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

Zusammenfassung

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

Inhaltsverzeichnis

1 Einleitung **3**

2 Umfeldanalyse und Zielgruppe **6**

3 Intergenerationelles Spielen **7**

4 Videochats in Videospielen **8**

5 Aufstellung der Forschungsfragen **9**
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

6 Methoden **12**

7 Spieletests und Resultate **13**
7.1 Resultate 13
7.2 Planung und Durchführung der Spieletests 13

8 Gamedesign **15**
8.1 Spielkonzept 15
8.1.1 Gameplay-Loops 15
8.2 Spielvarianten 16
8.3 Design und Usability 16

9 Implementation **17**
9.1 Technologien 17
9.1.1 WebRTC 17
9.1.2 Three.js 22
9.2 Architektur 23
9.2.1 Gampeplay-Synchronisation 23

9.3 Testing 24
9.4 Deployment und Betrieb (Anhang?) 24

10 Fazit **25**

11 Literaturverzeichnis **26**

A Ehrlichkeitserklärung **29**

B Testprotokolle, weitere Spielkonzepte/Ideenfindung... **30**

Kapitel 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

Kapitel 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.

Kapitel 3

Intergenerationelles Spielen

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

Kapitel 4

Videochats in Videospielen

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

Kapitel 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 intergenerationellem Spieledesign 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 intergenerationelle 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.

Kapitel 6

Methoden

In Palim-Palim verwendete Methoden

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

Kapitel 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*

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 [9] 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.

Kapitel 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

Kapitel 9

Implementation

9.1 Technologien

Palim-Palim kombiniert die Funktionaliäten einer Video-Chat-Applikation mit einem Multiplayer-Spiel. Um diesen Anforderungen gerecht zu werden, wurden entsprechende Technologien zur Umsetzung ausgewählt. Dabei fiel die Wahl auf WebRTC zur Übertragung der Multimedia-Daten, sowie Three.js als 3D-Library für die Gestaltung und Umsetzung der 3D-Gameplay-Szene.

9.1.1 WebRTC

WebRTC ist ein Open-Source-Projekt, das die Echtzeitkommunikation von Audio, Video und Daten in Web- und nativen Anwendungen ermöglicht [10]. Die Technologie ist in allen modernen Browsern sowie auf nativen Clients für alle wichtigen Plattformen verfügbar, ist eine Empfehlung des World Wide Web Consortium (W3C) [11] und ein Standard der Internet Engineering Task Force (IETF) [12]. 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 Daten-Kanäle erstellen, die z.B. zur Übertragung von Text-Nachrichten, Positions-Daten von Game-Objekten oder sogar Files zwischen den Peers verwendet werden können [13].

Signaling-Server

WebRTC verwendet die RTCPeerConnection JavaScript API, um Streaming-Daten direkt zwischen Browsern zu kommunizieren [14]. Doch wie wird dieser Prozess initiiert? Die beiden Peers - genannt Alice und Bob - kennen sich zu Beginn ja gar noch nicht. Zusätzlich wird also einen Mechanismus benötigt,

welcher die Kommunikation der beiden Peers koordiniert und Kontrollnachrichten sendet. Dieser Prozess wird als in WebRTC als *Signaling* bezeichnet. Signaling-Methoden und -protokolle sind von WebRTC nicht spezifiziert und können je nach Anwendungsfall entsprechend gewählt werden.
In Palim-Palim wurde die JavaScript-Library Socket.io für die Signalisierung verwendt [15]. Socket.io erlaubt eine einfache und bidirektionale Kommunkation zwischen den Clients und einem Signaling-Server. Mit seinem integrierten Room-Konzept eignet sich Socket.io zudem sehr gut für eine Video-Chat-App. Eine Node.js Server-Applikation fungiert als Signaling-Server für Palim-Palim. Der Server hat dabei folgende zwei Aufgaben: er dient als Message Relay und verwaltet alle Videochat-Räume.

Die Funktionaliät als Message Relay ist wichtig, da sich Alice und Bob vor dem Verbindungsaufbau noch nicht kennen. Ein Signaling-Server, welcher beiden Clients bekannt ist, wird benötigt. So können initiale Informationen von Alice zu Bob ausgetauscht werden, damit diese untereinander eine WebRTC-Peer-Verbindung aufbauen können:

```
1  socket.on('message', function (message) {
2    socket.to(room).emit('message', message);
3  });
```

Zusätzlich verwaltet der Signaling-Server alle WebRTC-Videochat-'Räume'. Tritt z.B. Alice einem Raum bei, so überprüft der Server, ob die Raumkapazität schon erreicht wurde und sendet entsprechende Antworten an sie:

```
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 Palim-Palim maximal zwei Peers in einem Raum. Ist Alice die erste Spielerin, erhält sie die Antwort 'created' vom Signaling-Server. Ist bereits ein Client in dem Raum, wird ihr ein 'joined' zurückgegeben. Wurde die Raum-Kapazität überschritten, sendet der Palim-Palim Server Alice ein 'full'. Diese Nachrichten werden dann Client-seitig im Browser von Alice entsprechend behandelt.

STUN- und TURN-Server

WebRTC ist grundsätzlich so konzipiert, dass es Peer-to-Peer funktioniert. Alice und Bob können sich also auf dem direktesten Weg verbinden. Die Technologie ist jedoch auch bewusst darauf ausgelegt, mit realen Netzwerken zurechtzukommen: Client-Anwendungen müssen NAT-Gateways (Network Address Translation) und Firewalls überwinden, und Peer-to-Peer-Netzwerke benötigen Fallbacks, falls die direkte Verbindung ausfällt. Als Teil dieses Prozesses verwendet die WebRTC-API zwei Netzwerkprotokolle als Hilfsmittel:

- *STUN* (Session Traversal Utilities for NAT), um die öffentliche IP-Adresse und Port-Nummer für den direkten Kontaktaufbau zu ermitteln [16].
- *TURN* (Traversal Using Relay NAT), welches die Kommunikation über NAT- oder Firewallgrenzen hinweg ermöglicht und als Fallback-Relay genutzt werden kann, sollte keine Peer-to-Peer-Verbindung möglich sein [17].

Für beide Protokolle müssen Server bereitgestellt werden, welche den WebRTC-Clients bekannt sein müssen. Die STUN- und TURN-Adressen müssen bei dem Aufbau der Peer-Connection der Clients an den Signaling-Server übermittelt werden [18]. Dazu setzt jeder Palim-Palim-Client beim Verbindungsaufbau in den Konfigurations-Werten die Addressen der STUN- und TURN-Server:

```
1       this.peerConnectionConfig = {
2             'iceServers': [
3                 {
4                   'urls': 'stun:stun.l.google.com:19302'
5                 },
6                 {
7                   'urls': 'turn:86.119.43.130:3478',
8                   'credential': '*****************',
9                   'username': 'palimpalim'
10                }
11            ]
12        };
```

Nur mit diesen beiden Server-Addressen im entsprechenden Konstruktor kann die Funktionalität des Video- und Audio-Streams über das Internet gewährleistet werden. Palim-Palim verwendet als STUN-Server einen öffentlich verfügbaren Server von Google. Da der STUN-Server nur zur Ermittlung der eigenen öffentlichen IP-Addresse dient, lässt sich hier ohne gros-

se Auswirkungen auf den Datenschutz oder die Sicherheit ein öffentlich gehosteter Server verwenden.

Als Fallback-Verbindungen wird allerdings im produktiven Betrieb einer WebRTC-Applikation unbedingt ein eigener TURN-Server mit einer öffentlich sichtbaren IP-Adresse benötigt [19]. Es gibt keine kostenlosen, öffentlich gehosteten TURN-Server, da der Netzwerkverkehr über so einen Server sehr stark ansteigen kann. Daher lohnt es sich für niemanden, fremden Verkehr über seinen TURN-Server zuzulassen. Es ist also ebenfalls von Vorteil, seinen TURN-Server mit einer ensprechenden Authentifizierung zu versehen. Palim-Palim verwendet einen auf SwitchEngines gehosteten TURN-Server. Dieser Linux-Server verwendet Coturn, eine Open-Source Implementierung des TURN-Protokolls [20].

Gemäss Auswertungen des WebRTC Call Quality Dienstleisters callstats.io haben im Durchschnitt etwa 30 Prozent aller WebRTC-Sessions einen Client, der sich über einen TURN-Server verbindet [21]. Das heisst in etwa 70 Prozent der Fälle ist die Peer-To-Peer-Verbindung stabil genug, und ein TURN-Server wird überhaupt nicht benötigt. Jedoch müssen sie trotzdem bei jeder Verbindung angegeben werden. Denn gewisse Benutzer*innen wären ohne die Unterstützung eines TURN-Relay-Servers nicht in der Lage zu kommunizieren.

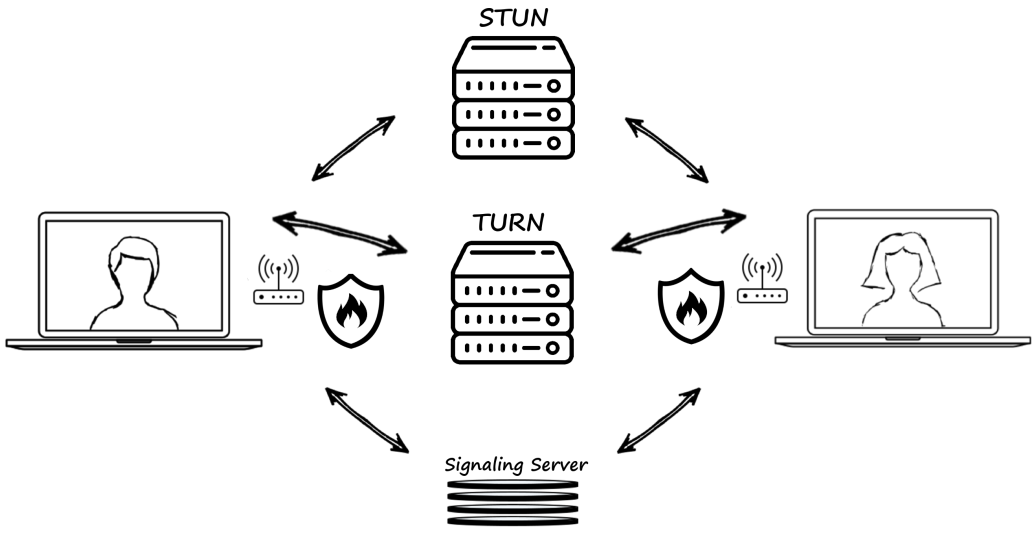

Abbildung 1: STUN, TURN, und Signalisierung in WebRTC (eigene Darstellung).

Mit STUN-, TURN- und Signaling-Server besitzt Palim-Palim ein stabiles Backend für die Implementierung des Video-Streams mit WebRTC. Zusätzlich kann die Peer-To-Peer-Funktionalität von WebRTC auch zur Synchronisation von Gameplay-Daten direkt zwischen Alice und Bob genutzt werden. Die Multiplayer-Funktionalität kann so auch unabhängig von einem zentralen Server als gemeinsame Authorität ermöglicht werden.

Sicherheit

Der Open-Source-Charakter von WebRTC kann bei potenziellen Anwender*innen der Technologie Sicherheitsbedenken hervorrufen. Diese Bedenken sind berechtigt und wurden bei der Entwicklung von Palim-Palim ebenfalls berücksichtigt. Damit der Datenschutz für die Spieler*innen sowie die Stabilität des Games gewährleistet werden kann, müsssen bei der Implementation einige Punkte berücksichtigt werden.

Auf der Protokoll-Ebene ist WebRTC als Standard sehr sicher und auch bereits etabliert. Verschlüsselung ist für alle WebRTC-Komponenten obligatorisch, und seine JavaScript-APIs können nur von sicheren Quellen (HTTPS oder localhost) aus verwendet werden [22]. Das verwendete Verschlüsselungs-Protokoll hängt dabei vom Kanaltyp ab. Daten-Kanäle werden mit Datagram Transport Layer Security (DTLS) und Medien-Kanäle mit Secure Real-time Transport Protocol (SRTP) verschlüsselt. Unverschlüsselte Kommunikation gibt es in WebRTC also nicht. Die Schlüssel zur Entschlüsselung der Median-Kanäle werden zudem nicht über den Signaling-Server ausgetauscht, sonder nur direkt zwischen den Peers.

Ebenfalls werden bei der Umleitung des Datenverkehrs über den TURN-Server die Daten nicht interpretiert oder modifiziert. Dies ist so im TURN-Standard definiert [23]. Das heisst, die Verwendung des TURN-Servers fügt dem WebRTC-Datenverkehr keine Sicherheitsschwachstellen hinzu. Die Verbindung zum Signaling-Server ist ebenfalls mit HTTPS geschützt, was als sicher genug für Online-Banking und Behörden-Websites gilt [24].

Somit lässt sich WebRTC als eigenständiges Framework für Videotelefonie als sehr sicher bezeichnen. Allerdings hängt diese Sicherheit auch von der umliegenden Web-Applikation und diese wiederum von dem verwendeten Browser ab. Als eine im Internet zugängliches Spiel sollte Palim-Palim deshalb über eine entsprechende User-Authentifizierung erweitert werden, sollte das Spiel dauerhaft produktiv betrieben werden wollen. Ebenfalls empfiehlt es sich, die WebRTC-Bibliothek und andere Abhängigkeiten regelmässig zu

aktualisieren.

9.1.2 Three.js

Three.js Zusammenfassung

Grundlagen

- *Aufbau einer 3D-Szene in Three.js erklären*

3D-Objekte laden

3D-Objekte in Three.js nutzen (https://discoverthreejs.com/book/first-steps/load-models/)

- *Beschaffenheit und Laden von GLTF-Files in Palim-Palim erklären*

Interaktionen

- *Interaktionskonzept DragControls und 3D-Szene skizzieren und erklären*

Physik

- *Versuche mit ammojs und headless Möglichkeiten auf Server*

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 Authorität 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 Erstllung 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 inital 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 Authorität. 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 Testing

9.4 Deployment und Betrieb (Anhang?)

Kapitel 10

Fazit

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

Kapitel 11

Literaturverzeichnis

[1] Bianca Glaab. "Silver Surfer erobern das Netz". In: *NovaCura* 46.4 (2015), S. 24–25. URL: https://so.prosenectute.ch/dam/jcr:c906daad-93fc-47f0-9900-c5b18b6d3a3c/Glaab_Video_Chat_gegen_Einsamkeit.pdf (besucht am 15. 03. 2021).

[2] Carly Tulloch. *7 Ways to Engage Young Children On Video Calls*. Wee Talkers. 19. Okt. 2020. URL: https://www.weetalkers.com/blog/engage-young-children-on-video-calls (besucht am 14. 06. 2021).

[3] Sean Follmer u. a. "Video Play: Playful Interactions in Video Conferencing for Long-Distance Families with Young Children". In: *Proceedings of the 9th International Conference on Interaction Design and Children*. IDC '10. New York, NY, USA: Association for Computing Machinery, 2010, S. 49–58. ISBN: 978-1-60558-951-0. DOI: 10.1145/1810543.1810550. URL: https://doi.org/10.1145/1810543.1810550.

[4] Puay-Hoe Chua u. a. "Let's play together: Effects of video-game play on intergenerational perceptions among youth and elderly participants". In: *Computers in Human Behavior* 29.6 (Nov. 2013), S. 2303–2311. ISSN: 07475632. DOI: 10.1016/j.chb.2013.04.037. URL: https://linkinghub.elsevier.com/retrieve/pii/S0747563213001490 (besucht am 15. 03. 2021).

[5] Teresa De la Hera u. a. "Benefits and Factors Influencing the Design of Intergenerational Digital Games: A Systematic Literature Review". In: *Societies* 7.3 (2017). ISSN: 2075-4698. DOI: 10.3390/soc7030018. URL: https://www.mdpi.com/2075-4698/7/3/18.

[6] Marco Soldati u. a. "Create Video Games to Promote Well-Being of Elderly People – A Practice-Driven Guideline". In: *Human Aspects of IT for the Aged Population. Healthy and Active Aging*. Hrsg. von Qin

Gao und Jia Zhou. Cham: Springer International Publishing, 2020, S. 401–418. ISBN: 978-3-030-50249-2.

[7] Jan Derboven, Mieke Van Gils und Dirk De Grooff. "Designing for collaboration: a study in intergenerational social game design". In: *Universal Access in the Information Society* 11.1 (März 2012), S. 57–65. ISSN: 1615-5289, 1615-5297. DOI: 10.1007/s10209-011-0233-0. URL: http://link.springer.com/10.1007/s10209-011-0233-0 (besucht am 15. 03. 2021).

[8] Marco Soldati. *FHNW Myosotis-Garden*. FHNW Myosotis-Garden. 2015. URL: https://www.fhnw.ch/de/die-fhnw/hochschulen/ht/institute/institut-fuer-data-science/fhnw-myosotis-garden (besucht am 14. 06. 2021).

[9] *terzStiftung, Interessensvertreter reiferer Menschen*. terzStiftung. URL: https://www.terzstiftung.ch/ (besucht am 14. 06. 2021).

[10] *WebRTC*. WebRTC. 2011. URL: https://webrtc.org/?hl=de (besucht am 14. 06. 2021).

[11] *WebRTC 1.0: Real-Time Communication Between Browsers*. 26. Jan. 2021. URL: https://www.w3.org/TR/webrtc/ (besucht am 14. 06. 2021).

[12] *Support for WebRTC*. IETF News. 26. Jan. 2021. URL: /blog/webrtc-support/ (besucht am 14. 06. 2021).

[13] *WebRTC DataChannel*. WebRTC. URL: https://webrtc.org/getting-started/data-channels?hl=de (besucht am 14. 06. 2021).

[14] *RTCPeerConnection - Web APIs | MDN*. URL: https://developer.mozilla.org/en-US/docs/Web/API/RTCPeerConnection (besucht am 14. 06. 2021).

[15] Damien Arrachequesne. *Socket.IO*. Socket.IO. 11. Juni 2021. URL: https://socket.io/index.html (besucht am 14. 06. 2021).

[16] *STUN*. In: *Wikipedia*. Page Version ID: 1016984334. 10. Apr. 2021. URL: https://en.wikipedia.org/w/index.php?title=STUN&oldid=1016984334 (besucht am 14. 06. 2021).

[17] *Traversal Using Relays around NAT*. In: *Wikipedia*. Page Version ID: 1001583358. 20. Jan. 2021. URL: https://en.wikipedia.org/w/index.php?title=Traversal_Using_Relays_around_NAT&oldid=1001583358 (besucht am 14. 06. 2021).

[18] *TURN-Server*. WebRTC. URL: https://webrtc.org/getting-started/turn-server?hl=de (besucht am 14. 06. 2021).

[19] *WebRTC TURN Servers: When you NEED it • BlogGeek.me*. BlogGeek.me. 20. Juli 2020. URL: https://bloggeek.me/webrtc-turn/ (besucht am 14. 06. 2021).

[20] *coturn/coturn*. original-date: 2015-07-17T08:15:16Z. 14. Juni 2021. URL: https://github.com/coturn/coturn (besucht am 14. 06. 2021).

[21] callstats. *Why Does Your WebRTC Product Need a TURN Server?* URL: https://www.callstats.io/blog/2017/10/26/webrtc-product-turn-server (besucht am 14. 06. 2021).

[22] *WebRTC Encryption and Security: Everything You Need to Know | Wowza*. Wowza Media Systems. 5. Nov. 2020. URL: https://www.wowza.com/blog/webrtc-encryption-and-security (besucht am 14. 06. 2021).

[23] *rfc5766*. URL: https://datatracker.ietf.org/doc/html/rfc5766 (besucht am 14. 06. 2021).

[24] *A Study of WebRTC Security · A Study of WebRTC Security*. URL: https://webrtc-security.github.io/ (besucht am 14. 06. 2021).

Anhang A

Ehrlichkeitserklärung

Anhang B

Testprotokolle, weitere Spielkonzepte/Ideenfindung...