

UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Matematica

Corso di Laurea Magistrale in Informatica

**Sviluppo di un sistema per la diagnosi
e la riabilitazione di bambini
affetti da CVI**

Relatore:

Prof.ssa Ombretta Gaggi

Laureando:

Matteo Ciman

Matricola: 1014530

Indice

1	Introduzione	13
1.1	CVI	14
1.1.1	Cause e diagnosi di CVI	15
1.1.2	La riabilitazione allo stato dell'arte	18
1.2	Qualcosa di nuovo	22
2	I Serious Games	25
2.1	L'uso dei serious games	25
2.1.1	Aree applicative	26
2.2	Accessibilità stradale	28
2.3	Progetti esistenti	29
2.3.1	PlayWithEyes	29
2.3.2	Giochiamo	31
2.3.3	Junior Blind of America	33
3	Definizione del progetto	37
3.1	Descrizione generale	37
3.2	Obiettivi dei giochi	38
3.2.1	Prendimi!	39
3.2.2	Aiutami!	40
3.3	Ambienti di utilizzo	41
3.4	Eye-tracking	44
3.5	Analisi dei requisiti	44
3.5.1	Requisiti non funzionali	45
3.5.2	Requisiti funzionali	45
3.5.3	Requisiti di interfaccia	46
3.6	Utenti coinvolti	46
3.7	Design del sistema	46
3.7.1	Obiettivi Architettura	47
3.7.2	Definizione Architettura	48

4	Tecnologie di sviluppo	51
4.1	Rich Internet Application	52
4.2	HTML5	55
4.3	CSS3	65
4.4	AJAX	66
4.5	Librerie di supporto	68
4.6	Tecnologie lato server	69
4.6.1	PHP	69
4.6.2	Java	69
4.6.3	MySQL	69
 5	 Eye Tracking	 71
5.1	Il problema	72
5.2	Tracciamento basato su video	72
5.3	Tracciamento tramite webcam	73
5.4	Tobii	75
 6	 Sviluppo del progetto	 77
6.1	Costruzione dell'architettura	78
6.1.1	ServerManager	80
6.1.2	Gestore Eye-Tracker	85
6.1.3	Gestore client medico	86
6.1.4	Gestore client bambino	86
6.2	Il problema della sincronizzazione	86
6.3	Gestione della comunicazione	92
6.4	Sviluppo gioco Prendimi!	92
6.4.1	Definizione	92
6.4.2	Impostazioni di gioco	93
6.4.3	Informazioni per il medico	95
6.4.4	Valutazione delle prestazioni	97
6.5	Sviluppo gioco Aiutami!	99
6.5.1	Definizione	99
6.5.2	Parametri di gioco	99
6.5.3	Archiviazione delle informazioni	101
6.5.4	Informazioni per il medico	103
6.5.5	Valutazione delle prestazioni	104
6.6	Gestione offline dei giochi	106
6.6.1	Da online ad applicazione nativa	106
6.6.2	Salvataggio dei dati	108
6.6.3	Invio dei dati salvati	109
6.7	Test dell'applicazione	110

INDICE

7	Conclusioni	115
A	Manuale Utente	117
A.1	Utente bambino	118
A.1.1	Credenziali di accesso	118
A.2	Utente Medico	118
A.2.1	Schermo bambino	119
A.2.2	Schermo medico	119
A.3	Gestione bambini	128
A.4	Avvio del server	130
B	Manutenzione del prodotto	131
B.1	Gioco Prendimi!	131
B.1.1	Inserimento nuove immagini	131
B.2	Aiutami!	131
B.2.1	Tipologie di livelli	132
B.2.2	Immagini e famiglie	132
B.2.3	Costruzione esempi	133
B.2.4	Definizione dei livelli di default	134

Elenco delle figure

1.1	Esempio di immagini utilizzate per il livello 4 di riabilitazione, con forme semplici e pochi dettagli	20
1.2	Esempio di parete a scacchiera con le immagini inserite all'interno dei quadrati bianchi	20
1.3	Esempio di immagini con qualche dettaglio in più rispetto alla semplice immagine con soli contorni	21
1.4	Esempio di immagine costituita da molti dettagli, somministrata alla fine del livello 6 in caso di notevoli capacità da parte del bambino	21
2.1	Esempio di schermata di controllo per il medico nel progetto PlayWithEyes	31
2.2	Esempio di esercizio che il bambino deve compiere nel progetto PlayWithEyes	32
2.3	Esempio di gioco sviluppato per il progetto Giochiamo	33
2.4	Fasce d'età dei pazienti della società Junior Blind of America .	34
2.5	Esempio di schermata del gioco Bubbles Magic	35
2.6	Schermata di gioco e delle impostazioni del gioco Tap-n-See Zoo	35
2.7	Esempio di schermata del gioco Baby Aquarium	36
3.1	Fase di personalizzazione delle impostazioni del gioco Prendimi!	40
3.2	Esempio di architettura del sistema	49
4.1	Supporto delle features HTML5	54
4.2	Supporto per le feature grafiche di HTML5	55
4.3	Formati supportati dai vari browser per il tag audio	57
4.4	Prima fase del WebSocket handshake tra client e server	62
4.5	Risposta positiva del server alla richiesta di cambio protocollo da HTTP a WebSocket	62
4.6	Confronto tra l'overhead introdotto tra polling e WebSocket .	63
4.7	Confronto tra la latenza introdotta dal polling e da WebSocket	64

ELENCO DELLE FIGURE

4.8	Confronto tra il pattern di interazione classico sincrono con quello AJAX di tipo asincrono	66
5.1	Componenti di un sistema di eye-tracking basato su video . . .	73
5.2	Posizione della webcam per il tracciamento del movimento degli occhi	74
5.3	Occhiali Tobii per l'analisi del movimento degli occhi	75
6.1	Andamento dell'utilizzo dei browser nel periodo Agosto 2011 → Agosto 2012	77
6.2	Architettura finale del sistema realizzato	80
6.3	Indicazione delle informazioni spedite dal server al client del dottore	84
6.4	Esempio grafico costruito per il gioco Prendimi!	84
6.5	Esempio di valori messi a disposizione del medico per il gioco Aiutami!	85
6.6	Esempio di grafico costruito per il gioco Prendimi! attraverso le informazioni provenienti dalle diverse sorgenti	87
6.7	Scambio di pacchetti marcati temporalmente per il calcolo dell'offset tra gli orologi di due nodi	89
6.8	Rappresentazione dei pacchetti su un grafico come delle rette che definiscono dei vincoli sui valori finali	90
6.9	Pagina di personalizzazione delle impostazioni per il gioco Prendimi!	94
6.10	Messaggio che indica che i colori scelti per lo sfondo e per l'immagine hanno basso contrasto	94
6.11	Grafico riassuntivo dell'errore commesso dal bambino per il gioco Prendimi!	96
6.12	Esempio di tooltip rappresentante le posizioni di occhio e tocco rispetto all'immagine per il gioco Prendimi!	97
6.13	Sistema di controllo a disposizione del medico per il gioco Prendimi!	97
6.14	Divisione dell'immagine in aree per la valutazione del tocco e della vista	98
6.15	Schermata di personalizzazione dei livelli per il gioco Aiutami!	100
6.16	Esempio di riepilogo per il medico per il gioco Aiutami!	104
6.17	Riassunto valutazioni per il gioco Aiutami!	104
6.18	Esempio di riepilogo dettagliato per il gioco Aiutami!	105
6.19	Creazione di un'applicazione nativa tramite PhoneGap	107
6.20	Messaggio che informa della presenza di visite in modalità offline da inviare al server per l'archiviazione	109

ELENCO DELLE FIGURE

6.21	Immagine di Bugs Bunny utilizzata per la demo di Prendimi! .	111
6.22	Esempio di immagini più adatte all'utilizzo per il gioco Prendimi!	112
6.23	Prima versione dell'ambientazione per il gioco Aiutami!, risultata essere troppo complessa e piena di dettagli inutili	112
6.24	Seconda versione dell'ambientazione per il gioco Aiutami! con molti particolari eliminati	113
A.1	Schermata principale del sistema per il medico	119
A.2	Selezione di un bambino precedentemente inserito nel sistema	120
A.3	Form per l'inserimento di un nuovo bambino	120
A.4	Selezione del gioco con descrizione	121
A.5	Schermata di personalizzazione delle impostazioni per il gioco Prendimi!	122
A.6	Messaggio visualizzato per comunicare che il sistema è pronto a partire	123
A.7	Esempio di grafico presentato al medico per il gioco Prendimi!	123
A.8	Esempio di tooltip di dettaglio per il gioco Prendimi!	124
A.9	Controllo per la modifica della velocità per il gioco Prendimi! .	124
A.10	Sistema di controllo del gioco Prendimi!	124
A.11	Schermata per la costruzione dei livelli del gioco Aiutami! . . .	125
A.12	Bottoni utilizzati per cambiare l'ordine delle immagini nel gioco Aiutami!	126
A.13	Form di inserimento di un nuovo livello per il gioco Aiutami! .	126
A.14	Nuovo livello pronto per essere impostato	127
A.15	Esempio di rappresentazione per il medico del comportamento del bambino durante il gioco Aiutami!	127
A.16	Report che riassume il comportamento del bambino durante il gioco	128
A.17	Selezione del bambino e del gioco nella gestione delle visite . .	128
A.18	Esempio di tabella riassuntiva per il gioco Aiutami!	129
A.19	Grafico di dettaglio per una visita con il gioco Prendimi! . . .	129
A.20	Esempio dettagliato di riepilogo per il gioco Aiutami!	130

Abstract

Questo progetto di tesi, realizzato in collaborazione con il Dipartimento di Salute della Donna e del Bambino ed il Dipartimento di Psicologia Generale dell'Università di Padova, ha visto la realizzazione di un sistema per la diagnosi e la riabilitazione di bambini affetti da Cerebral Visual Impairment (CVI), ovvero bambini la cui difficoltà visiva è riconducibile ad un problema del cervello e della zona adibita all'elaborazione degli stimoli visivi, piuttosto che a problemi riconducibili solamente all'occhio.

Il progetto ha visto la realizzazione di una Rich Internet Application (RIA), che integra due *serious games* atti a sviluppare capacità visive e motorie, oltre ad abilità di *problem solving*, e la sua integrazione con un sistema di eye-tracking, sviluppato in un altro progetto di tesi, utilizzato per tracciare lo spostamento degli occhi durante gli esercizi proposti per la riabilitazione. Viste le innumerevoli combinazioni di deficit che si possono presentare in un bambino affetto da CVI, per tutto il tempo della progettazione e della realizzazione del sistema si è mantenuto come obiettivo fondamentale quello di costruire un sistema completamente personalizzabile, in maniera tale da adattarlo al meglio alle caratteristiche di ciascun bambino.

Infine, attraverso il framework PhoneGap è stata fornita la possibilità di utilizzare l'applicazione web anche su dispositivi mobile come i tablet, per permettere la riabilitazione del bambino anche a casa attraverso lo svolgimento degli esercizi proposti dal medico.

Capitolo 1

Introduzione

Attualmente la tecnologia in tutte le sue forme sta diventando sempre più un fattore fondamentale all'interno della vita delle persone, con applicazioni in molteplici campi e nelle situazioni più comuni.

Un campo di applicazione che sta prepotentemente prendendo piede negli ultimi anni è quello medico, dove la tecnologia viene utilizzata in diverse situazioni come la gestione elettronica dei dati, l'elaborazione delle immagini mediche, il controllo dei robot impiegati durante le operazioni chirurgiche e la sostituzione dei medici, la riabilitazione dei pazienti, ecc.

Il lavoro svolto in questa tesi prende proprio in considerazione la riabilitazione dei pazienti, in particolare quella dei bambini, sviluppando un progetto in collaborazione con la Dott.ssa Pinello del Dipartimento di Salute della Donna e del Bambino dell'Università di Padova, e la Prof.ssa Nota e Dott.ssa Sgarabella del Centro d'Ateneo di Ricerca e Servizi per la Disabilità, la Riabilitazione e l'Integrazione dell'Università di Padova, Dipartimento di Psicologia Generale. L'obiettivo del lavoro svolto è quello di sviluppare un sistema di supporto per il medico nella diagnosi e la riabilitazione di bambini affetti da Cerebral Visual Impairment (CVI).

Alla base di questo progetto vi sono due necessità fondamentali nella diagnosi e nella riabilitazione di bambini affetti da CVI. Per prima cosa, si vuole costruire un sistema di supporto per il medico nella diagnosi di tale disturbo. Tale necessità deriva dal fatto che se prendiamo in considerazione i bambini, affetti o meno da CVI, essi sono pazienti tipicamente difficili da trattare. In particolare se gli esercizi o i test proposti non li entusiasmano, nella maggior parte dei casi i risultati che si ottengono sono spesso non veritieri in quanto le risposte ricevute da parte del bambino possono essere sbagliate, vista la scarsa attenzione posta negli esercizi.

Inoltre, oltre ad un aiuto in termini diagnostici, è necessario anche un aiuto in termini riabilitativi, in quanto molto spesso gli esercizi proposti ai pazienti

sono di difficile accettazione, perchè poco stimolanti e noiosi. Se questi sono i problemi che si possono riscontrare con bambini (o pazienti) non affetti da questa tipologia di problema, nel caso dei bambini affetti da CVI i problemi sono ancora maggiori, in quanto molto spesso oltre a problemi visivi vi sono problemi anche a livello neurologico, e quindi la gestione del bambino, l'interfacciarsi con lui e qualsiasi altra operazione risulta essere ancor più difficoltosa.

Per questi motivi, l'idea di questo progetto nasce dalla necessità di trovare un nuovo sistema per l'interazione con bambini affetti da questo tipo di malattia, con l'obiettivo di sostituire gli approcci "statici" utilizzati attualmente con dei sistemi maggiormente interattivi e che soprattutto stimolino maggiormente il bambino alla partecipazione. Gli strumenti a supporto di questo tipo di approccio sono i *serious games*, ovvero giochi interattivi sviluppati per computer o tablet, il cui scopo non è solo quello di far divertire il bambino, ma camuffare dietro il divertimento un insieme di esercizi, operazioni, pensieri o comportamenti che aiutano il bambino nel processo di riabilitazione.

1.1 CVI

Cerebral Visual Impairment (CVI), a volte indicata anche con Cortical Visual Impairment, è una disabilità che comporta un deficit visivo che si manifesta nei bambini per via di un danno cerebrale. La differenza fondamentale tra questo tipo di problema ed il deficit visivo (mancanza di diottrie/capacità visiva, indicato solitamente con Ocular Visual Impairment) risiede nel fatto che in questo caso il problema non coinvolge necessariamente l'occhio o il nervo ottico, ma la zona nel cervello adibita all'elaborazione delle immagini oppure i cammini visivi presenti all'interno del cervello. CVI è un termine che può essere utilizzato anche per indicare una condizione dove un bambino (o un adulto) è "visually unresponsive" ma da un esame oculare non si riscontrano problemi o l'esame in sé non è in grado di giustificare la carenza visiva di cui il bambino è affetto.

Per concludere quindi, CVI viene definito, secondo Skoczenski e Good (2004), come

Una condizione in cui il bambino ha una capacità visiva ridotta come risultato di un danneggiamento dei percorsi visivi nel cervello. Nella maggior parte dei casi, gli occhi di tali bambini sono normali a livello strutturale, tuttavia presentano una capacità visiva ridotta.

1.1 CVI

1.1.1 Cause e diagnosi di CVI

Come detto precedentemente, le cause del CVI possono essere generalmente pensate come una condizione che influisce sui cammini visivi o sul centro di processamento delle immagini del cervello.

Le principali cause da cui può verificarsi uno sviluppo della patologia del Cerebral Visual Impairment sono:

- asfissia (assenza o scarsità di ossigeno che impedisce una respirazione normale);
- ipossia (carenza di ossigeno all'interno dell'organismo);
- ischemia (mancanza assoluta o parziale di sangue in un organo);
- problematiche nello sviluppo del cervello;
- colpi alla testa.

dove le prime quattro condizioni si verificano durante il processo di sviluppo del feto.

Sebbene il modo in cui i bambini affetti da CVI rispondono a livello visivo e le cause dietro queste risposte sono ancor oggi soggette a notevoli discussioni, sono stati identificati un insieme di comportamenti visivi che accompagnano la CVI. Queste caratteristiche visive sono sintomi di disfunzioni visive, che a turno interferiscono con la capacità visiva e a diversi livelli, a seconda della gravità del problema.

Le caratteristiche principali sono:

- forte preferenza verso i colori, in particolare il giallo ed il rosso;
- necessità di movimento per suscitare l'interesse e l'attenzione da parte del bambino; inoltre è necessario che o il bambino o l'oggetto sia in movimento per massimizzare l'abilità da parte del bambino di vedere l'oggetto;
- ritardo visivo nel vedere gli oggetti;
- preferenza nel guardare in particolari direzioni o posti, in concomitanza con la perdita di vista in certe zone dell'occhio;
- difficoltà nel gestire la complessità degli oggetti, in particolare difficoltà con oggetti in ambienti disordinati/rumorosi, dove c'è competizione con gli altri organi sensoriali per avere attenzione da parte del bambino;

- tendenza a fissare fonti luminose;
- problemi con la percezione della distanza;
- difficoltà con novità visive, constatando nel bambino una tendenza a preferire oggetti a lui familiari;
- incapacità a raggiungere un oggetto contemporaneamente all'operazione di osservazione, implicando cioè che il bambino non è in grado di toccare e seguire con lo sguardo, ma queste due attività vengono eseguite in due momenti diversi.

In generale, maggiore è il livello di gravità del CVI, maggiore sarà il numero di caratteristiche presenti. Tuttavia, queste caratteristiche possono cambiare o migliorare. Alcuni medici ritengono inoltre che buona parte dei bambini affetti da CVI, se non la maggior parte di essi, è in grado di migliorare, mentre alcuni di loro sono anche in grado di sviluppare una vista normale, con un procedimento che può richiedere mesi o addirittura anni.

Per questo motivo, un intervento anticipato, attraverso un studio ed una diagnosi anticipata e successivamente con un intervento mirato, risulta essere particolarmente importante per la crescita e lo sviluppo del bambino. In particolare, risulta essere particolarmente importante costruire un percorso di riabilitazione che sia il più adatto possibile alle caratteristiche ed alle problematiche del bambino, per cercare quindi di correggere al meglio quelli che sono i difetti riscontrati e cercare di ridurre al minimo i deficit derivanti dalla malattia.

Viste le diverse forme di deficit e di problemi che il Cerebral Visual Impairment può causare, la natura dei disturbi che caratterizzano ogni bambino sono diversi da bambino a bambino, ed è quindi necessario costruire esercizi riabilitativi diversi che siano specifici per ciascuno di essi.

Per questo motivo, il sistema che si vuole realizzare deve adattarsi il meglio possibile alle esigenze dei bambini. Ciò significa quindi che il sistema non deve essere realizzato con un'ottica generale, che semplicemente va bene ed è utilizzabile per tutti i bambini, ma deve essere un'evoluzione, e cioè adattarsi il più possibile ed il meglio possibile a quelle che sono le caratteristiche del singolo bambino, facendo in modo che l'esperienza di gioco (e quindi di cura) sia il più possibile adatta alle caratteristiche ed ai deficit specifici del bambino.

Il risultato sarà quindi che se il bambino ha problemi con i colori e con il contrasto, il sistema adatterà il colore di sfondo e delle immagini affinché siano sufficientemente in contrasto tra loro, se invece il bambino ha problemi con la visione periferica le immagini si concentreranno nella zona centrale

1.1 CVI

dello schermo, se invece il bambino avrà problemi di fissazione di oggetti in movimento, le immagini si muoveranno a velocità molto ridotta, ecc.

Preferenza verso i colori

Gli studi effettuati su bambini affetti da CVI hanno evidenziato una tendenza a preferire oggetti di particolari colori. Anche se le motivazioni per cui nasca questa preferenza non sono ancora note, quello che è noto è che i colori preferiti sono il giallo ed il rosso. Tuttavia, a livello più generale, è probabile che il bambino affetto da CVI si affezioni ad un particolare colore di cui ha, ad esempio, esperienza da quando è nato, come il colore della stanza, di un oggetto vicino al letto, ecc. Per questo motivo, risulta necessario integrare il più possibile il colore preferito all'interno degli esercizi giornalieri.

Necessità di movimento

Nei bambini affetti da CVI, si è visto come vi è un maggiore interesse per oggetti in movimento rispetto a quelli che invece sono fermi. Alcuni pazienti inoltre sono in grado di vedere gli oggetti solo nel caso in cui o essi stessi sono in movimento, oppure è l'oggetto che si muove. L'interesse da parte del bambino al movimento viene solitamente dimostrata o attraverso un sorriso, o diventando più tranquillo, oppure girandosi verso l'oggetto stesso. Per generare interesse prolungato quindi da parte dei bambini è necessario o usare oggetti in movimento, oppure oggetti con una superficie riflettente che simuli il movimento. Alcuni bambini hanno anche la tendenza a muoversi in maniera quasi casuale all'interno dell'ambiente: questo può trovare spiegazione nella necessità di simulare il movimento di un oggetto che è fermo o che non ha una superficie riflettente. Altra dimostrazione della necessità di movimento deriva ad esempio dal guardare fuori dal finestrino di una macchina, o guardare la tv soprattutto durante programmi sportivi, dove abbiamo un notevole movimento di forme e colori.

Latenza visiva

La latenza visiva si riferisce al tempo che passa da quando un oggetto viene presentato al bambino, a quando effettivamente il bambino osserva tale oggetto. Un individuo con latenza visiva ha una risposta molto lenta e molto spesso ritardata. È necessario quindi lasciare al bambino una sufficiente quantità di tempo a disposizione per poter avere una reazione, che può essere molto prolungato anche di fronte ad un oggetto "comune" e del colore preferito.

Preferenza verso alcuni campi visivi

La maggior parte dei bambini affetti da CVI si è visto avere dei campi visivi preferiti rispetto ad altri. Alcuni individui inoltre, tendono a vedere un oggetto con un solo occhio, e poi girare la testa per utilizzare l'altro occhio per esaminare i dettagli dell'oggetto e identificarlo.

Difficoltà con complessità visiva

I bambini affetti da CVI evidenziano una certa difficoltà ad interfacciarsi ed interagire con scenari ed ambienti ad elevata complessità visiva, in quanto il loro sistema visivo viene sovraccaricato da stimoli multipli ed in competizione tra loro, con il risultato che il cervello non è in grado di elaborare contemporaneamente tutti gli stimoli ricevuti, ma solamente pochi alla volta.

I bambini affetti da CVI dimostrano una migliore risposta quando hanno a che fare con oggetti con forme semplici. Inoltre, oggetti che hanno superfici semplici e con un singolo colore sembrano aiutare i bambini a mantenere l'attenzione verso di loro.

Risulta comunque importante notare come anche presentare un singolo oggetto, rispetto una sequenza di oggetti, è molto più semplice da gestire per il bambino affetto da CVI. Inoltre, anche lo sfondo in cui si trova l'oggetto deve essere considerato con molta attenzione, in quanto uno sfondo più semplice consente un migliore riconoscimento dell'oggetto da parte del bambino. Per finire, un bambino affetto da CVI non riesce a distinguere oggetti che sono troppo vicini tra loro, ed è quindi necessario mantenerli sempre a sufficiente distanza uno dall'altro per evitare che il cervello del bambino sovrapponga e unisca tali oggetti.

Anche l'ambiente in cui si lavora è molto importante, visto che fornire un ambiente con molti stimoli sensoriali per un bambino può essere molto difficile, in quanto si rischia che non sia in grado di gestire tutti gli stimoli e quindi che perda la propria attenzione da quello che invece dovrebbe essere l'oggetto al centro del proprio interesse.

1.1.2 La riabilitazione allo stato dell'arte

Come detto precedentemente, la riabilitazione dei bambini affetti da CVI inizia con un'accurata visita che sia in grado di determinare qual è il livello di gravità del bambino, a cui segue la definizione di un percorso di esercizi che sia il più aderente possibile ai problemi riscontrati durante la visita.

Un esempio di procedimento di cura di pazienti affetti da CVI è presentato

1.1 CVI

in [18]. L'approccio generale è quello di partire da una classificazione del bambino sulla base delle sue caratteristiche visive e cerebrali, in maniera tale da definire il punto di partenza a partire dal quale definire poi gli interventi successivi.

Una volta determinato il livello di gravità del bambino, viene costruito un percorso personalizzato e specifico, che è basato sulla gravità del problema e su quelle che sono le caratteristiche del bambino stesso, come ad esempio colori o forme preferite, capacità di riconoscere degli elementi ed interagire con gli stessi ecc.

Una volta definito il programma di lavoro, sarà compito del genitore far svolgere al bambino gli esercizi richiesti presso il proprio domicilio, mantenendo un costante contatto con il medico in maniera tale da aggiornarlo sui progressi o sulle difficoltà riscontrate nella somministrazione degli esercizi.

Nello studio effettuato in [18] viene preso in considerazione un campione di 21 bambini, tutti affetti sia da CVI che da altri disturbi neurologici. Ogni bambino è caratterizzato da diversi livelli di capacità sensoriali, sia visive che di movimento, e dopo una visita atta a valutarne il livello di deficit, viene definito il livello di partenza del programma da seguire, dove a livelli diversi corrispondono livelli diversi di deficit visivo. Tale programma può durare tra i 4 ed i 15 mesi. I diversi esercizi proposti, in scala di difficoltà a partire dal più semplice per arrivare al più complesso, vengono svolti sia in casa che presso la clinica. Gli esercizi sono i seguenti:

- **Livello 1:** all'interno di una stanza completamente buia e senza disturbo sonoro, si utilizza una sorgente luminosa (come ad esempio una penna con la luce, tipicamente utilizzata dagli oculisti) per inviare dei flash ai due occhi del bambino;
- **Livello 2:** per sviluppare nel bambino la capacità di vedere il contrasto tra colori, il bambino viene collocato all'interno di una stanza senza arredamenti, all'interno della quale nelle pareti sono affissi dei poster costituiti da dei quadrati alternati di colore nero e bianco;
- **Livello 3:** all'interno di una stanza buia, attraverso l'uso di una penna illuminante si indica un quadrato alla volta nei poster utilizzati per il livello precedente, e l'obiettivo del bambino è quello di osservare il quadrato illuminato;
- **Livello 4:** in questo livello si iniziano ad introdurre delle forme molto semplici, come ad esempio un quadrato, un triangolo, un cerchio ecc. Queste figure vengono disegnate su schede quadrate, con immagini bianche su sfondo nero o immagini nere su sfondo bianco, per

poter avere il massimo contrasto possibile, e vengono somministrate al bambino affinché le possa osservare (Figura 1.1);

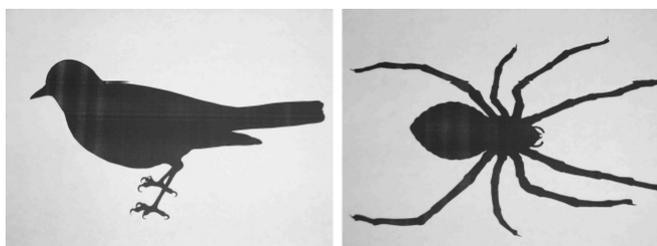


Figura 1.1: Esempio di immagini utilizzate per il livello 4 di riabilitazione, con forme semplici e pochi dettagli

- **Livello 5:** le immagini bianche e nere, o anche colorate, vengono inserite all'interno della scacchiera costruita al livello 2, e vengono illuminate una alla volta affinché il bambino si concentri su una sola immagine alla volta, anche all'interno di una configurazione complessa contenente più di un'immagine (Figura 1.2);



Figura 1.2: Esempio di parete a scacchiera con le immagini inserite all'interno dei quadrati bianchi

- **Livello 6:** le immagini utilizzate precedentemente vengono sostituite da immagini più complesse, per sviluppare nel bambino la capacità di

1.1 CVI

individuare anche i dettagli che possono caratterizzare un'immagine (Figure 1.3 e 1.4);

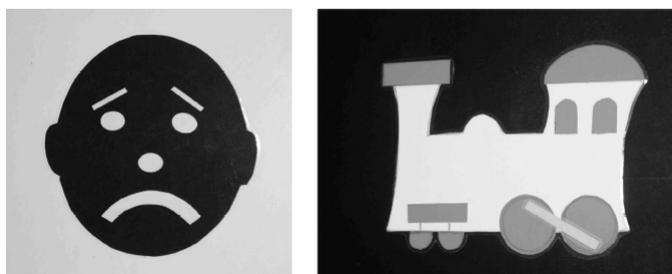


Figura 1.3: Esempio di immagini con qualche dettaglio in più rispetto alla semplice immagine con soli contorni



Figura 1.4: Esempio di immagine costituita da molti dettagli, somministrata alla fine del livello 6 in caso di notevoli capacità da parte del bambino

- **Livello 7:** le schede con le diverse figure vengono sostituite da schede che contengono delle semplici parole. Questa fase è l'unica fase personalizzata.

I risultati ottenuti da questo tipo di procedimento, rispetto all'esperimento svolto da Hoyt nel 2003 [24] dove valutava il miglioramento dei bambini affetti da CVI senza alcun stimolo visivo / riabilitazione, sono decisamente positivi. Nel 76% dei casi infatti, il miglioramento apportato è stato pari a tre livelli nella capacità visiva, che equivale ad un passaggio da una capacità di semplice percezione della luce o delle forme, alla capacità di identificare simboli o lettere.

Tuttavia, come possiamo osservare, tale approccio ha una forte limitazione:

la differenza tra i percorsi svolti dai pazienti consiste solamente nel livello di partenza, ossia da quale esercizio partire per lo stimolo visivo. Tutto il restante percorso, eccezion fatta per l'ultimo livello che invece presenta parole personalizzate, risulterà essere uguale per ciascun bambino, differendo eventualmente nella durata di un particolare livello sulla base delle risposte e dei miglioramenti del bambino stesso.

1.2 Qualcosa di nuovo

Come detto precedentemente, il procedimento di riabilitazione di un bambino affetto da CVI è basato su una diagnosi accurata dei problemi del bambino (che molto spesso risulta comunque essere molto difficoltosa), e sullo svolgimento a casa di esercizi specifici per ogni bambino sotto la supervisione del genitore, in costante contatto con il medico.

I problemi che questo tipo di approccio comporta sono molteplici, che rischiano di rendere difficoltosa sia una diagnosi che una riabilitazione accurata.

In primo luogo, il problema fondamentale è quello di riuscire a catturare l'attenzione da parte dei bambini. Infatti, i pazienti più giovani (insieme a quelli più anziani) sono spesso quelli più difficili da cui ottenere una collaborazione, visto che molto spesso, se non attratti o coinvolti dall'esercizio che stanno svolgendo, o anche semplicemente annoiati, hanno un comportamento che non li porta ad affrontare al meglio gli esercizi proposti, facendo sì che la visita e gli esercizi svolti siano totalmente inutili.

In secondo luogo, un altro problema è quello della valutazione dei progressi. Poiché non è chiaramente pensabile chiedere al bambino se sta migliorando e come migliora con gli esercizi proposti, il medico deve basarsi solamente su una sua valutazione soggettiva (e sulle indicazioni fornite dal genitore che segue il bambino durante gli esercizi casalinghi) per decidere quali strumenti e quali strategie seguire per la riabilitazione.

Come è facile osservare, questo tipo di approccio può e deve essere migliorato, soprattutto secondo due direttive fondamentali:

- il sistema deve catturare l'attenzione del bambino e far sì che questi resti il più possibile interessato ed attratto dagli esercizi proposti, al fine di ottenere un comportamento più propositivo nei confronti della riabilitazione da parte del bambino ;
- definire un sistema di valutazione delle prestazioni che permetta di valutare quelli che sono i progressi (o i peggioramenti) del bambino, con la possibilità di adattare le caratteristiche degli esercizi proposti sulla base dello stato di riabilitazione del bambino.

1.2 Qualcosa di nuovo

Per questo motivo, insieme al team di ricerca che gestisce il progetto si è deciso di costruire un'applicazione, fruibile in diversi contesti e situazioni (come esplicitato in 3.3), che funga sia da supporto al medico nella diagnosi, valutazione e programmazione del lavoro, sia al bambino, introducendo degli esercizi tramite lo sviluppo di *serious games* con l'obiettivo di invogliare il più possibile il bambino al gioco e quindi alla riabilitazione.

Questa scelta, di introdurre strumenti di riabilitazione il più interattivi possibile, anche per pazienti affetti da CVI, è una scelta che sta prendendo sempre più piede nel processo di riabilitazione. Questo perchè è stato possibile osservare come attraverso questa tipologia di sistemi, l'approccio di un qualsiasi bambino è molto più propositivo e disponibile rispetto a quelli che sono i metodi standard di riabilitazione, come ad esempio esercizi in palestra. Un esempio di applicazione è quello proveniente dalla riabilitazione di pazienti colpiti da trauma agli arti, che attraverso dei giochi costruiti appositamente effettuando riabilitazione casalinga giocando [6].

Capitolo 2

I Serious Games

Il concetto di *serious games* non è attualmente un concetto definito in maniera assoluta e dalla definizione precisa, ma cambia specifiche e obiettivi a seconda del contesto in cui tale termine viene utilizzato.

Volendone evidenziare le caratteristiche comuni, i *serious games* possono essere considerati come dei giochi o delle applicazioni, sviluppati per poter essere eseguiti su computer o dispositivi mobile come tablet o smartphone, il cui obiettivo non è semplicemente quello di intrattenere il giocatore, ma anche quello di far sviluppare al giocatore particolari capacità nascondendole sotto il semplice gioco. Alcune volte, il gioco può addirittura sacrificare il divertimento e l'intrattenimento con l'obiettivo di raggiungere un particolare scopo da parte del giocatore. Solitamente un *serious game* si presenta sotto forma di una simulazione che ha la stessa apparenza di un videogioco normale, ma in realtà esso non è altro che una simulazione di eventi e processi relativi alla realtà. L'obiettivo finale è solitamente quello di insegnare o educare gli utenti, anche se può essere utilizzato in altri ambiti meno pratici come il marketing e la pubblicità, fornendo agli utenti un'esperienza molto più coinvolgente della semplice pubblicità stampata.

2.1 L'uso dei serious games

Prendendo in considerazione solamente l'insieme dei *serious games* costruiti per uno scopo diverso da quello di marketing, sebbene non vi siano studi specifici che attestino in maniera inconfutabile l'utilità dei *serious games* nei diversi ambiti di utilizzo, essi, se costruiti ed utilizzati correttamente, sono attualmente considerati un ottimo strumento a supporto della formazione e dell'apprendimento. Le competenze che possono essere sviluppate

attraverso questa tipologia di simulazione sono molteplici, come ad esempio capacità analitiche, visive, strategiche ecc.

Alla base dei *serious games* c'è la logica di apprendimento attraverso il principio del *learning by doing*, che letteralmente sta per “imparare facendo”. Negli ultimi anni, questa strategia di apprendimento è risultata essere la migliore strategia di apprendimento in molti ambiti, sostituendo l'idea del semplice memorizzare con quella del comprendere. Il *learning by doing* è già molto utilizzato all'interno delle attività pedagogiche, ed è sostanzialmente l'insegna delle attività didattiche più aggiornate, seppur con alcune precisazioni. Ovviamente, l'apprendimento completo non si ha con il semplice fare, ma il fare deve essere accompagnato dal pensiero e dalla riflessione sull'argomento trattato, altrimenti la semplice attività non risulterà essere formativa ma semplicemente una memorizzazione dei passi. Per questo motivo quindi, i *serious games* non devono solo portare all'esecuzione di procedure meccaniche, ma scatenare nell'utente anche la riflessione, il pensiero ed il ragionamento, quello che viene definito *learning by thinking*.

2.1.1 Aree applicative

Le aree di applicazione dei *serious games* sono molteplici, e tutte con obiettivi diversi volte a sviluppare specifiche capacità.

In ambito **militare**, i *serious games* vengono utilizzati soprattutto per l'addestramento delle persone, in particolare prima delle missioni con l'obiettivo di riprodurre attraverso il gioco le condizioni ambientali e gli ostacoli che i soldati si troveranno ad affrontare, allo scopo di sviluppare quelle capacità decisionali che gli permettano poi di risolvere i problemi durante l'esperienza reale: un esempio a riguardo è “Army Battlezone”, primo videogioco appartenente a questa categoria. L'importanza di questa tipologia di addestramento e simulazione è sottolineata dal fatto che il Pentagono ogni anno stanziava circa 4 milioni di dollari da investire sotto la voce “*simulation equipment and war-games*”, a dimostrazione quindi dell'enorme interesse verso questa materia.

In ambito **formativo**, le principali direttive seguite dai *serious games* sono due. La prima riguarda la costruzione di simulatori di velivoli, treni, metropolitane ecc, utilizzati nella formazione del personale prima di inserirlo all'interno del contesto reale, con la possibilità di simulare situazioni particolari come ad esempio guasti o inconvenienti, per sviluppare nei lavoratori le capacità decisionali soprattutto in tempi molto rapidi.

Il secondo ambito, più prettamente aziendale, riguarda invece quello della formazione del personale, sia nuovo che vecchio. Le capacità che in questo caso si vogliono sviluppare sono molteplici, a partire dalle capacità di pro-

2.1 L'uso dei serious games

blem solving, per arrivare a quelle organizzative, comunicative e strategiche. Attraverso questa nuova modalità di formazione, il dipendente passa da un semplice ruolo passivo di acquirente di informazioni, ad un ruolo più attivo dove anche la formazione più noiosa può trasformarsi in una situazione maggiormente stimolante, con ovvi benefici anche per l'azienda, in particolare per quel che riguarda il risultato finale di apprendimento del dipendente.

In ambito **governativo**, i *serious games* vengono utilizzati come simulatori di scenari. Quello che si vuole sviluppare attraverso questa tipologia di simulazioni è la capacità, in questo caso da parte di governanti e politici, di prendere le decisioni migliori per la comunità sulla base degli eventi che si verificano e sulla base anche delle reazioni che le loro scelte comportano, come ad esempio attacchi terroristici, gestione del traffico, dell'assistenza sanitaria, la gestione del bilancio ecc.

In ambito **educativo**, i *serious games* vengono utilizzati con i bambini per sviluppare molteplici capacità, come quelle di memoria, comunicazione, apprendimento ecc. Sebbene non esistano prove scientifiche dell'effettiva validità di questi metodi di apprendimento, questi giochi, se opportunamente e correttamente sviluppati, possono diventare dei validi strumenti nell'accompagnamento del bambino nel proprio percorso scolastico, anche con l'introduzione di tali giochi all'interno delle scuole. L'introduzione di questi *serious games* all'interno delle scuole è supportato anche dal fatto che i bambini contemporanei vivono nella tecnologia fin da quando sono nati, venendo così definiti come *digital native speaker*, ovvero "di madrelingua digitale", indicando il fatto che questi bambini crescono in un ambiente immerso nella tecnologia, e quindi la conoscenza dell'uso del computer o tablet o smartphone è parte del bagaglio culturale che possiede un bambino sin dai primi anni. Per questo motivo, l'introduzione di metodi alternativi che sfruttino i media digitali risulta essere sicuramente un valido approccio rispetto a quello classico, con i serious games che possono essere utilizzati a questo scopo [28].

L'ultimo ambito di utilizzo, ma secondo in termini di soldi investiti dietro solo a quello militare, è l'ambito **medico**. I campi di applicazione in questo caso sono molteplici, con uno sviluppo rivolto sia ai pazienti che direttamente ai medici. In particolare:

- formazione e allenamento per personale medico, con simulazione di interventi, malattie, diagnosi e cure;
- monitoraggio delle funzioni vitali per il controllo di reazioni o stati emotivi;
- recupero e riabilitazione di attività o arti a seguito di traumi;

- educazione alla salute ed a pratiche di vita corrette;
- stimolo a praticare un costante esercizio fisico.

Il progetto presentato in questa tesi si colloca all'interno dell'ambito medico, con lo sviluppo di *serious games* con l'obiettivo di riabilitare i bambini affetti da CVI con uno strumento tecnologico ed innovativo che cerchi di superare la scarsa interattività degli esercizi riabilitativi attuali con un sistema interattivo che stimoli il più possibile il bambino al gioco e contestualmente alla riabilitazione.

2.2 Accessibilità stradale

Un esempio di *serious game* che combina il divertimento con l'utilità, in questo caso dedicata a persone affette da cecità, è il progetto presentato in [4]. L'idea alla base di tale progetto è molto semplice: si vuole costruire un database di informazioni relative agli attraversamenti pedonali delle diverse città dotati di segnalatore acustico, al fine di costruire successivamente dei percorsi accessibili che possano essere utilizzati da persone affette da cecità. Il punto di partenza di tale progetto è stato un progetto dagli intenti simili, ma con un target di utenti più generali, chiamato *Path 2.0* [5]. L'obiettivo di tale progetto è quello di costruire, attraverso l'esperienza di persone disabili nel muoversi all'interno delle città e sfruttando le capacità degli smartphone attuali (in particolare il GPS), un database di informazioni contenente, per ogni città, un elenco di percorsi accessibili per le persone disabili, categorizzando tale informazioni sulla base della disabilità stessa. In questo modo, una persona con problemi motori che necessita, ad esempio, di una carrozzina per muoversi, può utilizzare tale servizio per trovare, nel momento in cui visita in una nuova città, il percorso verso la propria destinazione che sia il più adatto possibile alla propria disabilità, sulla base dell'esperienza degli altri utenti che tale percorso l'hanno già affrontato.

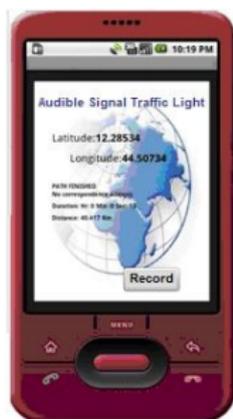
Il problema principale di tale applicazione è quello relativo allo scarso coinvolgimento di persone senza alcuna disabilità, che quindi comporta uno scarso numero di percorsi disponibili, perdendo quindi tutta l'utilità che essa può avere.

Per questo motivo, il secondo progetto è stato sviluppato nell'ottica di cercare di invogliare maggiormente l'utilizzatore dell'applicazione, al fine di ottenere un maggior numero di informazioni per la costruzione del database. La scelta è stata quella di sviluppare un *serious game* che costruisse una sorta di sfida a punti tra gli utenti, dove ogni nuovo incrocio accessibile a persone non vedenti inserito nel sistema fornisce un guadagno in termini di punti,

2.3 Progetti esistenti

permettendo di scalare posizioni nella classifica globale. Ovviamente, il sistema prevede anche la penalizzazione per utenti che segnalano più volte lo stesso semaforo.

In questo modo, si nasconde un'applicazione di pubblica utilità dietro un gioco, invogliando quindi le persone a parteciparvi, ed aumentando le informazioni a disposizione per chi poi ne avrà bisogno, ovvero persone affette da cecità. La Figura 2.1a fornisce un esempio dell'applicazione per smartphone, mentre la Figura 2.1b del servizio per l'indicazione dei semafori accessibili lungo il percorso verso la destinazione.



(a) L'applicazione sviluppata per il progetto *Accessible Crossroads*



(b) Esempio di costruzione di un percorso con indicazione dei semafori accessibili

2.3 Progetti esistenti

Nel momento in cui si sviluppa un nuovo progetto software, risulta spesso molto utile muovere i primi passi analizzando le soluzioni già presenti sul mercato, allo scopo eventualmente di estrarre dei punti di partenza e valutare le scelte fatte (con aspetti positivi e negativi) in progetti simili. In particolare, vista la natura del progetto e le parti coinvolte, si sono valutati due progetti: *PlayWithEyes* e *Giochiamo*.

2.3.1 PlayWithEyes

PlayWithEyes [3] [1] è uno strumento per lo screening visivo per bambini in età pediatrica sviluppato da Alberto De Bortoli nell'ambito della sua tesi di Laurea Magistrale in Informatica presso l'Università di Padova.

L'obiettivo di questo sistema è quello di misurare l'acutezza visiva dei bambini sottoponendoli a dei giochi, attraverso un'interfaccia touch sviluppata in particolare per dispositivi Apple[©] come iPod Touch[©], iPhone[©] e iPad[©]. Sebbene il modello tecnologico adottato per questo tipo di soluzione non è praticabile per il nostro progetto, visto che l'obiettivo principale è quello di realizzare un'applicazione che sia il più portabile possibile e quindi fruibile dalla maggior quantità possibile di dispositivi, siano essi tablet o computer, è stato preso in considerazione in quanto, oltre ad essere un *serious game* in ambito medico, il target di utenti, in termini di età, è molto simile a quello relativo a questa tesi. Questo progetto ci permette quindi di valutare se l'uso dei *serious games* è efficace e permette di ottenere buoni risultati in termini di risposta ed interesse da parte dei bambini.

Il sistema è costituito fondamentalmente da due unità distinte, una dedicata al medico (che utilizza un iPad[©]) ed una dedicata invece al bambino, che utilizza un iPod[©] Touch o un iPhone[©]. La schermata del dottore è collegata ad un proiettore, posto ad una determinata distanza dal bambino, ed attraverso il gioco il bambino deve eseguire una serie di test oculistici per valutarne le capacità visive.

La Figura 2.1 mostra la schermata a disposizione del medico per il controllo dell'esercizio, mentre la Figura 2.2 mostra un esempio di esercizio che il bambino deve compiere, in risposta a quello proiettato sullo schermo. Dai risultati presentati, possiamo notare come su un campione di 65 bambini (di età compresa tra i 3 ed i 6 anni), il 92% di loro hanno dichiarato che l'esperienza di gioco è stata gradevole e che avrebbero rigiocato volentieri. Ciò dimostra in maniera evidente come i bambini abbiano trovato l'esperienza piacevole, sebbene in realtà l'esperienza era finalizzata ad un test medico, oltre a dimostrare come un'interazione touch anche con i più piccoli non sia una scelta che può comportare delle difficoltà nell'utilizzo.

Per quel che riguarda invece i risultati dei test a livello medico, i responsi ottenuti sono stati molto positivi, permettendo di individuare bambini con difficoltà visive e di indirizzarli verso una più approfondita visita da parte di medici oculisti. Questo dimostra anche come, attraverso i *serious games*, si possano ottenere informazioni e dati molto importanti per l'analisi e lo studio medico.

Questi risultati permettono quindi di confermare come l'adozione dei *serious games* per far svolgere ai bambini particolari task con l'obiettivo di ottenere dei risultati anche medici sia sicuramente una scelta molto azzeccata, sfruttando anche la modalità di interazione touch che risulta essere molto semplice da utilizzare anche per i bambini.

2.3 Progetti esistenti

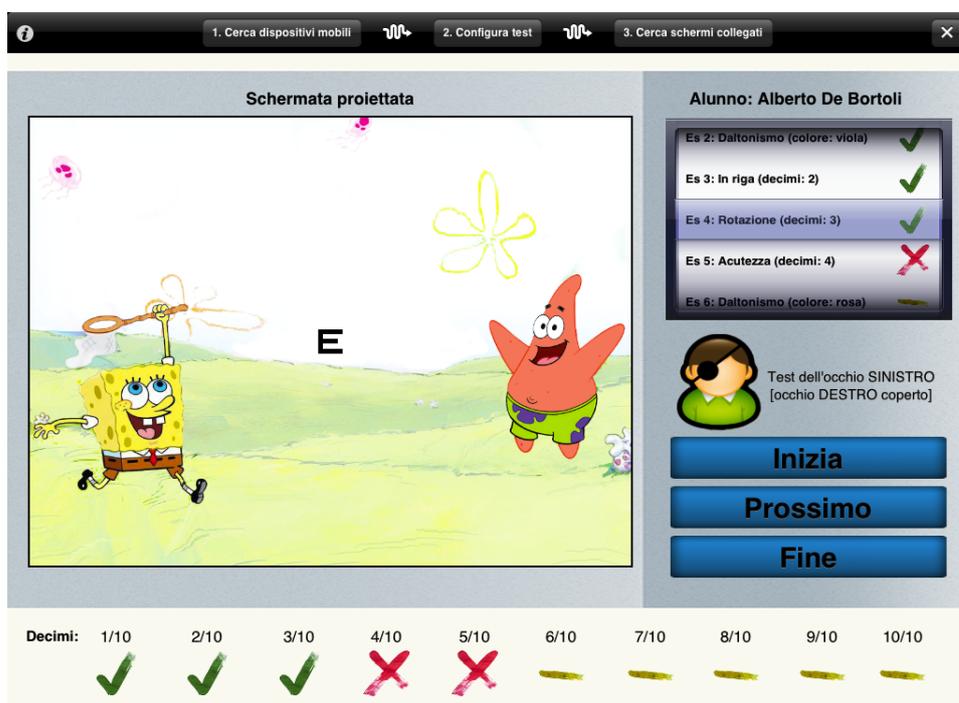


Figura 2.1: Esempio di schermata di controllo per il medico nel progetto *PlayWithEyes*

2.3.2 Giochiamo

Giochiamo [9] [21] è un progetto sviluppato da Giorgia Galiazzi come tesi di Laurea Magistrale in Informatica presso l'Università di Padova, in collaborazione con il Dipartimento di Psicologia Generale.

L'obiettivo di questo progetto è quello di sviluppare un pacchetto di *serious games*, il cui obiettivo è il trattamento della dislessia evolutiva in età precoce, cioè in bambini in età prescolare in cui è stato riscontrato un rischio di sviluppo del disturbo.

A differenza del progetto *PlayWithEyes*, sviluppato solamente per dispositivi Apple[®], in questo caso il progetto è stato sviluppato sotto forma di Rich Internet Application (RIA) portabile, in maniera tale da poter essere usufruito sia con dispositivi desktop che dispositivi mobile.

In particolare, si è ricorso a tecnologie come HTML5, CSS3 e Javascript per realizzare applicazioni web interattive usufruibili da dispositivi desktop, e con l'aggiunta di un framework aggiuntivo (PhoneGap [22]), si è trasformato tale applicazione web in un'applicazione nativa per la maggior parte dei dispositivi touch, quindi non solo dispositivi Apple[®], ma ad esempio anche Android[®]. Come vedremo più avanti, gli scenari di utilizzo e gli obiettivi

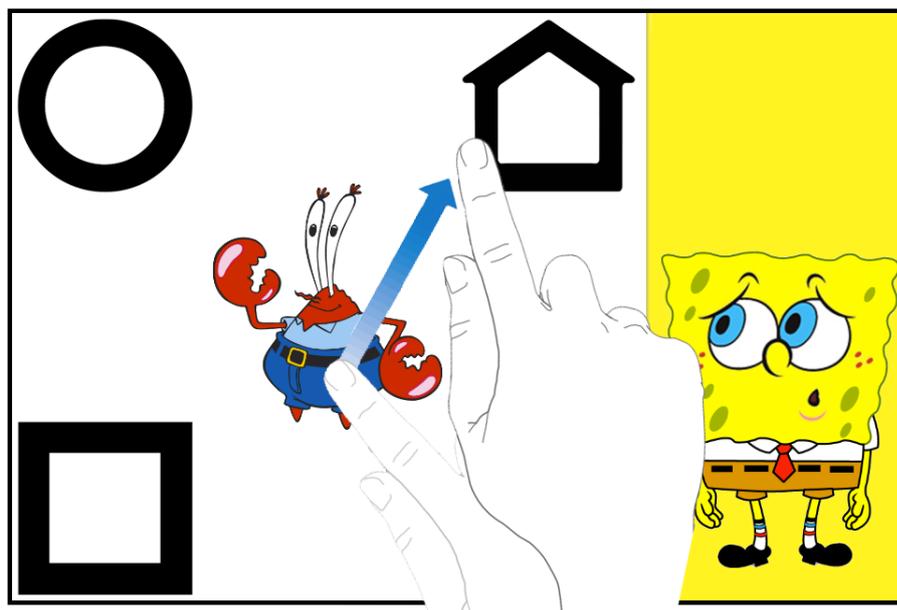


Figura 2.2: Esempio di esercizio che il bambino deve compiere nel progetto *PlayWithEyes*

considerati per questo progetto sono molto simili a quelli di questa tesi. Per quel che riguarda l'indice di gradimento di questo progetto, anche in questo caso il divertimento da parte dei bambini è stato molto elevato. Come risultato dai test compiuti su un campione di 24 bambini, più del 50% di essi si è dichiarato molto contento dell'esperienza di gioco, mentre più di un 75% semplicemente contenta. Questo dato indica quindi che l'esperienza di gioco è sicuramente positiva, e non vi saranno quindi problemi nel proporre tali giochi in un percorso riabilitativo di durata mensile o superiore. Inoltre, i risultati dei test hanno evidenziato come i giochi proposti ai bambini non sono mai risultati facili per tutti i pazienti. Non va infatti dimenticato che anche se si parla di *games*, tali giochi hanno uno scopo riabilitativo / medico più profondo del semplice divertimento. In questo caso quindi, la valutazione come giochi non facili fa sì che l'esperienza di gioco abbia portato anche ad una sfida per il bambino, con conseguente allenamento delle capacità fonologiche carenti, cosa che non sarebbe avvenuta nel caso in cui i giochi fossero risultati troppo semplici.

Quello che quindi questo progetto ha realizzato è un sistema utilizzabile da più dispositivi, a seconda delle esigenze o delle necessità dei diversi fruitori, che si è dimostrato in grado di essere utilizzato per il trattamento della dislessia evolutiva, allenando abilità coinvolte nella lettura in maniera piacevole e non pesante per il bambino.

2.3 Progetti esistenti

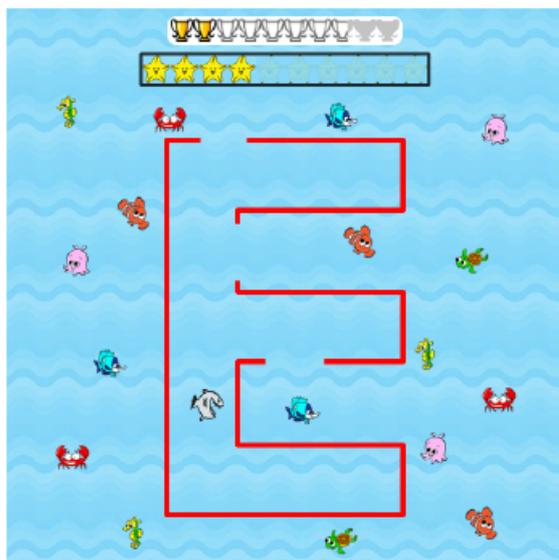


Figura 2.3: Esempio di gioco sviluppato per il progetto Giochiamo

2.3.3 Junior Blind of America

Junior Blind of America è una società americana fondata nel 1953, il cui compito è quello di seguire bambini, ragazzi o adulti affetti da ipovisione più o meno grave.

La società, nel biennio 2010-2011, ha aiutato 11.382 persone tra bambini e adulti, senza alcun costo per le famiglie. I programmi messi in campo da questa società sono molteplici, a partire dalla nascita del bambino, dove si contribuisce alla presa di consapevolezza anche del genitore del problema di CVI del proprio figlio, al supporto del bambino nel suo ingresso nella scuola, fino all'aiuto nella scelta del percorso scolastico e lavorativo che meglio si adatta alle proprie caratteristiche.

In Figura 2.4 possiamo notare come le fasce d'età siano molteplici, anche se la maggior parte delle persone assistite comprende bambini di età compresa tra gli 0 e 5 anni. Questo probabilmente significa che il percorso intrapreso da tale struttura nell'accompagnamento dei bambini a partire dai primissimi anni li aiuta a diventare sufficientemente indipendenti per potersi interfacciare con il mondo esterno senza il bisogno di una struttura particolare, quindi o in maniera totalmente autonoma oppure eventualmente supportati dai genitori. Particolarmente interessante per questo progetto è l'analisi fatta in [2], dove si discute dell'apporto molto positivo introdotto dall'uso di strumenti touch, come ad esempio l'iPad[©] nella cura dei bambini affetti da CVI. In particolare, quello che è stato visto è che con la sostituzione dei precedenti

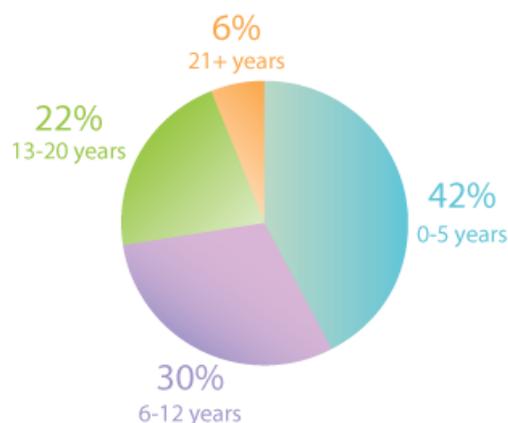


Figura 2.4: Fasce d'età dei pazienti della società Junior Blind of America

strumenti utilizzati per la riabilitazione con un tablet con installate applicazioni particolari adatte ai bambini con questo tipo di disabilità, come ad esempio *Tap-n-See Zoo*, *Bubbles Magic* ecc, si ottengono delle risposte e delle attenzioni da parte dei bambini migliori rispetto agli strumenti precedenti. Il tutto comporta quindi una migliore interazione, una migliore risposta e quindi anche dei risultati migliori nel trattamento dei problemi di CVI.

Dalla presentazione delle applicazioni installate all'interno di un dispositivo iPad© utilizzato per la riabilitazione dei bambini affetti da CVI, la più semplice utilizzata è sicuramente *Bubbles Magic* di *Squishy Things Inc.* [11]. Questo gioco, che disegna delle bolle sullo schermo quanto il bambino lo tocca, e le fa scoppiare se il bambino tocca nuovamente la bolla, non ha in sé un valore formativo, ma viene utilizzato per far avvicinare per la prima volta un bambino affetto da CVI all'iPad©. Tramite questi effetti visivi, ma soprattutto sonori, l'applicazione è particolarmente adatta a richiamare l'attenzione del bambino verso il dispositivo tablet e stimolare in lui un interesse. Essa è quindi sicuramente il punto di partenza per costruire un'interazione tra tablet e bambino per poter poi iniziare con delle applicazioni più formative o propedeutiche. La Figura 2.5 mostra un esempio di schermata che si viene a creare dopo una sequenza di tocchi sullo schermo. Un altro esempio che è possibile prendere in considerazione è il gioco *Tap-n-see Zoo* sviluppato da *Little Bear Sees LLC* [17]. Questo gioco, sviluppato appositamente per bambini affetti da CVI, è costituito da un'immagine sullo schermo che si muove, e attende che il bambino tocchi sull'immagine stessa per fornire una risposta visiva (immagine che aumenta di dimensioni) e uditiva (riproduzione di un suono di approvazione come l'applauso).

2.3 Progetti esistenti



Figura 2.5: Esempio di schermata del gioco Bubbles Magic

Tale gioco risulta essere molto utile per lo sviluppo della coordinazione occhio-tocco, raramente presente nei bambini affetti da CVI. Inoltre, l'applicazione permette la modifica delle impostazioni di velocità, colore di sfondo, colore dell'immagine, per cercare di adattarsi il meglio possibile al bambino che la utilizza.

L'unico problema di questa applicazione è che tutte le combinazioni di colori possono essere utilizzate, e quindi, data la possibilità a chiunque di modificare le impostazioni di gioco, c'è il rischio che la combinazione di colori scelti (come ad esempio arancione e verde per i daltonici) possa peggiorare ancor di più le prestazioni del bambino e quindi la relativa riabilitazione. La Figura 2.6 mostra sia un esempio di schermata di gioco che di selezione delle impostazioni. Un altro esempio che si adatta invece maggiormente allo sviluppo

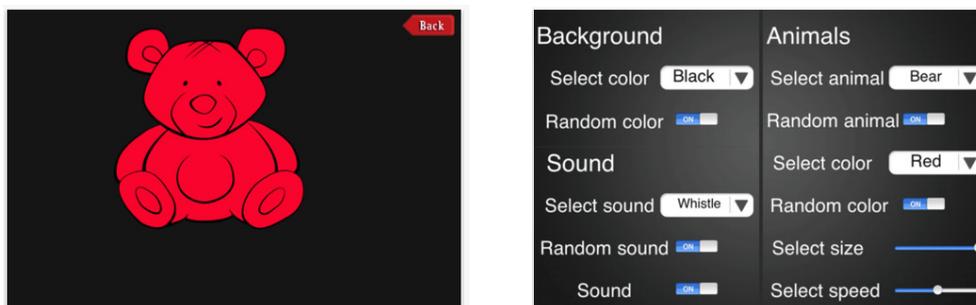


Figura 2.6: Schermata di gioco e delle impostazioni del gioco Tap-n-See Zoo

delle capacità dei bambini affetti da CVI, visto che solitamente vi sono altri danni cerebrali oltre quelli relativi alla vista, è *Baby Aquarium* sviluppato da *Kids Place* [23]. Tale gioco, ambientato in un acquario con dei pesciolini, vuole aiutare il bambino nel processo di apprendimento, in particolare dei colori e dei numeri. In tale ambientazione sottomarina, per ogni colore che si vuole insegnare al bambino vengono presentati due stimoli distinti: uno uditivo con lettura del colore, ed uno visivo, facendo nuotare un pesce del colore indicato.

Tale applicazione risulta essere maggiormente formativa rispetto a *Tap-n-see Zoo*, in quanto introduce un aspetto formativo per il bambino, non solo a livello di stimoli visivi o uditivi, ma anche di parole e concetti che quindi introducono in maniera molto semplice il bambino all'apprendimento. La Figura 2.7 mostra un esempio di schermata del gioco con presentazione del colore rosso insieme al pesce dello stesso colore. Questo tipo di analisi suggerì-

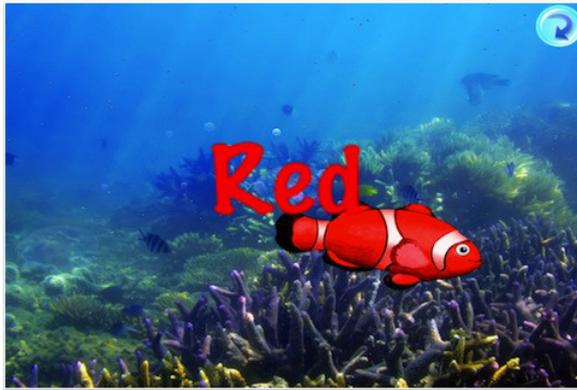


Figura 2.7: Esempio di schermata del gioco Baby Aquarium

sce quindi che, per cercare di ottenere risposte migliori da parte dei bambini e quindi anche un'efficacia migliore degli esercizi, è consigliabile costruire un'applicazione che sia prima di tutto naturale e facilmente utilizzabile da parte del bambino, e che sia in grado di catturare l'attenzione del bambino con suoni ed immagini.

Capitolo 3

Definizione del progetto

Cortical Visual Impairment (CVI) è la causa primaria di perdita della vista nella popolazione mondiale di bambini [26], e quindi qualsiasi strumento che possa aiutare i medici a diagnosticare e successivamente riabilitare bambini affetti da questo tipo di patologia può risultare molto utile e di aiuto per un numero considerevole di bambini.

Quello che si vuole fornire è uno strumento riabilitativo, ma anche di diagnosi e monitoraggio dell'andamento delle prestazioni del bambino, sotto forma di *serious game*, e quindi di videogioco, per riuscire ad ottenere una maggiore attenzione da parte del bambino, e quindi sperare in risultati migliori nel trattamento.

3.1 Descrizione generale

Il sistema che si vuole realizzare è costituito da due giochi distinti, uno proposto dalla Dott.ssa Pinello (Dipartimento di Salute della Donna e del Bambino) e l'altro proposto dalla Prof.ssa Nota e dalla Dott.ssa Sgaramella (Dipartimento di Psicologia Generale), con l'obiettivo di sviluppare nel bambino diverse capacità.

In entrambi i casi i due giochi dovranno poter essere usufruiti sia in ambiente ospedaliero, sotto la supervisione del medico che somministra il gioco al bambino per svolgere una visita, ad esempio per monitorare l'andamento dei progressi del bambino, che in ambiente casalingo, dove il bambino segue un percorso formativo di apprendimento svolgendo gli esercizi assegnati sotto la supervisione di un genitore.

In qualsiasi situazione i giochi vengano utilizzati, risulta essere fondamentale la registrazione e la successiva archiviazione delle prestazioni di ciascun bambino per ciascun gioco, questo per permettere il monitoraggio di eventuali

miglioramenti o peggioramenti del bambino nell'arco temporale del trattamento. Questo sistema di monitoraggio dovrà fornire sia un sistema di visualizzazione dei dati raccolti, per permettere al medico di fare le proprie analisi sulla base della propria esperienza, sia un sistema di valutazione delle prestazioni il più automatico possibile, in maniera tale da fornire una valutazione numerica oggettiva di quelli che sono i progressi del bambino.

La tipologia di esercizi da eseguire, le impostazioni di gioco e la durata della riabilitazione (oltre alla frequenza delle visite presso la struttura ospedaliera) sono di definizione da parte del medico e non oggetto di questa tesi, in quanto la risposta dei pazienti ai diversi stimoli a cui vengono sottoposti con il gioco possono avere risposte diverse, la cui comprensione esula dalle nostre competenze. L'unico obiettivo fondamentale è quello di fornire un sistema che sia il più personalizzabile possibile a seconda delle esigenze del medico e del bambino, in maniera tale da adattare ciascun esercizio/gioco in funzione delle caratteristiche del bambino.

3.2 Obiettivi dei giochi

In questa sezione verranno presentati i due giochi proposti, presentandone l'impostazione generale e quelle che sono le caratteristiche richieste dagli specialisti interessati allo sviluppo dei due giochi. Ognuno dei due giochi sviluppa delle capacità diverse, e sono orientati a bambini con difficoltà di livelli diversi. La complessità del gioco viene definita dal medico manualmente, e tale difficoltà verrà mantenuta fino a quando non vi sarà una modifica da parte del medico stesso. Ogni azione compiuta dal bambino sarà sempre seguita da un suono che descriva la valutazione dell'azione intrapresa, fornendo quindi un feedback immediato tale da incentivare il bambino a continuare nel gioco, ed eventualmente migliorare le proprie risposte ed i propri comportamenti nel caso in cui questi risultassero errati. Ovviamente, i suoni prodotti nel caso di risposte errate non devono essere suoni che comportino uno sconforto da parte del bambino, in quanto altrimenti si otterrebbe il risultato opposto rispetto a quello atteso, ovvero l'abbandono del gioco da parte del bambino perchè scoraggiato.

La definizione dei giochi da parte dei medici è stata a livello generale, descrivendone semplicemente quello che il sistema deve fare e quella che deve essere la risposta da parte del bambino. La costruzione del sistema, quale tecnologie utilizzare, come definire la grafica e altri aspetti tecnici sono stati oggetto dell'analisi e della progettazione del sistema.

3.2 Obiettivi dei giochi

3.2.1 Prendimi!

Il gioco ribattezzato *Prendimi!* è stato proposto dalla Dott.ssa Pinello del Dipartimento di Salute della Donna e del Bambino. Scopo del gioco è quello di sviluppare le capacità di focalizzazione, di coordinamento e di osservazione del bambino affetto da CVI.

In questo gioco vi è un'immagine posta all'interno dello schermo, che si muove da un punto di origine ad un punto di destinazione secondo una velocità definita, con l'intento di far seguire tale immagine dal bambino sia con gli occhi che con il tocco, fornito o tramite un dispositivo touch (che può essere o un tablet oppure uno schermo touch) oppure tramite click del mouse.

L'abilità che si vuole sviluppare con questo gioco è quello di riuscire a focalizzare la propria vista su un particolare punto, cosa molto difficile per bambini affetti da questo tipo di patologia, oltre a cercare di coordinare sia il contatto visivo con un oggetto sia l'azione di tocco, in quanto in questi bambini solitamente le due operazioni avvengono in momenti separati, di conseguenza il bambino per prima cosa instaurerà un contatto visivo con l'oggetto, successivamente tale contatto visivo verrà perso in favore però del raggiungimento tramite il dito dell'oggetto precedentemente osservato.

La variabilità dei problemi che possono affliggere i bambini richiedono una notevole quantità di impostazioni di personalizzazione, in particolare si vuole avere la possibilità di:

- personalizzare l'immagine utilizzata, per poter utilizzare soggetti o figure che possano attrarre maggiormente il bambino, come ad esempio il viso della mamma oppure un oggetto o giocattolo da lui conosciuto;
- personalizzare il colore di sfondo, vista la difficoltà di distinguere lo sfondo dall'immagine ed il contrasto dei colori;
- personalizzare i movimenti, in quanto diversi livelli di patologie possono compromettere diverse capacità di osservazione, come ad esempio vi possono essere bambini in grado di vedere solamente movimenti da destra a sinistra, solo alto verso il basso, o non avere una visione periferica e quindi necessitare di immagini che partono sempre dal centro;
- personalizzare la dimensione dell'immagine, in quanto livelli diversi di patologia possono influenzare più o meno gravemente la capacità visiva, in particolare degli oggetti più o meno grandi
- modificare la velocità, in quanto se si hanno risposte positive si vuole aumentare la difficoltà per continuare l'esercizio per il bambino e quindi non annoiarlo, mentre se le risposte sono negative si vuole cercare di

ridurre il senso di frustrazione nel bambino che non riesce a svolgere l'esercizio, partendo quindi da un livello più basso di riabilitazione.

Un esempio di personalizzazione per questo gioco è presentata in Figura 3.1, dove possiamo osservare come sia possibile modificare i valori delle impostazioni precedentemente elencate, facendo sì quindi che il gioco si adatti il più possibile al particolare bambino.



Figura 3.1: Fase di personalizzazione delle impostazioni del gioco *Prendimi!*

3.2.2 Aiutami!

In questo gioco, proposto dalla Dott.ssa Nota e dalla Dott.ssa Sgaramella del Dipartimento di Psicologia Generale, si vogliono allenare e sviluppare diverse capacità nei bambini affetti da CVI, non solamente quelle relative alla capacità di osservazione.

Il gioco prevede la definizione di un insieme di famiglie di oggetti, come ad esempio Animali, Fiori, Macchine ecc, e di una serie di immagini appartenenti a tali famiglie.

L'ambientazione descritta dalle specifiche parla di un elfo di Babbo Natale che deve mettere, all'interno di un sacco da preparare, solamente gli oggetti relativi ad una particolare famiglia indicata tramite audio all'inizio del livello. Ogni livello è caratterizzato da una sequenza di immagini di due tipi:

3.3 Ambienti di utilizzo

immagini target e immagini distrattori. Le immagini target sono le immagini che rappresentano oggetti appartenenti alla famiglia target del livello in questione, mentre le immagini distrattori rappresentano immagini che invece non appartengono alla famiglia specifica del livello, e che quindi non devono essere inserite nel sacco.

Il compito del bambino sarà quindi quello di comprendere la famiglia relativa al livello, riconoscere ogni singola immagine che viene proposta e decidere se tale immagine deve essere inserita nel sacco oppure eliminata perchè non appartenente alla famiglia specificata.

Rispetto al gioco precedente, questo è di un livello di difficoltà maggiore, in quanto le abilità che si vogliono sviluppare sono diverse e richiedono una maggiore attenzione da parte del bambino, in particolare capacità di giudizio, di comprensione e capacità di svolgere un'azione come quella di inserire delle immagini nel sacco, abilità che sicuramente richiedono molto esercizio. Per questo motivo, per cercare di massimizzare il tempo dedicato dal bambino al gioco, e rendere tale tempo il più proficuo possibile, è necessario introdurre, oltre ad un'ambientazione ed un'esperienza di gioco piacevole, anche una buona quantità di suoni che incentivino e premino il bambino a continuare a giocare, per riuscire quindi ad ottenere una maggiore attenzione ed un miglior livello di riabilitazione.

Anche in questo caso vi è la necessità di una personalizzazione molto accurata delle impostazioni di gioco, in particolare è necessario poter:

- Personalizzare il numero di livelli da giocare, la tipologia di livello e la sequenza di immagini target/distrattori, l'ordine di svolgimento dei livelli
- Modificare il tempo massimo atteso prima di passare all'immagine successiva

3.3 Ambienti di utilizzo

Lo sviluppo dell'applicazione e del sistema deve tenere in considerazione anche i contesti e gli ambienti all'interno dei quali tale applicazione verrà poi utilizzata. L'analisi ha individuato due ambienti fondamentali di utilizzo: lo studio medico oppure l'ambiente casalingo.

Studio medico

Lo studio medico, che verrà frequentato con una frequenza molto bassa da parte del bambino, circa una volta al mese, è l'ambiente più completo

all'interno del quale l'applicazione opererà, sia da un punto di vista tecnologico che da un punto di vista di apparecchiature. L'obiettivo principale degli esercizi nello studio medico, vista la scarsa frequenza delle visite, è quello di ottenere la maggior quantità possibile di informazioni, al fine di riuscire poi ad avere un quadro il più completo possibile della situazione del bambino.

Questo ambiente inoltre è quello su cui è possibile avere maggiore controllo, in quanto, grazie alla collaborazione con il medico, è possibile installare gli strumenti hardware e software necessari per poter utilizzare l'applicazione al completo. Per prima cosa vi è la necessità di avere a disposizione (almeno) un computer che sia in grado di connettersi al server ospitante l'applicazione, in maniera tale da costruire il sistema nella sua interezza.

Successivamente è necessario un browser recente, in quanto le funzionalità che verranno utilizzate per il progetto sono supportate solamente dalle versioni più recenti. Anche in questo caso, l'installazione del browser adatto non costituisce motivo di impedimento nello sviluppo.

Inoltre, all'interno dello studio medico è possibile realizzare un'ambientazione neutra, sufficientemente illuminata, tale da rendere possibile l'operazione di tracciamento del movimento degli occhi.

Affinchè tale operazione dia dei buoni risultati grazie ai quali poi effettuare diverse analisi, è infatti necessario che il bambino sia collocato in una posizione tale da avere dietro di sé sfondo neutro (come ad esempio il bianco), un'adeguata illuminazione, e la possibilità di installare il sistema hardware necessario per il sistema di tracciamento degli occhi.

Un'altra considerazione può essere fatta anche sullo schermo e sulla modalità di interazione disponibile. Quella più semplice, disponibile con qualsiasi computer, è la modalità che utilizza il mouse per selezionare gli oggetti sullo schermo. Tuttavia, nello studio medico è possibile considerare anche un'altra modalità di interazione, e cioè quella attraverso il tocco. Tale modalità può essere introdotta in diversi modi, ad esempio con un dispositivo tablet, che funziona naturalmente attraverso l'interazione con il touch, oppure attraverso i più recenti schermi touch, che sostituiscono l'interazione con il mouse con quella invece touch direttamente sullo schermo.

Nei bambini affetti da CVI, uno dei problemi che si riscontrano sono anche quelli relativi all'incapacità di vedere oggetti di piccole dimensioni. Per questo motivo, può essere utile installare presso lo studio medico uno schermo di dimensioni sufficientemente grandi, in maniera tale da poter ingrandire a piacimento la grandezza degli oggetti e renderli visibili ai bambini.

Questi requisiti sono facilmente soddisficibili nel caso dello studio medico, non essendoci particolari problemi nell'installazione di tutta la strumentazione necessaria, al meglio delle possibilità e delle richieste del sistema, permettendo quindi l'acquisizione della maggior quantità possibile di

3.3 Ambienti di utilizzo

informazioni.

Ambiente casalingo

L'ambiente casalingo è l'ambiente che sarà maggiormente visitato ed utilizzato dal bambino per i suoi esercizi quotidiani. Tuttavia tale ambiente risulta essere sicuramente più variabile rispetto allo studio medico, rendendo quindi molto difficile fare particolari assunzioni.

Per prima cosa, non è detto che sia sempre disponibile una connessione alla rete. Infatti, per i motivi più disparati, la connessione alla rete può non essere disponibile, e quindi non è chiaramente possibile fare affidamento su di essa per poter far funzionare l'applicazione.

Un altro problema è relativo al tracciamento degli occhi. Per poter fare questo tipo di acquisizione, è necessaria sia la strumentazione adatta in termini hardware, sia in termini di ambientazione. Per quel che riguarda l'ambientazione, è difficile ipotizzare l'esistenza, all'interno dell'ambiente casalingo, di una stanza con sfondo neutro e con l'illuminazione adatta per l'analisi. Oltre ad un problema di ambientazione, c'è anche un problema di attrezzatura. Come vedremo al capitolo 5, gli strumenti hardware per un buon tracciamento degli occhi sono molto costosi e delicati, e non è pensabile sia che l'azienda ospedaliera sia in grado di attrezzare i bambini in cura con i dispositivi necessari, sia che la famiglia sia in grado di sostenere la relativa spesa.

Per questi motivi, l'analisi del tracciamento degli occhi rispetto al gioco ed alle sue immagini, in ambiente casalingo non è praticabile.

Un altro problema riguarda il dispositivo utilizzato e le sue specifiche. Per prima cosa, non è detto che la famiglia abbia a disposizione un computer per poter utilizzare l'applicazione, oppure che il computer disponibile non sia in grado di far girare l'applicazione.

Questa possibilità permette di introdurre anche un'ulteriore situazione che si presenta all'interno dell'ambiente casalingo, ovvero la presenza di innumerevoli configurazioni di schermo dove tale applicazione viene eseguita, che possono essere più o meno grandi, con formato 16:9 oppure 4:3 ecc, senza dimenticare quelle situazioni in cui vengono utilizzati dei dispositivi tablet.

Non potendo conoscere la configurazione dell'ambiente casalingo, è necessario che l'applicazione sia in grado di funzionare sulla maggior quantità possibile di dispositivi, non imponendo quindi solamente alcune configurazioni, escludendo quindi di fatto tutte quelle famiglie non conformi alle specifiche imposte.

3.4 Eye-tracking

Come detto precedentemente, il sistema deve essere in grado di tracciare il movimento degli occhi durante i diversi esercizi. Questa informazione risulta essere molto importante perchè permette di capire qual è la capacità del bambino di osservare un'immagine.

Le informazioni che si possono ricavare con questa tipologia di analisi, in particolare permette di capire quanto il bambino è in grado di osservare un'immagine, se ci sono zone del campo visivo dove vi è assenza di capacità visiva, ecc.

Le modalità per il tracciamento degli occhi sono molteplici, tutte con costi e benefici e precisione diversa, ed a seconda dell'analisi che si vuole produrre è necessario scegliere accuratamente quale strumento utilizzare. Per un'analisi più approfondita di questa tematica, si rimanda al capitolo 5.

3.5 Analisi dei requisiti

I requisiti del progetto presentati sono emersi sia da un'analisi della descrizione dei giochi fornita dal committente dei giochi, sia da colloqui successivi, e ve ne sono sia di comuni a tutti e due i giochi che specifici per ognuno di essi. Ciascun requisito è classificato secondo la sua importanza, in particolare vi sono requisiti:

- *obbligatoria*: requisiti irrinunciabili per il committente;
- *desiderabili*: requisiti non strettamente necessari per il funzionamento del progetto, ma la cui presenza aumenta la qualità ed il valore del prodotto finale;
- *opzionali*: requisiti relativamente utili, la cui assenza non riduce la qualità del prodotto finale.

I requisiti saranno definiti nel modo seguente:

REQ_TIPOLOGIA_GIOCO_IMPORTANZA_NUMERO

dove:

- *TIPOLOGIA* indica la tipologia del requisito e può assumere valori *FUN* (funzionali), *NONFUN* (non funzionali) e *INT* (interfaccia);
- *GIOCO* indica per quale gioco tale requisito è definito, e può assumere valori *ALL* (per tutti i giochi), *CATCHME* (gioco *Prendimi!*) e *HELPME* (gioco *Aiutami!*);

3.5 Analisi dei requisiti

- *IMPORTANZA* indica se il requisito è obbligatorio (*OBB*), desiderabile (*DES*) o opzionale (*OPZ*).

3.5.1 Requisiti non funzionali

REQ_NONFUN_ALL_OBB_1 l'applicazione deve poter essere utilizzata dal maggior numero possibile di dispositivi, siano essi mobile o desktop, con il minor numero possibile di requisiti hardware o software.

3.5.2 Requisiti funzionali

REQ_FUN_ALL_OBB_1 deve essere presente un sistema di login che permetta di identificare il medico o il bambino e modificare il comportamento del sistema a seconda delle credenziali inserite.

REQ_FUN_ALL_OBB_2: l'interfaccia del medico deve essere diversa dall'interfaccia del bambino, permettendo al secondo solamente di giocare senza la possibilità di modificare eventuali impostazioni.

REQ_FUN_ALL_OBB_3: le prestazioni di gioco di ciascun utente devono essere salvate affinché siano consultabili in un secondo momento dal medico.

REQ_FUN_ALL_OBB_4: ogni bambino deve avere associato, per ciascun gioco, la propria particolare configurazione, in maniera tale da riproporla in un momento successivo

REQ_FUN_ALL_OBB_5: ogni nuovo livello costruito per ciascun bambino deve essere salvato e trasformato nel livello attuale, da utilizzare successivamente

REQ_FUN_ALL_OBB_6: il sistema deve tracciare l'andamento del tocco e della vista durante il gioco, e presentare i dati acquisiti al medico in un formato comprensibile per l'analisi, come grafici, rappresentazioni grafiche o tabelle riepilogative.

REQ_FUN_ALL_OBB_7: il sistema deve fornire una modalità di training per il sistema di eye-tracking, per poter calibrare al meglio il sistema senza che questo comporti un affaticamento per il bambino.

REQ_FUN_ALL_OBB_8: il sistema dovrà fornire un sistema di personalizzazione delle impostazioni di gioco, per rendere l'esperienza di gioco il più aderente possibile alle necessità dal bambino.

REQ_FUN_CATCHME_OBB_1: il sistema deve permettere la personalizzazione dei movimenti di gioco, della posizione di partenza, del colore di sfondo e del colore dell'immagine, della dimensione dell'immagine e della velocità di spostamento.

REQ_FUN_CATCHME_OBB_1: il sistema deve disporre di un insieme di immagini da utilizzare per lo spostamento, di forma, natura e colori diversi.

REQ_FUN_HELPME_OBB_1: il sistema deve permettere l'inserimento di nuovi livelli, la modifica dell'ordine delle immagini presentate, la modifica delle immagini, l'eliminazione di vecchi livelli e la modifica della durata massima dell'immagine sullo schermo.

3.5.3 Requisiti di interfaccia

REQ_INT_ALL_OBB_01 l'interfaccia dei giochi deve essere minimale, non troppo complicata per evitare che i bambini possano distrarsi con particolari e dettagli non importanti ai fini dell'esercizio.

REQ_INT_ALL_OBB_02 il layout dei giochi deve essere fluido, così da adattarsi a qualsiasi dimensione degli schermi su cui saranno utilizzati, e gestire eventi come il cambio di orientamento di un dispositivo tablet.

REQ_INT_ALL_DES_01 i giochi devono utilizzare dei suoni per accompagnare il bambino nel gioco e per fornirgli feedback riguardo le sue prestazioni.

3.6 Utenti coinvolti

Il sistema è utilizzabile dalle persone seguenti:

- *giocatore*: si tratta di un bambino, di età variabile compresa tra i 2 ed gli 8 anni affetto da CVI, che giocherà con i giochi costruiti
- *medico*: si occupa della gestione dei pazienti, in particolare della costruzione degli esercizi da somministrare al bambino e all'analisi dei dati raccolti
- *genitore*: si occupa di seguire il bambino a casa nello svolgimento degli esercizi quotidiani, preparando il dispositivo al bambino in maniera tale che possa poi essere utilizzato senza l'intervento del bambino stesso

3.7 Design del sistema

Una volta individuati i requisiti del sistema dalla descrizione del sistema, si è proceduto alla progettazione del sistema da realizzare, in particolare alla progettazione delle diverse componenti che lo compongono.

3.7 Design del sistema

3.7.1 Obiettivi Architettura

L'obiettivo principale dell'architettura del sistema è quella che viene definita "trasformazione elegante". Questo significa che le pagine rimangono accessibili nonostante possano verificarsi limitazioni nella fruizione delle pagine stesse, come ad esempio disabilità fisiche, sensoriali, oppure limitazioni tecnologiche. Questo significa quindi che il sistema deve rimanere accessibile indipendentemente dal dispositivo utilizzato per la fruizione.

L'ambiente ottimale per l'applicazione sarà lo studio medico, dove è possibile avere maggiore controllo per quel che riguarda l'apparecchiatura e l'ambientazione. Infatti, in questo caso vi sarà la disponibilità di effettuare eye-tracking attraverso l'apparecchiatura costruita e utilizzando un'ambientazione che faciliti tale operazione, come ad esempio l'uso di uno sfondo neutro ed il corretto orientamento delle luci.

Nel caso invece dell'ambientazione casalinga, è difficile fare considerazioni ed assunzioni di ogni genere. Per prima cosa, è chiaramente impossibile usare strumenti di eye-tracking, in quanto la strumentazione necessaria non è disponibile al bambino, ed inoltre anche se si volesse utilizzare ad esempio una webcam oppure la fotocamera frontale presente in diversi dispositivi tablet, le condizioni di luce, l'ambientazione, la distanza del bambino dal dispositivo sono tutti fattori che renderebbero impossibile il tracciamento degli occhi (o l'ottenimento di risultati in qualche modo sensati) e quindi è necessario che il sistema sia in grado di funzionare anche senza eye-tracker.

Un altro aspetto da considerare è quello relativo alla gestione del tocco. Anche in questo caso, le situazioni che si possono verificare sono molteplici. La prima situazione è quella in cui si ha a disposizione un dispositivo touch, che può essere o uno schermo touch oppure un tablet, ed un bambino in grado di utilizzare le mani per interagire con lo schermo. In questo caso sarà possibile rilevare le informazioni relative al tocco ed utilizzarle per analisi successive. Tuttavia, quest'ultima situazione può non verificarsi in tutte le condizioni possibili per diversi motivi. Per prima cosa, è possibile che uno schermo touch o un tablet non sia disponibile, e che quindi l'unica modalità di interazione possibile sia quella che prevede l'uso del mouse. In questo caso, poiché anche come viene utilizzato il mouse da parte del bambino può essere un'informazione importante, il sistema dovrà essere in grado di acquisire informazioni relative allo spostamento del mouse.

Inoltre, nei bambini affetti da CVI non è raro osservare, associate alla patologia relativa alla vista, altre patologie non relative all'apparato visivo ed alla capacità di elaborare le immagini. Infatti, essendo la CVI una malattia che colpisce il cervello, si verificano spesso anche altri problemi, ad esempio motori. Questo fa sì che i bambini abbiano anche problemi di movimento, e

quindi non siano in grado di muoversi, o di muovere correttamente mani e braccia.

Per questo motivo, l'applicazione deve essere in grado sia di funzionare senza la necessità del tocco o dell'uso del mouse, in maniera tale da allenare e riabilitare solamente l'apparato visivo nel caso in cui il bambino sia impossibilitato ad usare gli arti, oppure essere utilizzata anche come strumento per la riabilitazione delle capacità motorie del bambino, oltre a quelle visive.

3.7.2 Definizione Architettura

Sulla base delle considerazioni precedenti e dei requisiti precedentemente definiti, l'architettura del sistema è stata divisa in quattro parti, con la necessità che esse siano indipendenti tra loro:

- un server centrale, all'interno del quale è inserito il programma che si occupa di gestire tutti i pacchetti con le informazioni;
- il sistema che si occupa dell'eye-tracker, che gestisce l'analisi del movimento degli occhi;
- il client destinato al medico;
- il client destinato al bambino.

Questi quattro moduli sono stati definiti come indipendenti uno dall'altro per far sì che essi possano essere installati liberamente su quante macchine e su quali dispositivi si preferisce, al fine di garantire la maggior portabilità possibile del sistema. L'unico vincolo imposto è che il programma che funge da gestore dei pacchetti deve essere collegato su un server raggiungibile attraverso un indirizzo Internet. L'architettura generale può essere schematizzata come in Figura 3.2.

Server Centrale

Il server centrale, raggiungibile attraverso un indirizzo Internet, si occupa della gestione dei pacchetti che devono passare da un modulo all'altro. Inoltre, all'interno del server verrà installato il database per la memorizzazione delle informazioni utilizzate dal sistema. È infatti necessario tenere traccia di tutti i bambini che vengono seguiti dal medico, quali sono le loro caratteristiche e gli esercizi che hanno svolto, come sono stati valutati gli esercizi ecc.

3.7 Design del sistema

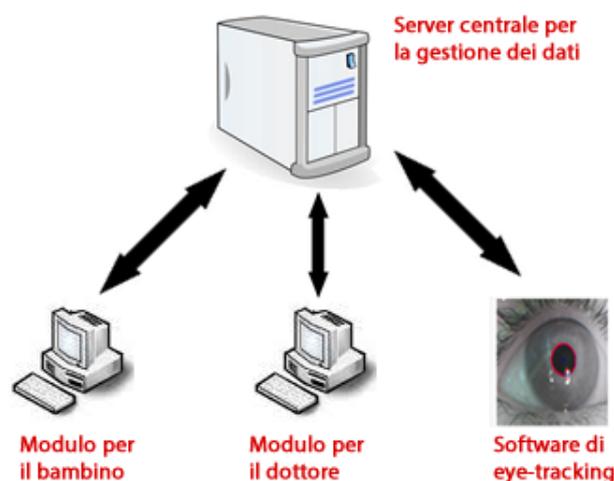


Figura 3.2: Esempio di architettura del sistema

Software eye-tracker

Il software di eye-tracker si occupa dell'analisi del movimento e della posizione degli occhi durante gli esercizi svolti presso lo studio medico.

Tale sistema si deve collegare al server tramite Internet, e deve svolgere tutti i task necessari ad un corretto tracciamento. Questo significa che per prima cosa si occuperà del procedimento di training, per potersi calibrare rispetto alle caratteristiche del bambino e della posizione degli occhi, dove viene richiesto al bambino di guardare in determinati punti dello schermo, e successivamente deve analizzare le immagini ricevute per capire dove il bambino guarda durante l'esercizio.

Particolare attenzione deve essere posta sulla gestione del procedimento di training. I software tradizionali, durante la fase di training, richiedono alla persona seduta di fronte alla telecamera di guardare sullo schermo dei punti, solitamente quattro / cinque relativi agli estremi dello schermo più il punto centrale, per una certa quantità di tempo. Tali software assumono però che il comportamento della persona sia di totale partecipazione, e che quindi il punto in cui sta guardando la persona è proprio quello indicato sullo schermo. Nel caso dei bambini affetti da CVI invece, tale assunzione non è più valida, o difficilmente lo è. Nei bambini infatti, soprattutto quelli affetti da CVI, il punto più problematico è quello di ottenere da parte loro attenzione ed interesse. Questo significa quindi che una semplice indicazione a guardare verso il punto dello schermo, indicato ad esempio con una croce, non è as-

solamente una modalità di approccio nei confronti del bambino che può portare alla sua attenzione, e quindi ad effettuare il training per il software. Per questi motivi, è necessario che la fase di preparazione del software venga proposta al bambino come forma di gioco o video, o qualsiasi altra modalità possibile, per far sì che il training sia il più preciso possibile, per ottenere i migliori dati possibili, ma che al tempo stesso non causi affaticamento o scarso interesse nei confronti del bambino.

Modulo del medico

Il modulo del medico gestisce l'interfaccia messa a disposizione per la gestione dei giochi.

Tale interfaccia deve essere il più semplice possibile, e guidare il medico nella costruzione degli esercizi per il bambino. Una volta impostati i parametri e iniziato l'esercizio, tale interfaccia deve fornire al medico un resoconto in tempo reale delle prestazioni del bambino, oltre a degli eventuali strumenti di controllo per la modifica sul posto degli esercizi proposti.

Inoltre, in un momento successivo a quello delle visite il sistema deve fornire la possibilità di visualizzare uno storico degli esercizi svolti dai bambini e delle loro prestazioni, corredate da immagini o informazioni aggiuntive tali da aiutare il più possibile il medico nella propria valutazione.

Modulo del bambino

Il compito del modulo dedicato al bambino è quello di interfacciarsi con il bambino, proponendo i diversi esercizi impostati secondo la volontà del medico.

Prima dell'esercizio vero e proprio, il client del bambino si occuperà anche di visualizzare a video i punti che il bambino dovrà osservare affinché il sistema di eye-tracking sia in grado di calibrarsi.

Una volta completata la fase di training, il client attenderà la comunicazione da parte del server per avviare l'esercizio, e quindi iniziare ad inviare le informazioni necessarie per l'analisi. Tali informazioni saranno relative sia agli elementi presenti sullo schermo, come ad esempio le immagini con cui il bambino si deve interfacciare, sia quelle eventualmente relative all'interazione che il bambino ha con tali immagini attraverso il mouse o il tocco.

Capitolo 4

Tecnologie di sviluppo

Come descritto nella sezione 3.3, l'uso sia di computer desktop che di dispositivi tablet, introduce il problema della gestione della modalità di interazione tra bambino e applicazione Web. Come osservato anche dalla società *Junior Blind of America*, la modalità di interazione con il tocco è, per i bambini più piccoli, la modalità più semplice che possono utilizzare, rispetto a quella con il mouse, che per i bambini può non essere immediata.

Inoltre, l'uso del tocco come modalità di interazione può risultare molto utile anche in situazioni in cui il bambino presenta altri problemi oltre a quelli visivi (possibilità non rara in questi bambini, visto che spesso i deficit coinvolgono diverse parti del cervello, e quindi anche parti di esso adibite al controllo di altri aspetti, come ad esempio il movimento degli arti). In questi casi l'uso del tocco può essere utile per un processo di riabilitazione o sviluppo del controllo degli arti. Risulta infatti essere molto più semplice del mouse, che richiede una coordinazione molto più complessa. Per questi motivi, l'inclusione del tocco come modalità di interazione è fondamentale per facilitare il gioco da parte del bambino.

Anche se la modalità di interazione preferita è il sistema touch, è chiaro come imporre all'intera applicazione quella sola modalità è riduttivo, con conseguente degrado della qualità e della fruibilità dell'applicazione stessa. Per questo motivo, nel momento in cui un'interazione touch non è possibile (per assenza del dispositivo, quindi tipicamente in un ambiente casalingo) è necessario definirne un'altra che possa sostituirla. La modalità più naturale e più facilmente comprensibile, utilizzabile con dispositivi desktop, è l'interazione con il mouse, che non è sicuramente così semplice come l'interazione touch ma rappresenta sicuramente il miglior compromesso raggiungibile.

Per questo motivo, la scelta della tecnologia da utilizzare deve tenere conto della necessità di definire queste due tipologie di interazione, facendole coesistere tra di loro senza che questo comporti una notevole quanto pesante

riscrittura di codice.

Per questi motivi, il miglior strumento attualmente disponibile per lo sviluppo di questa tipologia di applicazioni è il web: esso infatti può essere fruito indipendentemente dalla modalità di interazione che viene utilizzata, e quindi sia da dispositivi desktop con l'uso del mouse o da dispositivi touch, senza che questo comporti l'aggiunta di codice specifico per le due situazioni.

Inoltre, le applicazioni web hanno un'altra caratteristica fondamentale che risponde ad una necessità espressa precedentemente: se correttamente progettate e sviluppate, sono in grado di adattarsi a quasi tutte le risoluzioni e dimensioni dello schermo. Questa modalità di progettazione prevede l'utilizzo di layout fluidi, con direttive di presentazione espresse in forma relative anziché assolute (utilizzando quindi dimensioni espresse in % anziché in pixel), che fanno sì che il layout delle pagine si adatti in maniera automatica alla dimensione dello schermo su cui vengono visualizzate.

Anche se la scelta di un'applicazione web è quella più sensata per la tipologia di progetto che si vuole realizzare, è necessario tenere in considerazione come gli attuali browser presenti sui dispositivi mobile non siano maturi come quelli utilizzati in ambito desktop. Poiché l'introduzione del sistema anche all'interno dei dispositivi mobile è un requisito obbligatorio, è stato introdotto il framework di sviluppo PhoneGap [22] che, attraverso poche operazioni, permette di convertire un'applicazione web in un'applicazione nativa direttamente installabile sui dispositivi.

Inoltre, poiché una connessione ad Internet può non essere sempre disponibile, come ad esempio in un contesto casalingo con il bambino che svolge gli esercizi a casa, può essere necessario che l'applicazione funzioni anche senza essere collegata ad Internet. Per questo motivo, vi è la necessità di creare un'applicazione che sia in grado di funzionare anche in modalità offline, salvandosi tutte le informazioni necessarie e senza quindi necessità di recupero delle stesse da un server collegato in rete.

Sulla base di queste considerazioni, il progetto è stato sviluppato come applicazione web, o semplicemente come Rich Internet Application (RIA), che fornisce tutti gli strumenti necessari a soddisfare i diversi requisiti richiesti.

4.1 Rich Internet Application

Le Rich Internet Application (RIA) sono applicazioni web che utilizzano dati che possono essere elaborati sia dal server che dal client. Questi dati vengono scambiati in maniera asincrona in maniera tale che l'utente possa continuare la navigazione e la fruizione dell'applicazione web mentre quest'ultima si aggiorna o modifica parti della propria interfaccia grafica.

4.1 Rich Internet Application

Dal lato client, le RIA forniscono un look-and-feel molto simile alle applicazioni desktop, tanto che la parola “rich” indica proprio una marcata differenza con le applicazioni web precedenti. Le RIA sono inoltre caratterizzate da sistemi di controllo interattivi come menù o bottoni, la possibilità di utilizzare l’applicazione in modalità online o offline e l’uso della capacità di calcolo messa a disposizione sia dal client che dal server.

Per poter realizzare RIA, si sono inizialmente diffuse piattaforme realizzate appositamente per questo scopo, come Adobe Flash©, Silverlight© e JavaFX©. In tutti i casi però, questo tipo di tecnologie richiedono l’installazione lato client di plugin specifici in grado di interpretare la sequenza di istruzioni definite attraverso queste tecnologie.

A sostituzione di queste tecnologie sta però entrando in maniera prepotente nel mondo del web lo standard (non ancora completamente definito) HTML5. HTML5 aggiunge al linguaggio HTML precedentemente definito nuovi tag, alcuni che permettono di definire sia nuovi elementi strutturali (come i tag *nav*, *footer*, *header*¹ ecc) sia nuovi elementi più interattivi ed interessanti, come l’elemento *canvas*, *audio* o *video*. In congiunzione con CSS3 e JavaScript, HTML5 permette lo sviluppo di Rich Internet Application la cui portabilità è assicurata da tutti i browser che ne supportano le specifiche senza l’installazione di alcun plugin aggiuntivo.

È forse questo, attualmente, il maggior problema che affligge HTML5. Poiché le sue specifiche sono in costante evoluzione, essendo ancora in fase Draft per gli sviluppatori del W3C, anche il supporto alle diverse specifiche è limitato, e non fornito da tutti i browser presenti sul mercato. In particolare, il supporto è fornito a partire da Internet Explorer 9, Chrome 10, Firefox 3.6, Safari 5 e Opera 11.1, e attualmente le ultime versioni disponibili sul mercato sono Chrome 22, Firefox 15, Safari 6 e Opera 12. Internet Explorer rimane ancora fermo alla versione 9.

Anche se i diversi browser introducono quindi il supporto ad HTML5, questo non implica il fatto che tutte le funzionalità siano implementate, ma il supporto varia da browser a browser, come mostrato nel riepilogo in Figura 4.1 e 4.2.

¹La scelta di inserire questi nuovi elementi nasce da un’analisi effettuata da Google nel 2005 su miliardi di pagine della rete, per comprendere quali fossero i nomi assegnati alle classe degli elementi *div* maggiormente utilizzati, ottenendo che i nomi maggiormente utilizzati sono appunto *footer*, *nav* o *navigation*, *header*. Da qui la decisione di introdurre all’interno della specifica HTML5 tali elementi.

	WIN								
									
	FIREFOX	OPERA	CHROME	SAFARI	IE				
	11	11.61	18	5.1	6	7	8	9	
Local Storage	✓	✓	✓	✓	✗	✗	✓	✓	95%
Session Storage	✓	✓	✓	✓	✗	✗	✓	✓	95%
Post Message	✓	✓	✓	✓	✗	✗	✓	✓	97%
Offline Applications	✓	✓	✓	✓	✗	✗	✗	✗	85%
Workers	✓	✓	✓	✓	✗	✗	✗	✗	82%
Query Selector	✓	✓	✓	✓	✗	✗	✓	✓	94%
WebSQL Database	✗	✓	✓	✓	✗	✗	✗	✗	47%
IndexedDB Database	✓	✗	✓	✗	✗	✗	✗	✗	9%
Drag and Drop	✓	✗	✓	✓	✓	✓	✓	✓	95%
Hash Change (Event)	✓	✓	✓	✓	✗	✗	✓	✓	93%
History Management	✓	✓	✓	✓	✗	✗	✗	✗	64%
WebSockets	✓	✗	✓	✓	✗	✗	✗	✗	47%
Geolocation	✓	✓	✓	✓	✗	✗	✗	✓	88%
Touch	✗	✗	✗	✗	✗	✗	✗	✗	5%
File API	✓	✓	✓	✗	✗	✗	✗	✗	10%
meter element	✗	✓	✓	✗	✗	✗	✗	✗	6%
progress element	✓	✓	✓	✗	✗	✗	✗	✗	10%

Figura 4.1: Supporto delle features HTML5

4.2 HTML5

WIN									
									
	FIREFOX	OPERA	CHROME	SAFARI	IE				
	11	11.61	18	5.1	6	7	8	9	
Canvas	✓	✓	✓	✓	✗	✗	✗	✓	90%
Canvas Text	✓	✓	✓	✓	✗	✗	✗	✓	89%
SVG	✓	✓	✓	✓	✗	✗	✗	✓	88%
SVG Clipping Paths	✓	✓	✓	✓	✗	✗	✗	✓	88%
SVG Inline	✓	✓	✓	✓	✗	✗	✗	✓	54%
SMIL	✓	✓	✓	✓	✗	✗	✗	✓	69%
WebGL	✓	✓	✓	✓	✗	✗	✗	✓	61%
Audio	✓	✓	✓	✓	✗	✗	✗	✓	88%
Video	✓	✓	✓	✓	✗	✗	✗	✓	88%

Figura 4.2: Supporto per le feature grafiche di HTML5

Per quel che riguarda invece i browser mobile (quelli cioè installati su dispositivi mobile come tablet), il supporto è ancora ad un livello più basso rispetto a quello presente nei dispositivi desktop, con una quantità di funzionalità implementate ridotta. Inoltre, tali implementazioni possono differire molto da browser a browser, oppure produrre risultati non attesi, come nel caso della riproduzione di un audio che al momento risulta essere ritardata. Per questo motivo, si è reso necessario la trasformazione dell'applicazione web in un'applicazione nativa fruibile dai dispositivi mobile.

4.2 HTML5

HTML5 è un linguaggio di markup per le pagine web che si prefigge di aggiornare il linguaggio HTML introducendo nuovi costrutti e nuove funzionalità che possano essere utilizzati senza plugin o particolari installazioni aggiuntive da parte degli utenti (fatta eccezione per il browser che supporti tali specifiche).

La storia di HTML5 nasce nel 2004, quando il *Web HyperText Application Technology Working Group* (WHATWG) iniziò a lavorare su un nuovo standard HTML. Prima di tale data, lo standard HTML era alla versione 4.01, dalla quale venne poi costruito il linguaggio XHTML 1.0, dove X sta per

“eXtensible”, derivante proprio da XML. Il contenuto della specifica XHTML 1.0 era identica a quella dell’HTML 4.01, con la sola differenza che XHTML imponeva il rispetto delle specifiche XML, eliminando quindi le libertà nella scrittura di elementi ed attributi tipiche dell’HTML (quindi tag maiuscoli e minuscoli, spesso non chiusi, ecc).

Una volta completata la scrittura della versione 4 di HTML, il W3C si concentrò sulla definizione di un nuovo standard, XHTML 2, il cui obiettivo era quello di portare il web ad una nuova struttura XML. Tuttavia, tale standard non voleva essere retrocompatibile, bensì definire un nuovo linguaggio che sorpassasse quello precedente, con il risultato ovvio che tale operazione risultò un disastro.

Sebbene il W3C continuò nello sviluppo di tale linguaggio, anche se chiaramente senza futuro, dall’altro lato il WHATWG venne fondato da Ian Hickson, che si dedicò allo sviluppo della nuova versione dello standard, allo scopo di fornire la possibilità di sviluppare applicazioni web utilizzando HTML. A partire da tale data, il WHATWG iniziò lo sviluppo di tale standard, culminato nel 2008 con la pubblicazione della prima versione di HTML5.

Allo stato attuale, le specifiche HTML5 sono ancora in fase di sviluppo, anche se la maggior parte delle funzionalità sono implementate e supportate dalla maggior parte dei browser più recenti. Tra le diverse funzionalità e caratteristiche di HTML5, le più importanti sono:

- nuovi elementi specifici per la costruzione delle pagine, come *header*, *nav*, *footer*, ecc;
- nuovi elementi *audio* e *video* per la riproduzione di audio o video;
- supporto per la memorizzazione offline dei dati di navigazione;
- miglioramento delle form, introducendo nuovi controlli come *calendar*, *date*, *email*, *url*, *search* ecc;
- introduzione della possibilità di usufruire delle applicazioni in modalità offline.

Specificatamente per l’applicazione realizzata per questo progetto, si è fatto ricorso all’uso dell’audio, del sistema di disegno, della modalità offline dell’applicazione e lo storage locale dei dati.

Il tag canvas

Quando si definisce un elemento *canvas* all’interno di una pagina web, quello che viene costruita è un’area rettangolare dedicata al disegno, ma non

4.2 HTML5

solo.

Attraverso infatti opportuni comandi JavaScript è possibile manipolare il canvas, più in particolare quello che viene chiamato *context* del canvas, per poter rappresentare linee, forme geometriche, testo, immagini, gradienti ecc. L'elemento canvas è supportato dalla quasi totalità dei browser attualmente sul mercato, ed il suo impiego più profondo è nell'ambito nei videogiochi, dove attraverso delle librerie di supporto e semplici comandi è possibile rappresentare ambientazioni complesse. Tuttavia, il problema con questo elemento deriva dal fatto che ogni qual volta si vuole modificarne il contenuto, quello che succede è che viene ridisegnato completamente tutto, e non solo la parte modificata. Per questo motivo, per realizzare animazioni e ambientazioni complesse è necessario ricorrere a librerie di supporto come ad esempio **EaselJS**, che rappresentano gli elementi disegnati all'interno del *canvas* come una gerarchia di elementi, modificando solamente quelli che devono essere modificati e lasciando inalterati tutti gli altri.

Il tag audio

Il tag *audio* introduce la possibilità di riprodurre all'interno di una pagina HTML dei suoni. I formati audio (o video in caso di necessità) che possono essere riprodotti sono molteplici, ma il supporto da parte dei browser è molto variabile.

WIN									
									
	FIREFOX	OPERA	CHROME	SAFARI	6	7	8	9	
Audio: ogg/vorbis	✓	✓	✓	✗	✗	✗	✗	✗	72%
Audio: mp3	✗	✗	✓	✓	✗	✗	✗	✓	45%
Audio: wav	✓	✓	✓	✓	✗	✗	✗	✗	76%
Audio: AAC	✗	✗	✓	✓	✗	✗	✗	✓	45%

Figura 4.3: Formati supportati dai vari browser per il tag audio

Come possiamo osservare nella Figura 4.3, volendo supportare il maggior numero di browser possibili, non esiste un formato audio che sia riproducibile con tutti i browser in circolazione. Il motivo di questo non supporto

completo a tutti i formati audio nasce dal fatto che anche se il formato MP3 è il formato più diffuso ed utilizzato, esso non è un formato aperto. Ciò significa quindi che se un browser vuole supportare la riproduzione di file audio in formato MP3, la società sviluppatrice è costretta a pagare per tale possibilità. Per società quindi come Adobe, Apple o Google tale pagamento non ha chiaramente ripercussioni, mentre nel caso di Firefox tale richiesta economica non è supportabile. Per questo motivo, al formato MP3 è stato affiancato un altro formato, questa volta aperto chiamato Ogg Vorbis, che Firefox è in grado di riprodurre. Però Safari no. Fortunatamente, vi è la possibilità di specificare, per ogni audio presente nella pagina e tramite il tag *source* inserito all'interno di ciascun tag *audio*, i diversi file in formati diversi che devono essere riprodotti a seconda di quelli che sono supportati dal browser. Questo significa che codificando il file audio in formato .mp3 ed in formato .ogg, si ha una copertura completa dei browser che supportano il tag *audio*.

La necessità di un supporto totale al tag *audio* risulta molto importante, quasi fondamentale, per i *serious games* e per il progetto, in quanto il coinvolgimento del bambino anche attraverso stimoli uditivi, sia di accompagnamento al gioco che di incentivo o di approvazione per le operazioni svolte, è fondamentale per la buona riuscita della terapia riabilitativa.

Applicazioni offline

Un'altra funzionalità introdotta da HTML5 utilizzata nel progetto è la fruizione offline dell'applicazione web. Questa funzionalità viene introdotta da due elementi fondamentali: il file *manifest* e l'oggetto DOM *window.applicationCache*. Questi due elementi, combinati tra loro, permettono la fruizione di un'applicazione web anche se non si è connessi ad Internet, come avviene per qualsiasi altra applicazione installata ad esempio nei dispositivi tablet.

Il file *manifest* contiene un elenco di tutte i file, siano essi file html, JavaScript o semplici immagini, indispensabili affinché l'applicazione possa funzionare, mentre l'oggetto *window.applicationCache* permette di gestire gli eventi e lo stato della cache. Quando ci si collega per la prima volta all'applicazione web tramite un browser che supporta questa funzionalità, il file *manifest* viene scaricato, letto, e tutte le risorse indicate su tale file vengono scaricate e memorizzate lato client. Quando a questo punto ci si collega nuovamente a tale applicazione, i possibili scenari sono due: se non si è collegati ad Internet, l'applicazione viene caricata utilizzando i file precedentemente scaricati; in caso contrario invece, per prima cosa vengono caricate le risorse presenti in

4.2 HTML5

cache, e successivamente viene controllato il file *manifest*. Se il file presente sul server è diverso da quello in possesso del browser, la nuova versione viene scaricata contestualmente ai nuovi file, svolgendo comunque tali operazioni in background in maniera tale da non influenzare in maniera significativa le prestazioni dell'applicazione.

Come possiamo vedere, questa possibilità risulta essere molto utile per la tipologia di progetto sviluppato, in quanto fornisce ai genitori ed al bambino la totale libertà di svolgimento degli esercizi, non imponendo quindi una connessione ad Internet per poter utilizzare l'applicazione. Ovviamente, solamente collegando nuovamente il dispositivo alla rete sarà possibile scaricare nuovi aggiornamenti, come ad esempio nuovi parametri di gioco eventualmente modificati dal medico.

localStorage e sessionStorage

Un'ulteriore feature introdotta da HTML5 è quella relativa alla memorizzazione dei dati lato client. Con le versioni precedenti, l'unica possibilità a disposizione per memorizzare delle informazioni lato client è quella di utilizzare i cookie, piccoli file memorizzati dal browser sul computer dell'utente che vengono solitamente utilizzati per memorizzare semplici informazioni come ad esempio login e password dell'utente. HTML5 introduce un'evoluzione a questa tipologia di memorizzazione introducendo due distinte modalità: *localStorage* e *sessionStorage*.

sessionStorage e *localStorage* sono delle coppie <chiave, valore> che permettono di memorizzare, lato client, diverse informazioni tramite il metodo *set(chiave, valore)*, per poi essere recuperate attraverso il metodo *get(chiave)*. La differenza fondamentale tra le due modalità è che mentre il *localStorage* non ha data di scadenza delle informazioni, e quindi le informazioni salvate rimangono disponibili fino a quando non vengono effettivamente eliminate, *sessionStorage* elimina i dati non appena la sessione finisce. Rispetto ai cookie, questa tipologia di memorizzazione offre notevoli vantaggi, sia per quel che riguarda le prestazioni, essendo molto più veloce rispetto ai cookie, sia per quel che riguarda la sicurezza, ed infine per la quantità di informazione che può essere memorizzata. Con questa modalità di memorizzazione infatti, non vi sono limiti alle informazioni salvate, e quindi la quantità memorizzata con questa modalità può essere estremamente grande senza che questo influenzi le performance dell'applicazione web (i cookie fornisce una possibilità di memorizzazione di massimo 4KB). Infine, nel caso di memorizzazione tramite cookie le informazioni salvate vengono spedite ad ogni richiesta effettuata al server, mentre nel caso di *sessionStorage* e *localStorage* le informazioni ven-

gono inviate solo quando esplicitamente richieste).

File API

Un'evoluzione di *localStorage* e *sessionStorage* sono le File API. Questa funzionalità introdotta da HTML5 supera un'altra limitazione che affliggeva le applicazioni precedenti, e cioè l'impossibilità di salvare file lato client. Sebbene non vi sia comunque la possibilità di salvare dei file direttamente sul disco del computer dell'utente (per ovvie ragioni di sicurezza) quello che introducono i browser in congiunzione al supporto per le File API è una sandbox che funge da disco, all'interno della quale è possibile inserire qualsiasi elemento verrebbe comunemente salvato in qualsiasi disco, quindi file (testuali e non), cartelle, immagini ecc. Le possibilità di utilizzo sono molteplici, in particolare queste API possono essere utilizzate per:

- salvare temporaneamente una directory che si vuole caricare sul server, per poi spedire i file un pezzo alla volta;
- salvare localmente file compressi, per poi espanderli in un secondo momento riducendo la quantità di dati che è necessario trasferire;
- scaricare file di grosse dimensione, come file video, per poterli guardare in un secondo momento;
- scaricare allegati di posta per poterli poi aprire e visualizzare in un secondo momento.

Attualmente, solo Google Chrome supporta questo tipo di feature, e sarà quindi necessario ridurre al minimo la necessità di tale implementazione per non ridurre la portabilità dell'applicazione ad un singolo browser.

Il protocollo WebSocket

L'ultima evoluzione inserita da HTML5 utilizzata per il progetto è il protocollo di comunicazione HTML5 *WebSocket*. Questo protocollo permette di definire un canale di comunicazione di tipo full-duplex² che opera su un singolo canale di connessione. Questa tipologia di comunicazione risulta essere molto utile soprattutto per applicazioni web di tipo real-time.

²Con il termine full-duplex si indica una modalità di trasmissione e ricezione che ha caratteristiche di completa bidirezionalità, ossia che permette la comunicazione in entrambe le direzioni simultaneamente, come ad esempio nella rete cellulare.

4.2 HTML5

Per capire la vera utilità dei *WebSocket* facciamo un esempio. Quando un browser visita una pagina web, esso invia al server una richiesta HTTP che ospita tale pagina, che restituisce la risposta al client. Per applicazioni real time come ad esempio quelli relativi alla borsa, alle news, o alla vendita dei biglietti, per simulare il real time (evitando quindi di dover aggiornare a cadenze regolari la pagina per fare una nuova richiesta) l'applicazione esegue il *polling*, ovvero l'invio di richieste HTTP a intervalli regolari, ricevendo ogni volta immediata risposta da parte del server. Questa strategia, che è sicuramente una buona soluzione se si conosce a priori la frequenza di aggiornamento delle informazioni, ma per applicazioni real-time, dove tale scadenza non è nota a priori, il rischio è di avere una notevole quantità di messaggi spediti inutilmente che occupano la rete, con comunicazioni aperte e chiuse continuamente nel caso di situazioni con message-rate basso.

L'evoluzione proposta per il *polling* è il *long-polling*, dove in questo caso il browser invia una richiesta ed il server risponde (entro un certo tempo limite) solo se vi sono delle notifiche da spedire. Tuttavia, il *long-polling* non fornisce questo miglioramento nelle performance che ci si potrebbe attendere rispetto al *polling*, soprattutto in presenza di una notevole quantità di messaggi.

L'ultima possibilità precedente è l'uso dello *streaming*, dove in questo caso il browser invia una richiesta, ed il server risponde mantenendo aperta la connessione e spedendo nuove informazioni ogni qual volta vi sono degli aggiornamenti. Il problema di questo approccio risiede però nel fatto che lo *streaming* è incapsulato all'interno delle richieste HTTP, e questo fa sì che i firewall possano decidere di bufferizzare le risposte, aumentando quindi la latenza, che in un'applicazione real-time non è chiaramente accettabile.

Quello che fanno i *WebSocket* è migliorare tutti i problemi che caratterizzano le modalità precedenti di simulazione del real-time.

Il protocollo *WebSocket* è stato definito per poter funzionare bene con l'infrastruttura Web corrente. Come parte di questo principio, la specifica del protocollo definisce che una connessione *WebSocket* inizia ad esistere come una semplice connessione HTTP, in maniera tale da garantire retrocompatibilità completa. Il passaggio di protocollo da HTTP a *WebSocket* viene identificato con il termine di *WebSocket handshake*.

Il browser invia al server una richiesta al server, indicando la sua volontà di passare da protocollo HTTP a *WebSocket*. Questa volontà viene espressa dal client attraverso l'*Upgrade Header* (Figura 4.4).

```
GET ws://echo.websocket.org/?encoding=text HTTP/1.1
Origin: http://websocket.org
Cookie: __utma=99as
Connection: Upgrade
Host: echo.websocket.org
Sec-WebSocket-Key: uRovscZjNol/umbTt5uKmw==
Upgrade: websocket
Sec-WebSocket-Version: 13
```

Figura 4.4: Prima fase del WebSocket handshake tra client e server

Se il server è in grado di comprendere ed utilizzare il protocollo *WebSocket*, attraverso l'*Upgrade header* risponde in maniera affermativa al cambio di protocollo (Figura 4.5).

```
HTTP/1.1 101 WebSocket Protocol Handshake
Date: Fri, 10 Feb 2012 17:38:18 GMT
Connection: Upgrade
Server: Kaazing Gateway
Upgrade: WebSocket
Access-Control-Allow-Origin: http://websocket.org
Access-Control-Allow-Credentials: true
Sec-WebSocket-Accept: rLHCkw/SKsO9GAH/ZSFhBATDKrU=
Access-Control-Allow-Headers: content-type
```

Figura 4.5: Risposta positiva del server alla richiesta di cambio protocollo da HTTP a WebSocket

A questo punto, la connessione HTTP viene chiusa, e sostituita da una connessione *WebSocket* costruita sopra la stessa connessione TCP/IP. Ma quanto elevata è l'efficienza della comunicazione tramite *WebSocket*? Per farlo, confrontiamo la stessa applicazione realizzata sia tramite polling che tramite *WebSocket*, dove il polling viene fatto ogni secondo, e supponiamo che ad accedere a tale applicazione accedo 1.000, 10.000 e 100.000 client contemporaneamente. Nel caso di richieste HTTP, l'header contiene informazioni aggiuntive, che non riguardano cioè dati, che si è misurato sperimentalmente essere pari a 871 bytes [30]. Nel caso quindi delle richieste HTTP:

- **Caso 1:** 1,000 client che effettuano polling ogni secondo: il traffico di rete è $(871 \times 1,000) = 871.000$ byte = 6.968.000 bit per second (6,6 Mbps)
- **Caso 2:** 10.000 client che effettuano polling ogni secondo: il traffico di rete è $(871 \times 10.000) = 8.710.000$ byte = 69.680.000 bit per second (66 Mbps)

4.2 HTML5

- **Caso 3:** 100.000 client che effettuano polling ogni secondo: il traffico di rete è $(871 \times 100.000) = 87.100.000$ byte = 696.800.000 bit per second (665 Mbps)

Come è facile notare, la quantità di overhead introdotto è tremendamente elevato. Nel caso dei *Websocket* invece, che introducono solamente due byte di overhead nei propri frame [30], abbiamo che:

- **Caso 1:** 1.000 client che ricevono un messaggio al secondo: il traffico di rete è $(2 \times 1.000) = 2.000$ byte = 16.000 bit per second (0,015 Mbps)
- **Caso 2:** 10.000 client che ricevono un messaggio al secondo: il traffico di rete è $(2 \times 10.000) = 20.000$ byte = 160.000 bit per second (0,153 Mbps)
- **Caso 3:** 100.000 client che ricevono un messaggio al secondo: il traffico di rete è $(2 \times 100.000) = 200.000$ byte = 1.600.000 bit per second (1,526 Mbps)

Il risultato è quindi che i WebSocket riducono in maniera drammatica la quantità di overhead rispetto a quella introdotta invece dal *polling*, come mostrato anche in Figura 4.6.

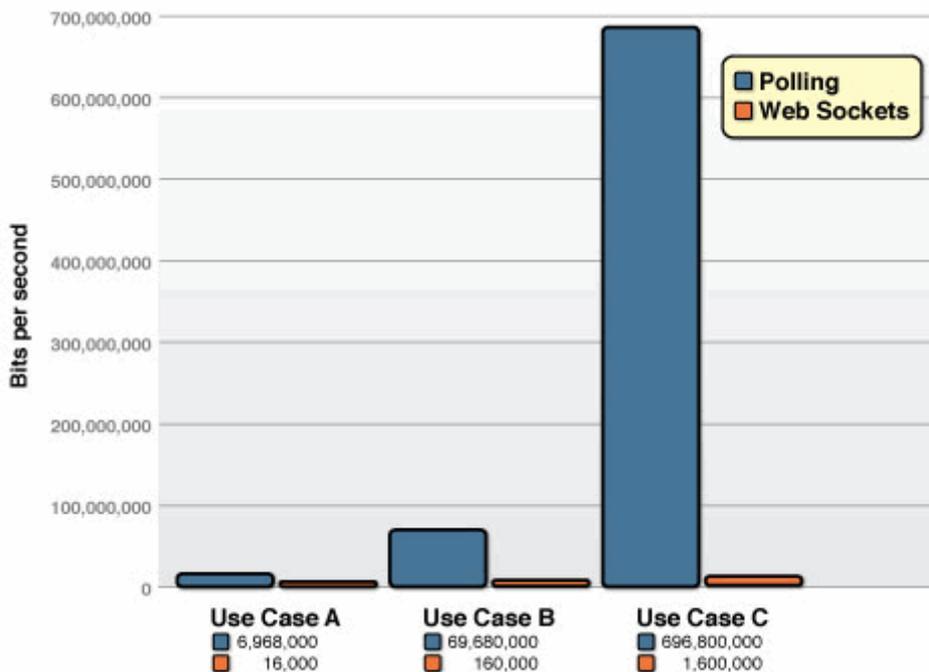


Figura 4.6: Confronto tra l'overhead introdotto tra polling e WebSocket

Come possiamo osservare, la quantità di bit al secondo necessari ai due protocolli per gestire tutte le connessioni entranti non sono nemmeno paragonabili, con il protocollo *WebSocket* che riduce di più del 90% la quantità di dati che devono essere spediti, con conseguente miglioramento delle prestazioni che possono essere raggiunte.

Per quel che riguarda invece la latenza, se assumiamo ad esempio che sono necessari 50ms per un messaggio per andare dal server al client, un'applicazione che utilizza il *polling* introduce una notevole quantità di latenza, in quanto la nuova richiesta può essere inviata solo quando la risposta precedente è stata ricevuta. Nel caso invece dei *WebSocket*, nel momento in cui è stata effettuata la connessione, le risposte possono arrivare al client senza che questo comporti una nuova richiesta da parte del client. La Figura 4.7 mostra le due situazioni proposte.

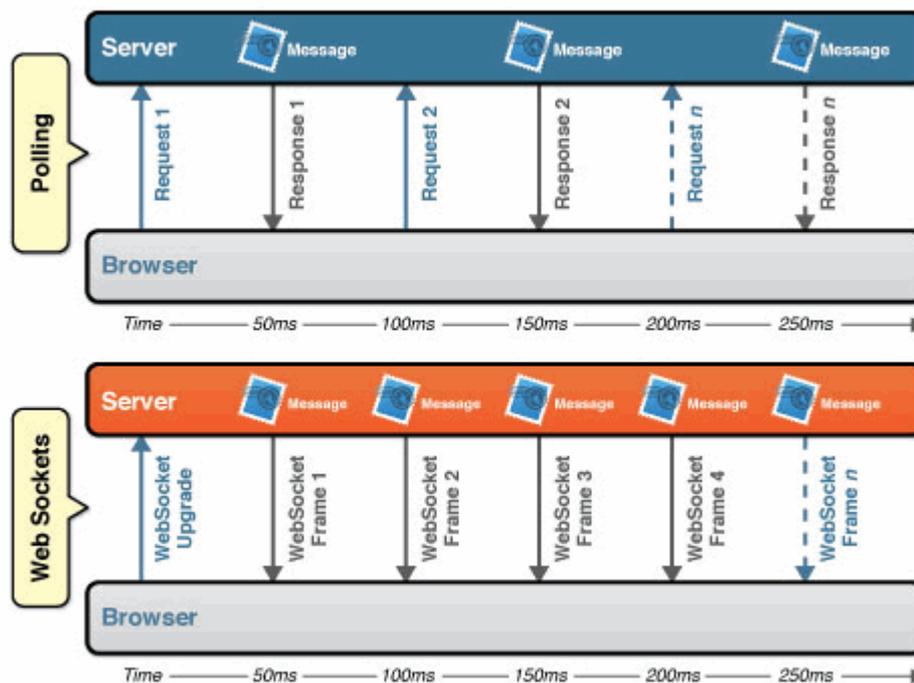


Figura 4.7: Confronto tra la latenza introdotta dal polling e da WebSocket

Nel caso del *polling*, il browser è costretto ad attendere la risposta da parte del server prima di poter fare una nuova richiesta. Dall'altro lato, il protocollo *WebSocket* non necessita di una risposta da parte del server, ma una volta che il protocollo ha subito l'upgrade a *WebSocket*, il server invierà dati al browser ogni qualvolta questi risultino essere disponibili. Per quel che

4.3 CSS3

riguarda il supporto ai *WebSocket*, Google Chrome e Firefox li supportano dalla versione 4, Safari dalla versione 5 mentre Internet Explorer e Opera al momento non implementano ancora tale soluzione.

4.3 CSS3

I fogli di stile CSS vengono utilizzati contestualmente alle pagine HTML, XHTML e XML per fornire supporto alla presentazione degli elementi, e quindi separare il contenuto dalla presentazione.

La nuova versione, ancora in fase di sviluppo, dei fogli di stile è la versione CSS3, che introduce diverse novità alla versione precedente.

In particolare, è stato introdotto:

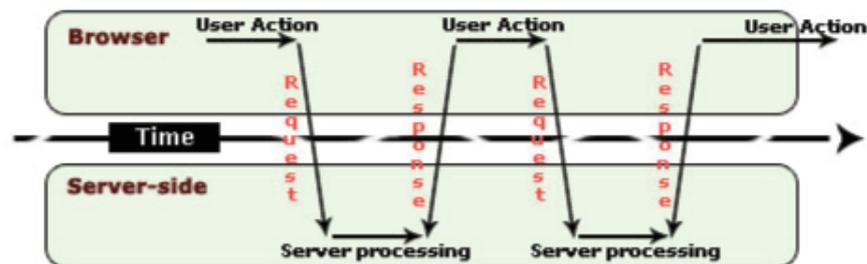
- Animazioni
- Transizioni di stile
- Trasformazioni 2D e 3D, come rotazione, ingrandimento, inclinazione
- Ombreggiatura degli elementi
- Bordi arrotondati
- Miglioramento della possibilità di personalizzazione del background degli elementi

Le funzionalità utilizzate all'interno del progetto sono le prime due, ovvero animazioni e transizioni di stile, che permettono di modificare lo stato degli elementi di una pagina senza necessariamente richiedere l'uso di JavaScript. Poiché le specifiche CSS3 sono ancora in fase di *Draft*, e quindi non completamente definite e soggette a possibili cambiamenti, non tutti i browser implementano le diverse proprietà in maniera completa, e anche l'implementazione può essere diversa. Per questo motivo quindi è necessario, quando si ricorre a regole CSS3, aggiungere prima della proprietà base un prefisso, in maniera tale che il browser utilizzi la proprietà in esso definita. Quindi, se vogliamo inserire la proprietà *transition* all'interno di un foglio di stile, dovranno necessariamente essere incluse anche le proprietà *-moz-transition* (per Firefox), *-webkit-transition* (per Google Chrome e Safari), *-o-transition* (per Opera) e *-ms-transition* (per Internet Explorer, anche se in questo ultimo caso il supporto alle regole CSS3 è ancora molto basso).

4.4 AJAX

AJAX, Asynchronous JavaScript and XML, è una tecnica di sviluppo per la realizzazione di applicazioni web interattive, ossia Rich Internet Application. Prima dell'introduzione di AJAX, ogni qual volta un browser voleva scambiare dati con il server per aggiornare ad esempio il contenuto di una pagina, era necessario che tale pagina venisse ricaricata totalmente. L'idea invece che sta alla base di AJAX è quello di introdurre lo scambio di dati tra browser e server in background, in maniera tale da permettere l'eventuale aggiornamento delle informazioni presentate in una pagina senza che questo comporti la necessità di aggiornare la pagina stessa. Inoltre, la chiamata verso il server effettuata tramite AJAX non è una chiamata bloccante, ma lascia libertà all'utente di eseguire altre operazioni in attesa di una risposta.

None-Ajax Request



Ajax Request

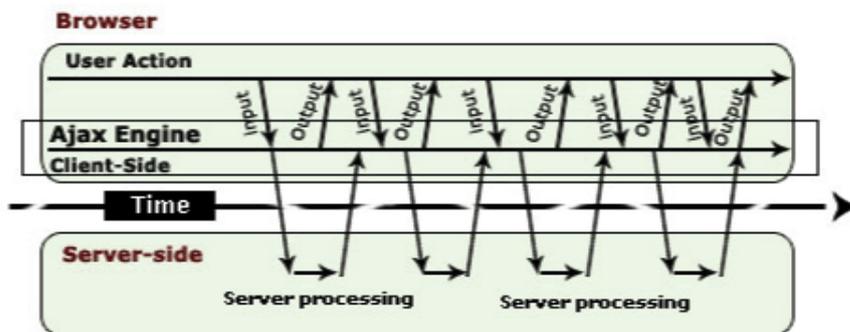


Figura 4.8: Confronto tra il pattern di interazione classico sincrono con quello AJAX di tipo asincrono

4.4 AJAX

La Figura 4.8 mette a confronto delle due situazioni. La parte superiore dell'immagine rappresenta la situazione prima dell'introduzione di AJAX, o di tutte quelle pagine che non utilizzano tale tecnologia. Ogni qualvolta il browser effettua una richiesta, attende la risposta da parte del server, senza lasciare possibilità al client di eseguire altre operazioni contemporaneamente, ed una volta ricevuta la risposta da parte del server aggiornerà la nuova pagina ricaricandola.

Nel caso invece di richieste effettuate attraverso AJAX, una volta inviata tale richiesta al server, il browser non attende la risposta bloccandosi, ma permette all'utente di eseguire altre operazioni, come ad esempio continuare la navigazione della pagina. Una volta ottenuta dal server una risposta, il motore AJAX si occuperà di aggiornare la pagina con le informazioni ricevute, senza che però questo comporti che la pagina debba essere ricaricata. AJAX è attualmente supportato da tutti i browser, e risulta essere indipendente dalla piattaforma e dal browser di utilizzo in quanto è basato sull'oggetto *XMLHttpRequest*, implementato da ogni browser attualmente in uso (eccetto per Internet Explorer 5 e 6 che utilizza l'oggetto ActiveX, mentre le versioni successive implementano l'oggetto *XMLHttpRequest*)³.

L'oggetto *XMLHttpRequest* è alla base dell'intero protocollo AJAX, in quanto esso è alla base dello scambio di dati tra client e server. Ogni volta che il client lo richiede, viene creato un oggetto *XMLHttpRequest* che si occupa di inviare una richiesta HTTP al server, spedendo eventualmente anche delle informazioni o parametri aggiuntivi attraverso le due modalità GET o POST. Una volta che il server riceve tale richiesta, essa viene elaborata e rispedita al browser, che avrà il compito di interpretare la risposta ricevuta ed eseguire le operazioni relative.

In congiunzione ad AJAX viene utilizzato, ed è stato utilizzato anche per questo progetto, il formato JSON [15]. Il formato JSON è un formato molto leggero per lo scambio dei dati, molto facile da utilizzare per i diversi dispositivi (non solo browser Web, ma anche applicazioni Java o C#) attraverso due semplici metodi di codifica e decodifica (*JSON.stringify()* e *JSON.decode()* nel caso di JavaScript).

JSON è costruito su due strutture:

- Una collezione di coppie nome(chiave)/valore
- Una lista ordinata di valori, vale a dire array o liste o vector

³L'oggetto *XMLHttpRequest* è attualmente solamente uno standard *de facto*, per via del fatto che tutti i browser implementano tale oggetto. Tuttavia, l'oggetto *XMLHttpRequest level 2* è attualmente in *Working Draft* da parte del W3C, che sta quindi procedendo ad una standardizzazione di tale oggetto

Attraverso questo formato, lo scambio di dati può essere effettuato facilmente tra le diverse applicazioni, anche di linguaggi differenti, in quanto il formato è universalmente riconosciuto e gestibile dalle diverse macchine, e soprattutto perchè sostituisce l'altro formato di scambio dati, e cioè XML, riducendone notevolmente l'overhead introdotto da XML per la propria configurazione.

4.5 Librerie di supporto

Per sviluppare il progetto, sono state utilizzate anche delle librerie aggiuntive allo scopo o di migliorare la gestione delle pagine web e degli elementi, oppure introdurre nuove funzionalità non presenti nativamente in JavaScript o in HTML5.

jQuery e **jQueryUI** [13] [14] sono due librerie appartenenti alla stessa famiglia, dove la prima semplifica notevolmente la gestione e la manipolazione del DOM, la gestione degli eventi, le animazioni e chiamate di tipo Ajax, mentre la seconda introduce nuovi elementi grafici come bottoni, temi, messaggi di dialogo, slider, ecc.

Modernizer [19] è una libreria che semplifica notevolmente il testing delle funzionalità HTML5 e CSS3 del browser utilizzato dall'utente. L'idea alla base di questa libreria è quello di effettuare dei test creando degli elementi con determinate proprietà, e a seconda della risposta ricevuta dal browser viene compreso se viene fornito il supporto a tali funzionalità o meno. Anche se la libreria mette a disposizione test per tutte le funzionalità offerte da HTML5 e CSS3, l'uso consigliato è quello di testare solamente le funzionalità di cui vi è effettivamente necessità, alleggerendo quindi la dimensione del file da includere per i test, e migliorando anche l'efficienza del software, in quanto la pesantezza dei test da eseguire ogni volta è sicuramente minore.

Flot [8] è una libreria costruita per disegnare grafici, utilizzata in congiunzione con jQuery. La libreria mette a disposizione una notevole quantità di grafici tra cui scegliere, come ad esempio torte, grafi a colonne, a righe ecc. Tramite ulteriori plugin sviluppati, è possibile modificare come vengono segnalati i punti sul grafo, dare dei nomi agli assi, abilitare le opzioni di zoom o di scorrimento del grafo. Tale libreria è stata scelta rispetto agli altri concorrenti perchè risulta essere molto leggera, semplice da usare e la gestione di una notevole quantità di dati non produce un degrado nella velocità di risposta e di calcolo.

4.6 Tecnologie lato server

Una volta descritte quelle che sono state le scelte tecnologie fatte per la parte del client del progetto, è stato necessario definire gli strumenti tecnologici utilizzati nel server.

4.6.1 PHP

PHP (PHP: Hypertext Processor) è un linguaggio interpretato utilizzato maggiormente per la costruzione di applicazioni web lato server. PHP all'interno del progetto è stato utilizzato sia per gestire le richieste AJAX proveniente dal browser, ma soprattutto per interagire con database e con i file XML presenti sul server. Ad esempio, è stato utilizzato per gestire l'inserimento di nuovi pazienti, il recupero delle informazioni relative ad un esercizio per il bambino, oppure per il salvataggio di nuovi esercizi.

4.6.2 Java

Il secondo problema da affrontare è stato quello relativo alla definizione del gestore delle comunicazioni di tipo *WebSocket*. Sono state sviluppate molteplici versioni di server in grado di rispondere a richieste di connessione di tipo *WebSocket*, sia in PHP, Java, C# e Node.js [20]. La scelta più naturale e sensata, vista la necessità di utilizzare PHP per la gestione delle richieste dell'applicazione (come ad esempio interrogazioni del database), sarebbe quella di utilizzare tale linguaggio anche per la gestione della comunicazione tramite *WebSocket*. Tuttavia, l'implementazione PHP presente in rete non introduce il supporto all'ultima versione del protocollo, ma solo a quelle precedenti, così come per Node.js. Per questo motivo, la scelta del linguaggio da utilizzare è ricaduta su Java, essendo l'unica in grado di supportare l'ultima specifica (RFC 6455) del protocollo.

Inoltre, i possibili problemi in termini di prestazioni di Java che potrebbero sorgere con un'applicazione che deve lavorare quasi in tempo reale sono stati quasi eliminati con la versione 1.7, dove la capacità di elaborazione è molto vicina a quella di un linguaggio compilato.

4.6.3 MySQL

In congiunzione con PHP, per il salvataggio dei dati e delle informazioni di gioco, si è reso necessario l'uso di MySQL. MySQL è un RDBMS (Relational

DataBase Management System) ampiamente diffuso in diverse applicazioni, non solo web, ed è opensource e compatibile con quasi la totalità delle piattaforme attualmente sul mercato.

Capitolo 5

Eye Tracking

Una componente fondamentale nell'analisi del comportamento dei bambini affetti da CVI è quella relativa alle capacità visive e di osservazione degli oggetti.

Il problema dei bambini affetti da CVI è quello di non essere in grado di processare, tramite il cervello, le informazioni visive acquisite tramite gli occhi. Questo fa sì che le loro capacità visive siano molto basse, ed è necessario quindi una riabilitazione per cercare in qualche modo di recuperare capacità visiva, anche se difficilmente tali bambini saranno in grado di riacquistare totalmente la vista.

Per questo motivo, all'interno del progetto si è sentita la necessità di introdurre un sistema di tracciamento degli occhi che sia in grado di fornire informazioni precise relative al movimento dell'occhio ed al punto in cui tale occhio è rivolto sullo schermo, al fine di valutare se ci sono, ad esempio, zone del campo visivo che risultano oscure al bambino.

Un'altra informazione importante che può essere acquisita tramite tracciamento degli occhi è quella relativa ai movimenti saccadici. La saccade viene definita come un movimento oculare che consiste in rapidi movimenti degli occhi, in questo caso del tutto involontari per il bambino. Un sistema di eye-tracking, utilizzato su bambini affetti da CVI e movimenti saccadici, risulta essere particolarmente importante in quanto permette di valutare sia il livello di tali movimenti, l'ampiezza, e se eventualmente vi è una diminuzione nella frequenza o nell'ampiezza nel tempo.

All'interno di questo capitolo verrà data una rapida introduzione al problema ed alcune soluzioni proposte, in quanto questa parte del progetto è stata seguita da un altro tesista, Nicola Riparelli, che si è occupato di analizzare a fondo le diverse soluzioni possibili e di selezionare la migliore possibile per poterla adattare al nostro problema.

5.1 Il problema

Con il termine eye tracking si indica il processo di misurazione sia del punto in cui una persona sta guardando, sia del movimento dell'occhio relativamente alla testa. Tale analisi risulta essere particolarmente interessante per diverse applicazioni, come ad esempio l'analisi del comportamento dei visitatori di un sito web, in psicologia, in ambito medico, automobilistico, ecc.

Sebbene inizialmente studi sul movimento degli occhi erano compiuti manualmente dal medico, che si posizionava di fronte al paziente e ne tracciava in maniera manuale quelli che erano i movimenti dell'occhio, le tecnologie attuali utilizzate per il tracciamento degli occhi e del loro movimento ricadono fondamentalmente in due categorie:

- la prima tipologia utilizza delle particolari lenti a contatto, poste sull'occhio, all'interno delle quali vi è installato uno specchio oppure un sensore magnetico, e tali lenti vengono utilizzate per valutare il movimento dell'occhio, partendo dall'assunzione che se la lente si sta muovendo è perchè anche l'occhio si sta muovendo, e non perchè la lente si muove in maniera autonoma;
- la seconda categoria non utilizza delle lenti a contatto per il tracciamento del movimento degli occhi, ma viene utilizzata della luce, tipicamente ad infrarossi, che viene riflessa dall'occhio e catturata da una telecamera o un altro sensore ottico. L'informazione ottenuta viene quindi analizzata per ottenere informazioni riguardo il movimento dell'occhio a partire dai cambiamenti del riflesso catturato. Gli eye tracker basati sul video utilizzano il riflesso della cornea ed il centro della pupilla per determinare lo spostamento dell'occhio.

5.2 Tracciamento basato su video

Il metodo attualmente più utilizzato è quello basato su video. Una telecamera si focalizza su uno o entrambi gli occhi, registrando i movimenti dell'occhio della persona in risposta a qualche stimolo proveniente ad esempio dallo schermo. La maggior parte dei sistemi di eye-tracking utilizzano il centro della pupilla e una luce ad infrarossi per creare quello che viene chiamato riflesso corneale. A questo punto, il vettore costruito tra il centro della pupilla ed il riflesso corneale può essere utilizzato per calcolare o la posizione in cui la persona sta guardando oppure la direzione dello sguardo.

La Figura 5.1 rappresenta quelli che sono i componenti fondamentali che

5.3 Tracciamento tramite webcam

caratterizzano un sistema di eye-tracking basato su video [10]. Il sistema

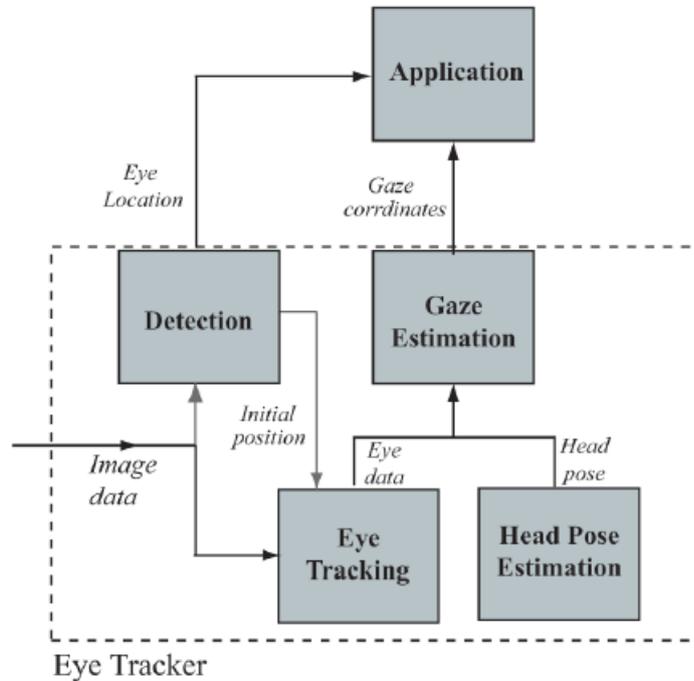


Figura 5.1: Componenti di un sistema di eye-tracking basato su video

ottiene le informazioni a partire da una o più videocamere (*Image Data*). La posizione dell'occhio all'interno dell'immagine viene determinata e viene utilizzata sia direttamente dall'applicazione oppure tracciata nei frame successivi. Basandosi poi sulla posizione degli occhi precedentemente calcolata e sulla possibile posizione della testa, viene stimato non solo la posizione dell'occhio ma anche dove gli occhi stanno guardando. Tale informazione può essere quindi utilizzata in qualsiasi applicazione basata sul movimento degli occhi, come ad esempio muovere il cursore del mouse attraverso il movimento degli occhi.

5.3 Tracciamento tramite webcam

Il tracciamento degli occhi basato su video ed una telecamera ad infrarossi assicura dei buoni risultati ma ad un prezzo, che è quello del costo dell'apparecchiatura necessaria per poter costruire il sistema.

Una soluzione molto più economica, basata sull'analisi di immagini senza telecamera ad infrarossi, è quella proposta in [31], che utilizza solamente una

comune webcam presente in tutti i computer moderni. L'architettura del sistema è presentata in Figura 5.2. Come possiamo osser-



Figura 5.2: Posizione della webcam per il tracciamento del movimento degli occhi

vare, la webcam, unico strumento utilizzato in questa soluzione, viene posta sopra lo schermo.

Le fasi del processo di analisi sono fondamentalmente tre, ossia:

1. identificazione della posizione della faccia;
2. identificazione della posizione degli occhi;
3. analisi del movimento degli occhi.

Nella fase di identificazione della posizione della faccia, il sistema si occupa di estrarre, da ciascun frame video ricevuto in input, la posizione della faccia. Questa operazione viene fatta per poter così eliminare dal frame tutta la parte di immagine non relativa alla faccia, ma semplicemente allo sfondo, che non è rilevante per l'analisi, riducendo quindi il tempo di computazione dei passi successivi, visto che si concentreranno solamente nell'analisi della faccia e non di qualche particolare esterno ad essa.

Nella fase successiva, quella relativa all'identificazione della posizione degli occhi, si vuole estrarre dall'immagine contenente la faccia della persona la posizione degli occhi. Le fasi principali di questo procedimento riguardano l'eliminazione dell'influenza della luce sull'immagine, seguita dall'identificazione della regione dell'occhio basandosi sulla simmetria della faccia (e quindi dei due occhi rispetto al naso), ed infine sull'angolo che si può costruire tra il naso e gli occhi, che è compreso tra i 30° ed i 60° .

Infine, l'ultimo passo è quello dell'identificazione del movimento degli occhi. Poiché la risoluzione di una webcam è chiaramente bassa in termini di qualità di immagine, questa operazione è quella più difficile in termini computazionali.

5.4 Tobii

Per prima cosa, il sistema viene calibrato chiedendo alla persona di guardare in nove punti differenti dello schermo, divisi in tre righe da tre punti ciascuna. Infine, operazioni con la trasformata di Fourier e analisi con Support Vector Machine permettono di stimare il movimento degli occhi.

Tramite questo metodo, sicuramente poco costoso, i test effettuati hanno dimostrato che mediamente l'accuratezza di questo tipo di analisi è intorno al 90%, calcolato come media delle valutazioni dei tracciamenti fatti con o senza occhiali, dove l'analisi senza occhiali ha dato i risultati migliori. Dipenderà quindi dall'applicazione che utilizza il tracciamento degli occhi valutare se la precisione raggiunta con questo metodo è sufficiente oppure no.

5.4 Tobii

Quando si parla di soluzioni professionali per il tracciamento degli occhi e del loro movimento, particolare interesse è rivolto verso *Tobii Technologies* [27], la più grossa azienda che fornisce sistemi per il tracciamento degli occhi e del loro movimento.

L'azienda sviluppa una notevole quantità di sistemi adatti al tracciamento degli occhi, a partire da semplici dispositivi che si possono collegare ai computer, a quelli molto più complessi come ad esempio televisioni oppure occhiali operanti in congiunzione con marker ad infrarossi.

Le applicazioni delle tecnologie Tobii sono le più disparate, come assistenza alla guida, giochi, test di usabilità dei sistemi, immagini mediche oppure operazioni mediche, ecc.

Il problema maggiore di questa tecnologia deriva dal costo dei diversi dispositivi, con cifre decisamente molto alte, e molto spesso insostenibili per gli enti di ricerca (soprattutto italiani). Nella Figura 5.3 viene presentato un modello di occhiale per il tracciamento degli occhi, utilizzato soprattutto nell'analisi del comportamento visivo nella scelta dei prodotti in un supermercato.



Figura 5.3: Occhiali Tobii per l'analisi del movimento degli occhi

Capitolo 6

Sviluppo del progetto

La prima decisione da prendere in merito allo sviluppo del progetto riguarda i browser da supportare e quali invece non considerare. Per fare ciò, è necessario considerare due elementi: gli scenari di utilizzo e la diffusione dei browser sui diversi dispositivi.

Per quel che riguarda la diffusione, il primo dato da prendere in considerazione è quello relativo alla diffusione sul suolo italiano. Per fare ciò, si è valutato l'andamento delle statistiche di utilizzo dei browser dell'ultimo anno. Come possiamo osservare dalla Figura 6.1, Chrome risulta essere il browser maggiormente utilizzato, seguito da Internet Explorer, Firefox e Safari.

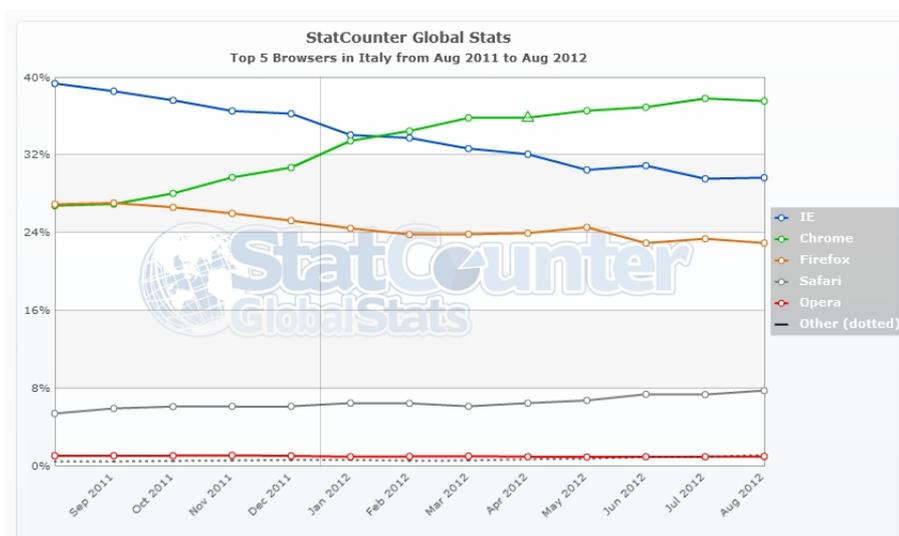


Figura 6.1: Andamento dell'utilizzo dei browser nel periodo Agosto 2011 → Agosto 2012

All'interno di questa classifica, Internet Explorer costituisce il maggior

problema, in quanto è l'unico, tra quelli sopra citati, a non supportare ancora (almeno prima della versione 8) le specifiche HTML5 e CSS3. Questo significa che, sebbene per una larga fetta di utenti il browser utilizzato è adatto all'applicazione sviluppata, per tutti quelli che utilizzano Internet Explorer 8 sarà necessario o l'aggiornamento alla versione successiva, oppure una sostituzione con un browser che supporti tali funzionalità.

6.1 Costruzione dell'architettura

Come descritto nella sezione 3.7.2, l'architettura generale del sistema è costituita da quattro moduli distinti, ovvero:

- il server centrale;
- il software che gestisce l'eye-tracker;
- il client dedicato al medico;
- il client dedicato al bambino.

Questi quattro elementi devono essere collegati tra loro attraverso una rete, ed è necessario che il software centrale, che si occupa di smistare tutti i messaggi che devono essere spediti tra i diversi elementi, venga collocato in un server raggiungibile attraverso un indirizzo IP.

Il problema successivo riguarda dove installare e far funzionare gli altri tre moduli, ovvero il software eye-tracker, il modulo per l'interfaccia del medico ed il modulo per l'interfaccia del bambino.

La configurazione più semplice che è possibile utilizzare è quella attraverso l'uso di tre computer differenti, dove ogni computer è dedicato allo svolgimento di un particolare task. Tuttavia, questo tipo di configurazione è molto dispendiosa sia in termini economici, visto il costo che sarebbe necessario sostenere per l'acquisto dei tre computer, sia in termini di spazio, visto che tre computer con tre schermi occupano sicuramente uno spazio non indifferente, senza dimenticare lo spazio occupato dai cavi che potrebbero rendere difficile la convivenza dell'apparecchiatura con il bambino.

Un'altra soluzione per la costruzione del sistema è quella di utilizzare una singola macchina per i tre moduli, con due diversi schermi collegati, uno touch per la parte del bambino ed uno normale per quella invece del medico, con il sistema di eye-tracking che opera in background.

La scelta di quest'ultima soluzione è però quella più difficile da realizzare in termini di prestazioni, in quanto l'uso di una sola macchina per le operazioni

6.1 Costruzione dell'architettura

di tracciamento degli occhi e di gestione dei giochi richiede un'elevata capacità computazionale, non presente sulle macchine meno recenti.

Per la costruzione dell'architettura migliore, si è valutato quello che è il carico di lavoro per ciascun modulo precedentemente descritto. Il modulo relativo al client del dottore è quello che richiede la capacità computazionale minore, in quanto non dovrà svolgere alcuna operazione dal costo elevato in termini di risorse, visto che dovrà semplicemente visualizzare dei dati ricevuti dal server, senza la necessità di compiere particolari operazioni. Dall'altro lato, il software di eye-tracking ed il client per il bambino sono i due moduli che richiedono maggiori capacità computazionali, nel primo caso perchè l'analisi di immagini acquisite dalla videocamera per la successiva analisi è chiaramente un'operazione dispendiosa, mentre nel secondo caso perchè il modulo deve gestire sia le animazioni dei giochi, sia l'invio dei pacchetti con i dati. Questa analisi ci permette quindi di costruire un'architettura che prevede l'inserimento del software di eye-tracking su un computer, ed il client per il bambino su un altro computer. Dove collocare poi il client dedicato al medico è una questione semplicemente pratica.

Se si vuole dare la possibilità al medico di interagire con il sistema durante il gioco del bambino, ed eventualmente cambiare qualche parametro di gioco, bisogna considerare il fatto che nei sistemi operativi attuali, è possibile mantenere il focus (e quindi avere interazione) solamente su una finestra alla volta, anche se sono due gli schermi collegati. Ciò significa quindi che il client del medico non può essere collocato nello stesso computer del client del bambino, visto che il focus in quel caso sarà sempre occupato dalla finestra dove è presente il gioco con il quale il bambino interagisce. Per questo motivo, il client relativo al dottore viene collocato nello stesso computer del software di eye-tracker, che una volta lanciato potrà liberamente lavorare in background. In Figura 6.2 viene presentata l'architettura realizzata per il sistema finale. Come è possibile vedere, due sono i client utilizzati, uno utilizzato dal medico insieme al software di eye-tracking, l'altro utilizzato dal bambino per giocare.

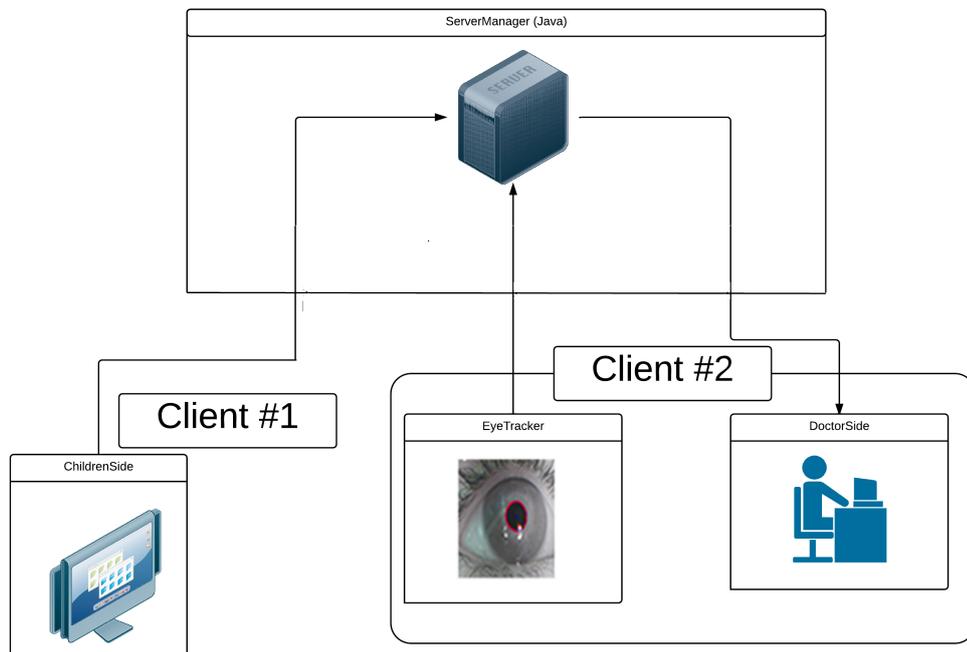


Figura 6.2: Architettura finale del sistema realizzato

6.1.1 ServerManager

Il *ServerManager* è il gestore che si occupa dello scambio di informazioni, pacchetti e della memorizzazione dei dati e delle prestazioni di gioco ed è installato sul server che ospita l'applicazione web. Questo gestore è a sua volta costituito da tre elementi distinti, ognuno dei quali si occupa di una connessione diversa. Questo significa che avremo un gestore per l'eye-tracker, un gestore per il client del medico ed un gestore per il client del bambino. La motivazione alla base di questa divisione nasce dal fatto che vista l'enorme quantità di pacchetti che verranno spediti, ricevuti ed elaborati, lasciare un unico gestore ad occuparsi di tutte le operazioni di tutti e tre i moduli connessi comporta sicuramente dei problemi in termini di prestazioni. Inoltre, poichè il *WebSocket* si mette in ascolto per pacchetti in ingresso su una particolare porta (8000, 8001 ecc), usarne solo una per spedire e ricevere un'enorme quantità di pacchetti potrebbe generare un enorme traffico di rete, con conseguente peggioramento delle prestazioni. Per questo motivo, i tre gestori distinti vengono inoltre messi in ascolto su tre porte di rete distinte, in maniera tale da cercare di ridurre al minimo la quantità di pacchetti circolante.

Perchè questi tre gestori funzionino attraverso il protocollo HTML5 *WebSoc-*

6.1 Costruzione dell'architettura

ket, è stato necessario costruirli in maniera tale che fossero in grado di gestire correttamente l'handshake iniziale tra client e server per poter definire tale connessione. Tra le diverse possibilità di server *WebSocket* disponibili liberamente in rete, quello utilizzato per il progetto è stato *Java-WebSocket* [12], scritto in Java.

Tale implementazione, che definisce la struttura di base di ciascun gestore (che può essere pensato come un server) introducendo il supporto all'handshake necessario per aprire correttamente la connessione *WebSocket*, fornisce il supporto per molteplici *Draft* del protocollo che si sono susseguiti nel tempo, aumentando così il bacino di browser che supportano questa comunicazione, visto che solo le versioni più recenti implementano l'ultimo *Draft* proposto per questa modalità di comunicazione.

Per poter implementare il proprio *WebServer* che supporti la comunicazione tramite *WebSocket*, è necessario estendere (ed implementare) la classe astratta *WebSocketServer*, la quale si occuperà di stabilire la connessione con il client, implementando quattro metodi necessari al corretto funzionamento del server:

- *onOpen(WebSocket, ClientHandshake)*: metodo invocato al momento che è stata stabilita una connessione con il client
- *onClose(WebSocket, int, String, boolean)*: invocato quanto la connessione con il client viene chiusa
- *onMessage(WebSocket, String)*: invocato ogni qual volta il client invia un messaggio al server attraverso i *WebSocket*
- *onError(WebSocket, Exception)*: gestisce eventuali problemi nella connessione

Ovviamente, il metodo maggiormente interessante è quello relativo alla gestione dei pacchetti, in quanto ciascun gestore diverso, a seconda del pacchetto ricevuto, dovrà eseguire le relative operazioni.

Durante la fase di gioco, il client del bambino ed il sistema di tracciamento degli occhi invieranno, a cadenze regolari e prefissate, dei pacchetti con l'informazione relativa alla posizione dell'immagine e del tocco per il primo, informazioni relative alla vista per il secondo.

Poichè il momento di arrivo di tali pacchetti al server non può essere noto a priori, e non è possibile che tale momento sia contemporaneo ai due sistemi, all'interno del *ServerManager* sono stati introdotti dei gestori di pacchetti relativi alle informazioni di gioco. Questi gestori, diversi a seconda del gioco selezionato, utilizzano due buffer distinti, uno per le informazioni provenienti

dal tracciamento degli occhi e l'altro per le informazioni del gioco, per sincronizzare tra di loro i vari pacchetti e ricondurre ad un solo pacchetto le due informazioni presenti. Ogni pacchetto spedito o ricevuto da questi gestori è marcato con il tempo, che indica quanto tempo è passato dall'inizio del gioco, e tale informazione viene utilizzata per mettere insieme due pacchetti provenienti da due sorgenti diverse e mostrare al medico la posizione ad un particolare istante temporale di occhi, tocco e immagine utilizzata.

Il sistema è anche in grado di gestire l'assenza di pacchetti da una delle due sorgenti, in quanto non c'è alcuna sicurezza che tutti i pacchetti siano spediti e ricevuti dal server, per molteplici motivi. L'operazione eseguita in questo caso dai gestori è quella di inserire un pacchetto fittizio, con valori uguali a -1, per la sorgente non collegata o non in grado di inviare pacchetti; in questo modo, il sistema potrà comunque fornire al medico qualche informazione, ovvero i dati con valore diverso da -1, relativa alla sorgente collegata di cui si è ricevuto il pacchetto con le informazioni.

Poichè è necessario poter visualizzare i risultati dei bambini in un secondo momento da parte del medico, tutti i pacchetti ricevuti vengono salvati, all'interno di file diversi, in una directory di archiviazione dedicata a ciascun bambino. All'interno di tale directory, diversa per ciascuna rilevazione effettuata e rinominata secondo la convenzione "YYYY-MM-DD-HH-MM-IDVisita", sono presenti i seguenti file:

- *GameSpecs.ini*: contiene informazioni relative al gioco, come ad esempio la dimensione dello schermo o la grandezza dell'immagine;
- *InputEyeTracking.txt*: memorizza i pacchetti relativi al tracciamento degli occhi;
- *InputImage.txt*: memorizza i pacchetti relativi alla posizione dell'immagine sullo schermo;
- *InputTouch.txt*: memorizza i pacchetti relativi alla posizione del tocco;
- *DeltaValues.txt*: informazioni relative al gioco *Prendimi!*;
- *Results.txt*: informazioni relative al gioco *Aiutami!*.

Per tutti i pacchetti, la convenzione utilizzata per memorizzare le informazioni è:

(informazioni _ separate _ da _ virgola)

6.1 Costruzione dell'architettura

Nel caso dei file *InputEyeTracking.txt*, *InputImage.txt*, *InputTouch.txt*, le informazioni memorizzate sono:

(offset_temporale, left, top)

dove:

- *offset_temporale*: rappresenta il tempo in cui sono state rilevate le informazioni spedite
- *left* e *top* indicano invece la posizione x e y dell'immagine, tocco o vista rispetto all'angolo in alto a sinistra dello schermo.

Nel caso del file *DeltaValues.txt*, le informazioni memorizzate sono:

(time, ΔT , ΔE , left, top, movimento)

dove:

- *time*: fornisce l'indicazione temporale relativa al pacchetto;
- ΔT : rappresenta l'errore commesso dal tocco (-1 nel caso in cui il tocco non sia presente), calcolato come la differenza tra la posizione del tocco e la posizione del centro dell'immagine;
- ΔE : rappresenta l'errore commesso dalla vista (-1 nel caso in cui questa non sia rilevata), calcolato come la differenza tra la posizione dell'occhio ed il centro dell'immagine;
- *left*: rappresenta il valore *left* della posizione dell'immagine;
- *top*: rappresenta il valore *top* della posizione dell'immagine;
- *movimento*: indica la direzione che sta seguendo l'immagine nello spostamento.

La necessità di salvare tali informazioni nasce dalla volontà di rappresentare, come informazioni per il medico, un grafico contenente un insieme di valori rappresentanti la distanza che intercorre tra la posizione del centro dell'immagine e la posizione o del tocco o della vista (per questo motivo vengono memorizzati i valori *time*, ΔT e ΔE).

Per costruire il tooltip visualizzato quando viene posto il mouse sopra un punto del grafo, sono invece necessarie le informazioni relative alla posizione dell'immagine (*top* e *left*) e il movimento seguito (campo *movimento*). La Figura 6.3 presenta un possibile stato dello schermo del bambino, con indicate

le informazioni che il server invierà al client del medico per la costruzione del grafico.

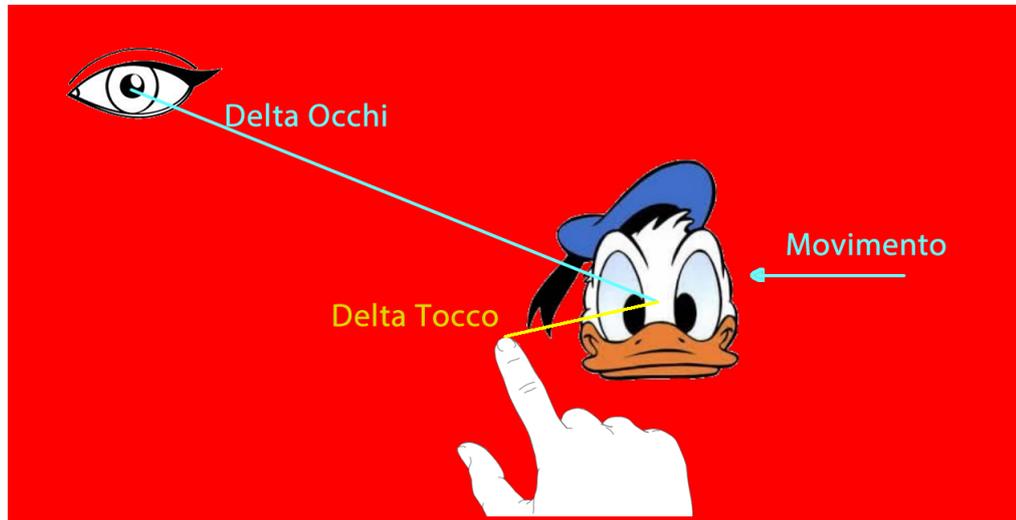


Figura 6.3: Indicazione delle informazioni spediti dal server al client del dottore

A questo punto il client del dottore, avendo a disposizione le informazioni ricevute, può costruire un grafico dell'andamento dell'esercizio del bambino come quello mostrato in Figura 6.4. Da notare inoltre come attraverso quest'ultimo tooltip, venga riprodotta la situazione presentata in Figura 6.3, fornendo quindi al medico la possibilità di "rivedere" quello che è accaduto durante l'esercizio svolto dal bambino.

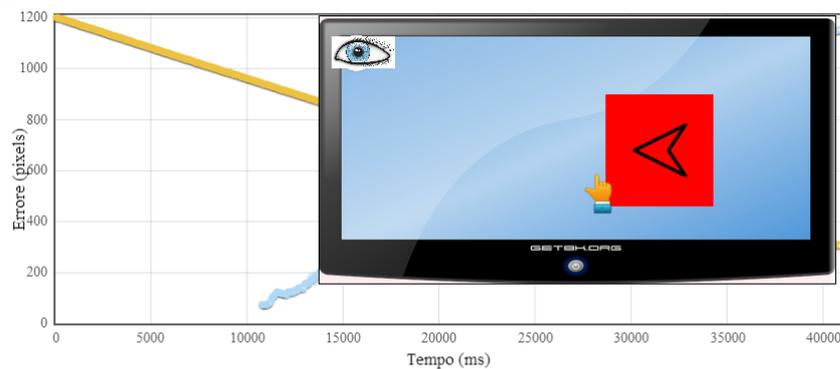


Figura 6.4: Esempio grafico costruito per il gioco Prendimi!

Nel caso del file *Results.txt*, contenente informazioni relative al gioco *Aiutammi!*, le informazioni salvate sono:

6.1 Costruzione dell'architettura

(famiglia_target, is_target, nome_immagine, FRT, CT, risposta_esatta)

dove:

- *famiglia_target*: rappresenta la famiglia target del livello;
- *is_target*: indica se l'oggetto fa parte della famiglia target o meno;
- *nome_immagine*: indica il nome dell'immagine corrente;
- *FRT*: rappresenta il valore relativo al First Response Time;
- *CT*: rappresenta il valore relativo al Completion Time;
- *risposta_esatta*: indica se la risposta fornita dal bambino è stata corretta o meno.

Queste informazioni vengono utilizzate per costruire il riepilogo fornito al medico durante il gioco oppure in un secondo momento per un'analisi più approfondita (Figura 6.5).

Oggetto	Famiglia Target	First Response Time (ms)	Tempo di completamento (ms)	Esatte / Errate
Riepilogo	animali	3110.67	7936.00	2/1

Figura 6.5: Esempio di valori messi a disposizione del medico per il gioco *Aiutami!*

6.1.2 Gestore Eye-Tracker

Il gestore del sistema di eye-tracking si occupa di instradare e gestire correttamente i pacchetti provenienti dal software di tracciamento degli occhi. Esso si mette in ascolto sulla porta assegnata 8000, ed attende che il software si colleghi a tale porta. Per poter essere sicuri che non sia un intruso ma effettivamente il software gestore del tracciamento, il primo pacchetto spedito (e lo stesso procedimento viene applicato anche dagli altri gestori) richiede al software di identificarsi, attendendo come risposta la stringa identificativa "EyeTrackerClient". Solo nel caso in cui il server riceva tale risposta allora la connessione verrà mantenuta attiva, altrimenti essa verrà chiusa. Una volta che tutto il sistema è stato settato e pronto a partire, il software di tracciamento degli occhi ha la necessità di iniziare un'insieme di stimolazioni

visive, attraverso la somministrazione di immagini in determinati punti dello schermo, per poter calibrare il sistema e poter quindi fare successivamente il tracciamento. In questo caso il server si occuperà di instradare i pacchetti contenenti le posizioni di visualizzazione verso il client del bambino, che si occuperà a sua volta di visualizzare le immagini nella posizione indicata.

6.1.3 Gestore client medico

Il gestore del client medico (che si mette in attesa di connessioni sulla porta 8002) si occupa principalmente di due aspetti fondamentali: riceve le impostazioni di gioco per farle arrivare poi al client del bambino, oppure spedire al client del medico i risultati che il client deve visualizzare per un resoconto in tempo reale delle prestazioni.

Una volta che il medico ha scelto il gioco e definito le impostazioni, e quindi il sistema è pronto per partire, il gestore si occupa di inserire nel database la nuova visita, memorizzando in particolare data, bambino ed il gioco scelto. Inoltre, nello stesso momento viene anche inizializzato il gestore che si occuperà di organizzare e salvare i messaggi con le informazioni relative alle diverse posizioni.

6.1.4 Gestore client bambino

Il gestore del client bambino, collegato alla porta 8001, si occupa fondamentalmente di ricevere i messaggi con le posizioni di tocco e dell'immagine da parte del client, e di inoltrare verso il client del medico delle informazioni che sono utili al settaggio delle partire, come ad esempio le dimensioni dello schermo utilizzato per il gioco del bambino.

6.2 Il problema della sincronizzazione

Uno dei problemi di maggior spessore che è stato necessario affrontare durante lo sviluppo del progetto è stato il problema della sincronizzazione tra i diversi dispositivi, che in particolare riguarda il client del bambino ed il software di gestione del tracciamento degli occhi. Come è facile intuire, affinché le informazioni contenute all'interno dei pacchetti provenienti da sorgenti diverse (eye-tracking e gioco) possano essere unite per ottenere dei dati utili al medico per la propria analisi, come ad esempio la costruzione del grafico in Figura 6.6, è necessario che, se due pacchetti provenienti dalle due sorgenti si riferiscono allo stesso istante temporale (in termini assoluti, indipendentemente cioè dall'orologio della singola sorgente), i valori relativi

6.2 Il problema della sincronizzazione

al tempo associati a ciascuno dei due pacchetti siano molto vicino, entro un certo margine di errore E fissato.

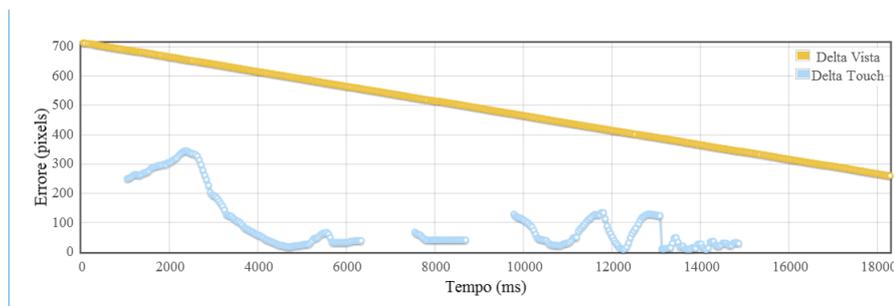


Figura 6.6: Esempio di grafico costruito per il gioco *Prendimi!* attraverso le informazioni provenienti dalle diverse sorgenti

La soluzione di marcare ogni pacchetto con l'orario in cui le informazioni vengono rilevate non è una strada percorribile, in quanto i due dispositivi non hanno necessariamente lo stesso orario impostato, e non è chiaramente possibile impostare l'orario manualmente, visto che lavorando a circa 25fps la precisione dev'essere dell'ordine di qualche decina di millisecondo, che con un'impostazione manuale è chiaramente impossibile da raggiungere. L'alternativa potrebbe essere quella di utilizzare il protocollo NTP [7], protocollo il cui funzionamento è basato sul rilevamento dei tempi di latenza nel transito dei pacchetti sulla rete. E' un protocollo di tipo client-server, in grado di sincronizzare gli orologi dei computer su internet entro un margine di 10 millisecondi e con un'accuratezza di 200 microsecondi all'interno di una rete LAN in condizioni ottimali, utilizzando il tempo UTC ¹. Il problema di questo protocollo è che la sua installazione non è molto agevole e quindi difficilmente realizzabile da persone con poche capacità informatiche.

Per questo motivo, l'unico modo possibile per avere una sincronizzazione tra i pacchetti è quello di utilizzare, anziché il valore assoluto del tempo, un valore relativo, ovvero, posto t_0 il tempo di inizio del gioco e t_i il tempo a cui si rilevano le informazioni, il valore temporale inserito nel pacchetto sarà pari a $t_i - t_0$. Il problema introdotto da questo tipo di soluzione è però dato dal fatto che, essendo i due orologi diversi, se indichiamo con t'_0 e t''_0 gli orari di partenza dell'eye-tracker e del gioco, essi sono necessariamente diversi. La necessità è quindi quella che, sebbene t'_0 e t''_0 siano per definizione diversi,

¹Il tempo UTC è anche definito come tempo civile o tempo coordinato universale, ed è il fuso orario di riferimento a partire dal quale sono calcolati tutti gli altri fusi orari del mondo. Esso è derivato dal GMT (Greenwich Mean Time).

essi rappresentino lo stesso istante temporale (ad esempio rispetto il GMT) e quindi facciano partire contemporaneamente il software di tracciamento ed il gioco. Per definire tale istante, la soluzione più semplice è quella di utilizzare un orologio neutro, indipendente cioè dagli altri due, che in questo caso è l'orologio dove si trova il *ServerManager*, definire un orario di partenza per quest'ultimo, chiamato t_{0u} , e attraverso un'opportuna funzione di trasformazione dipendente da opportuni parametri, convertire t_{0u} in t'_0 e t''_0 . In questo modo, sebbene t'_0 e t''_0 siano diversi, essi rappresenteranno lo stesso istante temporale, con il risultato che utilizzando tali valori come orari di partenza delle rispettive operazioni, i due sistemi risulteranno essere sincronizzati e partiranno quindi nello stesso momento.

La costruzione della funzione di conversione e dei parametri da utilizzare è un argomento trattato molto spesso in letteratura e sotto diversi punti di vista e con diverse soluzioni.

Prendiamo in considerazione due nodi distinti, 1 e 2, con i rispettivi orologi $t_1(t)$ e $t_2(t)$ dove t è il tempo universale UTC. Un qualsiasi orologio di un elemento $t_i(t)$ può essere espresso, in funzione di t , come

$$t_i(t) = a_i t + b_i$$

dove a_i viene definito come il parametro relativo alla deviazione dell'orologio, e che nasce dalla differenza di cadenza dell'oscillatore che genera i secondi, mentre b_i è relativo invece alla differenza in termini di secondi tra i due orologi (ovvero, quanto un orologio si discosta da t). In generale a_i e b_i sono diversi per ciascun orologio.

Se indichiamo con t_2 l'orario del server e t_1 l'orario di uno dei due end-point, è possibile definire $t_1(t)$ come:

$$t_1(t) = a_{12} t_2(t) + b_{12} \quad (1)$$

dove a_{12} e b_{12} sono rispettivamente la deviazione relativa e l'offset relativo tra i due orologi. L'obiettivo è quello di calcolare i valori di a_{12} e b_{12} sia per la macchina dove è installato il software di tracciamento degli occhi sia in quella per il gioco del bambino.

Il primo passo per ognuna delle possibili soluzioni di questo problema è quello dello scambio di pacchetti tra il nodo 1 ed il nodo 2, per ottenere un'insieme di informazioni temporali (Figura 6.7). Quello che è richiesto per questa prima fase è che il nodo 1 invii al nodo 2 un primo pacchetto contenente come informazione il proprio orario (t_0). Il nodo 2, riceve tale pacchetto ed immediatamente lo rispedisce al mittente, dopo averlo marcato con una

6.2 Il problema della sincronizzazione

nuova informazione temporale (t_b) relativa al momento in cui il pacchetto è stato ricevuto (ovviamente, secondo il proprio orologio).

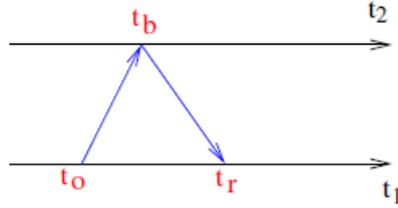


Figura 6.7: Scambio di pacchetti marcati temporalmente per il calcolo dell'offset tra gli orologi di due nodi

Il nodo 1 a questo punto marca il pacchetto ricevuto con l'orario di ricezione (t_r), secondo il proprio orologio, ottenendo una terna costituita da:

$$(t_0, t_b, t_r)$$

dove:

- t_0 è l'orario di partenza dal nodo 1;
- t_b è l'orario di ricezione del nodo 2;
- t_r è l'orario di ricezione nel nodo 1.

Questo scambio di pacchetti viene ripetuto n volte, con n dipendente dalla soluzione che si vuole utilizzare per poi calcolare i valori di a_{12} e b_{12} .

Le soluzioni proposte sono molteplici, come [29] o [16] oppure [25].

L'idea proposta in [16] prevede, attraverso i pacchetti ottenuti precedentemente, la costruzione di un modello che rappresenti l'orologio del nodo di cui si vogliono calcolare i parametri. Per la costruzione di tale modello, si vuole calcolare, dato l'insieme delle terne acquisite precedentemente, i valori di $\overline{a_{ij}}$ e $\underline{a_{ij}}$ (che rappresentano rispettivamente un upper bound ed un lower bound per il valore di a_{ij}), ed i valori di $\overline{b_{ij}}$ e $\underline{b_{ij}}$ (che rappresentano rispettivamente un upper bound ed un lower bound per i valori di b_{ij}). Per determinare tali valori, la soluzione proposta prevede la costruzione e la soluzione di due problemi di programmazione lineare, il cui numero di vincoli aumenta all'aumentare del numero di pacchetti acquisiti precedentemente.

Tale soluzione ha però problemi in termini di prestazioni, in quanto il tempo necessario per risolvere i due problemi di programmazione lineare non è chiaramente trascurabile, e aumenta all'aumentare del numero dei pacchetti

scambiati tra i due nodi, con il rischio quindi di avere un tempo di setup troppo elevato nel caso si voglia ottenere valori precisi, oppure dei valori poco precisi a fronte di un tempo di setup meno elevato.

Per il progetto la soluzione adottata è stata quella proposta in [25], che a fronte di una buona precisione (ordine di qualche decina di millisecondi, comunque accettabile per lo scopo del progetto) richiede una quantità di risorse molto limitata ed un tempo di elaborazione dell'ordine di qualche secondo. L'idea della soluzione è la seguente. Ogni terna di dati ottenuta con un pacchetto, definisce un insieme di vincoli sui possibili valori che può assumere a_{12} e b_{12} , ossia:

$$t_0 < a_{12}t_b + b_{12} \quad (2) \quad \text{e}$$

$$t_r > a_{12}t_b + b_{12} \quad (3)$$

La dipendenza lineare tra t_1 e t_2 ed i vincoli imposti da ciascuna terna di valori, possono essere rappresentati in un grafico (Figura 6.8), dove ogni punto può essere rappresentato da due vincoli distinti.

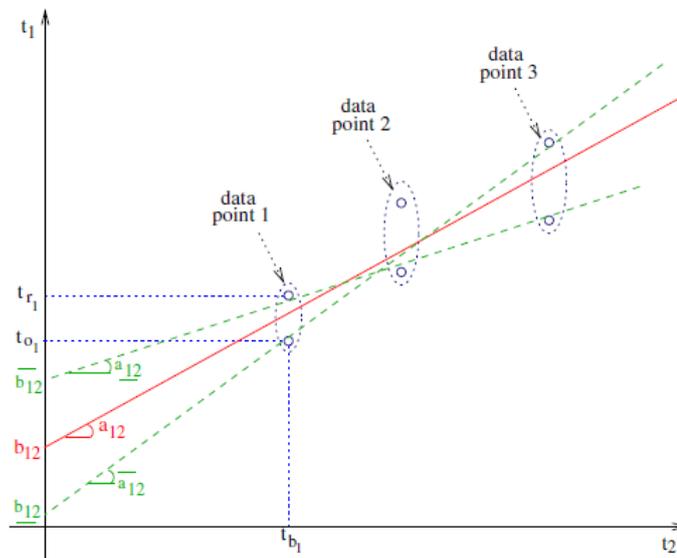


Figura 6.8: Rappresentazione dei pacchetti su un grafico come delle rette che definiscono dei vincoli sui valori finali

Se infatti rappresentiamo (1) all'interno del grafo come una retta, avremo che le disuguaglianze (2) e (3) impongono appunto dei vincoli sull'inclinazione e la posizione della retta rappresentate (1) (tale retta dovrà infatti collocarsi tra la retta rappresentante la disuguaglianza (2) e la disuguaglianza (3)).

6.2 Il problema della sincronizzazione

Anche se i valori esatti di a_{12} e b_{12} non possono essere determinati, possono essere definiti dei bound per tali valori:

$$\frac{a_{12} \leq a_{12} \leq \overline{a_{12}}}{b_{12} \leq b_{12} \leq \overline{b_{12}}} \text{ e}$$

dove $\underline{a_{12}}$ e $\overline{a_{12}}$ rappresentano un lower-bound ed un upper-bound per l'inclinazione della retta soddisfacente tutti i vincoli, mentre $\underline{b_{12}}$ e $\overline{b_{12}}$ rappresentano l'intersezione con l'asse y della retta dall'angolazione imposta da $\underline{a_{12}}$ e $\overline{a_{12}}$. A questo punto, l'obiettivo dell'algoritmo è quello di determinare i valori di $\underline{a_{12}}$, $\overline{a_{12}}$, $\underline{b_{12}}$ e $\overline{b_{12}}$ che siano i più stretti possibile, in maniera tale da definire il più precisamente possibile i valori di a_{12} e b_{12} . Per fare ciò l'algoritmo prende in considerazione una coppia di pacchetti (nel nostro caso il primo e l'ultimo scambiati, in quanto sono i due pacchetti che ci forniscono il risultato migliore) e determina i valori di $\underline{a_{12}}$, $\overline{a_{12}}$, $\underline{b_{12}}$ e $\overline{b_{12}}$ attraverso le seguenti formule:

$$\begin{aligned} \underline{a_{12}} &= \frac{\Delta t_o - RTT}{\Delta t_b}, \\ \overline{a_{12}} &= \frac{\Delta t_o + RTT}{\Delta t_b}, \\ \underline{b_{12}} &= t_{o_1} - t_{b_1} \overline{a_{12}}, \\ \overline{b_{12}} &= t_{o_1} + RTT - t_{b_1} \underline{a_{12}}. \end{aligned}$$

dove:

$$\begin{aligned} \Delta t_o &= t_{o_2} - t_{o_1} \\ \Delta t_b &= t_{b_2} - t_{b_1}. \end{aligned}$$

Rispetto alla versione originale dell'algoritmo, poichè nei vari calcoli effettuati vi è la necessità del valore di RTT (Round Trip Time), anzichè calcolarlo utilizzando l'ultimo pacchetto ricevuto, in questo caso vengono utilizzati 500 pacchetti iniziali usati solo per una stima del valore del valore di RTT.

Grazie a questa soluzione, con un errore molto basso (ed entro comunque i limiti imposti dalla definizione del problema) è possibile avviare sia l'eye-tracking che il gioco allo stesso istante e quindi avere dei pacchetti il cui valore temporale è uguale o molto simile. In particolare, affinchè due pacchetti vengano messi insieme, il loro valore temporale non può discostarsi di un ΔT , nel nostro caso definito uguale a 60ms. Questo perchè lavorando a 25 frame al secondo, il client del bambino e l'eye-tracker invieranno un

pacchetto ogni 40ms, e quindi la distanza temporale tra un pacchetto ed il successivo è almeno 40ms. Da questo possiamo dedurre che due pacchetti fanno riferimento allo stesso istante temporale se la loro differenza è più piccola di 40ms, in quanto in caso contrario quello con marcatore temporale più elevato farà riferimento ad un istante temporale successivo.

Questo significa quindi che il massimo scostamento temporale che è possibile avere tra due pacchetti che rappresentano informazioni relative allo stesso istante temporale è compreso tra 0ms (che ci indica che le informazioni sono state acquisite nello stesso istante temporale) e 40ms (più di tale soglia vorrebbe dire mettere insieme pacchetti appartenenti a due istanti temporali diversi).

6.3 Gestione della comunicazione

L'intero sistema costruito è basato sull'uso di pacchetti di comunicazione. Questi pacchetti sono costruiti seguendo lo standard JSON [15], sia per ridurre al minimo la loro dimensione e quindi la quantità di dati trasmessa nella rete, sia per la sua facile interoperabilità tra i diversi linguaggi quali in questo caso Java e JavaScript.

Ogni pacchetto è identificato da una chiave "TYPE" che identifica la tipologia del pacchetto, e quindi quelle che sono le informazioni contenute all'interno di esso. In questo modo, ogni volta che un gestore riceve un pacchetto dal proprio client designato, a seconda del valore all'interno del campo "TYPE" è in grado di decidere quelle che sono le operazioni da eseguire con tale pacchetto.

6.4 Sviluppo gioco Prendimi!

Il gioco ribattezzato *Prendimi!* ha come scopo quello di allenare la capacità di fissazione e di osservazione del bambino, oltre alla capacità di coordinamento, se possibile, tra vista e arti superiori (mani e braccia).

6.4.1 Definizione

L'idea alla base del gioco è quella di inserire un'immagine all'interno dello schermo, che esegue una serie di movimenti, eventualmente sempre più complessi, da un punto di origine ad un punto di destinazione. Il punto di origine può essere o sempre il centro dello schermo, visto che è il punto di osservazione più semplice per un bambino, oppure da un estremo dello schermo stesso.

6.4 Sviluppo gioco Prendimi!

I movimenti più semplici a cui è necessario sottoporre il bambino sono quelli solo orizzontali o solo verticali. Solo in un secondo momento, nel caso in cui le prestazioni del bambino migliorino insieme alle sue capacità, è possibile combinare i diversi movimenti tra di loro, e quindi ad esempio proporre un movimento verso destra e verso l'alto, oppure verso il basso e verso sinistra, fino ad arrivare alla complessità massima che compie un giro completo dello schermo toccando tutti e 4 i bordi.

6.4.2 Impostazioni di gioco

L'insieme delle impostazioni di gioco personalizzabili sono emerse da un'analisi di quelle che sono i problemi che caratterizzano i bambini affetti da CVI. Vi sono infatti bambini che hanno difficoltà a distinguere colori poco contrastanti tra loro, altri con problemi di gestione di spostamenti a velocità troppo elevate, o sono in grado di osservare solo particolari spostamenti ecc. Per questi motivi, la pagina di personalizzazione del gioco è molto corposa, includendo al suo interno diverse possibilità di modifica. La parte di sinistra della pagina mostrata in Figura 6.9 viene utilizzata per modificare:

- il colore di sfondo;
- se, ed eventualmente con quale, modificare il colore dell'immagine in movimento;
- la dimensione dell'immagine.

Nel caso in cui si decida di modificare il colore dell'immagine, grazie alla possibilità dell'elemento *canvas* di accedere ad ogni singolo pixel che caratterizza l'immagine visualizzata, tutti i colori presenti nell'immagine originale vengono sostituiti con il nuovo colore, eccezion fatta per il bianco che rimane tale. Questa soluzione può essere adottata nel caso in cui vi siano delle difficoltà nel bambino tali da necessitare dell'introduzione di immagini a forte contrasto con lo sfondo. Nel caso inoltre il colore scelto per l'immagine non sia sufficientemente in contrasto con il colore di sfondo, viene visualizzato un messaggio di errore (non bloccante per il proseguimento dell'esercizio) che informa della possibilità che il bambino abbia problemi a vedere l'immagine per via del basso contrasto (Figura 6.10).



Figura 6.9: Pagina di personalizzazione delle impostazioni per il gioco *Prendimi!*



Figura 6.10: Messaggio che indica che i colori scelti per lo sfondo e per l'immagine hanno basso contrasto

Nella parte destra della pagina in Figura 6.9 è possibile invece:

- scegliere i movimenti che l'immagine dovrà seguire, con la possibilità di scegliere tra:

6.4 Sviluppo gioco Prendimi!

- movimento verso il basso;
 - movimento verso l'alto;
 - movimento verso destra;
 - movimento verso sinistra;
 - combinare tra loro i movimenti selezionati precedentemente;
 - far partire l'immagine nel suo spostamento sempre dal centro.
- modificare la velocità di movimento;
 - selezionare l'immagine da utilizzare.

Tutte queste possibilità di scelta sono state inserite per riuscire a programmare un esercizio che si adatti il meglio possibile a ciascun bambino.

I valori di default impostati inizialmente vengono recuperati dal database e sono associati a ciascun bambino. Se il medico decide di modificare tale valori e utilizzarne di nuovi per l'esercizio, il sistema in automatico rileva che c'è stata una modifica rispetto alla versione precedente, e provvede a inserire all'interno del database il nuovo esercizio inizializzandolo ad esercizio di default, in maniera tale che la prossima volta sia direttamente accessibile il nuovo esercizio. Tale funzionalità si rende necessaria in quanto per valutare sensatamente le prestazioni del bambino, è necessario farlo sempre sullo stesso esercizio, e questo salvataggio nel Database permette di volta in volta di recuperare velocemente l'esercizio precedentemente proposto per riproporlo nuovamente.

6.4.3 Informazioni per il medico

Durante il gioco da parte del bambino, lo schermo del medico visualizzerà diverse informazioni, oltre a dei bottoni di controllo per la modifica "al volo" del gioco.

La parte centrale della pagina è occupata da un grafo che visualizza, con aggiornamento delle informazioni ogni 5 secondi, l'andamento del gioco e del comportamento del bambino. In particolare, quello che viene visualizzato rispetto al tempo è l'andamento dell'errore commesso dal bambino nel guardare o toccare l'immagine. Avremo quindi che sull'asse x è rappresentato il tempo (in millisecondi), mentre sull'asse y il valore dell'errore commesso (in pixel) rappresentando sia l'errore per il tocco che per la vista (Figura 6.11). Il client riceve già i valori calcolati e si preoccupa solamente di visualizzarli nella forma di grafo. Il medico ha anche la possibilità, posizionando il mouse sopra un punto all'interno del grafo, di visualizzare un tooltip (Figura

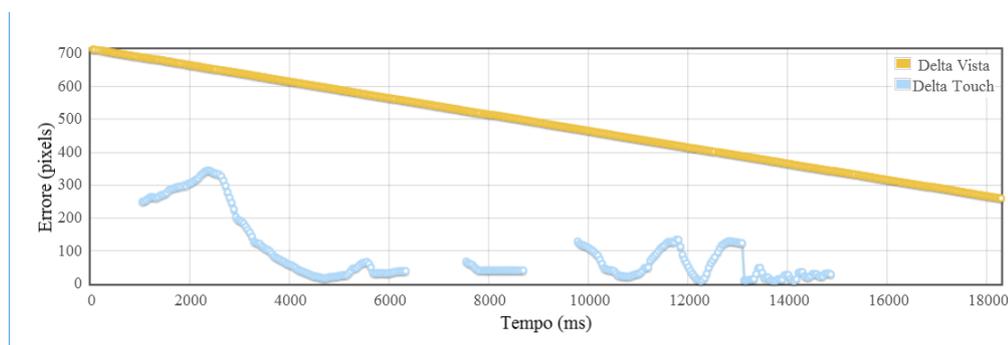


Figura 6.11: Grafico riassuntivo dell'errore commesso dal bambino per il gioco *Prendimi!*

A.8) che, mantenendo invariate le proporzioni dello schermo utilizzato per il gioco, visualizza, contestualmente alla posizione dell'immagine in quel dato istante, anche la posizione dell'occhio e la posizione del tocco (se questi sono presenti). Inoltre, se con il mouse viene seguito l'andamento dei punti del grafo, l'effetto che si otterrà è un'animazione che riproduce quello che è stato l'andamento dell'esercizio svolto dal bambino, mostrando lo spostamento seguito dall'immagine, dal tocco e dagli occhi.

Le informazioni visualizzate dal medico durante la simulazione saranno comunque accessibili anche in un secondo momento, attraverso la sezione apposita, e saranno presentate sempre attraverso lo stesso grafo, per permettere un'analisi più dettagliata, da parte del medico, del comportamento del bambino. Come possiamo osservare in Figura 6.13, oltre alla visualizzazione del grafico è possibile:

- modificare la velocità di spostamento, nel caso il medico si accorga che quella impostata è troppo elevata o troppo lenta;
- navigare il grafico spostandosi sull'asse temporale, in quanto esso visualizza solamente le informazioni relative all'ultimo minuto, e quindi se si vuole visualizzare un periodo temporale precedente è possibile tornare indietro nel tempo;
- interrompere il gioco.

6.4 Sviluppo gioco Prendimi!

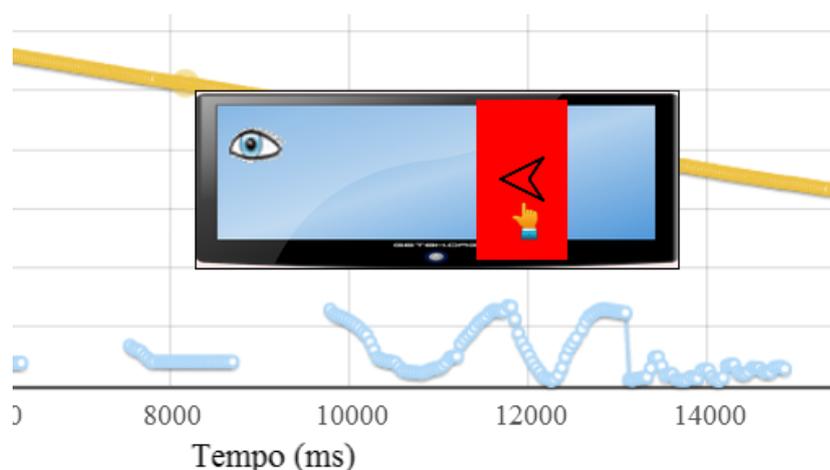


Figura 6.12: Esempio di tooltip rappresentante le posizioni di occhio e tocco rispetto all'immagine per il gioco Prendimi!



Figura 6.13: Sistema di controllo a disposizione del medico per il gioco Prendimi!

6.4.4 Valutazione delle prestazioni

Una volta completato il gioco e la sequenza di animazioni da sottoporre al bambino, è necessario fornire una valutazione numerica oggettiva dei risultati ottenuti dal bambino, in maniera tale da riuscire a valutare in maniera molto più agevole i progressi del bambino durante il periodo di riabilitazione. La modalità di valutazione proposta per questa tipologia di esercizio è basata sulla vicinanza della vista o del tocco rispetto al centro dell'immagine in movimento. Questa valutazione è utile per capire come, e quanto, il bambino è in grado di indirizzare il proprio sguardo ed il proprio tocco verso la figura. Poiché verificare se il bambino guarda o tocca perfettamente il centro dell'immagine non è chiaramente una valutazione sensata, risultando essere molto restrittiva in termini di abilità, la figura utilizzata sullo schermo, in congiunzione ad una parte dello spazio intorno ad essa, è stata suddivisa in tre aree.

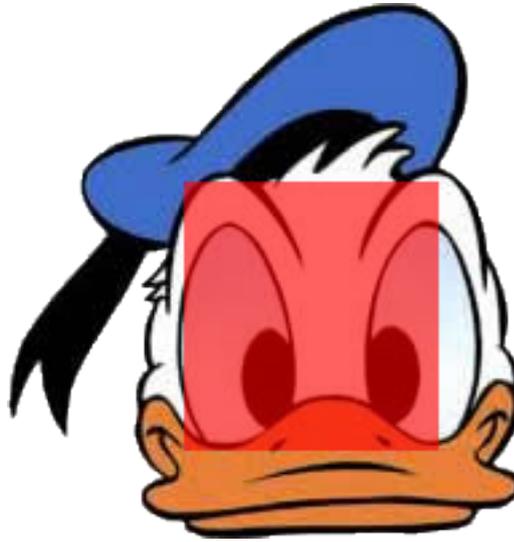


Figura 6.14: Divisione dell'immagine in aree per la valutazione del tocco e della vista

L'area più centrale, rappresentata in Figura 6.14 dalla zona colorata di rosso, è centrata al centro dell'immagine, e si estende per un'area pari a $h/2 * w/2$ dove h rappresenta l'altezza dell'immagine mentre w rappresenta la larghezza. Quest'area, essendo la zona più vicina al centro dell'immagine, e quindi è positivo che il bambino guardi in questa zona, viene valutata con un punto. L'area esterna alla zona rossa, ma comunque all'interno dei limiti dell'immagine, viene valutata 0.5 punti. La parte esterna all'immagine invece, che comprende un'area maggiore di $h/2$ e $w/2$ rispetto alla dimensione dell'immagine, viene valutata 0.25 punti. A questo punto, per valutare il comportamento del bambino si definiscono due valori:

$$V(e) = \frac{(\sum_{m \in M} p(e)) * 100}{|M|}$$

e

$$V(t) = \frac{(\sum_{m \in M} p(t)) * 100}{|M|}$$

dove:

- $V(e)$ rappresenta la valutazione della vista (espressa in %);
- $V(t)$ rappresenta la valutazione del tocco (espressa in %);
- $p \in P$ rappresenta un pacchetto(m) all'interno dell'insieme contenente tutti i pacchetti (M);

6.5 Sviluppo gioco Aiutami!

- $p(e)$ punteggio acquisito dalla vista, basandosi sulla posizione dell'occhio rispetto all'immagine;
- $p(t)$ punteggio acquisito dal tocco sulla base della posizione rispetto all'immagine.

Attraverso quindi la valutazione di tocco e vista sulla base di valori oggettivi, il medico è in grado di comprendere in maniera immediata se e come gli esercizi proposti per il bambino stiano avendo effetti positivi o negativi sulla sua capacità visiva.

6.5 Sviluppo gioco Aiutami!

Il gioco ribattezzato *Aiutami!* è stato sviluppato secondo le indicazioni fornite dalla Prof.ssa Nota e dalla Dott.ssa Sgaramella del Dipartimento di Psicologia Generale dell'Università di Padova. Lo scopo del gioco è quello di sviluppare nel bambino la capacità di problem solving e di coordinamento degli arti e della vista, in risposta a degli stimoli visivi e richieste di svolgimento di particolari operazioni.

6.5.1 Definizione

L'ambientazione del gioco porta il bambino al Polo Nord, a dover aiutare l'elfo di Babbo Natale a riempire dei sacchi con dei regali da portare ad altri bambini.

La costruzione del gioco consiste per prima cosa di un insieme di famiglie, come ad esempio "Animali", "Macchine", "Fiori" ecc, dove ogni famiglia contiene un insieme di immagini di particolari oggetti.

Il gioco richiede che il bambino, dopo aver ascoltato un'indicazione vocale indicante qual è la famiglia definita target per il livello corrente, inserisca all'interno del sacco di Babbo Natale solamente le immagini appartenenti a tale famiglia, oppure scarti le immagini non appartenenti a tale famiglia.

Prima del gioco vero e proprio, viene presentata un'animazione, con l'elfo che arriva in sella alla slitta, sia per coinvolgere ed attrarre il più possibile il bambino a giocare, e successivamente per spiegare al bambino le regole del gioco e quelli che sono gli obiettivi/operazioni che dovrà eseguire.

6.5.2 Parametri di gioco

L'impostazione fondamentale del gioco riguarda la costruzione dei livelli di gioco.

Ogni livello è identificato da una tipologia, dove la tipologia identifica il numero di elementi target (appartenenti cioè alla famiglia designata per quel livello) e di elementi chiamati distrattori (elementi cioè non appartenenti alla famiglia designata). Al livello, vi è poi associata la famiglia *target*, ed una sequenza di immagini che verranno visualizzate sullo schermo per stimolare il bambino al riconoscimento ed alla scelta dell'operazione da eseguire.

La Figura 6.15 mostra l'esempio di schermata somministrata al medico per settare i livelli di gioco. I livelli di default, presentati inizialmente, sono quelli associati al bambino scelto, oppure quelli di default se al bambino non è ancora stata associata una particolare sequenza di livelli.

The screenshot shows a user interface for customizing game levels. At the top, there is a section titled "Inserisci nuovo livello:" with a dropdown menu set to "6x4" and a blue button labeled "Aggiungi livello". Below this is a main panel for "Livello 1 - T: 6 x D: 4". It includes a "Famiglia target:" dropdown set to "animali" and a "Tempo massimo immagine:" dropdown set to "25". The main area contains a table with 8 rows, each representing a level element. Each row has a column for a letter (T or D), a dropdown menu for the element name, and a corresponding image icon.

T	Gatto	
T	Coniglio	
D	Occhiali	
D	Orologio bambini	
T	Tartaruga	
D	Ombrello	
T	Maiale	

On the right side of the interface, there are two arrows (up and down), a blue button "Elimina livello", and another blue button "Livelli completati".

Figura 6.15: Schermata di personalizzazione dei livelli per il gioco *Aiutami!*

Per ciascuna sequenza di immagini, composta da immagini *target* e *distrattori*, è possibile definire:

- la famiglia *target* per quella sequenza;
- il numero massimo di secondi attesi per una risposta da parte del bambino;
- l'immagine associata a ciascun elemento che compone la sequenza, scelta tra tutte quelle inserite nel sistema.

6.5 Sviluppo gioco Aiutami!

In questo modo, vi è la possibilità di definire nella più totale libertà i livelli di gioco, adattandoli nel migliore dei modi alle esigenze del bambino. È possibile inoltre inserire una nuova sequenza di immagini, eliminarne una presente e modificare l'ordine di presentazione degli oggetti al bambino, spostando prima o dopo gli oggetti target/distrattori.

Una volta completate la modifica delle impostazioni, il bottone "Livelli completati" invia i livelli creati al client del bambino, che potrà quindi istanziare tale livelli durante il gioco. Nel caso in cui i livelli definiti modifichino quelli precedentemente assegnati al bambino, tali informazioni verranno salvate all'interno di un file XML, secondo le specifiche definite in 6.5.3, per poterle recuperare la volta successiva e sottoporre eventualmente il bambino agli stessi esercizi.

6.5.3 Archiviazione delle informazioni

Per archiviare le diverse informazioni, come ad esempio le immagini a disposizione, i livelli di gioco (di default o associati a ciascun bambino) il sistema utilizza dei file XML all'interno della cartella *settings* dove è salvato il gioco *Aiutami!*.

Definizione delle immagini

Il sistema mette a disposizione del medico un insieme di immagini, suddivise per famiglia di appartenenza. Ogni famiglia ha associato anche un file audio, riprodotto prima dell'inizio del livello, che viene utilizzato per dire al bambino quale sarà la famiglia di oggetti che dovrà essere inserita all'interno del sacco. Il file XML utilizzato per l'archiviazione delle informazioni è organizzato nel modo seguente:

```
<imageMaterials>
  <family type="animali" audioFile="intro_animali">
    <image id="1" name="Oca" fileName="oca.jpg" />
    <image id="2" name="Asino" fileName="asino.png" />
    ...
  </family>
  <family type="macchine" audioFile="intro_macchine">
    <image id="8" name="Ferrari" fileName="ferrari.png" />
    ...
  </family>
  ...
</imageMaterials>
```

A questo punto, nella definizione del livello all'interno del file XML basterà specificare l'ID dell'immagine affinché poi essa venga utilizzata durante il gioco.

Definizione delle sequenze

I file XML adibiti al salvataggio delle informazioni relative alle sequenze di gioco associate a ciascun bambino sono fondamentalmente di due tipi: generali o specifici. I file generali contengono delle sequenze associate al livello di disabilità che può avere un bambino. I file specifici sono invece assegnati a ciascun bambino, e contengono delle sequenze costruite specificatamente per lui.

Ogni volta che devono essere caricate le sequenze per un bambino, il sistema si comporta nel modo seguente:

- se all'interno del database è assegnato un file specifico al bambino, verranno recuperate le sequenze presenti all'interno di tale file, in maniera tale da dare la possibilità al medico di riproporre al bambino la stessa sequenza, e valutare eventuali miglioramenti;
- se non è presente tale associazione, viene recuperato il livello di disabilità associato al bambino, e tramite tale informazione vengono recuperate le sequenze contenute nel file generale associato a tale livello di disabilità.

Ogni file specifico (sono più di uno in quanto se vengono creati delle nuove sequenze per lo stesso bambino non si sovrascrive il file contenente le sequenze correnti, ma se ne crea uno nuovo) è archiviato nella cartella *settings/ID_BAMBINO/* dove *ID_BAMBINO* è l'ID del bambino all'interno del database. Quando viene richiesto il file con le sequenze per un particolare bambino, il sistema recupera dal database il file attualmente attivo, ovvero l'ultimo selezionato come attivo per quel bambino, e successivamente legge il contenuto per costruire praticamente le sequenze. La struttura utilizzata per l'archiviazione dei livelli è la seguente:

```
<levels>
  <level targetFamily="animali" type="6x4" maxTimeImage="25">
    <image type="T" imageID="5" />
    <image type="T" imageID="3" />
    <image type="D" imageID="11" />
    <image type="D" imageID="13" />
    <image type="T" imageID="7" />
    <image type="D" imageID="12" />
    <image type="T" imageID="6" />
  </level>
</levels>
```

6.5 Sviluppo gioco Aiutami!

```
<image type="T" imageID="2" />
<image type="D" imageID="14" />
<image type="T" imageID="4" />
</level>
</levels>
```

dove:

- *targetFamily* indica il nome della famiglia target;
- *type* indica la tipologia di livello (numero_target x numero_distrattori);
- *maxTimeImage* indica il numero di secondi a disposizione del bambino per fornire una risposta.

6.5.4 Informazioni per il medico

Rispetto al gioco *Prendimi!*, le informazioni fornite al medico durante la simulazione sono completamente diverse. Risulta infatti essere inutile memorizzare la distanza che c'è tra il centro delle immagini e la vista o il tocco, in quanto quello che in questo caso interessa è come il bambino si comporta di fronte alla sequenza di oggetti che gli vengono presentati.

Per questo motivo, durante il gioco, sfruttando le capacità e velocità di comunicazione attraverso il protocollo *WebSocket*, il medico può vedere in tempo reale il comportamento del bambino, ovvero vedere, in una riproduzione dello schermo utilizzato, la posizione dell'occhio e del suo movimento, il movimento del centro dell'immagine e del tocco del bambino. In questo modo il medico è in grado di capire dove il bambino guarda e come riesce a coordinare sguardo e tocco rispetto ad un'immagine collocata sullo schermo.

Oltre a questo, il sistema permette di visualizzare anche un semplice report delle prestazioni del bambino, analizzando le risposte corrette o meno nella classificazione degli oggetti. Come possiamo vedere in Figura 6.16, le informazioni visualizzate sono:

- *First Response Time*: indica in millisecondi il tempo necessario per ottenere dal bambino una reazione, ossia il tocco dello schermo;
- *Completion Time*: indica in millisecondi il tempo necessario al bambino per completare l'operazione di inserimento (o di eliminazione) dell'immagine;
- *Esatte/Errate*: indica il numero di risposte esatte o errate fornite fino a questo momento.

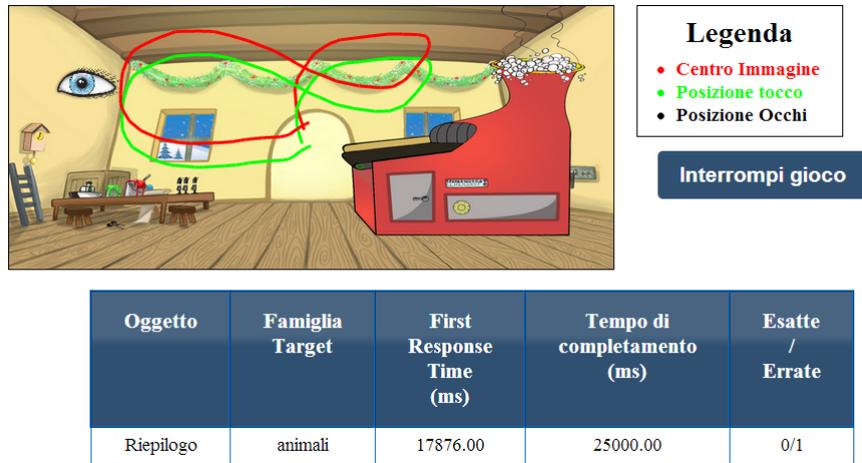


Figura 6.16: Esempio di riepilogo per il medico per il gioco Aiutami!

6.5.5 Valutazione delle prestazioni

Per questi esercizi, la valutazione finale fornisce una media dei punteggi totalizzati dal bambino nei diversi livelli proposti. Come prima possibilità, mostrata in Figura 6.17, è possibile conoscere, per ogni visita, la media del tempo di risposta (First Response Time), la media del tempo di completamento (Completion Time) ed il numero di risposte corrette o errate date per tutti gli esercizi.

Gioco Aiutami!					
Data	First Response Time (ms)	Completion Time (ms)	Ris. Corrette	Ris. Errate	Visualizza Esercizio
23/09/2012	4294	15929	1	0	>
19/09/2012	4807	7363	1	0	>
19/09/2012	4652	5316	1	0	>
19/09/2012	4087	4739	1	0	>
19/09/2012	3698	4087	1	0	>
19/09/2012	4502	7036	1	0	>
10/09/2012	1626	4652	4	0	>
10/09/2012	2172	4960	4	0	>

Figura 6.17: Riassunto valutazioni per il gioco Aiutami!

Nel momento in cui invece si vogliono visualizzare delle informazioni più

6.5 Sviluppo gioco Aiutami!

puntuali riguardo una particolare sessione di gioco, quello che viene presentato al medico è un riepilogo, suddiviso per livello, delle immagini presentate e dei singoli valori di *First Response Time*, *Completion Time* e se la risposta fornita è stata corretta o meno. In questo modo il medico può avere un'indicazione più precisa circa quelli che sono i comportamenti del bambino con le immagini, individuando magari set di immagini particolarmente difficili da riconoscere e trattare per il bambino.

 **Famiglia target: animali**

Nome Oggetto	Ogg. Target	FRT (ms)	CT (ms)	Risp. Corretta
Oca	true	3544	13973	true
Oca1	true	1226	1758	true
	Valori medi:	2385	7866	2/0

 **Famiglia target: macchine**

Nome Oggetto	Ogg. Target	FRT (ms)	CT (ms)	Risp. Corretta
Ferrari	true	1069	1608	true
	Valori medi:	1069	1608	1/0

 **Famiglia target: fiori**

Nome Oggetto	Ogg. Target	FRT (ms)	CT (ms)	Risp. Corretta
Rosa	true	668	1271	true
	Valori medi:	668	1271	1/0

Figura 6.18: Esempio di riepilogo dettagliato per il gioco Aiutami!

6.6 Gestione offline dei giochi

I requisiti del sistema sviluppato chiedono la possibilità di poter far giocare (e quindi riabilitare) il bambino non solo in un contesto ospedaliero, ma anche in un contesto casalingo, seguendo degli esercizi assegnati dal medico, al fine di ottenere almeno una mezz'ora di attività riabilitativa al giorno.

Lo sviluppo di un'applicazione web raggiungibile tramite qualsiasi browser (che supporti HTML5) fa sì che qualsiasi famiglia dotata di un computer con connessione alla rete possa collegarsi per far giocare il proprio bambino.

Tuttavia, due sono i motivi principali per cui la sola applicazione web può non essere sufficiente alla riabilitazione casalinga. Il primo deriva dal fatto che non tutte hanno la possibilità di avere a casa un computer con connessione alla rete, o più semplicemente un computer sufficientemente recente da supportare i browser più recenti (che supportano cioè HTML5), e quindi in grado di riprodurre i giochi. Dall'altro lato, l'uso di un computer per far giocare un bambino può non essere la scelta più felice, in quanto un computer può non stimolare appieno l'interesse e la curiosità del bambino, non invogliandolo quindi a giocare, e di conseguenza a cercare di ottenere i risultati migliori.

Per risolvere questo problema, e come detto anche in [2], l'uso dei dispositivi tablet, in questo caso particolare gli iPad©, stimolerebbero maggiormente i bambini al gioco, e quindi possono essere utilizzati come ottimo strumento per attrarre i bambini verso il gioco.

6.6.1 Da online ad applicazione nativa

La prima possibilità per poter utilizzare l'applicazione web su un dispositivo tablet è quella di sfruttare il browser di cui tutti i dispositivi sono dotati. Tuttavia tale possibilità incontra subito diversi problemi, in particolare quelli relativi allo sviluppo di tali browser.

La maggior parte delle funzionalità relative ad HTML5 non sono state infatti ancora implementate, con il rischio quindi che il browser non sia in grado di gestire correttamente tutte le funzionalità dell'applicazione (in particolare, la riproduzione dei suoni, che è un punto fondamentale soprattutto nei *serious games*, è uno dei maggiori problemi riscontrati).

La seconda possibilità per lo sviluppo di applicazioni per tablet è lo sviluppo di applicazioni native. Questo tipo di approccio, vista la necessità dell'uso dell'applicazione sul maggior numero possibile di tablet, è chiaramente impraticabile, vista la necessità di sviluppare una versione per i dispositivi Apple© ed un'altra per i dispositivi Android©, sviluppando quindi un tota-

6.6 Gestione offline dei giochi

le di almeno tre applicazioni, una versione web più due versioni distinte per i tablet.

Per poter quindi riuscire a trasportare un'applicazione web in un'applicazione per tablet, con l'obiettivo fondamentale di avere il maggior livello di portabilità a fronte di una minima quantità di codice modificato, è possibile utilizzare degli strumenti che trasformano il codice HTML, CSS e Javascript in applicazione nativa, in particolare sfruttando il framework PhoneGap [22].

Apache Cordova PhoneGap

Apache Cordova PhoneGap (alias PhoneGap) è il framework più famoso utilizzato per lo sviluppo di applicazioni cross-platform non native a partire da applicazioni web. Esso permette la conversione di tali applicazioni web in applicazioni native compatibili con la maggior parte dei dispositivi mobile, come Android, Apple, Blackberry, Symbian, Windows Mobile etc.

L'idea alla base del progetto è molto semplice: lo sviluppatore scrive la propria applicazione utilizzando il linguaggio HTML (e quindi anche HTML5), CSS (e CSS3) e Javascript. Una volta completato il progetto, tramite phonegap si effettua un porting dell'applicazione web, trasformandola in un'applicazione nativa per il dispositivo scelto. Quello che fa PhoneGap è impacchettare tutti i file necessari all'applicazione all'interno di un motore webkit (Figura 6.19), facendo sì che tale applicazione sia eseguibile senza un browser.

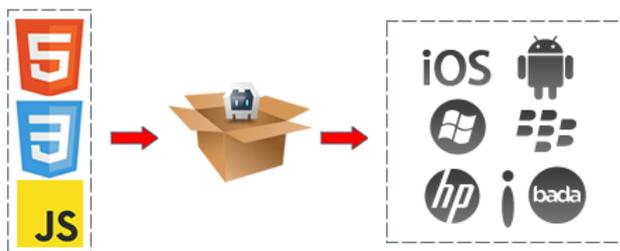


Figura 6.19: Creazione di un'applicazione nativa tramite PhoneGap

Oltre a questo porting, PhoneGap mette a disposizione anche un discreto numero di API che permettono, tramite opportuni oggetti messi a disposizione da questo framework, di accedere a diverse funzionalità tipiche dei dispositivi mobile. Si avrà quindi la possibilità di accedere alla fotocamera, al GPS, alla rubrica ed ai messaggi, senza dimenticare l'accelerometro, la rete o la scheda di memoria.

Da considerare comunque il fatto che non realizzando un'applicazione nativa, ma racchiudendo un'applicazione web all'interno di un wrapper che permette

l'esecuzione dell'applicazione senza browser, le prestazioni decadono notevolmente. Questo risulta essere particolarmente importante quando si realizzano applicazioni che sfruttano notevolmente la CPU e che richiedono una discreta quantità di risorse, in quanto il porting dell'applicazione potrebbe comportare una drastica riduzione delle prestazioni, con conseguente peggioramento dell'esperienza utente. Tuttavia, i due giochi sviluppati per questo progetto non appartengono a tale categoria, e si prestano molto bene al porting tramite questo framework verso i dispositivi mobile.

6.6.2 Salvataggio dei dati

Gli esercizi svolti a casa dal bambino risultano essere il più grande bacino di informazioni e di dati per il medico, visto che il loro numero è di gran lunga superiore rispetto alle visite effettuate presso lo studio medico.

Come detto precedentemente, sebbene l'ambiente casalingo non si presti al processo di tracciamento del movimento degli occhi, anche le sole informazioni relative al tocco del bambino rispetto alle immagini mostrate a video risultano essere molto importanti.

Per questo motivo, quando il gioco non viene eseguito in ambiente ospedaliero (significa cioè che l'utente loggato nell'applicazione non è un medico, o si sta giocando in modalità offline) le informazioni non vengono scartate o semplicemente non acquisite, ma vengono salvate tramite le *File API* messe a disposizione da HTML5.

Il sistema infatti sfrutta tale spazio messo a disposizione per creare una sorta di archivio del browser, dove ogni qual volta il bambino gioca, anziché spedire nella rete i pacchetti relativi alle posizioni del tocco o dell'immagine, questi vengono salvati localmente, all'interno di un file che conterrà quindi tutti i pacchetti che in una situazione ottimale (cioè in studio del medico) sarebbero stati inviati al sistema.

A questo punto, la prima volta che, tramite browser (o tablet nel caso in cui si stia usando tale dispositivo) ci si ricollega all'applicazione web, un messaggio visualizzato in home page (Figura 6.20) indicherà della presenza di sessioni di gioco archiviate e pronte per l'invio. A questo punto, il genitore da casa, se in presenza di connessione alla rete, oppure il medico direttamente dallo studio (usando quindi il tablet del bambino) potranno inviare al server le informazioni salvate.

6.6 Gestione offline dei giochi



Figura 6.20: Messaggio che informa della presenza di visite in modalità offline da inviare al server per l'archiviazione

6.6.3 Invio dei dati salvati

Affinchè gli esercizi effettuati a casa, salvati e spediti al server possano essere utili per una corretta analisi, è necessario estrapolare dei dati che mostrino quindi l'andamento dei risultati ottenuti dal bambino nel tempo. Come descritto in 6.6.2, per ogni esercizio svolto in ambiente casalingo il sistema si occupa di archiviare i pacchetti che in situazione ottimale sarebbero stati spediti al server per il loro allacciamento ai dati forniti dal sistema di eye-tracker. Quando si decide di inviare i dati al server, il sistema recupera l'insieme di tutti i pacchetti e li spedisce ad una pagina PHP designata per la gestione di queste informazioni.

Quello di cui si occupa tale pagina è per prima cosa il parsing dei pacchetti, da semplice testo a pacchetti JSON, molto più semplici da gestire. A questo punto, dal primo pacchetto dell'insieme il sistema riconosce a quale gioco fanno riferimento i pacchetti successivi, e delega ad un altro gestore (diverso per ogni gioco) la gestione dei restanti pacchetti.

Quello di cui si occupa ogni gestore è prendere ogni singolo pacchetto e trattarlo nel modo corretto, scrivendolo quindi ad esempio all'interno del file *InputTouch.txt* se le informazioni in esso contenute sono relative al tocco, oppure scrivendolo nel file *Results.txt* se invece fa riferimento al risultato di inserimento di oggetti nel sacco per il gioco *Aiutami!*.

Una volta completata la gestione (e quindi la scrittura) di tutti i pacchetti ricevuti, il sistema estrapola da tali pacchetti tutte le informazioni relative invece alla valutazione dell'esercizio, calcolando quindi la distanza del centro

dell'immagine dal tocco nel caso del gioco *Prendimi!* (e fornendo poi una valutazione secondo quanto descritto in 6.4.4), oppure costruendo il riepilogo di risposte corrette/errate o di tempi di risposta nel caso del gioco *Aiutami!* (come descritto in 6.5.5).

Una volta completata anche la valutazione dell'esercizio, la visita a domicilio viene archiviata all'interno del database affinché possa essere consultata dal medico per una successiva analisi.

6.7 Test dell'applicazione

Per poter avere una prima valutazione del sistema e dell'architettura implementata, oltre a tutte le scelte fatte relativamente ai due *serious games* da implementare, è stato organizzato un incontro con i committenti affinché potessero fornire un parere, soprattutto medico.

Il primo problema affrontato è stato quello relativo al frame rate delle applicazioni e quindi alla frequenza di generazione dei pacchetti con le relative informazioni. Come impostazione di base infatti, il frame rate è di 25fps, che equivale a generare un pacchetto informativo ogni 40ms. È proprio tale frequenza il problema maggiore. Se infatti consideriamo una visita della durata media di dieci minuti, il numero totale di pacchetti generato per una visita è pari a 15.000. Tale valore è chiaramente molto elevato, soprattutto se si considera che l'intero ammontare di pacchetti viene visualizzato attraverso un grafo, che quindi può richiedere una discreta quantità di tempo per poter elaborare tutti i punti.

Per questo motivo, si è valutata la possibilità di ridurre il numero di frame al secondo, riducendo di conseguenza il numero totale dei pacchetti. Tuttavia, il numero di frame al secondo, e quindi l'intervallo temporale tra un pacchetto e l'altro, è fortemente vincolato alla velocità dei movimenti saccadici che si possono verificare in un bambino. Poiché vogliamo infatti valutare le capacità visive del bambino, e riuscire a cogliere anche tali movimenti, è necessario che la frequenza dei pacchetti sia tale da riuscire a cogliere tali informazioni, e quindi non troppo bassa da non permettere tale analisi.

L'esito della discussione è stato quello di inserire, nelle impostazioni del gioco a disposizione del medico, il numero di frame al secondo da utilizzare per quella determinata sessione. In questo modo, se il bambino non è affetto da movimenti saccadici, o nel caso in cui tali movimenti fossero sufficientemente lenti, è possibile ridurre il numero di frame al secondo, e di conseguenza il numero totale dei pacchetti che devono essere rappresentati dal grafo.

La successiva analisi del sistema si è concentrata sulle immagini utilizzate.

6.7 Test dell'applicazione

Nel caso del gioco *Prendimi!*, era inizialmente proposta, a titolo di esempio, l'immagine presentata in Figura 6.21.

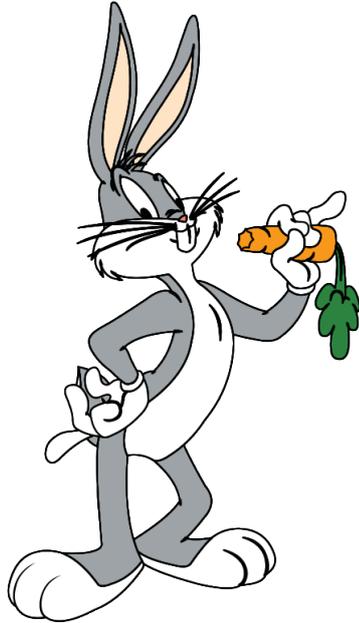


Figura 6.21: Immagine di Bugs Bunny utilizzata per la demo di Prendimi!

Tale immagine però ha suscitato diverse reazioni negative, in particolare perchè:

- è un'immagine complessa, contenente molti dettagli, e c'è il rischio che il bambino si perda nell'osservare i dettagli dell'immagine, perdendo quindi l'attenzione verso quello che è il compito che deve eseguire;
- è troppo sproporzionata nel rapporto tra altezza e larghezza, risultando molto più alta che larga. Questo fa sì che l'identificazione del centro dell'immagine, punto rispetto al quale viene valutato il comportamento del bambino, corrisponda in questo caso alla pancia di Bugs Bunny, mentre la reazione naturale di un bambino sarebbe quella di toccare ad esempio il naso, o per lo meno una parte del corpo più significativa.

Per questo motivo, si è deciso di eliminare tale immagine, e di sostituirla con immagini più adatte alla situazione, che catturassero l'attenzione del bambino soprattutto verso il centro dell'immagine stessa. Un esempio di immagini più adatte sono presentate in Figura 6.22.



Figura 6.22: Esempio di immagini più adatte all'utilizzo per il gioco Prendimi!

Per quel che riguarda il gioco *Aiutami!*, un primo problema segnalato è stato quello relativo all'ambientazione. La prima versione, presentata in Figura 6.23, è risultata essere troppo complessa, con una notevole quantità di dettagli inutili, che per i bambini rischiavano di essere addirittura distraenti, facendogli concentrare l'attenzione più sull'ambientazione che non sull'oggetto e sul task da svolgere.



Figura 6.23: Prima versione dell'ambientazione per il gioco Aiutami!, risultata essere troppo complessa e piena di dettagli inutili

Per questo motivo, l'ambientazione è stata rivista, e sono stati eliminati tutti quei particolari che potevano risultare distraenti per i bambini. La seconda versione è presentata in Figura 6.24.

6.7 Test dell'applicazione



Figura 6.24: Seconda versione dell'ambientazione per il gioco Aiutami! con molti particolari eliminati

Un altro problema emerso nel gioco *Aiutami!* è stato quello relativo alla barra, posta in alto sullo schermo, utilizzata per indicare il trascorrere del tempo, e quindi quanto tempo il bambino aveva ancora a disposizione per eventualmente inserire l'oggetto dentro il sacco. Tale barra correva però il rischio sia di distrarre il bambino, che poteva essere attratto da questo oggetto "in movimento", oppure poteva essere addirittura angosciato e quindi rifiutarsi di giocare. Per questo motivo, tale barra è stata eliminata dall'ambientazione, e si è deciso di sostituirla con dei messaggi sonori che incitino il bambino a giocare ed a compiere l'operazione richiesta.

Capitolo 7

Conclusioni

Il progetto realizzato ha visto la realizzazione di un sistema per la diagnosi e la riabilitazione di bambini affetti da Cerebral Visual Impairment (CVI). Gli obiettivi principali del progetto sono fondamentalmente due: sviluppare un sistema a supporto del dottore per la diagnosi e la riabilitazione del bambino, con particolare attenzione all'analisi della vista e del coordinamento occhio/movimento, e sviluppare un'applicazione che si adattasse il più possibile alle esigenze del bambino affinché il suo coinvolgimento fosse tale da avvicinarlo il più possibile al gioco e quindi alla riabilitazione.

Sin dalla parte iniziale della progettazione, la scelta è ricaduta sullo sviluppo di una Rich Internet Application (RIA, applicazioni web che hanno le stesse caratteristiche e funzionalità delle applicazioni desktop) in congiunzione con lo sviluppo di *serious games*, in maniera tale da nascondere dietro ad un gioco il più allettante possibile la riabilitazione del bambino e delle sue disabilità.

La scelta per lo sviluppo delle applicazioni e dell'intero sistema è ricaduta su HTML5 e le sue feature, che introducono diverse possibilità per la gestione di animazioni, suoni e trasformazioni che si adattano perfettamente allo sviluppo di giochi interattivi.

Oltre all'applicazione web fruibile attraverso i dispositivi desktop, è stata sviluppata anche una versione mobile utilizzabile attraverso dispositivi tablet come iPad© o tablet Android©.

Vista la scarsa maturità dei browser disponibili su questi dispositivi, si è reso necessario lo sviluppo di un'applicazione ad hoc. Per evitare la riscrittura completa del codice a seconda del dispositivo per cui si vuole sviluppare, con conseguente spreco di tempo e risorse, si è utilizzato il framework PhoneGap per effettuare un porting dell'applicazione web sui dispositivi tablet, trasformandola in applicazione ad hoc senza che questo comporti una riscrittura totale (ma solo a minimi pezzi) dell'applicazione.

Il sistema, dopo una prima versione, è stato presentato ai medici committenti dei due giochi, che hanno espresso le loro opinioni e proposto delle modifiche atte a migliorare l'esperienza utente dei bambini, riducendo quindi i problemi che potevano eventualmente nascere da una loro esperienza di gioco.

Dopo tale incontro, al sistema sono state apportate un insieme di modifiche volte a migliorare l'esperienza utente, sia per quel che riguarda le necessità dei dottori, soprattutto per aspetti medici a me sconosciuti, sia per quel che riguarda l'esperienza di gioco dei bambini, al fine di renderla il più aderente possibile alle caratteristiche ed alla necessità dei bambini stessi.

Questo progetto non rappresenta chiaramente un punto di arrivo o un lavoro fine a se stesso, ma rappresenta un punto di partenza attraverso il quale creare un sistema sempre più complesso e adatto alla diagnosi e riabilitazione di bambini affetti da deficit visivi.

In particolare, il sistema potrebbe integrare al proprio interno anche un secondo progetto sviluppato per la diagnosi dei problemi visivi dei bambini, ovvero *PlayWithEyes* [1], per creare una suite di applicazioni che possano diventare un valido strumento per destinato a medico e bambino per la cura o riabilitazione di bambini con problemi visivi.

Inoltre, ai due giochi già presenti possono essere introdotti nuovi giochi, con i loro obiettivi e con le loro operazioni da eseguire, allo scopo di aumentare quindi sia le possibilità di riabilitazione del bambino, sia quelle relative invece al divertimento ed all'interesse del bambino stesso, aspetto fondamentale nello sviluppo dei *serious games*.

Appendice A

Manuale Utente

L'applicazione è fruibile in due ambienti distinti, quello dello studio medico e quello casalingo nel caso degli esercizi riabilitativi.

Per quel che riguarda lo studio medico, per poter utilizzare l'applicazione nella sua interezza, corredata quindi anche il sistema di eye-tracking, è necessario avere a disposizione due computer connessi alla rete, e che utilizzino entrambi uno dei seguenti browser:

- Chrome, versione 10 o superiore
- Firefox, versione 3.6 o superiore
- Safari, versione 5 o superiore

Nel caso invece dell'ambientazione casalinga, le possibili modalità di utilizzo sono o attraverso un computer oppure attraverso un tablet. Se viene utilizzato un computer, è necessario avere a disposizione un browser che soddisfi i requisiti minimi specificati precedentemente, mentre per i tablet è possibile utilizzare o un iPad© o un dispositivo Android©.

Nel caso di utilizzo presso l'ambiente domestico, affinché l'applicazione possa funzionare anche in modalità offline, quindi senza connessione ad Internet attiva, è necessario essere connessi alla rete almeno la prima volta, così da scaricare le impostazioni di gioco per il bambino ed adattare gli esercizi alle sue caratteristiche.

La pagina iniziale dell'applicazione presenta la schermata di login. Le informazioni da inserire sono *username* e *password*, dove:

- nel caso del medico, sono quelle concordate al momento della sua registrazione nel sistema;
- nel caso del bambino, le credenziali vengono definite al momento della registrazione da parte del medico, come specificato nella sezione A.1.1.

A.1 Utente bambino

Una volta inserite le credenziali corrette, la pagina successiva permette di scegliere il gioco per la riabilitazione. Al momento, i due giochi disponibili sono *Aiutammi!*, dove bisogna aiutare un elfo nella preparazione del sacco di Babbo Natale, e *Prendimi!*, dove bisogna rincorrere, con il mouse o con il dito, una figura che si muove sullo schermo.

Una volta scelto il gioco a cui prendere parte, verrà mostrata un'immagine di "attesa", che permette al bambino di prepararsi e di posizionarsi nel modo a lui più congeniale. A questo punto, cliccando in qualsiasi zona dello schermo (o toccando con un dito nel caso in cui si stia utilizzando un'interfaccia touch, o attraverso un tablet o uno schermo touch) sarà possibile iniziare il gioco.

Nel caso del gioco *Prendimi!*, in alto a destra dello schermo è possibile visualizzare, previo click o tocco, un'icona che permette, attraverso un ulteriore click, di interrompere immediatamente il gioco.

A.1.1 Credenziali di accesso

Alla registrazione del bambino nel sistema, vengono assegnati automaticamente *username* e *password*.

- *username*: è definito come "*nome.cognome*" tutto minuscolo
- *password*: definita come "*cognomeGGMM*" dove *GG* e *MM* indicano rispettivamente il giorno ed il mese di nascita.

A.2 Utente Medico

Nel caso in cui le credenziali inserite nel form di login corrispondano ad un medico preventivamente registrato all'interno del sistema, la schermata presentata è quella in Figura A.1. A questo punto il medico avrà a disposizione tre possibilità:

- "Gestione bambini" permette di accedere all'area contenente lo storico degli esercizi svolti dai bambini, e relative valutazioni;
- "Schermo dottore" definisce il computer in uso come ospitante l'interfaccia che verrà utilizzata dal medico durante gli esercizi;
- "Schermo bambino" definisce il computer in uso come ospitante l'interfaccia che verrà utilizzata dal bambino durante gli esercizi, che si occuperà quindi di proporre gli esercizi.

A.2 Utente Medico

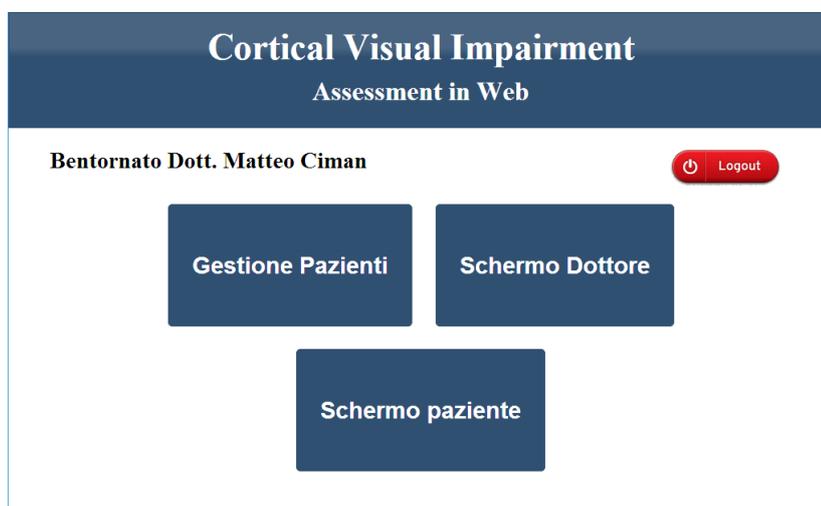


Figura A.1: Schermata principale del sistema per il medico

A.2.1 Schermo bambino

Selezionando l'opzione "Schermo bambino", si indica al sistema come la pagina in questione verrà utilizzata per presentare al bambino gli esercizi. Tale scelta dovrebbe essere effettuata nel computer dotato di schermo touch e dedicato al bambino.

Questo significa che tale pagina si metterà "in attesa" di ulteriori istruzioni provenienti dal server, ovvero quelle indicanti quale gioco è stato scelto dal dottore.

Una volta ricevuta tale informazione, il sistema si collegherà all'home page del gioco selezionato, attendendo ulteriori istruzioni, che in questo caso saranno le impostazioni di gioco necessarie per istanziare un nuovo esercizio. Come possiamo osservare, fino a quando non inizia l'esercizio vero e proprio, non sarà necessaria alcuna interazione con l'applicazione, in quanto si comporterà in maniera automatica sulla base di quelle che sono le istruzioni fornite dal dottore.

A.2.2 Schermo medico

Selezionando l'opzione "Schermo Medico", si indica all'applicazione che la schermata corrente verrà utilizzata dal medico. Questo significa che l'applicazione fornirà la possibilità di selezionare il paziente, preparare gli esercizi ed infine di seguire, durante lo svolgimento dell'esercizio, il comportamento del bambino.

Il primo passo che il medico deve compiere è quello di selezionare il bambino a cui far svolgere l'esercizio. Le possibilità in questo caso sono due:

1. selezionare un bambino tra quelli già inseriti all'interno del sistema e associati a quel dottore, utilizzando il menù di selezione posto nella parte sinistra della pagina (Figura A.2);



Seleziona un bambino tra quelli già presenti

Ciman Matteo
Paolino Paperino
Rossi Mario

Figura A.2: Selezione di un bambino precedentemente inserito nel sistema

2. inserire un nuovo bambino all'interno del sistema, cliccando prima sul bottone "Inserisci" nella parte destra della pagina, e completando poi la form visualizzata con tutte le informazioni richieste (Figura A.3).



Oppure puoi inserirne uno nuovo

Nome:

Cognome:

Maschio Femmina

Data Nascita:

Disabilità

Inserisci

Figura A.3: Form per l'inserimento di un nuovo bambino

Nell'inserimento delle informazioni relative al nuovo bambino, particolare attenzione deve essere posta nella definizione del livello di disabilità, in quanto attraverso tale informazione è possibile associare degli esercizi di default che possono essere utilizzati come punto di partenza per la diagnosi e la riabilitazione. Ovviamente, a livello di disabilità più elevato corrisponde un grado di complessità dell'esercizio più basso, sulla base delle indicazioni fornite dai

A.2 Utente Medico

medici.

Successivamente è richiesto al medico di selezionare il gioco tra quelli presenti (Figura A.4) ed infine, cliccando sul bottone “Comincia”, si verrà reindirizzati alla pagina di definizione dei parametri.

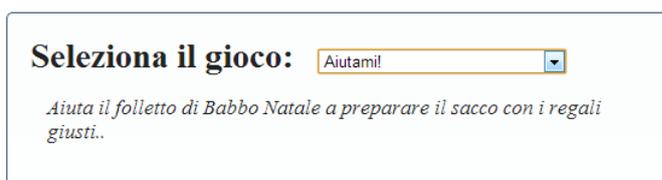


Figura A.4: Selezione del gioco con descrizione

Parametri di gioco Prendimi!

Selezionando il gioco *Prendimi!* si verrà reindirizzati alla pagina che permette di definire le impostazioni per l'esercizio, come quella presentata in Figura A.5. Le impostazioni che verranno inizialmente presentate sono quelle attualmente associate al bambino all'interno del database, che possono essere o specifiche per il bambino stesso (ovvero, è stato definito un particolare set di valori) o quelle default per il suo livello di disabilità (nel caso in cui non sia ancora stato impostato un esercizio specifico). Questo significa che il medico deve inserire le impostazioni di gioco per un bambino solamente la prima volta, in quanto per le visite successive il sistema presenterà l'ultimo insieme di impostazioni inserito, permettendo quindi di far ripetere al bambino lo stesso esercizio. Inoltre, se i parametri di default si adattano al meglio alle caratteristiche del bambino, è possibile utilizzare tali impostazioni per tutto il tempo necessario, senza che questo comporti un intervento da parte del medico. Tramite la schermata presentata in Figura A.5 è possibile:

- definire gli spostamenti possibili per la figura, ovvero:
 - movimento verso destra
 - movimento verso sinistra
 - movimento verso l'alto
 - movimento verso il basso
- imporre alla figura di cominciare dal centro dello schermo all'inizio di ogni animazione;

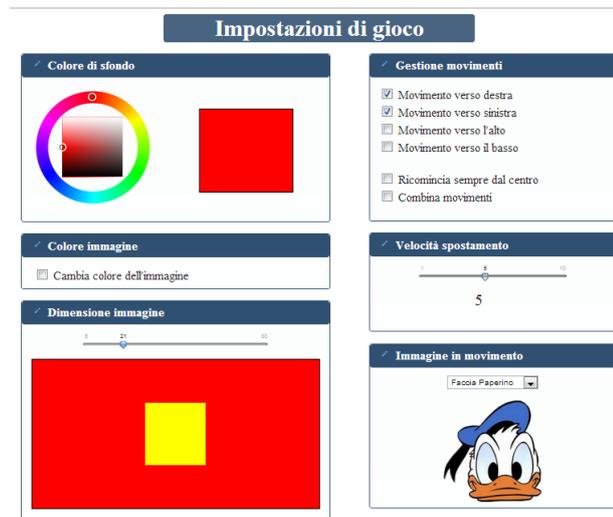


Figura A.5: Schermata di personalizzazione delle impostazioni per il gioco *Prendimi!*

- permettere o meno la combinazione dei movimenti selezionati precedentemente. Questo permette di inserire dei movimenti più complessi e quindi non adatti a tutti i bambini;
- modificare il colore di sfondo;
- modificare o meno il colore dell'immagine utilizzata;
- modificare la dimensione dell'immagine;
- modificare la velocità di spostamento dell'immagine;
- modificare l'immagine utilizzata per lo spostamento.

Una volta completata la definizione delle impostazioni, il click sul bottone “Comincia” comunica al client del bambino le impostazioni di gioco, permettendo quindi la costruzione dell'esercizio.

Nella schermata successiva, cliccando sul bottone “Ok” del messaggio che comunica che il sistema è pronto per iniziare, è possibile dare il via all'applicazione ed al gioco per il bambino (Figura A.6).

A.2 Utente Medico

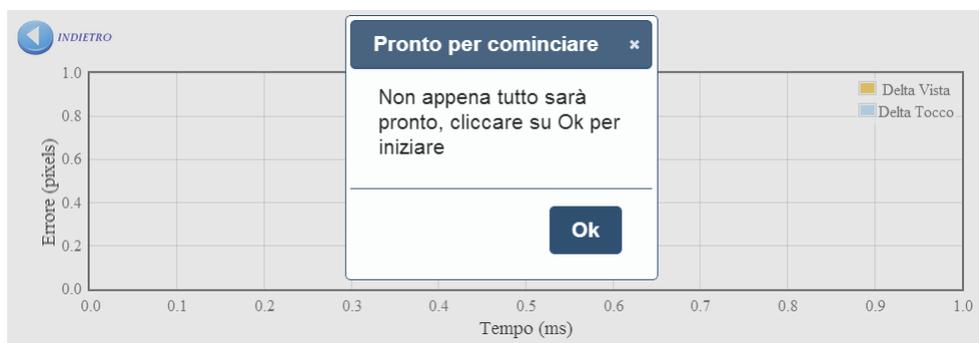


Figura A.6: Messaggio visualizzato per comunicare che il sistema è pronto a partire

Report online Prendimi!

Durante l'esecuzione dell'esercizio da parte del bambino, il medico ha la possibilità di vedere in tempo reale quelle che sono le prestazioni ed il comportamento del bambino.

L'indicazione fornita al medico è in questo caso un grafico che visualizza, rispetto al tempo trascorso, la distanza tra il centro dell'immagine e dove il bambino sta guardando o toccando. Un esempio di grafico è presentato in Figura A.7.

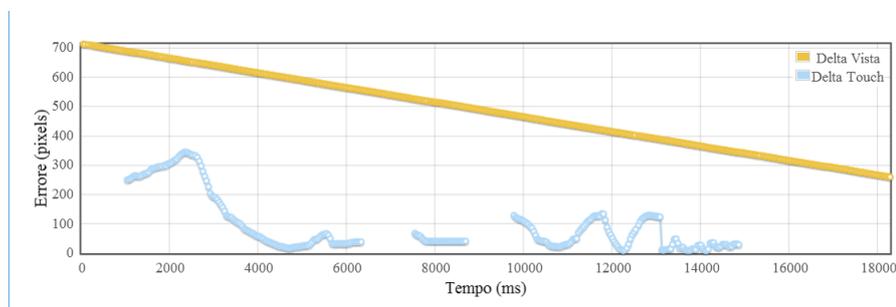


Figura A.7: Esempio di grafico presentato al medico per il gioco Prendimi!

In Figura A.7 l'insieme di punti di colore azzurro indica l'errore commesso dal bambino nell'uso del tocco, mentre l'insieme di punti arancioni rappresenta l'andamento della vista durante l'esercizio (nel caso degli esercizi svolti a casa tale sequenza di punti non è presente, in quanto non viene effettuato tracciamento degli occhi). Collocando il mouse sopra uno dei punti presenti nel grafico, il sistema dà la possibilità di valutare, in quell'istante di tempo, qual era la posizione dell'immagine, dell'occhio e del tocco (Figura A.8), mentre seguendo sempre con il mouse l'andamento della curva disegnata dai punti, è possibile vedere un'animazione che simula come l'esercizio è stato

svolto dal bambino.

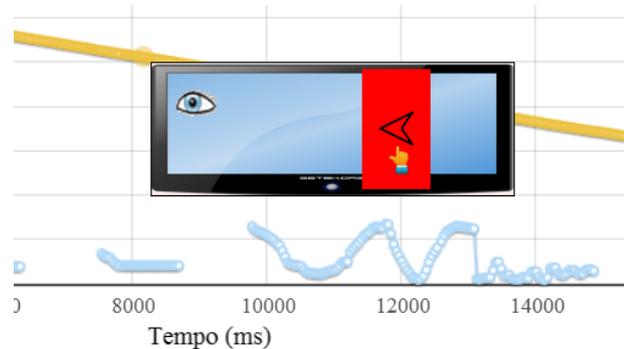


Figura A.8: Esempio di tooltip di dettaglio per il gioco Prendimi!

Se il medico si rende conto che la velocità impostata inizialmente è troppo elevata per il bambino, è possibile modificarla (con effetto valido a partire dal prossimo cambio di direzione dell’immagine) attraverso il pannello, mostrato in Figura A.9, presente sotto il grafico.

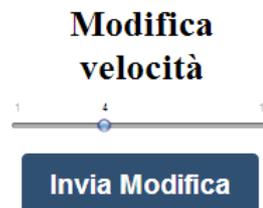


Figura A.9: Controllo per la modifica della velocità per il gioco Prendimi!

Spostando la barra che indica la velocità attuale, il bottone “Invia Modifica” verrà reso attivo e sarà quindi possibile inviare il nuovo valore della velocità al client del bambino che gestisce il gioco.

Infine, è possibile navigare il grafico attraverso le frecce mostrate in Figura A.10a, e, nel caso in cui il medico lo ritenga necessario, interrompere il gioco attraverso il bottone “Interrompi Gioco” mostrato in Figura A.10b.



(a) Frecce di navigazione del grafico



(b) Bottone per l’interruzione del gioco

Figura A.10: Sistema di controllo del gioco Prendimi!

A.2 Utente Medico

Parametri gioco Aiutami!

La schermata relativa alle impostazioni del gioco *Aiutami!* permette la definizione delle sequenze di immagini che si susseguono durante il gioco. Se chiamiamo una sequenza di immagini un livello, ogni livello è costituito da un certo numero di immagini chiamate *target* ed un certo numero di immagini chiamate *distrattori*. Le immagini *target* (indicate con la lettera T) sono le immagini che appartengono alla famiglia *target* per quel determinato livello, ovvero la famiglia di oggetti che il bambino deve identificare all'interno della sequenza di immagini che vengono presentate. La costruzione dei livelli è possibile attraverso la schermata in Figura A.11.

Lettera	Nome	Immagine
T	Gatto	
T	Coniglio	
D	Occhiali	
D	Orologio bambini	
T	Tartaruga	
D	Ombrello	
T	Maiale	

Figura A.11: Schermata per la costruzione dei livelli del gioco *Aiutami!*

Come possiamo osservare, per ogni livello vi è definita la famiglia *target*, scelta tra tutte le famiglie di immagini definite, il tempo massimo lasciato al bambino per decidere se inserire o meno un'immagine all'interno del sacco, ed infine la sequenza di immagini che saranno presentate al bambino. Oltre alla scelta di quale immagini, tra tutte quelle a disposizione, presentare al bambino, è data la possibilità, attraverso i due bottoni mostrati in Figura A.12, di modificare l'ordine con cui immagini *target* e *distrattori* vengono presentate.



Figura A.12: Bottoni utilizzati per cambiare l'ordine delle immagini nel gioco *Aiutami!*

Per modificare l'ordine di un'immagine *target* o *distrattore*, è sufficiente cliccare sulla riga contenente l'immagine che si vuole spostare ed utilizzare i due bottoni per scambiarla con l'immagine immediatamente precedente o immediatamente successiva.

È possibile inoltre inserire un nuovo livello nell'esercizio proposto al bambino, come mostrato in Figura A.13. Per inserire un nuovo livello, è necessario scegliere la tipologia di livello tra quelle disponibili (dove il primo indice indica il numero di immagini *target*, mentre il secondo il numero di immagini *distrattori*), e poi cliccare su "Aggiungi Livello".

Figura A.13: Form di inserimento di un nuovo livello per il gioco *Aiutami!*

Il nuovo livello inserito sarà un livello completamente vuoto, senza cioè alcuna famiglia *target* definita, e senza un numero di secondi massimo di attesa definiti.

La sequenza di immagini sarà solamente costituita dal numero di immagini *target* e di immagini *distrattori* definiti dal livello, e sarà compito del medico andare a scegliere quale immagine utilizzare e come riordinarle per costruire la sequenza desiderata. La Figura A.14 mostra un nuovo livello inserito pronto per essere personalizzato.

Report online *Aiutami!*

Il report creato per il dottore per il gioco *Aiutami!* ha come scopo quello di mostrare al dottore il comportamento seguito dal bambino durante gli esercizi, e di fornire un riepilogo con le risposte date.

Per prima cosa, viene presentato un disegno che riproduce, simulando lo schermo visto dal bambino, l'interazione che ha il bambino con l'immagine visualizzata. Attraverso tale disegno, il medico può osservare il tracciato della posizione del tocco e del centro dell'immagine, per osservare quali movimenti compie il bambino durante il gioco. La Figura A.15 mostra un esempio di tale rappresentazione.

A.2 Utente Medico

1 2

Livello 1 - T: 4 x D: 4

Famiglia target:

Tempo massimo immagine:

T	<input type="text"/>
D	<input type="text"/>

Figura A.14: Nuovo livello pronto per essere impostato



Figura A.15: Esempio di rappresentazione per il medico del comportamento del bambino durante il gioco *Aiutami!*

Nella parte inferiore dello schermo viene visualizzato un report simile a quello in Figura A.16 che riassume, suddividendo per livello di gioco, il comportamento del bambino, e riporta il tempo medio di risposta (*First Response Time*, che indica il tempo atteso prima di avere una risposta da parte del bambino), il tempo medio di completamento (*Tempo di Completamento*, che indica il tempo necessario al bambino per fornire una risposta) ed il numero di risposte esatte e di risposte sbagliate.

Oggetto	Famiglia Target	First Response Time (ms)	Tempo di completamento (ms)	Esatte / Errate
Riepilogo	animali	3110.67	7936.00	2/1

Figura A.16: Report che riassume il comportamento del bambino durante il gioco

A.3 Gestione bambini

Entrando nella sezione “Gestione bambini” è possibile accedere allo storico che permette di visualizzare tutte le visite e gli esercizi casalinghi svolti dal bambino, con relativa valutazione.

Per prima cosa, è necessario selezionare il bambino di recuperare le informazioni, ed eventualmente filtrare i risultati ottenuti in base ad un particolare gioco (Figura A.17).

Selezione paziente: ---
 Selezione gioco: ---

Matteo Ciman
 Paperino Paolino
 Mano Rossi

Matteo Ciman
 Prendimi!

 Prendimi!
 Aiutami!

Figura A.17: Selezione del bambino e del gioco nella gestione delle visite

Una volta completata la selezione del bambino, vengono presentate le visite (o esercizi) sostenute, con relativa valutazione, come mostrato in Figura A.18.

A.3 Gestione bambini

Gioco Aiutami!					
Data	First Response Time (ms)	Completion Time (ms)	Ris. Corrette	Ris. Errate	Visualizza Esercizio
23/09/2012	4294	15929	1	0	➤
19/09/2012	4807	7363	1	0	➤
19/09/2012	4652	5316	1	0	➤
19/09/2012	4087	4739	1	0	➤
19/09/2012	3698	4087	1	0	➤
19/09/2012	4502	7036	1	0	➤
10/09/2012	1626	4652	4	0	➤
10/09/2012	2172	4960	4	0	➤

Figura A.18: Esempio di tabella riassuntiva per il gioco *Aiutami!*

Per ogni visita, è possibile visualizzare, cliccando sulla freccia corrispondente, delle informazioni più dettagliate che sono state memorizzate. Nel caso del gioco *Prendimi!*, vien riproposto il grafico costruito con le distanze di occhi e tocco rispetto al centro dell'immagine (Figura A.19), mentre nel caso del gioco *Aiutami!* viene proposto un riepilogo, livello per livello, del comportamento del bambino per ogni singola immagine visualizzata (Figura A.20).

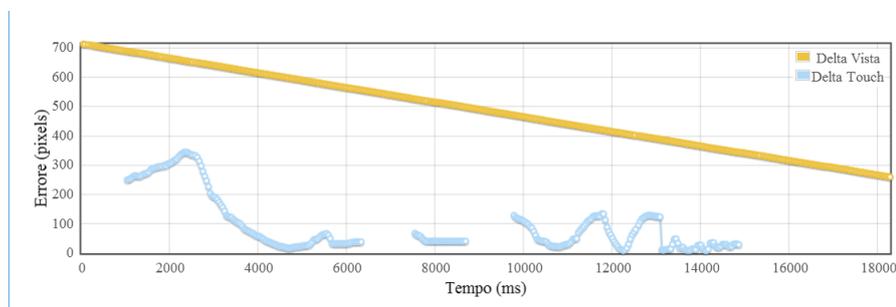


Figura A.19: Grafico di dettaglio per una visita con il gioco *Prendimi!*

 Famiglia target: animali				
Nome Oggetto	Ogg. Target	FRT (ms)	CT (ms)	Risp. Corretta
Oca	true	3544	13973	true
Oca1	true	1226	1758	true
	Valori medi:	2385	7866	2/0

 Famiglia target: macchine				
Nome Oggetto	Ogg. Target	FRT (ms)	CT (ms)	Risp. Corretta
Ferrari	true	1069	1608	true
	Valori medi:	1069	1608	1/0

 Famiglia target: fiori				
Nome Oggetto	Ogg. Target	FRT (ms)	CT (ms)	Risp. Corretta
Rosa	true	668	1271	true
	Valori medi:	668	1271	1/0

Figura A.20: Esempio dettagliato di riepilogo per il gioco *Aiutami!*

A.4 Avvio del server

Per avviare il server che gestisce la comunicazione tramite *WebSocket* e quindi l'intera infrastruttura dell'applicazione, è necessario eseguire il seguente programma:

```
java -jar /var/www/server/WebSocketServer.jar indirizzoIP
```

dove *indirizzoIP* indica l'indirizzo (ad esempio *ciman.math.unipd.it*) a cui il server è raggiungibile da parte dei diversi moduli dell'applicazione, ovvero il software di eye-tracking, il computer del medico e quello del bambino. Nel caso in cui tale indirizzo non venga fornito, il sistema utilizzerà *localhost* come indirizzo.

Appendice B

Manutenzione del prodotto

L'applicazione è stata sviluppata affinché il sistema sia in grado di funzionare senza particolari operazioni di manutenzione. Tuttavia, potrebbe presentarsi la necessità di modificare alcuni parametri, soprattutto per quel che riguarda le impostazioni dei giochi.

B.1 Gioco Prendimi!

B.1.1 Inserimento nuove immagini

Per poter inserire nuove immagini da poter utilizzare nel gioco, è necessario modificare il file XML “*catchMe/settings/images.xml*”.

Una nuova immagine, identificata dal tag XML `<image>`, è caratterizzata dai seguenti attributi:

- *id*: numero identificativo dell'immagine;
- *imageName*: nome dell'immagine;
- *fileName*: nome del file, compreso di estensione;
- *size*: dimensione dell'immagine, espressa come “*WxH*” dove *W* indica la larghezza mentre *H* l'altezza.

Le immagini specificate all'interno del file XML dovranno poi essere caricate, per poter essere utilizzate, all'interno della cartella “*catchMe/images/*”.

B.2 Aiutami!

La manutenzione relativa al gioco *Aiutami!* riguarda la gestione dei livelli e delle immagini associate.

B.2.1 Tipologie di livelli

Se si vuole introdurre una nuova tipologia di livello da utilizzare all'interno del gioco, con un particolare numero di immagini *target* e *distrattori*, è necessario modificare il file *“helpMe/settings/difficulties.xml”*.

Ogni livello, identificato dal tag `<typeLevel>`, ha un singolo attributo, *type*, definito come “TxD” dove *T* indica il numero di elementi *target*, mentre *D* il numero di elementi distrattori. un esempio di file è il seguente:

```
<difficulties >
  <typeLevel type="4x4" />
  <typeLevel type="6x4" />
  <typeLevel type="8x4" />
  <typeLevel type="4x6" />
  <typeLevel type="4x8" />
</difficulties >
```

B.2.2 Immagini e famiglie

L'elemento di base del gioco *Aiutami!* sono le famiglie, dove ogni famiglia è costituita da un insieme di immagini.

Ogni immagine è identificata dal tag `<image>`, con i seguenti attributi:

- *id*: identificativo numerico dell'immagine;
- *name*: nome associato all'immagine, come “Orologio”, “Ferrari”, ecc;
- *fileName*: nome del file dell'immagine.

Ogni immagine apparterrà ad una particolare famiglia. Una famiglia è identificata dal tag `<family>`, ed è costituita da una sequenza di tag `<image>` (che rappresentano la sequenza di immagini) e dai seguenti attributi:

- *type*: nome della famiglia di immagini, come “Animali”, “Fiori”, ecc;
- *audioFile*: nome del file audio (senza estensione) che riproduce l'indicazione per il bambino di inserire tale famiglia di oggetti dentro al sacco.

Un esempio di definizione di famiglia è la seguente:

```
<family type="animali" audioFile="sound_prova">
  <image id="1" name="Oca" fileName="oca.jpg" />
  <image id="2" name="Asino" fileName="asino.png" />
  <image id="3" name="Coniglio" fileName="coniglio.png" />
  <image id="4" name="Gallina" fileName="gallina.png" />
```

B.2 Aiutami!

```
<image id="5" name="Gatto" fileName="gatto.jpg" />
<image id="6" name="Maiale" fileName="maiale.png" />
<image id="7" name="Tartaruga" fileName="tartaruga.png" />
</family>
```

B.2.3 Costruzione esempi

Il gioco *Aiutami!* prevede prima del gioco vero e proprio una presentazione dove viene spiegato al bambino quelle che sono le operazioni che dovrà svolgere.

Per costruire tale presentazione, viene utilizzato il file XML *“helpMe/settings/examples.xml”* che definisce le immagini che vengono utilizzate e le operazioni che tali immagini dovranno svolgere. Ogni immagine è identificata dal tag *<example>*, ed ogni tag *<example>* è costituito dai seguenti attributi:

- *image*: indica il nome del file dell’immagine da utilizzare;
- *withHelp*: assume valore *true* o *false*, ed indica se deve essere visualizzata anche la freccia che indica di inserire l’immagine dentro il sacco;
- *insertIntoBag*: assume valore *true* o *false*, ed indica se l’immagine deve essere inserita o meno dentro il sacco;
- *soundBefore*: **opzionale**, contiene il nome del file audio (senza estensione) che deve essere riprodotto una volta che l’immagine è stata collocata al centro dello schermo (utilizzato ad esempio per riprodurre l’indicazione “Inseriamo il biscotto dentro il sacco”);
- *soundAfter*: **opzionale**, indica il nome del file audio (senza estensione) che deve essere riprodotto o una volta completata l’animazione che inserisce l’immagine dentro il sacco oppure per informare il bambino che tale oggetto non va inserito all’interno del sacco.

Un esempio di sequenza di immagini di esempio è il seguente:

```
<examples>
  <example image="rosa.png" withHelp="true "
    insertIntoBag="true" soundBefore="sound_prova"
    soundAfter="sound_prova"/>
  <example image="biscuit.png" withHelp="false "
    insertIntoBag="true" soundBefore="sound_prova"
    soundAfter="sound_prova"/>
```

```
<example image="ferrari.png" withHelp="false"
  insertIntoBag="false" soundAfter="sound_prova"/>
<example image="orologio.png" withHelp="false"
  insertIntoBag="false" soundAfter="sound_prova"/>
</examples>
```

B.2.4 Definizione dei livelli di default

L'applicazione permette di associare a ciascun bambino un livello di disabilità, dove tale livello associa degli esercizi di default che risultano essere standard ed utilizzati fino a quando il medico non decide di modificare tali esercizi in alcuni personalizzati sulle caratteristiche del bambino. Mentre gli esercizi specifici vengono salvati in modo automatico dal sistema, quelli generali devono essere gestiti manualmente.

Per questo motivo vi sono tre file XML distinti, *"helpMe/settings/low.xml"*, *"helpMe/settings/medium.xml"* e *"helpMe/settings/high.xml"* (rispettivamente per un livello di disabilità basso, medio o alto), che specificano le sequenze di immagini usate come sequenze di default.

Ogni esercizio è costituito da un insieme di livelli, indicati con il tag *<level>*, a cui sono associati i seguenti attributi:

- *targetFamily*: indica la famiglia di immagini *target*;
- *type*: indica la tipologia del livello, specificata nella sezione B.2.1;
- *maxTimeImage*: indica il massimo numero di secondi attesi prima di passare all'immagine successiva.

Ogni livello è costituito da una sequenza di immagini, dove ogni immagine è identificata dal tag *<image>* con i seguenti attributi:

- *type*: indica la tipologia di immagine, ovvero se l'immagine è un'immagine *target* o *distrattore*, e assume valori *T* o *D*;
- *imageID*: indica l'ID dell'immagine da utilizzare, dove l'ID è definito nel file *"helpMe/settings/images.xml"* (Sezione B.2.2).

Un esempio di livello è il seguente:

```
<level targetFamily="animali" type="6x4" maxTimeImage="25">
  <image type="T" imageID="5" />
  <image type="T" imageID="3" />
  <image type="D" imageID="11" />
  <image type="D" imageID="13" />
  <image type="T" imageID="7" />
```

B.2 Aiutami!

```
<image type="D" imageID="12" />
<image type="T" imageID="6" />
<image type="T" imageID="2" />
<image type="D" imageID="14" />
<image type="T" imageID="4" />
</level >
```


Bibliografia

- [1] O. Gaggi A. De Bortoli. PlayWithEyes: a new way to test children eyes. In *Proceedings of the IEEE International Conference on Serious Games and Applications for Health*, pages 190–193. SeGAH 2011, November 2011.
- [2] Junior Blind Of America. How ipads could help 'blind' children to see, http://www.juniorblind.org/site/images/stories/pdfs/news_articles/2011/ipadsmailonline-10-6-11.pdf, November 2011.
- [3] Alberto De Bortoli. PlayWithEyes. Master's thesis, Università di Padova, 2011.
- [4] M. Rocchetti C. E. Palazzi, G. Marfia. Combining web squared and serious games for crossroad accessibility. In *Proceedings of the IEEE International Conference on Serious Games and Applications for Health*, 2011.
- [5] M. Rocchetti C. E. Palazzi, L. Teodori. Path 2.0: A participatory system for the generation of accessible routes. In *of the IEEE International Conference on Multimedia and Expo (ICME'10)*, 2010.
- [6] Goauich Abdelkader Di Loreto Ines. Mixed reality serious games: The therapist perspective. In *Proceedings of the IEEE International Conference on Serious Games and Applications for Health*, 2011.
- [7] Mills D.L. Internet time synchronization: the network time protocol. *IEEE Transactions on Communications*, Volume 39(Issue: 10):pages 1482–1493, October 1991.
- [8] Flot. Flot plotting plugin, <http://www.flotcharts.org>.
- [9] Giorgia Galiazzo. Portable rich internet application: Un caso di studio applicato ai videogiochi. Master's thesis, Università di Padova, 2012.

- [10] Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions On Pattern Analysis and Machine Intelligence*.
- [11] Squishy Things Inc. Bubbles magic, <http://quinly.com/prod.html>.
- [12] Java-WebSocket. Java-websocket, <http://java-websocket.org>.
- [13] jQuery. jquery, <http://jquery.com>.
- [14] jQueryUI. jqueryui, <http://jqueryui.com>.
- [15] JSON. Specifiche del formato json, <http://json.org>.
- [16] M.D. Lemmon. Model-based clock synchronization in networks with drifting clocks. In *Proceedings of the International Symposium on Dependable Computing*, 2000.
- [17] Little Bear Sees LLC. Tap-n-see zoo, <http://www.littlebearsesees.org>.
- [18] Leisman G. Malkowicz DE, Myers G. Rehabilitation of cortical visual impairment in childred. *International Journal of Neuroscience*, 2006.
- [19] Modernizer. Modernizer, <http://www.modernizer.com>.
- [20] Node.js. Node.js, <http://nodejs.org>.
- [21] C.E. Palazzi O. Gaggi, G.Galiazzo. A serious game for predicting the risk of developmental dyslexia in pre-readers children. In *Proceedings of the 8th International Workshop on Networkin Issues in Multimedia Entertainment*. NIME 2012, July 2012.
- [22] PhoneGap. Phonegap, <http://phonegap.com>, 2009.
- [23] Kids Place. Baby aquarium, <http://kidsbestplace.com/>.
- [24] Hoyt C. S. Visual function in the brain-damaged child. *Eye*, 2003.
- [25] M.L. Sichitiu. Simple, accurate time synchronization for wireless sensor networks. In *IEEE Conference on Wireless Communications and Networking*, 2003.
- [26] Lehman SS. Cortical visual impairment in children: identification, evaluation and diagnosis. *Current Opinion in Ophthalmology*, 2012.
- [27] Tobii Technology. <http://www.tobii.com>.

BIBLIOGRAFIA

- [28] Andrea Torrente. Video gioco, serious games e gioco di simulazione a scuola, http://www.edscuola.it/archivio/software/videogiochi_a_scuola.pdf.
- [29] D. Veitch. Robust synchronization of absolute and difference clocks over networks. *IEEE/ACM Transactions on Networking*, 2009.
- [30] WebSocket. Websocket.org, <http://www.websocket.org>.
- [31] Yu-Chih Lin Greg C. Lee Yu-Tzu Lin, Ruei-Yan Lin. Real-time eye-gaze estimation using a low-resolution webcam. *Springer Science*.

Ringraziamenti

Per prima cosa, vorrei ringraziare la mia relatrice, la Professoressa Ombretta Gaggi. Sebbene ogni volta abbia sostenuto che fosse il suo lavoro, per entrambe le tesi svolte insieme a lei si è dimostrata sempre disponibile e professionale, in un modo difficilmente riscontrabile in altri docenti. Sicuramente è grazie a lei se ho avuto la possibilità di realizzare due progetti di cui sono particolarmente soddisfatto.

Ringrazi poi i miei genitori, perchè è grazie a loro se questi cinque anni, che segnano un passo sicuramente molto importante della mia vita, sono potuti diventare realtà, e ho potuto completare i miei studi raggiungendo questi risultati.

Ringrazio poi Anna, che in questi anni mi è sempre stata vicino, anche nei momenti più difficili e stressanti, sopportando con il sorriso tutte le varie peripezie che ho dovuto affrontare.

Ringrazio infine il gruppo dell'aula tesisti, Alberto C., Alberto F., Fabrizio, Andrea e Marco per la compagnia ed il divertimento di questi ultimi mesi. A loro voglio augurare un "In bocca al lupo" per il futuro.

Matteo