

**Masterarbeit**  
**Entwicklung einer auf Web-Standards basierenden**  
**Multiplayer-Online-Plattform für emulierte**  
**(S)NES-Konsolenspiele**

Kai Kühne  
798797

Beuth Hochschule für Technik Berlin  
Fachbereich VI – Informatik und Medien

Berlin, 2. Juli 2017

Betreuung: Herr Dipl.-Inform. Hans-Georg Reimer (LB)  
Begutachtung: Herr Prof. Dr. Hildebrand



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	.	.	.	1
1.1	Motivation	.	.	.	3
1.2	Zielsetzung	.	.	.	4
1.3	Vorgehensweise	.	.	.	5
<b>2</b>	<b>Grundlagen</b>	.	.	.	7
2.1	Emulation von Konsolenspielen	.	.	.	8
2.2	Der Multiplayer-Modus	.	.	.	8
2.3	Grundbegriffe der Netzwerk-Kommunikation	.	.	.	9
2.4	Das Realtime-Web	.	.	.	9
<b>3</b>	<b>Spezifikation</b>	.	.	.	11
3.1	Funktionsumfang	.	.	.	13
3.2	Bedienung	.	.	.	13
3.3	Anforderungen an die Umgebung	.	.	.	14
3.4	Leistung- & Qualitätsanforderungen	.	.	.	14
3.5	Betriebs- & Nutzungsbedingungen	.	.	.	14
3.6	Randbedingungen	.	.	.	14

<b>4 Herausforderungen</b> . . . . .	15
<b>5 Lösungsansätze</b> . . . . .	17
5.1 Vorhandene Teillösungen . . . . .	18
5.2 Lösungsansatz 1: Game-Server . . . . .	19
5.3 Lösungsansatz 2: Emulation im Browser . . . . .	20
<b>6 Methodik</b> . . . . .	21
<b>7 Konzeption</b> . . . . .	23
7.1 Entwurf der Benutzeroberfläche . . . . .	24
7.2 Software-Architektur . . . . .	24
<b>8 Implementierung</b> . . . . .	25
8.1 Entwicklungsumgebung . . . . .	26
8.2 Verwendete Software-Komponenten . . . . .	26
8.3 Ausgewählte Implementierungsdetails . . . . .	26
8.4 Vollständigkeit . . . . .	27
<b>9 Evaluation</b> . . . . .	29
9.1 Aufbau der Messumgebung . . . . .	30
9.2 Ergebnisse und Beobachtungen . . . . .	30
9.3 Diskussion und Bewertung . . . . .	30
<b>10 Fazit</b> . . . . .	31
10.1 Zusammenfassung . . . . .	32
10.2 Bewertung . . . . .	32
10.3 Ausblick . . . . .	32

<b>I Verzeichnisse</b> . . . . .	35
<b>Abbildungsverzeichnis</b> . . . . .	37
<b>Literatur</b> . . . . .	39
<b>Glossar</b> . . . . .	43
<b>Akronyme</b> . . . . .	45
<b>II Anhang</b> . . . . .	47
<b>Entwürfe</b> . . . . .	49



# 1

## Einleitung

*“Begin at the beginning,” the King said, gravely, “and go on till you come to an end; then stop.”*

— Lewis Carroll, *Alice in Wonderland*

Nach der Krise der Videospiel-Industrie Mitte der achtziger Jahre hatte Nintendo mit der Veröffentlichung des **Nintendo Entertainment System (NES)** eine Konsole für Videospiele auf den Markt gebracht, die sich millionenfach verkaufte. Dabei führte das Unternehmen weltbekannte Marken wie Mario, und Zelda ein, mit denen das Unternehmen bis zum heutigen Tag erfolgreich ist. In den 90er Jahren wurde der Nachfolger auf den Markt gebracht, das **Super Nintendo Entertainment System (SNES)**. Beide Systeme verkauften sich zusammen weltweit über 100 Millionen mal und sind damit die erfolgreichsten Systeme ihrer Generation. Insgesamt wurden für beide Systeme ca. 1495 Spiele entwickelt und lizenziert; darunter Singleplayer-Klassiker wie Super Me-

troid oder »Mega Man«. Gespielt werden konnte aber nicht nur allein: Die Konsolen wurden von vornherein für mehrere Spieler ausgelegt und verfügen über zwei Anschlüsse für Controller. Dies führte dazu, dass viele Spiele sowohl einen Singleplayer- als auch einen Multiplayer-Modus beinhalteten, der es ermöglichte, mit mehreren Spielern an der Konsole zu spielen.

Mittlerweile existieren mindestens zwei weitere erfolgreiche Videospiel-Konsolen: Die Xbox One von Microsoft und die PlayStation 4 von Sony. Wie auch schon das **NES** unterstützen beide Systeme mehrere Spieler, über entsprechende Controller, die mit der Konsole gekoppelt werden. Die technologische Entwicklung seit den 80er und 90er Jahren ist gut am Internet abzulesen: Damals galt ein 28.8k-Modem als Highspeed-Gerät. Heutzutage sind breitbandige Internetanschlüsse großflächig verfügbar. Dieser technologische Fortschritt ermöglicht es, größere Datenmengen übers Internet zu übertragen – z. B. Daten eines Multiplayer-Spiels. Man spielt nicht mehr exklusiv im eigenen Wohnzimmer oder auf einer LAN-Party gemeinsam miteinander, sondern kann dies über das Internet und über Landesgrenzen hinweg tun. Dazu haben Microsoft und Sony jeweils kostenpflichtige Online-Plattformen geschaffen, über die Spieler miteinander über verschiedenste Kanäle interagieren – und, viel wichtiger – miteinander spielen können. Die Plattformen bieten einen Treffpunkt der Kommunikation und Vermittlung und stellen für die Spiele-Entwickler Dienste wie Spiel-Lobbys und Matchmaking bereit.

Eine bemerkenswerte Eigenschaft beider Plattformen ist deren einfache Handhabung: Sie machen es den Spielern sehr einfach, gemeinsam über das Internet miteinander zu spielen. Die Plattformnutzer müssen über keine speziellen Kenntnisse verfügen, müssen keine zusätzliche Software installieren oder etwa darin geübt sein, einen dedizierten Game-Server aufzusetzen. Mitglieder der Plattform können sich auf das Spielen konzentrieren. Aktuelle Spiele-Titel können ohne große Hürden über die beschriebenen Plattformen gespielt werden. Eine Plattform für **SNES**-Spiele, die einen vergleichbaren Komfort bietet, existiert bislang nicht.

## 1.1 Motivation

Möchte man am PC ein Konsolenspiel, z. B. Super Mario, spielen, benötigt man neben dem eigentlichen Spiel auch ein Programm, mit dem das Spiel ausgeführt werden kann: einen Emulator. Einen geeigneten Emulator auszuwählen ist nicht einfach: Allein für das **SNES** existieren mehr als ein Dutzend Emulatoren für verschiedenste Betriebssysteme (vgl. Wikipedia, [List of video game emulators](#)), die sich in der Genauigkeit der Emulation, der Kompatibilität und weiteren Aspekten unterschieden und von denen partiell zusätzliche Ableger (Forks) existieren (FIXME). Ein Teil der Emulatoren verfügt über einen Multiplayer-Modus. Mit dem Problem, dass die verschiedenen Emulatoren untereinander meist inkompatibel sind (FIXME). Auch bei komplexeren Emulationssystemen wie RetroArch, das viele Emulatoren enthält und um eine eigens geschaffene Netplay-Funktion erweitert, ist die Kompatibilität nicht garantiert (vgl. Richards, [Netplay core testing](#)). Um erfolgreich ein **SNES**-Spiel im Multiplayer-Modus gespielt werden, müssen alle Spieler im Idealfall den gleichen Emulator in der selben Version verwenden. Sofern die verwendeten Emulatoren untereinander kompatibel sind, existieren weitere Hürden, die für einen Anwender ohne technischen Hintergrund nicht ohne Weiteres zu lösen sind. Ein möglicher Ablauf zum Erstellen eines Spiels:

- Einer der Spieler wird als **Host** bestimmt und startet den Emulator im Netzwerk-Modus.
- Der Host muss dabei sicherstellen, dass der spezifische Port, auf dem der Prozess läuft, von allen Mitspielern über das Netzwerk erreichbar ist. Falls der Host-Spieler sich in einem Netzwerk befindet, dessen Routing auf **Network Address Translation (NAT)** basiert, muss eine Port-Weiterleitung für den Emulator-Prozess eingerichtet werden, der auf dem PC des Host-Spielers läuft. Dazu muss die verwendete Portnummer des Emulators bekannt und eine Port-Weiterleitung im Administrationsbereich des Routers konfiguriert sein. Für das beschriebene Szenario ist die korrekte **NAT**-Konfiguration Voraussetzung für das Spielen im Multiplayer-Modus.
- Wenn der ausgewählte Emulator über keine Funktion verfügt, um Netzwerkspiele aufzulisten, benötigen alle Mitspieler die IP-Adresse des Host-Spielers, um eine Verbindung herzustellen.
- Der Host-Spieler muss die eigene öffentliche IP-Adresse ermitteln und seinen Mitspielern mitteilen.

- Die Mitspieler verbinden sich mit dem Host-Spieler über die Eingabe der IP-Adresse des Hosts im entsprechenden Dialog der Benutzeroberfläche des verwendeten Emulators.

## 1.2 Zielsetzung

Im Zuge der immer größer werdenden Verfügbarkeit von breitbandigen Internetanschlüssen entstehen neue Chancen für die Entwicklung interessanter neuer Lösungen. Mit den Chancen stellen sich aber auch neue Herausforderungen, für deren Lösung neue Technologien geschaffen werden. Ein Teilbereich ist das [World Wide Web \(Www\)](#), das eine kontinuierliche Weiterentwicklung erfährt und sich die Computer-Nutzung immer mehr in den Internet-Browser verlagert.

Ablesen lässt sich die Weiterentwicklung der Web-Technologien zum Beispiel durch die Entwicklung neuer Standards- und Protokolle. Der Internet-Browser hat nahezu den Funktionsumfang eines Desktop-Systems und kann ein breites Spektrum von Anwendungsfällen abdecken. Das Abspielen von Audio- und Videodateien, die direkte Kommunikation zwischen Browsern und das Ausführen von komplexen 3D-Anwendungen ist mittlerweile möglich.

Diese Arbeit untersucht die Frage, ob die aktuell verfügbaren Web-Standards die Anforderungen für die Entwicklung einer Multiplayer-Plattform für [SNES](#)-Spiele erfüllen und welche Technologien wie kombiniert werden müssen, um ein funktionales System zu schaffen. Die zentralen Aspekte sind die Auswahl, die Kombination und die Integration der verschiedenen Technologien zu einem Gesamtsystem bestehend aus verschiedenen Komponenten, die alle im Zuge dieser Arbeit entwickelt werden.

Der Fokus liegt dabei auf der Entwicklung einer funktionalen Web-Plattform mit den wichtigsten Basisfunktionen, die aufgrund einer sauberen Architektur leicht um neue Funktionen zu erweitern ist.

## 1.3 Vorgehensweise

1. **Grundlagen** Einarbeitung und Erläuterung der technischen Grundlagen, die für die Lösung der Zielsetzung notwendig sein können. Dies umfasst verschiedene Web-Technologien auf der einen Seite und Emulationssysteme auf der anderen Seite.
2. **Zieldefinition** Genaue Festlegung des Funktionsumfangs des zu erstellenden Systems.
3. **Herausforderungen** Ableitung der zu lösenden Teilprobleme, die sich aus den definierten Zielen ergeben.
4. **Lösungsansätze** Untersuchen, ob und welche Teillösungen für die im vorherigen Abschnitt beschriebenen Probleme existieren. Vorstellen von Lösungsansätzen, die ggf. auf vorhandenen Teillösungen aufbauen oder diese erweitern. Gewählten Lösungsansatz vorstellen und mögliche Alternativen aufzeigen.
5. **Methodik** Beschreibung der Umsetzung.
6. **Konzeption** Entwurf der Benutzeroberfläche, Definition der System-Komponenten und der Software-Architektur.
7. **Implementierung** Technische Beschreibung der entwickelten Lösung mit Fokus auf ausgewählte Aspekte der Implementierung.
8. **Evaluation** Überprüfung und Bewertung der Lösung im Hinblick auf das Erreichen der definierten Ziele.
9. **Fazit** Zusammenfassung und Bewertung der Arbeit sowie Vorstellen von Erweiterungsmöglichkeiten.



# 2

## Grundlagen

Inhalt dieses Kapitels ist eine Einführung in die technischen Grundlagen dieser Arbeit. Es werden die wichtigsten Aspekte und Technologien erläutert und für das Lösungsumfeld essentielle Begriffe erklärt. Die Erläuterungen schaffen ein Grundverständnis für die in den nachfolgenden Kapiteln behandelten Aspekte und sind für eine Beurteilung der entwickelten Lösung notwendig. Schwerpunkte sind für die Zielsetzung potentiell geeignete Web-Technologien, Emulatoren & ROMs und Grundlagenthemen zur Kommunikation in Computer-Netzwerken.

## 2.1 Emulation von Konsolenspielen

Wie eingangs erwähnt, sind zum Spielen von SNES-Spielen am Computer zwei Dinge notwendig: Ein Emulationsprogramm — auch [Emulator](#) genannt — und das jeweilige Konsolenspiel, das ausgeführt werden soll. Das Spiel liegt dabei in Form einer Datei vor. Sie beinhaltet die exakt gleichen Daten, die sich für gewöhnlich auf den Modulen befinden, die zum Spielen in die Konsolen gesteckt werden müssen. Zum Kopieren der Daten existieren spezielle Geräte, mit denen eine solche Datei erzeugt werden kann. Dabei wird ein genaues Abbild des Moduls erzeugt, inklusive der Sektoren und der jeweiligen Dateisystem-Struktur. Die Dateien werden oft als ROMs bezeichnet, da die Daten des Spiels innerhalb des Moduls in einem ROM-Chip gespeichert sind. Um ein Spiel auszuführen zu können, muss die Spiel-Datei in den Emulator geladen werden.

Der Emulator ist ein Software-Programm, das es ermöglicht Konsolentitel auf einem Computer auszuführen. Dies funktioniert, in dem die Hardware der jeweiligen Konsole nachgeahmt wird. Die verschiedenen Komponenten der Konsole (CPU, Soundchip, etc.) sind dabei so in Software implementiert dass sie sich möglichst so verhalten, wie die nachzuahmende Hardware. Erst eine möglichst exakte Emulation der ursprünglichen Laufzeitumgebung ermöglicht die Ausführung von unveränderten Spielen in Form von ROM-Dateien. Die Spiele, bzw. das Abbild der Software, ist eine exakte Kopie des Originals.

- Roms
- Emulatoren

## 2.2 Der Multiplayer-Modus

Das erste entwickelte Computerspiel war ein Multiplayer-Spiel und benötigte damit mindestens zwei Spieler, die gemeinsam an einem Gerät spielen. Die Verarbeitung der Benutzereingaben erfolgt zentral an dem einzigen Gerät, so wie auch die Darstellung der grafischen Oberfläche.

## 2.3 Grundbegriffe der Netzwerk-Kommunikation

### 2.3.1 Latenz

- Minimalste Latenz essentiell beim Spielen
- Alles über N ms ist nicht gut genug. Um angepeilten Maximalwert zu erreichen, System entsprechend konstruieren und Technologien wählen, die eine hinreichende Lösung erst möglich machen.
- Wo von hängt Latenz ab? Was kann man da überhaupt machen?
- RTT, TCP vs. UDP?

### 2.3.2 RTT

### 2.3.3 Realtime

## 2.4 Das Realtime-Web

Vergleich zur alten Herangehensweise. Frontend viel Logik, Backends eigentlich nur noch API.

## 2.4.1 Websockets

## 2.4.2 HTTP/2

## 2.4.3 SSE

## 2.4.4 WebRTC

Hier nur beschreiben. Hier kann ich letztlich auch nur raten, dass die gewählten Technologien sinnvoll sein werden. Ob die getroffene Auswahl gut war (wurden Ziele erreicht?), zeichnet sich ggf. schon während der Implementierung ab. Ergebnisse und Erkenntnisse, die die Bildung eines ersten Urteils ermöglichen, werden erst in der Evaluationsphase geschaffen.

NICHT:

- Vergleichen/Abwägen
- Entscheiden
- Begründen

# 3

## Spezifikation

Dieses Kapitel beinhaltet die Problembeschreibung und die Motivation, die Anstoß zur Bearbeitung dieses Themas war. Der Hauptteil besteht aus einer Zieldefinition der zu erstellenden Software. Dies beinhaltet auch Aspekte, die ausdrücklich nicht Teil dieser Arbeit sind.

Dieser Abschnitt definiert die zu erfüllenden System-Anforderungen, die im späteren Verlauf dieser Arbeit während der Evaluation überprüft werden. Zunächst wird definiert, welche Grundfunktionen das System haben soll. Weiter gelistet sind die Leistungs- und Qualitätsanforderungen sowie die Randbedingungen, innerhalb derer die definierten Anforderungen gelten. Abgeschlossen wird das Kapitel durch eine Auflistung von Kriterien, die nicht Gegenstand dieser Arbeit sind.

### 3 Spezifikation

---



Abbildung 3.1: This flower was photographed at my home town in 2010.

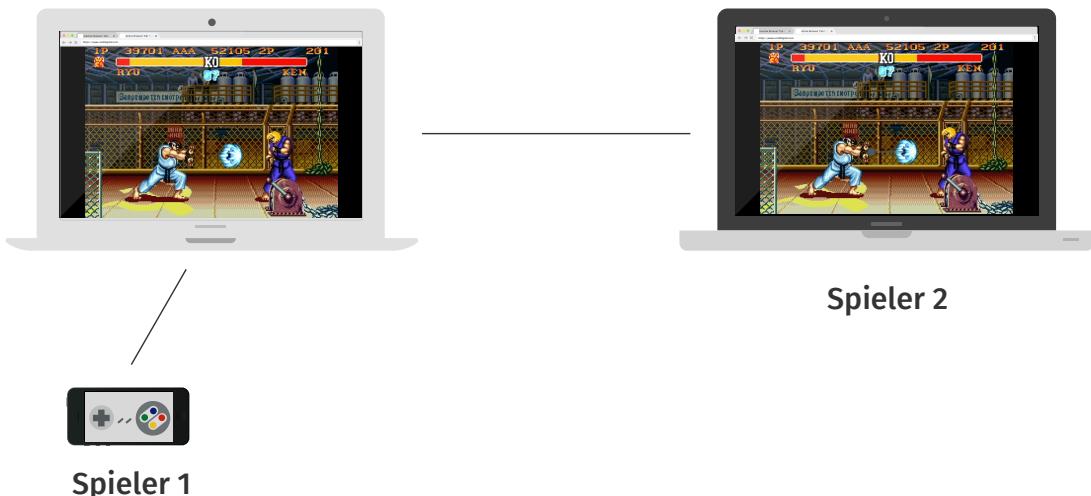


Abbildung 3.2: This flower was photographed at my home town in 2010.

Das Ziel dieser Masterarbeit ist die Entwicklung einer Online-Plattform, die es den Nutzern einfach macht, gemeinsam über das Netzwerk miteinander SNES-Spiele zu spielen. Dafür existieren Emulations-Programme in Hülle und Fülle. Warum muss man das neu machen?

- Völlig neu? Alt?
- Einfache Lösung? Total schwer?
- Forschung? Anwendung?

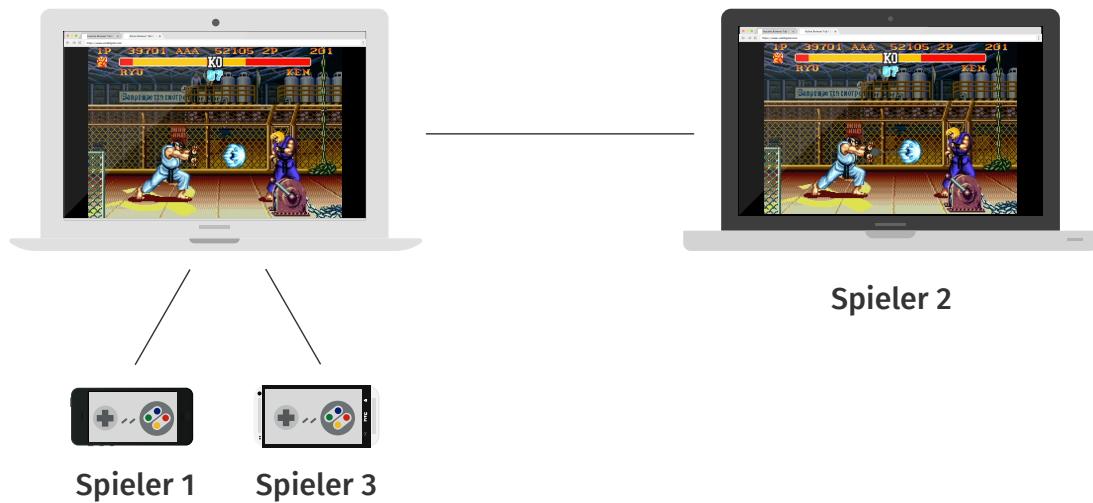


Abbildung 3.3: This flower was photographed at my home town in 2010.

## 3.1 Funktionsumfang

### 3.1.1 Muss-Kriterien

### 3.1.2 Kann-Kriterien

### 3.1.3 Abgrenzungskriterien

## 3.2 Bedienung



### **3.3 Anforderungen an die Umgebung**

### **3.4 Leistung- & Qualitätsanforderungen**

### **3.5 Betriebs- & Nutzungsbedingungen**

### **3.6 Randbedingungen**

# 4

## Herausforderungen

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.



# 5

## Lösungsansätze

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

## 5.1 Vorhandene Teillösungen

### 5.1.1 melody-jsnes

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

### 5.1.2 webtendo

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

### 5.1.3 retroarch-web

- Zu kompliziert, snes9x nur ein Emulator unter vielen
  - Nicht benutzerfreundlich

- Reagiert nicht auf Touch-Events

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

## 5.2 Lösungsansatz 1: Game-Server

Definition list

Is something people use sometimes.

Markdown in HTML

Does *not* work **very** well. Use HTML tags.

**Pro** Client-Seite sehr einfach und portabel (jsmpeg spielt den mpeg-ts stream [mpeg1/mp2] ab und fertig).

**Pro** Super Browser-Unterstützung. Habe keinen Browser gefunden, der den Stream nicht abgespielt hat.

**Contra** Minimaler Prototyp (mit statischem Video) hat nur funktioniert, in dem Audio und Video getrennt kodiert und zum jsmpeg-WebSocket-Relay geschickt wurde.

**Contra** Der Quellcode des Emulator(frontend)s müsste angepasst und erweitert werden:

**Contra** retroarch-ffmpeg unterstützt Live-Streaming und es sind auch verschiedenste Optionen konfigurierbar. Problem: mpeg1/mp2 wird nicht unterstützt, zumindest habe ich keine funktionierende Konfiguration gefunden. Für die Unterstüt-

zung von mpeg1/mp2 müsste also der retroarch-ffmpeg-core angepasst werden (viel C, wenig Kommentare). Ist prinzipiell machbar, aber nicht mit meiner C-Expertise, und nicht innerhalb des gegebenen Zeitrahmens.

**Contra** Auch schwierig: Die Inputs müssten von den Clients an retroarch geschickt werden. Hier müsste wieder das Emulator-Frontend (retroarch) angepasst werden, C/C++. Man hätte wahrscheinlich einen WebSocket-Server bauen können, der dann Nachrichten mit Payload=Controller-Input enthält und diese Nachrichten dann auf retroarch-Controller-Eingaben mappt und an das Emulator-Frontend weitergibt.

**Contra** ffmpeg, libav, Encodings, (spezielles) Muxing, A/V-Sync, Doku jeweils katastrophal

Noch zu testen: Paralleles Encoden von Audio/Video mit retroarch-ffmpeg. Das war für jsmpeg sowieso nötig.

### 5.3 Lösungsansatz 2: Emulation im Browser

- Warum sind die Smartphones mit dem “eigenen Bildschirm” verbunden, und nicht mit dem Host-PC?
  - Minimierung des lokalen Input-Lags zwischen Anzeige und Steuerung

Tamarys WebRTC-Plugin

# 6

## Methodik

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.



# 7

## Konzeption

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

- Technologien nicht neu beschreiben, verweisen auf Kapitel 3
- Zeigen, wie diese zusammen spielen, um das Ziel zu erreichen

## 7.1 Entwurf der Benutzeroberfläche

## 7.2 Software-Architektur

# 8

## Implementierung

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

## 8.1 Entwicklungsumgebung

- Gitlab, Tickets, CI, etc.
- Heroku

## 8.2 Verwendete Software-Komponenten

- Ionic/Quasar: Macht Sinn, wenn *eine* App im Browser und mobil laufen muss.  
Hier gibt es zwei alleinstehende Apps, die völlig verschiedene Dinge tun

## 8.3 Ausgewählte Implementierungsdetails

### 8.3.1 Cool #1

### 8.3.2 Cool #2

### 8.3.3 Integration von xnes/snes9x.js und vue.js

Drei notwendige Schritte:

1. Deaktivieren von eslint für Kompilierungsergebnis snes9x.js,
2. Webpack-Konfiguration: `node: {fs: 'empty'}`,
3. Anfügen von `export {Module as default}` ans Ende von snes9x.js, damit die Datei korrekt von Webpack erkannt wird.

## **8.4 Vollständigkeit**



# 9

## Evaluation

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

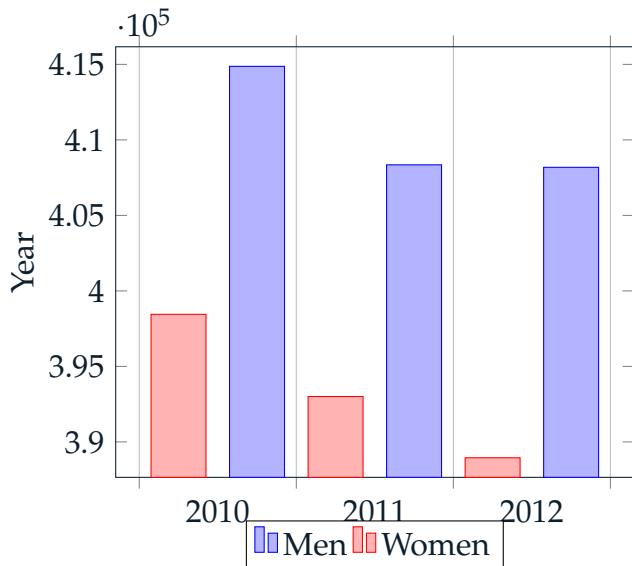
- Spielbar?
- Nur im LAN, oder auch übers Internet?

- Welche Latenz war angestrebt? Erreicht?
- WebRTC vs. WebSockets
  - Wenn LAN-Spiel: Welchen Vorteil bringt WebRTC? (Wie viel weniger Latenz?)
- Emulator: emscripten-Backends vergleichen. asm.js vs. WebAssembly. Emulator funktioniert mit beidem in den zu unterstützenden Browsern (Chrome, Firefox). Bringt es Vorteile? Bessere Framerate? Wie viel besser? Andere Metriken?

## 9.1 Aufbau der Messumgebung

## 9.2 Ergebnisse und Beobachtungen

[https://www.sharelatex.com/learn/Pgfplots\\_package](https://www.sharelatex.com/learn/Pgfplots_package)



## 9.3 Diskussion und Bewertung

# 10

## Fazit

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

## 10.1 Zusammenfassung

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

## 10.2 Bewertung

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

## 10.3 Ausblick

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich

die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.



# Verzeichnisse



# Abbildungsverzeichnis

3.1 Home town flower . . . . .	12
3.2 Home town flower . . . . .	12
3.3 Home town Test . . . . .	13



# Literatur

## Print

Grigorik, Ilya. *High Performance Browser Networking: What every web developer should know about networking and web performance*. 1. Aufl. O'Reilly Media, Sep. 2013. ISBN: 9781449344764. URL: <https://hpbn.co>.

Hebert, Fred. *Learn You Some Erlang for Great Good!: A Beginner's Guide*. 1. Aufl. No Starch Press, Jan. 2013. ISBN: 9781593274351. URL: <http://learnyousomeerlang.com>.

Jurić, Saša. *Elixir in Action*. 1. Aufl. Manning Publications, Juni 2015. ISBN: 9781617292019. URL: <https://manning.com/books/elixir-in-action>.

Krug, Steve. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability (3rd Edition) (Voices That Matter)*. 3. Aufl. New Riders, Jan. 2014. ISBN: 9780321965516. URL: <https://sensible.com/dmmt.html>.

McCord, Chris, Bruce Tate und José Valim. *Programming Phoenix: Productive | > Reliable | > Fast*. 1. Aufl. Pragmatic Bookshelf, Apr. 2016. ISBN: 9781680501452. URL: <https://pragprog.com/book/phoenix/programming-phoenix>.

Tan Wei Hao, Benjamin. *The Little Elixir & OTP Guidebook*. 1. Aufl. Manning Publications, Okt. 2016. ISBN: 9781633430112. URL: <https://manning.com/books/the-little-elixir-and-otp-guidebook>.

## Web

- emulation-general.wikia.com. *Super Nintendo emulators*. URL: [http://emulation-general.wikia.com/wiki/Super\\_Nintendo\\_emulators](http://emulation-general.wikia.com/wiki/Super_Nintendo_emulators) (besucht am 10.06.2017).
- Fablet, Youenn. *Announcing WebRTC and Media Capture*. URL: <https://webkit.org/blog/7726/announcing-webrtc-and-media-capture> (besucht am 08.06.2017).
- Inc., Apple. *What's New in Safari: Safari 11.0*. URL: [https://developer.apple.com/library/content/releasenotes/General/WhatsNewInSafari/Safari\\_11\\_0/Safari\\_11\\_0.html](https://developer.apple.com/library/content/releasenotes/General/WhatsNewInSafari/Safari_11_0/Safari_11_0.html) (besucht am 08.06.2017).
- Jurić, Saša. *Observing low latency in Phoenix with wrk*. Juni 2016. URL: [http://theerlangelist.com/article/phoenix\\_latency](http://theerlangelist.com/article/phoenix_latency) (besucht am 07.06.2017).
- Mann, Ben. *Multiplayer JS game platform*. URL: <https://github.com/8enmann/webtendo> (besucht am 07.06.2017).
- Richards, Gregor. *Netplay core testing*. URL: <https://github.com/GregorR/RetroArch/wiki/Netplay-core-testing> (besucht am 10.06.2017) (siehe S. 3).
- Rouyrre, Maxime. *WebAssembly 101: a developer's first steps*. URL: <https://blog.openbloc.fr/webassembly-first-steps> (besucht am 01.07.2017).
- Wikipedia. *List of video game emulators*. URL: [https://en.wikipedia.org/wiki/List\\_of\\_video\\_game\\_emulators#Super\\_NES](https://en.wikipedia.org/wiki/List_of_video_game_emulators#Super_NES) (besucht am 10.06.2017) (siehe S. 3).

# Index

## D

Donkey Kong, 1



# Glossar

**Django** Web-Framework für die Programmiersprache Python

**Emulator** Computer-Programm zur Emulation von Konsolenspielen

**Host** Dienstanbietender Teilnehmer in einem Computer-Netwerk



# Akronyme

**GUI** Graphical User Interface

**HTTP** Hyper Text Transfer Protocol

**MPEG-1** Moving Picture Experts Group Phase 1

**NAT** Network Address Translation

**NES** Nintendo Entertainment System

**ROM** Read-Only Memory

**SNES** Super Nintendo Entertainment System

**WWW** World Wide Web



# Anhang



# Entwürfe



DROP GAME ROM HERE

PICK A NAME

SELECT NO. OF PLAYERS

2    3    4

HOST GAME



GAME  
**Super Tennis**

HOST  
Alice

PLAYERS  
1/2

Join this Game



GAME  
**Mario World**

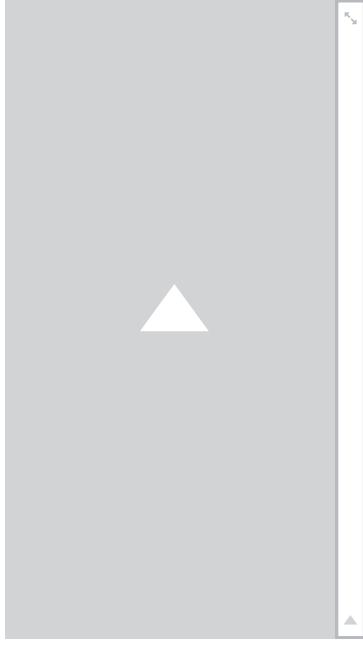
HOST  
Bob

PLAYERS  
2/2

Join this Game



## Super Bomberman



**Alice42:** Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

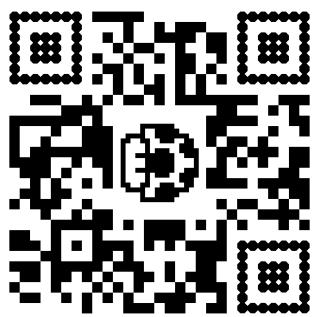
Write a Message

Send



## Super Bomberma

Use your smartphone as a controller!  
Just scan the code using the **SMES app**.



**Alice42:** Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Write a Message

Send

Want to join...	X	V
Local Player	X	I



