

Software Engineering

WS2015/2016

TIM BIERENBREIER

Matrikel Nr.: 43235

JONAS ROTTMANN

Matrikel Nr.: 44501

JONAS WEBER

Matrikel Nr.: 43399

21. Dezember 2015

Inhaltsverzeichnis

1	Design	2
1.1	Aufbereitung der Analyse und Systemoperationen	2
1.1.1	Use-Case-Beschreibung	2
1.1.2	Use-Case-Diagramm	4
1.1.3	Aktivitäts-Diagramme	4
1.1.4	System-Sequenz-Diagramme	4
1.1.5	System-Operationen	4
1.2	Zustandsautomat	6
1.3	Detailed Design und das Objektmodell	6

Kapitel 1

Design

1.1 Aufbereitung der Analyse und Systemoperationen

Beschreiben Sie die zu realisierenden Use Cases in Form von System-Use-Cases.

- *Use-Case-Beschreibung*
- *Use-Case-Diagramm*
- *detaillierte Beschreibung in Form von Activity-Diagrammen*

Orientieren Sie sich hierzu an Ihren Analyseergebnissen und berücksichtigen Sie die gewählte Architektur und Ihr Bedienkonzept.

1.1.1 Use-Case-Beschreibung

1.1.1.1 Zug spielen

Akteure

View, Controller

Priorität

essentiell
Hauptbestandteil des Spiels

Beschreibung

1. Würfelzähler auf 0 zurücksetzen, Spieler der am Zug ist aus Variable auslesen und vom View ausgeben lassen.
2. Pseudozufallszahl zwischen 1 und 6 als Würfelergebnis bestimmen, den Würfelzähler inkrementieren und Würfelergebnis vom View anzeigen lassen.

3. Prüfen ob der Spieler am Zug schon mindestens einen Wissensstreiter auf dem Spielfeld hat.

Fall: Spieler am Zug hat schon mindestens einen Wissensstreiter auf dem Feld.

- 4a. Es sind noch nicht alle Wissensstreiter auf dem Feld und das Würfelerggebnis $== 6$, falls ja muss ein Wissensstreiter aus dem Heimatfeld auf das Startfeld gebracht werden und der Zug ist beendet.
- 5a. Würfelerggebnis und Positionen aller Wissensstreiter von View ausgeben lassen.
- 6a. View zeigt Frage welcher Wissensstreiter bewegt werden soll an und der Controller erwartet eine Eingabe.
- 7a. Eingabe wird auf Gültigkeit geprüft, man kehrt ggf. zu 6a zurück.
- 8a. Die Position des Wissensstreiters wird auf die alte Position + Würfelerggebnis geändert, es wird geprüft ob dieses neue Feld schon besetzt ist, und ggf. der Use-Case „Wissen testen“ gestartet. Falls dieses Feld noch nicht besetzt ist, wird dieser Spielzug beendet.

Fall: Spieler am Zug hat noch keine eigenen Wissensstreiter auf dem Feld.

- 4b. Wird geprüft ob das Würfelerggebnis $== 6$ ist, falls ja wird ein Wissensstreiter aus dem Heimatfeld auf das Startfeld gesetzt und der Zug ist beendet. Wenn dies nicht der Fall ist wird geprüft ob der Würfelzähler < 3 , falls ja wird zurück zu Schritt 2. gesprungen, falls nicht ist der Spielzug ebenso beendet.

Vorbedingungen Spieler der am Zug ist wurde in einem Feld gespeichert.

Nachbedingung Nächster Spieler wird im Spieler am Zug Feld vermerkt.

1.1.1.2 Wissen Testen

Akteure

View

Priorität

essentiell

Hauptbestandteil des Spiels

Beschreibung

1. Präsentationsschicht fragt den prüfenden Spieler nach einer Kategorie.
2. Applikationssschicht wählt pseudozufällig eine Frage dieser Kategorie.
3. Präsentationsschicht zeigt die Frage und 4 Antwortmöglichkeiten für den geprüften Spieler an und fordert Eingabe.

4. Applikationsschicht wertet Antwort aus. Ist diese falsch wird der Wissensstandsanzeiger dekrementiert (wobei der Minimalwert 0 beträgt). Ist die Antwort richtig wird geprüft ob der Wissensstandsanzeiger schon den maximal Wert erreicht hat, ist dies der Fall fragt die Präsentationsschicht welche Kategorie ansonsten erhöht werden soll.
5. Der Wissenstreiter des geprüften Spielers wird bewegt:
 - Antwort war richtig: Wissensstreiter wird auf das Startfeld des Spielers gesetzt, ist dieses bereits besetzt muss der Wissensstreiter ins Heimatfeld.
 - Antwort war falsch: Wissensstreiter wird zurück ins Heimatfeld gesetzt.
6. Die Applikation vergleicht ob der prüfende Spieler != dem geprüften Spieler, falls nicht wurde der Wissenstest beendet. Stimmen prüfender Spieler und geprüfter Spieler nicht überein fragt die Präsentationsschicht den prüfenden Spieler ob er sich selbst testen will. Falls ja geht es wieder bei Schritt 1. weiter, falls nicht ist der Wissenstest beendet.

Vorbedingungen Wissensstreiter wurde auf Feld gezogen, welches schon besetzt ist.

1.1.2 Use-Case-Diagramm

1.1.3 Aktivitäts-Diagramme

1.1.4 System-Sequenz-Diagramme

1.1.5 System-Operationen

1.1.5.1 Zug spielen

1.1.5.1.1 wissensstreiterBewegen (Wissensstreiter wissensstreiter)

Verantwortlichkeit Das aktuelle Feld des vom Controller entgegengenommenen Wissensstreiters wird um die zuvor gespeicherte Augenzahl inkrementiert. Das aktuelle Feld des geschlagenen Spielers wird auf sein Heimatfeld gesetzt. Anschließend wird überprüft, ob das neue Feld leer oder belegt ist.

Bemerkungen Gewählter Wissensstreiter wird bewegt.

Ausnahmen keine

Vorbedingung Spiel ist initialisiert, eine Augenzahl wurde automatisch am Anfang des Zuges generiert und ein Spieler hat eine Eingabe gemacht, welchen Wissensstreiter bewegt werden soll.

Nachbedingung Wenn das neue Feld (altes Feld + Augenzahl) leer ist, ist der nächste Spieler am Zug. Wenn nicht, tritt der Use Case „Wissen testen“ ein.

1.1.5.2 Wissen testen

1.1.5.2.1 kategorieSetzen (Kategorie kategorie)

Verantwortlichkeit Die vom Controller entgegen genommene Kategorie wird gesetzt, um daraus die folgende Frage zu entnehmen.

Bemerkungen Kategorie wird für Fragerunde gesetzt.

Ausnahmen keine

Vorbedingung Wissensstreiter wurde auf Feld gezogen, auf dem schon ein Wissensstreiter steht.

Nachbedingung Antwort muss gewählt werden.

1.1.5.2.2 antwortPrüfen(int antwortNummer)

Verantwortlichkeit Die vom Controller entgegen genommene Antwortnummer wird mit der Antwortnummer der richtigen Antwort der Frage verglichen.

Bemerkungen Antwort wird überprüft.

Ausnahmen keine

Vorbedingung Eine Kategorie wurde gesetzt.

Nachbedingung Wenn die Antwort richtig ist, wird der Wissensstandszeiger des Spielers in dieser Kategorie inkrementiert. Wenn der Wissensstandszeiger bereits den Maximalwert 3 hat, wird nochmals eine Kategorie entgegen genommen, um den Wissensstandszeiger stattdessen in dieser Kategorie zu erhöhen.

Wenn die Antwort falsch ist, wird der Wissensstandszeiger dekrementiert. Wenn der befragte Spieler darüber hinaus ungleich dem fragenden Spieler ist, wird die Funktion frageBeantworten(antwortNummer) erneut für den fragenden Spieler aufgerufen.

1.1.5.2.3 wissensstandszeigerErhöhen(Kategorie kategorie)

Verantwortlichkeit Der Controller nimmt eine Kategorie entgegen, in der für den aktuellen Spieler die Wissensstandszeiger-Variable inkrementiert werden soll.

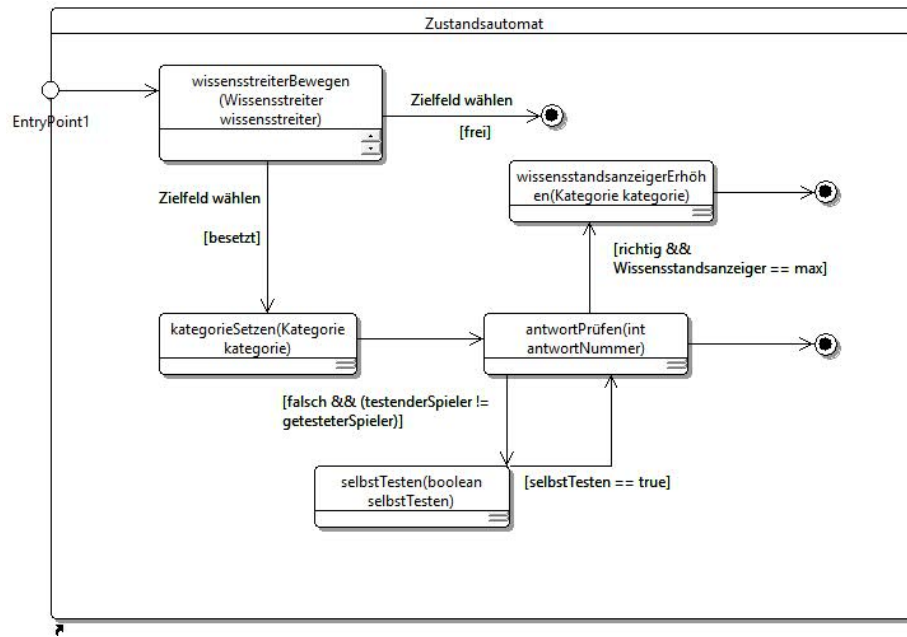
Bemerkungen keine

Ausnahmen keine

Vorbedingung Prüfen der Antwort lieferte true und die Wissensstandszeiger-Variable für die aktuelle Kategorie des aktuellen Spielers steht auf dem Max-Wert 3.

Nachbedingung Nächster Spieler ist an der Reihe.

1.2 Zustandsautomat



1.3 Detailed Design und das Objektmodell