



A Media Engineering Project

// Sommersemester 2012
// Bachelor Media Engineering
// Abschlusspräsentation am 20.07.2012
// Postbankgebäude / Keßlerstr. 1 / Raum 4020

Projektteilnehmer

- Sebastian Adam
- Sergej Bjakow
- Michael Kao
- Pavlina Pavlova
- Maximilian Seyfert

0. INHALT

1. Ausgangssituation

2. Soll - Zustand

3. Projektorganisation

4. Sprite Engine

5. Umsetzung

6. Zusammenfassung

0. INHALT

1. Ausgangssituation

2. Soll - Zustand

3. Projektorganisation

4. Sprite Engine

5. Umsetzung

6. Zusammenfassung

0. INHALT

1. Ausgangssituation

2. Soll - Zustand

3. Projektorganisation

4. Sprite Engine

5. Umsetzung

6. Zusammenfassung

0. INHALT

- 1. Ausgangssituation
- 2. Soll - Zustand
- 3. Projektorganisation
- 4. Sprite Engine**
- 5. Umsetzung
- 6. Zusammenfassung

0. INHALT

- 1. Ausgangssituation
- 2. Soll - Zustand
- 3. Projektorganisation
- 4. Sprite Engine
- 5. Umsetzung**
- 6. Zusammenfassung

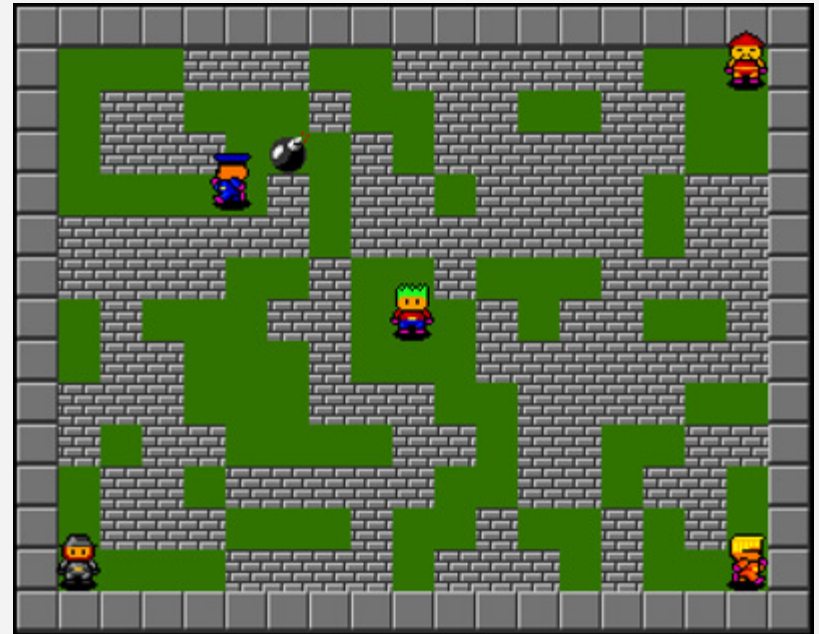
0. INHALT

-
1. Ausgangssituation
 2. Soll - Zustand
 3. Projektorganisation
 4. Sprite Engine
 5. Umsetzung
 - 6. Zusammenfassung**

1. AUSGANGSSITUATION

1.1 MasterBlaster auf der AMIGA

- 1994 von Alexander Ivanof entwickelt
- unterstützt bis zu 5 Spieler
- Ziel ist es Bomben zu legen und dadurch andere Gegenspieler vom Spielfeld zu entfernen
- Zudem gibt es diverse Extras, die nach dem Sprengen eines Mauernblocks zufällig erscheinen
- Zusätzlich zu Extras gibt es Geldmünzen mit denen man sich Extras in einem Shop nach Ende einer Runde kaufen kann
- unterstützt werden 1 - 5 Spieler (mittels Tastatur und Joysticks)

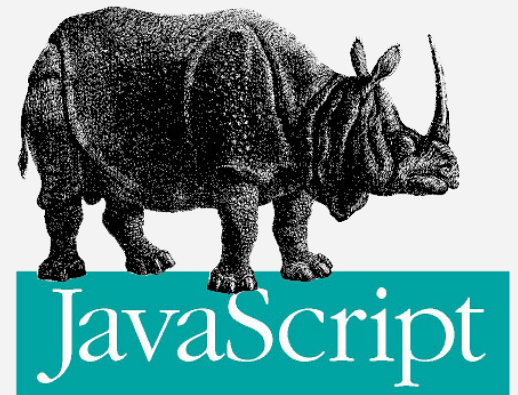


1. AUSGANGSSITUATION

1.2 Projektidee

- Rekonstruktion des Spieleklassikers mittels **Reverse Engineering**
- Umsetzung durch neue Webtechnologien
HTML 5 + CSS + JavaScript + jQuery
- **Reverse Engineering (Auszug aus Wikipedia):**
"(...)bezeichnet den Vorgang, aus einem bestehenden, fertigen System oder einem meistens industriell gefertigten Produkt durch Untersuchung der Strukturen, Zustände und Verhaltensweisen, die Konstruktionselemente zu extrahieren. Aus dem fertigen Objekt wird somit wieder ein Plan erstellt.(...)"

HTML CSS



2. SOLL - ZUSTAND

Muss - Kriterien

- Spielbares Produkt von "ReMasterBlaster" muss vorliegen
- 2 Spielfiguren müssen auf dem Spielfeld mit einer Tastatur steuerbar sein
- Es müssen Bomben gelegt und Wände gesprengt werden können
- Ein Gegenspieler muss vom Spielfeld entfernt werden, wenn dieser innerhalb des Bombenradius einer Bombe getroffen wird

Soll - Kriterien

- Es sollen diverse Extras (Goodies) mit eingebunden werden
 - Es soll vor Spielbeginn ein Konfigurationsmenü geben
- Szenenablauf: Startscreen --> Konfiguration --> Spiel --> Hall of Fame

Kann - Kriterien

- mehrere Spieler, andere Eingabegeräte, Shop für Extras

3. PROJEKTORGANISATION

3.1 Tools

- Projektablauf und Zeitplanung mit GanttProject
- Projektumsetzung in öffentlichem git-basierten Content Management System
- Facebook Gruppe für kurzfristige Absprachen
- Entwicklungsblog
<http://www.remasterblaster.com>
- Wöchentliche Projektmeetings und Besprechungen
- Google Kalender



3. PROJEKTORGANISATION

3.2 Zeitlicher Ablauf

Planungsphase

26.03. - 10.04.2012

- Einarbeitung in Sprite Engines
- TestBenches mit diversen Engines
- Erstellung des Projektplans
- Definition des Lastenhefts

Entwicklungsphase

10.04. - 03.07.2012

- Implementierung der Spielelogik
- Realisierung der Spieldetails
- Definition verschiedener Szenen
- Merging von Spielelogik und Szenen

Dokumentationsphase

03.07. - 17.07.2012

- Erstellung der Dokumentation, PrStA's, Abschlusspräsentation

3. PROJEKTORGANISATION

3.2 Zeitlicher Ablauf

Planungsphase

26.03. - 10.04.2012

- Einarbeitung in Sprite Engines
- TestBenches mit diversen Engines
- Erstellung des Projektplans
- Definition des Lastenhefts

Entwicklungsphase

10.04. - 18.07.2012

- Implementierung der Spielelogik
- Realisierung der Spieldetails
- Definition verschiedener Szenen
- Merging von Spielelogik und Szenen

+ 17 Tage
↓

Dokumentationsphase

18.07. - 03.08.2012

- Erstellung der Dokumentation, PrStA's, Abschlusspräsentation

4. SPRITE - ENGINE

4.1 Definition

- *"Eine Sprite - Engine ist eine Bibliothek, die eine Zusammenstellung von vorgefertigten Funktionen bereit stellt, welche es ermöglichen Sprites in Spielen zu verwenden."*

4.2 Kriterien

- HTML5 / JavaScript kompatibel
- möglichst performant
- z-Index Tiefenwerte müssen berücksichtigt werden
- gute Dokumentation
- vorhandene Tutorials

4. SPRITE - ENGINE

4.3 Crafty JavaScript Sprite - Engine

Key Features:

- Entities & Components:
Einfache Möglichkeit Spielelemente ohne Vererbung zu organisieren
- Unterstützt Canvas und DOM
- Eventbinding:
Custom Events die einfach an jeder Stelle zu jedem Zeitpunkt ausgelöst werden können

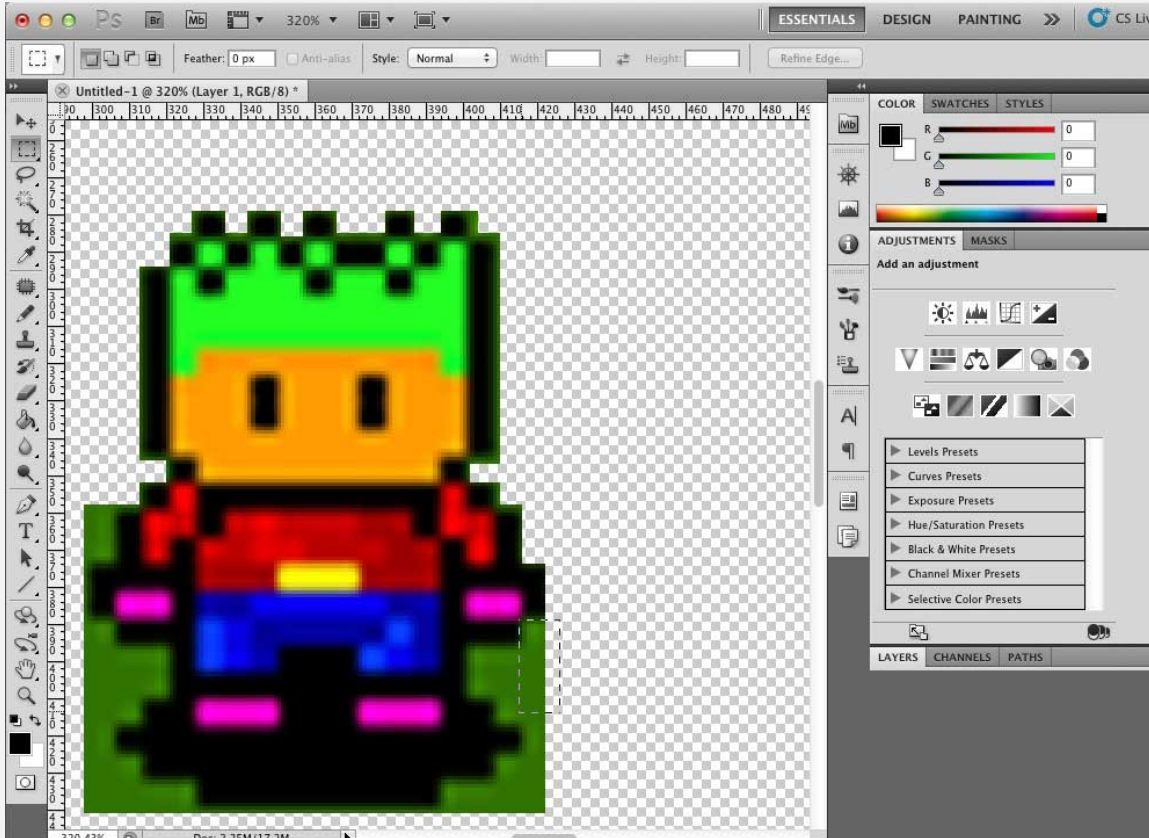


Weitere Goodies (von Crafty):

- aktive Community mit wachsendem Forum
- Pures JavaScript - keine Magie!
- sehr gute Dokumentation und kleine Tutorials

5. UMSETZUNG

5.1 Reverse Engineering



5. UMSETZUNG

5.1 Reverse Engineering



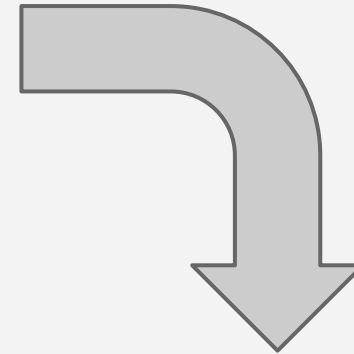
5. UMSETZUNG

5.2 Testumgebung



5. UMSETZUNG

5.2 Testumgebung



**Festlegung auf Google
Chrome als "der Browser"
für ReMasterBlaster**



5. UMSETZUNG

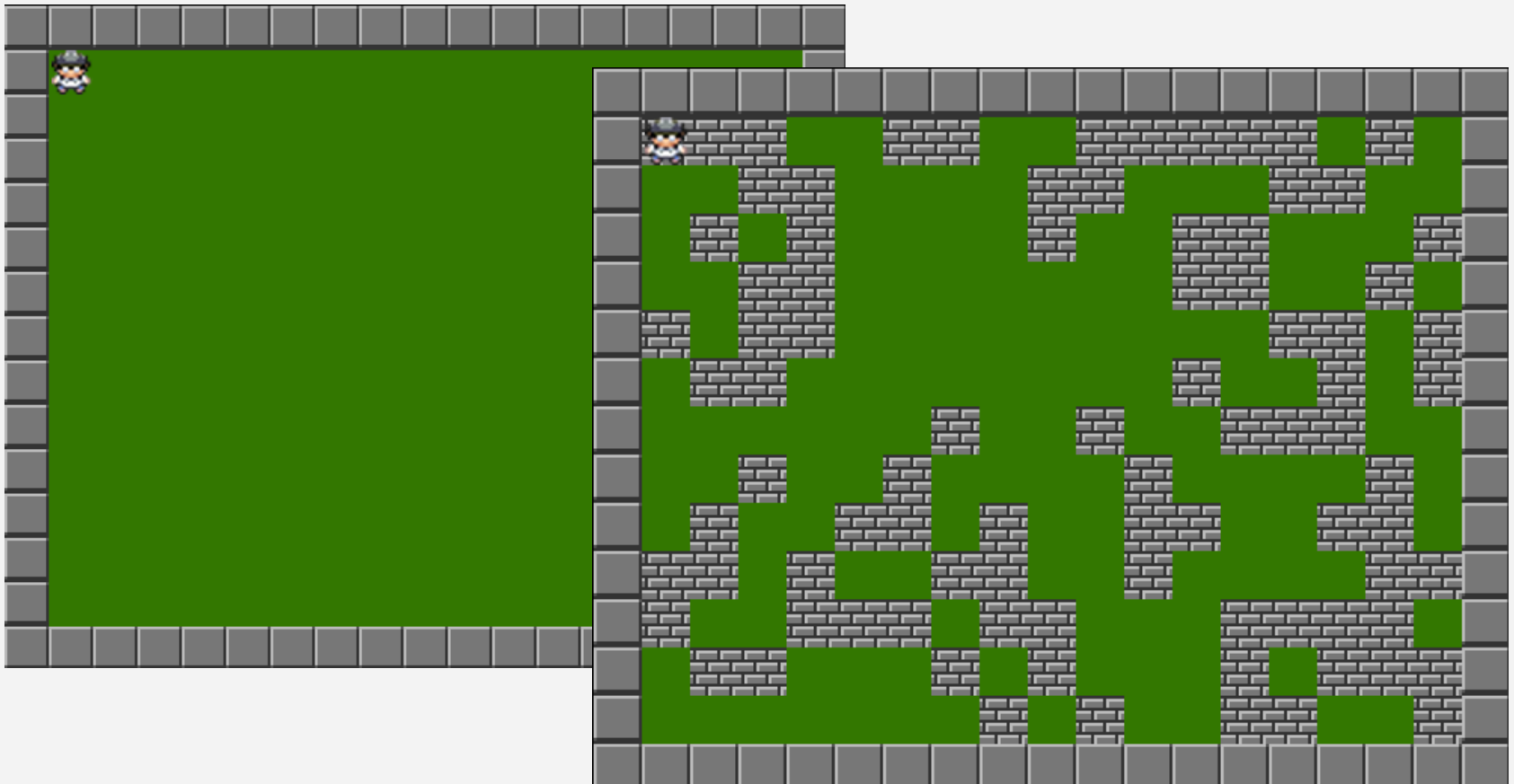
5.3 Implementierung der Details

- **Erstellung der Welt**
- Bewegung
- Animation
- Kollisionserkennung
- Goody Erstellung - Auswirkungen
- über 1000 Zeilen Code

5. UMSETZUNG

5.3 Implementierung der Details

Erstellung der Welt



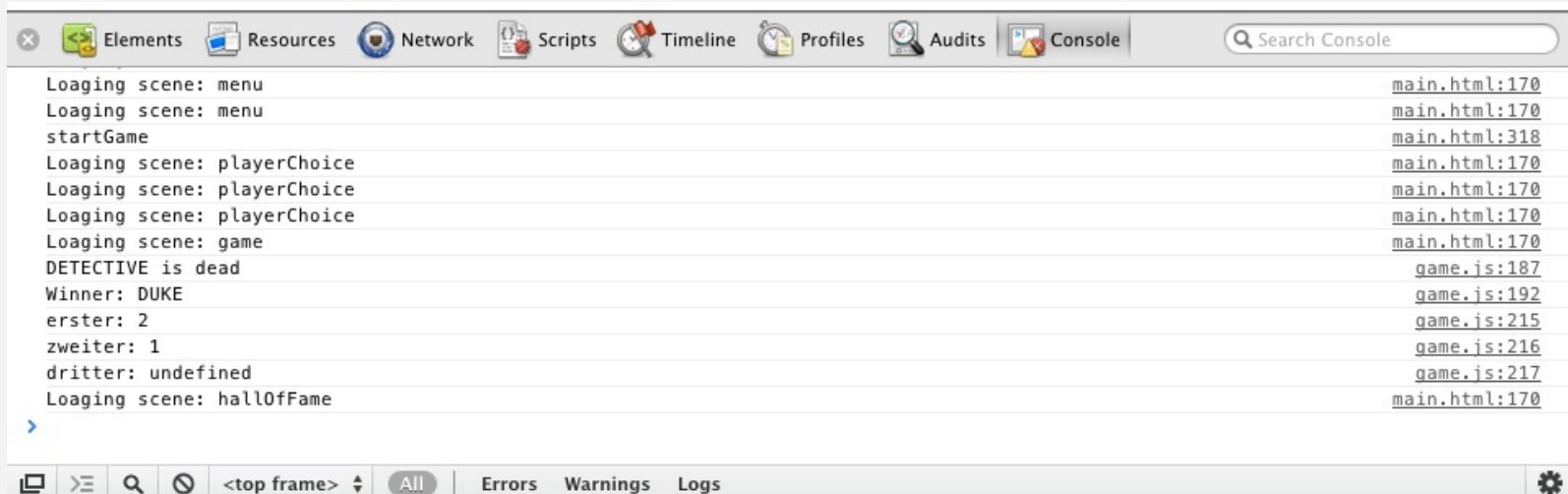
5. UMSETZUNG



5. UMSETZUNG

5.3 Implementierung der Details

"Debugging" mit der JavaScript Console



```

Loading scene: menu                                main.html:170
Loading scene: menu                                main.html:170
startGame                                           main.html:318
Loading scene: playerChoice                         main.html:170
Loading scene: playerChoice                         main.html:170
Loading scene: playerChoice                         main.html:170
Loading scene: game                                 main.html:170
DETECTIVE is dead                                  game.js:187
Winner: DUKE                                        game.js:192
erster: 2                                           game.js:215
zweiter: 1                                          game.js:216
dritter: undefined                                 game.js:217
Loading scene: hallOfFame                           main.html:170
>
```

Versioning!



103 commits

0,94 commits/day

5. UMSETZUNG

5.3 Implementierung der Details

- Erstellung der Welt
- **Bewegung**
- **Kollisionserkennung**
- Animation
- Goody Erstellung - Auswirkungen
- ~ 1300 Zeilen Code

5. UMSETZUNG

5.3 Implementierung der Details



5. UMSETZUNG

5.3 Implementierung der Details

- Erstellung der Welt
- Bewegung
- Kollisionserkennung
- **Animation**
- Goody Erstellung - Auswirkungen
- ~ 1300 Zeilen Code

AMB A Media Engineering Project



5. UMSETZUNG

5.3 Implementierung der Details

- Erstellung der Welt
- Bewegung
- Kollisionserkennung
- Animation
- **Goody Erstellung - Auswirkungen**
- ~ 1300 Zeilen Code

5. UMSETZUNG

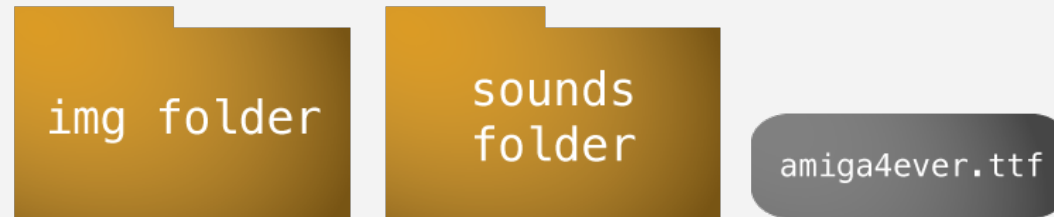
5.3 Implementierung der Details



5. UMSETZUNG

5.3 Implementierung der Details

Folders:



Libraries:



5. UMSETZUNG

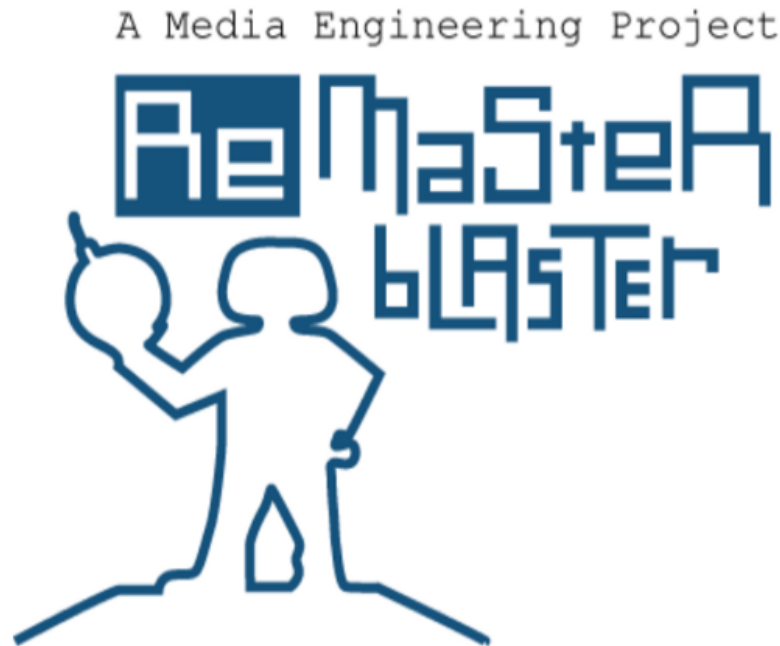


5. UMSETZUNG

5.4 Definition der Szenen

- Was sind Szenen?
- Original MasterBlaster Look übernommen und angepasst
- Konfigurationsmöglichkeit mit JSON-Objekt eingebunden
- eigenes Logo und Credits
- verbesserung der Szenen (Toggle, farbliche Trennung)
- da nur 2 Spieler als Ziel - Spriteauswahlmenu
- Loop: Shop, Countdown, GameUI, Hall Of Fame

5. UMSETZUNG



REMASTERBLASTER

BY

SEBASTIAN ADAM

SERGEJ BJAKOW

MICHAEL KAO

PAVLINA PAVLOVA

MAXIMILIAN SEYFERT

THIS GAME IS A REMAKE OF THE GAME
"MASTERBLASTER" BY ALEXANDER IVANOF

YOU CAN FIND US AT
WWW.REMASTERBLASTER.COM

SPECIAL THANKS TO THE CREATOR OF THE
ORIGINAL GAME ALEXANDER IVANOF

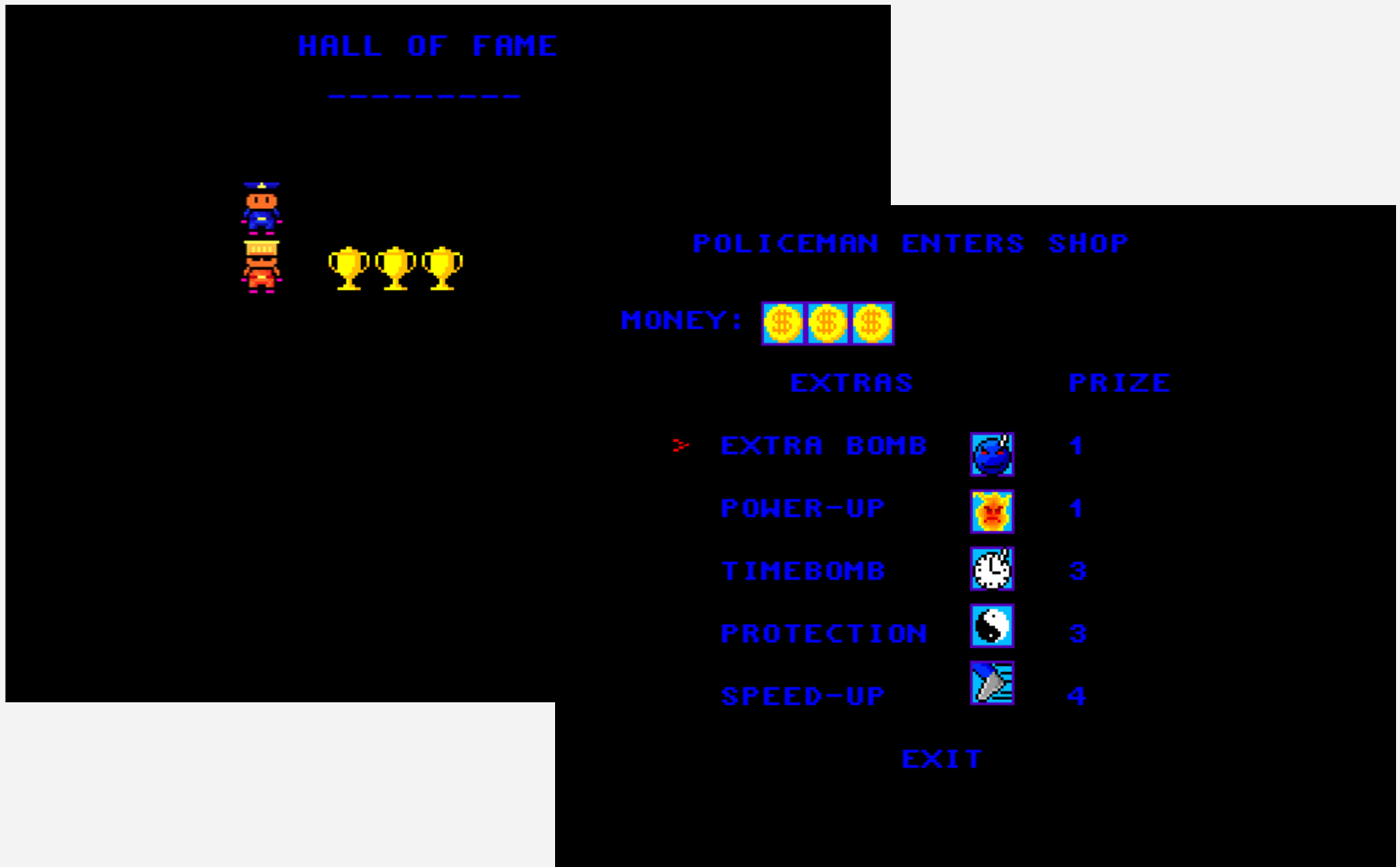
5. UMSETZUNG



A Media Engineering Project ■






5. UMSETZUNG



6. ZUSAMMENFASSUNG

6.1 Soll - Ist Analyse

Muss - Kriterien	<ul style="list-style-type: none">• Spielbares Produkt• 2 Steuerbare Spieler mit Tastatur• Bomben legen und Wände sprengen• Gegenspieler bei Treffer entfernen	
Soll - Kriterien	<ul style="list-style-type: none">• Einbindung der Extras (Goodies)• Konfigurationsmenü• Szenenabfolge• Hall of Fame	
Kann - Kriterien	<ul style="list-style-type: none">• Alle Goodies implementieren	
Kann - Kriterien:	<ul style="list-style-type: none">• mehrere Spieler• zusätzliche Eingabegeräte• Shop für Extras• Sprite-Auswahl für die Spieler	

6. ZUSAMMENFASSUNG

6.2 Let's play ... ReMasterBlaster!

