

Bomberman

Dokumentation Teil 2

Version 1.0.0, 26. Februar 2018 | Projektteam: Joey Sciamanna, Nico Janner

Inhalt

[1 Testen 2](#_Toc507412023)

[1.1 Testkonzept 2](#_Toc507412024)

[1.2 Testumgebung 2](#_Toc507412025)

[1.3 User Acceptance Testfälle 2](#_Toc507412026)

[1.4 Testprotokoll 5](#_Toc507412027)

[2 Installationsanleitung 6](#_Toc507412028)

[3 Benutzerhandbuch 6](#_Toc507412029)

[4 Fazit 7](#_Toc507412030)

# Testen

## Testkonzept

Mit den JUnit – Tests haben wir 4 Methoden getestet, die wir als «gefährlich» einstuften. Die Timer-Methode, die richtigen Spielernummern (bomberman:1, crazybomber: 2) oder die Namens-Methode wird mit den JUnit – Tests getestet. Mit den User Acceptance Testfällen wurden unser User-Stories getestet.

## Testumgebung

Die Testfälle wurden auf dem Betriebssystem Windows 10 mit der Java Version «JavaSE-1.8» getestet und ausgeführt. Unsere Testdaten basieren auf unseren User-Stories, die wir zu Beginn des Projektes festgelegt haben. Zum Testen haben wir die Klassen «Server» und «Client» gebraucht.

## User Acceptance Testfälle

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | T-01 |
| Anforderungen | US-01 |
| Testdaten | Spielername: bomberman |
| Ablauf | 1. Spieler gibt einen beliebigen Spielernamen ein und klickt den „Start“ Knopf. |
| Erwartetes Resultat | Eingabefeld erscheint (Anforderung, Textfeld und Button). Spielername kann eingegeben werden. |

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | T-02 |
| Anforderungen | US-02 |
| Vorbedingungen | US-01 : Als Spieler möchte ich mir vor dem Spiel einen Namen geben. |
| Testdaten | Spielername1: bomberman, Spielername2: crazybomber |
| Ablauf | 1. Spieler1 gibt einen beliebigen Spielernamen ein und klickt den „Start“ Knopf. 2. Spieler2 gibt einen beliebigen Spielernamen ein und klickt „Start“ Knopf. |
| Erwartetes Resultat | Nachdem der erste Spieler seinen Namen eingegeben hat wird das Spiel noch nicht gestartet, erst dann wenn auch der zweite Spieler seinen Namen eingibt und auf „Start“ Knopf drückt. |

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | T-03 |
| Anforderungen | US-03 |
| Testdaten | Spieler: bomberman, crazybomber |
| Ablauf | 1. Nach der Eingabe des Spielernamens wird der «Start» Knopf gedrückt. 2. Das Spielfeld (Map) erscheint mit dem Timer, welcher gleich runterzählt. |
| Erwartetes Resultat | Nach Drücken des «Start» Knopfes erscheint bei beiden Spielern das Spielfeld und der Timer fängt an herunterzuzählen. |

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | T-04 |
| Anforderungen | US-04 |
| Testdaten | Spieler: bomberman |
| Ablauf | 1. Spieler drückt entsprechende Tasten (Caps Lock oder nicht spielt keine Rolle). |
| Erwartetes Resultat | Spieler bewegt sich mit «w/W» einen Block nach vorne, mit «d/D» nach rechts und so weiter. Die Bombe kann mit der Leertaste gesetzt werden, aber nur, wenn er vorher eine Bewegung gemacht hat. |

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | T-05 |
| Anforderungen | US-05 |
| Testdaten | Spieler: bomberman |
| Ablauf | 1. Spieler bewegt sich und legt mit der Leertaste eine Bombe. 2. Die gelegte Bombe explodiert nach 2 Sekunden. |
| Erwartetes Resultat | Der Spieler kann mit der Leertaste eine Bombe legen, die dann nach 2 Sekunden explodiert. Bevor er aber die Bombe setzen kann muss er sich bewegen, somit wird verhindert, dass an der gleichen Stelle mehrere Bomben gelegt werden. |

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | T-06 |
| Anforderungen | US-06 |
| Testdaten | Spieler: bomberman, crazybomber |
| Ablauf | 1. Server wird auf einem PC gestartet. 2. Beide Clients verbinden sich mit dem Server durch die IP-Adresse. 3. Spieler können sich auf dem Spielfeld sehen und Aktionen des anderen Spielers werden aktualisiert. |
| Erwartetes Resultat | Beide Spieler können sich über das Netzwerk verbinden und sehen sich gegenseitig auf dem Spielfeld. Aktionen von einzelnen Spielern werden aktualisiert, somit haben beide Spieler die gleichen Informationen zum Spiel. |

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | T-07 |
| Anforderungen | US-07 |
| Vorbedingungen | US-05: Als Spieler möchte ich eine Bombe legen können, die nach einem Timer von 3 Sekunden explodiert. |
| Testdaten | Spieler: bomberman |
| Ablauf | 1. Spieler legt eine Bombe, während er im Radius der Explosion steht. 2. Spieler erleidet Schaden (je nach Entfernung zur Bombe), der beim Lebensbalken des entsprechenden Spielers abgezogen wird. |
| Erwartetes Resultat | Der erlittene Schaden wird beim richtigen Spieler abgezogen und der grüne Lebensbalken passt sich dementsprechend an. |

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | T-08 |
| Anforderungen | US-08 |
| Testdaten | Spieler: bomberman, crazybomber |
| Vorbedingungen | US-06 und US-07 |
| Ablauf | 1. Wenn ein Spieler Schaden erleidet, wird der entsprechende Lebensbalken bei beiden angepasst. |
| Erwartetes Resultat | «bomberman» erleidet 50 Schadenspunkte und der Lebensbalken von «bomberman» wird bei beiden Clients angepasst. |

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | T-09 |
| Anforderungen | US-09 |
| Testdaten | Spieler: bomberman, crazybomber |
| Vorbedingungen | US-06, US-07 und US-10 |
| Ablauf | 1. Spieler1 stirbt und der Todes-Screen wird angezeigt. 2. Wenn man den Todes-Screen anklickt, wird das Spiel neu gestartet. |
| Erwartetes Resultat | Spieler: bomberman stirbt durch eine Bombe und der Todesscreen wird auf beiden PC’s angezeigt. Der Todes-Screen wird angeklickt und das Spiel startet neu. Ansonsten schliesse ich das Programm. |

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | T-10 |
| Anforderungen | US-10 |
| Testdaten | Spieler: bomberman, crazybomber |
| Vorbedingungen | US-06 und US-07 |
| Ablauf | 1. Ein Spieler stirbt oder hat weniger Lebenspunkte, als der andere, wenn die Zeit abläuft. 2. Entsprechender Todesscreen wird angezeigt. |
| Erwartetes Resultat | «bomberman» hat weniger Lebenspunkte nachdem der Timer abgelaufen ist. Der entsprechende Screen wird auf beiden PC’s angezeigt, damit beide wissen wer verloren hat. |

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | T-11 |
| Anforderungen | US-11 |
| Testdaten | Spieler: bomberman, crazybomber |
| Ablauf | 1. Beide Spieler drücken den «Start» Knopf 2. Der Timer startet zu zählen (2min) |
| Erwartetes Resultat | Beide Spieler geben sich einen Namen und drücken «Start» Knopf. Wenn die Map erscheint startet auch der Timer und zählt herunter. |

## Testprotokoll

Name des Testers: Nico Janner

Release: v1.0b

Datum und Uhrzeit: 26. Februar 2018, 12:38

|  |  |  |
| --- | --- | --- |
| ID | Erfolgreich | Bemerkungen |
| T-01 | Ja | Der Testfall war erfolgreich, beide konnten einen beliebigen Spielernamen eingeben. |
| T-02 | Ja | Der Testfall war erfolgreich, das Spiel wird erst gestartet, wenn beide Spieler «Start» geklickt haben. |
| T-03 | Ja | Der Testfall war erfolgreich, die Map erscheint und der Timer zählt herunter. |
| T-04 | Ja | Der Testfall war erfolgreich, Spieler kann sich fehlerfrei bewegen. |
| T-05 | Ja | Der Testfall war erfolgreich, Spieler kann Bombe legen, die nach 2 Sekunden explodiert. |
| T-06 | Ja | Der Testfall war erfolgreich, beide können sich mit dem Server verbinden und den Client starten. |
| T-07 | Ja | Der Testfall war erfolgreich, die richtige Anzahl Lebenspunkte wurde abgezogen. |
| T-08 | Ja | Der Testfall war erfolgreich, beide sehen die aktuellen Lebensbalken. |
| T-09 | Ja | Der Testfall war erfolgreich, ein Neustart des Spiels ist möglich. |
| T-10 | Ja | Der Testfall war erfolgreich, beide Spieler wissen wer verloren hat oder ob unentschieden steht. |
| T-11 | Ja | Der Testfall war erfolgreich, der Timer zählt von 2 Minuten herunter. |

Name des Testers: Reangsi Im

Release: v1.0b

Datum und Uhrzeit: 26. Februar 2018, 12:38

|  |  |  |
| --- | --- | --- |
| ID | Erfolgreich | Bemerkungen |
| T-01 | Ja | Ich konnte beliebigen Namen eingeben: Im |
| T-02 | Ja | Konnte Spiel erst starten, nachdem mein Gegner auch «Start» gedrückt hat. |
| T-03 | Ja | Ich habe die Map gesehen und der Timer hat begonnen zu zählen. |
| T-04 | Ja | Konnte mich bewegen, die Steuerung war aber ein bisschen komisch. |
| T-05 | Ja | Bombe explodiert nach 2 Sekunden |
| T-06 | Ja | Ich sehe meinen Gegner und mich über das Netzwerk. Bewegungen werden gleich übertragen. |
| T-07 | Ja | Leben wurden abgezogen nach explodierter Bombe. |
| T-08 | Ja | Ich sehe die Lebenspunkte von meinem Gegner, wenn er Schaden erlitten hat. |
| T-09 | Ja | Nach dem Spiel kann ich wählen, ob ich erneut spielen will. |
| T-10 | Ja | Todesanzeige wird bei beiden PC’s angezeigt. |
| T-11 | Ja | Timer zählt herunter von 2 Minuten. |

# Installationsanleitung

Bevor der Server gestartet werden kann, muss der richtige Port eingegeben werden, damit sich beide PC’s verbinden können. Beide Clients müssen dann bei sich den gleichen Port vom Server, sowie die IP-Adresse des Servers eingeben (im Sourcecode). Erst wenn der Server gestartet ist, können beide das eigentliche Programm starten.

# Benutzerhandbuch

Start-Menu / Nameneingabe:

Bevor das eigentliche Spiel beginnt, wird der Spieler nach einem in-game Namen gefragt, welcher für diese und folgende Runden gilt. Es wird ein JDialog – Fenster erscheinen und nachdem beide «Start» gedrückt haben, wird das Spielfeld angezeigt. Im oberen Bereich des Fensters sieht mal alle Spielerdaten, wie Name, Bild und Lebensanzeige. Auf der eigentlichen Map (Spielfeld) kann man sich mit den entsprechenden Tasten bewegen und Bomben legen.

Steuerung:

Die Steuerung funktioniert mit den Tasten «w/W,a/A,s/S,d/D» und die Bombe wird mit der Leertaste gelegt. Die Bombe wird dann nach 2 Sekunden explodieren. Die Explosion wird auf alle 4 Himmelsrichtungen über 3 Blöcke erfolgen. Dabei ist der Schaden unterschiedlich (je näher an der Bombe, desto höher der Schaden).

Spielende:

Das Spiel ist zu Ende, wenn ein Spieler keine Lebenspunkte mehr besitzt, indem er Leben durch eine Bombe verliert. Die zweite Möglichkeit ist, wenn der Timer von 2 Minuten abläuft. Jetzt wird geschaut welcher Spieler weniger Leben besitzt und ist somit auch der Verlierer. Aber wenn beide gleich viel Leben besitzen gibt es ein Unentschieden. Für einen Neustart des Spiels kann irgendwo auf den Todes-Screen geklickt werden.

# Fazit

Wir waren sehr schnell unterwegs schon zu Beginn, aber haben leider nochmals von Vorne begonnen, wegen dem RMI. Wir konnten alle unsere Muss-Ziele erreichen und sogar noch ein bisschen verbessern, jedoch hat uns der Neustart viel Zeit gekostet. Dieser Schritt war notwendig, um eine sinnvolle Einbindung des RMI zu erreichen. Die Explosion der Bombe hat viel Zeit in Kauf genommen, dass sie nur in die 4 Himmelsrichtungen sprengen soll. Aber nach langem Probieren haben wir ein Erfolgsgefühl erhalten und auf diese Codestelle sind wir demnach auch sehr stolz. Unser Programm hat da und dort noch einige kleine Mängel, die aber nicht sehr schlimm sind.

Wir sind beide sehr zufrieden und stolz auf unser Ergebnis. Trotz schweren Problemen und langen Stunden herumprobieren haben wir unser Schlussergebnis erreichen können. Leider konnten wir unsere Kann-Ziele nicht erreichen, da uns die Zeit dazu gefehlt hat.

In diesem Projekt durften wir viel lernen, auch gegenseitig im Team. Wir haben gelernt, dass das GANTT-Diagramm ein sehr nützliches Tool ist, um ein Projekt zu planen und die Zeit einzuteilen. Mit dem Scrum-Diagramm hatten wir den Überblick welche Ziele wir schon erreicht haben und welche noch offen sind. Wir haben auch gelernt, dass es wichtig ist vor dem Projekt Ziele zu definieren und regelmässig diese Ziele zu prüfen. Wenn man ein Programm schreibt, das über ein Netzwerk läuft ist RMI ein gutes Prinzip, wobei zu beachten ist, dass man es von Anfang an einsetzt.

Die Zusammenarbeit verlief sehr gut zwischen uns. Wir waren meistens einig und sonst haben wir einen anderen Weg gefunden, um ein Problem zu lösen oder etwas hinzuzufügen. Wir konnten voneinander viel lernen und profitieren. Da es schwierig war, dass jeder an einer Klasse arbeitete, wegen dem Zusammenhang aller Klassen, haben wir das Pair-Programming bevorzugt. Meistens hat einer programmiert und der andere hat nebendran mitgeholfen / kontrolliert.

In unserem Projekt ist soweit alles vorhanden, was wir uns vorgestellt und gesetzt haben. Leider konnten wir unsere Kann-Ziele nicht erreichen, weil uns die Zeit dafür gefehlt hat. Wir denken wir hätten ein paar Kann-Ziele erreichen können, wenn wir die RMI Präsentation früher gehabt hätten, denn wir haben von neu begonnen nachdem wir die Präsentation hatten. Aber für uns ist das nicht sehr schlimm, denn wir denken unser Programm ist auch ohne Kann-Ziele gut herausgekommen.

Das nächste Mal wäre es ein Vorteil, wenn wir früher wüssten welche Technologien wir einsetzen und wie wir diese einsetzen können(RMI). Ausserdem möchten wir uns das nächste Mal besser am UML orientieren und dies auch mehr nutzen, da dies ziemlich hilfreich war. Wir möchten nächstes Mal einen Weg finden, dass beide an einer eigenen Klasse arbeiten können, ohne das der andere behindert wird. So steigt die Produktivität und wir können das Ziel schneller erreichen. Wir haben gemerkt, dass wir während des codens keine Kommentare geschrieben haben, was zu Verständnisproblemen führen kann. Wir wollen auch JUnit-Test häufiger nutzen, auch schon beim Programmieren, um frühzeitig Fehler in Methoden zu indentifizieren.