

# BOMBER MAN



**Designdokument**

**PRIMA WiSe 20/21**

**Prof. Jirka Dell'Oro-Friedl**

**vorgelegt von  
Daniel Meisler**

**Matrikel: 263236**



## Inhaltsverzeichnis

Beschreibung:.....	3
Ursprung:.....	3
Der Prototyp:.....	4
Level:.....	5
Items:.....	6
Künstliche Intelligenz:.....	7
Problematik:.....	7
Anforderungen:.....	8

## Beschreibung:

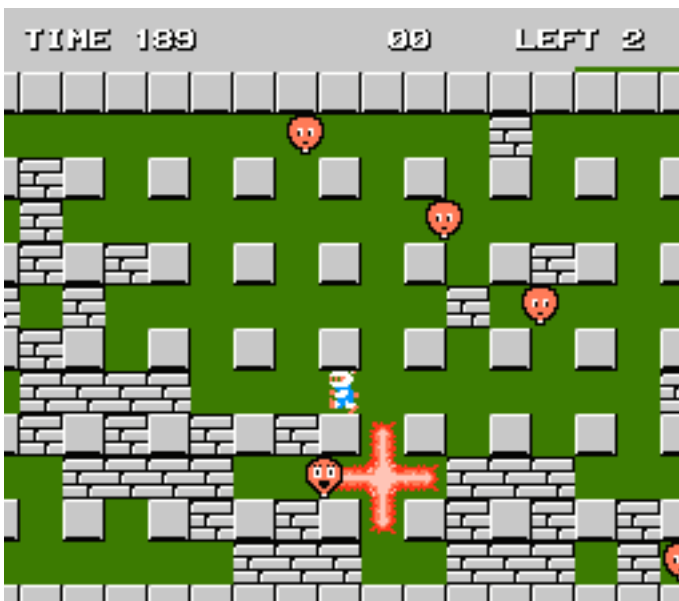
Das Spiel stellt einen Klon des allzu bekannten Bomberman - Genres dar. Man steuert eine Spielfigur in einem Labyrinth und versucht alle Feinde zu besiegen um in das nächste Level zu kommen. Dieser Klon ist ein Prototyp und somit in allen Aspekten klein gehalten. Programmiert wurde es in TypeScript/JavaScript und der Engine FUDGE. FUDGE ist eine open-source game engine von der Hochschule Furtwangen im Schwarzwald.



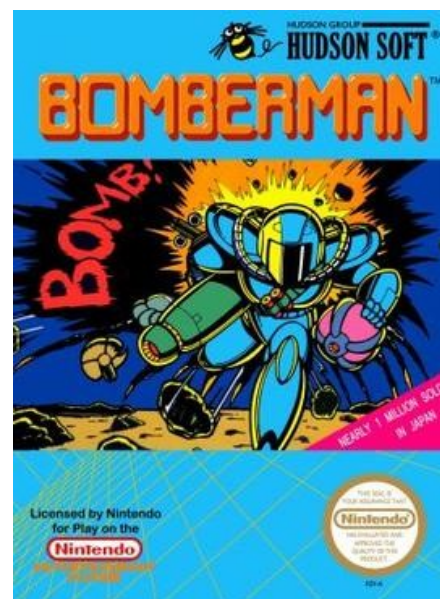
Prototyp 2021 (PRIMA)

## Ursprung:

Das Original stammt aus dem Jahre 1983 von Hudson Soft und wurde für den Heimcomputer herausgebracht. Im Jahre 1985 kam es auch auf der NES raus und bekam somit seine eigene Serie, welche bis heute aktiv ist. Seither gibt es zahlreiche Klone und verschiedene Variationen dieses Spiels und zählt zu den Klassikern schlechthin.



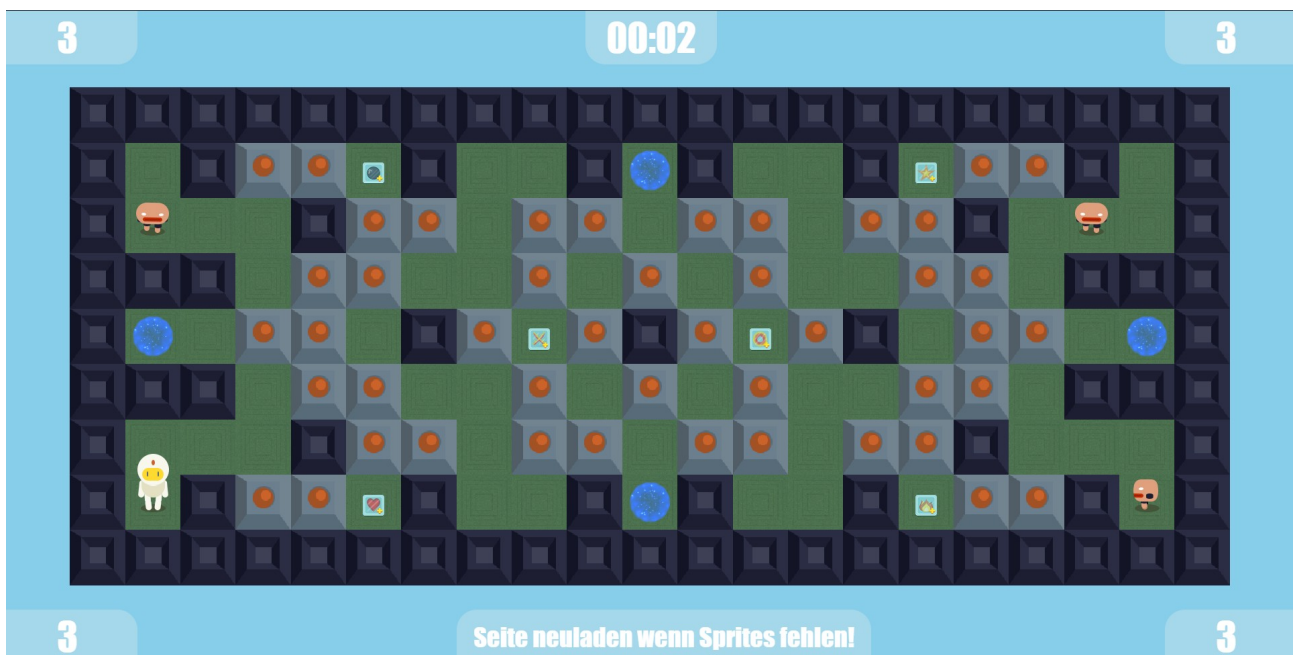
Bomberman 1985 (NES)



1987 Nord-Amerika (NES box art)

## Der Prototyp:

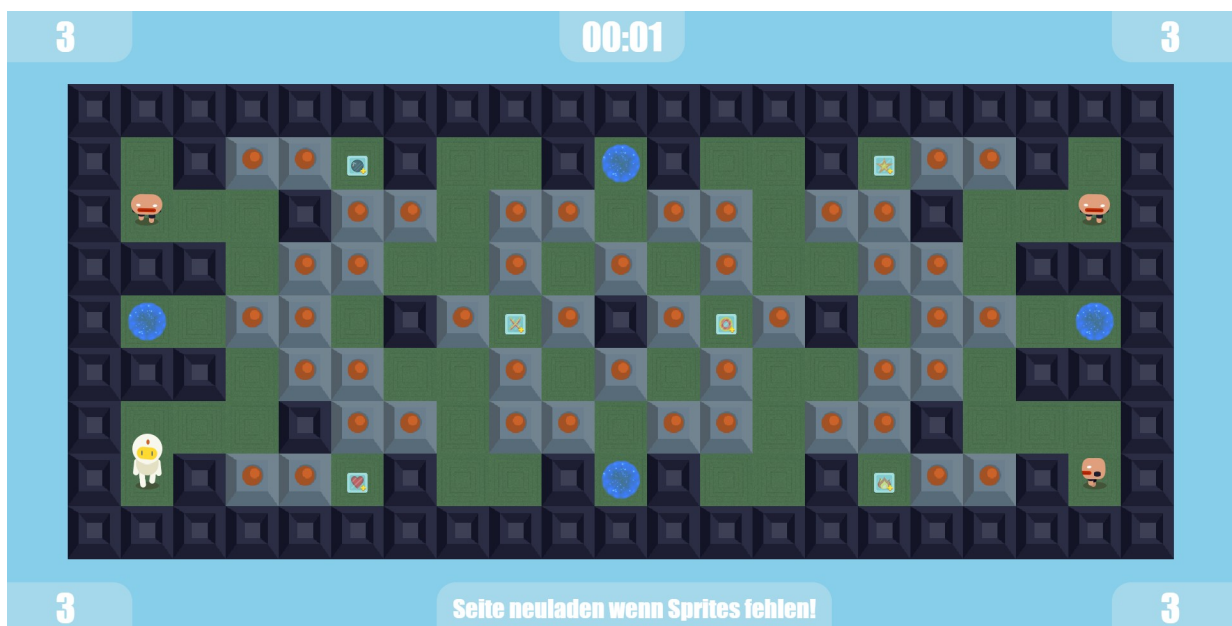
Die Abgabe ist eine funktionstüchtiges Spiel. Es werden alle Grundkonzepte eingehalten. Der Spieler kann sich mit den Tasten WASD bewegen und mit der Leertaste Bomben platzieren. Es gibt Blöcke, welche durch das Auftreffen der Bomben entweder zerstört werden oder eben nicht. Items können eingesammelt werden und mit verschiedenen „Fähigkeiten“ genutzt werden. Zur Variation wurden noch Portale eingefügt, welche das Teleportieren über die platzierten Stellen erlauben. Es befinden sich drei Gegner auf dem Spielfeld, welche zu besiegen gilt. Die Gegner besitzen eine sehr simple KI, welche sie dazu bringt den Spieler anzugreifen. Zur Auswahl gibt es drei verschiedene Level, welche man mit einer externen Datei (JSON) auswählen kann. Auf dem HUD kann man die aktuelle Spielzeit und wie viele Leben alle Spieler besitzen ablesen. Die Sprites gab es zur freien Benutzung als Download im Internet. Fehlende Sprites wurden selbst erstellt. Das gleiche gilt für die Sounds und Musik.



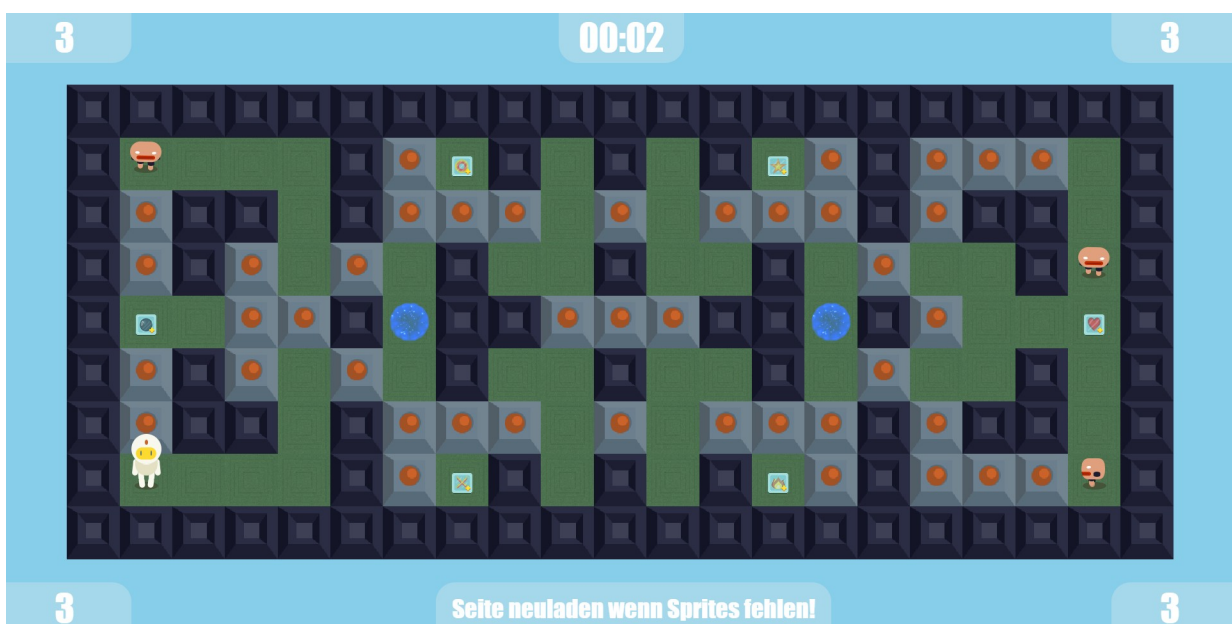
Bomberman Prototyp (PRIMA)

**Level:**

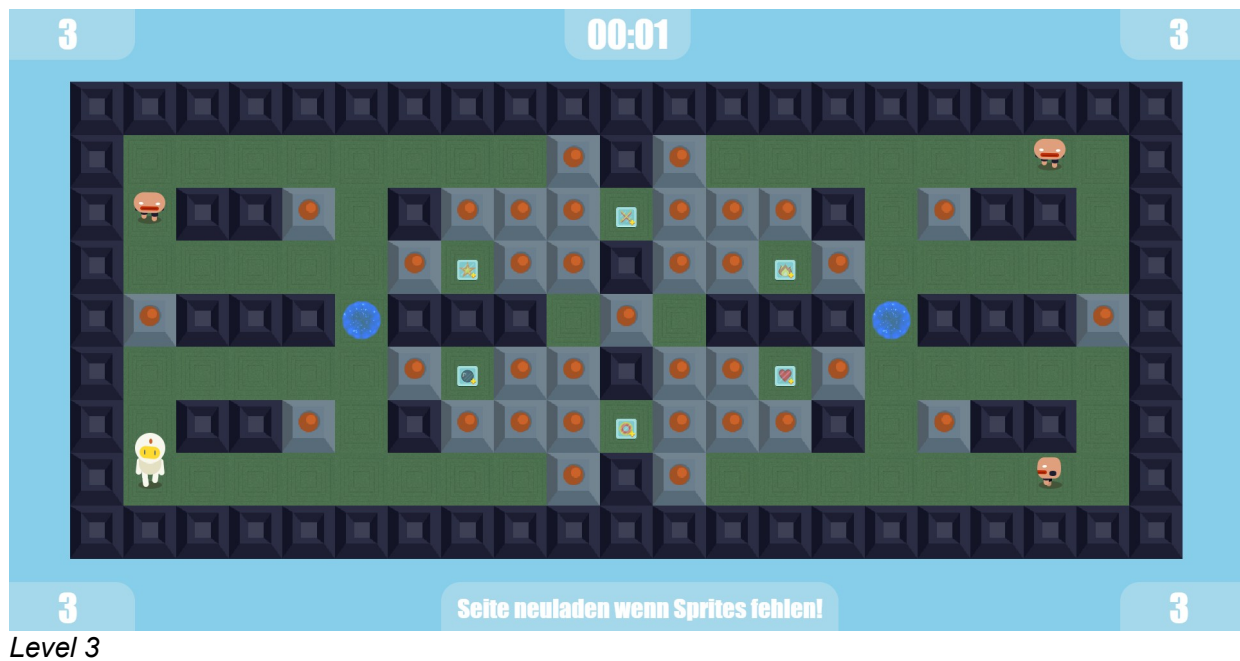
Das Spiel beinhaltet, wie bereits beschrieben, drei Level. Diese lassen sich aus dem JSON auswählen. Es wurde darauf geachtet die Level unterschiedlich wie möglich zu gestalten. Die Items und Portale sind anders platziert, als auch die zerstörbaren und unzerstörbaren Blöcke. Bei einem Level wurde mehr auf die Mechanik des frei „bombens“ geachtet, während das andere mehr an das Kämpfen oder probieren der KI gedacht ist. Die Level fallen aus Zeitgründen alle in der selben Größe aus.



Level 1



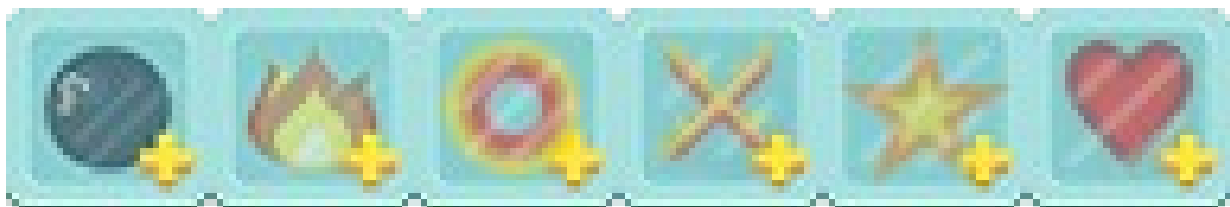
Level 2



## Items:

Es befinden sich sechs Items in diesem Spiel. Von links nach rechts:

- Extra Bombe: Erhöht die Anzahl der gleichzeitig platzierten Bomben.
- Extra Flamme: Erhöht die Reichweite in alle Richtungen.
- Kreis-Explosion: Ermöglicht einmalige Explosion durch Wände in Kreisform.
- Diagonal-Explosion: Ermöglicht einmalige Explosion durch Wände in die Diagonalen.
- Unsterblichkeit: Man ist für 10 Sekunden gegen alle Angriffe geschützt.
- Extra Leben: Das eigene Leben wird um eins erhöht.

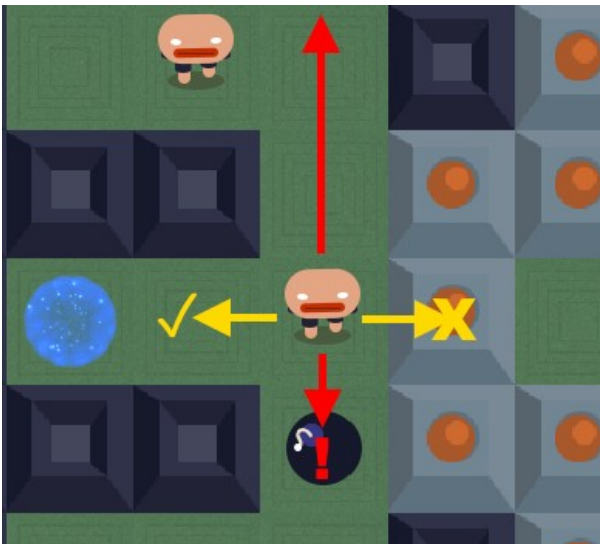


Items

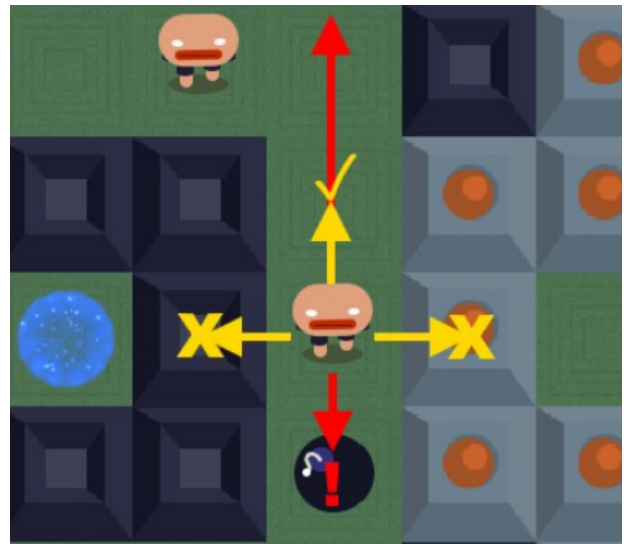


## Künstliche Intelligenz:

Die KI ist einfach gehalten und besitzt zwei States. In der einen befindet sich die KI im HUNT-STATE. Sie nimmt den Spieler als Ziel und versucht zu ihm zu gelangen. Sobald er den Spieler in Sicht hat und zwar im Abstand der Flammendistanz ohne Hindernisse, legt er eine Bombe und wechselt in den FLEE-STATE. In diesen wird auch gewechselt sobald seine dauerhafte Überprüfung eine Bombe in der selben Distanz entdeckt. Der Feind überprüft dann jeweils, je nach dem auf welcher Achse er sich befindet, erst die benachbarten Felder und flieht dort hin. Sollten diese blockiert sein geht er noch einen Schritt entgegen der Bombe und überprüft dort die benachbarten um möglichst immer um Ecken zu laufen. Diesen Vorgang wiederholt er, bis er einen sicheren Platz gefunden hat.



*FLEE-STATE: Überprüft benachbarte Felder und einer ist frei. Er geht auf dieses Feld.*



*FLEE-STATE: Überprüfung benachbarter Felder und beide sind blockiert. Er geht ein Feld nach hinten und wiederholt den Vorgang.*

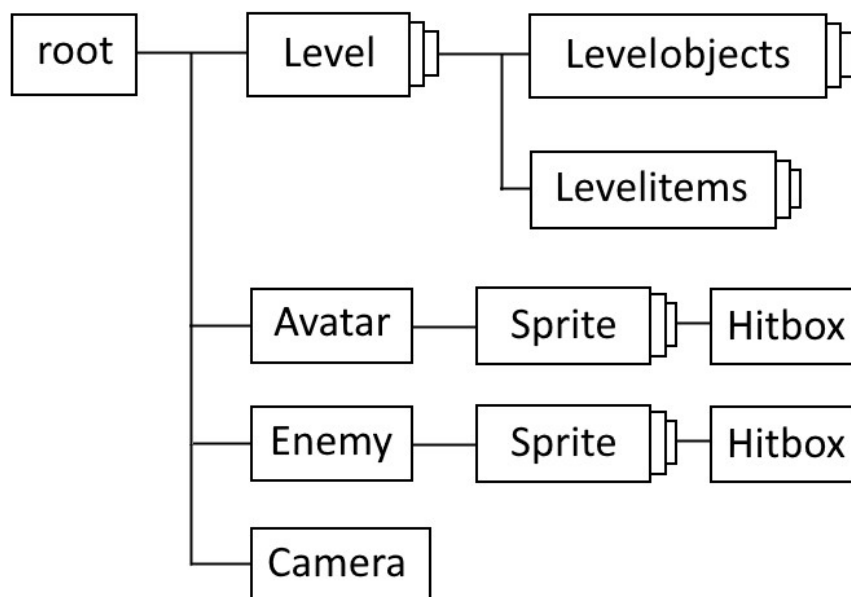
## Problematik:

Die KI wurde meinerseits ein wenig vom Aufwand unterschätzt. Sie ist simpel gehalten und ist spielbar, jedoch nicht intelligent genug um wirklich den Spielspaß zu erhalten. Sie spielt und reagiert, jedoch ohne richtige Intelligenz. Sie jagt den Spieler, kann aber dabei hängen bleiben, weil sie sich keinen Weg frei bomben kann. Sie sucht sich nicht den idealsten Weg, sondern den Kürzesten. Die Angriffsweise ist auch simpel und kann ausgetrickst werden. In ungeeigneten Fällen tötet sich KI selbst, falls sie den Spieler in einer Sackgasse trifft. Leider blieb mir keine Zeit übrig um diese Punkte zu überarbeiten und die Feinde intelligenter zu machen.

# Anforderungen:

1	Nutzerinteraktion	- Steuerung - Bomben -> Wände zerstören/Gegner töten
2	Objektinteraktion	- Spieler-Kollision mit Wänden - Bomben-Kollision mit Wänden/Spielern - Items aufsammeln
3	Objektanzahl variabel	- Levelgenerierung - Gegner - Items oder ähnliches
4	Szenenhierarchie	- Diagramm
5	Sound	- Hintergrundmusik - Sounds für Bomben, Spieler, Aktionen usw.
6	GUI	- Leben, Zeit
7	Externe Daten	- JSON
8	Verhaltensklassen	- GameObject -> z.B Collision - Avatar -> z.B Controls - Enemy -> z.B AI - Items -> z.B Effekte der Items
9	Subklassen	- GameObject -> Spieler, Enemy, Items
10	Maße & Positionen	- Mehr oder weniger Grid - Gridgröße - Nullpunkt unten links, zwei Achsen.
11	Event-System	- load, loop, controls usw.

## Checkliste für Leistungsnachweis



## Szenenhierarchie