

# Unity-Multiplayer

## Kurzbeschreibung

In diesem Dokument werden die einzelnen Komponenten des Projekts *Unity-Multiplayer* beschrieben. Bei dem Projekt handelt es sich um in VR-Projekt, in welchem man geplante Bauprojekte (Häuser und deren Umgebungen) gemeinsam in einer virtuellen Welt besichtigen kann. Dabei ist es möglich über die Anwendung zu kommunizieren und mit anderen Nutzern in der Welt zu interagieren. Die geplanten Bauprojekte können mit Hilfe eines Web-Interfaces in die Welt eingefügt werden.

Unity-Multiplayer .....	1
Kurzbeschreibung .....	1
Quickstart Guide .....	2
Webinterface .....	4
Sequenzdiagramme .....	5
Clients Interaktion .....	7
Modell-Platzierung .....	8
Modelldaten erhalten .....	9
IFC Modelle kombinieren .....	9
OpenStreetMap-Modelle .....	9
Dynamische Sonne .....	10

## Quickstart Guide

Um das Projekt in VR begehen zu können, wird SteamVR auf dem auszuführenden Computer benötigt. Dies ermöglicht die Verwendung von einer größeren Anzahl an VR-Headsets von verschiedenen Anbietern.

Nach Installation des Steam-Client (hier: <https://store.steampowered.com/about/>) und Anmeldung mit einem Account kann SteamVR über den Steam-Store heruntergeladen und installiert werden (hier: <https://store.steampowered.com/app/250820/SteamVR/>). Danach wird der Account nicht mehr benötigt. Der SteamVR Server ist nun im Hintergrund aktiv und startet SteamVR wenn eine VR-Anwendung gestartet wird. Dabei ist keine weitere Anmeldung notwendig.

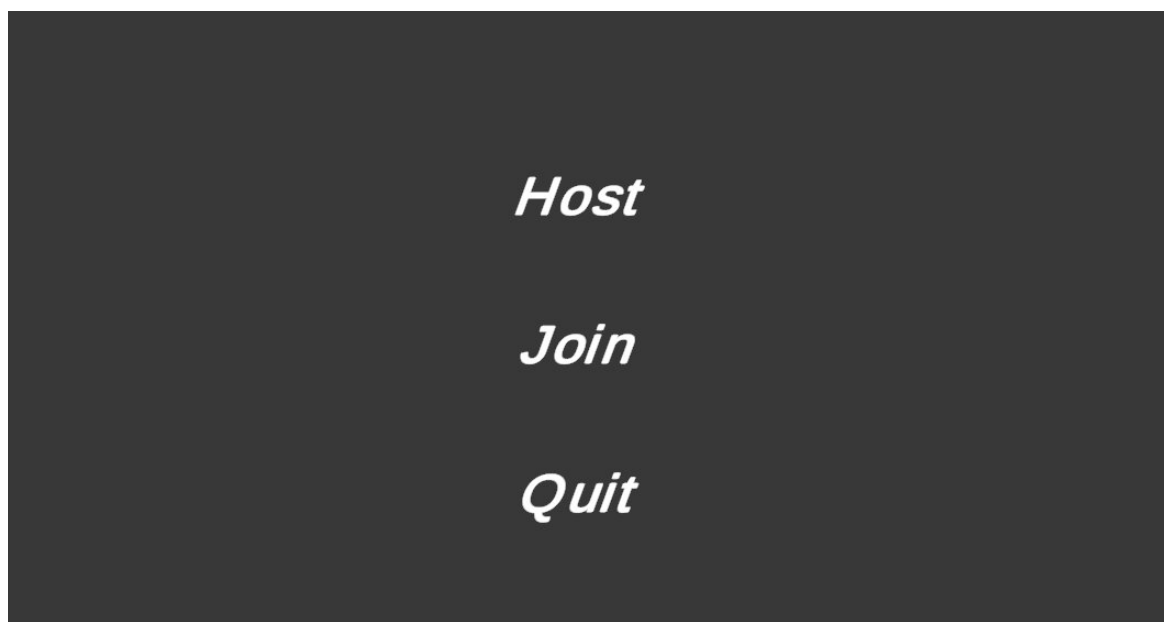
Bei Verwendung eines Headsets eines anderen Herstellers als VALVE, muss zunächst die VR-Software des Herstellers gestartet werden. Im Falle von Oculus der Oculus-Client.



Nun kann die Software durch Ausführen der Datei *Multiplayer.exe* gestartet werden.

Name	Änderungsdatum	Typ	Größe
MonoBleedingEdge	22.11.2022 11:53	Dateiordner	
Multiplayer_BurstDebugInformation_Do...	22.11.2022 11:53	Dateiordner	
Multiplayer_Data	22.11.2022 11:53	Dateiordner	
Multiplayer.exe	16.03.2022 03:18	Anwendung	639 KB
UnityCrashHandler64.exe	16.03.2022 03:25	Anwendung	1.205 KB
UnityPlayer.dll	16.03.2022 03:25	Anwendungserwe...	27.612 KB

Anschließend wird man mit folgendem Menü begrüßt. Dort muss man sich entscheiden, ob man der Host (Gastgeber) einer Sitzung sein möchte oder einer vorhandenen Sitzung über Join beitreten möchte.



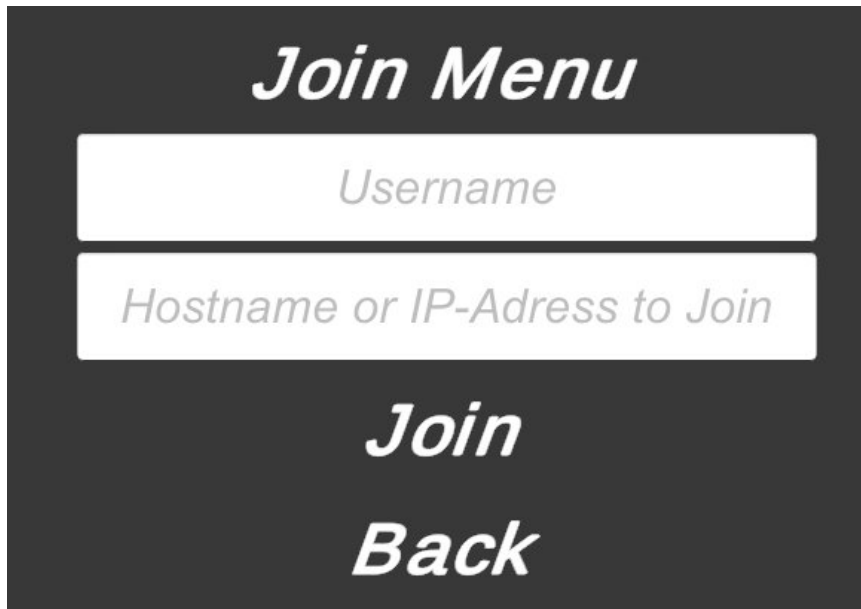
Host:

The image shows a dark-themed user interface titled "Host Menu" in a large, white, italicized font. Below the title are two white rectangular input fields. The first field contains the placeholder text "Username" in a light gray font. The second field contains the placeholder text "Own IP-Adress" in a light gray font. To the right of the second input field is a large white question mark "?". Below the input fields are two white buttons with the text "Create" and "Back" in a bold, italicized font.

Als Host müssen zwei Felder ausgefüllt werden. Bei dem Feld „Username“ kann der Name des Hosts eingetragen werden. Dieser wird dann in der Anwendung über dem Körper angezeigt. In das Feld „Own IP-Adress“ muss die IP-Adresse des Host PCs eingetragen werden um festzulegen, auf welcher IP die Sitzung bereitgestellt werden soll. Wo man diese finden kann, wird mit einem Klick auf das Fragezeichen erklärt.

Sobald die Anwendung als Host gestartet wurde, kann unter [http://\"Own IP-Adress\":8888](http://\) das Webinterface aufgerufen werden. Das Passwort lautet 123. Darin können die Modelldaten ausgetauscht werden.

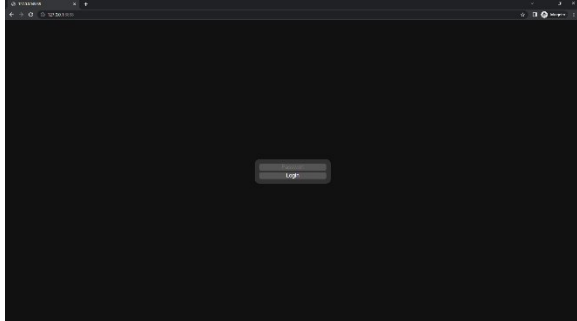
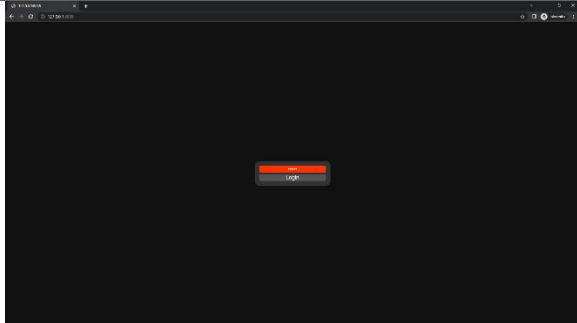
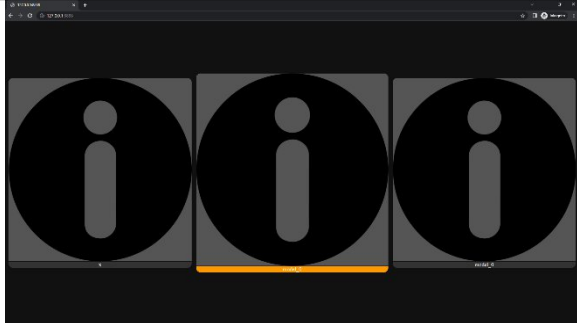
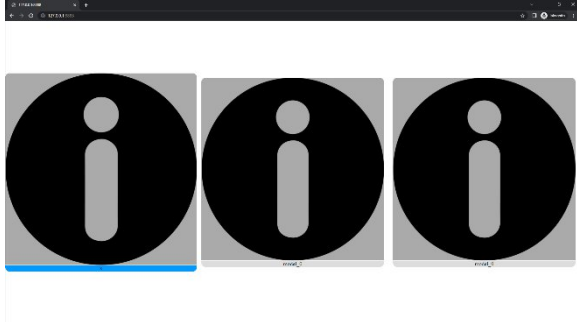
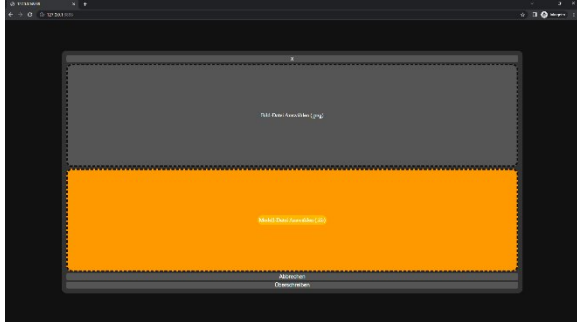
Join:

The image shows a dark-themed user interface titled "Join Menu" in a large, white, italicized font. Below the title are two white rectangular input fields. The first field contains the placeholder text "Username" in a light gray font. The second field contains the placeholder text "Hostname or IP-Adress to Join" in a light gray font. Below the input fields are two white buttons with the text "Join" and "Back" in a bold, italicized font.

Unter Join kann man, wie bei dem Host, einen „Username“ festlegen, der in der Anwendung angezeigt wird. In „Hostname or IP-Adress to Join“ kann der Hostname dank des DNS-Resolvers oder die IP-Adresse des Host-PCs eingetragen werden. Mit „Join“ kann dann der Sitzung beigetreten werden.

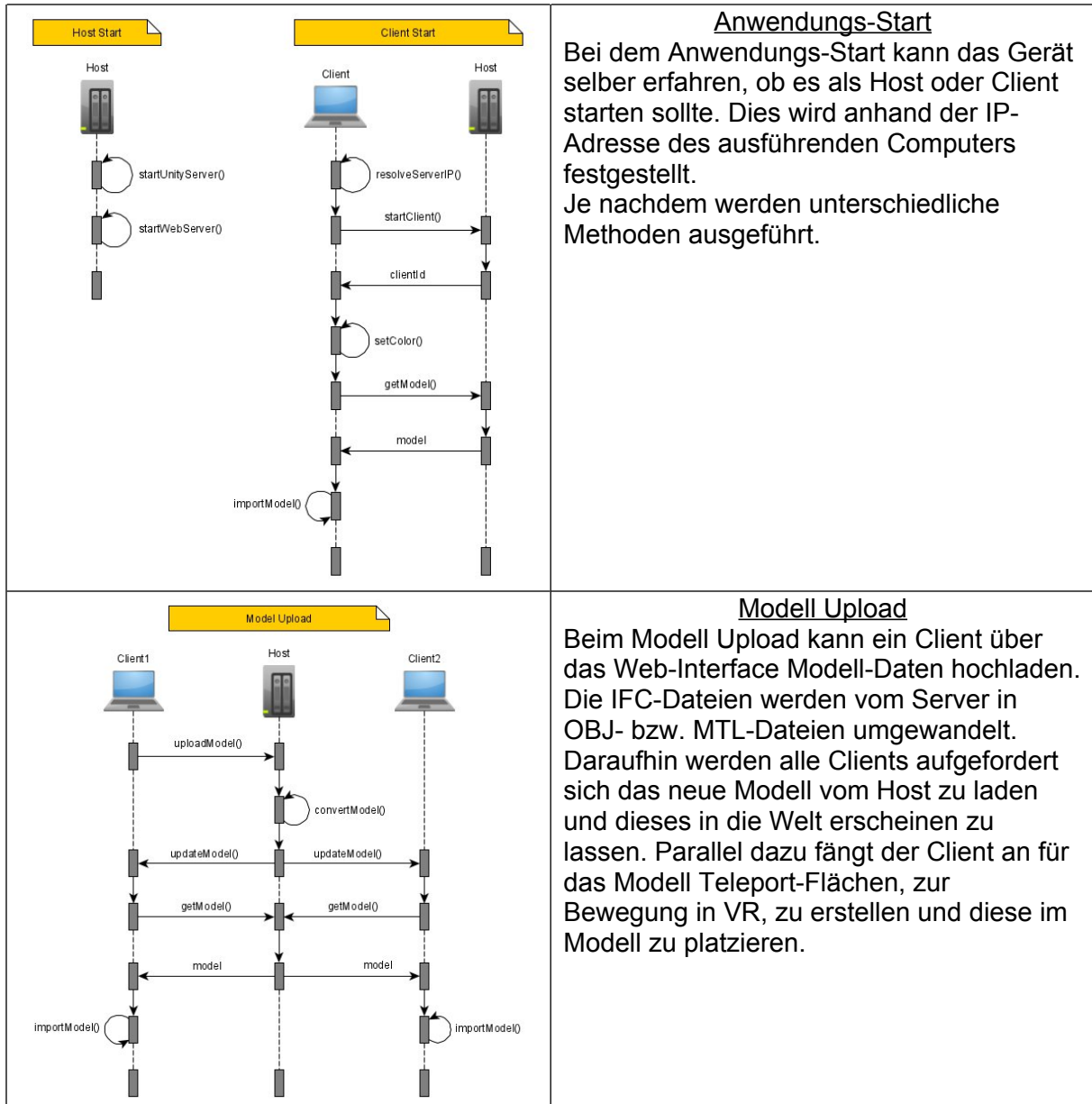
## Webinterface

Das Webinterface dient zur Verwaltung der Bauwerke und Pläne, welche in der Welt dargestellt werden sollen. Dieses wird von dem Host-Computer bereitgestellt und ist mit einem Passwort vor unbefugter Nutzung geschützt.

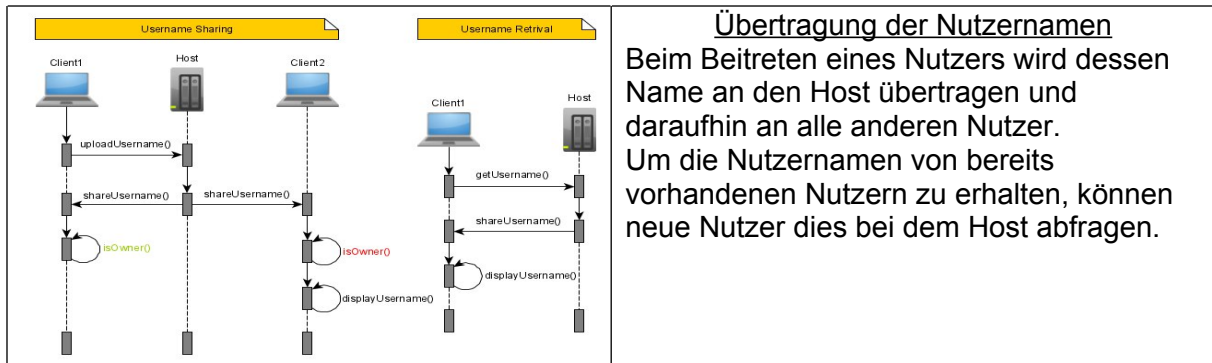
	<p><u>Login</u></p> <p>Das Web-Interface bietet einen Login an, welcher mit einem Passwort abgesichert ist.</p>
	<p><u>Falsche Passwort</u></p> <p>Bei falschem Passwort wird der Nutzer durch Rot-Werden des Textfeldes informiert.</p>
	<p><u>Dunkles Thema</u></p> <p>Das Webinterface hat ein dunkles Thema mit der Kontrastfarbe orange.</p>
	<p><u>Helles Thema</u></p> <p>Das Webinterface hat ein helles Thema mit der Kontrastfarbe blau.</p>
	<p><u>Bearbeitungsmodus</u></p> <p>Im Bearbeitungsmodus können der Titel, der auszustellende Plan und das Modell, welches Präsentiert werden soll, ausgewählt werden.</p> <p>Der Plan muss als png-Datei vorliegen. Das Modell muss als ifc-Datei vorliegen.</p>

## Sequenzdiagramme

Hier werden einige Sequenzdiagramme zum besseren Verständnis der Anwendung dargestellt.

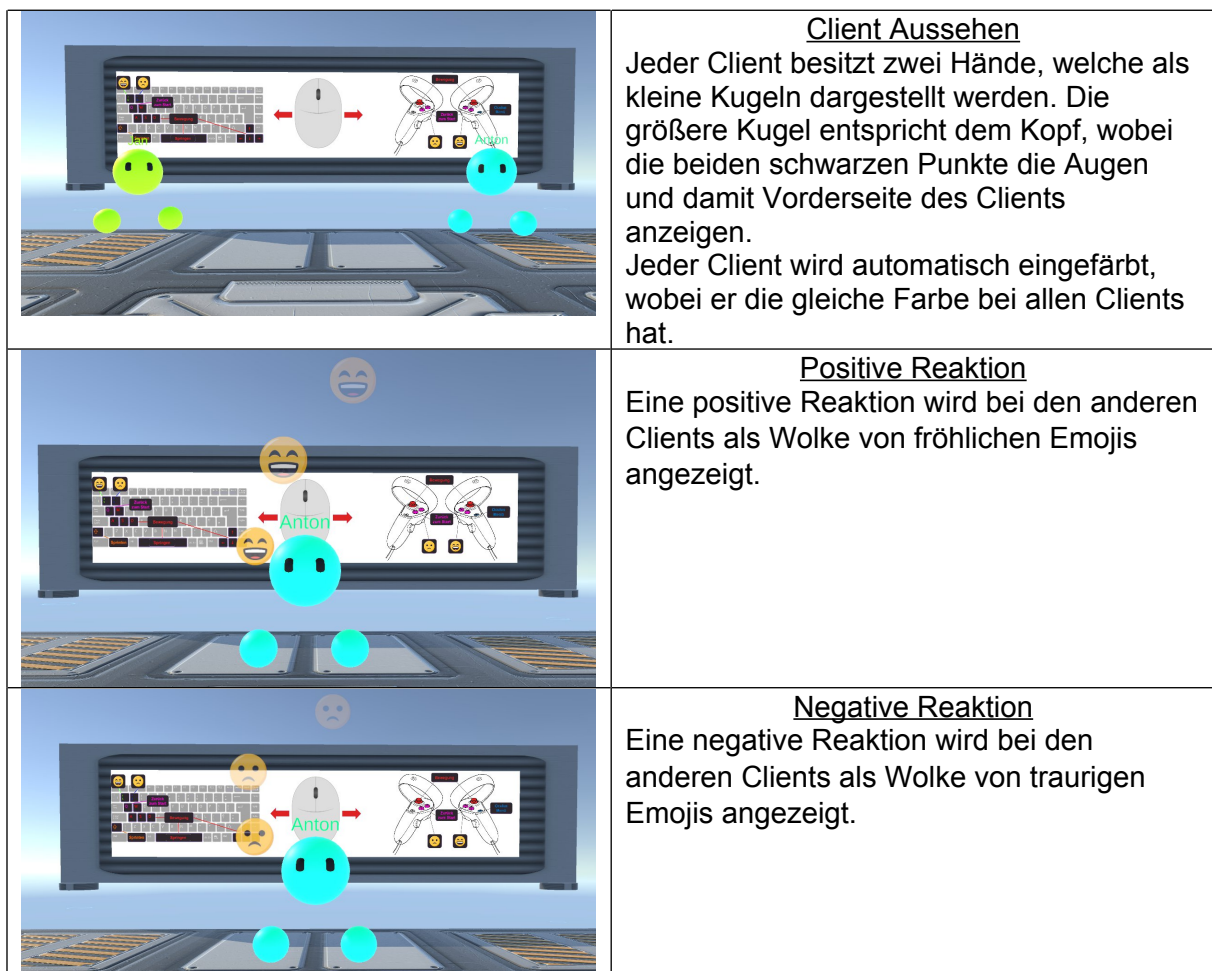


<p style="text-align: center;"><b>Client Syncing</b></p> <pre> sequenceDiagram     participant C1 as Client1     participant H as Host     participant C2 as Client2      C1-&gt;&gt;H: uploadMovement()     H-&gt;&gt;C2: shareMovement()     H-&gt;&gt;C1: shareMovement()     C1-&gt;&gt;C1: isOwner()     C2-&gt;&gt;C2: isOwner()     C2-&gt;&gt;C2: interpolatePosition()     C2-&gt;&gt;C2: updateClientPosition()     </pre>	<p style="text-align: center;"><u><b>Übertragung der Position</b></u></p> <p>Beim Client Syncing überträgt ein Client seine Bewegungen an den Host. Dieser verteilt anschließend die Bewegungsdaten an alle Clients, die mit ihm verbunden sind. Jeder Client, der die Bewegungsdaten erhält, prüft ob die empfangenen Bewegungsdaten seine eigenen sind, und updatet diese gegeben falls entsprechend.</p>
<p style="text-align: center;"><b>Voice Chat</b></p> <pre> sequenceDiagram     participant C1 as Client1     participant H as Host     participant C2 as Client2      C1-&gt;&gt;C1: recordAudio()     C1-&gt;&gt;C1: isRelevant()     C1-&gt;&gt;C1: downSampleAudio()     C1-&gt;&gt;H: uploadAudio()     H-&gt;&gt;C1: shareAudio()     H-&gt;&gt;C2: shareAudio()     C1-&gt;&gt;C1: isOwner()     C2-&gt;&gt;C2: isOwner()     C2-&gt;&gt;C2: playAudio()     </pre>	<p style="text-align: center;"><u><b>Übertragung des Sprachchat</b></u></p> <p>Beim Sprachchat wird kontinuierlich Audio aufgenommen. Dabei wird über mit Hilfe der maximalen Amplitude überprüft, ob das Audio relevant ist. Wenn das der Fall ist, wird die Qualität des Audios heruntergesetzt, um Bandbreite beim Übertragen zu sparen. Abgesehen vom Client, welcher das Audio gesendet hat, können alle die Nachricht hören.</p>
<p style="text-align: center;"><b>Reaction</b></p> <pre> sequenceDiagram     participant C1 as Client1     participant H as Host     participant C2 as Client2      C1-&gt;&gt;H: uploadReaction()     H-&gt;&gt;C1: shareReaction()     H-&gt;&gt;C2: shareReaction()     C1-&gt;&gt;C1: isOwner()     C2-&gt;&gt;C2: isOwner()     C2-&gt;&gt;C2: playReaction()     </pre>	<p style="text-align: center;"><u><b>Übertragung der Reaktionen</b></u></p> <p>Wenn ein Client eine Reaktion teilen möchte, wird eine Nachricht an den Host geschickt. Dieser leitet die Nachricht an alle Clients weiter. Abgesehen vom Client, welcher die Reaktion getätigt hat, können alle die Reaktion sehen.</p>

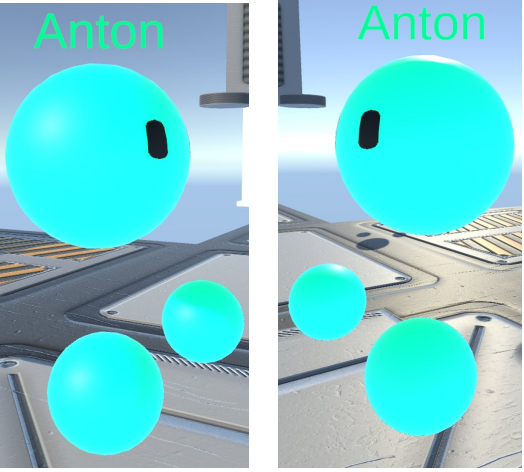
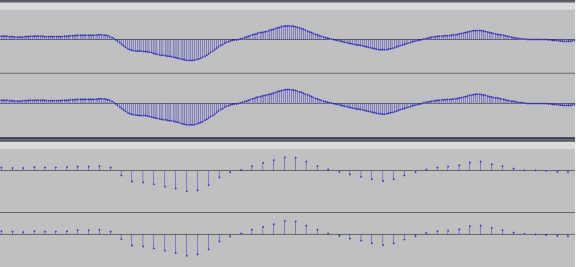


## Clients Interaktion

Nutzer können sich gegenseitig in der Welt sehen und miteinander sprechen und durch einen Tastendruck Reaktionen ausführen, welche vom Gegenüber gesehen werden können. Die Sprache eines Nutzers wird zu allen anderen Nutzern übertragen. Dabei wird die Lautstärke der Sprache an die Distanz des Gegenübers angepasst. Das bedeutet, umso weiter eine Person von einer anderen entfernt steht, umso leiser ist diese zu hören, bis sie komplett verstummt.

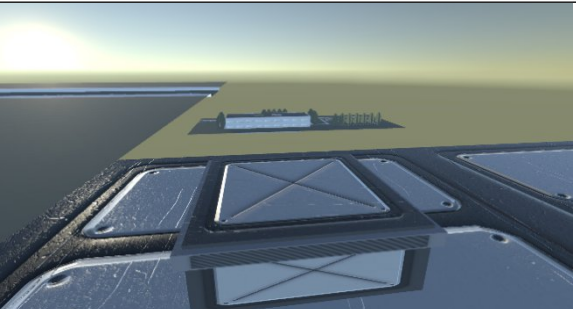
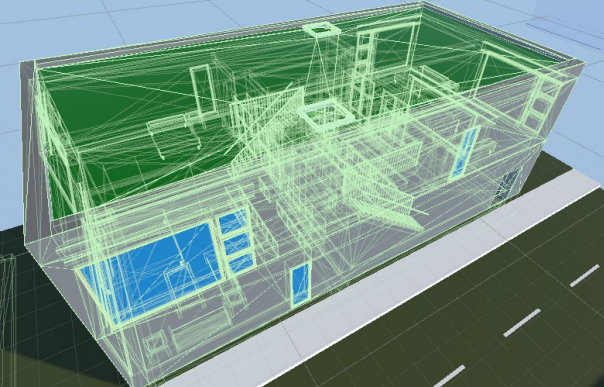




	<p><u>Name eines Nutzers</u></p> <p>Der Name eines Nutzers wird in der Anwendung angezeigt und richtet sich immer für dich Sicht des Betrachters aus. Dies ermöglicht immer die beste Sicht auf den Namen des Nutzers.</p>
	<p><u>Sprachchat Resample</u></p> <p>Das Audio des Sprachchats wird durch ein downsample komprimiert. <i>(erstellt mit Audacity)</i></p>

## Modell-Platzierung

Die Modelle werden auf zwei Arten präsentiert. Einerseits auf einem kleinen Podium als Miniaturversion und andererseits in normaler Größe, welche besichtigt werden kann.

	<p><u>Podium Modell</u></p> <p>Das Podium Modell wird zur Orientierung runterskaliert und auf einem Podest dargestellt.</p>
	<p><u>Teleport-Flächen</u></p> <p>Aus Performance Gründen wurden die Teleport-Flächen aus der vorherigen Version entfernt. Stattdessen wird das Modell doppelt in die Welt geladen. Dabei wird der Klon leicht verschoben, sein Renderer wird deaktiviert und es wird ein Skript angefügt, welches das Teleportieren ermöglicht.</p>



## Modelldaten erhalten IFC-Repository

Es gibt mehrere Möglichkeiten kompatible Modelldaten zu erhalten. Für einfache Tests können ifc-Dateien auf folgender Webseite heruntergeladen werden:

<http://smartlab1.elis.ugent.be:8889/IFC-repo/>

## IFC Modelle kombinieren

Modelle die im IFC-Format vorliegen zu kombinieren ist nicht so trivial, wie es im ersten Moment erscheint. Dies liegt daran, dass IFC ein Format ist, welches die baulichen Eigenschaften eines Objekts beschreiben soll. Dies beinhaltet auch ein 3D Modell des Objekts. Will man also beispielsweise zwei IFC Dateien kombinieren, muss nicht nur das 3D Objekt kombiniert werden, sondern auch alle anderen Eigenschaften der beiden Dateien.

Um dies nicht machen zu müssen, gibt es auch die Möglichkeit die Modelle die in den IFC Dateien sind zu exportieren und diese dann als ein neues Modell zu kombinieren und als IFC Projekt zu speichern.

Dafür müssen die Modelle mit Hilfe des IFC-Converters, der in unserem Projekt verwendet wird, zu eine „.obj und .mtl“ Datei konvertiert werden. Diese können im Anschluss dann in ein 3D-Programm, zum Beispiel Blender, importiert werden. Dort können diese wie gewünscht angeordnet werden. Wenn man mit seinem neuen Modell zufrieden ist, muss dieses als „.dae“ exportiert werden. Dieses kann anschließend in das Programm „[CAD Exchanger](#)“ importiert werden und im Anschluss als .ifc exportiert werden. Diese fertige IFC-Datei kann dann einfach über das Web-Interface des Unity-Multiplayers an die gewünschte Stelle platziert werden.

## OpenStreetMap-Modelle

Für reale Modelldaten kann das kostenfreie Blender-Plugin „Blender-OSM“ verwendet werden. Dieses Plugin kann Daten von OpenStreetMap herunterladen und zu einem dreidimensionalen Modell konvertieren. Es ist unter folgender Webseite verfügbar:

<https://prochitecture.gumroad.com/l/blender-osm>

Das Problem hierbei ist, dass Blender das Modell nicht als ifc-Datei exportieren kann. Eine Lösung für dieses Problem besteht darin, das Modell als fbx-Datei zu exportieren und erneut zu importieren um alle Modifier, welche das Blender-OSM-Plugin erstellt hat zu entfernen.

Nun kann das Modell als dae-Datei exportiert und in dem Programm FreeCAD importiert werden. FreeCAD kann aus den einzelnen Meshes „Bauteile“ erstellen. Hierfür wird das Mesh ausgewählt und unter „*Part (Workbench) > Part > Part from mesh*“ umgewandelt. Das Mesh kann danach gelöscht werden.

Nachdem alle Meshes konvertiert wurden, können alle „Bauteile“ ausgewählt und gemeinsam als ifc-Datei exportiert werden. Dieses Modell kann in die Anwendung importiert werden, jedoch werden alle Farben durch die Standard Farbe ausgetauscht. Durch Bearbeitung der ifc-Datei von Hand kann mit sehr viel Trial and Error die Farbe wieder hinzugefügt werden.

In dem Repository ist unter „*doc/models/Rheinfelden.ifc*“ das resultierende Modell, welches ungefähr einen Quadratkilometer von Rheinfelden abdeckt. Dort befinden sich auch andere fertige Modelle, die verwendet werden können.

## Dynamische Sonne

Die Sonne und somit das Licht in der Welt werden abhängig von der Systemzeit des ausführenden Computers dargestellt. Somit ist es möglich die Bauwerke an verschiedenen Tageszeiten zu besichtigen beziehungsweise darzustellen.

	<p>Sonnenstand Der Sonnenstand um 10 Uhr Systemzeit.</p>
	<p>Sonnenstand Der Sonnenstand um 13 Uhr Systemzeit.</p>