

# Konzeptpapier Bomberman?

Bösl, Florian  
f.boesl@oth-aw.de

Chernysheva Anastasia  
a.chernysheva@oth-aw.de

Kohl Helge  
h.kohl@oth-aw.de

Korinth Patrice  
p.korinth@oth-aw.de

Porsch Philipp  
p.porsch@oth-aw.de

**Zusammenfassung—Ziel des Papers ist es ein Spiel zu entwickeln bei dem mehrere Spieler in Echtzeit in ihrem Browser gegeneinander antreten können. Das Spielprinzip soll dem Spiel „Bomberman“ ähneln.**

## I. EINFÜHRUNG

Bereits 1983 kam das labyrinthartig aufgebaute Computerspiel Bomberman, das vom japanischen Hersteller Hudson Soft (wurde 2012 von Konami übernommen) entwickelt wurde, auf den Markt. Seitdem erschienen bis heute zahlreiche offizielle und daran angelehnte Spiele. Dabei hat sich das grundsätzliche Spielprinzip nie verändert. Ein beliebter Spielmodus ist dabei der Multiplayer-Modus, in welchem alle Mitspieler besiegt werden müssen. Im Rahmen der Vorlesung Big Data und Cloud-basiertes Computing soll nun cloudbasiert ein Online-Multiplayer-Spiel, das dem zeitlosen Klassiker Bomberman ähnelt, entwickelt werden.

## II. VERWANDTE ARBEITEN

Nachdem in der Einleitung bereits die Motivation erläutert wurde, werden hier verwandte Arbeiten und Systeme vorgestellt. Bei der Recherche wurden einige Bomberman inspirierte Online-Spiele gefunden, die auch kostenlos verfügbar sind. Spiel [?], Bomber-Friends, stellt dabei eine komplexe Variante des Spiels dar, aufgrund von RPG-Elementen, die den klassischen Spielstil mit der Möglichkeit erweitert, aus zerstörten Boxen Edelsteine zu gewinnen, mit denen sich Verbesserungen freischalten lassen. Power-Ups sind ebenfalls erhältlich. Bomber-Mouse [?] hingegen ist eine simplere Version, in der sich der Charakter nur sehr langsam fortbewegt, was möglicherweise nicht beabsichtigt ist. Auch die Fenstergröße passt sich nicht automatisch an und das Game Design weist allgemein deutliche Mängel auf. Playing with Fire 2 [?] weist die meisten Individualisierungsmöglichkeiten in der Lobby auf. So kann die Anzahl der Spieler und die der Gegner festgelegt werden sowie die Auswahl des Levels. Zusätzlich läuft ein Timer ab, der die Schwierigkeit erhöht. Allstar-Blast [?] ist eine MMO-Variante von Ubisoft, in der in der Probe 78 Spieler gleichzeitig gegeneinander antraten. Die Besonderheit hierbei ist, dass das Spielfeld mit der Zeit schrumpft. Power-Ups sind hier ebenfalls verfügbar.

## III. ANFORDERUNGEN

### A. Anforderung 1

(mvp) Als Spieler möchte ich ein Spiel eröffnen können, um ein Spiel spielen zu können.

### B. Anforderung 2

(mvp) Als Spieler möchte ich mich auf dem Spielfeld bewegen können, um mit der Umgebung zu interagieren.

### C. Anforderung 3

(mvp) Als Spieler möchte ich Bomben legen können, damit sie später explodieren können.

### D. Anforderung 4

(mvp) Als Spieler möchte ich, dass die Bomben explodieren und verschwinden, um ihren Effekt visuell zu erleben (und das Spiel zu gewinnen).

### E. Anforderung 5

Als Spieler möchte ich mit Hindernissen kollidieren können, um Einschränkungen in der Spielwelt zu haben.

### F. Anforderung 6

Als Spieler möchte ich, dass die Bomben bei ihrer Explosion Hindernisse zerstören, um die Spielwelt zu manipulieren.

### G. Anforderung 7

Als Spieler möchte ich einem Spiel beitreten können, um in einem bestehenden Spiel mitspielen zu können.

### H. Anforderung 8

Als Spieler möchte ich, dass meine Spielfigur bei Kontakt mit anderen Spielfiguren kollidiert, um den Bewegungsspielraum des Gegners einzuschränken.

### I. Anforderung 9

Als Spieler möchte ich, dass der Kontakt von Bombenexplosionen und Spielfiguren erkannt wird, um diese zu zerstören.

### J. Anforderung 10

Als Spieler möchte ich, dass ich wenn der letzte Gegner besiegt ist, das Spiel beendet wird, um den Abschluss der Spielziels zu erkennen.

### K. Anforderung 11

Als Spieler möchte ich jederzeit spielen können, um bei hohen Spielerzahlen keine Wartezeiten zu haben.

### L. Anforderung 12

Als Anbieter möchte ich, dass Spiele bei zu langer Spieldauer beendet werden, um Rechenkapazitäten zu sparen.

### *M. Anforderung 13*

Als Spieler möchte ich mit mehr als einem weiteren Spieler spielen können, um den Komplexitätsgrad des Spieles zu erhöhen.

### *N. Anforderung 14*

Als Spieler möchte ich eine dynamische Spielfeldgröße, um den Umfang des Spiels zu erhöhen (nochmal drübernachdenken).

### *O. Anforderung 15*

Als Spieler möchte ich meinen Spielernamen festlegen können, um einen Wiedererkennungswert zu schaffen.

### *P. Anforderung 16*

Als Spieler möchte ich ein Leaderboard einsehen können, um meine Leistung kompetitiv einordnen zu können.

### *Q. Anforderung 17*

Als Spieler möchte ich meine Spielfigur personalisieren können, um meine Figur an meinen Geschmack anzupassen.

### *R. Anforderung 18*

Als Spieler möchte ich Power-Ups einsammeln können, um das Spielerlebnis aufzuwerten.

## LITERATUR

- [1] Bomber-Friends [Online] <https://www.spielaffe.de/Spiel/Bomber-Friends> (visited on May. 03, 2022)
- [2] Bomber-Mouse [Online] [https://de.y8.com/games/bomber\\_mouse](https://de.y8.com/games/bomber_mouse) (visited on May. 03, 2022)
- [3] Playing-with-Fire [Online] <https://www.kibagames.com/Game/Playing-with-Fire-2> (visited on May. 03, 2022)
- [4] Allstar-blast [Online] <https://www.kibagames.com/Game/All-Star-Blast> (visited on May. 03, 2022)

## IV. METHODEN

### *Todo*

#### *Über die Daten*

In einer lokalen Graphdatenbank werden die aktuellen Filme und Serien diverser Streaminganbieter (Netflix, Amazon Prime, Hulu, Disney+) abgespeichert. Außerdem werden diese durch Informationen aus der IMDB (Internet Movie Database) ergänzt. Mithilfe dieses Datenbestandes soll es möglich sein, komplexe Abfragen zu formulieren. Durch den SPARQL-Endpoint der DBpedia ist der Zugriff auf weitere Details möglich.

#### *Backend*

Als Backend wird ASP.NET Core verwendet. Dieses von Microsoft entwickelte Framework hat sich in der Welt der Microservices in den letzten Jahren etabliert. Über noch nicht definierte Graph Query Languages soll auf die verwendete Graphdatenbank bzw. DBpedia zugegriffen werden können, um Informationen zu den Filmen und Schauspielern auszulesen.

#### *Frontend*

Für die Interaktion mit dem Benutzer wird React verwendet. Über eine REST-Schnittstelle des Backends kann auf benötigte Informationen zugegriffen werden.