



Projekt: JBombberman  
DomainAnalyse

Pascal Kistler  
Silvan Adrian  
Fabian Binna

## 1 Änderungshistorie

Datum	Version	Änderung	Autor
09.03.15	1.00	Erstellung des Dokuments	Gruppe
20.03.15	1.01	Vollendung des Dokuments	Fabian Binna

## Inhaltsverzeichnis

<b>1</b>	<b>Änderungshistorie</b>	<b>2</b>
<b>2</b>	<b>Einführung</b>	<b>4</b>
2.1	Zweck . . . . .	4
2.2	Gültigkeit . . . . .	4
2.3	Übersicht . . . . .	4
<b>3</b>	<b>DomainModell</b>	<b>5</b>
3.1	Strukturdiagramm . . . . .	5
3.2	Konzeptbeschreibung . . . . .	6
3.2.1	Game . . . . .	6
3.2.2	Party . . . . .	6
3.2.3	Player . . . . .	6
3.2.4	Sprite . . . . .	7
3.2.5	AnimatedSprite . . . . .	7
3.2.6	Bomberman . . . . .	7
3.2.7	PowerUp . . . . .	7
3.2.8	Bomb . . . . .	8
3.2.9	Explosion . . . . .	8
3.2.10	DestroyableBlock . . . . .	8
<b>4</b>	<b>Systemsequenzdiagramme</b>	<b>9</b>
4.1	UC01: Bomberman spielen . . . . .	9
<b>5</b>	<b>Systemoperationen</b>	<b>10</b>
5.1	Contracts . . . . .	10

## **2 Einführung**

### **2.1 Zweck**

Dieses Dokument beschreibt die Domainanalyse für das Projekt JBombberman.

### **2.2 Gültigkeit**

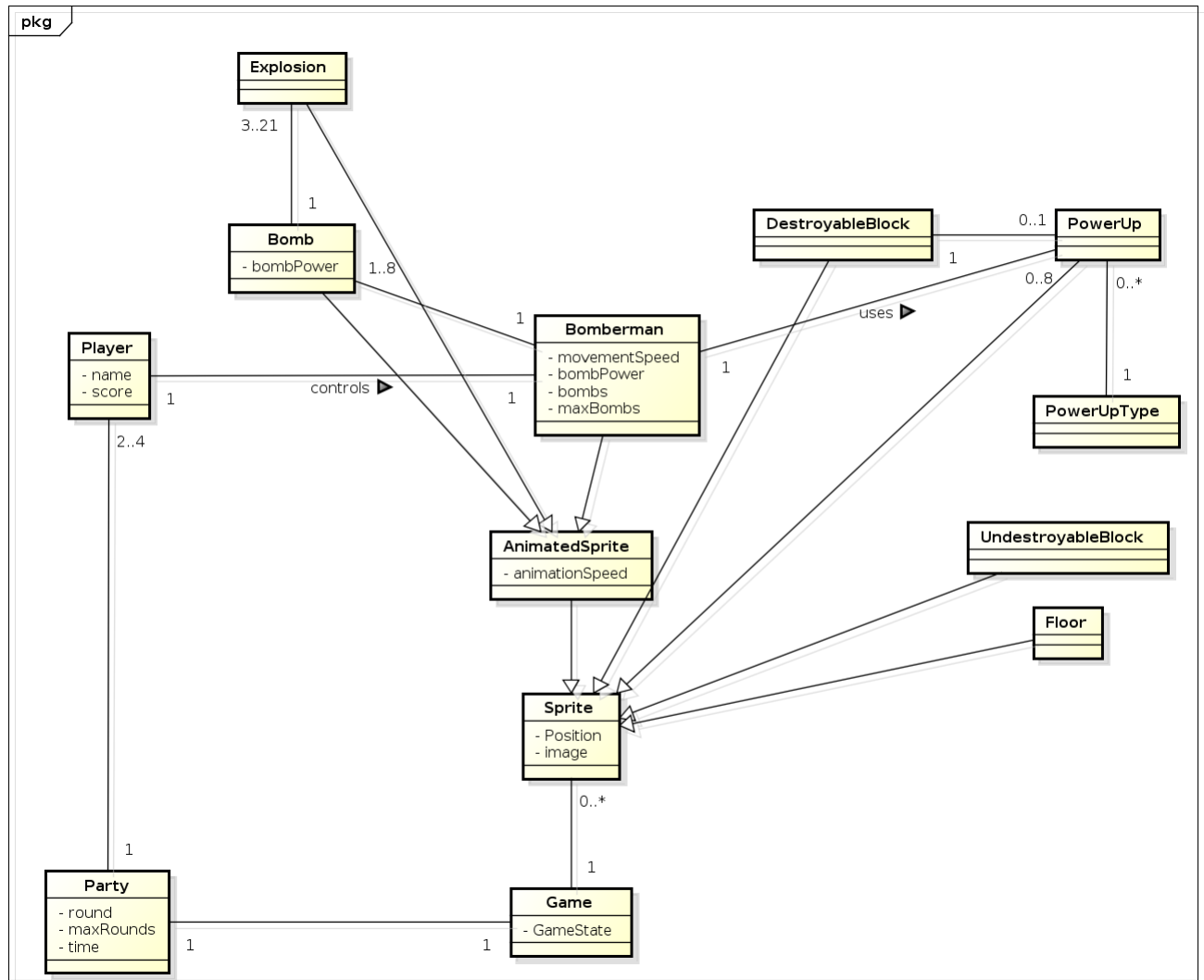
Dieses Dokument ist während des ganzen Projekts gültig und wird laufend aktualisiert.

### **2.3 Übersicht**

Dieses Dokument soll eine erste Analyse der Software zeigen. Das Strukturdiagramm stellt die wichtigsten Klassen dar, die miteinander interagieren müssen. Das Systemsequenzdiagramm beschreibt die wichtigsten Abläufe der Use Cases.

### 3 DomainModell

#### 3.1 Strukturdiagramm



powered by Astah

## 3.2 Konzeptbeschreibung

### 3.2.1 Game

Die Klasse Game ist die oberste Klasse und kontrolliert den Ablauf des Spiels.

#### **Gamestate**

Der GameState weist auf den aktuelle Spielestatus hin. Je nachdem in welchem Status sich das Spiel befindet werden andere Routinen durchlaufen.

### 3.2.2 Party

Die Party beinhaltet alle Spieler die aktuell am Spiel teilnehmen, und kontrolliert Zeit sowie die Spielrunden.

#### **round**

Die aktuelle Spielrunde.

#### **maxRounds**

Maximale Anzahl Runden die gespielt werden müssen.

#### **time**

Time zählt die Zeit von 3 Minuten herunter. Bei Ende der Zeit kriegen alle Bombermans die vollen PowerUps.

### 3.2.3 Player

Der Player dient zur identifizierung der Teilnehmer. Jedem Spieler ist ein Bomberman zugewiesen. Die Messages werden über die jeweilige Playerinstanz dem korrekten Bomberman zugewiesen.

#### **name**

Speichert den Namen des Spielers. Wird auch zur identifizierung verwendet.

#### **score**

Speichert die aktuelle Punktezahl des Spielers.

### 3.2.4 Sprite

Die Spriteklasse implementiert die Grundvoraussetzungen für jedes Spielelement. Jedes visuelle Element muss von der Spriteklasse erben.

#### **Position**

Position ist eine Klasse, die x und y Koordinaten speichert.

#### **image**

Image ist ein String, der den Namen des Bildes speichert, welcher intern zur Identifizierung dient und bei einem ResourceManager die Referenz zum BufferedImage abholen kann.

### 3.2.5 AnimatedSprite

Das AnimatedSprite erbt von der Spriteklasse und implementiert weitere Funktionen, die ein animiertes Objekt ermöglicht.

#### **animationSpeed**

Die Geschwindigkeit mit der durch die einzelnen Bildteile geschaltet wird.

### 3.2.6 Bomberman

Die Bombermannklasse beschreibt den Zustand und die Fähigkeiten der Spielfigur.

#### **movementSpeed**

Wie schnell kann sich der Bomberman fortbewegen.

#### **bombPower**

Wie weit reicht der Sprengradius.

#### **bombs**

Anzahl Bomben die der Bomberman zeitgleich auf dem Spielfeld verteilen kann.

#### **maxBombs**

Maximale Anzahl Bomben die der Bomberman tragen kann.

### 3.2.7 PowerUp

Das PowerUp beschreibt die Art der Fähigkeit die den Bomberman verbessert.

### **3.2.8 Bomb**

Die Bombklasse repräsentiert die Bombe auf dem Spielfeld.

#### **bombPower**

Die Reichweite der Explosionsarme.

### **3.2.9 Explosion**

Beschreibt einen Teil eines Explosionsarms. Die bombPower der Bombe definiert wie viele Explosionen in jede Richtung erstellt werden müssen.

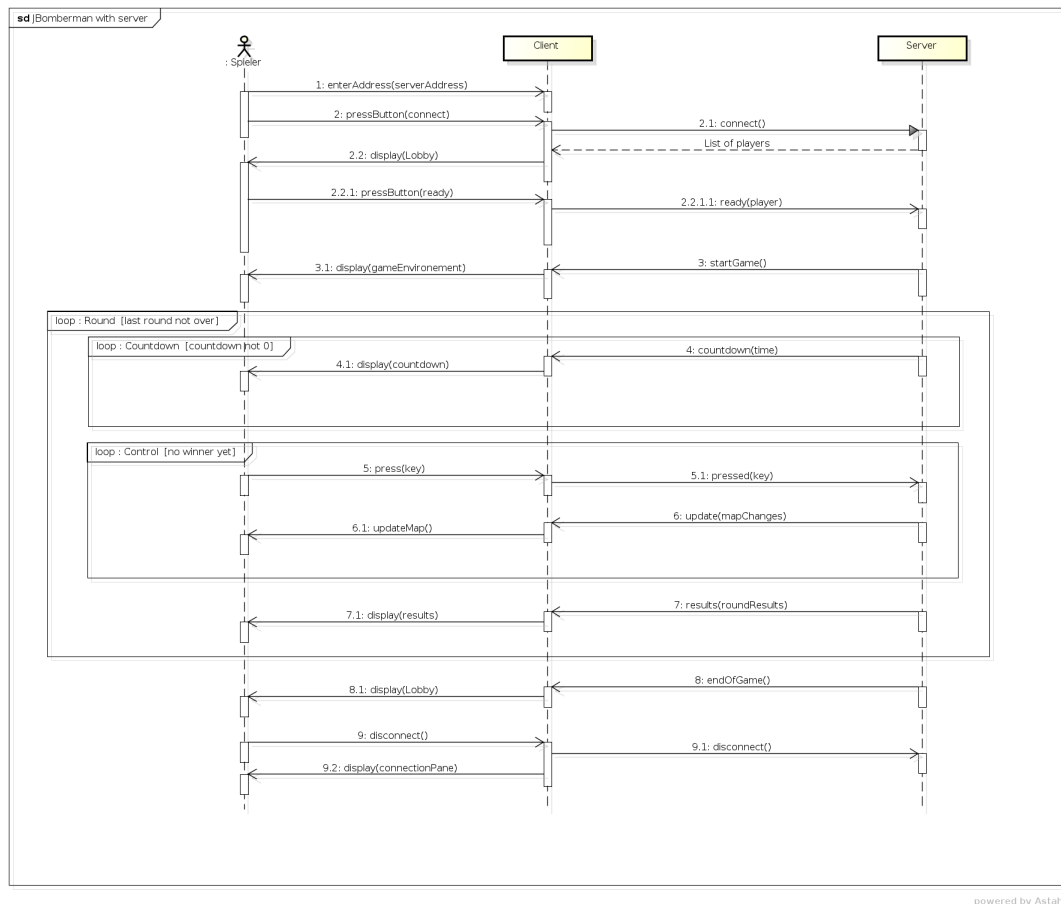
### **3.2.10 DestroyableBlock**

Der DestroyableBlock kann von einer Bombe zerstört werden und hinterlässt manchmal ein PowerUp.



## 4 Systemsequenzdiagramme

### 4.1 UC01: Bomberman spielen



## 5 Systemoperationen

- connect(server: Address)
- ready(player: Player)
- startGame()
- countdown(time: integer)
- pressed(key: Message)
- update(mapChanges: Message)
- results(results: Array)
- endOfGame()
- disconnect()

### 5.1 Contracts

<b>Operation</b>	connect(server : Address)
<b>Cross References</b>	UC01
<b>Preconditions</b>	Eine Serveradresse wurde eingegeben
<b>Postconditions</b>	<ul style="list-style-type: none"><li>• Der Client ist mit dem Server verbunden</li><li>• Der Server kennt den Client</li></ul>
<b>Operation</b>	ready(player : Player)
<b>Cross References</b>	UC01
<b>Preconditions</b>	Der Client befindet sich in der Lobby
<b>Postconditions</b>	<ul style="list-style-type: none"><li>• Der Status des Clients wurde auf dem Server auf <b>ready</b> gesetzt</li></ul>
<b>Operation</b>	startGame()
<b>Cross References</b>	UC01
<b>Preconditions</b>	Alle Client sind ready
<b>Postconditions</b>	<ul style="list-style-type: none"><li>• Bei den Clients wurde die Spielumgebung gestartet</li><li>• Der Server befindet sich im Gamemode</li></ul>

<b>Operation</b>	countdown(time : integer)
<b>Cross References</b>	UC01
<b>Preconditions</b>	Das Spiel wurde gestartet
<b>Postconditions</b>	<ul style="list-style-type: none"><li>• Ein Countdown wird den Spielern angezeigt</li></ul>
<b>Operation</b>	pressed(key : Message)
<b>Cross References</b>	UC01
<b>Preconditions</b>	Der Client befindet sich im Spiel und der Countdown ist abgelaufen
<b>Postconditions</b>	<ul style="list-style-type: none"><li>• Der Event ist beim Server angekommen</li><li>• Der Event befindet sich in der Eventqueue</li></ul>
<b>Operation</b>	update(mapChanges : Message)
<b>Cross References</b>	UC01
<b>Preconditions</b>	Der Client befindet sich im Spiel
<b>Postconditions</b>	<ul style="list-style-type: none"><li>• Die Informationen über Änderungen in der Map wurden an den Client übertragen</li><li>• Der Client hat seine Map angepasst</li></ul>
<b>Operation</b>	results(results : Array)
<b>Cross References</b>	UC01
<b>Preconditions</b>	Die Runde ist vorbei
<b>Postconditions</b>	<ul style="list-style-type: none"><li>• Die Clients haben die Rangliste erhalten</li><li>• Die Clients zeigen die Rangliste an</li></ul>
<b>Operation</b>	endOfGame()
<b>Cross References</b>	UC01
<b>Preconditions</b>	Alle Runden sind vorbei
<b>Postconditions</b>	<ul style="list-style-type: none"><li>• Alle Clients wurden über das Ende des Spiels informiert</li><li>• Die Clients zeigen die Rangliste an</li></ul>

<b>Operation</b>	disconnect()
------------------	--------------

<b>Cross References</b>	UC01
-------------------------	------

<b>Preconditions</b>	Der Client befindet sich in der Lobby
----------------------	---------------------------------------

<b>Postconditions</b>	<ul style="list-style-type: none"><li>• Der Client ist nicht mehr mit dem Server verbunden</li><li>• Der Server hat den Client aus der Clientliste entfernt</li></ul>
-----------------------	---