

## **Abstract**

Cafés are the home to many interesting ideas, it also a place where people go to meet new interesting people. In this report it is discussed how a café environment can be realized online. The problem faced during the project includes overseeing, how one can look around and see other groups of people talking to each other. Overhearing, how one can listen to and find inspiration from other conversations going on in the café. Mingle, how one can move between these conversations in a subtle way.

A prototype was built using WebRTC, HTML5 and JavaScript in order to solve these problems. Multiple solutions for each problem, together with 2- and 3-dimensional views are being presented. The prototype consists of a video chat with multiple extra features for collaboration, like shared napkin to paint on and synchronized YouTube watching, implemented to enhance the feeling of sitting at the same table in a café-like environment online.

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Problem description . . . . .	4
1.2	Project goals . . . . .	5
1.2.1	Overseeing . . . . .	5
1.2.2	Overhearing . . . . .	5
1.2.3	Mingle . . . . .	5
1.3	Project requirements . . . . .	6
1.4	Project delimitations . . . . .	6
1.5	Research questions . . . . .	6
1.6	Project transparency . . . . .	7
<b>2</b>	<b>Related work</b>	<b>10</b>
2.1	Traditional video conference . . . . .	10
2.2	3D Virtual environments . . . . .	10
2.3	Mixed reality . . . . .	10
2.4	Criteria . . . . .	11
<b>3</b>	<b>Related services</b>	<b>12</b>
3.1	Google WebRTC . . . . .	12
3.2	Browser meeting . . . . .	12
3.3	frisB . . . . .	12
3.4	Video Conference in 3D-environment . . . . .	12
3.5	Protosphere . . . . .	12
3.6	O.L.I.V.E . . . . .	13
3.7	OpenQwaq . . . . .	13
3.8	Paltalk . . . . .	13
3.9	Tinychat . . . . .	13
3.10	Skype . . . . .	13
3.11	Google Hangout . . . . .	14
3.12	ooVoo . . . . .	14
3.13	Second Life . . . . .	14
3.14	The Word Cafe online community . . . . .	14
<b>4</b>	<b>Related papers</b>	<b>15</b>

<b>5</b>	<b>What the site looks like</b>	<b>17</b>
5.1	Index/start page . . . . .	17
5.2	Caféview . . . . .	19
5.3	Tableview . . . . .	20
5.4	Use cases . . . . .	21
5.4.1	Use case 1: Enter a café . . . . .	21
5.4.2	Use case 2: Oversee and overhear . . . . .	21
5.4.3	Use case 3: Sit down at table . . . . .	21
<b>6</b>	<b>Technologies</b>	<b>22</b>
6.1	Node.js . . . . .	22
6.2	WebRTC . . . . .	22
6.3	Licode . . . . .	22
6.3.1	Erizo . . . . .	22
6.3.2	Nuve . . . . .	22
6.4	MongoDB . . . . .	23
6.5	Mongoose . . . . .	23
6.6	Express . . . . .	23
6.7	Twitter Bootstrap . . . . .	23
6.8	HTML5 . . . . .	23
6.9	Three.js . . . . .	23
6.9.1	WebGL . . . . .	24
6.10	GitHub . . . . .	24
<b>7</b>	<b>Method</b>	<b>25</b>
7.1	WebRTC comparison . . . . .	25
7.1.1	Easy RTC . . . . .	25
7.1.2	WebRTC.io . . . . .	25
7.1.3	Holla . . . . .	25
7.2	Survey . . . . .	26
7.2.1	Evaluation . . . . .	26
7.3	Prestudy conclusion . . . . .	26
7.4	System architecture . . . . .	27
7.4.1	Backend . . . . .	27
7.4.2	Database . . . . .	28
7.4.3	Server . . . . .	28
7.4.4	Nuve . . . . .	28
7.4.5	Erizo Controller . . . . .	28

7.4.6	Frontend . . . . .	29
7.4.7	Erizo Client . . . . .	29
7.5	Setting up a connection . . . . .	29
<b>8</b>	<b>Result</b>	<b>30</b>
8.1	Design . . . . .	30
8.1.1	The 2D-view . . . . .	30
8.1.2	The 3D-view . . . . .	30
8.2	Leader . . . . .	31
8.3	Peer-to-peer . . . . .	31
8.4	Overhearing . . . . .	32
8.5	Overseeing . . . . .	33
8.6	Mingle . . . . .	34
8.7	Paint . . . . .	35
8.8	YouTube . . . . .	36
8.9	Text chat . . . . .	37
8.10	Server performance . . . . .	37
8.10.1	PeerConnection vs MCU . . . . .	38
8.10.2	Performance test . . . . .	40
<b>9</b>	<b>Discussion</b>	<b>42</b>
9.1	Issues . . . . .	42
9.2	The design . . . . .	42
9.3	User test . . . . .	42
9.4	Privacy Issues . . . . .	43
9.5	Performance test . . . . .	44
<b>10</b>	<b>Conclusion</b>	<b>45</b>
<b>11</b>	<b>Future work and improvements</b>	<b>46</b>
<b>Appendices</b>		<b>47</b>
<b>A</b>	<b>Comparison</b>	<b>47</b>
<b>B</b>	<b>Survey</b>	<b>49</b>

# 1 Introduction

A café is a very stimulating environment to many people. It is a place where you can easily meet other interesting people. A place where you can walk by a table, overhear a conversation and oversee the whole café. Everyone has seen the group of girls sitting at a table drinking coffee and having conversations about anything, or the writer who finds inspiration for his next book from the environment and people around himself. Another scenario is all the people travelling to conferences each year. Businesses could save a lot of money if there existed a stimulating environment where one can meet people, collaborate in groups and mingle just like in the real world.

The goal is to create an online video conference system that transmits the feeling of being in a real café. With the ability of seeing each other through video conference, collaborate together on a drawing, watch YouTube videos, reading pdfs together and of course being able to communicate through a text chat with all the participants sitting down at a table. Three other important aspects of transferring this feeling is overseeing, overhearing and mingling. These parts will be investigated and tested with a prototype.

When you are walking by a table in a real café you will hear what their conversation is about and also see all the participants, this is called overhearing and overseeing. Before you can sit down at a table and join a conversation you will gently ask if it is okay without trying to disturb their conversation too much.

The prototype contains a 2D- and 3D-view where overseeing, overhearing and mingling is realized.

Tables are often mentioned in this report. A table is referred to a video conferencing room where up to six people can video chat.

## 1.1 Problem description

The problem that was set out to be solved was how to create a web based environment for sharing the feeling of being in a Paris Café including how to convey a feeling of overseeing, overhearing and mingling. Unlike traditional video chats where there is only one group of people participating, the goal is to create a video chat service with multiple rooms and multiple groups in each room. Where users can see the other groups and overhear conversations and move between the tables in an easy way.

## 1.2 Project goals

The goal of the project is to:

- Create a working prototype
- Implement some kind of overhearing
- Implement some kind of overseeing
- Implement some kind of mingle
- 2D- and 3D-view
- Evaluate the system with a user test

Implementing some kind of overhearing, overseeing and mingle is vague but here are some ideas for how it can be done.

### 1.2.1 Overseeing

How can one see the other participants at the same table while at the same time also see other groups of visitors in the same café. This could be realized with a 3D-environment with videos organized around different tables.

### 1.2.2 Overhearing

How can one hear the conversations from the other groups sitting at another table or just walking by that table. This could be realized with spatial sound in a 3D-environment or by hover the mouse over a table in a 2D-view.

### 1.2.3 Mingle

How can participants easily move between different conversations? I.e. if I oversee and overhear another interesting conversation, how can I gently move into that without too much interruption? This can be done by, for example, knocking, avatar gestures in a 3D world or writing a message to the group.

### **1.3 Project requirements**

The project had a few restrictions of what technologies to use, WebRTC for media and data communication between clients, JavaScript and HTML5 for frontend. At the moment WebRTC is still under development. These technologies are used to promote using native browsers without any plugins.

### **1.4 Project delimitations**

This is a big project and it need some limitations, it is our imagination and time that decides. It was decided to limit the prototype to have six cafés, six tables in each café and six persons at each table maximum. The reason for having six tables in each café was to keep the intimate feeling of a café. Having more than six people in each table would make the bandwidth requirement really high for the clients.

### **1.5 Research questions**

The questions to be answered are:

- Does the ability to oversee other groups of people in the same virtual room affect the experience of online video chat?
  - What is the effect?
- Does the ability to overhear other conversations in the same virtual room affect the experience of online video chat?
  - What is the effect?
- Does knowing that you and your group can be overheard by other people affect the experience of online video chat?
  - What is the effect?
- Does knowing that you and your group can be overseen by other people affect the experience of online video chat?
  - What is the effect?

Imagine a café in Paris where people come and sit down for a coffee and conversation. Around them other things will happen; they will see people, they will overhear conversations, they might meet old friends and engage in conversation with them for a short while. This can be done online with video chat, text chat, view videos and draw together for example on a napkin as you might do in a real café. Good video quality is needed and the fewer steps from the index-page to the actual conversation is better, due to the simplicity for the user and therefore easier to make the users experience as good as possible and hopefully they will come back.

As a part of the research, the prototype will be tested in both 2D- and 3D-view to discover which of them that achieves the most realistic way of presenting a café in Paris online.

## 1.6 Project transparency

One thing that our supervisor, Peter Parnes usually requests in all his courses is that the progress of the project should be periodically updated in blog posts[1]. The idea is that everyone should understand the posts and it makes it easy for us to see what problems we run into during the project.

Before this project started a Gantt-chart was made, a rough plan of our milestones of all 20 weeks. The first five weeks was dedicated to competitive intelligence, research what technologies that is suitable for our prototype and feature planning. The next twelve weeks were spent implementing the prototype, bugfix, find improvements and writing report. The last three weeks were used to fix some bugs, writing report and the final presentation.

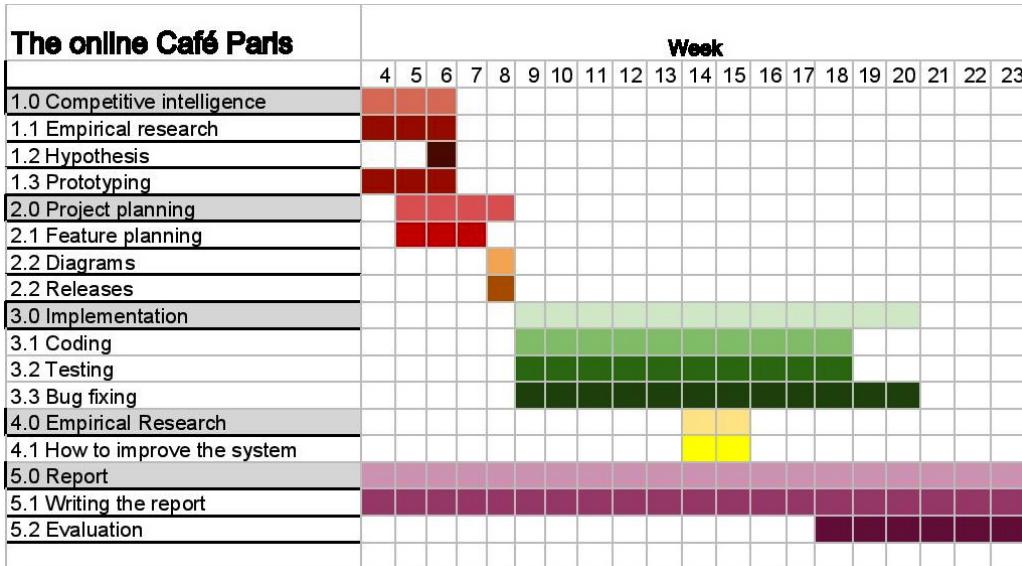


Figure 1: A Gantt-chart to guide us through the project and manage deadlines.

The work has been shared equally between the two participants in this project. All the prestudy, planning and implementing of the prototype was done working closely together.

A huge part of project was new for us, we had no prior experience with WebGL, close to no experience in design and we had only written one application in JavaScript before. Because of this, and because we are only two people in the project, we decided not to have any specific roles. All the planning and research were done together and any important decision were discussed before implementing it. We tried to split the programming equally between the two of us, because both wanted to learn everything that was new to us. We also practised pair programming which means that we both programmed on the same computer at the same time. This method works really good and we added as much functionality as if we were working separately, but with much higher quality of the code.

However, Patrik worked more on the backend, creating cafés, setting up routes, the RESTful API and modifying Licode, while Alexandra set up the database and had more focus on the design, and additional features like paint, text chat and YouTube. She also focused more on keeping track of the progress and what needed to be done in order to meet the deadlines.

All the solutions to our main problems overseeing, overhearing and mingling were solved together but Alexandra had more focus on the overseeing part and Patrik the overhearing. The mingling part were mostly solved together. We practised pair programming on the harder parts and when someone got stuck.

## **2 Related work**

Online group collaboration is an area that has been investigated and researched for many years and a number of different tools have been made available, both from research and from commercial entities. However, in a video chat, the focus has mostly been in one group chat and not to have several groups in one chat room that allows overhearing and overseeing and an easy way to move between these groups.

Video conferencing tools comes in many forms. Some are web-based like Google Hangout [2], other require software clients, like Skype[3]. Some uses avatars in a virtual environments or a mixed reality.

### **2.1 Traditional video conference**

There are tons of group video conferencing services out on the market today. In services such as Google Hangout[2] or ooVoo[4], users transmit video and audio to each other using normal video cameras or web cameras and microphones.

### **2.2 3D Virtual environments**

In 3D virtual environment you can create an avatar and walk around in a world with other avatars. Depending on the service you might be able to talk using spatial voice chat, text chat, mingle with other participants or explore the world together. An example of this is Second Life[5].

The key difference from traditional video conference is that one cannot see the faces of the other participants, you can only see their avatars. This can affect the social trust between participants but a study showed that participants get a feeling of being ‘there’ that traditional video conference does not support[6].

### **2.3 Mixed reality**

Mixed reality means merging a real world with a virtual world to produce a new world where physical and virtual objects co-exists. Usage of this mixed reality has been done in order to enable human expressions and gestures on avatars in 3D virtual environment[7].

## 2.4 Criteria

The criteria for the comparison of the different services are based on three things. What are the important aspects of a real life café, what features would be useful in an online café and the key features of the services in question.

- Video chat - If it has video chat or not.
- Free - If it is free or paid.
- Record - If it is possible to record your chats.
- Share screen - If it is possible to share your screen with other participants.
- Broadcast - If it is possible to broadcast your video.
- Collaboration - If it is possible to ex. write documents or view presentations together.
- Requires software - If there is a client that needs to be installed.
- Mobile version - If there is a mobile version of the service.
- File transfer - If it is possible to share files.
- Overhearing - If it is possible to overhear conversations of other people using the same service.
- Overseeing - If it is possible to oversee other people using the same service.
- Mingling - If it is possible to easily join a conversation without too much disturbance.

## **3 Related services**

Here follow some information about each service and what makes them unique. Each service is compared to the list of criterias[Appendix A].

### **3.1 Google WebRTC**

WebRTC[8] is a new open-source project which aims to enable real-time communication(RTC) in web browsers with simple JavaScript APIs. It's currently under development but Google Chrome and Mozilla Firefox have recently managed to communicate between the browsers[9].

### **3.2 Browser meeting**

Browser meeting[10] is a web application using WebRTC. It's a web conference tool which is very easy to use. Simply create a conference room and invite your friends.

### **3.3 frisB**

FrisB[11] is a web based application and a voice channel that freely 'ring invites' any telephone user on the planet into a conversation. No video or text chat is offered. This application uses WebRTC.

### **3.4 Video Conference in 3D-environment**

By using both WebRTC and WebGL you can for example video chat with your friends in a virtual 3D-environment. One application, WebGL Meeting[12], let you chat in real-time with the people you invite to your room.

### **3.5 Protosphere**

Protosphere[13] is a secure private 3D virtual environment where you control an avatar in order to collaborate and socialise with other participants.

### **3.6 O.L.I.V.E**

SAIC O.L.I.V.E[14] is an online Interactive virtual environment software platform that delivers interactive multimedia communication capabilities for collaboration, training, operations and education. It is possible to video and text chat through an avatar with voice, you can also record the conversations. During training and education with O.L.I.V.E broadcasts and collaboration is used. You are able see, hear and mingle with the other participants.

### **3.7 OpenQwaq**

OpenQwaq[15] is an open source software which aims to allow businesses to implement their own virtual world workspaces adjusted for their specific needs. It provides all the tools, data, and interactivity that people need to explore ideas, resolve issues, track progress, and be more productive.

### **3.8 Paltalk**

Paltalk[16] is an instant messaging service which allows users to communicate via text, voice and video chat. It lets users create their own public chat rooms where they can talk with their friends or find new ones. Paltalk exists in three different forms. Paltalk messenger which is a downloadable client, Paltalk Mobile, the phone version and Paltalk Express, a web version.

### **3.9 Tinychat**

Tinychat[17] is a small simple video chat. Create a chat room, invite your friends or make new ones. An interesting part of Tinychat is their API which allows you to implement your own fully functional chat room wherever you want. Other features are screen sharing, broadcast and more.

### **3.10 Skype**

Skype[3] is a service which aims to make it easy to stay in touch. It does this by enabling free internet calls, instant messages and video chat. What makes Skype special is the ability to buy Skype Credit for which you can make cheap calls to phones and mobiles, get online at public WiFi hotspots and send SMS worldwide. Skype require a client and has a mobile version.

### **3.11 Google Hangout**

Google hangout[2] allows you to video chat with up to nine other people in a hangout. One interesting thing with Google hangout is that it has apps, for example YouTube, Poker, and Google Docs. These makes it possible to for example, watch videos together or view presentations and diagrams with your co-workers. You can also broadcast your hangout with Hangout on air. Everything you broadcast is recorded and stored on your YouTube channel.

### **3.12 ooVoo**

OoVoo[4] is another interesting video chatting system. In contrast to Google hangout, ooVoo requires a client to be installed. Once installed you can video chat with up to eleven other people, send video messages and text chat. The video chatting can be recorded and uploaded to YouTube.

### **3.13 Second Life**

Second Life[5] is a 3D virtual environment where everyone you see is a real person and every place you visit is built by people in the virtual world. Setup and design your own 3D-avatar to join a 3D world. After downloading and installing the client you can start text chat with other people online in the world that you see. It exists different types of room you can visit, for example a night club, beach or London city town. There exists both paid and free versions. Some of the rooms also have voice chat to offer their visitors.

### **3.14 The Word Cafe online community**

The world cafe online community[18] is a website who offers a place online to have great and meaningful conversations about things you care to discuss about. There is no video chat or sharing media with each other, only text chat, more like a forum. Everybody gets their own blog to share their interest with everybody and start a conversation. The site is free to use, they offer their users to take online courses, some of them are free. The courses are held at the site. It exists a mobile page for your mobile phone or tablet.

## 4 Related papers

Online video collaboration has been researched for a long time by many different teams. There exists a large number of solutions for video conferencing systems each using different technologies to solve their specific problems. Back in 1995 some researchers in Japan created a video conferencing system called MAJIC[19]. It was a multi participant system that enabled eye-contact with life-sized images of each other. It also had a shared workspace which enhances the collaboration. Evaluation of the system showed that the background influences the sense of presence and that life-sized images gave a sense of reality.

The MAJIC system was big due to the life-sized images and eye-contact solution. Some other researchers proposed a solution for eye-contact using a single Kinect sensor[20]. They do this by rendering a gaze-corrected 3D model of the scene and transfer the gaze-corrected facial portion onto the original image using a face- tracker. The result was okay but not good enough to compete with reality.

Another gaze correction study is done to establish if it is possible to do this correction even with movement during a video conference[21]. The paper aims to show that, while both integrating eye-trackers into an Immersive collaborative virtual environments (ICVE) and video conference, allow people to distinguish being looked at and what else is looked at, when someone gazes into their space from another location, ICVE alone can continue to do this as people move. The result shows that only the ICVE supports eye gaze correction movement of the observer.

Other researchers tried to improve this feeling of presence and realism by using a movable camera[22]. The system has a remote camera which moves forward when a local user approaches the screen. This gives both users a good perspective when looking at the other person during a video chat. It gives the feeling of facing a remote person in the same room.

Microsoft researchers tried a different solution. Instead of using hardware they came up with a software solution[23]. They tracked the participants head and eye movements and used this information to place the head and eyes in a 3D-environment. Although the resulting system were slow, it showed that it is definitely possible to do with only software.

Group video conferencing requires some hardware and bandwidth in order to transfer the audio and video between all the participants. One paper

proposed an architecture for a peer-to-peer multipoint video conferencing system with layered video aimed for end-points with low bandwidth[24]. The system enables the participants to create a group conference using no more bandwidth than a point-to-point system. A prototype is created and the system is validated.

DigiMetro[25] is a proposed design for an application-level multicast system for a small-scale video conferencing tool. It takes use of multiple source-specific trees in order to keep the delay down. DigiMetro allows different bit-rates on different sources which makes it possible to use even with low bandwidth.

Many solutions by, for example Cisco and HP, require a large stationary and expensive setup in order to work and does not even then solve the problems good enough to give the feeling of sitting in the same room. The European FP7 project 3DPresence[26] came up with a concept for a high-end 3D Video conferencing system that aims to solve several of the problems that comes with video conferencing, including life sized participants, mutual gaze and gesture awareness.

In contrast, TAM TAM[27] is a minimalistic video conferencing system designed for the browser and mobile devices. It is built using Adobe Flash and Adobe Air, which makes it possible to run on platforms such as Windows, Android and iOS.

Online mingle and overhearing have received little attention from researchers. One of the few services which allows mingling is Second Life[5]. An evaluation of a big conference in Second Life done by IBM[1] showed that although the avatars helped created a feeling of being there, the sound of people talking travelled too far. This meant groups of people had to walk far away from each other or end up interrupting another or multiple other conversations. Some also said that it was hard to know if it's okay to join a conversation or not due to the lack of human expressions and gestures. An advantage of online mingling is that each avatar may have a name over his head which makes it easy to recognize acquaintances. A disadvantage however is that it does not feel as personal as if you had met face to face[6].

Overseeing in 3D worlds already exists[5][13][14][15], in a 2D world this cannot be realized in the same way as in a 3D world and this motivates the goal of the project.

## 5 What the site looks like

These pictures will give you an idea of how the site is build up and it is easier to understand the idea, technologies, features and of course the design.

### 5.1 Index/start page

The start page consists of three parts. When you enter the page you will first see the welcome message[Figure 1]. If you scroll down you will see the links to the cafés of the 2D-view[Figure 2]. If you continue scrolling you will see the links to the cafés of the 3D-view[Figure 3].

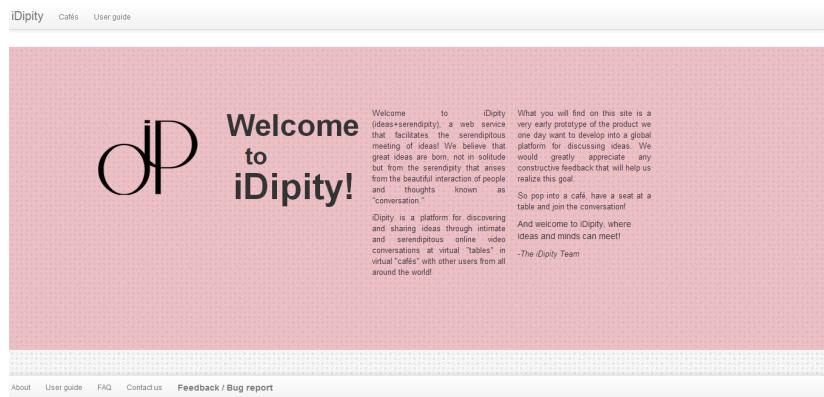


Figure 2: The start page with a welcome message and some information about the site.



Figure 3: The start page, the buttons are the six cafés the user can visit, this is the 2D-view.

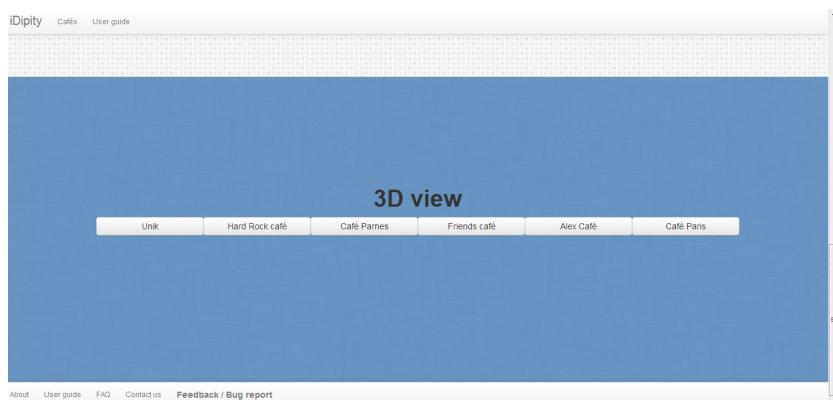


Figure 4: The start page, the buttons are the six cafés the user can visit, this is the 3D-view.

## 5.2 Caféview

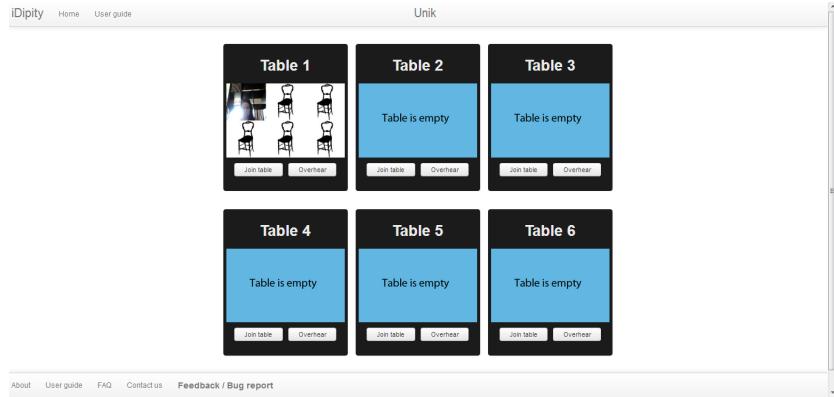


Figure 5: The 2D-view of the tables in the café Unik. In table 1 there is one person sitting down, the other seats is empty at that table.



Figure 6: The 3D-view of all the tables in café Unik. At the moment all the tables are empty. The left middle table has been turned around with the mouse, if you click the on the turned table you will overhear the conversation if the table is not empty. If you click at a table you will sit down at the table.

### 5.3 Tableview



Figure 7: This is how it looks like when you sitting down at a table having a video conversation with someone in the 2D-view. You see the text chat and the napkin.

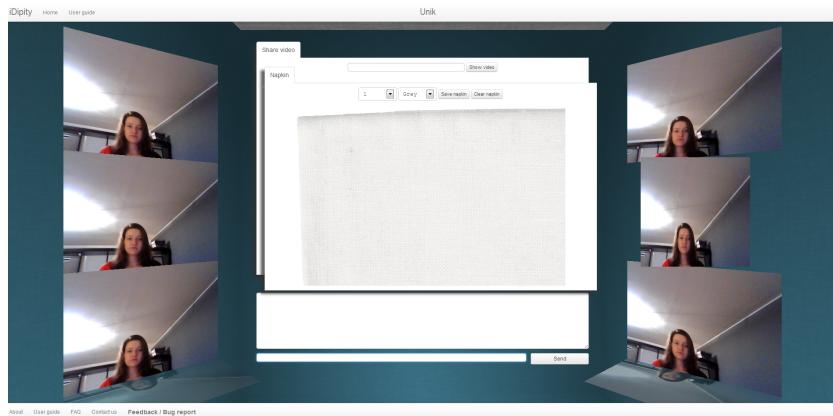


Figure 8: This is how it looks like when you sitting down at a table having a video conversation with someone in the 3D-view. You see the text chat and the napkin. The video stream in middle to the right has a mouse over it. That is why it looks different.

## **5.4 Use cases**

These use cases are made for a better understanding of how the system can be used. Three example will explain this.

### **5.4.1 Use case 1: Enter a café**

Lisa has a new idea for a product she wants to discuss. She starts her web browser and open Paris Café web page. She is met with a welcoming message and a list of the available cafés. After choosing a café she will be asked to enter her name into a form before continuing to the tables of the café. Once done she will be met by six “tables”.

### **5.4.2 Use case 2: Oversee and overhear**

When Lisa is inside a café she can look at the six tables. Each table will display an image of the persons, if any, sitting at the tables. When she has found a table that looks interesting she can choose to overhear and oversee the persons at the table by clicking the button “overhear”. The image will then be replaced by live video streams of the participants, including audio.

### **5.4.3 Use case 3: Sit down at table**

After finding a table to sit down at she clicks “Join table”. Participants of the group will hear a knocking sound and find a notification in the upper right corner. They will now answer yes or no. If the majority answer yes she will sit down at a table where the video chat begins.

## 6 Technologies

### 6.1 Node.js

Node.js[28] is a framework built on Chrome's JavaScript runtime, this enables JavaScript on the server side. Node.js allows us to quickly install Node modules which makes it easy to build fast and scalable web applications.

### 6.2 WebRTC

WebRTC[8] (Web Real-Time Communication) enable browser to browser applications for voice calling, video chat and peer-to-peer file sharing without any plugins. A requirement for WebRTC is to use it together with HTML5. WebRTC is available in Chrome's stable version and in Firefox's Nightly version.

WebRTC allow the browsers to connect directly to each other using peer-to-peer technology. This removes the need for a powerful server. Although server is still needed to establish the connection between the participants.

### 6.3 Licode

Licode[29] is an open-source project that is based upon WebRTC technologies. It comes with Erizo, a Multipoint Control Unit(MCU) and an easy to use API for creating video conference rooms directly in the browser.

#### 6.3.1 Erizo

Erizo is an MCU written in C++. It's fully compatible with WebRTC standards and protocols. Erizo does not currently provide any transcoding of videos but it's being worked on right now.

#### 6.3.2 Nuve

Nuve makes it possible to create and manage video conference rooms. It holds information about each room and let's you ask for example a user list. It is designed to scale on the cloud.

## **6.4 MongoDB**

MongoDB[30] is an open-source, document-oriented database designed for ease of development and scaling. Stores data as JSON-like documents dynamically and making the integration of data and our prototype easier and faster.

## **6.5 Mongoose**

Mongoose[31] is an object data modelling library for MongoDB. It allows you to create models for your data. This gives better structure to MongoDB while maintaining it's flexibility.

## **6.6 Express**

Express[32] is a fast and small node.js web application framework, providing a powerful routing system which allows you to easily create single and multi-page web applications. It helps you manage everything, from routes, to handling requests and views.

## **6.7 Twitter Bootstrap**

Twitter Bootstrap[33] is a collection of tools for creating beautiful web applications without being an expert on design. It includes HTML, JS and CSS-based design templates for typography, forms, buttons, charts, navigation and other interface components. The reason for using Twitter Bootstrap is to get a nice interface and it is easy to use.

## **6.8 HTML5**

HTML is used to structure and presents information on a website, HTML5 is the newest HTML standard. It's designed to make web programming easier and more interactive. Some of the new features that are interesting for our project is the <video> and <canvas> elements.

## **6.9 Three.js**

Three.js[34] is a lightweight cross-browser JavaScript library/API used to create and display animated 3D computer graphics on a Web browser. Three.js

scripts may be used in conjunction with the HTML5 canvas element, SVG or WebGL.

### 6.9.1 WebGL

WebGL (Web Graphics Library) is a JavaScript API that allows the client to render powerful 3D graphics within the browser without plugins. It runs on computers GPU.

Our purpose of using WebGL is to try the café Paris website with a virtual 3D-environment. This makes a good case to try out a mixed reality, by combining a virtual 3D-environment with the users video stream it may be possible to transfer the feeling of actually being in a real café into the website.

## 6.10 GitHub

Git is used for revision control and code management. Git is a open source distributed revision control system. A respository was hosted at GitHub[35] in order to simplify the setup.

## 7 Method

Some information was collected before implementing the service. A study was made in order to find suitable ways of overhearing, overseeing and mingle. And a comparison of different WebRTC frameworks was necessary for us due to the video conference features.

### 7.1 WebRTC comparison

WebRTC is a new web standard for peer-to-peer communication. WebRTC allow browsers to send information to each other, without going through a server. The focus is currently on audio and video communication but they will implement a data channel in the future. This will drastically decrease the server load.

Many different WebRTC frameworks were compared before settling with Licode. Licode offered something that others didn't, and something that is necessary if the system would be used on a larger scale. That feature is a Multipoint Control Unit(MCU). Here follows a short summary of the runner ups.

#### 7.1.1 Easy RTC

EasyRTC[36] is an open source WebRTC framework with cross browser support. It allows you to easily setup a Node.js server and provides an API for making advanced web applications in no time. EasyRTC uses websockets for fast message passing between clients.

#### 7.1.2 WebRTC.io

WebRTC.io[37] aims to make it easier to use the new webstandard. It does this by providing a simple abstraction layer for the otherwise semi-low level WebRTC.

#### 7.1.3 Holla

Holla[38] is another abstraction layer with the same goal, to make it easier to work with WebRTC. Holla provides a client and a Node.js server module. The server is used to initiate the communication between the clients. Each

client registers a username that is used for making calls or sending data to each other.

## 7.2 Survey

A survey was made during the early stages of the project[Appendix B]. The idea was to get a grasp of how people feel about visiting a virtual café online as well as to capture their opinion on what aspects of a real café makes it such a good place for intimate discussions.

### 7.2.1 Evaluation

The most important aspects of a café is the intimate environment and the other people in the café. This means that in order to capture that intimate feeling of a café, a design that gives a similar feeling has to be created.

The result showed that people like the ability to overhear other conversations going on in the café. But it also showed that it can be a disturbing factor when you cannot block out the noise.

Mingle is big part of the master thesis. How can you ask to participate in a conversation without disturbing the group of people having the conversation. The answers shows that a subtle knock or a short personal message to the group would get their attention without drawing them away from the conversation.

There were a few questions concerning a 3D-environment. The answers however were pretty much evenly distributed between “yes” and “no” and due to the amount of participants there are no clear answer. However, it can be hard to imagine going to a café in a 3D virtual environment and a new survey when the prototype is up and running could result in different answers.

## 7.3 Prestudy conclusion

This prestudy show the huge amount of different video conferencing systems each with their own solution for specific problems. All from high-end systems designed for big businesses, to small systems for personal use.

The result however, strongly motivates the research questions. While there exists a few good services for video conference and online group collabora-

ration the only ones that support multiple groups is within a 3D-environment and not one of these have strong focus on mingling.

There is still room for improvements in overhearing and mingling, both inside and outside a 3D-environment.

If using virtual 3D-environments, a mixed reality is interesting to mix real peoples video streams with a virtual environment. If a person who has joined a conversation at a table does not want to share his video stream that person could be represented by an avatar instead of the video stream. It would also make it possible to enable human expressions and gestures[9].

## 7.4 System architecture

The system consists of two different parts, the backend and the frontend.

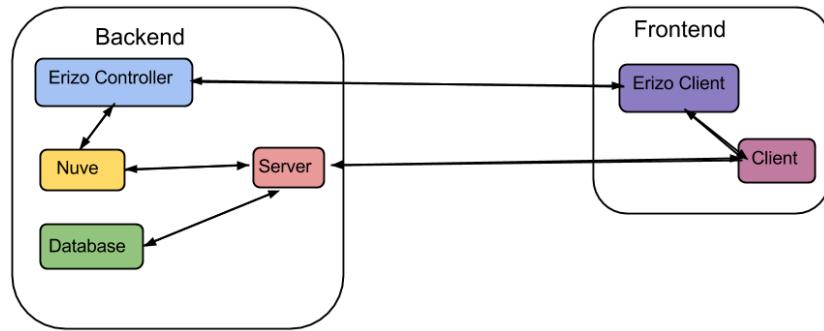


Figure 9: A chart over the system architecture.

### 7.4.1 Backend

The main part is the Node.js server. The server acts like a middleman for the Licode modules and the frontend. It communicates with the clients using standard HTTP requests and a simple RESTful API together with Express.js. It allows the client to easily GET/POST data from/to the server. The server also communicates with Nuve via a token based communication mechanism.

Example: GET /api/getCafe/Unik from the database results in:

```
{  
    "name": "Unik",  
    "table1": "513dd08a07aa2f143700001f",  
    "table2": "513dd08b07aa2f1437000020",  
    "table3": "513dd08c07aa2f1437000021",  
    "table4": "513dd08d07aa2f1437000022",  
    "table5": "513dd08e07aa2f1437000023",  
    "table6": "513dd08f07aa2f1437000024"  
}
```

#### 7.4.2 Database

The database stores all the information about the cafés, all the rooms and what rooms that belongs to what café and the snapshot taken by all leaders in each room. Each café consists of six rooms. The client communicates with the server in which the server communicated with the database.

#### 7.4.3 Server

All the cafés and rooms are created in the server using Nuve Client. An access request will be sent and accepted via Nuve for each user trying to connect to a room via a token-based authentication mechanism. This mechanism allows the server to create access tokens, and it will provide these tokens to the clients.

#### 7.4.4 Nuve

Nuve is a Licode module that manages the rooms or the tokens to access to a determined room. Nuve Client API allows the server to, for example, create rooms, get rooms, and get users of a certain room.

#### 7.4.5 Erizo Controller

Erizo controller communicates with Erizo. It handles the subscription of streams, sends the streams to all subscribers and controls the rooms. Erizo controller scales well and spawns new instances when the pressure is high.

#### **7.4.6 Frontend**

The frontend is built using HTML5, JQuery, JavaScript and Erizo. The frontend communicates with two different parts of the system. It speaks to the server for HTML/CSS/JS files, tokens and database access. All the communication between different clients goes through Erizo controller. This means that all the data (audio, video or data) from a stream is sent to Erizo controller and then forwarded to all the client who subscribes to said stream.

#### **7.4.7 Erizo Client**

Erizo client provides a JavaScript API for communicating with Erizo Controller. It allows the client to, for example, publish and subscribe streams and create video elements.

### **7.5 Setting up a connection**

The client requests a token from the server which in turn ask Nuve. The token includes an ip-address to Erizo Controller. When you connect to a room the client establishes a websocket connection with Erizo Controller. When you publish your own stream or subscribe to other streams the web client exchanges Session Description Protocol messages with Erizo Controller in order to negotiate media type, format and all associated properties. When the negotiation is done the client and Erizo Controller establish a Secure Real-time Transport Protocol connection to deliver audio and video. This process usually take five to ten seconds to accomplish.

## 8 Result

Here follows the result of the main features and different pages on the website.

### 8.1 Design

The design is an important feature of the system. In order to capture the feeling of being in a café, a modern design that is just as intimate as a real life café needs to be created, as well as finding a nice way to present overseeing, overhearing and mingling to get the feeling of being in a café. The design is mostly done using Twitter Bootstrap. Bootstrap helped attaining a clean and modern look. A few goals were set up for the design, one was that the site would look just as good and work just as well on big and small resolutions. Therefore, percentages were used on almost all sizes so everything keeps the aspect ratio when the window is resized.

Another goal was the site navigation. It should be self-explanatory, easy to navigate and accessible in a few clicks.

#### 8.1.1 The 2D-view

The website has three important pages, index, Caferview and Tableview. When the user first enters the site they will see a welcoming message, explaining the purpose of the site. If they scroll down they will see a list of café buttons for each of the view type, 2D and 3D. On the top bar is a link to a user guide and on the bottom a link to send the creators a feedback mail.

The Caferview is after you have entered a café. It displays the tables of the café and an image of the participants of each table. The user can chose to sit down at a table or overhear the table by pressing respective button.

The Tableview consists of two parts. There is what is called a table in the middle of the page, with room for three streams on each side of the table. There are tabs on the top side of the table. These are filled with features for the users. While video chatting, the users can draw, watch a YouTube clip and text- or voice-chat together.

#### 8.1.2 The 3D-view

The first idea was to create an environment with the feeling of being in a real café, a room with tables, the video streams in the room of all the

participants and also you can walk around in the café to see all the tables. After the first survey it was found out that this type of environment was not desirable. Because of this it was decided that much of the design of the 2D-view would be reused and represented in a 3D-view using WebGL to enhance the feeling of being in the same room, instead of going with the avatar-based 3D-environment that was considered earlier.

The Caferview has been completely remade. When you enter a cafe you will see a large room with blocks that represents tables. These blocks are rotatable. You will also see a big screen on the wall, after rotating a block and clicking on the backside, the live video streams will be displayed on that screen.

The Tableview has been improved in many ways. The video streams are slightly rotated towards the center of the screen. This gives the feeling of sitting around a table with your friends. The background has been given depth in order to give the feeling of being in a room. The tabs with the napkin and share YouTube video is now in different layers. They will switch places when you click on them.

## 8.2 Leader

Each room needs its own leader to enable some of the features below. If a room is empty when someone enters, he or she is automatically chosen to be the leader in that room. A new leader will be chosen if the current leader leaves the room, this is done by letting each user ping the server three times. The one with the lowest mean value becomes the leader. Doing this assures that the leader will have the lowest latency of the group. If two or more users have the same mean value, the one with the highest stream ID becomes the leader. By letting the leader take care of some of the features instead of letting the server handle it, the server load is reduced. The server does not even know about what the leader is up to due to the peer-to-peer connection.

## 8.3 Peer-to-peer

WebRTC allows data to be sent between the clients using peer-to-peer. This feature is taken advantage of in order to release some stress from the server. The media and data streams are separated in order to create a simple messaging service that can be used by all features on the site, to communicate between the clients without the server.

## 8.4 Overhearing

You can only block out noise to some extent in a real environment. However, in a virtual environment one can chose when and what to overhear. Just like one can focus the hearing to a certain point one can chose to listen to a certain table.

The audio cannot be separated from the video and because of that it was decided that audio as well video would be displayed when you overhear a table. When pressing the overhearing button, the client will subscribe to all the streams at the table without publishing a local stream. The snapshot image for the table will be replaced with live video streams. This means the user can hear and see the group without them being able to see or hear the user. When the user choses to end the overhearing, the streams will be removed and replaced with the snapshot image.



Figure 10: Real-time overseeing and overhearing in the 2D-view.



Figure 11: Real-time overseeing and overhearing in the 3D-view.

The reason for not displaying the text chat, drawing or if a video is watched was to give the users a bit of privacy.

It is not possible to overhear other tables when sitting in a table. This is because it would draw attention from the conversation.

## 8.5 Overseeing

The prototype has three kinds of overseeing. The first one is directly after entering a café. Images containing a snapshot of every stream for every table in the café[Figure 12] is displayed for each table. The snapshots updates every 30 seconds, when a new user sits down at a table and when a user leaves a table. The snapshot is sent to the server by a chosen leader of each table.

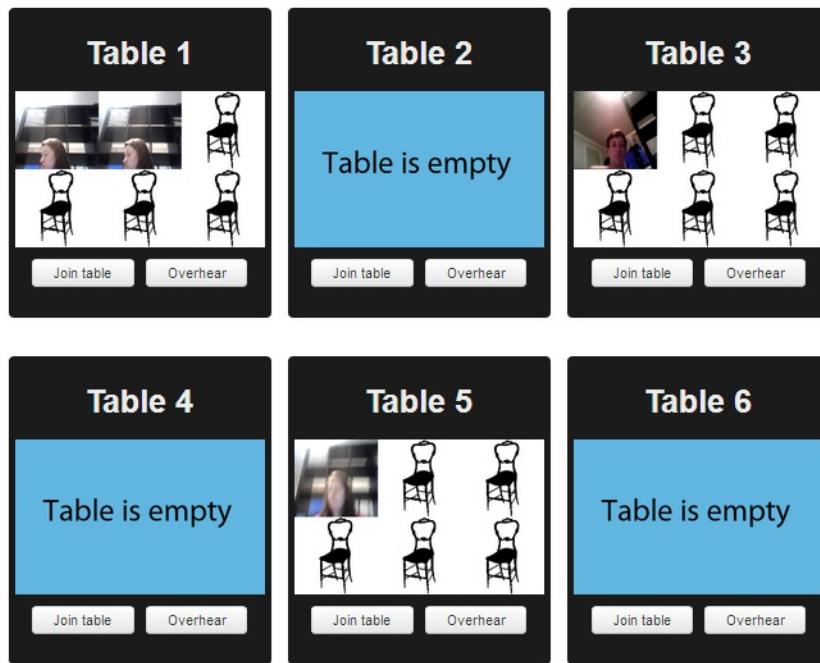


Figure 12: Shows overseeing when you enter a café and view the tables.

The second type of overseeing is in real-time. Before sitting down at a table the user can chose to oversee and overhear the by clicking a button. The user can then see and hear all the participants at the table in real-time[Figure 10, 11].

The third type of overseeing is when the user is sitting down at a table. In a real café one can actually look around and see the other participant while still have a conversation with the participants at the table. To simulate this experience, a bar was added to the top of the page that one can pull down[Figure 13] to see the snapshots of the other tables in the same café.

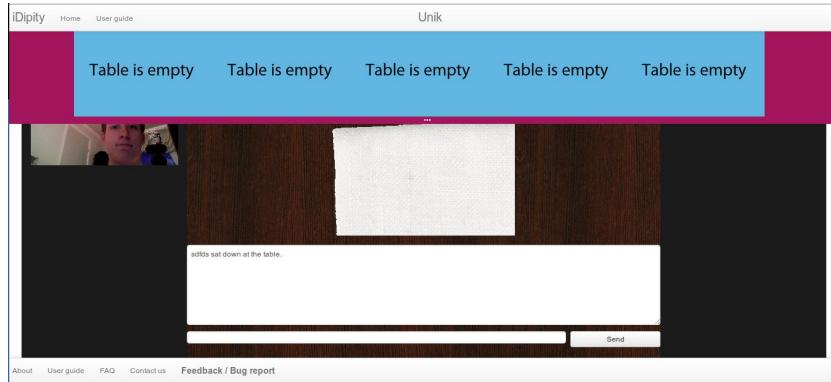


Figure 13: Pulldown overseeing from inside a café. Displays the snapshot images of the other tables in the café.

To enable overseeing on the website the leader takes a snapshot of every user in the room as mentioned above. The client will then encode the binary data from the image to a base64 string and send it to the server. The server will then store the string in the database.

To see all the snapshots in the Cafédview the client sends an HTTP request to the server and gets a JSON-object in response.

## 8.6 Mingle

The idea was to have a subtle way of telling the participants at a certain table a user wants to sit down and join the conversation. There were many ideas for how this could be realized, for example a personal text message to the group, record a voice message or just a simple knock. After the study it was decided that the knock would be best suited for the service because reading a text message or listening to a voice message would take the focus away from the conversation.

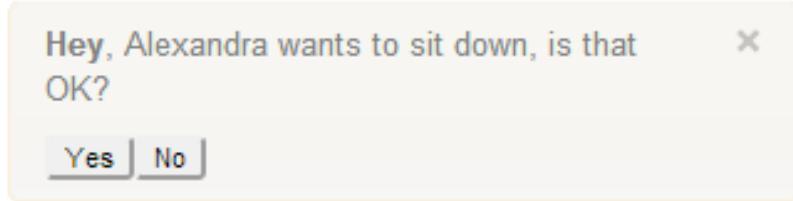


Figure 14: A knock notification. Alexandra wants to join the room.

Instead of having the server keep track of which user is in which table and in which café, the knocking and joining tables are being handled within the group.

After pressing the knock button, a data stream is published and other data streams are being subscribed to. A message is sent to all the users in the room and a notification[Figure 14] pops up in the upper right corner. All the users in the room gets to vote by pressing “Yes” or “No”. The notification disappears after 20 seconds. Any unanswered notification results in a “No”. The answers are sent to the requesters client and the majority has to answer “Yes” before a user can connect to the table and publish the local media stream.

## 8.7 Paint

Many ideas starts on a napkin, perhaps in a café just hanging out with colleagues or friends. The napkin is an HTML5 canvas that works together with mouse listeners to create basic painting functions. When finishing drawing a line, an array containing all the coordinates to be draw will be sent to all other participants at the table. The lines will scale to fit the other participants canvas sizes.

When a new user sits down at the table he or she will get the current napkin as a base64 string by each tables leader. The napkin can also be saved locally to your computer. The clear napkin button will reset the napkin for all users. Color and thickness can be chosen in the dropdown menus.

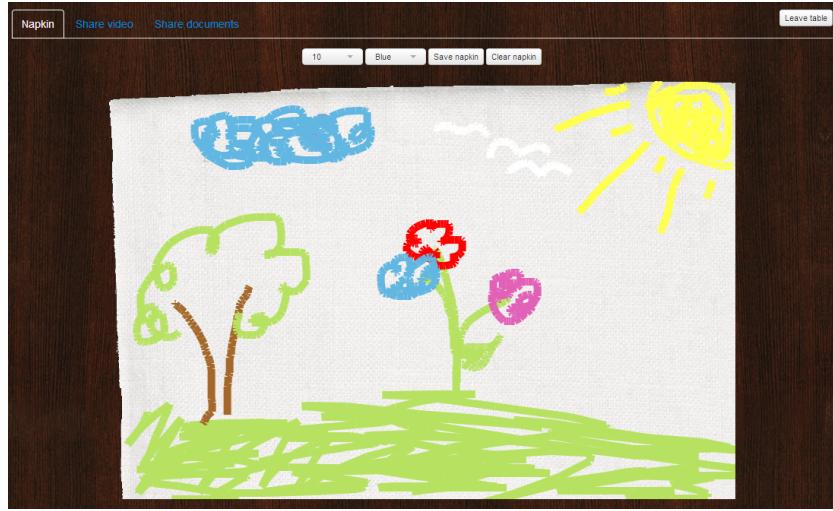


Figure 15: A drawing on the napkin shared with all participants in the room.

## 8.8 YouTube

This is a synchronized YouTube player that plays any video from YouTube [Figure 16], it's built using the YouTube API. To play a video one has to paste an URL into a field and it will be shared amongst the group. Pressing play or pause will play and pause the video for all participants in the room. If a user decides to close the video it will only close locally and the rest of the group can still watch the video together.



Figure 16: The youtube player in the 2D-view.

## 8.9 Text chat

The text chat is very basic[Figure 17]. Each user pick a username when entering a café. The username will be displayed whenever a message is written. The chat is also used for some notifications, for example when a new user sits down at a table. All the text messages are sent to all other participants through a data stream that is published when knocking on a table.

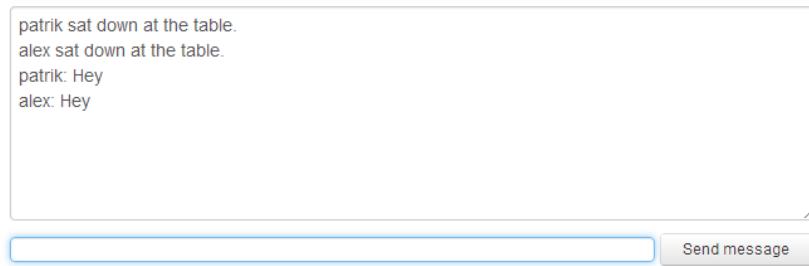


Figure 17: A knock notification. Alexandra wants to join the room.

## 8.10 Server performance

The reason for using an MCU is to reduce the required bandwidth for every client. This means that more people with not as good bandwidth can still

use the service even if they do not have very high bandwidth.

Using peer-to-peer would mean that every user sends his or her stream to every other user in the room, as well as to everyone that is overhearing. Instead by using an MCU, each user sends a stream to the MCU, Erizo. Erizo then perform some basic transcoding, if any, before forwarding the data to each other user in the same room, as well as to everyone who is overhearing.

#### 8.10.1 PeerConnection vs MCU

To understand the importance of an MCU, a comparison between the required bandwidth on both server and client, with and without the MCU will be made.

**Without MCU.** Scenario: A room with six users and two people overhearing and overseeing in real-time.

Let  $U$  denote amount of users in a room,  $O$  the amount of users overhearing a certain room and  $S$  the bandwidth of a stream.

Each client then send his or her stream to each other client in the room, plus each overhearing client.

The data sent from and received at each client can be calculated with the following equations.

The data sent from each client:

$$S * (U - 1) + S * O \quad (1)$$

The data received at each client:

$$S * (U - 1) \quad (2)$$

The data received at each overhearing client:

$$S * U \quad (3)$$

The data sent from and to the server is close to zero and will not be counted.

Let's say each stream has a bandwidth of 200 KB/s and calculate the scenario above.

$$200 * 5 + 200 * 3 = 1600KB/s = 12.8Mbps \quad (4)$$

$$200 * 5 = 1000KB/S = 8Mbps \quad (5)$$

$$200 * 6 = 1200KB/s = 9.6Mbps \quad (6)$$

Equation 4 shows the amount of data sent from each client in the room. Equation 5 shows the amount of data sent to each client in the room. Equation 6 shows the amount of data sent to each overhearing client.

The calculations show that for a client to work with our service with streams of 200 KB/s, it needs a bandwidth of atleast 8 Mbit downstream and atleast 8 Mbit upstream. And this is without anyone overhearing. This is a very high bandwidth requirement for one client and it also would not work on a larger scale if some table have too many people overhearing at the same time. If the system instead had an MCU it would not matter which table got overheard since the total amount of data sent from the server would not change depending on if five people are overhearing the same table or five different tables.

**With MCU.** The Multipoint Control Unit decreases the bandwidth load of the clients. Instead of sending the stream to all other participants, each client will send the local stream to the MCU. The MCU then forwards it to all subscribers. So same scenario again, this time with the MCU.

The data sent from and received at each client can be calculated with the following equations.

The MCU forward each stream to all the subscribers in a room and to all overhearing client. The data sent from the server:

$$U((U - 1) * S) + O * U * S \quad (7)$$

Each client sends the local stream to the MCU. The data sent to the server:

$$U * S \quad (8)$$

The data received at each client:

$$S * (U - 1) \quad (9)$$

The data received at each overhearing client:

$$S * U \quad (10)$$

An example. If there is one room with six users and three people overhearing with a bitrate of 200 KB/s results in:

$$6 * 200 = 1200KB/s = 9.6Mbps \quad (11)$$

9.6 Mbps sent to the server.

$$6 * (5 * 200) + 3 * 6 * 200 = 9600KB/s = 76.80Mbps \quad (12)$$

76.8 Mbps sent from the server.

$$200 * 5 = 1000KB/S = 8Mbps \quad (13)$$

8 Mbps sent to each client in the room.

$$200 * 6 = 1200KB/s = 9.6Mbps \quad (14)$$

9.6 Mbps sent to each overhearing client.

Now let's assume the server has a dedicated 1 Gbit connection.

$$1Gbit = 131072KB \quad (15)$$

A 1 Gbit can in theory send and receive up to 131 072 KB of data each second. The bandwidth needed to fill a room without overhearing is

$$6 * (5 * 200) = 6000KB/s \quad (16)$$

The maximum amount of video conferencing rooms that the system can handle with a 1 Gbit connection can now be calculated.

$$131072/6000 = 21.8453333333 \quad (17)$$

So, a 1 Gbit connection can manage almost 22 filled rooms at a time. But keep in mind that 200 KB/s equals to over 1.5 Mbps. In comparison, Skype recommends 500 kbps bandwidth for high quality video[39].

### 8.10.2 Performance test

A performance test was done in order to see how the system would perform under pressure and to get a feeling of the hardware requirements on a larger scale. Four computers including the server were used during the test.

Device	OS	Processor	Ram
iMac	Mac OS X	Intel Core i5 3.1 GHz	16 GB
Asus Laptop	Windows 7	Intel Core i7 2.2 GHz	8 GB
Mac mini	Mac OS X	Intel Core i7 2.7 GHz	4 GB
Server	Ubuntu 12.10	Intel Xeon i5 2.4 GHz	2 GB

As seen in the theory, the bottleneck lies in data sent from the server. The CPU usage peaked at a bit over 50% while the video quality of the streams lowered because the server couldn't send out more than 4 MB/s.

The result of the test followed the theory closely, as seen in the table below.

Streams(Nr)	Incoming data(KB/s)	Outgoing data(KB/s)	Comment
1	45	0	
2	250	500	
3	800	1600	
4	1000	3200	
5	1000	4000	
6	800	4000	Lower video quality

The video quality then continued to decrease for every new stream published or subscribed. The data sent from the server might be reduced and CPU usage increased once Licode start transcode/mixing the streams.

## 9 Discussion

During this project we discovered problems with some features. Along with that discussions we also present feedback from test users.

### 9.1 Issues

We have one issue with the knocking function. If a table has five participants, two more can be accepted to sit down, even though the maximum amount of users for a table is six. This happens because of the delay when connecting to a room and publishing a stream. The room does not count as being full until it has six published media streams. So if two or more users connect to the room at roughly the same time, they will both be able to publish their streams.

### 9.2 The design

We had almost no experience of web design and WebGL before starting this project. Here follows some improvements we think could be done to enhance the user experience.

- We have blocks representing tables in the 3D version of Caféview. These could be exchanged with models of a table.
- We think having models of the persons around the tables and map the video stream or snapshots to the face of the model would give a higher sense of reality to the service.
- Furnish the 3D Caféview with perhaps a bar, paintings and such, to make it look like a real room.

### 9.3 User test

We collected a group of people to test our prototype in an attempt to get answers on our research questions. The test begun with everybody entering the same café and sitting down at the same table in the 2D-view. The next step was to test all the features. A drawing was made, we watched a YouTube video together and chatted together using both text and voice. After a time of playing around we sent them out of the room, one by one,

to try the features outside the table, in the overview of all tables, overseeing and overhearing. We also tested overseeing when sitting at a table.

After the whole group tested all the features, they answered a questionnaire based on our research question.

The group found the snapshot images very practical and a good way to display the visitors or the café. They also liked being able to "look around" at the other tables while sitting at a table of their own. The overhearing part seems to have been a nice experience but they felt a bit freaky when listening to and seeing another conversation while not being a part of it. But they also said they would probably feel more comfortable overseeing and overhearing other people after using the service for a while.

People generally like the knocking because it was a good way see who wants to sit down at the table. However, they would like some more information about the person before deciding if he or she can join the conversation, for example a personal text message or being to check a user profile.

There were mixed feelings about having other people overhearing the conversation. Some thought it was exciting and a bit scary, others didn't even think about it. The group agreed on that the content of the conversation changes when they know other can be listening in on the conversation. That you have to think about what you say, just like in public places.

## 9.4 Privacy Issues

Privacy issues on the web has become a popular topic and we see many aspects of the prototype that could feel like invasion of privacy for many people. Most of these things can be done in real life as well, but not to the same extent.

Anyone can record any conversation that is taking place online using free software without the participants knowledge. In real life they would not be able record as easily without people noticing. Compare this to the recent discussion about the usage of Google Glass. Google Glass has even been banned in some cafés[40]. One big difference between our site and a real café is that you know someone could be overhearing and possible recording, before using the service.

The same goes with the overhearing. It is currently not possible to see if someone is overhearing and if so, who it is. It is much harder to overhear in a real café, as people can notice you and change subject or lower their voice.

When overhearing you will hear everything they are saying, even if they are whispering. It is just like sitting at the same table but being invisible. In a real café there is background noise, for example music playing or other groups of people talking. This together with the ability to whisper gives people privacy that isn't possible with the current state of the prototype.

There are many ways to make the overseeing and overhearing, for example making it possible to see if someone is overhearing or implementing a way for the users to whisper without being overheard. But for now it should be enough to know that you could be recorded or overheard and then decide if it is worth the risk.

## 9.5 Performance test

The test result showed that the system is really heavy on the bandwidth. The MCU helps a lot by taking the stress away from the clients and making it easier to decide on a minimum bandwidth requirement. But it is not needed as much when there are few people in a room with no one overhearing. In fact, if there are only two people in the room both alternatives uses as much bandwidth, but a peer-to-peer connection would send the stream directly to each other resulting in less stress on the server without increasing the load on the clients.

So we see that we need an MCU in order for the service to work well with a large amount of clients. But the problem is still the bandwidth requirement and we see two solutions to this problem. The first one is to transcode the streams. This would increase the CPU load but drastically decrease the data sent from the server. The second solution is to limit the bitrate of the streams. The video quality as it is now is really high compared to other online video conference services and limiting it to for example 125 KB/s would drastically increase the maximum number of users on the system.

## 10 Conclusion

We feel that we have reached the goals we set up earlier in the project. We have explored different solutions for overseeing, overhearing and mingling online, in a 2D and 3D world and created a working prototype using modern technologies. Although we have presented many different solutions for our main problems, we feel that we have only touched the surface of what could be done with each respective problem. But we hope this project makes a stable foundation for future research in the field.

We are happy with how both views turned out, even though both of us lack experience in web design. We believe that both views are perfectly suited for the service, however, there are pros and cons of both views. The 3D-view provides a more convincing feeling of being in a room, but the quality of the video streams inside the 3D world decreases. The 2D-view is easier to navigate for a first time user or a less experienced user.

All the key features of the service works just as well in both views. Which view works best depends on the target group. A good looking 3D-environment probably draws a younger crowd than the traditional 2D-view.

We had four research questions for this project that we now feel comfortable answering. These real life café features like overseeing, overhearing and mingling all has an effect on the experience of online video chat.

They all make it more like a real life experience like going to café. People will have a favorite café with regular visitors. Overseeing helps to recognize friends and help find new interesting people.

Overhearing makes the participants think before talking which can lead to more serious conversations compared to other video conference services, especially if it's an anonymous one. But it can also lead to shy people holding back even more. But we think that it is a good effect because people will get used to it and hopefully not have problem with it in the future.

We believe this is a step forward towards achieving more lifelike environments online. Lots of people go to conferences every year to meet interesting people. This costs a lot of money for the participants or their companies. Imagine if they instead could use a service like The Online Paris Café, meet and mingle online and get the same experience as you would on a conference. This may one day be possible if more research is done in this field.

## 11 Future work and improvements

We set out to build a prototype that includes overseeing, overhearing and mingle, and that's what we have done. But this is a project done in limited time so there are of course plenty of ways to improve the system.

We wanted to activate overhearing by simply holding the mouse over a table. We tried it and it worked, but it was far from optimal. Connecting to a room and subscribing to a stream actually takes some time to do. And holding the mouse over a table for all that time is simply not something you want to do. A solution to this problem would be to first connect directly to the other users while waiting for Erizo to connect and then switch over to Erizo when it's done. This would drastically decrease the time required to hover before you start overhearing the users in the room.

To get the feeling that you actually having a conversation face to face with a person online in a video conference, the eyes and body gestures is important. In the related work part we discussed an article about eye gaze. They looked at if it is possible to do eyes correction with movement during a video conference[20]. This can be interesting to test and see if the experience gets better for the user.

A new idea was given to us during the presentation of the project. When sitting in a real café, one can lean in towards another person and whisper in his or her ear. No one will be able to hear that conversation. This is something that our service lack, the ability to say something knowing that no other person will be able to hear it. This could be implemented either by providing a private text chat with another user or by establishing a peer connection to another user, while "muting" or unpublishing the local media stream so the other participants will not hear the conversation.

# **Appendices**

## **A Comparison**

This appendix is a comparison of all features of each related service.



## **B Survey**

This appendix is our first survey, the questions and the answers.

[Edit this form](#)

# Questionnaire

Hello! We, Alexandra Tsampikakis and Patrik Burström, are currently doing our master thesis, researching how an environment like a café can be realized online. We would appreciate it very much if you took a few minutes to answer some questions in order to make our work as good as possible.

## Question 1

A café can be a stimulating environment for many people. What do you think makes a café such a good place to sit down and have interesting conversations?

1 is not important

5 is very important

### The intimate environment of a café

1 2 3 4 5



### The other visitors in the café

1 2 3 4 5



### The ability to over-hear and get inspired by other conversations taking place in the same café

1 2 3 4 5



### The ability to over-see other visitors in the café

1 2 3 4 5



### The ability to mingle and participate in conversations with new and interesting people

1 2 3 4 5



## Question 2

Imagine sitting in a café online. You over-hear an interesting conversation you want to participate in.

What do you feel would be the best way to determine if it's okay to sit down at the table?

1 is not important  
5 is very important

**A simple knock, like knocking on a door**

1 2 3 4 5

**Write a personal message to the group**

1 2 3 4 5

**Record a personal voice message to the group**

1 2 3 4 5

**Just grab a chair, sit down and start talking**

1 2 3 4 5

**Other, please describe**

## Question 3

Imagine sitting in a café online. You're in the middle of an interesting conversation with your friends. A person over-hears your conversation and wants to participate. How would you like to be informed that the person wants to participate?

1 is not important  
5 is very important

**A simple knock like knocking on the door**

1 2 3 4 5

**A written message from the person**

1 2 3 4 5



**A personal voice message from the person**

1 2 3 4 5

---

    **Just grab a chair, sit down and start talking**

1 2 3 4 5

---

    **Other, please describe****Question 4****In a real café it can be hard to block out noise from the other tables, is this something you find disturbing?**

1 2 3 4 5

---

 no     yes**Question 5****How would you feel about visiting an online café in a virtual 3D environment?**

1 2 3 4 5

---

 uncomfortable     very comfortable**Question 6****Would you like the ability to maneuver a customizable avatar inside the café?**

An avatar is a graphical representation of the user in a virtual 3D environment.

**Question 7****If you're sitting down at a table with other participants, would it be important to see the faces of the other participants or would their avatars be sufficient?**

## Question 8

Are gestures important for conversations online?

[Submit](#)

Never submit passwords through Google Forms.

Powered by



This content is neither created nor endorsed by Google.

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

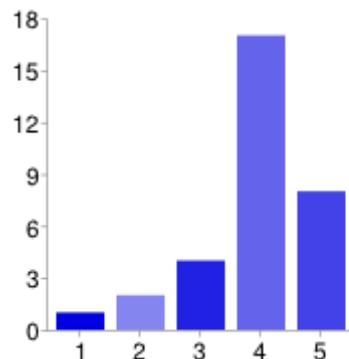
# 32 responses

[View all responses](#)[Publish analytics](#)

## Summary

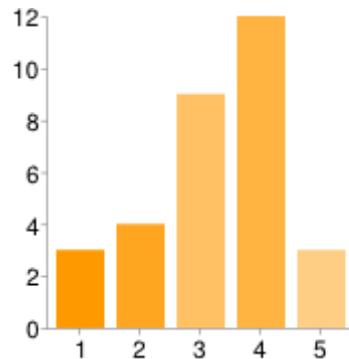
### Question 1

#### The intimate environment of a café



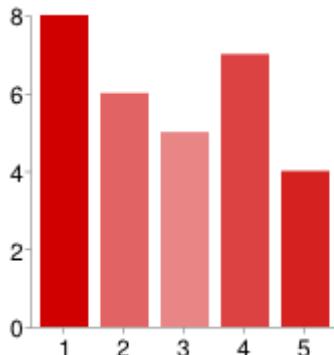
1	<b>1</b>	3%
2	<b>2</b>	6%
3	<b>4</b>	13%
4	<b>17</b>	53%
5	<b>8</b>	25%

#### The other visitors in the café



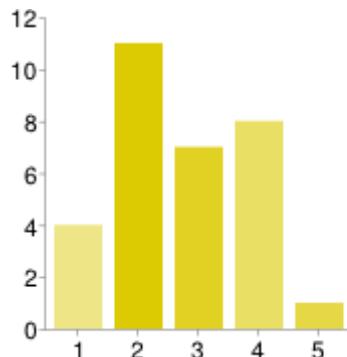
1	<b>3</b>	10%
2	<b>4</b>	13%
3	<b>9</b>	29%
4	<b>12</b>	39%
5	<b>3</b>	10%

#### The ability to over-hear and get inspired by other conversations taking place in the same café



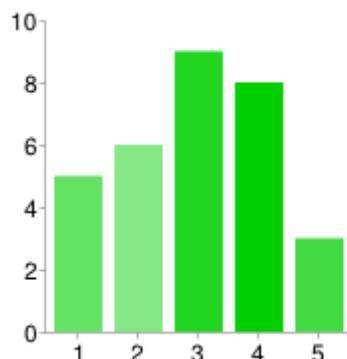
1	<b>8</b>	27%
2	<b>6</b>	20%
3	<b>5</b>	17%
4	<b>7</b>	23%
5	<b>4</b>	13%

### The ability to over-see other visitors in the café



1	<b>4</b>	13%
2	<b>11</b>	35%
3	<b>7</b>	23%
4	<b>8</b>	26%
5	<b>1</b>	3%

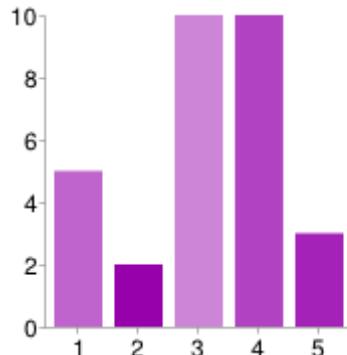
### The ability to mingle and participate in conversations with new and interesting people



1	<b>5</b>	16%
2	<b>6</b>	19%
3	<b>9</b>	29%
4	<b>8</b>	26%
5	<b>3</b>	10%

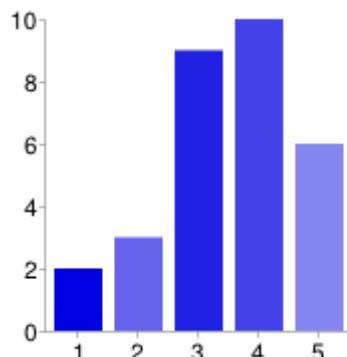
## Question 2

### A simple knock, like knocking on a door



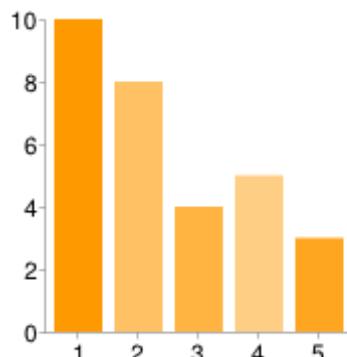
1	<b>5</b>	17%
2	<b>2</b>	7%
3	<b>10</b>	33%
4	<b>10</b>	33%
5	<b>3</b>	10%

### Write a personal message to the group



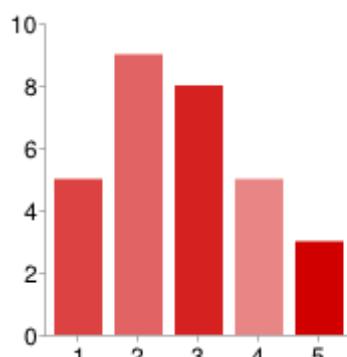
1	<b>2</b>	7%
2	<b>3</b>	10%
3	<b>9</b>	30%
4	<b>10</b>	33%
5	<b>6</b>	20%

### Record a personal voice message to the group



1	<b>10</b>	33%
2	<b>8</b>	27%
3	<b>4</b>	13%
4	<b>5</b>	17%
5	<b>3</b>	10%

### Just grab a chair, sit down and start talking



1	<b>5</b>	17%
2	<b>9</b>	30%
3	<b>8</b>	27%
4	<b>5</b>	17%
5	<b>3</b>	10%

### Other, please describe

Get their attention and talk from you seat, if its to far away, well then u cant hear them anyway never been to an online cafe

## Question 3

### A simple knock like knocking on the door



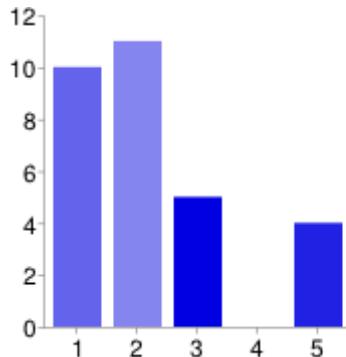
### A written message from the person



### A personal voice message from the person



### Just grab a chair, sit down and start talking



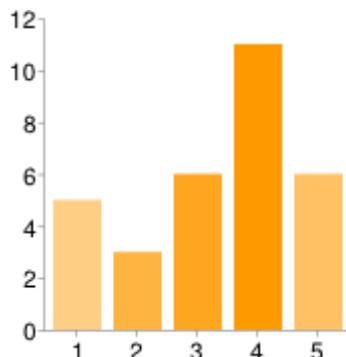
1	<b>10</b>	33%
2	<b>11</b>	37%
3	<b>5</b>	17%
4	<b>0</b>	0%
5	<b>4</b>	13%

### Other, please describe

What i would do in question 2 If I'm sitting at a table, I would want the option to give someone a cold shoulder. Knock, and if there is no rejection after 3 seconds, the person is "at the table"

## Question 4

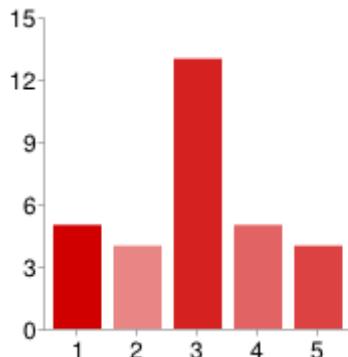
In a real café it can be hard to block out noise from the other tables, is this something you find disturbing?



1	<b>5</b>	16%
2	<b>3</b>	10%
3	<b>6</b>	19%
4	<b>11</b>	35%
5	<b>6</b>	19%

## Question 5

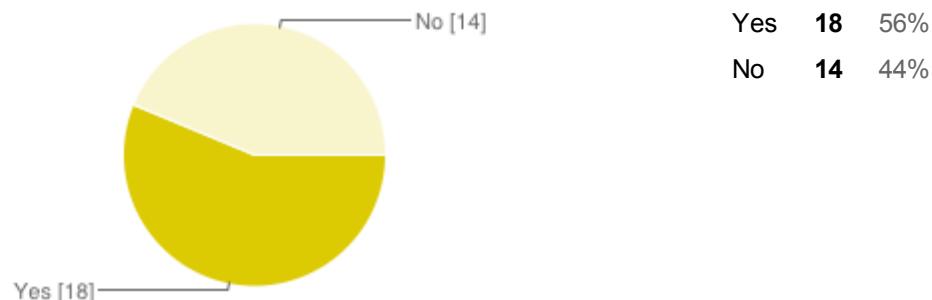
How would you feel about visiting an online café in a virtual 3D environment?



1	<b>5</b>	16%
2	<b>4</b>	13%
3	<b>13</b>	42%
4	<b>5</b>	16%
5	<b>4</b>	13%

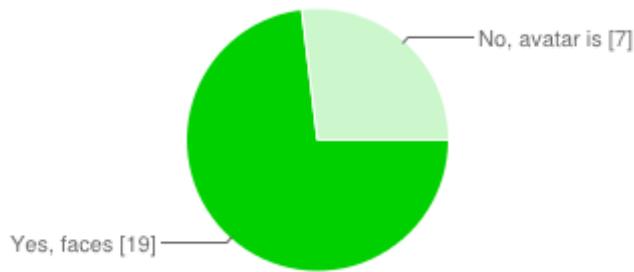
## Question 6

**Would you like the ability to maneuver a customizable avatar inside the café?**



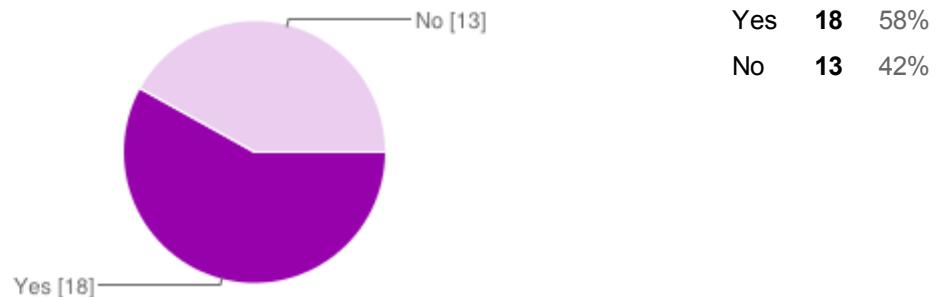
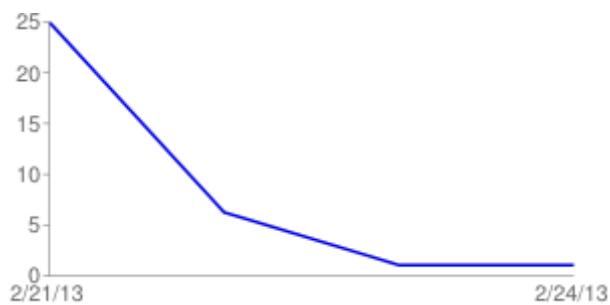
## Question 7

**If you're sitting down at a table with other participants, would it be important to see the faces of the other participants or would their avatars be sufficient?**



Yes, faces	<b>19</b>	73%
No, avatar is sufficient	<b>7</b>	27%

## Question 8

**Are gestures important for conversations online?****Number of daily responses**

## References

- [1] Project group, *Project blog*. May 2013, <http://mt-topc.blogspot.se/>
- [2] Google, *Google Hangout*. February 2013, <http://www.google.com/+/learnmore/hangouts/>
- [3] Microsoft, *Skype*. February 2013, <http://www.skype.com/intl/en/home>
- [4] ooVoo LLC, *ooVoo*. February 2013, <http://www.oovoo.com/home.aspx>
- [5] Linden Labs, inc, *Second Life*. February 2013, <http://www.secondlife.com/>
- [6] Thomas Erickson, N. Sadat Shami, Wendy A. Kellogg, David W. Levine, *Synchronous Interaction Among Hundreds: An Evaluation of a Conference in an Avatar-based Virtual Environment*. IBM T. J. Watson Research Center, Yorktown Heights, NY USA, May 2011, <http://dl.acm.org/citation.cfm?id=1979013>
- [7] Tuomas Kantonen, Charles Woodward, Neil Katz, *Mixed reality in virtual world teleconferencing*. VTT Technical Research Centre of Finland, March 2010, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5444792>
- [8] Google, *WebRTC*. May 2013, <http://www.webrtc.org/>
- [9] Google. *Hello Firefox this is Chrome calling*. February 2013, <http://blog.chromium.org/2013/02/hello-firefox-this-is-chrome-calling.html>
- [10] Team XSockets.NET. *Browser Meeting*. February 2013, <http://browsermeeting.com/>
- [11] frisB. *frisB*, February 2013. <http://www.frisb.com/>
- [12] Jerome Etienne, *WebGL Meeting*. October 2012, <http://learningthreejs.com/blog/2012/04/12/video-conference-on-top-of-webgl/>
- [13] Protonmedia, *Protosphere*. February 2013, <http://www.protonmedia.com/>

- [14] SAIC, *O.L.I.V.E.* February 2013,  
<http://www.saic.com/products/simulation/olive>
- [15] teleXLR8, *OpenQwaq*. February 2013, <http://telexlr8.net/openqwaq/>
- [16] Paltalk, *Paltalk*. May 2013, <http://www.paltalk.com/>
- [17] Tinychat Co, *Tinychat*. May 2013, <http://tinychat.com/>
- [18] The World Cafe Community, *The World Cafe Online Community*. February 2013, <http://www.theworldcafecommunity.org/>
- [19] Yusuke Ichikawa, Ken-ichi Okada, Giseok Jeong, Shunsuke Tanaka and Yutaka Matsushita, *MAJIC Videoconferencing System: Experiments, Evaluation and Improvement*. ECSCW'95 Proceedings of the fourth conference on European Conference on Computer-Supported Cooperative Work, Pages 279-292, September 2008. <http://dl.acm.org/citation.cfm?id=1241976>
- [20] Claudia Kuster, Tiberiu Popa, Jean-Charles Bazin, Craig Gotsman, Markus Gross, *Gaze Correction for Home Video Conferencing*. ETH Zurich, Technion - Israel Institute of Technology, November 2012. <http://ieeexplore.ieee.org.proxy.lib.ltu.se/stamp/stamp.jsp?tp=&arnumber=4811013>
- [21] David Roberts, Robin Wolff, John Rae, Anthony Steed, Rob Aspin, Moira McIntyre, Adriana Pena, Oyewole Oyekoya, and Will Steptoe, *Communicating Eye-gaze Across a Distance: Comparing an Eye-gaze enabled Immersive Collaborative Virtual Environment, Aligned Video Conferencing, and Being Together*. University of Salford, University of Roehampton, University College London, University of Reading, Lafayette, Louisiana, USA, March 2009. <http://ieeexplore.ieee.org.proxy.lib.ltu.se/stamp/stamp.jsp?tp=&arnumber=4811013>
- [22] Hideyuki Nakanishi, Yuki Murakami, Kei Kato, *Movable Cameras Enhance Social Telepresence in Media Spaces*. Department of Adaptive Machine Systems. Osaka University, Yamadaoka, Suita, Osaka, Japan, April 2009. [http://www.academia.edu/3274831/Movable\\_cameras\\_enhance\\_social\\_telepresence\\_in\\_media\\_spaces](http://www.academia.edu/3274831/Movable_cameras_enhance_social_telepresence_in_media_spaces)

- [23] Jim Gemmell, Kentaro Toyama, C. Lawrence Zitnick, Thomas Kang, Steven Seitz, *Gaze Awareness for Videoconferencing: A Software Approach* <http://ieeexplore.ieee.org.proxy.lib.ltu.se/stamp/stamp.jsp?tp=&arnumber=895152>
- [24] Istemı Ekin Akkus, Öznur Özkasap, M. Reha Civanlar *Peer-to-peer multipoint video conferencing with layered video* Journal of Network and Computer Applications, Volume 34 Issue 1, January, 2011, Academic Press Ltd. London, UK. <http://dl.acm.org/citation.cfm?id=1889430>
- [25] Chong Luo, Jiang Li, and Shipeng Li, *DigiMetro - An Application-Level Multicast System For Multi-Party Video Conferencing*. Microsoft Research Asia, 2004, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1378106>
- [26] O. Schreer, I. Feldmann, N. Atzpadin, P. Eisert, P. Kauff, H.J.W. Belt, *3DPRESENCE - A SYSTEM CONCEPT FOR MULTI-USER AND MULTI-PARTY IMMERSIVE 3D VIDEOCONFERENCING*. Fraunhofer Institute for Telecommunications/Heinrich-Hertz Institut, Berlin, Germany, November 2008. <http://ieeexplore.ieee.org.proxy.lib.ltu.se/stamp/stamp.jsp?tp=&arnumber=4778747>
- [27] Víctor Torres-Padrosa, Eusebi Calle, Jose L. Marzo, Mercè Rovira, *Towards a Mobile, Assistive and Intuitive Videoconferencing*. Broadband Communications and Distributed Systems Research Group Informatics and Applications Institute, Universitat de Girona (Spain), 2012. [http://www.thinkmind.org/download.php?articleid=etelemed\\_2012\\_1\\_50\\_40110](http://www.thinkmind.org/download.php?articleid=etelemed_2012_1_50_40110)
- [28] Joyent, Inc, *Node.js*. May 2013, <http://nodejs.org/>
- [29] Lynckia, *Licode*. May 2013, <http://lynckia.com/licode/>
- [30] 10gen, Inc, *MongoDB*. May 2013, <http://www.mongodb.org/>
- [31] LearnBoost, *Mongoose*. May 2013, <http://mongoosejs.com/>
- [32] Google, *Express*. May 2013, <http://expressjs.com/>

- [33] Twitter, *Bootstrap*. May 2013, <http://twitter.github.io/bootstrap/>
- [34] *Three.js*. May 2013, <http://threejs.org/>
- [35] GitHub, Inc, *GitHub*. May 2013, <https://github.com/>
- [36] Priologic Software Inc., *EasyRTC*. June 2013, <http://easyrtc-wp.com/prod/>
- [37] Ben Brittain, Dennis Mårtensson, David Peter, *WebRTC.io*. June 2013, <https://github.com/webRTC/webRTC.io>
- [38] Fractal, *Holla*. June 2013, <http://wearefractal.com/holla/>
- [39] Microsoft, *How much bandwidth does Skype need?*. May 2013, <https://support.skype.com/en/faq/FA1417/how-much-bandwidth-does-skype-need>
- [40] DailyMail, *Cafes ban Google glasses to protect customers' privacy: Fears users of futuristic eyewear can record without permission*. June 2013, <http://www.dailymail.co.uk/sciencetech/article-2323578/Cafes-ban-Google-glasses-protect-customers-privacy-Fears-users-futuristic-eyewear-record-permission.html>