen, die er selbst beantworten muss.

**Ausnahmen** Keine

**Vorbedingungen** Ein Spieler zieht einen seiner Wissensstreiter auf ein Feld, das von einem Wissensstreiter besetzt ist.

**Nachbedingungen** Die Frage muss beantwortet werden.

#### 1.2.3.3.2 Frage beantworten

**Verantwortlichkeit** Der Spieler beantwortet die Frage, die vom System gestellt wurde.

**Bemerkungen** Keine

**Ausnahmen** Keine

**Vorbedingungen** Eine Kategorie wurde gewählt.

**Nachbedingungen** Darauf wird die Position des Wissensstreiter und der Wissensstandanzeiger des Spielers verändert (siehe Use-Case „Wissenstest“).

## Kapitel 2

# Design