

Titelblatt

Projektinformationen

Titel: Palim-Palim

Projektnummer: 21FS_I4DS08

Projekt-Team

Daniel Obrist, 8iCbb

daniel.obrist@students.fhnw.ch

Severin Peyer, 8iCbb

severin.peyer@students.fhnw.ch

Auftraggeber und Betreuung FHNW

Marco Soldati

Fachhochschule Nordwestschweiz FHNW

Hochschule für Technik

Bahnhofstrasse 6

CH-5210 Windisch

+41 56 202 77 31

marco.soldati@fhnw.ch

Tabea Iseli

Fachhochschule Nordwestschweiz FHNW

Hochschule für Technik

Bahnhofstrasse 6

CH-5210 Windisch

+41 56 202 86 53

tabea.iseli@fhnw.ch

Zeitbudget

Das Projekt wird im Rahmen des Frühlingssemesters 2021 durchgeführt.

Nominell sind für das Projekt 360 Arbeitsstunden pro Teammitglied veranschlagt.

Wichtigste Daten

Beginn: 22. Februar 2021

Ende: 20. August 2021

Abstract

- Ergebnisse aus den Spieletests, der Entwicklung und der Literaturrecherche zusammengefasst erläutern.
- Erst am Schluss schreiben!aa
- Wichtig für den ersten Eindruck

Contents

1	Einleitung	3
2	Spieletests und Resultate	6
2.1	Resultate	6
2.2	Planung und Durchführung der Spieletests	6
3	Gamedesign	7
3.1	Spielkonzept	7
3.1.1	Gameplay-Loops	7
3.2	Spielvarianten	7
3.3	Design und Usability	7
4	Implementation	8
4.1	Architektur	8
4.2	Technologien	8
4.2.1	WebRTC	8
4.3	Sicherheit	12
4.4	Testing	12
4.5	Deployment und Betrieb (Anhang?)	12
5	Fazit	13
6	Literaturverzeichnis	14
A	Ehrlichkeitserklärung	15
B	Testprotokolle, weitere Spielkonzepte/Ideenfindung...	16

Chapter 1

Einleitung

Teil 1 / Was wurde erreicht?

- *Beschreibung des Videospiels Palim-Palim (inkl. Screenshot)*
- *Aufstellung der Forschungsfragen und den Erkenntnissen*

Teil 2 / Warum wurde es gemacht?

- *Ausgangslage inkl. Forschungsstand*
- *Relevanz der Problemstellung*
- *Was ist das Umfeld?*

Teil 3 / Wie wurde es gemacht?

- *Grobe Beschreibung der angewendeten Methodik*
 - *Spielementwicklung (Architektur, Technologien)*
 - *Spieletests (Methoden)*

Teil 4

- *Aufbau des Dokuments und Überleitung in den theoretischen Teil*

Palim-Palim ist ein interaktives Multiplayer-Videospiel für Kinder und betagte Menschen. Zwei Spielende können damit Kinder-Kaufladen in einer virtuellen Umgebung spielen. Es verfügt über einen Video-Chat als inhärente Game-Mechanik, um die Kommunikation zwischen den Spielerinnen und Spielern zu fördern.

Mit dem Spiel wird untersucht, welchen Einfluss ein Video-Chat in Videospielen auf die User Experience und die Kommunikation zwischen den Spielenden hat. Im Speziellen werden die folgenden Fragestellungen untersucht:

- Welche Wirkung hat ein integrierter Video-Chat auf das Spielerlebnis von betagten Menschen?
- Welche Wirkung hat ein integrierter Video-Chat auf das Spielerlebnis von Kindern?
- Welche Auswirkungen hat die Art und Weise, wie der Video-Chat in das Spiel integriert ist, auf das Spielerlebnis?
- Welche Wirkung hat ein integrierter Video-Chat auf die Kommunikation zwischen den Spielenden?
- Wird ein integrierter Video-Chat aktiv als Kommunikationsmittel zur Bewältigung von Spielaufgaben genutzt?
- Welche Auswirkungen hat die Art und Weise, wie der Video-Chat in das Spiel integriert ist, auf die Förderung der Kommunikation?

Erkenntnisse Video-Chats werden heutzutage schon von vielen Familien benutzt, um mit Ihren Verwandten zu kommunizieren. Auch in der Kommunikation mit den Grosseltern wird dabei immer mehr auf Video-Telefonie gesetzt. Dies gilt besonders für Zeiten, in denen Besuche im Alters- oder Pflegeheim auf Grund kursierenden Viren wie Corona schwierig oder unmöglich werden. Betagte Menschen schätzen und nutzen diese moderne Art der Kommunikation mit der Familie auch immer mehr – besonders weil es öfters auf ältere Benutzergruppen angepasste Angebote gibt [1].

Allerdings hat die Video-Telefonie immer noch Grenzen, welche die Interaktionen unnatürlich und teilweise entfremdend wirken lassen. Vor allem für Kinder ist es schwierig, Gesprächsthemen und Kommunikationswege zu finden, die sich so lustig und verbindend anfühlen, wie die Zeit mit der Grossmutter oder dem Grossvater im echten Leben. Oft sind Kinder vom Gespräch

schnell gelangweilt – sie würden lieber etwas spielen [2]. Spielen kann ein Mittel sein, um Kinder besser einzubeziehen und die Interaktion mit ihnen zu unterstützen, wie bisherige Studien zu dem Thema Video-Calls mit Eltern und Kindern zeigen [3].

Ergänzend zum Video-Telefonie-Aspekt gibt es bereits viel Literatur bezüglich generationenübergreifender Computerspiele [4], [5], [6]. Erkenntnisse aus einer Studie von Derboven et al [7] deuten ausserdem drauf hin, dass in einem Multiplayer-Videospiel die zusätzliche Kommunikationsfunktionalität durch einen Video-Chat oft sowohl von älteren als auch von jüngeren Personen begrüsst wird. Allerdings bietet die Forschung bisher keine detaillierte Studie mit Kindern und betagten Personen in diesem Zusammenhang.

Am Institut für Data Science (I4DS) der Fachhochschule Nordwestschweiz wird im Rahmen des Projekts Myosotis schon seit einigen Jahren daran gearbeitet, Video-Spiele zu entwickeln, welche in unterhaltsamer Weise die soziale Interaktion zwischen betagten Menschen und ihren Angehörigen unterstützen [8]. Dabei wurden schon etliche Spiele umgesetzt und getestet [6]. Mit einer Integration eines Video-Chats in ein Spiel hat sich jedoch bisher noch kein Team explizit auseinandergesetzt. Palim-Palim schliesst diese Lücke und zeigt wertvolle Erkenntnisse über die Kombination von Video-Chats und Video-Spielen mit Kindern und betagten Personen.

Um die formulierten Fragen zu beantworten, wurden mehrere Varianten eines Videospiels implementiert, in welchen ein Video-Stream zu einem Teil des Spiels wird. Anschliessend wurden mit allen Varianten Spieltests durchgeführt, um herauszufinden, ob die Integration eines Video-Chats in einem Video-Spiel einen positiven oder negativen Einfluss auf die User Experience sowie die Kommunikation zwischen den Spielenden hat. Die Arbeit fokussiert sich speziell auch auf die Art und Weise, wie ein Video-Stream in ein Spiel integriert werden kann.

Grobe Systemarchitektur, verwendete Methoden und Konzepte

Chapter 2

Umfeldanalyse und Zielgruppe

- *Beschreibung des Umfelds / Anwendungsdomäne (betagte Personen und Kinder)*

Chapter 3

Intergenerationelles Spielen

- *Forschungsstand (Welche Methoden/Ansätze werden angewendet?)*
- *Bisherige Erkenntnisse zu intergenerationellem Spielen*

Chapter 4

Videochats in Videospielen

- *Forschungsstand (Welche Methoden/Ansätze werden angewendet?)*
- *Bisherige Erkenntnisse zu Videochats in Videospielen*

Chapter 5

Aufstellung der Forschungsfragen

- *Lücken der bisherigen Forschung*
- *Aufstellung der Forschungsfragen*

Chapter 6

Methoden

In Palim-Palim verwendete Methoden

- *Methode der Spieletest*
- *Methode der Spieletest-Auswertung*

Chapter 7

Spieletests und Resultate

7.1 Resultate

- Ergebnisse
- Beantwortung der aufgestellten Forschungsfragen

7.2 Planung und Durchführung der Spieletests

- Organisation der Spieletests (Aufbau, Ablauf, Testpersonen, Testszenarien, Testumgebung)
- Beobachtungen

Chapter 8

Gamedesign

8.1 Spielkonzept

8.1.1 Gameplay-Loops

8.2 Spielvarianten

8.3 Design und Usability

Chapter 9

Implementation

9.1 Architektur

9.2 Technologien

9.2.1 WebRTC

WebRTC [<https://webrtc.org/>] ist ein Open-Source-Projekt, das die Echtzeitkommunikation von Audio, Video und Daten in Web- und nativen Anwendungen ermöglicht. Die Technologie ist in allen modernen Browsern sowie auf nativen Clients für alle wichtigen Plattformen verfügbar. Dabei wird zwischen zwei Browsern eine Peer-To-Peer-Verbindung aufgebaut, worüber die Daten gestreamt werden.

Signaling

WebRTC verwendet die `RTCPeerConnection` [<https://developer.mozilla.org/en-US/docs/Web/API/RTCPeerConnection>] JavaScript API, um Streaming-Daten zwischen Browsern zu kommunizieren. Zusätzlich wird aber auch einen Mechanismus benötigt, um die Kommunikation zu koordinieren und Kontrollnachrichten zu senden. Dieser Prozess wird als *Signaling* bezeichnet. Signaling-Methoden und -protokolle sind von WebRTC nicht spezifiziert und können je nach Anwendungsfall entsprechend gewählt werden.

In diesem Projekt wurde Socket.IO für die Signalisierung verwendet, da es sich mit seinem integrierten Room-Konzept für eine Video-Chat-App anbietet. Es gibt aber viele Alternativen, welche sich teilweise besser für produktive Applikationen eignen (<https://bloggeek.me/siganling-protocol-webrtc/>). Palim-Palim verwendet eine Node.js Server-Applikation (implementiert in `server.js`),

welche als Signaling-Server fungiert. Der Server hat dabei folgende zwei Aufgaben.

Erstens fungiert er als Message Relay. Dies wird benötigt, um verschiedene Informationen von und zu den Clients zu senden, damit diese untereinander eine WebRTC-Peer-Verbindung aufbauen können:

```
1 socket.on('message', function (message) {
2   log('Got message: ', message);
3   socket.broadcast.emit('message', message);
4 });
```

Zweitens verwaltet er alle WebRTC-Videochat-'Räume':

```
1 if (numClients === 0) {
2   socket.join(room);
3   socket.emit('created', room, socket.id);
4 } else if (numClients === 1) {
5   socket.join(room);
6   socket.emit('joined', room, socket.id);
7   io.sockets.in(room).emit('ready');
8 } else { // max two clients
9   socket.emit('full', room);
10 }
```

Wie im Code ersichtlich ist, erlaubt unsere Applikation maximal zwei Peers in einem Raum.

Zusätzlich werden in unserer Applikation die Positionen der Gameobjekte über den Server synchronisiert (TODO).

STUN und TURN

WebRTC ist grundsätzlich so konzipiert, dass es Peer-to-Peer funktioniert. Benutzer können sich also auf dem direktesten Weg verbinden. Die Technologie ist jedoch darauf ausgelegt, mit realen Netzwerken zurechtzukommen: Client-Anwendungen müssen NAT-Gateways und Firewalls überwinden, und Peer-to-Peer-Netzwerke benötigen Fallbacks, falls die direkte Verbindung ausfällt. Als Teil dieses Prozesses verwenden WebRTC-APIs sogenannte *STUN-Server*, um die IP-Adresse ihres Computers zu ermitteln, und *TURN-Server* (Traversal Using Relay NAT), welche als Relay-Server fungieren, falls die Peer-to-Peer-Kommunikation fehlschlägt. [<https://webrtc.org/getting-started/turn-server>]

Palim-Palim verwendet als STUN-Server öffentlich verfügbare Server von Google. Als TURN UDP- und TCP-Fallback-Verbindungen werden die Konfigurationen aus dem WebRTC-Sample-Projekt übernommen. Diese Server

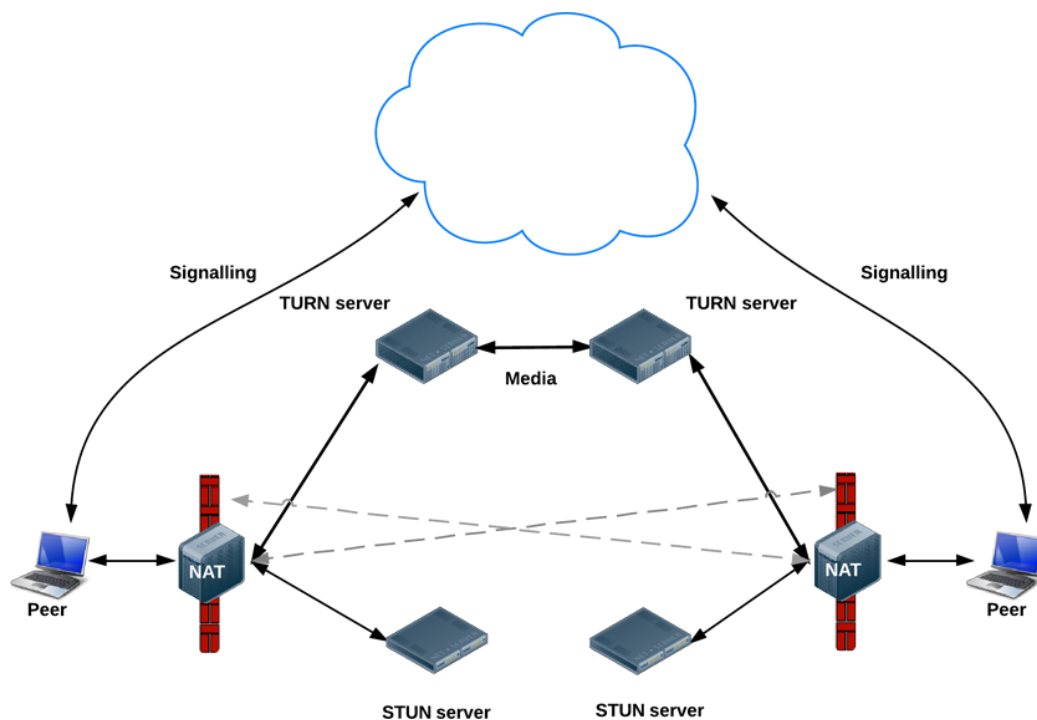


Figure 1: STUN, TURN, und Signalisierung in WebRTC. Quelle: <https://www.html5rocks.com/en/tutorials/webrtc/infrastructure/>.

werden gratis angeboten und können mit Tools wie Google's Trickle ICE sample [<https://webrtc.github.io/samples/src/content/peerconnection/trickle-ice/>] auf ihre Verfügbarkeit getestet werden [<https://bloggeek.me/webrtc-turn/>]. Die STUN- und TURN-Server-Adressen müssen bei dem Aufbau der Peer-Connection vom Client an den Signaling-Server übermittelt werden. Dazu übermittelt jeder Palim-Palim-Client beim Verbindungsaufbau folgende pcConfig-Werte an den Server:

```

1      var pcConfig = {
2      'iceServers': [
3      {
4      'urls': 'stun:stun.l.google.com:19302'
5      },
6      {
7      'urls': 'turn:192.158.29.39:3478?transport=udp',
8      'credential': 'JZE0Et2V3Qb0y27GRntt2u2PAYA=',
9      'username': '28224511:1379330808'
10     },
11     {
12     'urls': 'turn:192.158.29.39:3478?transport=tcp',
13     'credential': 'JZE0Et2V3Qb0y27GRntt2u2PAYA=',

```

```
14     'username': '28224511:1379330808',
15   }
16 ]
17 };
```

In etwa 82 Prozent der Fälle ist die Peer-To-Peer-Verbindung stabil genug, und die TURN-Server werden überhaupt nicht benötigt. Jedoch müssen sie trotzdem bei jeder Verbindung angegeben werden. TODO more infos <https://www.callstats.io/blog/2017/10/26/webrtc-product-turn-server>

Sicherheit

Verschlüsselung ist für alle WebRTC-Komponenten obligatorisch, und seine JavaScript-APIs können nur von sicheren Quellen (HTTPS oder localhost) aus verwendet werden. Die Signalisierungsmechanismen sind allerdings nicht in den WebRTC-Standards definiert. Hier liegt es an den Entwicklern, sichere Protokolle zu verwenden. TODO;Für unser Projekt noch definieren und ausformulieren;

9.3 Sicherheit

9.4 Testing

9.5 Deployment und Betrieb (Anhang?)

Chapter 10

Fazit

- Zusammenfassung des Erreichten / Zielerreichung
- Zentrale Erkenntnisse
- Reflektion
- Mögliche Weiterentwicklungen (bezogen auf die Software)
- Weiterführende Forschung

Chapter 11

Literaturverzeichnis

Appendix A

Ehrlichkeitserklärung

Appendix B

Testprotokolle, weitere
Spielkonzepte/Ideenfindung...