

Bomberman

Dokumentation Teil 1

Version 1.0.0, 29. Januar 2018 | Projektteam: Joey Sciamanna, Nico Janner

Inhalt

[1 Abstract 2](#_Toc504996778)

[2 Anforderungsanalyse 3](#_Toc504996779)

[2.1 User Stories 3](#_Toc504996780)

[3 Lieferumfang 5](#_Toc504996781)

[3.1 Applikation 5](#_Toc504996782)

[3.2 Dokumentation 5](#_Toc504996783)

[4 GUI Design 6](#_Toc504996784)

[4.1 MockUp 6](#_Toc504996785)

[5 OO Analyse- / Design 7](#_Toc504996786)

[5.1 Klassendiagramm 7](#_Toc504996787)

# Abstract

Über Sockets können zwei Spieler an verschiedenen Computern gegeneinander spielen. Auf einem Spielfeld befinden sich zwei Spieler, welche Bomben mit einer Taste legen können. Diese Bomben gehen nach einem Timer(2 Sekunden) hoch, dabei beschädigen sie Spieler, sowie Blöcke. Dabei ist zu beachten, dass einige Blöcke unzerstörbar sind. Hingegen gibt es Blöcke, die mit Bomben zerstört werden, um an den Gegenspieler heranzukommen. Wenn eine Bombe hochgeht, zerstört sie 3 Felder in jede Himmelsrichtung, wo es keinen unzerstörbaren Block hat. Wird ein zerstörbarer Block von der Bombe gesprengt, verschwindet dieser, aber wenn ein Spieler getroffen wird, zieht man ihm die entsprechende Anzahl an Lebenspunkten ab. Das Spiel ist zu Ende, wenn einer der Spieler keine Leben mehr hat, oder der Spieltimer (5 min) abgelaufen ist. Der Spieler, der zu diesem Zeitpunkt weniger Leben hat, hat verloren. Es wird gefragt, ob erneut gespielt werden möchte oder ob das Spiel beendet werden sollte.

# Anforderungsanalyse

## User Stories

|  |  |
| --- | --- |
| Funktionale-US-01 | Beschreibung |
| User Story | Als Spieler möchte ich mir vor dem Spiel einen Namen geben. |
| Art | Muss-Ziel |
| Aufwand | 2-3 Stunden |

|  |  |
| --- | --- |
| Funktionale-US-02 | Beschreibung |
| User Story | Als Spieler möchte ich erst das Spiel starten können, wenn mein Gegner auch bereit ist (Namen eingegeben hat und auf Spiel starten gedrückt hat). |
| Art | Muss-Ziel |
| Aufwand | 4 Stunden |
| Abhängigkeiten: | Abhängig von User Story F-US-01 |

|  |  |
| --- | --- |
| Funktionale-US-03 | Beschreibung |
| User Story | Als Spieler möchte ich am Anfang des Spiels die Map mit dem Spieltimer sehen können. |
| Art | Muss-Ziel |
| Aufwand | 3 Stunden |

|  |  |
| --- | --- |
| Funktionale-US-04 | Beschreibung |
| User Story | Als Spieler möchte ich mich mit entsprechenden Tasten bewegen können. |
| Art | Muss-Ziel |
| Aufwand | 3 Stunden |
| Abhängigkeiten: | Abhängig von User Story F-US-03 |

|  |  |
| --- | --- |
| Funktionale-US-05 | Beschreibung |
| User Story | Als Spieler möchte ich eine Bombe legen können, die nach einem Timer von 3 Sekunden explodiert. |
| Art | Muss-Ziel |
| Aufwand | 3 Stunden |
| Abhängigkeiten: | Abhängig von User Story F-US-04 |

|  |  |
| --- | --- |
| Funktionale-US-06 | Beschreibung |
| User Story | Als Spieler möchte ich den anderen Spieler sehen können, da das Spiel über das Netzwerk (Socket) läuft. |
| Art | Muss-Ziel |
| Aufwand | 4 Stunden |
| Abhängigkeiten: | Abhängig von User Story F-US-02 |

|  |  |
| --- | --- |
| Funktionale-US-07 | Beschreibung |
| User Story | Als Spieler möchte ich sehen wieviele Lebenspunkte mir abgezogen worden sind durch die explodierte Bombe. |
| Art | Muss-Ziel |
| Aufwand | 3 Stunden |
| Abhängigkeiten: | Abhängig von User Story F-US-05 |

|  |  |
| --- | --- |
| Funktionale-US-08 | Beschreibung |
| User Story | Als Spieler möchte ich sehen können wie viele Lebenspunkte ich und mein Gegner noch haben. |
| Art | Muss-Ziel |
| Aufwand | 2 Stunde(n) |
| Abhängigkeiten: | Abhängig von User Story F-US-04 und F-US-06 |

|  |  |
| --- | --- |
| Funktionale-US-09 | Beschreibung |
| User Story | Als Spieler möchte ich nach einem Spiel auswählen können, ob ich erneut spielen möchte oder das Programm beenden möchte. |
| Art | Muss-Ziel |
| Aufwand | 2 Stunde(n) |

|  |  |
| --- | --- |
| Funktionale-US-10 | Beschreibung |
| User Story | Als Spieler möchte ich zwischen 3 verschiedenen Maps auswählen können. |
| Art | Kann-Ziel |
| Aufwand | 3 Stunden |

|  |  |
| --- | --- |
| Funktionale-US-11 | Beschreibung |
| User Story | Als Spieler möchte ich mir die Highscores ansehen können, indem ich auf den Highscorebutton drücke. |
| Art | Kann-Ziel |
| Aufwand | 4Stunden |

|  |  |
| --- | --- |
| Nichtfunktionale-US-01 | Beschreibung |
| User Story | Als Projektleiter möchte ich, dass die Dokumentation(Teil1) am Mittwoch 12:00 Uhr komplett fertig ist. |
| Art | Muss-Ziel |
| Aufwand | 3 Stunden |

|  |  |
| --- | --- |
| Nichtfunktionale-US-02 | Beschreibung |
| User Story | Als Projektleiter möchte ich, dass der Sourcecode übersichtlich und strukturiert gehalten / geschrieben wird, um Komplikationen zu vermeiden. |
| Art | Muss-Ziel |
| Aufwand | Dauernd, am Schluss überprüfen |

# Lieferumfang

Am Ende des Projekts müssen folgende Produkte abgegeben werden:

## Applikation

Am Ende des Projektes muss folgendes abgegeben werden:

* Ein ausführbares Java JAR File
* Der gesamte Source Code in einem JAR File
* Ein vollständiger MySQL DB Dump File (SQL)

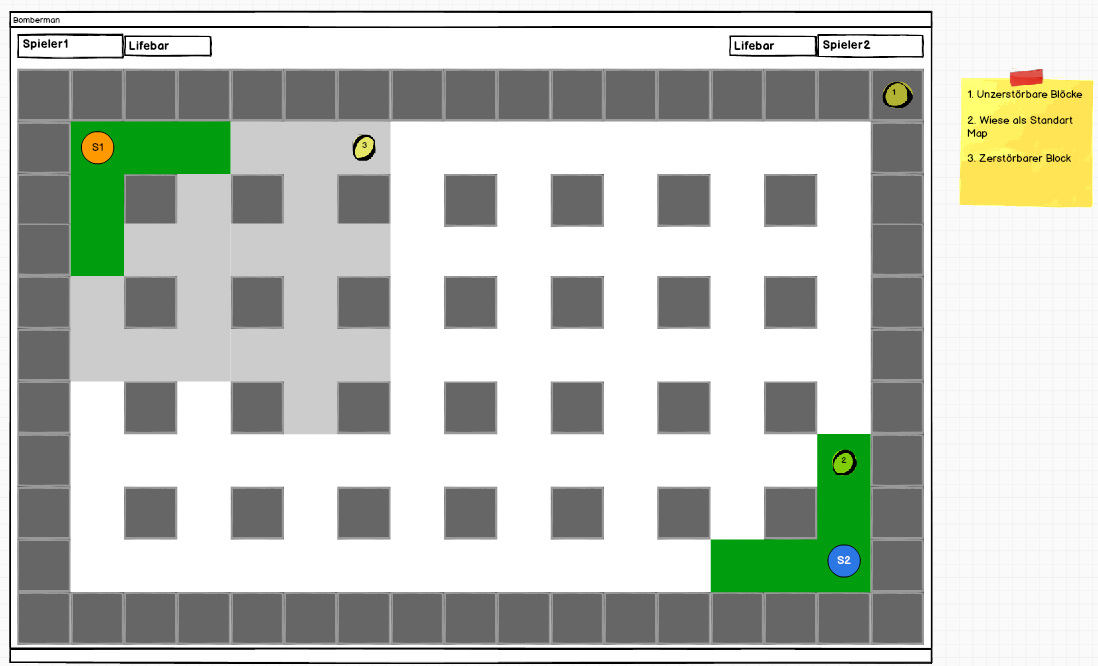
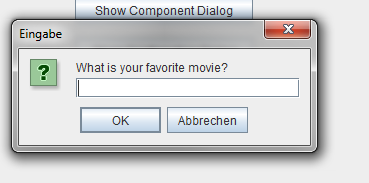
Nur für Java Projekte. Alle anderen nach Absprache mit Product Owner.

## Dokumentation

Die im Verlauf des Projekts entstandenen Dokumentationsteile 1 und 2 müssen jeweils zu den kommunizierten Terminen abgegeben werden. Alle geforderten Inhalte sind genügend genau beschrieben so, dass eine andere Gruppe das Projekt später weiterführen könnte.

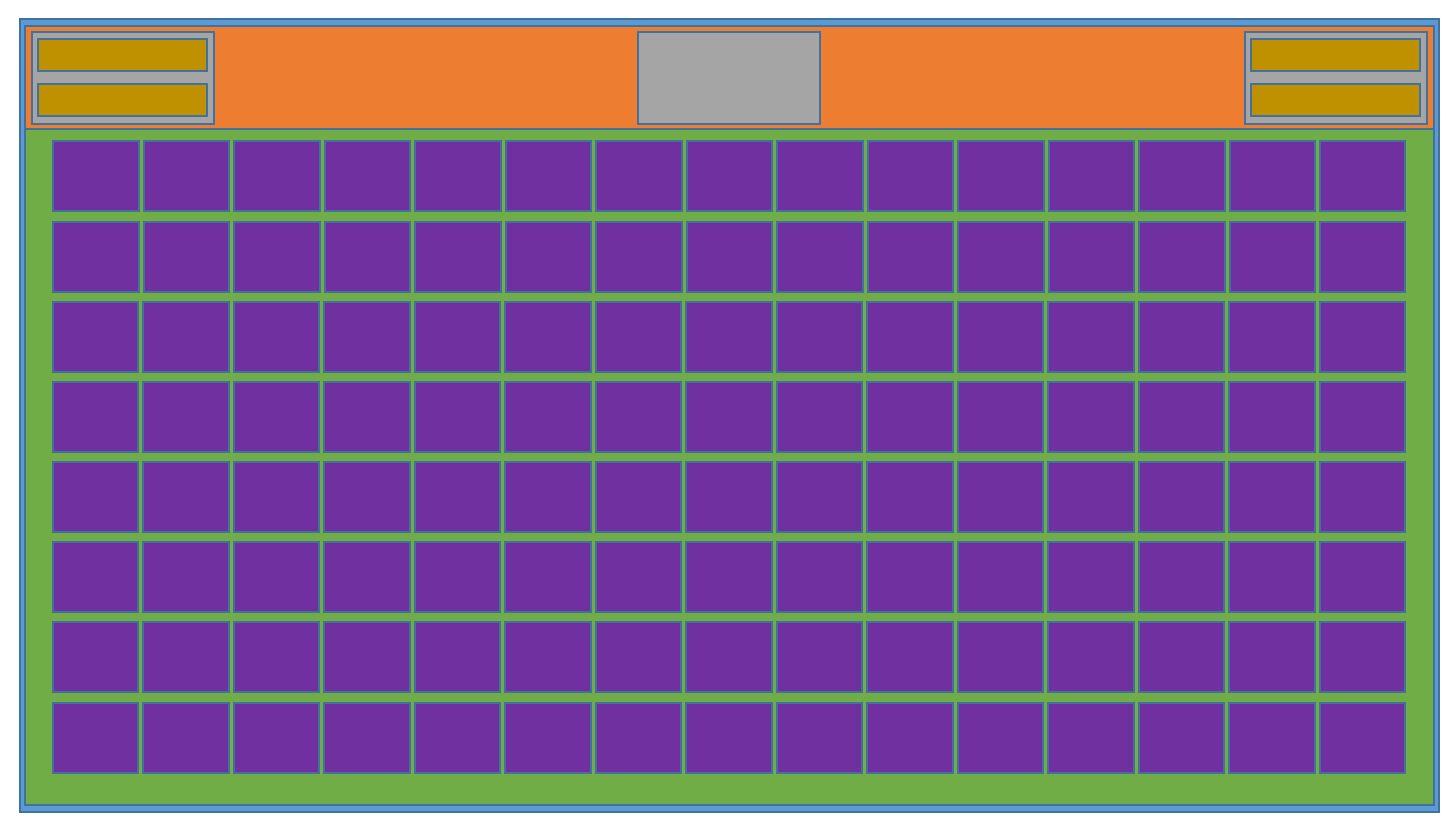
# GUI Design

## MockUp



Standart Map, welche nach dem eingeben der Namen erscheint.

Jeder Spieler gibt seinen (in-game) Namen ein und drückt auf den Button(JInputdialog), ist somit bereit zu spielen.



Layout:

Blau: JFrame, BorderLayout  
Orange: JPanel, BorderLayout  
Grau: JPanel, BorderLayout  
Gelb: JTextField  
Grün: JPanel, GridLayout  
Violet: JLabel

# OO Analyse- / Design

## Klassendiagramm

Vollständiges UML Klassendiagramm. Es muss nur die Modellschicht (Abstraktion der Realität) darstellen. Klassenbeziehung (Vererbung, Assoziation), Sichtbarkeitsmodifikatoren, static, Datentypen, Attribute und Methoden müssen vollständig sein. Bei Assoziationen müssen Richtung, Rollen und Kardinalitäten angegeben werden. Java Standard Klassen und Klassen aus verwendeten Frameworks müssen nicht modelliert werden, ausser es trägt zu einem besseren Verständnis bei.

