**SS**

15

**Karlsruhe 12.05.2015**

Analyse Softwareengineering SS15

Pascal van der Locht 43423  
Kishor Rana 43281  
Fabian Koffler 43047  
Gruppennr: 6

Funktionalität 3

Use Cases 5

Use Case Diagramm 5

Use Case Beschreibungen 5

Use Case Aktivitätsdiagramme 7

Objektmodell 10

System-Sequenz-Diagramme 11

Systemoperationen im Use Case „Spiel starten“ 11

Systemoperationen im Use Case „Spielzug durchführen“ 13

Systemoperationen im Use Case „Wissen testen“ 15

# Funktionalität

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr** | **Name** | **Beschreibung** | **Sichtbarkeit** | **Relevanz** |
| F-1 | Spielbrett aufbauen | Spielbrett aufbauen | Sichtbar | Notwendig |
| F-2 | Spielername eingeben  + Farbe auswählen | Mehrere Spieler möglich daher Namen | Sichtbar | Notwendig |
| F-3 | Kategorie festlegen | Auswählen welche Kategorien | Sichtbar | Notwendig |
| F-4 | Reihenfolge auswürfeln | Zufallszahl entscheidet wer beginnt | Nicht sichtbar | Notwendig |
| F-5 | Würfeln | Würfeln mit einem Würfel | Nicht sichtbar | Notwendig |
| F-6 | Würfelergebnis | Würfelergebnis anzeigen | Sichtbar | Notwendig |
| F-7 | Wissensstreiter aufs Feld setzen | Bei einer 6 und Streiter noch in Basis und keiner auf Startfeld🡪 Streiter aufs Feld | Sichtbar | Notwendig |
| F-8 | Wissensstreiter Startfeld belegt | Bei einer 6 und Streiter noch in Basis und einer auf Startfeld 🡪 Streiter muss Startfeld räumen | Sichtbar | Notwendig |
| F-9 | Kein Wissensstreiter in Basis | Bei einer 6 und keiner in Basis 🡪 beliebigen Streiter ziehen | Sichtbar | Notwendig |
| F-10 | Wissensstreiter bewegen | Bei allen anderen Fällen beliebigen Streiter bewegen | Sichtbar | Notwendig |
| F-11 | Ziel Feld belegt von eigenem Streiter | Streiter muss zurück in Basis | Sichtbar | Notwendig |
| F-11.1 | Variante: Zielfeld belegt von eigenem Streiter | Spieler stellt sich eigene Frage wenn frage richtig beantwortet 🡪 wissenstandsanzeiger erhöht  Wenn Frage falsch🡪Wissensstreiter zurück in Basis und Anzeiger wird erniedrigen | Sichtbar | Nicht notwendig |
| F-12 | Ziel Feld belegt von fremden Streiter | Frage aus einer der Kategorien stellen | Sichtbar | Notwendig |
| F-13 | Frage richtig beantwortet | Wissenstandsanzeiger der Kategorie um 1 erhöhen und Wissensstreiter auf Startfeld (auf Heimatfeld, wenn Startfeld besetzt) | Sichtbar | Notwendig |
| F-14 | Frage richtig beantwortet und Kategorie Maximum | Erhöhen einer beliebigen Kategorie um 1 und Wissensstreiter auf Startfeld (auf Heimatfeld, wenn Startfeld besetzt) | Sichtbar | Notwendig |
| F-15 | Frage Falsch beantwortet | Wissenstandsanzeiger der Kategorie um 1 verringern und Wissensstreiter auf Heimatfeld | Sichtbar | Notwendig |
| F-16 | Sieger bestimmen | Alle Kategorien eines Spielers auf Profi | Sichtbar | Notwendig |
| F-17 | Sieger in Highscore eintragen | Name und Punktzahl in Highscoreliste eintragen | Sichtbar | Notwendig |

# Use Cases

## Use Case Diagramm



## Use Case Beschreibungen

**Use Case 1 Spiel starten:**

Akteure: Spieler

Priorität: essentiell

Referenz: F1, F2, F3, F4

Beschreibung: Das Spielbrett wird dargestellt jeder Spieler gibt seinen Namen ein und wählt seine Farbe aus. Festlegen der Kategorien auswürfeln der Reihenfolge. Variante wählen

**Use Case 2 Zug durchführen:**

Akteure: Spieler

Priorität: essentiell

Referenz: F5, F6, F10

Beschreibung: Der Spieler würfelt und zieht einen beliebigen Wissensstreiter um entsprechende Augenzahl weiter.

**Use Case 3 Variante extends Use Case2:**

Akteure: Spieler

Priorität: unwichtig

Referenz: F11.1

Beschreibung: Spieler kommt nach würfeln auf Feld mit eigenem Wissensreiter darf sich selbst eine Frage aus einer beliebigen Kategorie stellen. Bei richtiger Beantwortung darf der Wissenstandsanzeiger erhöht werden bei falscher muss ein Wissensstreiter zurück ins Heimatfeld.

**Use Case 4 Spezialzug: Wissen testen extends Use Case 2:**

Akteure: Spieler

Priorität: essentiell

Referenz: F5,F6,F13,F14,F15

Beschreibung: Ein Wissensstreiter betritt ein durch einen anderen Wissensstreiter besetztes Feld. Er fragt nun den anderen Spieler eine Frage aus einer Kategorie seiner Wahl. Beantwortet der andere Spieler die Frage richtig, so erhöht sich der Wissensstandszeiger der gewählten Kategorie um 1. Beantwortet er die Frage falsch, so wird der Wissensstandzeiger um 1 erniedrigt und sein Wissensstreiter wird auf sein Heimatfeld zurückgesetzt. Des Weiteren ist bei falscher Antwort nun der andere Spieler an der Reihe eine Frage zu beantworten.

**Use Case 5 Siegerehrung:**

Akteure: Spieler

Priorität: unwichtig

Referenz: F16, F17

Beschreibung: Name und Punktzahl in die lokale Highscore liste eingetragen.

Auswahl der drei Use Cases für die erste Iteration, wurde anhand der obligatorischen Use Cases getroffen.

## Use Case Aktivitätsdiagramme



Aktivitätsdiagramm : zu Use Case 1, „Spiel starten“



Aktivitätsdiagramm 2: zu Use Case 2, „Spielzug durchführen“



Aktivitätsdiagramm 3: zu Use Case 3, „Wissen testen“

# Objektmodell

# System-Sequenz-Diagramme

## Systemoperationen im Use Case „Spiel starten“

## 

System-Sequenz-Diagramm 1: Zu Use Case „Spiel starten“

**Name: nameEingeben(name)**

|  |  |
| --- | --- |
| Verantwortlichkeit: | weist einem Spieler einen Namen zu |
| Referenzen: | Use Case „Spiel starten“ (UC-1) |
| Bemerkungen: |  |
| Ausnahmen: | Es sind bereits 4 Spielernamen eingegeben |
| Output: | Name des neuen Spielers |
| Vorbedingungen: | ein neues Spiel wurde gestartet |
| Nachbedingungen: | der Name des Spielers wurde in Spielerliste eingetragen |

**Name: farbeWählen(farbe)**

|  |  |
| --- | --- |
| Verantwortlichkeit: | weist einem Spieler eine Farbe zu |
| Referenzen: | Use Case „Spiel starten“ (UC-1) |
| Bemerkungen: | Es werden nur Farben angeboten, die bislang noch nicht gewählt wurden |
| Ausnahmen: | Jeder Spieler hat bereits seine Farbe gewählt |
| Output: | Zuweisung einer Farbe zum Spieler |
| Vorbedingungen: | Spieler wurden erstellt |
| Nachbedingungen: | Die gewählte Farbe wurde dem jeweiligen Spieler zugewiesen |

**Name: kategorieWählen(kategorie)**

|  |  |
| --- | --- |
| Verantwortlichkeit: | Fügt dem Spiel eine Fragekategorie hinzu |
| Referenzen: | Use Case „Spiel starten“ (UC-1) |
| Bemerkungen: | Es werden nur Kategorien angeboten, die bislang noch nicht gewählt wurden |
| Ausnahmen: | Jeder Spieler hat bereits eine Kategorie gewählt |
| Output: | Kategorie wird der Auswahl der Fragekategorien hinzugefügt |
| Vorbedingungen: | Spieler wurden erstellt |
| Nachbedingungen: | Die Kategorie wurde der Auswahl von Fragekategorien hinzugefügt |

## Systemoperationen im Use Case „Spielzug durchführen“



System-Sequenz-Diagramm 2: Zu Use Case „Spielzug durchführen“

**Name: würfeln()**

|  |  |
| --- | --- |
| Verantwortlichkeit: | Bestimmt zufällig eine Zahl zwischen [1-6] |
| Referenzen: | Use Case „Spielzug durchführen“ (UC-2) |
| Bemerkungen: |  |
| Ausnahmen: |  |
| Output: | Zufallszahl zwischen [1-6] |
| Vorbedingungen: | Ein Spieler beginnt einen Spielzug |
| Nachbedingungen: |  |

**Name: wissensstreiterWählen()**

|  |  |
| --- | --- |
| Verantwortlichkeit: | Der Spieler wählt einen Wissensstreiter aus mit dem der Spielzug fortgeführt wird |
| Referenzen: | Use Case „Spielzug durchführen“ (UC-2) |
| Bemerkungen: |  |
| Ausnahmen: |  |
| Output: | Der gewählte Wissensstreiter wird um die zuvor gewürfelte Anzahl an Spielfeldern fortbewegt |
| Vorbedingungen: | Es gibt mindestens einen Wissensstreiter, den der Spieler ziehen kann |
| Nachbedingungen: | Der Wissensstreiter ist um die zuvor gewürfelte Anzahl an Spielfeldern fortbewegt worden |

**Name: spielzugBeenden()**

|  |  |
| --- | --- |
| Verantwortlichkeit: | Beendet den Spielzug |
| Referenzen: | Use Case „Spielzug durchführen“ (UC-2) |
| Bemerkungen: |  |
| Ausnahmen: | Jeder Spieler hat bereits seine Farbe gewählt |
| Output: | Zuweisung einer Farbe zum Spieler |
| Vorbedingungen: | Der Spieler kann nicht mehr würfeln, bzw. keinen Wissensstreiter für seinen Zug auswählen. |
| Nachbedingungen: | Der Spielzug ist beendet und der nächste Spieler ist an der Reihe. (solang kein Sieger bestimmt ist) |

## Systemoperationen im Use Case „Wissen testen“

## 

System-Sequenz-Diagramm 3: Zu Use Case „Wissen testen“

**Name: kategorieWählen(kategorie)**

|  |  |
| --- | --- |
| Verantwortlichkeit: | Es wird eine Kategorie gewählt aus der nachfolgend eine Frage gestellt wird |
| Referenzen: | Use Case „Wissen testen“ (UC-3) |
| Bemerkungen: |  |
| Ausnahmen: |  |
| Output: |  |
| Vorbedingungen: | Ein Spieler beginnt einen Spielzug |
| Nachbedingungen: | Es ist eine Kategorie bestimmt worden |

**Name: frageBeantworten(antwort)**

|  |  |
| --- | --- |
| Verantwortlichkeit: | Der Spieler gibt seine Antwort auf die zuvor gestellte Frage ein |
| Referenzen: | Use Case „Wissen testen“ (UC-3) |
| Bemerkungen: |  |
| Ausnahmen: |  |
| Output: | Die Antwort wird ausgewertet und das Resultat zurückgegeben |
| Vorbedingungen: | Es ist eine Frage gestellt worden |
| Nachbedingungen: |  |