10.11.2018

Robin Lauenroth

Version: 1.0

Status: IN Bearbeitung

UniRisk

Testspezifikation

# Historie

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Status | Datum | Autor(en) | Erläuterung |
| 1.0 | In Bearbeitung | 10.11.18 | Robin Lauenroth | Erster Entwurf |

[Historie 1](#_Toc529642220)

[1 Testkonzept 3](#_Toc529642221)

[2 Testspezifikation 3](#_Toc529642222)

[2.1 Komponententest 3](#_Toc529642223)

[2.2 Integrationstest 3](#_Toc529642224)

[2.3 Systemtest 3](#_Toc529642225)

[2.3.1 Anwendungsfall „Neues Spiel anlegen“ 3](#_Toc529642226)

[2.3.2 Anwendungsfall „Spiel Starten“ 3](#_Toc529642227)

[2.3.2.1 Erfolgreicher Start eines neuen Spiels ohne Bots 4](#_Toc529642228)

[2.3.2.2 Erfolgreicher Start eines neuen Spiels mit Bots 4](#_Toc529642229)

[2.3.2.3 Spielstart geladenes Spiel mit zu wenig Spielern unmöglich 4](#_Toc529642230)

[2.3.2.4 Erfolgreicher Start eines geladenen Spiels 5](#_Toc529642231)

[2.3.3 Anwendungsfall „Phase 2“ 5](#_Toc529642232)

[2.3.3.1 Erfolgreicher Angriff gestartet 5](#_Toc529642233)

[2.3.2.2 Erfolgreiche Abwehr durch einen Mitspieler 6](#_Toc529642234)

[2.3.2.3 ? 6](#_Toc529642235)

[3 Testprotokoll 6](#_Toc529642236)

# 

# 1 Testkonzept

Es werden sowohl die einzelnen Komponenten der Anwendung „UniRisk“ als auch die gesamte Anwendung getestet. Die Komponententests finden auf Basis der detaillierten Schnittstellenbeschreibung der Komponenten statt und werden mit JUnit durchgeführt.

# 2 Testspezifikation

## 2.1 Komponententest

## 2.2 Integrationstest

## 2.3 Systemtest

Im Systemtest wird vorrangig die korrekte Funktionsweise bei korrekter Eingabe geprüft. Zweitrangig wird überprüft ob fehlerhafte Eingaben nicht zu einem Systemausfall führen.

### 2.3.1 Anwendungsfall „Neues Spiel anlegen“

### 2.3.2 Anwendungsfall „Spiel Starten“

**Testdaten**

Für die folgenden Testfälle werden fünf Dummy Spieler benötigt. Je nach Testfall werden dann alle oder nur einige auf “Bereit” geschaltet. Für den Testfall 4 werden zusätzlich noch Dummy Spieldaten eines geladenen Spiels benötigt.

*IP-Adresse Name Zustand*

192.168.0.100 Horst Bereit/Nicht bereit

192.168.0.101 M@rianne Bereit/Nicht bereit

192.168.0.102 Bartholomäus Bereit/Nicht bereit

192.168.0.103 P3t3r Bereit/Nicht bereit

192.168.0.104 Piet Bereit/Nicht bereit

*Spieldaten eines geladenen Spiels*. Der Inhalt der Spieldaten spielt bis auf die Namen der Spieler keine Rolle und kann zufällig entsprechend der Datenstruktur gewählt werden. Die oben genannten Spieler sollen als Spieler in den Spieldaten hinterlegt sein.

#### 2.3.2.1 Erfolgreicher Start eines neuen Spiels ohne Bots

**Testablauf**

-Im geladenen Spiel sind alle fünf Spieler hinterlegt

-Nur vier davon sind auf “Bereit” geschaltet

-Button “Spiel Starten” klicken

**Testergebnis**

-Keine Aufforderung an die Clients, dass sie die Karte aktualisieren sollen

-Meldung “Mindestens ein Spieler fehlt noch” an GUIServer gesendet

#### 2.3.2.2 Erfolgreicher Start eines neuen Spiels mit Bots

-Drei Dummy Spieler sind auf “Bereit” geschaltet. Beim Anlegen des Spiels wurden jedoch fünf Spieler angegeben

-Button “Spiel Starten” klicken

**Testergebnis**

-Es wurden Bots erstellt

-Ein zufälliger Spieler wurde für den ersten Zug ausgewählt

-Die Methode zur Aktualisierung der Karte an alle Spieler wurde aufgerufen

#### 2.3.2.3 Spielstart geladenes Spiel mit zu wenig Spielern unmöglich

**Testablauf**

-Im geladenen Spiel sind alle fünf Spieler hinterlegt

-Nur vier davon sind auf “Bereit” geschaltet

-Button “Spiel Starten” klicken

**Testergebnis**

-Keine Aufforderung an die Clients, dass sie die Karte aktualisieren sollen

-Meldung “Mindestens ein Spieler fehlt noch” an GUIServer gesendet

#### 2.3.2.4 Erfolgreicher Start eines geladenen Spiels

**Testablauf**

-Im geladenen Spiel sind alle fünf Spieler hinterlegt

-Alle sind auf “Bereit” geschaltet

-Button “Spiel Starten” klicken

**Testergebnis**

-Es wurden keine Bots erstellt

-Dem Spieler der laut Spieldaten am Zug ist, wurde der Zug zugeteilt

-Die Methode zur Aktualisierung der Karte an alle Spieler wurde aufgerufen

### 2.3.3 Anwendungsfall „Phase 2“

**Testdaten**

Für die folgenden Testfälle werden mindestens drei Dummy Spieler benötigt.

Der Spieler, der am Zug ist muss sich in Phase 2 befinden. Die anderen Spieler können Bots oder auch echte Mitspieler sein.

Der aktuelle Spieler muss mindestens ein Gebäude mit mehr als einem Erstie besitzen.

#### 2.3.3.1 Erfolgreicher Angriff gestartet

**Testablauf**

-Ein Spieler ist am Zug in Phase 2

-Button “Angreifen” klicken, eigenes Gebäude auswählen, Gegner-Gebäude auswählen, Erstie-Anzahl festlegen

-Testergebnis abwarten, dann Test mit anderen Gebäuden und anderer Erstie-Anzahl wiederholen

**Testergebnis**

-Die eingegebenen Daten wurden überprüft und ein Angriff gestartet

-Die richtige Anzahl an Würfel wird angezeigt

-Die Methode „angriffAbwehren“ wird aufgerufen und ausgeführt

#### 2.3.2.2 Erfolgreiche Abwehr durch einen Mitspieler

**Testablauf**

-Der Spieler des angegriffenen Gebäudes sieht den Dialog „Würfeln-Verteidiger“

-Button “Würfeln” klicken (Verteidiger)

-Button „Würfeln“ klicken (Angreifer)

**Testergebnis**

-Es werden die Würfel angezeigt.

-Die Würfel wurden miteinander verglichen und das Ergebnis wird angezeigt

-Die Methode zur Aktualisierung der Karte an alle Spieler wurde aufgerufen

#### 2.3.2.3 ?

### 2.3.3 Anwendungsfall „Ersties Verteilen“

**Testdaten**

Für die folgenden Testfällen werden…

2.3.3.1

**Testablauf**

**Testergebnis**

2.3.3.2

**Testablauf**

**Testergebnis**

2.3.3.3

**Testablauf**

**Testergebnis**

# 3 Testprotokoll