

## **Titelblatt**

# Projektinformationen

Titel: Palim-Palim  
Projektnummer: 21FS\_I4DS08

## Projekt-Team

Daniel Obrist, 8iCbb  
[daniel.obrist@students.fhnw.ch](mailto:daniel.obrist@students.fhnw.ch)

Severin Peyer, 8iCbb  
[severin.peyer@students.fhnw.ch](mailto: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](mailto: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](mailto: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 Literaturrechereche zusammengefasst erläutern.
- Erst am Schluss schreiben!aa
- Wichtig für den ersten Eindruck

# Contents

<b>1 Einleitung</b>	<b>3</b>
<b>2 Umfeldanalyse und Zielgruppe</b>	<b>6</b>
<b>3 Intergenerationelles Spielen</b>	<b>7</b>
<b>4 Videochats in Videospiele</b>	<b>8</b>
<b>5 Aufstellung der Forschungsfragen</b>	<b>9</b>
5.1 Hauptfragestellung . . . . .	9
5.2 Einzelfragen im thematischen Zusammenhang . . . . .	10
5.2.1 Spezifische Fragestellung zur User Experience: . . . . .	10
5.2.2 Fragestellung zur Kommunikation: . . . . .	10
<b>6 Methoden</b>	<b>12</b>
<b>7 Spieletests und Resultate</b>	<b>13</b>
7.1 Resultate . . . . .	13
7.2 Planung und Durchführung der Spieletests . . . . .	13
<b>8 Gamedesign</b>	<b>15</b>
8.1 Spielkonzept . . . . .	15
8.1.1 Gameplay-Loops . . . . .	15
8.2 Spielvarianten . . . . .	16
8.3 Design und Usability . . . . .	16
<b>9 Implementation</b>	<b>17</b>
9.1 Technologien . . . . .	17
9.1.1 WebRTC . . . . .	17
9.2 Architektur . . . . .	21
9.2.1 Gampeplay-Synchronisation . . . . .	21
9.3 Sicherheit . . . . .	22

9.4	Testing . . . . .	22
9.5	Deployment und Betrieb (Anhang?) . . . . .	22
<b>10</b>	<b>Fazit</b>	<b>23</b>
<b>11</b>	<b>Literaturverzeichnis</b>	<b>24</b>
<b>A</b>	<b>Ehrlichkeitserklärung</b>	<b>25</b>
<b>B</b>	<b>Testprotokolle, weitere Spielkonzepte/Ideenfindung...</b>	<b>26</b>

# 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*
  - *Spielentwicklung (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)*

Die verwendeten Begriffe Kind und betagte Person werden dabei im Rahmen des Projekts wie folgt definiert:

**Kind** Person zwischen fünf und acht Jahren.

**Betagte Person** Person ab einem Alter von 65 Jahren ohne grössere mentale Beeinträchtigung. Diese beiden Personengruppen stellen zugleich die Zielgruppen für die durchzuführenden Spieltests dar.

# Chapter 3

## Intergenerationelles Spielen

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

## Chapter 4

# Videochats in Videospiele

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

# Chapter 5

## Aufstellung der Forschungsfragen

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

Aus der in der Einleitung formulierten Ausgangslage stellt sich folgende Game-Design-Frage: Wie lässt sich ein Videospiel für unterschiedliche Altersgruppen ansprechend gestalten? Da diese Frage schon in vielen Studien bezüglich intergenerationallem Spieldesign behandelt wurde [4], [5], [6], konzentriert sich das Projekt Palim-Palim in seiner Aufgabenstellung speziell auf den Aspekt der Video-Telefonie in Videospielen. Dies ist besonders interessant, da die Kombination von Videospielen mit Video-Telefonie eine grosse Möglichkeit bietet, Video-Chats für intergenerationale Altersgruppen attraktiver zu gestalten.

### 5.1 Hauptfragestellung

Mit Palim-Palim soll herausgefunden werden, wie Videospiele und Video-Telefonie kombiniert werden können. Zusätzlich sollen für diese Kombinationen die Auswirkungen auf die Interaktionen zwischen betagten Menschen und Kindern erforscht werden. Deshalb befasst sich das Projekt mit folgender übergeordneter Fragestellung:

**Wie lässt sich Video-Telefonie mit Video-Spielen kombinieren, damit zwei Personen (Kind und betagte Person) übers Netz miteinander spielen und sich gleichzeitig unterhalten können?**

## 5.2 Einzelfragen im thematischen Zusammenhang

Um die übergeordnete Aufgabenstellung messbar zu machen, wird sich die begleitende Thesis von PalimPalim mit spezifischen Fragestellungen zu den Themen User Experience (siehe 2.2.1) und Kommunikation (siehe 2.2.2) auseinandersetzen. Die dazu formulierten Hypothesen können dabei durch die Auswertung der Resultate aus den Spieltests verifiziert oder widerlegt werden.

### 5.2.1 Spezifische Fragestellung zur User Experience:

1. Wie beeinflusst die Einbindung von Video-Telefonie die User Experience in Videospielen?

**Fragestellung 1a:** Welche Wirkung hat ein integrierter Video-Chat auf das Spielerlebnis von betagten Menschen?

**Hypothese 1a:** Ein integrierter Video-Chat hat eine positive Wirkung auf das Spielerlebnis von betagten Menschen.

**Fragestellung 1b:** Welche Wirkung hat ein integrierter Video-Chat auf das Spielerlebnis von Kindern?

**Hypothese 1b:** Ein integrierter Video-Chat hat eine positive Wirkung auf das Spielerlebnis von Kindern.

**Fragestellung 1c:** Welche Auswirkungen hat die Art und Weise, wie der Video-Chat in das Spiel integriert ist, auf das Spielerlebnis?

**Hypothese 1c:** Je stärker der Video-Chat ins Gameplay integriert ist, desto positiver ist das Spielerlebnis.

### 5.2.2 Fragestellung zur Kommunikation:

2. Wie beeinflusst die Einbindung von Video-Telefonie die Kommunikation in Videospielen?

**Fragestellung 2a:** Welche Wirkung hat ein integrierter Video-Chat auf die Kommunikation zwischen den Spielenden?

**Hypothese 2a:** Ein im Spiel integrierter Video-Chat fördert die Kommunikation zwischen den Spielenden.

**Fragestellung 2b:** Wird ein integrierter Video-Chat aktiv als Kommunikationsmittel zur Bewältigung von Spielaufgaben genutzt?

**Hypothese 2b:** Der Video-Chat wird aktiv als Kommunikationsmittel zur Bewältigung der Spielaufgabe verwendet.

**Fragestellung 2c:** Welche Auswirkungen hat die Art und Weise, wie der Video-Chat in das Spiel integriert ist, auf die Förderung der Kommunikation?

**Hypothese 2c:** Je stärker der Video-Chat ins Spiel integriert ist, desto angeregter ist der Austausch zwischen den Spielenden.

# Chapter 6

## Methoden

*In Palim-Palim verwendete Methoden*

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

# Chapter 7

## Spiletests und Resultate

### 7.1 Resultate

- *Ergebnisse*
- *Beantwortung der aufgestellten Forschungsfragen*

### 7.2 Planung und Durchführung der Spiletests

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

Die Überprüfung der im Kapitel 2 aufgestellten Hypothesen wird durch Spieltests mit den unterschiedlichen Spielvarianten geschehen. Das Ziel dabei ist es, mit den Methoden Beobachtung und Befragung die User Experience sowie die Spielenden-Kommunikation messbar zu machen. Das Beobachten dient einerseits dazu, sofortige und auch unterbewusste Reaktionen bei den Testpersonen zu erkennen. Andererseits möchten wir mittels Befragungen qualitatives Feedback zu den einzelnen Spielvarianten einholen. Dazu wird für die Spieltests vorab ein passender Fragenkatalog erarbeitet, bzw. ein entsprechendes Beobachtungs-Protokoll vorbereitet und geführt.

In der Projektwoche 12 wird mit einem funktionalen Prototyp ein erstes Mal Feedback zu den Grundmechaniken eingeholt. Diese erste Standortbestimmung wird mit Bekannten durchgeführt.

Für die tatsächlichen Spieltests werden sechs Test-Paare benötigt, wobei ein Test-Paar sich aus einer betagten Person sowie einem Kind zusammensetzt. In der Woche 19 und 20 wird mit allen Paaren getestet. Um die Testpersonen zu finden, wird mit der Organisation terzStiftung [14] zusammengearbeitet. Mithilfe eines Newsletters werden die Mitglieder der terzStiftung für ein Testing angefragt. Die angefragten betagten Menschen sind mindestens 65 Jahre alt und weisen eine höhere Technik-Affinität auf als ein Abbild der Gesamtheit dieser Altersgruppe.

Nach Möglichkeit (Situation SARS-CoV-2) werden die Spieltests bei den Probanden zuhause, an der FHNW in Brugg, an einem öffentlichen Ort oder remote durchgeführt (absteigende Priorität). Bei den drei erstgenannten Varianten würde das Projektteam die Beobachtungen direkt vor Ort vornehmen, sollten die Tests remote durchgeführt werden müssen, würden die Projektmitglieder sich online zuschalten. Die Probanden müssen nicht zwingend ein Tablet besitzen, dies kann Ihnen von uns zur Verfügung gestellt werden. Jedoch muss bereits vor den Tests ein solches benutzt worden sein.

# Chapter 8

## Gamedesign

### 8.1 Spielkonzept

Das dem Videospiel zugrunde liegende Spielkonzept lehnt sich dem Kinder-Kaufladen-Spielprinzip aus dem echten Leben an. Das Kind und die betagte Person schlüpfen dabei in die entsprechenden Rollen von Verkaufspersonal und Kundschaft in einem virtuellen Einkaufsladen. Die Spieler\*innen sehen dabei jeweils einen gemeinsamen Tresen aus der Perspektive ihrer Rolle (siehe Abbildung 1). Beide Seiten können Objekte auf dem Tresen platzieren, beziehungsweise Objekte vom Tresen wegziehen. Zusätzlich besteht die Möglichkeit, die Objekte auf den Video-Stream des Gegenübers zu ziehen. Durch all diese Aktionen können je nach Objekt diverse Animationen, Filter-Effekte und Gameplay-Events ausgelöst werden.

#### 8.1.1 Gameplay-Loops

Durch die Aufteilung in die beiden Rollen lassen sich für beide Spieler\*innen verschiedene Aktionen und somit ein unterschiedliches Gameplay definieren. Diese Art der Unterteilung eignet sich hervorragend, um das Gameplay jeweils auf die spezifischen Anforderungen von Kindern bzw. betagten Personen anpassen zu können.

Der Gameplay-Loop ist in der nachfolgenden Abbildung 2 skizziert. Die Haupt-Interaktionen zwischen den beiden Spieler\*innen bestehen aus der Kommunikation der Einkaufsliste und dem Hin- und Herreichen von Objekten auf der gemeinsamen Tresen-Fläche.

*Gameplay-Loops*

Dieses übergeordnete Gameplay soll als Leitfaden für den Spielverlauf dienen. Allerdings bietet das Spiel durch die gemeinsame Tresen-Fläche und dem interaktiven Video-Stream bewusst Möglichkeiten für sonstige Interaktionen mit dem Gegenüber. Damit soll für die Benutzerinnen und Benutzer ein gewisser Freiraum entstehen, in welchem sie den Verlauf des Spiels auf spielerische Art und Weise beeinflussen können. Dies soll das Spiel besonders für das Kind interessanter machen, um eine aktive und lustige Kommunikation mit der betagten Person zu fördern.

## 8.2 Spielvarianten

Damit der Einfluss der Video-Telefonie als Teil des Spiels messbar wird, werden unterschiedliche Varianten des Spiels implementiert und getestet:

1. Spiel ohne Video-Chat
2. Spiel mit Video-Chat, aber getrennt von der Spiel-Szene
3. Spiel mit Video-Chat, in die Spiel-Umgebung integriert
4. Spiel mit Video-Chat, in die Spiel-Umgebung integriert und optionale Interaktionsmöglichkeiten mit dem Video-Stream
5. Spiel mit Video-Chat, speziell in die Spiel-Umgebung integriert und zum Spielerfolg notwendige Interaktionsmöglichkeiten mit dem Video-Stream

*Tabelle Spielvarianten*

## 8.3 Design und Usability

# Chapter 9

## Implementation

### 9.1 Technologien

#### 9.1.1 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 und ist eine Empfehlung des World Wide Web Consortium (W3C) [<https://www.w3.org/TR/webrtc/>] und ein Standard der Internet Engineering Task Force (IETF) [<https://www.ietf.org/blog/webrtc-support/>]. Dabei wird zwischen zwei Browsern eine Peer-To-Peer-Verbindung aufgebaut, worüber die Daten gestreamt werden. Auf der Peer-To-Peer-Verbindung lassen sich auch eigene DataChannels erstellen, die z.B. zur Übertragung von Text-Nachrichten, Positions-Daten von Game-Objekten oder sogar Files zwischen den Peers verwendet werden können [<https://webrtc.ventures/2016/05/5-uses-of-the-data-channel/>].

#### 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/signal-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 wir 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 einen eigenen DataChannel zwischen den Peers synchronisiert (TODO DataChannel).

## 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>]

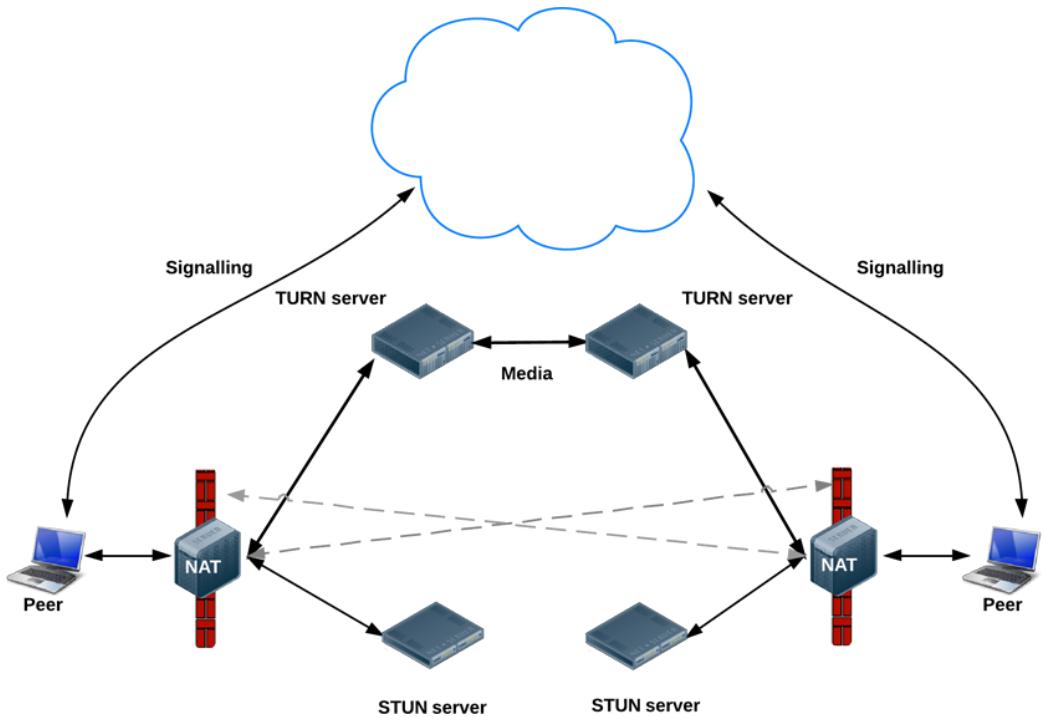


Figure 1: STUN, TURN, und Signalisierung in WebRTC. Quelle: <https://www.html5rocks.com/en/tutorials/webrtc/infrastructure/>.

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 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 Verbindungsauftbau 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. Video, Audio oder Daten zwischen zwei Peers werden mit Secure Real Time Protocol (SRTP) verschlüsselt. SRTP verschlüsselt die Session, sodass niemand die Nachricht ohne die richtigen Verschlüsselungsschlüssel entschlüsseln kann. Unverschlüsselte Kommunikation gibt es in WebRTC nicht. Schlüssel werden ebenfalls nicht über den Signaling Server ausgetauscht, sondern nur direkt zwischen den Peers. Dazu kommt HTTPS zum Signaling-Server und die im Browser und inn Betriebssystem vorgegebenen Sicherheiten. [https://webrtc-security.github.io/] [https://webrtc-security.github.io/] TODO; Für unser Projekt noch definieren und ausformulieren gemäss Quellen;

## 9.2 Architektur

### 9.2.1 Gampeplay-Synchronisation

Um die Interaktionen der beiden Spieler zu synchronisieren, wird normalerweise ein Server verwendet, der die Inputs der Spieler entgegennimmt, und als zentrale Autorität den Zustand des Spiels bestimmt. Bei Palim-Palim wurde dies aber anders gelöst. Das Spiel nutzt seine Peer-To-Peer-Funktionalität aus und kreiert neben dem Video- und Audiostream auch einen spezifischen DataChannel für Game-Updates. Dieser Channel wird genutzt, um die Objekte beider Szenen zu synchronisieren.

Der DataChannel überträgt JSON-Strings via UDP. UDP als Protokoll ist sehr schnell und deshalb sind die Änderungen auch fast ohne Verzögerung beim Peer sichtbar. Die Wahl des Protokolls kann beim Erstellen des DataChannels gewählt werden [Code Erstellung DataChannel]. Eine weitere Eigenschaft von UDP ist, dass die Reihenfolge der Übertragung nicht garantiert ist - im Gegensatz zu TCP z.B. Da die Game-Updates sehr oft geschickt werden, ist es in diesem Fall egal, in welcher Reihenfolge die Nachrichten ankommen. (Man muss sich das für jeden Fall überlegen...)

```
1 const dataChannel = peerConnection.createDataChannel('  
  gameUpdates', {  
2   ordered: false,  
3   id: room  
4 });  
5 dataChannel.onmessage = handleReceiveMessage;  
6 dataChannel.onerror = handleError;  
7 dataChannel.onopen = handleDataChannelStatusChange;  
8 dataChannel.onclose = handleDataChannelStatusChange;
```

Diese Art der Gameplay-Synchronisation erlaubt es den Clients, total unabhängig von einem Server zu spielen. Da der Server nur fürs Signaling benutzt wird, ist auch eine grosse Skalierung der Spieleranzahl denkbar. Der Server muss die Spieler nur initial vermitteln, was keine grosse Sache ist. Alle anderen Berechnungen sowie der Abgleich der Spielwelt werden auf den Clients direkt vorgenommen. Da kein Umweg über einen Server genommen werden muss, ist auch die Latenz niedrig. (Einziges Manko: es gibt keine zentrale Autorität. Das heißt Spieler könnten den Client-Code manipulieren und sich unfaire Vorteile verschaffen. Da Palim-Palim und sein Publikum aber nicht kompetitiv sind, ist diese Gefahr des Cheatings vernachlässigbar.)

**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...**