HelpMe!: A Serious Game for Rehabilitation of Children Affected by CVI

Matteo Ciman[1], Ombretta Gaggi[1,3], Laura Nota[2,3], Luisa Pinello[4], Nicola Riparelli[1]
and Teresa Maria Sgaramella[2,3]

[1]*Dept. of Mathematics, University of Padua, Padua, Italy*
[2]*Dept. of Developmental Psychology and Socialisation, University of Padua, Padua, Italy*
[3]*University Center for Research and Services on Disability, Rehabilitation and Inclusion, University of Padua, Padua, Italy*
[4]*Pediatric Low Vision Center of the Veneto Region, Dept. of Women's and Children's Healths, University of Padua, Padua, Italy*
{*mciman, gaggi*}*@math.unipd.it*, {*laura.nota, luisa.pinello*}*@unipd.it*

Keywords: Serious Games, Game-based Rehabilitation, HTML5, Web Technologies.

Abstract: CVI (Cerebral Visual Impairment) is the leading cause of visual impairment in the United States and the Western Europe. Due to the high number of different inabilities that children could have when affected by CVI, it is crucial for the rehabilitation process to start from a good assessment, especially at early ages. In this paper we present *HelpMe!*, a serious game to improve the rehabilitation process for these children with a system able to adapt the exercises to each particular child and to his/her improvements. The system integrates an eye tracker system to correctly measure the performances of the child and his/her capability to watch and touch a moving object at the same time.

1 INTRODUCTION

Cerebral Visual Impairment (CVI)[1] is a disability that entails a visual deficit, due to a brain damage (Roman-Lantzy, 2007). People affected by this disability need that an object is moving to be able to see it, have a reduced ocular field and ocular delay, and they find difficulties to understand complex images. Moreover they are not able to see and touch an object at the same time: they usually watch the object first, and then they try to touch it, often loosing the eye contact. A child affected by CVI can experience all (or a set of) this difficulties, with different level of graveness. In addition, since this disability is originated into the brain, these children could often experience other disorders, i.e. motor disabilities.

As shown by Malkowicz at al., an early intervention can increase the possibility for children to drastically reduce the effects of this disability (Malkowicz et al., 2006). For this reason, it is very important to identify this problem at early ages, to perform a diagnosis and a personalized rehabilitation program.

A good diagnosis can be achieved only with the total collaboration and attention of the patient. It is not easy to capture children attention during a visual acuity test, because test exercises (like Lea symbols) are extremely boring and children usually do not pay much attention to the questions of the doctor and so to the answers they give. As a result, the diagnosis may become inaccurate.

In this paper we propose a game for a personalized rehabilitation of children affected by CVI. We propose the *serious games* paradigm to get attention and a touch interface for a more natural interaction with the system. The use of a *serious game* allows to obtain as much attention as possible from the child and to reduce the *drop-out-from-therapy* phenomenon: the more the child has fun, the more attention he/she will pay to the game-exercises and he/she will spend much more time doing the rehabilitation exercises.

The game interface must be *fluid* to adapt itself to different situations: the child could play the game using different devices, e.g., a tablet, a computer with or without a touch monitor, etc. The game must work even in absence of the network connection. Finally the game must be configurable to adapt itself to any child with different difficulties, and must evolve together with the improvement of the child.

The paper is organized as follows: Section 2 discusses background and related works. The usage sce-

[1]Also known as *Cortical Visual Impairment.*

narios and the developed game are presented in Section 3 and Section 4. The architecture of the system is defined in Section 5. Section 6 discusses implementation issues. Finally, we conclude in Section 7.

2 RELATED WORKS

When we talk about *serious games*, we talk about games or applications which are developed not only for fun, but to hide, under the games, some exercises to develop particular skills of the user. Actually there are a lot of different applications of *serious games*, e. g., military field, educational field or governmental field (Michael and Chen, 2006).

Serious games are becoming very important, either for doctors, e. g., for their training or to simulate real-life experiences, and for patients, e. g., to hide rehabilitative exercises under a game. Esteban at al. developed a system to combine 3D computer simulation to the learning process for new doctors, to teach them particular procedures (Esteban et al., 2011).

Other approaches show the goodness of the usage of *serious games* with children. De Bortoli and Gaggi showed how a visual acuity test can be hidden under a much more interesting game, with the consequence that children pay much more attention to the exercises, and so to the answer for the doctor, with the result that the diagnosis will be more accurate (De Bortoli and Gaggi, 2011).

Other authors ((Di Loreto and Gouaich, 2011), (Gaggi et al., 2012)) showed that not expensive equipment and *serious games* can open opportunities for home rehabilitation, reducing the *drop-out-from-therapy* phenomenon, even with children, and for rehabilitation programs that could last several months.

To the best of our knowledge, the first attempt to use a game for rehabilitation of children affected by CVI was carried out by Laura Campaña of the Junior Blind of America. She has shown, that if traditional rehabilitation tools are replaced by iPads with some simple games, even without any specific goal and which are not created for this particular kind of children (i. e., *Bubbles Magic*) that present some words or pictures and play different sounds, the result is that the child is attracted by the pictures and the sounds on the device, he/she pays lot of attention to the tablet and to what happens on the screen (Campaña, 2012). *Tap-n-see Zoo* is an application specific for children with CVI which moves a teddy bear on the screen and plays a sound when the child taps on it.

Our solution improves this first approach since it integrates the serious game with an eye-tracker system to record the child's eyes movement during the rehabilitation program and to better measure his/her improvements and it proposes a game to train a specific skill: the *problem-solving* capabilities of the child.

3 USAGE SCENARIOS

Our goal is to train the ability to see and touch an object at the same time. This kind of rehabilitation requires a long period of training, which must be performed every day, therefore not only in the hospital, but also at child's home.

To measure if the training process has a positive result, the doctor needs a precise way to measure the performance of the children during all the rehabilitation program. For this reason, we need to be able to acquire as much data as possible about the behavior of the child, therefore our system uses an eye-tracker system to record the eyes gaze during all the game session. Two problems arise: the eye tracker system costs, therefore we cannot assume that every child has it at home. Moreover, it requires a sufficiently illuminated environment with a neutral background.

Since the rehabilitation program, for being effective, needs to be performed almost every day, we must allow the child to play the game with any device. For this reason,the application has to be *portable* over several platforms and devices, e.g. with different screen resolutions, and must support different input methods, e.g. touch interfaces and standard input interfaces. Further, the system has to be *robust* to the absence of a component, i.e., the eye tracker system, in order to reduce the minimum cost of the needed equipment, and to the absence of Internet connectivity, to allow children to play the game even with a tablet. Even in this cases, the system has to collect data, though incomplete, but still sufficient for further analysis.

For this reasons, our system was developed to be *adaptable* to any possible scenario, without the need to develop different specific-scenario applications. As well known, web interfaces are the best candidates to ensure the highest portability over different devices.

4 GAME DESCRIPTION

The developed game was called *Help Me!* and has two goals: to train the ability to see and touch an object at the same time and to train the *problem-solving* capabilities of the child.

The location of the game is the North Pole, in particular at Santa Claus' laboratory. The child has to help a pixie to fill, on time, the bag of Santa Claus with proper gifts. Each level of the game has a different family of objects as *target*, e. g., “musical instruments”, “clothes for dolls”, “cars” and so on, each of which is composed by a set of different images that show objects belonging to that particular family.

HelpMe! proposes to the child a set of images; the child has to discriminate between *target images*, i. e. images that belong to the *target family*, and *distractors*, i. e., images which belong to other families, putting the firsts inside the Santa Claus' bag, and throwing out of the screen the last ones. Therefore, the main tasks that the child has to complete are to understand what is presented on the screen, to decide the operation to perform and to move the object in the right position, i. e., in the bag or out of the screen.

The game is challenging for children affected by CVI because the first operation requires the child to be able to focus on an object, the second one involves his/her problem-solving capabilities, and the last operation requires the child to be able to see and touch an object at the same time. To avoid that the child becomes more interested on the background image instead of the objects presented on the screen, we try to keep the background image as easier as possible, avoiding the usage of lot of details.

Every level is described inside the system by a *target family*, the number of *target images* and *distractors* which compose that level and the maximum amount of time (in seconds) the child can use to give an answer. The system allows to configure every level with a sequence of images, both target and distractors, organized according to a specific pattern defined by the doctor and to adapt these pattern for each child.

Furthermore, the system provides a lot of sounds and audio to guide the child during the game, i.e. to give the first instructions, to tell to the child which is the *target family* of each level, to provide live feedback about child behavior and choices etc.

If the child fails to discriminate one of the images of a particular level, the game asks to the child to repeat that level. After three failures in the same level, the game proposes the next level anyway.

5 ARCHITECTURE OF THE SYSTEM

The design of the system architecture has been deeply influenced by the requirement, discussed in Section 3, that the system can be used in very different scenarios, and has to transform its user interfaces gracefully, in order to adapt itself to the different characteristics of each environment and to remain usable by all the children with any kind of devices. We must note here that the system does not need to provide all its features in all situations, but it must be *robust* to the absence of some component, e. g., even if the eye tracker is not available when a child is at home, the system must provide the possibility to play the game even with the child computer or tablet. In this case the system will record only the touch interactions of the child and not the movement of the eyes.

For this reason, the system has been structured in four different components, following a client-server architecture. In particular our system includes:

1. a *server*, which manages, synchronizes and stores data produced from the other components;
2. an *eye-tracker system* that produces information relative to eyes position;
3. the *user interface for the children*, i. e., a component responsible to present the game and manage child interactions;
4. the *user interface for the doctor*, i. e., a component which is able to present data collected on the child performances. Moreover, it gives to the doctor the possibility to change the game behavior to better adapt it to a specific patient.

All the components have been developed as much independent as possible, so that they can be removed, or replaced in the future. This is particular important both to adapt the system to particular situations, e. g., the absence of the eye tracker system or if the child uses a tablet instead of a computer to perform his/her rehabilitation exercises, and to allow an easy reconfiguration of the system. As an example, we use an open source eye tracker system. Commercially available eye tracker systems are more accurate but also more expensive. If the budget allows to buy one of this system, it can be easily integrated in our system.

The server component is the more important component of the system. It must be accessible through the net in order to receive and manage the packets from the other components. It even stores several information, like child performances or game settings.

The eye-tracker system is responsible to manage all the tasks necessary to determine where the child is watching on the screen during the exercises. The system is composed by a ThorLab web cam (Thorlabs, 2012), two infrared lamps and a software developer by the ITU GazeGroup from Denmark (ITU GazeGroup, 2012) that analyzes the information that comes from the camera.

Before the tracking process, the software requires a calibration step in which the system asks the child to focus on some specific points of the screen, defined by the software itself. We use some funny pictures, instead of a red dot in the original software, to maintain

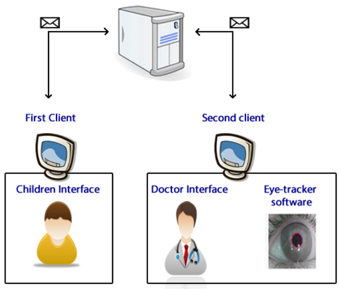


Figure 1: System architecture implemented for *HelpMe!*

the serious game paradigm also in this phase.

Users interfaces are managed by two specific components: a client for the child to play the game, and one for the doctor to manage the game settings and to provide online information about children behavior.

Given the particular type of users, children affected by CVI, the choice of the interaction modality is very important. As shown by Forlines at al., the touch interfaces are a really natural paradigm for children; indeed they interact with the screen as they would interact with a real object (Forlines et al., 2007). For this reason, the interaction method preferred for the children interface component is a touch interface, to better involve the children to the game and naturalize the interaction with it, through a tablet or a touch monitor. However, it is possible to interact with a mouse if a touch device is not available.

6 IMPLEMENTATION OF THE SYSTEM

As discussed in Section 3, to address *portability*, the game and the child interface must be usable on any device. For this reason, the application has been developed as a *Rich Internet Application (RIA)*, a web application that works with lots of data elaborated both by the client and the server, exchanged in an asynchronous way, with a *look and feel* similar to desktop application.

Our application has been developed using the new HTML5 standard, that does not requires to download a specific plugin, but only to use a browser that supports it. Unfortunately, not all the versions of the browsers support HTML5 standard, but its support is widespread over the actual browsers [2].

Another requirement for this system was to take up little space, so that it can be used in a doctor's office. For this reason, we managed to use the minimum number of computers and other devices. The final architecture of the system is composed, when all the components are available, by the server and two computers, one dedicated only for the child (which could be replaced by a tablet) and one for the doctor interface and the eye-tracker system (the last one will work in background), as shown in Figure 1. We cannot use only one device, because, even if we do not consider performances degradation, this solution would require to the doctor and the child to compete to get the focus on their own window to interact with the proper software interface.

6.1 Communication Protocol

The client-server communication is based on a simple protocol of packets exchanging encoded into the JSON format. Every packet is composed by a "type" field, which specifies which is the meaning of that packet, e. g., "ready_to_play" or "game_settings", and other fields, that depends on the "type" field, e. g., "data", "time", "pos_top" or "pos_left", that contain the real information of that packet.

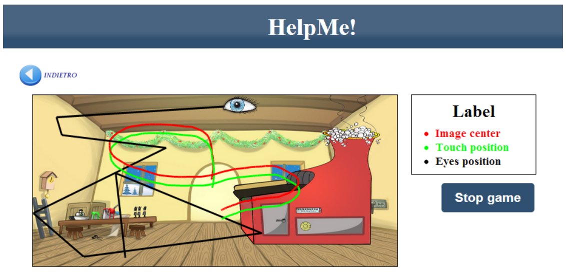
Communication between the clients and the server has been developed using *WebSocket API*, a new feature of HTML5 which allows to defines a new communication protocol that creates a full-duplex single socket connection between server and client (IETF, Internet Engineering Task Force, 2012).

The *WebSocket* protocol is implemented, at client side, by the browser, therefore, we only needed to develop a server. At the time of the development of this project, a only Java implementation of the server supported the last specification of the WebSocket handshake and the WebSocket protocol. For this reason, the server component was written in Java 7.

The usage of the WebSocket protocol provides several improvements in real-time application performances. Firstly, it reduces the amount of overhead introduced in each information packet, reducing the throughput necessary to send all the packets. Secondly, it reduces latency, because after the initial handshake between the client and the server, every time the server has new data for the client it can send it immediately, without waiting for a new request from the client (like the *polling* technique).

The use of WebSocket allows to provide to the doctor a real-time feedback about the behavior of the child, with information packets sent every 40 ms (limited by eye-tracker performances). Figure 2 shows the doctor interface: the server receives, records and sends to the doctor client the movements of the image (the red line), of the user touch (the light green line)

[2] Internet Explorer v. 9 (current 10), Chrome v. 10 (current 23), Firefox v. 3.6 (current 17), Safari v. 5 (current 6) and Opera v. 11.1 (current 12.1)



Object	Target Family	First Response Time (ms)	Completion Time (ms)	Correct / Wrong
Resume	**animali**	**2577**	**12340**	**1/0**
Gatto	animali	2577	12340	true

Figure 2: Simple feedback of child interaction with the system provided to the doctor in real-time way

and of the gaze (the black line) along the time. In this way, the doctor is able to watch what is happening on the child side, viewing where the child is touching, how he/she is moving the image on the screen, and where he/she is watching. Furthermore, we provide several summary information about child choices and interaction with the game.

6.2 Data Storage

The doctor can consult data during the child exercises or offline, for further analysis. For each session game a *.ini* file stores information about the used device, the screen size, etc. Information about the image position on the screen, the touch interaction of the child and where the eyes are watching on the screen are stored into three different text files, in which each entry, i. e., a packet, follows the pattern $\langle \Delta T, left, top \rangle$ where ΔT is the timestamp, and *left* and *top* contains the left position and the top position of the recorded element (the touch position, the center of the image or where the eyes are watching) on the screen.

Even the exercises settings used by the doctor for each child are stored in the server with an XML file (every child has his own settings associated), providing the possibility to the doctor to use several times the same exercise with the same child, avoiding him to insert every time the same settings (this is particular important for performances analysis for the doctor).

6.3 Portability Issues

To improve the portability of the system, we use other two important features of HTML5: *local storage* and *cache manifest*. When a *cache manifest* file is associated to an HTML page, the browser reads this files and downloads all the files listed in its cache, that are all the files necessary to provide the requested page. In this way, we have a performance improvement (the browser does not need to download all the file every time) and the browser can provide the application even without Internet connectivity.

The Internet connectivity is necessary only to send to the server the information stored during the rehabilitation made at home. The server receives this information, saves them and calculates an evaluation of each exercise. All the information are stored in the database to provide to the doctor the possibility to watch a simulation of the rehabilitation program performed by the user and his/her progress.

At the current state of art, browsers available on tablet are not able to reproduce mode than one audio at the same time, for this reason, we used the *PhoneGap* (Adobe Systems, 2012) framework to carry out the conversion from a web application to a native application for any available tablet.

The application has been successfully tested with Chrome from version 14.0, Firefox from version 11.0, Safari from version 5.0 and Opera from version 12.0.

6.4 Synchronization of the Components

The goal of the system is to train the child and to provide information useful to the doctor. This information are: the image position, where the child is touching and where he/she is watching on the screen. This information comes from two different sources: the first two from the client dedicated to the child interaction, and the last one from the eye-tracker component. Therefore, the main task of the server is to synchronize and store this information during the session game, in order to avoid loss of data.

To align all the packets along a common time line, every packet is timestamped with a value ΔT, representing the offset with referent to a common initial time zero which represents the beginning of the game or the beginning of the eye-gaze tracking.

Due to the natural difference between clocks in different autonomous computers, the starting time for the game and the eye-tracking software will be different. But this different times have to represent the same UCT time, in order to start their operation at the same moment. In this way, if two packets have the same ΔT (or with a difference less than 40ms) the information provided are relative to the same moment.

The server calculates the starting time for every session of the game using its own clock (basically, five seconds after the entire system is ready to start). Then it converts the starting time to the corresponding time for the each client. At this point, the two clients can start their job at the time received by the server. Every clients calculate the ΔT values based on the start time value received by the server.

The algorithm used to calculate the starting times for the two clients is based on the work by Sichitiu and Veerarittiphan, that defines the time of a client as a linear function of the server time based on two parameters, previously calculated during a synchronization phase where lots of packets that contains time information are exchanged between server and client (Sichitiu and Veerarittiphan, 2003). We use 500 packet for the estimation of the *Round Trip Time (RTT)*. This synchronization step usually requires less then a minute[3] and must be performed only once for each client, the first time it is connected to the system.

7 CONCLUSIONS AND FUTURE WORKS

This paper presents a system for rehabilitation of children affected by CVI with a serious game, called *HelpMe!*. The game aims at training the problem solving capabilities of these children and their ability to watch, touch and move an object at the same time. To the best of our knowledge, this system represents the first integration of an eye-tracker system with a serious game to train some capabilities. The game is *adaptable* to any kind of device, and *robust* to the absence of internet connection. Unfortunately, even if the game can be played, potentially, in any device, we must note that devices with very small screen, like the cell phones, strongly reduce the usability of the game.

We asked to an oculist and two psychologists to review the system, in order to evaluate the usability of the game for children affected by CVI. They suggested to remove some details from the background image of the game, initially crowded with gifts and Christmas decorations, that may act as distractors for children affected by CVI that can experience some difficulties in understanding the image itself and in recognizing the object to analyze for the game. This suggestion has been implemented in the system.

Moreover, a small group of children, with the same age of the target users, but not affected by CVI, play with the game, without the eye tracker system. In this second test, we wanted to study if the children got engaged with the game, and the result was positive.

The use of HTML5 standard and the PhoneGap framework has deeply influenced the development of the project since they allow to develop a unique application that can be made available on line, or built for different devices without the need to re-implement the game into another programming language.

Future works will be dedicated to study the precision of the system, which may depend on the calibration step. In this step, the system asks to the child to focus on an image placed in five different places of the screen, the four corners and its midpoint. Depending on the graveness of CVI, some children could not perform precisely the calibration step, reducing the quality of the eye gaze calculated. We plan to add an evaluation system of the training phase, in order to provide a measure of the precision of the child.

[3]The time required for the synchronization phase depends on the network bandwidth. We made some test using LAN, but since this process is performed only once for each client, this value does not affected the overall performance of the system.

REFERENCES

Adobe Systems (2012). Phonegap framework, http://phonegap.com/.

Campaña, L. (2012). iPad Apps for Children with Visual Impairments. In *Proc. of CTEVH, 2012*.

De Bortoli, A. and Gaggi, O. (2011). PlayWithEyes: A new way to test children eyes. In *Proc. of SeGAH, 2011*, pages 1–4.

Di Loreto, I. and Gouaich, A. (2011). Mixed reality serious games: The therapist perspective. In *Proc. of SeGAH, 2011*, pages 1–10.

Esteban, G., Fernandez, C., Matellan, V., and Gonzalo, J. (2011). Computer surgery 3D simulations for a new teaching-learning model. In *Proceedings of 1st SeGAH, 2011*, pages 1–4.

Forlines, C., Wigdor, D., Shen, C., and Balakrishnan, R. (2007). Direct-touch vs. mouse input for tabletop displays. In *Proc. of ACM CHI, 2007*, pages 647–656.

Gaggi, O., Galiazzo, G., Palazzi, C., Facoetti, A., and Franceschini, S. (2012). A serious game for predicting the risk of developmental dyslexia in pre-readers children. In *Proc. of ICCCN, 2012*, pages 1–5.

IETF, Internet Engineering Task Force (2012). The websocket protocol, RFC 6455.

ITU GazeGroup (2012). ITU Gaze Tracker, http://www.gazegroup.org/home.

Malkowicz, D., Myers, G., and Leisman, G. (2006). Rehabilitation of cortical visual impairment in children. *Journal of Neuroscience*, pages 116–135.

Michael, D. R. and Chen, S. (2006). *Serious games: games that educate, train and inform*. Thomson Course Technology, 1st edition.

Roman-Lantzy, C. (2007). *Cortical Visual Impairment: An Approach to Assessment and Intervention*. American Foundation for the Blind, 1st edition.

Sichitiu, M. and Veerarittiphan, C. (2003). Simple, accurate time synchronization for wireless sensor networks. In *Proc. of IEEE WCNC, 2003*, pages 1266–1273.

Thorlabs (2012). High Resolution USB 2.0 CMOS Cameras, http://www.thorlabs.de.