POLITECNICO
MILANO 1863

Advanced User Interfaces
Extending Wearable Immersive Virtual Reality with AURAL Interaction

Aural Immersive VR

Authors:
Marocchi Fabio
Corradini Maurizio Martin
Cremona Davide

Tutors:
Pr. Garzotto Franca
Dr. Occhiuto Daniele

Version 1.0
February 23, 2017

Abstract — In Wearable Immersive Virtual Reality (WIVR), the only interaction paradigm today is limited to eye-tracking based interaction, or in rare cases touch interaction by means of remote controllers (e.g. oculus touch, htc vive). Why not exploring new forms of VR interaction by exploiting natural speech? This project aims to develop a new form of VR interaction, exploiting WebSpeech APIs and the A-Frame WebVR framework to extend the classical visual interaction in WIVR applications. The project also aims to use this new VR interaction to help nonverbal children in their therapy, with the implementation of a Treasure Hunt in VR that can assist therapists in their work.

The Team

0.1 Fabio Marocchi

I've been passionate about technology since I was a child and that passion led me to enroll to an Electronics and Telecomunication course in high scool.
It was only during my studies that I realized that I loved programming. That's why I quickly decided to continue my education with a Bachelor Degree in Software Engineering.
After that I worked for a summer as a programmer but I realized I wanted to learn more, and delve deeper into more advanced topics.

I then decided to enter the Computer Science and Software Engineering course at Politecnico di Milano and my passion continued to grow, especially towards topics related to Artificial Intelligence, Machine Learning and videogames design.

Contact: fabio.marocchi@mail.polimi.it

0.2 Maurizio Martin Corradini

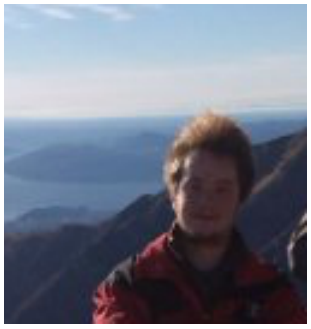
I was born in Caracas, Venezuela in 1992. Since 2003 I am living in Spain. My passion for the computers began when I was really young. Since I remember I have been always playing and using electronic devices so I knew that I wanted to study Computer Science very early.

I started my degree at "Universidad de La Laguna", Canary Islands, then I moved to Madrid to continue my career at "Universidad Complutense". I made my internship almost two years ago at EduWeb, an american software company with offices in Madrid, that distributes and supports a School Managment System. I'm

still working as Chief Programmer, in remote from Milano, while doing an Erasmus year experience.

I like to create Web pages and I want to learn more about the UX in order to improve the ways of interaction with the users.

0.3 Davide Cremona



Since the age of 12/13 years, I have developed the passion for Computer Science and this drove me to enroll in the J.M. Keynes (Computer Science High School) and then to the Politecnico di Milano both on Bachelor and Master of Science degrees on Computer Science and Engineering, focusing on Artificial Intelligence courses.

The interest in this field is given by the fact that I have always wanted to understand and to develop Artificial Intelligence algorithms that are more and more present in our life and in the Computer Science scene.

My experience have been enriched thanks to the development of various application (both for PCs and mobile) and web applications. My professional goal is to become a good Software Engineer and Computer Scientist in the fields of AI, Data Mining and Machine Learning.

You can find me at: davide.cremona@me.com

Contents

The Team **i**

0.1 Fabio Marocchi i

0.2 Maurizio Martin Corradini i

0.3 Davide Cremona ii

Introduction **1**

1 Target Groups and User Needs **3**

1.1 Main Target Groups 3

1.2 Context and Needs Addressed 4

1.3 Constraints 4

1.4 Goals 4

1.5 Requirements 5

2 State of the Art **7**

3 Solution - UX Design **9**

3.1 Approach 9

3.1.1 Interaction Paradigm 9

3.1.2 Content 9

3.2 Scenarios 10

3.2.1 Meta-Scenario: Story Description 10

3.2.2 Starting The Story 10

3.2.3 Treasure Hunt 10

3.2.4 A Difficult Clue 10
3.2.5 A Very Difficult Hunt 11

4 Solution - Implementation **13**
4.1 Hardware Architecture 13
4.2 Software Architecture 13
4.2.1 Components point of view 13
4.2.2 MVC point of view 17

5 Value Proposition **19**
5.1 Difficulties and Challanges 19
5.2 Is it a good Solution? 19
5.3 Is it the best Solution? 20

6 Future Work **21**

7 Bibliography **23**

Introduction

If we talk about Virtual Reality, the most prominent form of interaction is through eye-tracking, but in our attempt to improve the interaction possibilities, we decided to exploit speech APIs, complementing the classic visual interaction in Wearable Immersive Virtual Reality with natural speech.

One of the biggest challenge was understanding how to do it in a way that could help children with speech impairment improve their capabilities by practicing their pronounciation in a stimulating way. The answer we found for that is a classical "Treasure Hunt": the story of Roger, a Robot who is looking for his charger around the house with only a series of clues to help him.

Our main targets are therefore non verbal children and, more in general, children with speech difficulties. Our goal is to provide them with a tool that they can use and, hopefully, will want to use, to train their speech.

We decided to deploy this tool as a web application to take advantage of the fact that the users won't have to download and install any application and that the only required hardware are a smartphone with a gyroscope sensor and a cheap VR headset like Google Cardboard.

We created a Virtual Reality environment using A-Frame, and we populated it with 3D objects.
For the speech recognition, as well as for the speech synthesis, we used the Webkit Speech APIs implemented by Google Chrome, while for the semantic analysis of the recognized speech we used Api.ai, a conversational user experience platform.

The interaction between the components is provided through javascript code, that acts as a controller for the A-Frame virtual environment and the services APIs.

Chapter 1

Target Groups and User Needs

In this chapter, we will explain our analysis of the users and their needs, we will then present some constraints imposed by the technology and by the type of users; at the end we'll explain what goals the system needs to satisfy.

1.1 Main Target Groups

The system is intended to be used by **nonverbal children** that will use the application with the guidance of their therapists, family or tutors. Other **children with no speech impairment** can use the application to play and spend some time inside a virtual reality game. The use of this system in the context of nonverbal children therapy hardly modify the way **therapists**, **family** and **tutors** face their work. They also can help the future improvement of the application, by giving their opinion about what needs to be improved or modified.
So we can have this classification of stakeholders:

Primary	Nonverbal children
Secondary	Other children with no speech impairment
Tertiary	Therapists, Family, Tutors

Table 1.1: Classification of Stakeholders

1.2 Context and Needs Addressed

The context in which this application will be used is the therapy of nonverbal children. Knowing this, we have wondered what could be the Needs of our stakeholders. The first thing that comes into mind is that nonverbal children may need an instrument that can help them be more focused and involved during therapy sessions. Other children might need a tool that can be useful to help them in learning to speak and to read; they may also need a paricular type of game that can help them improving their concentration. Therapists may need a tool that helps them to work better with nonverbal children and make the work easier. In addition to this, we have found that the family or tutors may need an instrument that can be used also at home to be able to make their children play and train even outside therapy sessions. We can summarize the needs for each class of stakeholder with this table:

Nonverbal children	Instrument to improve concentration and speech capabilities
Other children	Learn to speak, Learn to read, Play/Improve concentration
Therapists	Make their work easier and more effective
Family/Tutors	Continuity of the therapy also at home

Table 1.2: Stakeholders Needs

1.3 Constraints

The first constraint is that the system must be portable and inexpensive, leading us to the use of a smartphone. This, in turn, gives us another constraint: the virtual reality world must be light enough to be rendered with the hardware of a smartphone. Furthermore, the system must be fast and responsive in the intepretation of vocal commands.
There are also compatibility constraints: we'll be able to use only technologies compatible with a smartphone. In our case we opted for the use of Webkit Speech Recognition APIs and A-Frame WebVR Framework, working on a browser (specifically Chrome).

1.4 Goals

The main goal of our system is to provide a new instrument that helps non verbal children to improve their speech capabilities and focus in a controlled environment on therapy sessions. The system can be used also to help children in learning to speak or reading using audio and visual feedback and speech recognition.

1.5 Requirements

To satisfy our targets needs our system must be able to:

- Listen to the user and recognize natural language
- Provide a virtual reality environment for the user to interact with
- Detect if the user is concentrating to specific areas within the VR environment
- Provide audio feedback to guide the user
- Let the therapist control some aspects of the configuration
- Implement an engaging story

Chapter 2

State of the Art

Virtual Reality has recently acquired a significant role in therapy treatment, mainly because it has become more and more inexpensive and available.
It's used in different fields with different goals. VR therapy is used with children and adults with NDD (Neuro Developmental Disorder), to help them learn how to behave in a variety of real-life situations and to increase focus capabilities in ADD (Attention Deficit Disorder) children. It's shown that the therapy is effective in these situations (Garzotto et al.).
Another application is for people with PTSD (Post Traumatic Stress Disorder), phobias and social anxiety, and it is used to let the user gradually grow comfortable with situations that would normally cause stress and anxiety, by introducing them in a controlled environment and without the possible risks associated to these situations in real life.
One of the social anxiety situations that are often faced in this second type of therapy, is stuttering related to the fear of public speaking. This is usually addressed by making the user speak in front of a virtual audience, and research has shown that this type of therapy can actually reduce the user's anxiety (Anderson et al.).
This type of treatment however requires either medical sensors to detect anxiety levels or self evaluation from the user, and are not focused on the speech capability itself but more on the anxiety point of view. Also there's no system that automatically detects if the user is speaking correctly, which is instead part of our goal.

This makes that kind of therapy inadequate for children with speech impairment and is the reason why in our project the main focus is instead on the spoken interaction with a conversational agent and not on the possible anxiety.

Chapter 3

Solution - UX Design

3.1 Approach

We addressed the problem thinking about who will mainly use the application (nonverbal children) and trying to find an attractive solution for them. We started by imagining the possible situations our users will face; then we have searched for the right interaction paradigm and contents that can help with the problem.

3.1.1 Interaction Paradigm

The users can interact with the application by wearing a VR visor (such as Google Cardboard) and visiting the application webpage. After that they can use vocal commands to proceed in the story. The application will also react to where the children are looking and what they're focusing on inside the environment. They will also receive visual and audio feedbacks to guide them and reward them.

3.1.2 Content

The content implemented is inspired by social stories given by the therapists. We have used the same simple style of narration to implement a Treasure Hunt configurable by the therapist. The story is about a robot (Roger) that has lost his charger. The child has to help him find his charger by searching for objects to solve the treasure hunt.

3.2 Scenarios

Here we present some scenarios that describe how the verbal interaction between the child (user) and Roger (the virtual agent) works.

3.2.1 Meta-Scenario: Story Description

The system implements a treasure hunt game guided by Roger the robot. The child has to answer Roger's questions and find answers to the clues that they find around the house, one after the other, in order to find Roger's precious charger.

3.2.2 Starting The Story

At the start, the Therapist can adjust some settings (objects in the hunt and their order). Marco wears the cardboard and the game starts. Roger the robot asks Marco: "My friend hid my charger, can you help me find it?" Marco replies "Yes!". Roger says: "Thank you! I found this clue that could help us locate my charger, can you read?" Marco replies:"No, I can't". Roger:"Ok then, let me read it for you".

3.2.3 Treasure Hunt

Annachiara is a child that can read easily. She's using our system to train to speak. After the beginning of the treasure hunt, she finds the first clue and Roger (the virtual agent) asks to Annachiara to read it. After she has read the clue, Roger interacts with her and asks: "good, now can you tell me what is it?". The child wants to help him so she reply. After that Roger asks: "perfect! can you help me find it? look around and tell me when you found it". Annachiara starts to look around and when she's looking at the right object she says: "I found the object!". Since she has found the object Roger is happy and replies: "very good! hey look, there's another clue hidden here! can you read it for me?" and the cycle repeats.

3.2.4 A Difficult Clue

Luca finished reading the clue and Roger asked if he understood it. Luca doesn't know what the answer is, so he says: "No" and Roger reply: "I'll give you another clue: it's black". Luca looks around and he sees a black cat and says: "It's a cat!". Roger says: "Very good, find it and tell me when you're looking at it". Luca is looking at the cat and says: "I found it". Roger replies: "Perfect! look there's another clue hidden behind the cat".

3.2.5 A Very Difficult Hunt

Giulio is a child that is using our system to train in speaking with the help of his therapist. He's searching for a key that is needed to proceed with the treasure hunt. After a certain amount of time he has not found the key and the system helps him by animating or illuminating the keys. But this is not enough so Roger helps him by saying "Hey Giulio I think they are on the table, aren't they?". Finally Giulio finds the keys and the story can continue.

Chapter 4

Solution - Implementation

4.1 Hardware Architecture

There isn't a real Hardware architecture, since the project is web based and it consists on a Web Application deployed on GitHub. Nevertheless, we can say that the user will need VR visor such as Google Cardboard and Chrome installed on a Smartphone to use the Web Application. The smartphone need also to have a gyroscope sensor, microphone and speakers to be used with the web application, in order to be able to explore the virtual reality world, receive audio feedback and send vocal commands.

4.2 Software Architecture

We can see the software structure in two ways: one under the point of view of the components and one under the point of view of the Model-View-Controller (MVC) design pattern.

4.2.1 Components point of view

We can say that the system is divided in two main components:

1. The **Web Application** is the core of the system, it renders objects in the virtual world, records vocal commands and contains the code to make the world reactive to the actions of the user. The control is distributed in two modules:

    - `api-controller.js` is responsible for handling the speech recognition and speech synthesis in order to provide a simpler interface to the actual controller. It also

handles the requests to the API.ai agent and dispatches its responses to the controller.

- `game-controller.js` is in charge of the main control flow of the application. It's the component that actually activates the speech recognition and, after receiving the outcome it sends it to API.ai.
  It then reacts to the response of the agent according to the current state, by showing/hiding the clues, highlighting the objects in the environment and call api-controller to synthesize the textual response.
  The detail of the flow of the interaction can be seen in the following image.

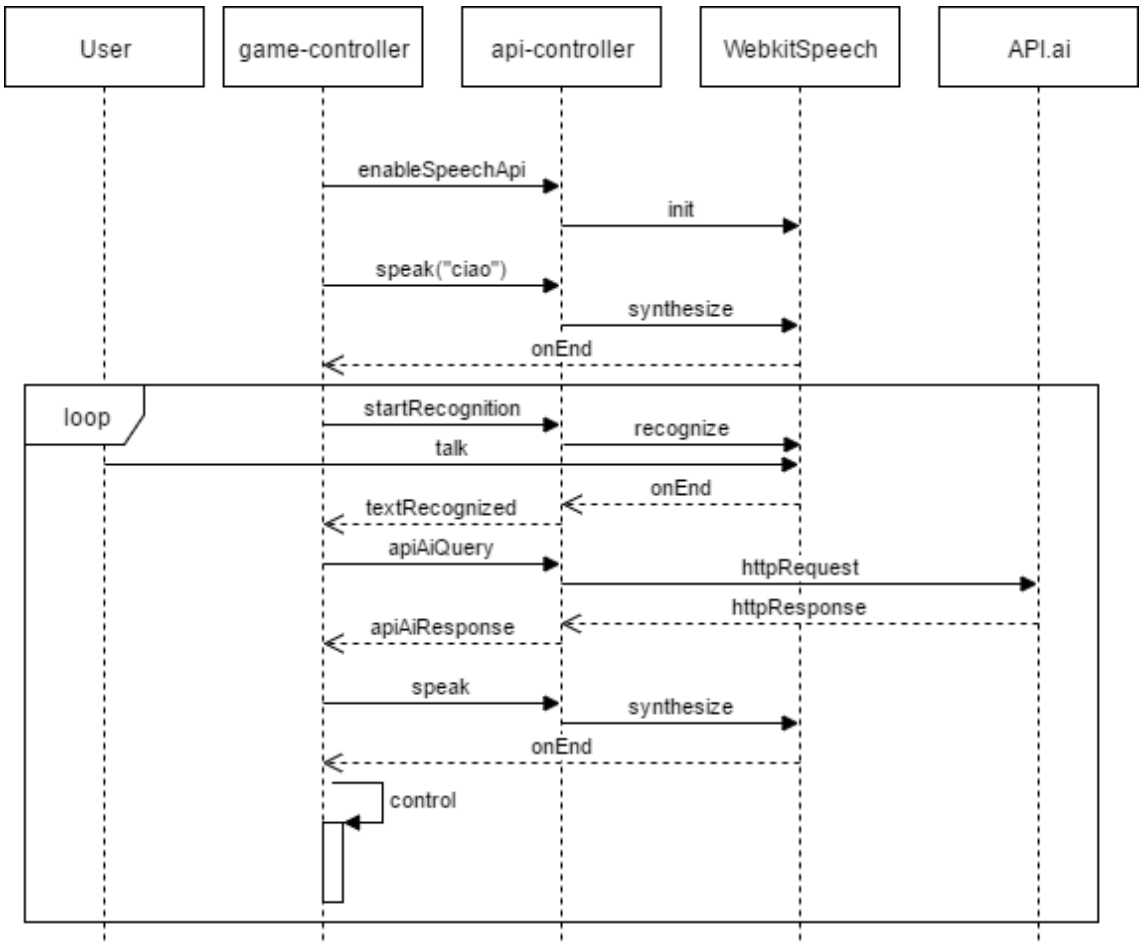


Figure 4.1: Main application interaction flow

Inside the control function the system reacts to the response of the Agent by applying changes to the environment or to the interaction flow itself according to the action type recognized by the Agent:

- `can_read` : remember if the child is able to read

- `get_clue` : display the clue on a piece of paper and if the child can read, wait for the speech (if the child doesn't read the clue, the system will read the clue and continue as if the child was unable to read). Otherwise read the clue for the child and wait for the solution (if the child doesn't solve the clue after a certain time the system will provide an additional suggestion).
- `read_clue` : the child has just read the clue so the system behaves as above
- `finding_object` : the child has solved the clue and is now looking for the object. The piece of paper is now hidden and the child has to focus on the object (after a certain amount of time a visual clue will be provided in the form of a rapid scaling up and down).
- `object_found` : the child has found the object and from here the system can go either in the final state, or select the next object in the list and start over.
- `end_game` : after the last object has been found roger will thank the child for the help and the application will terminate.

2. The **API.ai Agent** is in charge to decode the requests made by the Web Application and reply to them with an appropriate response message, built depending on the current advancement of the Treasure Hunt. An API.ai agent is composed by Intents. Every intent activates when the user says some specific words and the Agent is in a specific Context. Here it's described the state machine of the Agent (each node is a Context and each arrow is an event that triggers an Intent that changes the Context):

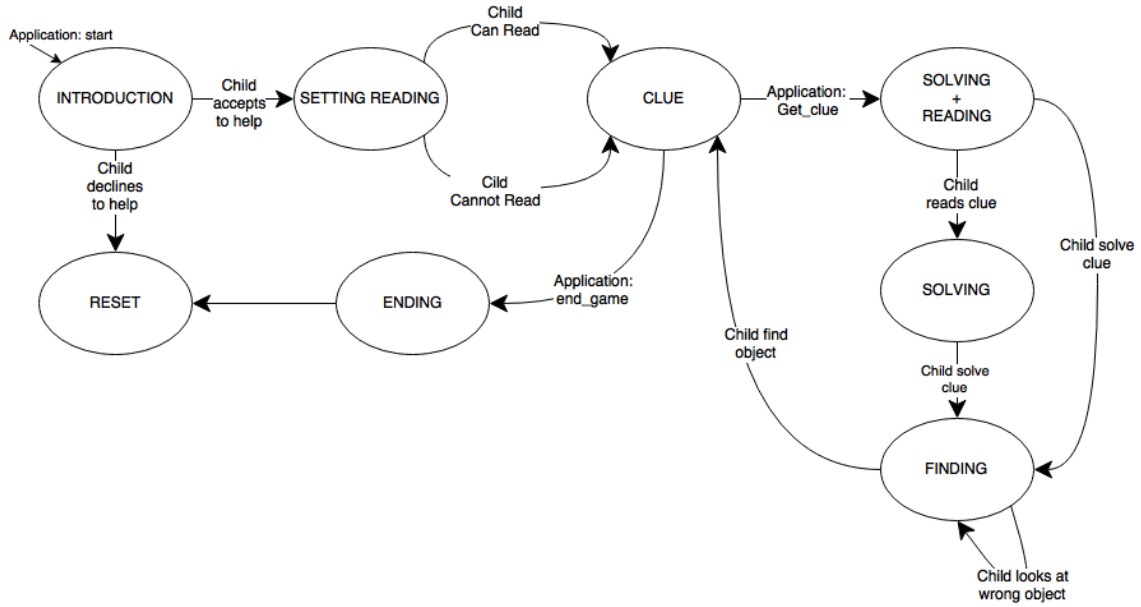


Figure 4.2: Agent Context State Machine

Our Agent is composed by two types of Intents: Global Intents and Object-Related Intents.

Global Intents

These are intents that are not related to a particular object of the treasure hunt. They are in charge to process general commands such as the start command. They are:

- **"TreasureHunt_intro"** it activates when the child salutes the Robot, starting the application. It takes the Agent in the "Introduction" Context meaning that the Web Application is introducing the child to the Treasure Hunt.
- **"TreasureHunt_declination"** it activates then the child says that he will not help the Robot. The Agent resets.
- **"TreasureHunt_settings"** it activates when the child says that he will help the Robot to find his lost charger. This Intent is called to ask the child if he can read or not. It takes the Agent in the "Settings" state meaning that the child has to say if he can read the clues or not.
- **"TreasureHunt_readok"** is activated if the child can read, it takes the Agent in the "Clue" state meaning that he's waiting a request for the next clue from the application.
- **"TreasureHunt_readko"** is activated id the child cannot read, it takes the agent in the "Clue" state like the previous intent.
- **"Object_found"** that activates when the child says that he has found the object, it takes the Agent in the "Clue" state, meaning that the application must ask for a new clue to continue the treasure hunt. It also rewards vocally the child.
- **"Looking_wrong_object"** that activates when the application says that the child is looking at the wrong object. The response of this intent is a phrase that says that the child is looking at the wrong object. It starts and ends with the "Finding" Context.
- **"TreasureHunt_end"** it activates when the application sends the "End" command and it resets the Agent.
- **"Default_Fallback_Intent"** this particular intent activates every time there is no other intent to activate. It says to the child that he has not understood what he's saying.

Object-Related Intents

These are intents that are related to a specific object of the treasure hunt. For each object we have:

- **"[id]_getclue"** intent that communicates to the Web Application the clue and the suggestions for the object with id "[id]", it takes the Agent in the "Solving" and "Reading" states that indicates that the child is either solving the clue or he is reading it.
- **"[id]_readclue"** that is activated only if the child reads the clue, it changes the Context in "Solving" meaning that the child is solving the clue.
- **"[id]_solving"** that id activated only if the child solves the clue saying the name of the object. It changes the Context from "Solving" to "Finding" meaning that the child is searching for the object in the VR world.

4.2.2 MVC point of view

The system can also be seen as a collection of components that compose the Model View Controller design Pattern.

1. The **Model** contains all the 3D models that are rendered in the VR world and the collection of intents of the API.ai Agent. They only describe the behaviour of what is present in the VR world.
2. The **View** is represented by the webapp.html page of the Treasure Hunt. This page is only in charge to render objects of the real world using A-Frame and to visualize their changes.
3. The **Controller** contains all the logic, the communication between the web pages and the controller.js that is in charge to make requests to API.ai and handle responses to change the state of the VR world.

The application is written in HTML/Javascript with the use of the A-Frame framework to build up the VR world. Speech recognition and Text-To-Speech is implemented using the Webkit Speech Recognition APIs. The semantic analysis of the vocal commands is made by the API.ai Agent (that is a black-box, so we cannot say how it's implemented).

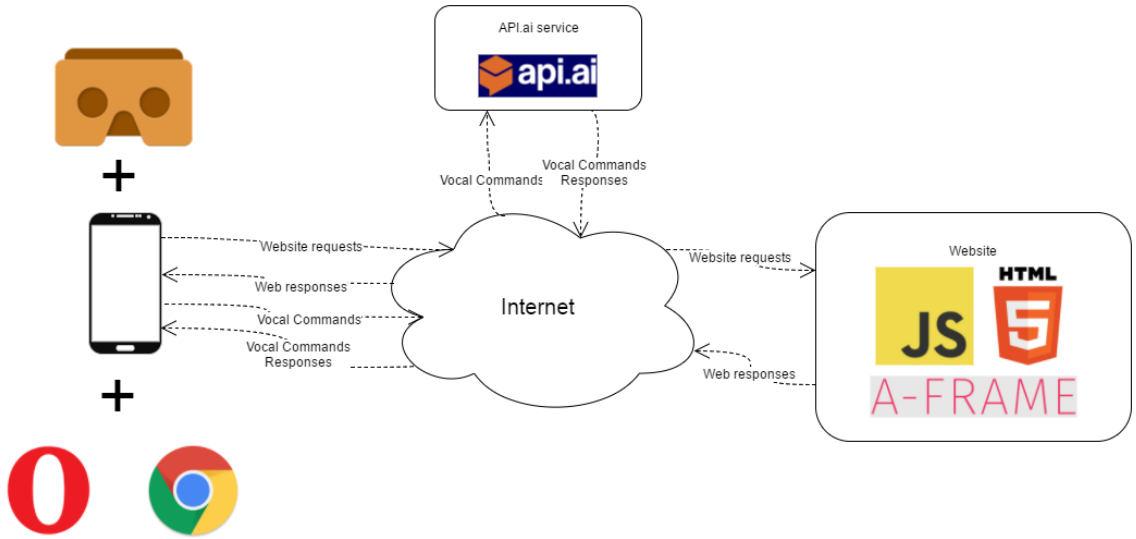


Figure 4.3: Overall view of the system

Chapter 5

Value Proposition

Here we present a critical reflection of our work, trying to identify the main challanges, difficulties and points of strenght.

5.1 Difficulties and Challanges

We have encountered various challenges, but the most important is certainly the fact that we had to model our system to be usable by a child with verbal difficulties and to give the right feedback to have an interaction that is acceptable from the children point of view. Another difficulty we found is that the system must be modular: the therapists must be able to change the treasure hunt according to the situations. This is feasible, but we have to constrain it to a predefined set of objects that can be put into the treasure hunt because the API.ai Agent needs to be expanded and a new 3D model for the VR environment must be developed for each new object that we want to insert.

5.2 Is it a good Solution?

We think that our system is a good solution as the virtual world can help the children focus on his task and training their speech inside a game, can help them be more motivated. We also think that this is a good way to face the problem of nonverbal children because since the interaction is with a virtual agent, they don't need to think about the personal relation that they might have with real people and can focus solely on the problem.

5.3 Is it the best Solution?

There is a lot of research going on in the use of VR to face the problem of nonverbal/ASD children. Our system is a simple solution that can be the start point of something bigger. As it is now it's not polished as a commercial product would be, but it's certainly a start towards new interaction paradigms.

Chapter 6

Future Work

Here we want to examine what the future development of this work could be, trying to imagine what can be done to improve the performance and the utility of our application.

- **Statistics**: to help the research and the work of therapists, the system can be expanded adding statistics about the focus time on particular objects, time to end the treasure hunt, most difficult objects etc...
- **Personalization**: the system can be personalized with a login for each children to allow the implementation of a sort of automatic system that learns the main difficulties of the user and can set timers, objects and their position in the virtual world according to these data.
- **More Modularity**: the modularity could bu further increased by building a configuration tool to automatically update the API.ai agent when a new object is added to the treasure hunt object collection. Up to now there are no public API to automatically update and modify the agent but maybe in future API.ai will add this feature to its agents.
- **Improve Compatibility**: the system compatibility can be improved or simply modified by using a different type of VR framework or by changing the Text-To-Speech/Speech-To-Text APIs.

Chapter 7

Bibliography

- GitHub repository: `https://github.com/fmarocchi92/AUI1617`
- Web Application: `https://fmarocchi92.github.io/AUI1617/`
- API.ai Platform: `https://api.ai`
- A-Frame WebVR Framework: `https://aframe.io`
- Webkit Speech APIs: `https://dvcs.w3.org/hg/speech-api/raw-file/tip/speechapi.html`
- Virtual reality exposure therapy for social anxiety disorder: a randomized controlled trial - Anderson PL, Price M, Edwards SM, Obasaju MA, Schmertz SK, Zimand E, Calamaras MR.
- Wearable Immersive Storytelling for Disabled Children - Garzotto F, Gelsomini M, Clasadonte F, Montesano D, Occhiuto D.