TECHNISCHE UNIVERSITÄT
CHEMNITZ

Bomberman

Pflichtenheft

Erstellt von:

DONGER SoftwareEngineering

(Gruppe 7)

11.12.2016 Version 1

Inhaltsverzeichnis

1. Anforderungen

1.1 Funktionale Anforderungen

1.2 Nicht-funktionale Anforderungen

2. Produkteinsatz

2.1. Beschreibung des Problembereichs

2.2. Modell des Problembereichs

2.3. Glossar

3. Vision

4. Produktfunktionen

4.1. Use-Cases

4.1.1. A1: Anmeldung

4.1.2. A2: Spielvorbereitung

4.1.3. A3: Spiel starten

4.1.4. A4: Bomberman steuern

4.1.5. A5: Spiel verwalten

4.1.6. A6: Spiel beenden

4.1.7. A7: Scoreboard anzeigen

4.1.8. A8: Verbindungsproblem

4.1.9. A9: Spieler trennen

4.2. Grafische Benutzeroberfläche

4.2.1. Menü Bildschirm

4.2.2. Serverlisten Bildschirm

4.2.3. Login Bildschirm

4.2.4. Spiel Bildschirm

4.2.5. Steuerungs Bildschirm

4.2.6. Fehler Bildschirm

5. Produktcharakteristiken

5.1. Systemanforderungen

5.1.1. Hardware-Anforderungen

5.1.2. Softwareumgebung

5.1.3. Entwicklungsumgebung

5.2. Nicht-funktionale Anforderungen

1. Anforderungen

1.1. Funktionale Anforderungen

1. Der Verwaltungsserver kann die Spielserver anzeigen.(muss)
2. Der Client kann Spielserver auswählen und sich mit ihnen verbinden.(muss)
3. Der Client kann einen Namen auswählen.(muss, impliziert 2.)
4. Der Spielserver beugt doppelten Namen vor.(muss, impliziert 3.)
5. Unser Client funktioniert mit unserem Spielserver.(muss)
6. Unser Client ist mit anderen Spielservern kompatibel.(soll)
7. Unser Server ist mit anderen Clienten kompatibel.(soll)
8. Das Spiel kann gestartet werden.(muss)
9. Die Spieldaten werden korrekt zum Clienten übertragen.(muss)
10. Ein Bomberman kann laufen.(muss)
11. Ein Bomberman kann nicht über Bomben laufen.(muss)
12. Ein Bomberman kann nicht über Mauern laufen.(muss)
13. Ein Bomberman kann Bomben legen.(muss)
14. Ein Bomberman darf nicht zu viel Bomben legen.(muss)
15. Ein Bomberman kann Boni sammeln.(muss)
16. Ein Bomberman kann sterben.(muss)
17. Ein Bomberman mit Rüstung wird unverwundbar.(muss, impliziert 16.)
18. Ein Bomberman wird nach Ablauf der Rüstung wieder verwundbar.(muss, impliziert 17.)
19. Die Bomben haben den korrekten Explosionsradius des Spielers der sie platziert hat. (muss)
20. Bomben können brüchige Wände zerstören.(muss)
21. Bomben können feste Wände nicht zerstören.(muss)
22. Explodierende Bomben lösen Kettenreaktionen von Explosionen aus.(muss)
23. Ein Spieler mit Beschleunigungs-Powerup läuft schneller.(soll, Modus B)
24. Ein Spieler mit Tritt-Powerup kann Bomben wegstoßen.(soll, Modus B)

25. Getretene Bomben stoppen bei Kontakt mit Objekten.(soll, Modus B)
26. Die Explosionen der Superbombe wird nicht durch Mauern gestoppt. (soll, Modus B)
27. Das Maximum-Radius-Powerup setzt den Explosionsradius auf das Maximum. (soll, Modus B)
28. Ein Spieler mit Bombenläufer-Powerup kann Bomben überqueren.(soll, Modus B)
29. Inaktive Spieler explodieren nach 30s.(muss)
30. Die KI kann auf Wunsch eingeschaltet werden.(soll)
31. Die KI versucht Powerups zu sammeln.(soll, impliziert 30.)
32. Die KI schließt sich nicht selbst ein.(soll, impliziert 30.)
33. Die KI tötet sich nicht selbst.(soll, impliziert 30.)
34. Die KI läuft aus Bombenradien heraus.(soll, impliziert 30.)
35. Die KI erkennt Kettenreaktionen von Bomben.(soll, impliziert 30.)
36. Die KI versucht Gegner zu töten.(soll, impliziert 30.)
37. Die KI versucht Kettenreaktionen von Explosionen zu nutzen.(soll, impliziert 30.)
38. Das Ranking wird korrekt berechnet.(soll)
39. Das Ranking wird korrekt angezeigt.(soll)
40. Unser Client ist mit den Servern anderer kompatibel.(soll)
41. Unser Server ist mit den Client anderer kompatibel.(soll)
42. Besiegte Spieler können dem Spiel zuschauen.(muss)
43. Brüchige Mauern werden bei Spielstart generiert.(muss)
44. Powerups werden unter brüchigen Mauern generiert.(muss)
45. Powerups sind nicht sichtbar bevor die Mauer zerstört ist.(muss)
46. Spieler der alten Partie werden in die Lobby des nächsten Spieles eingefügt.(soll)
47. Am Server können Spieleinstellungen vorgenommen werden.(soll)

1.2. Nicht-funktionale Anforderungen

48. Das Handbuch steht als PDF-Datei zur Verfügung.(muss)
49. Das Handbuch ist grafisch visualisiert.(kann)
50. Die Fehlermeldungen klären aussagekräftig und verständlich über Probleme auf(soll)
51. Der Client erreicht mindestens 15 fps.(muss)
52. Der Client erreicht mindestens 30 fps.(soll, impliziert 4.)
53. Der Client erreicht mindestens 60 fps.(kann, impliziert 5.)
54. Die Verbindung wird mindestens 15 mal pro Sekunde aktualisiert.(muss)
55. Die Verbindung wird mindestens 30 mal pro Sekunde aktualisiert.(soll impliziert 7)
56. Die Verbindung wird mindestens 60 mal pro Sekunde aktualisiert.(kann impliziert 8)
57. Das Programm Bomberman wird mit der Programmiersprache Java programmiert.(muss)
58. Der Programmcode wird auf englisch verfasst.(muss)
59. Die Codebegleitenden Kommentare werden auf deutsch verfasst.(soll)
60. Kommentare werden mittels Javadoc Format erstellt.(soll)
61. Programmtests werden mittel Junit durchgeführt.(soll)

2. Produkteinsatz

Bei der Entwicklung des Spieles ist es zu Beginn wichtig, den grundlegenden Problembereich - also das klassische "Super Bomberman" von Nintendo - zu analysieren und zu verstehen.

2.1 Beschreibung des Problembereichs

Der Problembereich basiert auf dem klassischen Bombermanspiel, das 1993 herausgebracht wurde. Dabei soll es 2 bis 4 Spielern möglich sein, auf einem Spielfeld (Map) gegeneinander anzutreten. Ziel dabei ist es, die gegnerischen Spieler mit Bomben zu zerstören und dabei so viele Punkte wie nur möglich zu sammeln.

2.2 Modell des Problembereichs

(Abbildung 2.2.a: Modell des Problembereichs)

Die Abbildung 2.2.a zeigt das Domänenmodell, welches grob den Aufbau des Problembereichs darstellt.

Einer Menge Spielern (Player) ist es dabei möglich sich mit verschiedenen Servern zu verbinden, auf denen dann unterschiedliche Spiele angeboten werden.

Die Klasse "Game" modelliert dabei ein Bombermanspiel, an dem sich 2 bis 4 Spieler anmelden können, um gegeneinander anzutreten. Dazu werden ein Spielfeld ("GameField") und verschiedene Spielobjekte erstellt, die dann zusammen das Spiel komplettieren. Zu den Spielobjekten gehören ein einzelnes leeres Feld (emptyField), die Spielfiguren (Bomberman), Bomben (Bomb), verschiedene Boni (Boni) und natürlich Mauern (Wall). Diese werden dann auf einem Spielfeld platziert, um ein bestimmtes Spiel zu realisieren.

Die Spielfiguren können sich dann frei über das Spielfeld bewegen, Bomben legen, Mauern sprengen und Boni sammeln.

2.3. Glossar

Das Glossar beinhaltet die wichtigsten Begriffe, die in diesem Pflichtenheft Verwendung finden.

Muss-Kriterien:

Diese Kriterien werden in der finalen Software in jedem Fall verfügbar sein.

Soll-Kriterien:

Unser Entwicklerteam versucht diese Kriterien umzusetzen.

Kann-Kriterien:

Diese Kriterien werden eventuell nicht in der finalen Software verfügbar sein.

Spieler:

Die Person, die das Spiel Bomberman spielt.

Client:

Der Spieler benutzt einen Client (Software) auf seinen lokalen Rechner, um zu spielen.

Spielserver:

Server, der spielrelevante Daten berechnet und bereitstellt.

Verwaltungsserver:
Server, der die Spielserver auflistet und den Clients ermöglicht einen von diesen auszuwählen.

Spiel:

Synonym: Game

Eine Instanz von „Bomberman“, welche mit der Kommunikation von Server und Client spielbar ist.s

Administrator (des Spiels):
Spieler, welcher zuerst den Spielserver betreten hat und damit das recht hat, ein Spiel zu starten.

Grafischen (Benutzer) Oberfläche (GUI):
Die grafische Benutzeroberfläche visualisiert das Programm anschaulich und übersichtlich, um dem Benutzer die Bedienung zu vereinfachen.

Spielmenü:
Das Spielmenü zeigt die verschiedenen Optionen an, welche der Benutzer nach dem Starten des Spiels auswählen kann.

Internetprotokoll (IP):
Mit Hilfe des Internetprotokolls werden die Verbindungen und die Kommunikation zwischen Client und Server realisiert.

Multiplayer:

Synonym: Mehrspieler

Die Möglichkeit das Spiel „Bomberman“ über ein Netzwerk von Computern zu spielen.

Frames per second (FPS):
Anzahl gerenderte Bilder, die Pro Sekunde auf einem Bildschirm angezeigt werden.

Java:
Java ist die Programmiersprache in der unsere Software geschrieben wird.

Spielraum:

Synonym: Spiellobby

Wenn der Spieler sich mit einem Spielserver verbindet, werden ihm vor dem Spielstart alle anderen Mitspieler angezeigt, sowie seine Startposition und Farbe.

Spielobjekte: (GAME OBJECTS)

Bomberman:
Synonym: Spieler, Spielfigur
Die vom Benutzer gesteuerte Spielfigur. Er kann Bomben legen, sich auf dem Spielfeld bewegen und Power-Ups aufnehmen.

Zerstörbare/Feste Mauer: (WALL)
Ein Hindernis welches entweder mit einer Bombe zerstört werden kann, oder unzerstörbar ist.

Bombe: (BOMB)
Eine vom Spieler gelegte Bombe explodiert nach N Sekunden, und zerstört jede zerstörbare Mauer und jeden Spieler in ihrem Explosionsradius.

Boni:
Spezielle aufsammelbare Objekte im Spiel, welche den Spieler einen Vorteil verschaffen, zB. Explosionsradius erhöhen oder Explosionszeit verringern.

Feld: (EMPTY FIELD)
Auf einem Feld kann entweder eine Zerstörbare/Feste Mauer,eine Bombe,ein Powerup oder ein Spieler stehen.

Spielfeld: (GAMEFIELD)

Synonyme: Spielkarte

Besteht aus N*M Feldern. Auf einem Spielfeld wird das Spiel ausgetragen. Beinhaltet alle Spielobjekte.

Scoreboard:

Synonym: Ranking
Spieler bekommen Punkte für das Interagieren mit Spielobjekten. Diese Punkte werden aufaddiert und immer wieder aktualisiert. Am Ende eines Spiels kann man alle Punkte in einer geordneten Liste (Scoreboard) einsehen.

3. Vision

Unser Ziel ist die Entwicklung eines funktionierenden Bomberman-Spiels, das sich an dem von Nintendo 1993 herausgebrachten "Super Bomberman" orientiert. Dabei werden wir eine Client-Server-Anwendung erschaffen, die das verteilte Spielen des Bomberman-Games ermöglichen soll.

Wir werden eine Client-Applikation entwickeln, die es mehreren Spielern ermöglicht gegeneinander in einer Bomberman-Arena anzutreten. Dabei solle es den Usern möglich sein verschiedene Spieleserver auszuwählen, die von einem zentralen Verwaltungsserver angeboten werden. Weiterhin sollen sich die Benutzer auf den Spieleservern anmelden können, auf denen das konkrete Spiel abgehandelt wird. Die Spieleserver sorgen dann für die korrekten Interaktionen mit den Clients und überwachen das Spielgeschehen auf Korrektheit. Es soll den Spielern somit nicht möglich sein, unerlaubte Aktionen durchzuführen.
Des Weiteren werden wir eine Lobby erschaffen, die die bereits beigetretenen Spieler anzeigt, denn die Spiele sollen erst starten, wenn mindestens zwei User anwesend sind. Jedoch wir das ganze auf maximal vier Spieler pro Game begrenzt. In der Lobby wird es einen Spielleiter geben, der verschiedene Konfigurationen an der Arena vornehmen kann. Außerdem sollen Administratoren die Möglichkeit haben, Spiele zu starten.

Nachdem das Spielfeld und die Spielobjekte erstellt wurden, soll den jeweiligen Spielern in Echtzeit die relevanten Informationen und Änderungen des laufenden Spiels übermittelt werden. Es soll den Usern nach Möglichkeit ein flüssiges Spielerlebnis geboten werden, wobei wir uns auf mindestens konstante 20 FPS geeinigt haben.
Im Spiel bieten wir den Spielern die Möglichkeit, sich frei über das Spielfeld zu bewegen und Bomben zu legen. Sollten zerstörbare Mauern durch Bomben vernichtet werden, so wird die Möglichkeit bestehen, dass verschiedene Boni auftauchen, die die jeweiligen Spieler aufnehmen können. Zu den Standardboni, die wir implementieren werden, gehören extra Bomben, eine Erhöhung des Sprengradius und Rüstungen für die Bomberman. Eventuell werden wir noch weiter Boni im Verkauf der Entwicklung der Software implementieren.
Wird ein Spieler im Verlauf des Games getötet, so soll ihm die Möglichkeit geboten werden weiterhin zuzuschauen, aber nicht mehr aktiv am Verlauf teilzunehmen.

Sind alle Spieler bis auf einen tot oder die Zeit abgelaufen, so soll das Spiel beendet und das Ranking angezeigt werden. Dabei werden wir ein geeignetes Punktesystem implementieren, was entsprechende Aktionen des Nutzers unterschiedlich bewertet. Danach soll es den Usern möglich sein eine weitere Runde zu starten, wobei die alten Spieler bei der Lobbyerstellung dann neuen bevorzugt werden.

Auch Netzwerk- und Verbindungsprobleme werden wir in unserer Software behandeln. Der entsprechende Spieler soll bei auftretenden Verbindungsfehlern nach 30 Sekunden gleich einer Bombe gesprengt werden.

Weiterhin soll der Server das jeweilige laufende Spiel protokolieren und den Spielverlauf abspeichern.

Außerdem werden wir eine KI entwickeln, denn der Spieler soll die Möglichkeit haben durch eine KI abgelöst zu werden, die dann die Kontrolle für den Spieler übernimmt.

Schließlich soll es noch möglich sein, die Clients und Server unter den verschiedenen Projektgruppen austauschbar zu machen, was zu einer Einigung auf gleiche Interfaces führt.

4. Produktfunktionen

In diesem Abschnitt werden alle Funktionen behandelt welche dem Benutzer zur Verfügung stehen. Im Vordergrund stehen Anwendungsfälle (Use-Cases), Aktivitäten und Szenarien.

4.1. Use-Cases

Im folgenden werden die Produktfunktionen des Spiels „Bomberman“ vorgestellt. Einführend beschreibt ein Use-Case-Diagram die Funktionalität der zu entwickelnden Software. Anschließend wird jeder Anwendungsfall nochmal expliziert in seiner Funktionalität erläutert.

(Abbildung 4.1.a: Use-Case-Diagramm)

Mittels Aktivitäten Diagramm werden die Anwendungsfälle A1-A9 dargestellt und erläutert. Anschließend wird an Beispielszenarien ein bestimmter Ablauf im „Bomberman“-System durchgespielt.

4.1.1. A1: Anmeldung

Hier wird der Anmelde- und Verbindungsvorgang erklärt. Zuerst verbindet sich der Client mit dem Verwaltungsserver, welcher eine Liste der zu erreichenden Spielserver anzeigt. Die Anzahl der verbundenen Clients mit dem jeweiligen Spielserver werden ebenfalls angezeigt. Dann wählt der Client einen Spielserver und verbindet sich mit diesen. Als vorletzten Schritt wählt der Client noch einen Benutzernamen, unter dem er auf den Spielserver registriert wird. Der Spielserver bereitet nun das Spiel vor.

(Abbildung 4.1.b: Anmeldung)

4.1.2. A2: Spielvorbereitung

Nach dem betreten des Spielservers kommt der Spieler in einen sogenannten Spielraum (Lobby), wo er auf das Starten des Spiels wartet. Wenn sich der erste Spieler zu einem Spielserver verbindet, startet dieser Server einen Initialvorgang, indem er die Spielkarte erstellt und den ersten und jeden weiteren Spieler für alle anderen sichtbar macht; also seinen Benutzernamen anzeigt. Hinzu kommt, dass der Server jedem Spieler eine Startposition und Farbe zuweist.

(Abbildung 4.1.c: Spielvorbereitung)

4.1.3. A3: Spiel starten

Der Spielstart wird vom Spielserver berwaltet und ist von zwei Faktoren abhängig. Entweder die maximale Spieleranzahl (4) ist erreicht oder es sind mindestens zwei Spieler verbunden und der Spieler, der sich zu erst mit dem Spielserver verbunden hat (Spieladministrator) startet das Spiel. Wenn einer dieser zwei Optionen eingetreten ist, startet der Server das Spiel

(Abbildung 4.1.d: Spiel starten)

4.1.4. A4: Bomberman steuern

Das Steuern der Spielfigur im Spiel ist vom Client abhängig. Es werden die Eingaben des Spielers vom Client eingelesen und zunächst geprüft, ob der Benutzer sich durch eine KI ersetzen lassen möchte. Danach generiert der Client entweder einen Spielzug der KI oder nimmt die Eingaben des Spielers und sendet diese an den Spielserver weiter. Dieser verarbeitet dann - wie in A5 beschrieben - die Eingaben und schickt daraufhin die Spielinformationen an den Client zurück. Als letzten Schritt zeichnet der Client nun die aktuelle Spielkarte und gibt diese auf dem Bildschirm aus.

(Abbildung 4.1.e: Bomberman steuern)

4.1.5. A5: Spiel verwalten

Die Spielverwaltung läuft vollständig auf dem Spielserver ab. Dieser prüft immer wieder, ob noch genügend Spieler im Spiel vorhanden sind und berechnet danach aus den Eingaben, die ihm jeder Client sendet das Spielfeld neu. Nach dem errechnen des neuen Spielfelds sendet der Spielserver die aktuellen Spielfelddaten an alle Clients.

(Abbildung 4.1.f: Spiel verwalten)

4.1.6. A6: Spiel beenden

Um ein Spiel zu beenden, muss entweder ein oder kein Bomberman auf der Spielkarte verbleiben. Falls eine dieser genannten Bedingungen eintritt, beendet der Server die Spielverwaltung und zeigt die erreichten Punkte jedes einzelnen Spielers in einem Scoreboard an. Danach werden die verbundenen Clients in die Spiellobby zurückgebracht

(Abbildung 4.1.g: Spiel beenden)

Szenario für Use Cases „Spiel starten“ und „Spiel verwalten“

Zweck:

Das Szenario beschreibt, wie sich Verbindungsprobleme auf den Spielverlauf/-ausgang auswirken.

Akteure und Ausrüstung:

Zwei beliebige Personen, die Zugang zum Spiel haben und über eine funktionierende Internetverbindung verfügen.

Szenario:

Spieler 1 loggt sich als Erster ins Spiel ein und wird dadurch vom Server zum Administrator des aktuellen Spiels gemacht. Er hat nun die Möglichkeit das Spiel bei Anwesenheit von minindestens einem weiteren Spieler zu starten.
Nun Spieler 2 loggt sich ein.
Der Server prüft die Anzahl der Spieler und erkennt, dass sich zwei Spieler eingeloggt haben, was die Mindestanforderung für den Start des Spiels erfüllt.
Spieler 1 sieht den Startbutton in der Mitte seines Bildschirms grün aufleuchten und drückt ihn.

Der Server erkennt einen Spielstart durch den Administrator und startet das Spiel.
Spieler 1 befindet sich an Position 1 und Spieler 2 an Position 2. Es gibt kein Durchkommen zu einander, weil der gesamte mittlere Bereich des Spielfeldes durch sprengbare und nichtsprengbare Mauern blockiert ist.
Spieler 1 fängt an Bomben zu legen und Mauern zu zerstören. Bei Spieler 2 treten Verbindungsprobleme auf. Somit kan derr Server keine Daten mehr von ihm erhalten, was als „keine Aktion“ angesehen wird.
Dadurch, dass keine Aktionen vom Spieler 2 kommt, wird sein „keine Aktion“-Timer nicht mehr auf Null gesetzt.
Nach dem Timeout hat es Spieler 1 noch nicht geschafft Spieler 2 aus dem Spiel zu befördern.
Da der Timeout abgelaufen ist, wird Spieler 2 vom Server automatisch zur Explosion gebracht.
Weil sich jetzt nur noch ein Spieler auf dem Spielfeld befindet, bricht der Server das aktuelle Spiel ab, berechnet die Statistik inklusive des Ranges und sendet schleißlich die Daten an alle Clients.
Spieler 1 hat durch das Sprengen von Mauern, und dadurch, dass er als Letzter im Spiel war, die meisten Punkte gesammelt und somit gewonnen.

4.1.7. A7: Scoreboard anzeigen

Scoreboard wird vom Spielserver schon während des Spieles gepflegt. Jeder Spieler bekommt Punkte für das Interagieren mit Spielobjekten, also beispielsweise das Zerstören einer Mauer oder Töten eines gegnerischen Bombermans. Am Ende eines Spiels werden diese Punkte in einem Ranking von entsprechend der Spieleranzahl aufgelistet.

(Abbildung 4.1.h: Scoreboard anzeigen)

4.1.8. A8: Verbindungsproblem

Verbindungsprobleme werden vom Spielserver behandelt. Hat sich ein Client seit mehr als 30 Sekunden nicht bei dem Spielserver gemeldet, so wird zuerst unterschieden, ob das Spiel gestartet ist oder ob man noch auf den Start des Spiels wartet. Beim ersten Fall wird ein Countdown von drei Sekunden gesetzt, woraufhin dann der Bomberman des Spielers explodiert, welcher die Verbindung verloren hat. Dieser scheidet damit aus dem Spiel aus. Im zweiten Fall (Spiel hat noch nicht begonnen) wird der Client mit der fehlenden Verbindung aus der Spielerliste der Spielvorbereitung entfernt.

(Abbildung 4.1.i: Verbindungsproblem)

4.1.9. A9: Spieler trennen:

Bei der Trennung eines Spielers handelt es sich einfach um das Entfernen eines Clients vom Spielserver, falls dieser die Verbindung mit dem Spielserver absichtlich oder auch unabsichtlich beendet haben sollte. Danach schickt der Server keine Daten mehr an diesen Client.

Beispielablauf (aus Nutzersicht)

Der folgende Abschnitt soll einen zusammenfassenden Überblick über den gesamten Ablauf des Spiels, vom Anfang bis zum Ende geben und die Nutzeroberfläche aus Nutzersicht beschreiben.

Nachdem der Spieler das Spiel gestartet hat erscheint ein Menü, dass ihn auffordert, einen Namen anzugeben und die Möglichkeit bietet, das Spiel zu verlassen. Der Spieler, nennen wir ihn einfach Spieler 1, gibt seinen Namen ein und bestätigt diesen mit der Eingabetaste. Daraufhin kommt er ins nächste Hauptmenü.

Zu sehen sind in folgender Reihenfolge von unten nach oben die Optionen: „Multiplayer“, „How to play“ und „Exit“. Exit bietet wieder die Möglichkeit, das Spiel zu verlassen. Durch die Auswahl von „How to play“ hat der Spieler die Möglichkeit, die Steuerung einzusehen und diese entsprechend zu konfigurieren. Nehmen wir an das Spieler 1 mit der Steuerung zufrieden ist, so kommt er durch Klicken von „Back to menu“ wieder ins Hauptmenü.

Nun wählt Spieler 1 „Multiplayer“ und bekommt eine Liste von Spielservern, die zur Auswahl stehen, angezeigt. Die Liste zeigt außerdem, wie viele Spieler bereits auf dem Server sind und welche Karte verwendet wird. Spieler 1 kann sich einem Spiel anschließen, entscheidet sich aber dafür einen leeren Server - Server 1 - zu nehmen, was ihm zum Spielleiter des aktuellen Servers macht. Da er momentan noch alleine ist, kann er kein Spiel starten, doch nach wenigen Sekunden verbindet sich Spieler 2 mit dem Server 1. Dadurch bekommt Spieler 1 die Möglichkeit das Spiel zu starten, was er schließlich auch tut.

Nun wird das Spiel gestartet und beide Spieler sehen die gleiche Karte und zwei Bomberman, die jeweils eine eigene Farbe haben. Weiterhin sind die Anzahl der Spieler, der eigenen Punktestand und die Spieldauer zu sehen. Auch gibt es wieder einen Exitbutton, falls ein Spieler das Spiel vorzeitig verlassen möchte.

Beide Spieler beginnen nun durch Eingabe von Befehlen ihre Bomberman zu bewegen.

Durch das Legen von Bomben und Sprengen von Mauern, sammeln beide Spieler Punkte und können evtl. Boni sammeln. Diese werden zufällig auf den Stellen erzeugt wo zuvor Mauern gesprengt wurden. Nach einigen Sprengungen entscheidet sich Spieler 1 die KI für ihn spielen zu lassen, was er auch durch eine im „How to play“-Menü festgelegte Taste kann. Die KI fängt an selbstständig Mauern zu sprengen und Boni zu sammeln und versucht Spieler 2 zu eliminieren,

was ihr auch schließlich gelingt. Die aktuelle Karte wird geschlossen, da sich nur noch Spieler 1 im Spiel befindet und beide Spieler bekommt ihren Punktestand und die Platzierung angezeigt.
Spieler 1 hat die meisten Punkte sammeln können und ist somit Sieger und auf dem ersten Platz.

4.2. Grafische Benutzeroberfläche

Das Spiel wird über eine grafische Benutzeroberfläche (GUI) seitens des Clients gesteuert. Die folgenden Seiten werden einen Einblick in den ersten Entwurf dieser GUI aufzeigen. Die Abbildungen dienen ausschließlich als Beispiel bzw. Entwurf und können im finalen Produkt abweichend sein.

Beiliegend finden Sie außerdem eine PDF in der Sie die Verlinkungen zwischen den einzelnen Bildschirmen selbst ausprobieren und damit besser verstehen können.

4.2.1. Menü Bildschirm

Nachdem das Programm gestartet wurde, wird in der GUI der Menü Bildschirm angezeigt. Dort hat man drei Möglichkeiten zur Auswahl.

Unter dem Menüeintrag „Play“ gelangt man zur Liste der verfügbaren Server (Abb. 4.2.b).

Der Button „How to play“ führt zum Spielprinzip und zur Erklärung der Steuerung(Abb. 4.2.e).

Schließlich beendet „Exit“ das Programm und man kehrt zu Windows zurück.

(Abbildung 4.2.a: Menü Bildschirm)

4.2.2. Serverlisten Bildschirm

Im Serverlisten Bildschirm (Abb. 4.2.b) werden alle verfügbaren Spielserver in einer Tabelle angezeigt. Zu jedem Server wird neben dem Servernamen auch der Spielmodus, die Karte und die Spieleranzahl aufgeführt. Wenn man einen Spielserver ausgewählt hat kann man mit ENTER eine Verbindung zu genau diesem Server (Abb. 4.2.d) aufbauen.

Der Button „Back to Menu“ führt zurück in das Hauptmenü (Abb. 4.2.a).

(Abbildung 4.2.b: Serverlisten Bildschirm)

4.2.3. Login Bildschirm

Nachdem man sich mit einem Server verbunden hat wird in der GUI der Login Bildschirm angezeigt. Dort hat man die Option, das Spiel zu beenden oder einen Benutzernamen einzugeben. Dieser Name wird zunächst vom System (Server) auf unzulässige Zeichen und Verfügbarkeit geprüft. Wenn kein Fehler auftritt wird man zum Spiel Bildschirm (Abb. 4.2.d) weitergeleitet.

(Abbildung 4.2.c: Login Bildschirm)

4.2.4. Spiel Bildschirm

Dieser Bildschirm (Abb. 4.2.d) zeigt wie es aussieht, wenn man sich in einem Spiel auf einem Spielserver befindet. Dieses Bild gilt als Beispiel wie der Spiel Bildschirm im finalen Spiel aussehen könnte. Da die Entwicklung noch nicht so weit vorangeschritten ist, kann man noch keine genaue Abbildung zeigen.

(Abbildung 4.2.d: Spiel Bildschirm)

4.2.5. Steuerungs Bildschirm

Im Steuerungs Bildschirm (Abb. 4.2.e) wird das Spielprinzip und -ziel kurz und knapp erklärt. Außerdem kann man alle Tastenbelegungen einsehen.

Mit dem „Back to Menu“ Button gelangt man zurück ins Hauptmenü (Abb. 4.2.a).

(Abbildung 4.2.e: Steuerungs Bildschirm)

4.2.6. Fehler Bildschirm

Falls es einen Fehler geben sollte (z.B. Verbindungsabbruch) wird dieser Bildschirm

(Abb. 4.2.f) angezeigt.

(Abbildung 4.2.f: Fehler Bildschirm)

5. Produktcharakteristiken

In diesem Abschnitt wird erst die Systemumgebung spezifiziert und anschließend werden die nicht-funktionalen Anforderungen definiert, welche an die Software gestellt werden.

5.1. Systemanforderungen

5.1.1. Hardware-Anforderungen

Betriebssystem: Windows 7

Arbeitsspeicher: minimal 1GB

Festplattenspeicher: minimal 100 MB

5.1.2. Softwareumgebung

Betriebssystem: Windows 7

Aktuelle Java Runtime Environment für das entsprechende Betriebssystem (Version 1.8 oder höher)

5.1.3. Entwicklungsumgebung

Programmierung: Eclipse Neon.1

Laufzeitumgebung: Java Runtime Environment (1.8.0_65-b17)

Tests: JUnit Framework

5.2. Nicht-funktionale Anforderungen

Name: Programmiersprache

Typ: Pflege

Beschreibung: Die Programmiersprache der Software „Bomberman“ ist Java

Name:Entwicklungssprache

Typ: Pflege

Beschreibung: Code Kommentare werden auf Deutsch verfasst, der Code selber ist komplett in Englisch gehalten.

Name: Format der Quellcode-Kommentare

Typ: Pflege

Beschreibung: Die Kommentare im Code werden im Javadoc-Format erstellt.

Name: Umsetzung der Modultests
Typ: Pflege

Beschreibung: Modultests werden mit Hilfe des JUnit Frameworks erstellt und durchgeführt.