



SOFTWARE-ENTWICKLUNGSPRAKTIKUM

OFFICEMANIA SEC0

Software-Entwicklungspraktikum (SEP)
Sommersemester 2021

Fachentwurf

Auftraggeber
Technische Universität Braunschweig
Institut für Systemsicherheit
Prof. Dr. Konrad Rieck
Mühlenpfordtstraße 23
38106 Braunschweig

Betreuer: Erwin Quiring, Alexander Warnecke, Jonas Möller

Auftragnehmer:

Name	E-Mail-Adresse
Michael Goslar	m.goslar@tu-bs.de
Paul Hagedorn	paul.hagedorn@tu-bs.de
Johan Kolms	j.kolms@tu-bs.de
Eva Nortmann	e.nortmann@tu-bs.de
Joel Schaub	j.schaub@tu-bs.de
Han Thang	h.thang@tu-bs.de
Fabian Vizi	f.vizi@tu-bs.de
Lukas Wieland	l.wieland@tu-bs.de
Giulia Woywod	g.woywod@tu-bs.de

Braunschweig, 9. Juni 2021

Bearbeiterübersicht

Kapitel	Autoren	Kommentare
1	Michael Goslar, Han Thang, Joel Schaub, Eva Nortmann	...
1.1	Michael Goslar, Han Thang, Joel Schaub, Eva Nortmann	...
1.2	Michael Goslar, Han Thang, Joel Schaub, Eva Nortmann	...
2	Lukas Wieland	...
2.1 bis 2.8	Lukas Wieland, Johan Kolms, Giulia Woywod	...
3	Fabian Vizi	
4	Paul Hagedorn	...
5	Alle	...

Inhaltsverzeichnis

1	Einleitung	5
1.1	Projektübersicht	5
1.2	Projektdetails	6
1.2.1	Spielerbewegung	7
1.2.2	Videochat	8
2	Analyse der Produktfunktionen	9
2.1	Analyse von Funktionalität F10: Anwendungsaufruf	10
2.2	Analyse von Funktionalität F20: Kartenzeichnung	11
2.3	Analyse von Funktionalität F30: Spielerbewegung	12
2.4	Analyse von Funktionalität F40: Start Videokonferenz	13
2.5	Analyse von Funktionalität F50: Stop Videokonferenz	14
2.6	Analyse von Funktionalität F60: Anzeige aktueller Spielerzahl	15
2.7	Analyse von Funktionalität F70: Namensbestimmung	16
2.8	Analyse von Funktionalität F80: Spielerinteraktion	17
3	Datenmodell	18
4	Konfiguration	19
5	Glossar	20

Abbildungsverzeichnis

1.1	Systemablauf	6
1.2	Spieler bewegen	7
1.3	Videochat	8
2.1	Anwendungsaufruf	10
2.2	Kartenzeichnung	11
2.3	Spielerbewegung	12
2.4	Start Videokonferenz	13
2.5	Stop Videokonferenz	14
2.6	Anzeige aktueller Spielerzahl	15
2.7	Namensbestimmung	16
2.8	Spielerinteraktion	17

1 Einleitung

In diesem Dokument erklären wir, in einer fachlichen Beschreibung, die Funktionalität unseres Projekts. Hierfür wurde in Kapitel 1 ein Statechart angefertigt, welches einen Überblick über die Software gibt. Außerdem gibt es noch Aktivitätsdiagramme die komplizierte Zustände aus dem Statechart genauer erläutern. In Kapitel 2 werden die Produktfunktionen analysiert und mit Hilfe von diversen, dazugehörigen Sequenzdiagrammen, graphisch genauer erläutert. Im letzten Kapitel werden die Konfigurationen des Servers und config-Dateien und deren Funktion angegeben.

1.1 Projektübersicht

Unsere Webapplikation „*Officemania*“ ist eine virtuelle Büroumgebung, welche im Rahmen des Softwareentwicklungspraktikums 2021 erstellt wird.

Im Folgenden wird die Funktionsweise der Software anhand der verschiedenen Zustände des Statecharts (**Abbildung 1.1**) erklärt.

Die Software wird über einen Webaufruf gestartet. Zum Anfang wird überprüft, ob schon Cookies vorhanden sind. Ist dies nicht der Fall, wird man aufgefordert einen Namen einzugeben. Währenddessen wird im Hintergrund die Map berechnet und nach dem Setzen des Namens gezeichnet.

Dann wird überprüft, ob Spieler in der Nähe sind. Ist das der Fall, wird sofort der Videochat mit den Spielern in der Nähe gestartet. Als nächstes befindet sich der Spieler in einem Zustand, in dem er bereit für Aktionen ist. Hier ist Ton und Video standardmäßig angeschaltet, wobei der Stream zu Beginn immer ausgeschaltet ist. Per Tastendruck können die Zustände von angeschaltet zu ausgeschaltet und andersherum gewechselt werden.

Dauerhaft wird getestet, ob Spieler in der Nähe sind und ob somit ein Videochat gestartet wird. Außerdem kann sich ein Spieler jederzeit bewegen, mit Objekten interagieren, den Skin wechseln und seinen Namen ändern.

Möchte man das Programm beenden, kann man die Anwendung schließen und kommt somit im Statechart in den Endzustand.

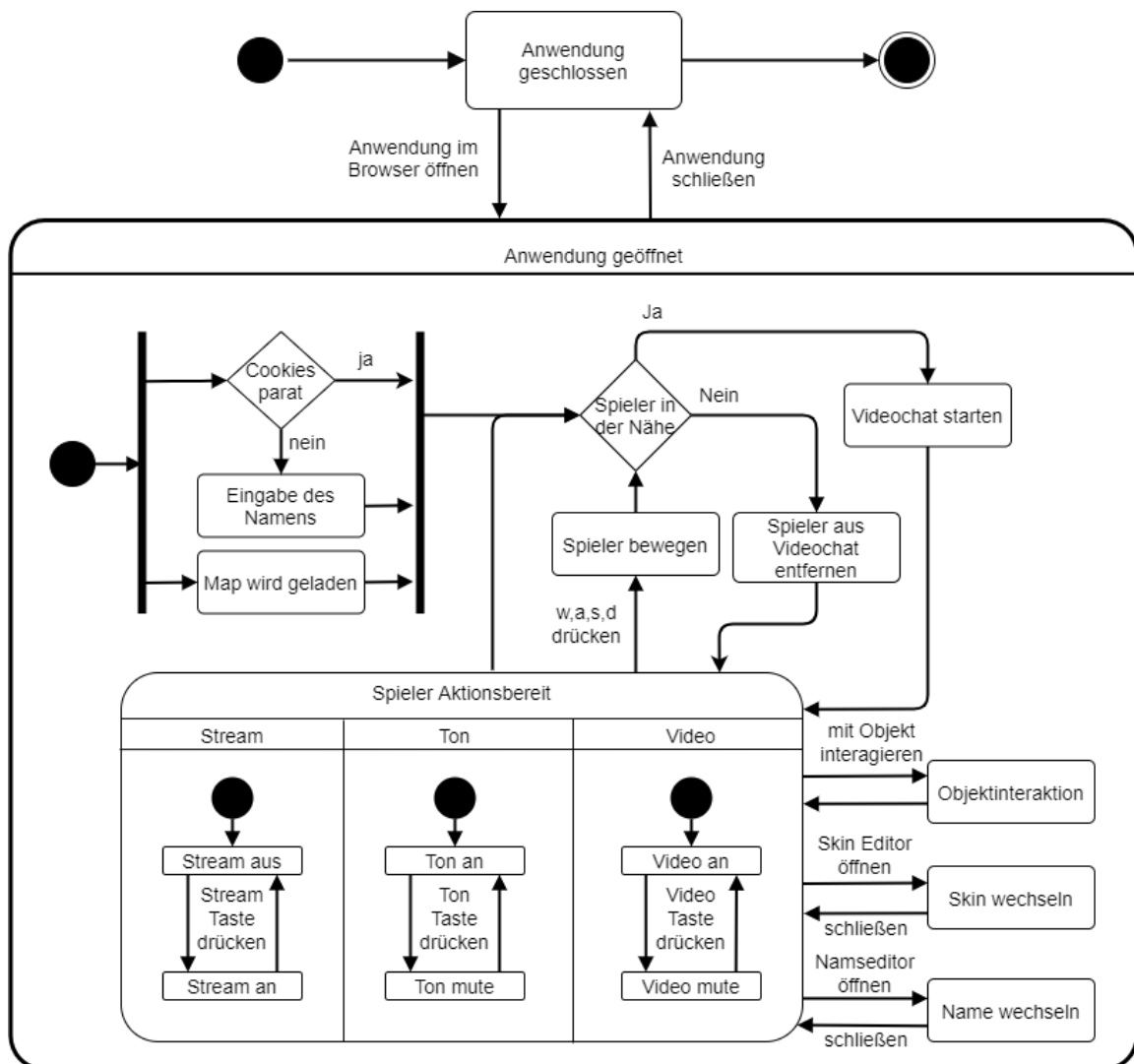


Abbildung 1.1: Systemablauf

1.2 Projektdetails

In diesem Abschnitt werden bestimmte Vorgehenweisen aus dem Statechart genauer betrachtet und fachlich anhand eines Textes und dem zugehörigen Aktivitätsdiagramm erläutert.

1.2.1 Spielerbewegung

Die Spielerbewegung startet mit der Eingabe des Spielers durch das betätigen einer der Tasten w, a, s oder d. Je nach Eingabe vom Spieler werden die entsprechenden Daten, in dem Fall Koordinatenwechsel und Richtungswechsel, an den Server übergeben.

Danach wird auch im Client die Koordinate geändert, um die Map und die neue Position des Avatar zu zeichnen.

Zuletzt wird der Avatar des Spielers gezeichnet, welcher sich immer in der Mitte des Bildschirm befindet. Dafür werden vom Server die Koordinate und die Richtung, in die der Spieler guckt, genommen. Hierbei unterscheidet man zwei Fälle.

In Fall eins wird die Bewegungsanimation gezeichnet. Dies geschieht nur, wenn sich der Avatar bewegt.

In Fall zwei wird die Atembewegung gezeichnet. Dies tritt nur auf wenn der Spieler sich nicht bewegt, welches uninteressant für die Spielerbewegung ist.

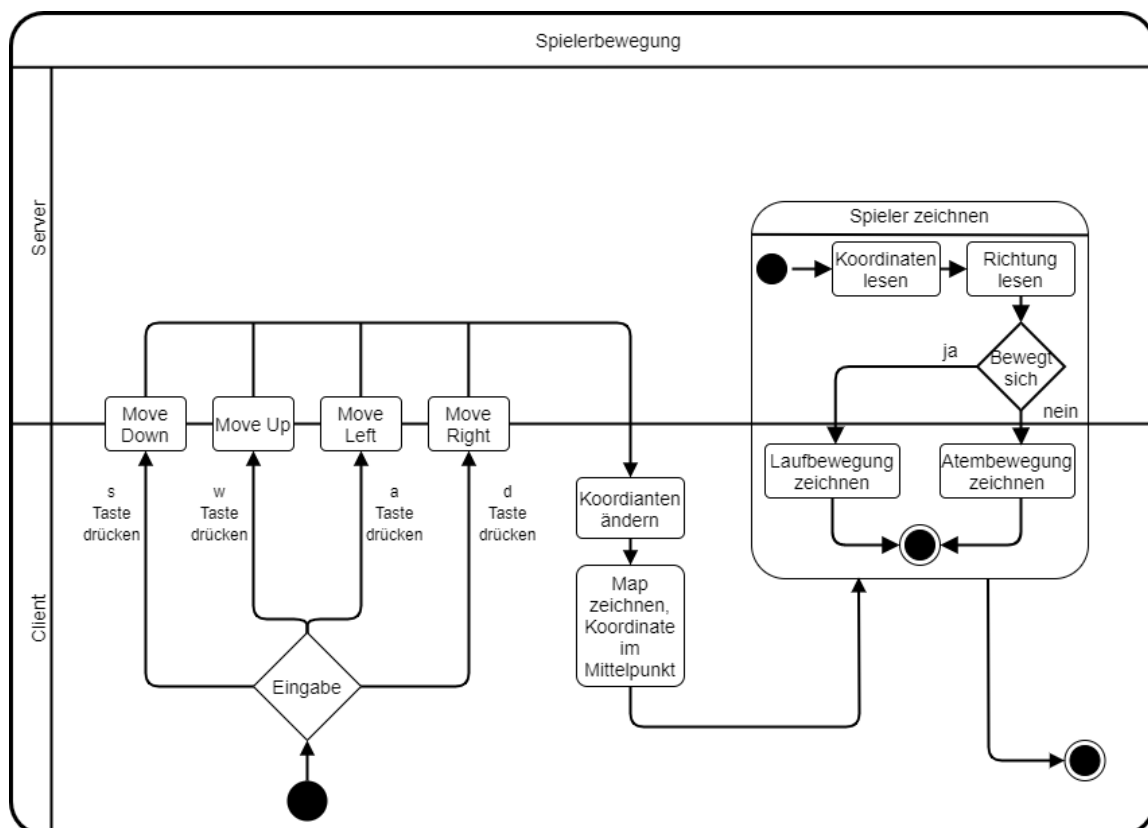


Abbildung 1.2: Spieler bewegen

1.2.2 Videochat

Beim Betreten des Servers werden alle Spieler mit dem selben Videochat verbunden. Audio und Video der Spieler, die nicht in der Nähe sind, sind deaktiviert.

Um andere Spieler anzeigen zu lassen, werden Spieler in Reichweite ermittelt. Befinden diese sich in der gegebenen Reichweite, so wird sowohl Audio als auch Video aktiviert, sodass diese Spieler miteinander kommunizieren können.

Bewegt sich jedoch einer dieser Spieler wieder aus der gegebenen Reichweite, so werden Audio und Video wieder deaktiviert.

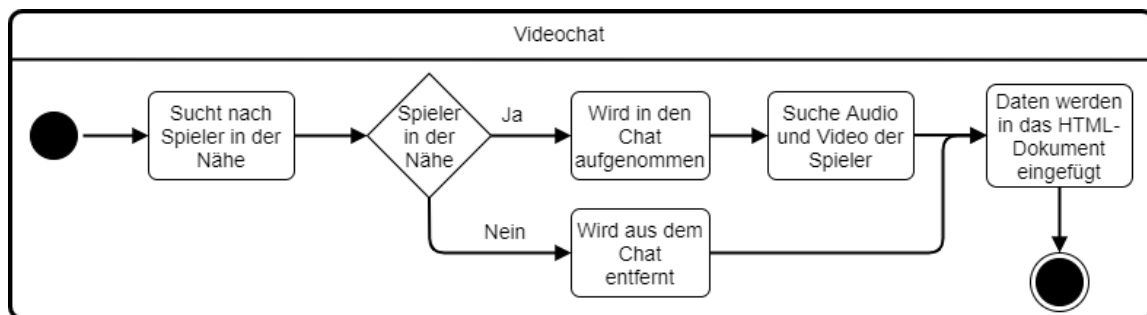


Abbildung 1.3: Videochat

2 Analyse der Produktfunktionen

Dieses Kapitel befasst sich mit der Analyse und graphischen Darstellung der Produktfunktionen aus dem Pflichtenheft. Hierbei werden Sequenzdiagramme als Graphiken verwendet. Diese sollen beteiligte Komponenten, sowie deren Interaktionen und Abhängigkeiten von relevanten Funktionalitäten darstellen.

2.1 Analyse von Funktionalität F10: Anwendungsaufruf

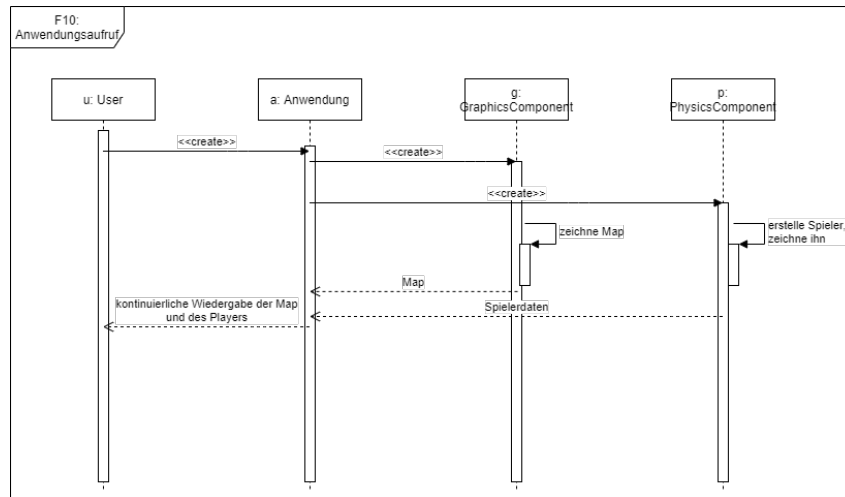


Abbildung 2.1: Anwendungsaufruf

Die Funktion <F10>, die in Abbildung 2.1 mit einem Sequenzdiagramm beschrieben wird, beschreibt das Starten der Anwendung. Durch das Aufrufen des Links wird die Anwendung gestartet. Diese lässt die Karte von der GraphicsComponent zeichnen und den Spieler von der PhysicsComponent erstellen und zeichnen. Diese Zeichnungen werden im Browser dargestellt und so dem Nutzer indirekt wiedergegeben.

2.2 Analyse von Funktionalität F20: Kartenzeichnung

Die Funktion <F20>, die in Abbildung 2.2 mit einem Sequenzdiagramm beschrieben wird, beschreibt die Kartenzeichnung. Dabei gibt der User der Anwendung (durch eine Bewegung von sich selbst oder jemand anderen ausgelöst) den Impuls, die Map erneut zu zeichnen. Die Anwendung gibt diesen Impuls an die GraphicsComponent und die PhysicsComponent weiter, welche die Map und die andern Spieler darstellen. Auch hier wird alles wieder im Browser dargestellt.

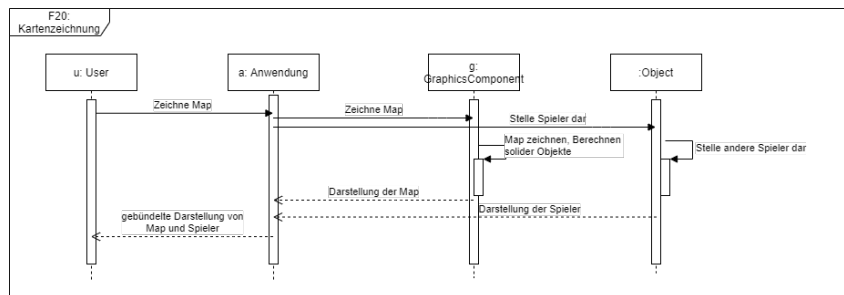


Abbildung 2.2: Kartenzeichnung

2.3 Analyse von Funktionalität F30: Spielerbewegung

Die Funktion <F30>, die in Abbildung 2.3 mit einem Sequenzdiagramm beschrieben wird, beschreibt die Spielerbewegung. Der Nutzer nutzt die Tasten W, A, S und D, um den Character zu bewegen. Die Anwendung nimmt dies wahr und leitet die Informationen über den Tastendruck an die PhysicsComponent weiter. Diese berechnet die neue Spielerposition, startet eine Laufanimation und zeichnet diese auf ein Canvas. Die berechnete Spielerposition bekommt die Anwendung wieder und lässt damit die GraphicsComponent einen neuen Ausschnitt der Map zeichnen. Am Ende wird alles wieder im Browser dargestellt und so dem User wiedergegeben.

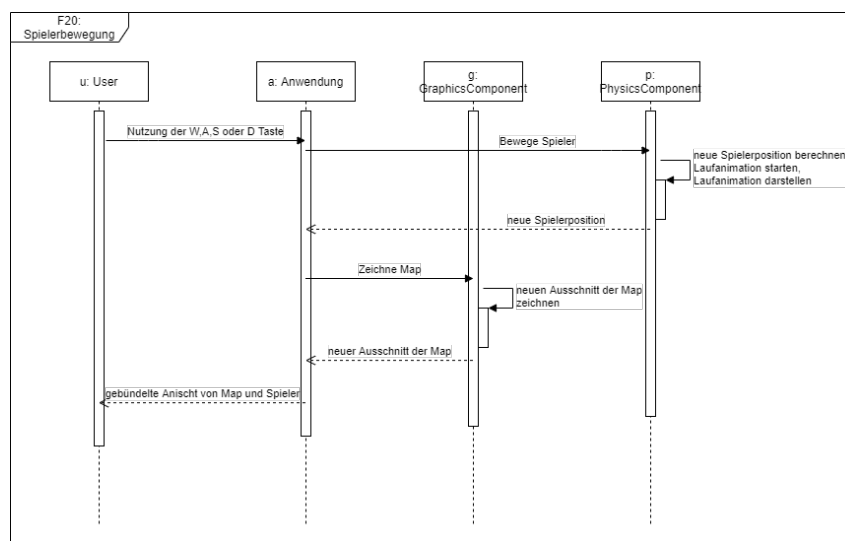


Abbildung 2.3: Spielerbewegung

2.4 Analyse von Funktionalität F40: Start Videokonferenz

Die Funktion <F40>, die in Abbildung 2.4 mit einem Sequenzdiagramm beschrieben wird, beschreibt das Starten einer Videokonferenz. Die Anwendung übermittelt kontinuierlich Audio und Video an den Jitsi-Server und dieser übermittelt kontinuierlich Audio und Video der anderen User. Der OfficeMania-Server stellt die Positionen der anderen Spieler zur Verfügung. Die Anwendung berechnet so den Abstand zu den anderen. Ist der Abstand klein genug, werden dem User Video und Audio der entsprechenden anderen User eingeblendet.

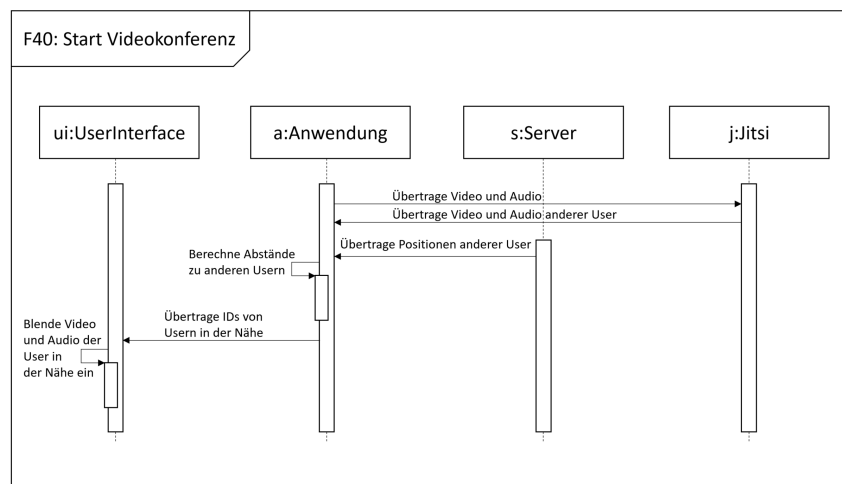


Abbildung 2.4: Start Videokonferenz

2.5 Analyse von Funktionalität F50: Stop Videokonferenz

Die Funktion <F50>, die in Abbildung 2.8 mit einem Sequenzdiagramm beschrieben wird, beschreibt das Beenden einer Videokonferenz. Die Anwendung übermittelt kontinuierlich Audio und Video an den Jitsi-Server und dieser übermittelt kontinuierlich Audio und Video der anderen User. Der OfficeMania-Server stellt die Positionen der anderen Spieler zur Verfügung. Die Anwendung berechnet so den Abstand zu den anderen. Wird der Abstand zu groß, werden Video und Audio der entsprechenden anderen User ausgeblendet.

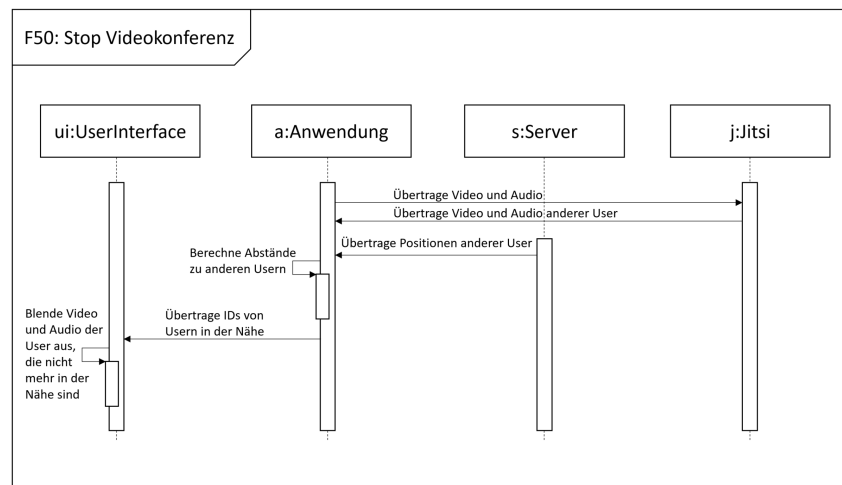


Abbildung 2.5: Stop Videokonferenz

2.6 Analyse von Funktionalität F60: Anzeige aktueller Spielerzahl

Die Funktion <F60>, die in Abbildung 2.6 mit einem Sequenzdiagramm beschrieben wird, beschreibt die Anzeige der aktuellen Spielerzahl. Wenn ein Nutzer die Anwendung startet, wird der Server durch die Anwendung von der Spielerkreation informiert. Anschließend speichert die Anwendung den Spieler in der Spielerliste. Bei Beendigung der Anwendung, wird der Server wieder informiert, und der Spieler wird aus der Liste gelöscht. Bei jedem Anwendungsstart und -ende wird die aktuelle Spieleranzahl anhand der eingetragenen Spieler ermittelt und ausgegeben.

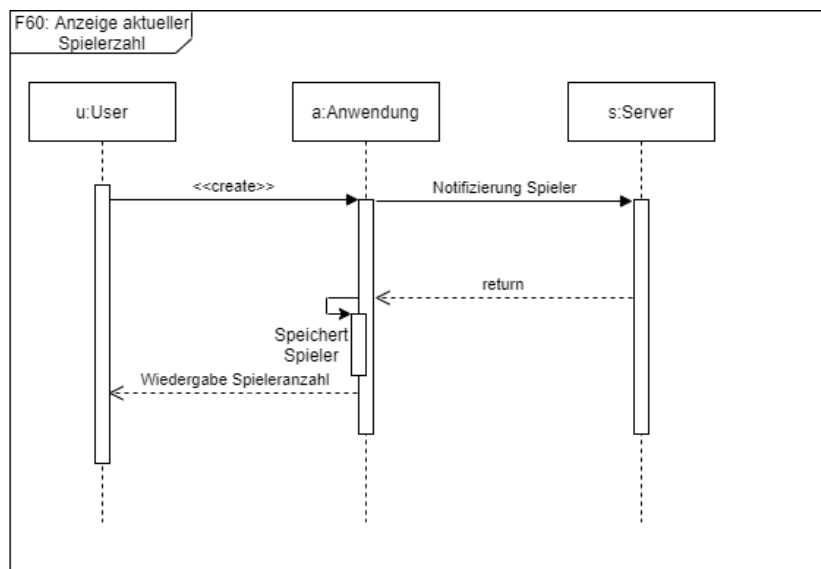


Abbildung 2.6: Anzeige aktueller Spielerzahl

2.7 Analyse von Funktionalität F70: Namensbestimmung

Die Funktion <F70>, die in Abbildung 2.7 mit einem Sequenzdiagramm beschrieben wird, beschreibt die Namensbestimmung. Wenn der Nutzer die Option zur Namensbestimmung ausführt, wird von der Anwendung ein Textfeld für den einzugebenden Namen geöffnet. Nach Eingabe und Bestätigung des gewünschten Namens, wird dieser anschließend im Server für den Spieler gespeichert. Die PhysicsComponent der Anwendung hat Zugriff auf den neuen Namen. Der neue Name wird fortan vom UserInterface bei allen Nutzern wiedergegeben.

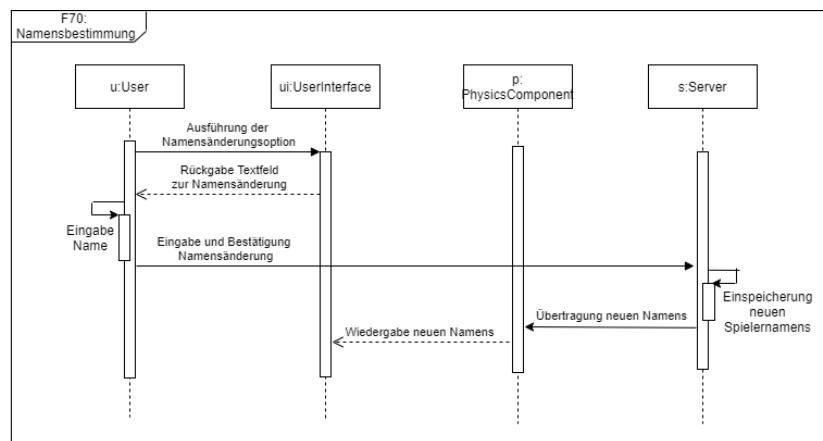


Abbildung 2.7: Namensbestimmung

2.8 Analyse von Funktionalität F80: Spielerinteraktion

Die Funktion <F80>, die in Abbildung 2.8 mit einem Sequenzdiagramm beschrieben wird, beschreibt die Spielerinteraktion. Wenn der Spieler sich an einem interagierbaren Objekt befindet und die Option zur Interaktion ausführt, startet die Spielerinteraktion. Hierbei führt die Anwendung bei dem relevanten Objekt im Server eine Statusänderung durch. Je nach Objekt und Interaktion, sowie der folgenden Serveranpassung, führt die PhysicsComponent die Interaktion durch, welche visuell vom UserInterface dargestellt wird.

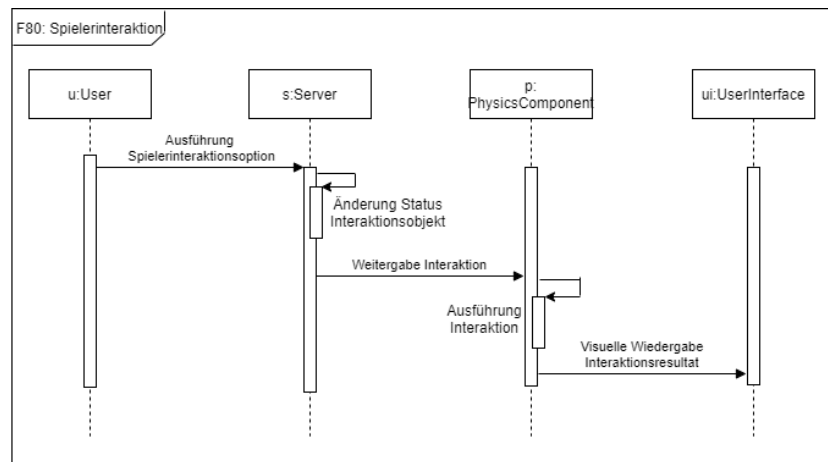


Abbildung 2.8: Spielerinteraktion

3 Datenmodell

Die Anwendung benötigt keine dauerhaft gespeicherten Daten. Das bedeutet, dass beim Schließen des Servers keine Daten bis zum nächsten Start aufbewahrt werden. Es werden temporäre Daten im Server und in Browser-Cookies gespeichert, wie Spielernamen, deren Position und Objektplatzierungen.

4 Konfiguration

Als grundlegende Konfiguration reicht ein Server aus, auf dem eine aktuelle Version der Docker Engine läuft, die die amd64 Architektur und Linux Docker Container unterstützt.

Das Projekt ist deshalb ziemlich plattformunabhängig, was die Konfiguration des Servers angeht, da die Konfiguration des Projekts innerhalb des Docker Images und der Umgebungsvariablen beim Starten vorgenommen werden kann.

Somit wird als explizite Konfigurationsdatei auf dem Server eine docker-compose.yml benötigt, die das Image des Projekts und die gewünschte Konfiguration des Kunden enthält.

Jedoch ist eine weitere explizite Nennung bestimmter Konfigurationsdateien für einen Server nicht ohne weiteres möglich, da weitere Dinge wie bestimmte Portfreigaben, FirewallEinstellungen etc. abhängig vom jeweiligen Betriebssystem und verwendeter Software auf einem Server ist.

5 Glossar

Avatar: Ein Avatar ist ein virtueller Charakter in einer virtuellen Welt.

Canvas: Ein Canvas ist ein HTML Element, auf welchem sich graphische Elemente darstellen lassen.

Client: Clients sind Programme oder Computer, die auf Server zugreifen können und von denen Daten oder Dienste abrufen.

Cookies: Cookies sind kleine Textdateien, die auf dem Rechner gespeichert werden, was beim Surfen im Internet von einer Webseite über dem Webbrowser passiert. Wird die Seite noch mal besucht, wird der Cookie wieder aufgerufen.

DockerEngine: Docker Engine ist eine Software, welche eine Schnittstelle zwischen den Ressourcen des Hosts und den laufenden Containern bildet.

Jitsi: Ist eine freie Open-Source-Videoconferencing-Software.

Statechart: Statechart beschreibt das Antwortverhalten eines Objekts.