

Ringraziamo il Professor Angelo Gargantini per i preziosi insegnamenti, inoltre
vogliamo ringraziare le nostre famiglie per il sostegno ed il grande aiuto.

Fabio e Matteo

Indice

1	Introduzione	5
2	L'Ambliopia	7
2.1	Il disturbo	7
2.2	Trattamenti	9
2.2.1	Occlusione	9
2.2.2	Penalizzazione ottica	9
2.2.3	Penalizzazione farmacologica	9
2.2.4	Settorizzazione	9
2.3	L'idea di 3D4Amb	11
3	La tecnologia per il trattamento	13
3.1	Sistema operativo ed ambiente di sviluppo:	13
3.1.1	Perchè Android:	14
3.1.2	Perchè Android Studio:	14
3.2	Google cardboard	16
4	Car Racing Cardboard: l'applicazione	19
4.1	Il principio del trattamento tramite il gioco:	19
4.2	Il gioco:	21
4.2.1	Obiettivo:	21
4.2.2	Doppia Viewport:	22
4.3	Guida all'utilizzo:	23

5	Il software	29
5.1	Architettura di sistema	29
5.2	Casi d'uso	31
5.3	Package Diagram	35
5.4	Component diagram	36
5.4.1	Il sistema di Car Racing Cardboard	36
5.4.2	L'applicazione Android	36
5.4.3	Il server	37
5.4.4	Dettaglio applicazione	37
5.5	Class diagram	38
5.6	Sequence diagram	39
6	Analisi degli Algoritmi	41
6.1	Penalizzazione progressiva delle immagini	41
6.2	Generazione random scena	43
6.2.1	Nemici	43
6.2.2	Panorama	44
6.3	Cifratura password	45
6.4	Servizio web	46
6.5	Thread	50
6.6	AsyncTask	51
7	Toll utilizzati	53
7.1	GitHub	53
7.2	Crashlitics	53
8	Social media	55
9	Conclusioni	57
9.1	Paper	58

Capitolo 1

Introduzione

Il progetto 3D4Amb mira a sviluppare un sistema basato sul 3D per la diagnosi e il trattamento dell'ambliopia nei bambini piccoli.

Sfrutta la tecnologia 3D active shutter per garantire una visione binoculare, cioè per mostrare immagini diverse all'occhio normale e all'occhio pigro. Essa dovrebbe consentire una facile diagnosi dell'ambliopia e il suo trattamento per mezzo di giochi interattivi e attività di intrattenimento. Non dovrebbe soffrire dei problemi del trattamento classico dell'occlusione, è adatto ad un uso domestico, e potrebbe, almeno in parte, sostituire l'occlusione dell'occhio normale.

L'obiettivo principale di questo progetto di ricerca, denominato 3D4Amb, è di sviluppare un sistema per la diagnosi e il trattamento di ambliopia, basata sulla visione binoculare in modo accessibile. Con il termine accessibile si intende: poco costoso, user friendly, adatto per uso domestico e facilmente estendibile.

Tutte le informazioni sul progetto sono reperibili sul sito: <http://3d4amb.unibg.it/> Car Racing Cardboard è un'applicazione per la piattaforma Android, il suo scopo è curare una patologia come l'ambliopia attraverso un gioco, in modo tale da far divertire il paziente ed allo stesso tempo sottoporlo al trattamento per la cura della sua malattia.

Capitolo 2

L'Ambliopia

2.1 Il disturbo

L'ambliopia è una condizione di ridotta acuità visiva mono o bilaterale e si manifesta indipendentemente da causa organica. È dovuta ad una inadeguata stimolazione visiva durante il periodo plastico del sistema visivo, ossia il periodo che va dalla nascita fino ai sette anni.

Il soggetto in cui è presente l'ambliopia soffre di un'alterazione della visione dello spazio: le immagini che provengono dagli occhi non vengono correttamente rielaborate all'interno del cervello. Questo causa una scorretta comprensione dello spazio che lo circonda e causa una percezione scorretta della profondità, dei movimenti e dei contrasti. È presente nel 2-4% della popolazione, la sua incidenza è più elevata in associazione con alcune condizioni quali prematurità, sindrome di Down, patologia neurologica e familiarità per ambliopia o strabismo. Spesso le persone non si accorgono nemmeno di esserne affette fino ai 20-30 anni, per questo è fondamentale la diagnosi.

Può colpire i bambini dalla nascita fino ai 7 anni, età in cui il sistema visivo raggiunge la sua maturità. Durante questo periodo iniziale l'ambliopia può essere trattata e prevenuta, mentre superata questa fase l'istaurazione della malattia diventa impossibile, ma, nel caso fosse presente, essa risulta irreversibile. L'ambliopia funzionale deve essere distinta dall'ambliopia organica, la quale è un impoverimento della visione, causata da anomalie strutturali dell'occhio o del cervello, che sono indipendenti

dagli input sensoriali.

L'ambliopia funzionale è reversibile se trattata con la stimolazione visiva adeguata, mentre quella organica non subisce alcun beneficio da una stimolazione visiva. Il videogame di rebalance ha quindi effetto solo sull'ambliopia funzionale e non su quella organica.

2.2 Trattamenti

Il trattamento precoce dell'ambliopia è fondamentale per ottenere buoni risultati, la correzione oculare avviene in diversi modi:

2.2.1 Occlusione

La terapia occlusiva si basa sulla copertura dell'occhio sano per stimolare l'occhio ambliope. Solitamente nei pazienti affetti anche da strabismo l'occlusione avviene a tempo pieno; tuttavia l'occlusione a tempo pieno, o occlusione totale, può causare un'ambliopia inversa nei soggetti sotto i 4-5 anni. Per evitare l'insorgere di questo ulteriore problema, la condizione dell'ambliopia deve essere monitorata ogni settimana per l'età del bambino. Per esempio, un bambino di due anni va monitorato ogni due settimane, uno di tre ogni tre settimane. Nel caso in cui venga riscontrato che il paziente non sopporta un'occlusione totale si applica un'occlusione parziale, ossia solo qualche ora al giorno.

2.2.2 Penalizzazione ottica

Attuata con filtri di Bangerter (lenti con gradi diversi di opacizzazione, a seconda della entità della penalizzazione che si vuole attuare) o con lenti più forti o più deboli poste davanti all'occhio sano per costringere quello malato a lavorare; come l'occlusione parziale, viene attuata o per coloro che necessitano di una occlusione "morbida" o come terapia di mantenimento.

2.2.3 Penalizzazione farmacologica

Viene effettuata con un collirio cicloplegico instillato nell'occhio sano per escluderlo dal processo di visione e costringere quello malato a lavorare.

2.2.4 Settorizzazione

Consiste nella copertura di parte del campo visivo dell'occhio sano con pellicole adesive traslucide sugli occhiali.

Per il trattamento dell'ambliopia sono stati proposti anche degli stimolatori visivi di tipo elettrico (trattamento CAM-Cambridge Stimulator e trattamento Flicker): vengono inviati stimoli luminosi di vario tipo sulla retina dell'occhio ambliope, forzandolo a trasmettere l'impulso luminoso al cervello, e riattivando così i canali impigriti dall'ambliopia. L'efficacia di queste metodiche è ancora oggi molto dibattuta. Recentemente si stanno utilizzando farmaci neuroprotettivi come sostegno alla terapia occlusiva: studi recenti indicano che in questo modo viene potenziato l'effetto della terapia occlusiva e viene più facilmente stabilizzato il miglioramento della funzione visiva.

2.3 L'idea di 3D4Amb

3D4Amb ha ideato un sistema basato sulle tecnologie 3D per consentire la visione binoculare. L'uso classico di un sistema 3D è quello di fornire ai due occhi diverse immagini della stessa scena con angoli di visualizzazione leggermente sfalsati che corrispondono ai diversi punti di vista dell'occhio destro e sinistro. Questa visione produce un'illusione di profondità della scena ed è la base della realtà virtuale.

Il principio primario del sistema è che all'occhio ambliope (o occhio pigro) e all'occhio normale sono mostrate due immagini differenti ma correlate. Questo principio può essere utilizzato nella pratica per il trattamento di ambliopia, andando a mostrare all'occhio ambliope la parte più interessante dei frame della clip o del gioco, mentre all'occhio non ambliope (o buono) viene mostrata la parte meno interessante. Il contenuto da mostrare al paziente (gioco o immagine) viene diviso da 3D4Amb in due parti, una per l'occhio destro (occhio ambliope, Figura 2.1) e una per l'occhio sinistro (occhio buono, Figura 2.1). Il software 3D4Amb deciderà cosa inviare ad entrambi gli occhi a seconda del tipo di trattamento suggerito dal medico. Si noti che l'occhio pigro del bambino è più stimolato a lavorare, ma l'occhio sano non è "patchato". Il cervello del paziente ha il compito di unire le due immagini per visualizzare il frame completo con successo ed eseguire correttamente operazioni semplici in caso di gioco interattivo. Per assicurarsi che il paziente possa unire le due immagini sono presenti un numero significativo di elementi comuni ad entrambe le immagini. Notare che il frame finale è una rappresentazione bidimensionale in quanto l'obiettivo è quello di non stimolare la visione stereo del paziente (almeno inizialmente).[6]

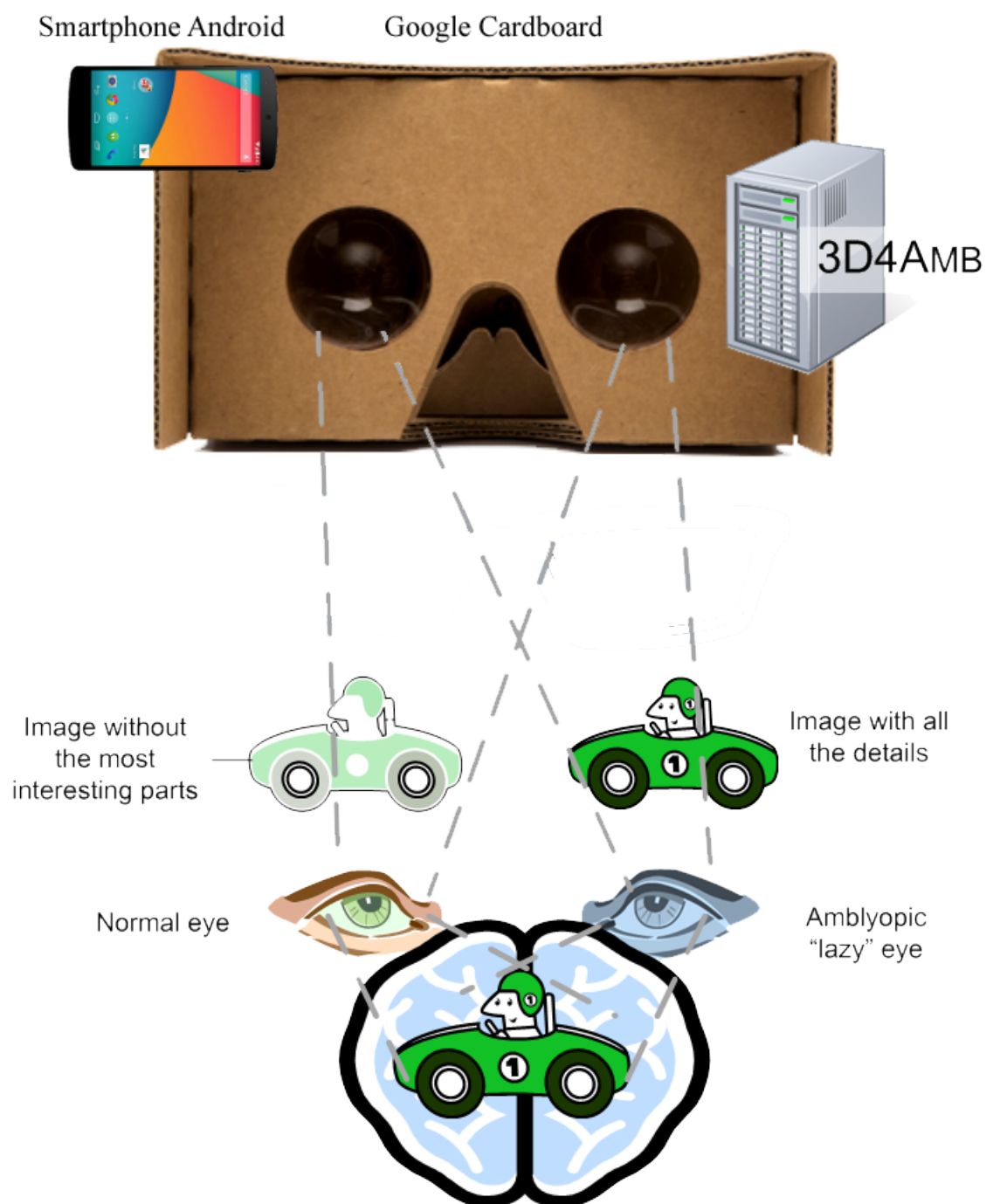


Figura 2.1: La tecnologia di 3D4Amb

Capitolo 3

La tecnologia per il trattamento

3.1 Sistema operativo ed ambiente di sviluppo:

L'applicazione è stata progettata e sviluppata per il mondo Android. I software utilizzati per l'implementazione sono stati:

- L'ultima versione del JDK, il kit di sviluppo per la tradizionale programmazione Java, reperibile dal sito Oracle¹.
- L'IDE (ambiente di sviluppo integrato) Android Studio, alternativa fresca e promettente ad Eclipse. La sua installazione include il pacchetto di strumenti Android SDK, il quale è costituito da programmi, emulatori e piattaforme per ogni versione di Android. La sua composizione inoltre non è immutabile, ma viene gestita tramite il programma Android SDK Manager, avviabile dall'IDE, grazie al quale siamo stati in grado di profilare le piattaforme e gli strumenti presenti nel SDK nella maniera più congeniale al nostro lavoro.

In particolare si è deciso che il requisito minimo che deve possedere lo smartphone dell'utente che utilizza l'applicazione sia un livello di API(Application Programming Interface) 15, ovvero dal sistema operativo Android 4.0.3² in poi.

¹<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

²(ICE.CREAM.SANDWICH_MR1)

3.1.1 Perchè Android:

La scelta di sviluppare l'applicazione per il sistema operativo Android, anziché ad esempio per iOS o Windows Phone, è figlia delle seguenti riflessioni: Innanzitutto il sistema Android sta dilagando. Non è più solo questione di smartphone o tablet. Si sta imponendo come sistema operativo in grado di animare qualsiasi dispositivo più o meno mobile tanto da apparire, in prospettiva non troppo futuristica, una presenza sempre più costante nel nostro quotidiano. Gli è stata inoltre attribuita la più veloce diffusione mai vista per un sistema operativo mobile. Gran parte del merito è dovuto alle radici ben salde nel mondo open source. Il sistema operativo Android, infatti, è figlio di Linux, ed ha attirato l'interesse di tanti sviluppatori che per anni si sono stretti intorno ai grandi bacini del software libero. Accoglie inoltre in sé tutto il meglio di quanto è stato ideato per supportare lo sviluppo del web, desktop e mobile sia in termini di pattern progettuali che di librerie software. Per decretare un successo tanto ampio è necessario che ci sia anche un forte riscontro di pubblico. Sicuramente un qualcosa che ha contraddistinto questo sistema è stata la sua adozione da parte di dispositivi molto diversi tra loro, non solo per tipologia (smartphone piuttosto che tablet), ma soprattutto per fasce di prezzo, da poche decine di euro fino a cifre piuttosto significative. Ciò ne ha permesso una diffusione molto diversificata trasversalmente alle diverse categorie sociali, ma ha causato di riflesso una frammentazione notevole dello scenario applicativo costringendo gli sviluppatori ad una particolare cura degli aspetti di adattamento alle caratteristiche del dispositivo ospite. Proprio in questo, Android ha dimostrato la sua grande modernità offrendo tutto il supporto necessario per permettere all'applicazione in esecuzione di adeguarsi ad ogni circostanza.

3.1.2 Perchè Android Studio:

Dopo aver scelto per quale sistema operativo sviluppare, un'altra importante scelta è stata quella di identificare l'IDE più adatto alle nostre esigenze. Android Studio è un'alternativa giovane ma molto promettente rispetto ad Eclipse. Esso è sponsorizzato direttamente da Google e pensato appositamente per Android. Gli elementi di



Figura 3.1: Android Logo

Android Studio che spiccano maggiormente sono:

- l'utilizzo di Gradle come strumento di build automation, atto quindi ad accompagnare lo sviluppatore nelle fasi di build, sviluppo, test e pubblicazione della propria app;
- disponibilità di un gran numero di template per la realizzazione di applicazioni già in linea con i più comuni pattern progettuali;
- editor grafico per la realizzazione di layout, molto pratico e dotato di un ottimo strumento di anteprima in grado di mostrare l'aspetto finale dell'interfaccia che si sta realizzando in una molteplicità di configurazioni (tablet e smartphone di vario tipo).

3.2 Google cardboard

Google cardboard è una piattaforma per la realtà virtuale sviluppata da Google. Creato da David Coz e Damien Henry, due ingegneri della Google, nel Google Cultural Institute a Parigi nel loro 20% “Innovation Time Off” [5]



Figura 3.2: Google Cardboard

Introdotta nel 2014 al “Google I/O developers conference for Android device”, è formata da un cartone ripiegabile e due lenti, l’utente deve inserirci lo smartphone (Figura 3.2). Questo sistema consente una visione stereoscopica, funziona con differenti smartphone e può essere facilmente usato dai bambini.

Google cardboard non consente un’ esperienza immersiva come Oculus rift, il quale necessita però di un PC collegato (ed è ancora in fase di sviluppo), ma è un modo facile ed economico per provare l’esperienza della realtà virtuale, inoltre la maggior parte delle applicazioni sono reperibili gratuitamente.

Esistono rischi per la salute?

Ad oggi non esistono evidenze scientifiche certe e rilevanti a supporto della tesi secondo cui le tecnologie 3D siano responsabili (o corresponsabili) della patogenesi di disturbi e difetti a carico dell’apparato della vista.

Tuttavia, è importante notare che l’assenza di prove a sostegno della tesi non implica

la sua totale infondatezza, cioè in altre parole almeno per il momento non siamo in grado di mettere in relazione con certezza i due elementi.

Capitolo 4

Car Racing Cardboard: l'applicazione

4.1 Il principio del trattamento tramite il gioco:

L' applicazione sviluppata in questo progetto di tesi, punta a sfruttare le potenzialità del Google Cardboard per correggere il disturbo dovuto all'ambliopia. Fino ad ora, il trattamento standard consolidato per la cura della malattia è stato l'occlusione visiva. Questa tecnica può però presentare alcuni limiti:

- in primo luogo la compliance (adesione del paziente alla terapia) è un fattore limitante da non trascurare, soprattutto quando la profondità dell'ambliopia richiede tempi lunghi di trattamento;
- Inoltre, l'occlusione è gravata dalla tendenza alla perdita di efficacia nel breve periodo. Infatti fino al 50% dei benefici ottenuti vengono persi, se i risultati funzionali non sono stati resi stabili, prolungando il trattamento per un tempo sufficientemente lungo, dopo che si è registrato il miglioramento del visus.

Probabilmente la tendenza alla perdita di effetto è dovuta, almeno in parte, al fatto che l'occlusione non affronta un altro importante aspetto della terapia dell'ambliopia, cioè la stimolazione diretta dell'occhio con il deficit visivo. Sulla base di questo, si è quindi pensato di realizzare un gioco interattivo che punti alla cura dell'occhio ambliope in modo più coinvolgente. Grazie quindi all'accessibilità dei

Google Cardboard, all'evoluzione del mondo Android ed all'intrattenimento volontario ed intrinsecamente motivato dato dal gioco, è stato possibile creare un metodo di cura innovativo.



Figura 4.1: Interactive game scene

4.2 Il gioco:

4.2.1 Obiettivo:

L' applicazione Car Racing Cardboard consiste in un gioco il cui obiettivo è quello di ottenere il maggior punteggio possibile. L'utente ha la possibilità di manovrare un veicolo attraverso tre corsie. Il punteggio del gameplayer incrementa ogni volta che si evita un ostacolo, costituito dalle macchine che pervengono nella nostra direzione sulle tre corsie. Inizialmente la posizione del veicolo controllato dall'utente è nella corsia centrale, ma non appena inizia il gioco, si ha la possibilità di spostarsi a destra o a sinistra. Lo spostamento del veicolo viene gestito tramite un dispositivo bluetooth collegato allo smartphone o tramite degli auricolari aventi i pulsanti UP, DOWN e PLAY. Oltre al punteggio, esistono altre due variabili dipendenti dal comportamento dell'utente: una che conteggia le vite a disposizione ed una che indica il livello raggiunto. Inizialmente il gameplayer ha a disposizione un totale di tre vite, le quali decrementano nel momento in cui avviene una colluttazione con una macchina proveniente dal verso opposto. Il livelli inoltre incrementano col passare del tempo. La difficoltà del gioco, caratterizzata dal numero e dalla velocità dei nemici, incrementa con l'aumentare dei livelli.

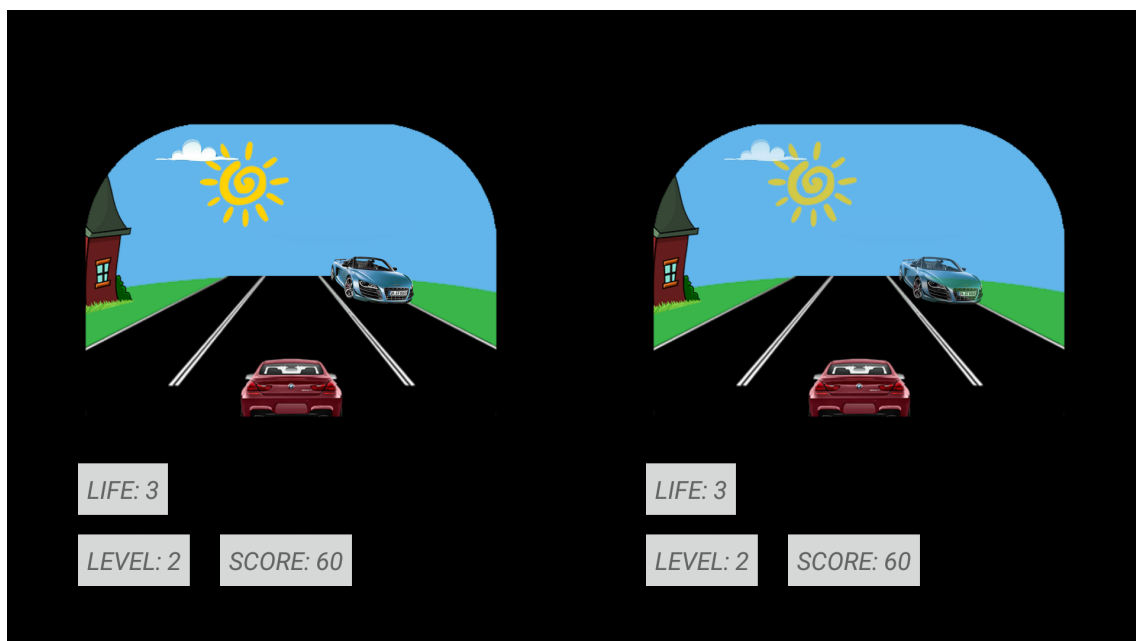


Figura 4.2: Game scene

4.2.2 Doppia Viewport:

Come si può vedere dall'immagine 5.2, l'applicazione viene realizzata renderizzando sullo smartphone dell'utente una doppia Viewport. Questo sdoppiamento, grazie alla tecnologia offerta dal Google Cardboard, permette all'utente una visione stereoscopica del gioco. Le due Viewport sono inoltre fondamentali per il principio di funzionamento della nostra applicazione. Infatti, grazie ad esse è possibile trattare in modo differente l'occhio sinistro dall'occhio destro, e più specificatamente l'occhio ambliope dall'occhio sano. Prima dell'inizio del gioco vero e proprio, l'applicazione dà la possibilità di scegliere l'occhio pigro: In questo modo si dà all'applicazione

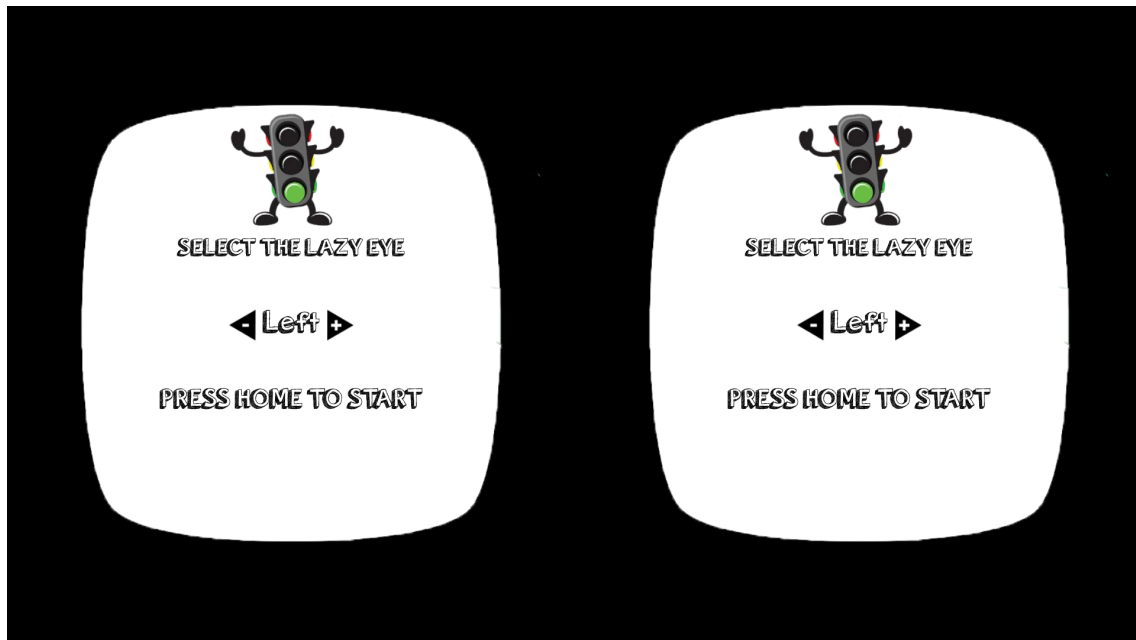
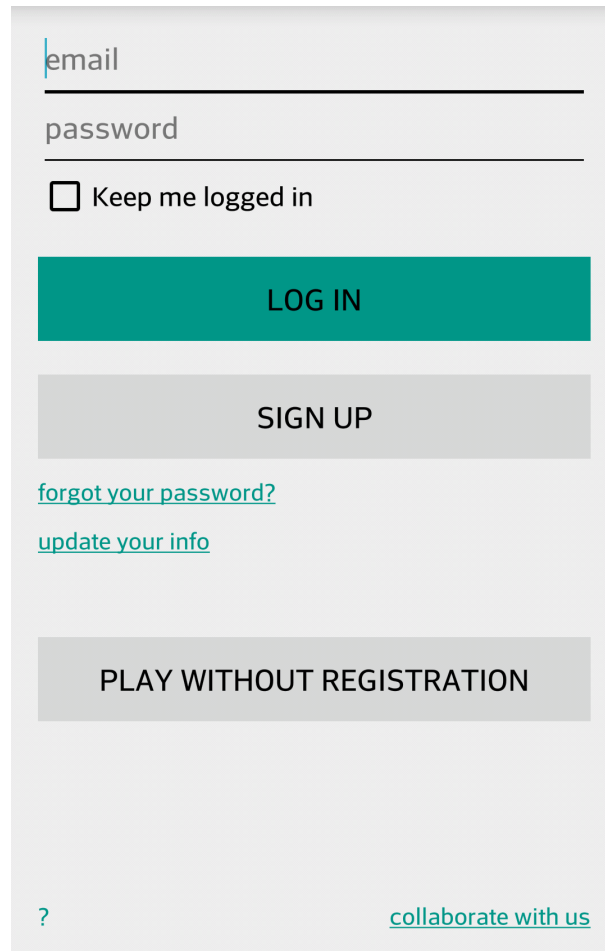


Figura 4.3: Choose the lazy eye

l'opportunità di trattare l'occhio pigro in modo da svilupparne la visione. In particolare l'applicazione andrà a penalizzare l'occhio sano riducendo l'opacità di alcuni elementi che caratterizzano la Viewport come ad esempio i veicoli nemici o il paesaggio, in modo da stimolare e sforzare l'occhio pigro ad allenarsi. La penalizzazione avviene in modo dinamico ed è la chiave della cura. Essa aumenta con l'aumentare dello score e dei livelli raggiunti dall'utente.

4.3 Guida all'utilizzo:

Dopo aver scaricato l'applicazione da Google Play Store ¹ è possibile fruire dei contenuti di Car Racing Cardboard. Connettere quindi lo smartphone agli auricolari, ma aspettare ad inserirlo nel visore 3D. La schermata di avvio è la seguente (Figura 4.4): Da questa schermata è possibile effettuare il log in, se l'utente è già registrato,



The screenshot shows the login interface of the Car Racing Cardboard app. It features a light gray background with a white login form. The form includes two input fields: 'email' and 'password', both with placeholder text. Below the password field is a checkbox labeled 'Keep me logged in'. There are two main buttons: a teal 'LOG IN' button and a gray 'SIGN UP' button. Below the 'SIGN UP' button are two links: 'forgot your password?' and 'update your info'. At the bottom of the form is a gray button labeled 'PLAY WITHOUT REGISTRATION'. In the bottom left corner of the screen is a question mark icon, and in the bottom right corner is a link labeled 'collaborate with us'.

Figura 4.4: Schemata di avvio

in caso contrario è possibile registrarsi premendo sul bottone “SIGN UP”. Prima della schermata di registrazione il sistema mostra un messaggio per ricordare all'utente (qualora esso non sia il paziente, ad esempio un genitore) di inserire i dati del paziente, poichè non avrebbe senso raccogliere informazioni forvianti (Figura 4.5) Cliccando su “CONTINUE” si entra nella vera schermata di registrazione (Figura 4.6):

¹<https://play.google.com/store/apps/details?id=it.unibg.p3d4amb.carracingcardboard&hl=it>

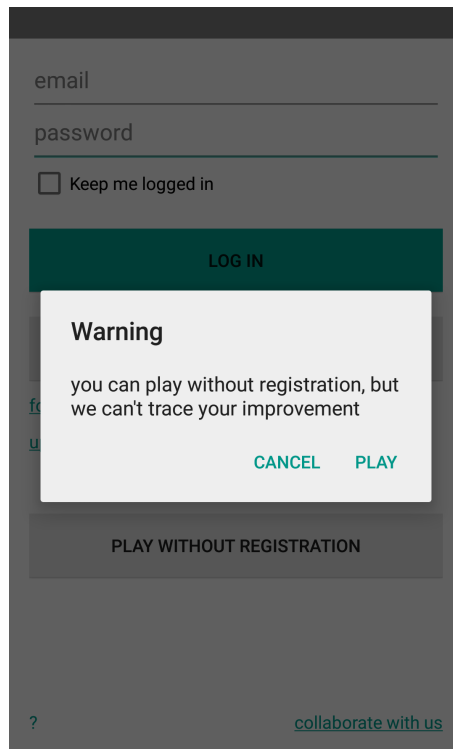


Figura 4.5: Pop up di warning

Qui bisogna inserire: Nome, Cognome, email, data di nascita, colore preferito e password. Il colore viene utilizzato in caso di password dimenticata, come domanda di sicurezza. Nell'eventualità in cui un utente si dimentichi la sua password, dalla schermata iniziale (Figura 4.4) premendo su “forgot your password?” si accede al pop up di ripristino (Figura 4.7).

Inserendo l'indirizzo email valido ed il colore corretto, il sistema invia una mail con la nuova password, che sarà possibile ri-personalizzare premendo su “update your info” (Figura 4.4). Sempre dalla schermata iniziale è possibile, premendo su “?” aprire il sito dedicato a Car Racing Cardboard². Cliccando su “collaborate with us” è possibile inviare una mail allo staff di 3D4Amb, questa possibilità è stata pensata per collaborazioni con medici e/o centri di ipovisione, interessati a collaborazioni(cFigura 4.8).

In questo caso, è stata prevista la possibilità per i dottori di gestire più pazienti con un unico account, infatti, il sistema è in grado di riconoscere un dottore registrato e lo reindirizza ad una pagina apposita (Figura 4.9).

²<http://3d4amb.unibg.it/3dcar/cardboard%20site/>

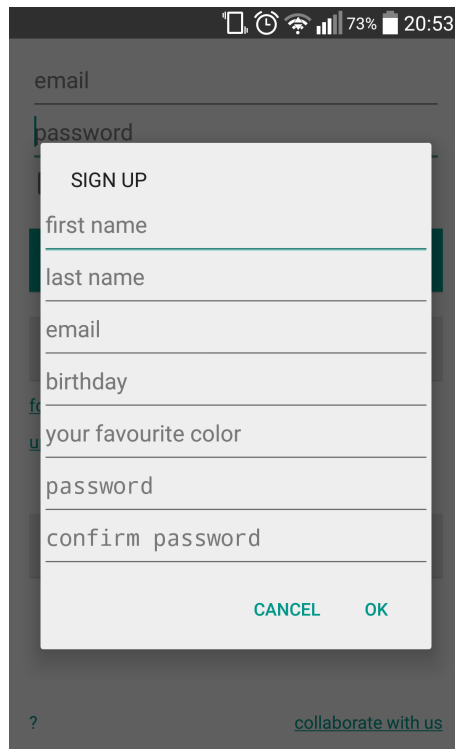


Figura 4.6: Registrazione nuovi utenti

Un utente può anche accedere al gioco senza registrarsi sul nostro sistema, cliccando su “PLAY WITHOUT REGISTRATION”, in questo caso però non godrà degli stessi servizi di un utente loggato.

Avvio del gioco: dopo aver effettuato il log in o aver avviato l'applicazione con l'opzione senza registrazione, inserire lo smartphone nel visore 3D. Inserire nel sistema a quale occhio si desidera somministrare il trattamento (Figura 4.3), e premere conferma. A questo punto inizia il gioco vero e proprio.

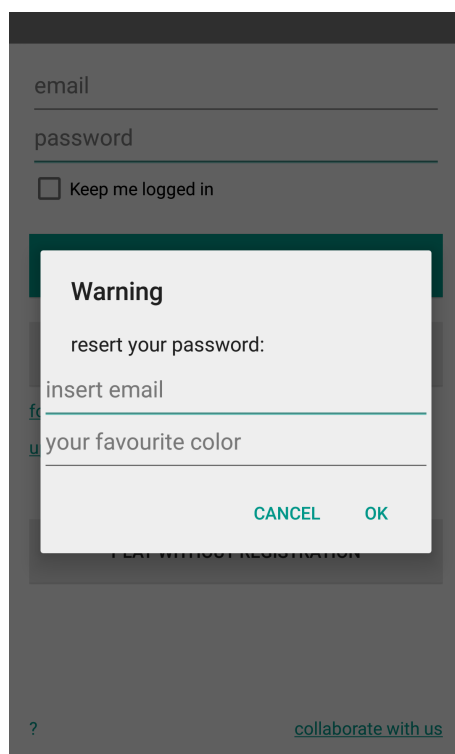


Figura 4.7: Pop up di reset della password

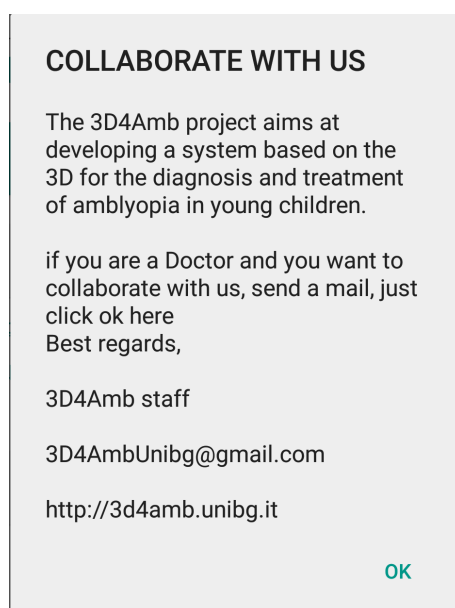


Figura 4.8: Pop up collaborate with us

Insert here your patient data:

firstname

lastname

birthday

START

Figura 4.9: Schermata per i dottori

Capitolo 5

Il software

5.1 Architettura di sistema

Il sistema prevede le seguenti componenti:

- Google Cardboard (vedi sezione 3.2)
- Smartphone Android con connessione internet attiva
- Auricolari con tasti di controllo del volume
- Server con database MySql

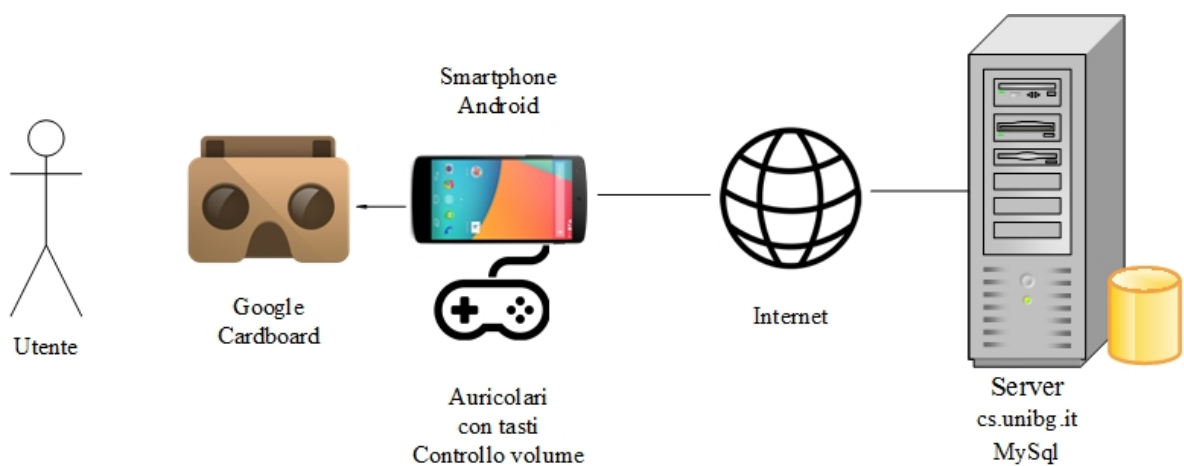


Figura 5.1: Architettura di sistema

Lo smartphone (con S.O. Android) va inserito nel visore, ad esempio Google Cardboard, ma qualora si disponesse di un qualsiasi visore 3D per smartphone il risultato sarà lo stesso, al dispositivo vanno collegati degli auricolari con il controllo del volume (come quelli che si trovano nelle confezioni dei telefoni all'acquisto), saranno necessari per controllare il movimento della macchinina.

La scelta degli auricolari come controller è nata dopo aver notato come l'utilizzo dell'accelerometro (sensore presente su ogni smartphone) come strumento per catturare i comandi dell'utente, causasse un senso di nausea dopo pochi minuti di gioco, si è quindi deciso di utilizzare un controllore esterno.

Dopo di che si è dovuto scegliere il tipo di controllore, dato che i movimenti da controllare erano solo due (sposta a destra e a sinistra l'auto) per essere coerenti con la scelta di un sistema **low cost**, la scelta non è ricaduta su un joy-pad, ma piuttosto sugli auricolari dotati del controllo volume, presenti nella confezione degli smartphone o acquistabili a pochi euro.

Il server è utilizzato per salvare lo storico dei risultati di tutti gli utenti che ne hanno espresso il desiderio, in questo modo è possibile tenere traccia dei loro miglioramenti; è quindi necessario che lo smartphone sia connesso ad internet per poter trasmettere i risultati. Il server è il cs.unibg.it e utilizza un database MySQL è installato presso il dipartimento di ingegneria dell'Università degli Studi di Bergamo¹ a Dalmine(Bg).

¹<http://www.unibg.it/>

5.2 Casi d'uso

Sign Up:

Attori: Utente

Precondizioni: E' necessaria una connessione ad internet.

Passi principali:

1. L'utente lancia l'applicazione.
2. L'utente clicca sul pulsante di Sign Up.
3. L'utente inserisce i propri dati (first name, last name, email, birthday, favourite color, password, password di conferma) e clicca Ok.

Situazioni eccezionali:

- 1a. L'utente lascia vuoti alcuni campi.
- 1b. L'applicazione risponde con un toggle che segnala che alcuni campi sono da compilare e non permette il Sign Up.
- 2a. L' email inserita dall' utente non esiste.
- 2b. L'applicazione risponde tramite una warning, segnalando all'utente l'insistenza dell'email inserita, non permettendo la registrazione.
- 3a. L' email inserita dall' utente è già presente nel database.
- 3b. L'applicazione risponde tramite una warning, segnalando all'utente che la mail non è valida, non permettendo la registrazione.
- 4a. La data di nascita inserita non è valida.
- 4b. L'applicazione risponde tramite una warning, non permettendo la registrazione.

5a. La password di conferma non corrisponde alla password inserita.

5b. L'applicazione risponde tramite una warning, segnalando l'errore e non permettendo il Sign Up.

Postcondizioni: I dati inseriti dall'utente sono validi e vengono registrati sul database MySQL. L'utente è in grado sfruttare tutti i servizi offerti dall'applicazione accedendo tramite la fase di Log In.

Log In utente comune:

Attori: Utente

Precondizioni: E' necessaria una connessione ad internet.

Passi principali:

1. L'utente lancia l'applicazione.
2. L'utente inserisce i campi email e password utilizzati nella fase di Sign Up.
3. L'utente clicca sul pulsante di Log In.

Situazioni eccezionali:

- 1a. L'utente lascia vuoto il campo email o il campo password.
- 1b. L'applicazione risponde con un toggle, il quale segnala che alcuni campi sono da compilare e non permette il Log In.
- 2a. L'utente inserisce un email non presente nel database.
- 2b. L'applicazione risponde tramite una warning, segnalando all'utente la non presenza dell' email nel database MySQL.
- 3a. L'utente inserisce un email presente nel database ma con password scorretta.
- 3b. L'applicazione risponde tramite una warning, segnalando all'utente che la password inserita non è corretta, non permettendo il Log In.

Postcondizioni: Le credenziali di accesso inserite dall'utente sono valide ed il Log In viene eseguito.

Log In medico:

Attori: Medico

Precondizioni: E' necessaria una connessione ad internet. E' necessario che il medico abbia sottomano i dati del paziente.

Passi principali:

1. Il medico lancia l'applicazione.
2. Il medico inserisce i campi email e password utilizzati nella fase di Sign Up.
3. Il medico inserisce i dati del paziente nella form successiva alla fase di Log In.

Situazioni eccezionali:

- 1a. Il medico lascia vuoti alcuni dati del paziente
- 1b. L'applicazione risponde con un toggle, il quale segnala che alcuni campi sono da compilare e non permette l'avvio del gioco.
- 2a. L'utente inserisce una data di nascita non valida per il paziente.
- 2b. L'applicazione risponde tramite una warning, segnalando che la data non è valida e non permettendo l'avvio del gioco.

Postcondizioni: Le credenziali del paziente inserite dal medico sono corrette ed il gioco viene avviato.

Update Info:

Attori: Utente

Precondizioni: E' necessaria una connessione ad internet.

Passi principali:

1. L'utente avvia l' applicazione.
2. L'utente clicca sul link Update Your Info.
3. L'utente sceglie la credenziale di cui vuole fare l' update.
4. L'utente effettua il Log In utilizzando le credenziali precenti all'update.
5. L'utente inserisce il nuovo valore della credenziale da modificare e clicca Ok.

Situazioni eccezionali:

- 1a. Le credenziali di Log In non sono valide
- 1b. L'applicazione risponde con un warning e rimanda alla schermata principale.
- 2a. La nuova credenziale inserita non è valida.
- 2b. L'applicazione risponde tramite una warning, rimandando alla schermata principale.

Postcondizioni: L'update delle credenziali selezionate viene eseguito correttamente.

5.3 Package Diagram

Ecco una panoramica sui package della nostra app, adottiamo una strategia top-down, e partendo da un livello alto, aumentiamo via via il grado di dettaglio.

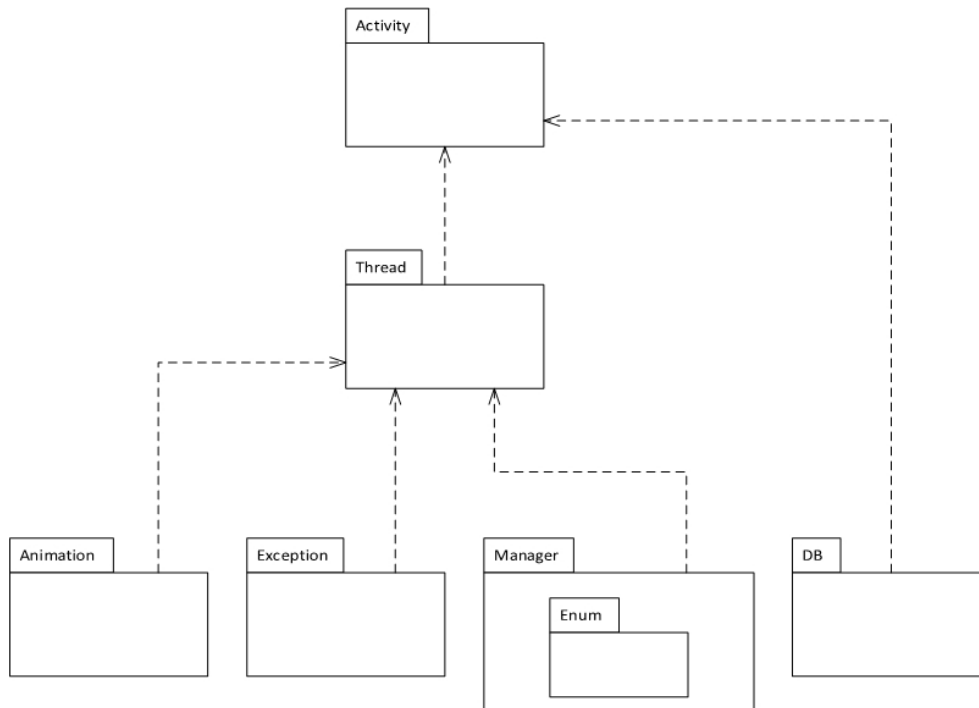


Figura 5.2: Package diagram

5.4 Component diagram

5.4.1 Il sistema di Car Racing Cardboard

Le due componenti principali del sistema Car Racing Cardboard sono il server cs.unibg.it e l'applicazione Andoid, queste due componenti comunicano con chiamate di tipo http post.

A differenza di GET, che permette a un server di mandare informazioni a un client, l'operazione POST fornisce ai client un modo per trasmettere informazioni ai server. La maggior parte dei browser web usa comunemente tale metodo per inviare forms ai server Web. Il client manda un messaggio POST e include l'informazione che desidera mandare al server. Il server può rimandare l'informazione stessa come parte della risposta [3].

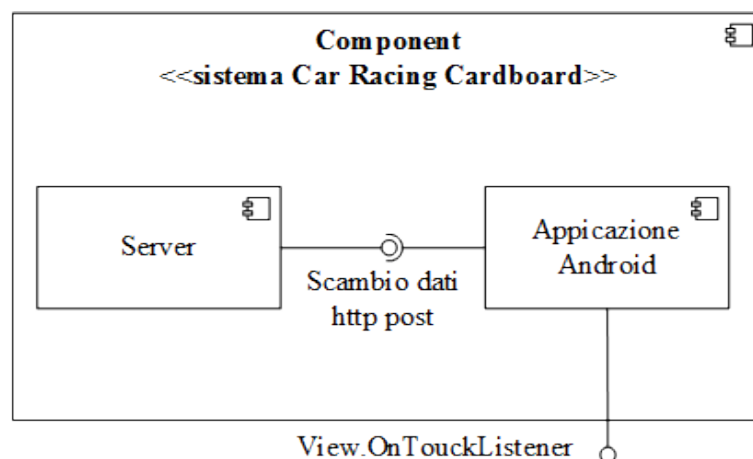


Figura 5.3: Component diagram sistema

5.4.2 L'applicazione Android

Entranto nel dettaglio della prima componete, l'applicazione Android, offre il servizio di interfaccia grafica touch, mentre riceve dal server il servizio di scambio dati http. I componenti grafici di Android si dividono in due categorie: View e ViewGroup. Una View è un qualunque componente utilizzabile (TextView, EditView, CheckBox, Button, ImageView, ?) in una interfaccia. Una ViewGroup è solita-

mente un layout che contiene un insieme di View, i più usati sono LinearLayout, RelativeLayout e TableLayout. Un ViewGroup è a sua volta una View, per questo possono essere creati layout complessi annidando a più livelli i vari componenti. View e ViewGroup sono un esempio di pattern composite. Le interfacce grafiche in Android possono essere definite in due modi:

- definendo un file xml: ad ogni View corrisponde un tag xml, i ViewGroup sono tag xml che contengono all'interno altri tag;
- scrivendo codice Java: ogni View ha una classe associata, è possibile aggiungere una View a un ViewGroup usando il metodo `addView`.

L'applicazione Car Racing Cardboard sfrutta entrambe le forme di componenti grafiche.

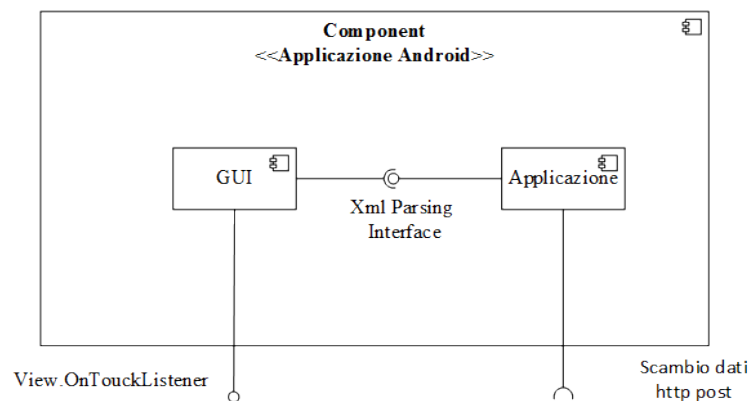


Figura 5.4: Component diagram applicazione Android

5.4.3 Il server

5.4.4 Dettaglio applicazione

5.5 Class diagram

5.6 Sequence diagram

Capitolo 6

Analisi degli Algoritmi

6.1 Penalizzazione progressiva delle immagini

L'alpha value di un immagine è un valore che va da 0 a 255 e descrive la sua trasparenza. Un'immagine con alpha=0 è trasparente, mentre con alpha=255 conserva tutte le sue caratteristiche, con questo codice all'aumentare del livello, aumenta anche la trasparenza dell'immagine, dopo l'ottavo livello l'alpha rimane=50, continua però ad aumentare la difficoltà del gioco, in termini di nemici e velocità d'azione.

```
1  /**
2  *
3  * @return Alpha value to penalize the image
4  */
5  private int getLevelPenalization(){
6      switch(globalData.getLevel()){
7
8          case 1:{
9              return 200;
10             }
11         case 2:{
12             return 180;
13         }
```

```
14     case 3:{
15         return 160;
16     }
17     case 4:{
18         return 140;
19     }
20     case 5:{
21         return 120;
22     }
23     case 6:{
24         return 100;
25     }
26     case 7:{
27         return 80;
28     }
29     case 8:{
30         return 50;
31     }
32     default:{
33         return 50;
34     }
35 }
36 }
```

6.2 Generazione random scena

6.2.1 Nemici

Nella scena ci sono tre “corsie” dalle quali possono arrivare i “nemici”, questo è il codice dal quale viene estratta la corsia scelta. Inizialmente la corsia viene scelta a caso con un’ estrazione random, ma per evitare che il giocatore resti troppo a lungo fermo nella stessa corsia senza muoversi viene fatto un controllo sulla scelta precedente. Un nemico non viene mai mandato più di due volte sulla stessa corsia.

```
1  public void randomFunction(){
2      selectedLane=(int)(lane*Math.random())+1;
3      if(previousLane==selectedLane){
4          countPrevious++;
5          if(countPrevious>3){
6              selectedLane++;
7              if(selectedLane>3){
8                  selectedLane=1;
9              }
10         }
11     }
12     previousLane=selectedLane;
13     selectedCar=(int)(numberOfCar*Math.random())+1;
14 }
```

6.2.2 Panorama

Analogamente ai nemici, anche il panorama viene modificato per rendere più avvincente l'ambientazione del gioco.

```
1 public void randomPanorama() {  
2     int pick = new Random().nextInt(Side.values().length);  
3     selectedSide= Side.values()[pick];  
4     idSubject= (int)(numberOfSubject*Math.random())+1;  
5 }
```

6.3 Cifratura password

Le password degli utenti non vengono memorizzate nel Data Base in chiaro ma vengono cifrate (dallo smartphone) con un algoritmo sha256.

```
1  public static String calculateHash(String data) {  
2      String hash = new String(Hex.encodeHex(DigestUtils.  
    sha256(data)));  
3      return hash;  
4  }
```

6.4 Servizio web

La comunicazione tra **client** (smarthphone) e **server** (cs.unibg.it) avviene tramite chiamate post (vedi anche 5.4.1), le chiamate sono sincrone, ossia ad un messaggio inviato dal client corrispondono sempre delle risposte dal servizio. Le chiamate partono sempre dal client, il server di suo non invia messaggi.

Il thread della chiamata post la chiamata avviene nel metodo *public void myPostCall(final TypeCall type, final Activity logInActivity)* grazie ad un thread al suo interno.

```
1 // Create a new HttpClient and Post Header
2 thread = new Thread(new Runnable() {
3     @Override
4     public void run() {
5         HttpClient httpclient = new DefaultHttpClient();
6         HttpPost httppost = new HttpPost("http://3d4amb.unibg
7         .it/3dcar/3d4ambService.php");
8         [...]
9     }
}
```

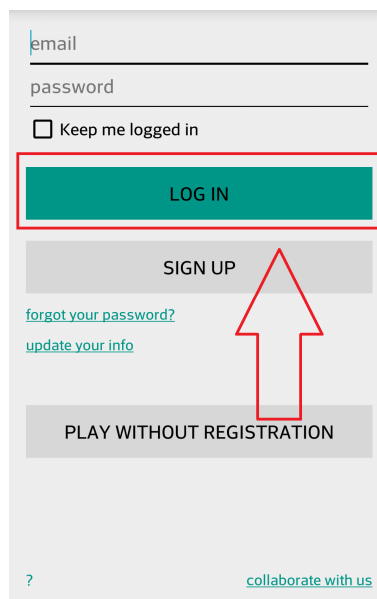
La chiamata viene fatta al servizio php di Car Racing Cardboard (3d4ambService.php).

Le chiamate possibili sono varie, per ognuna è stato predisposto una lista di stringhe che possono essere passate all'oggetto httpPost:

```
1 List<NameValuePair> nameValuePairsUpdateString = new
    ArrayList<NameValuePair>(3);
2 List<NameValuePair> nameValuePairsLogIn = new ArrayList<
    NameValuePair>(3);
3 List<NameValuePair> nameValuePairsReset = new ArrayList<
    NameValuePair>(5);
4 List<NameValuePair> nameValuePairsSignUp = new ArrayList<
    NameValuePair>(7);
5 List<NameValuePair> nameValuePairsDoctorCall = new
    ArrayList<NameValuePair>(5);
6 List<NameValuePair> nameValuePairsReport = new ArrayList<
    NameValuePair>(4);
```

Log in:

La chiamata post per il login avviene ogni volta che l'utente preme il bottone di login (Figura 6.1):



The image shows a login form with the following elements:

- Input field for "email"
- Input field for "password"
- Checkbox labeled "Keep me logged in"
- A teal "LOG IN" button, which is highlighted with a red rectangular box.
- A grey "SIGN UP" button.
- Two links: "forgot your password?" and "update your info".
- A grey "PLAY WITHOUT REGISTRATION" button.
- A red arrow pointing upwards towards the "LOG IN" button.
- At the bottom, there is a question mark icon and a link "collaborate with us".

Figura 6.1

Per il login vengono inviate al server email e password, che prima di essere inviata viene cifrata (vedi 6.3).

A questo punto viene lanciata la finestra di attesa (Figura 6.2):

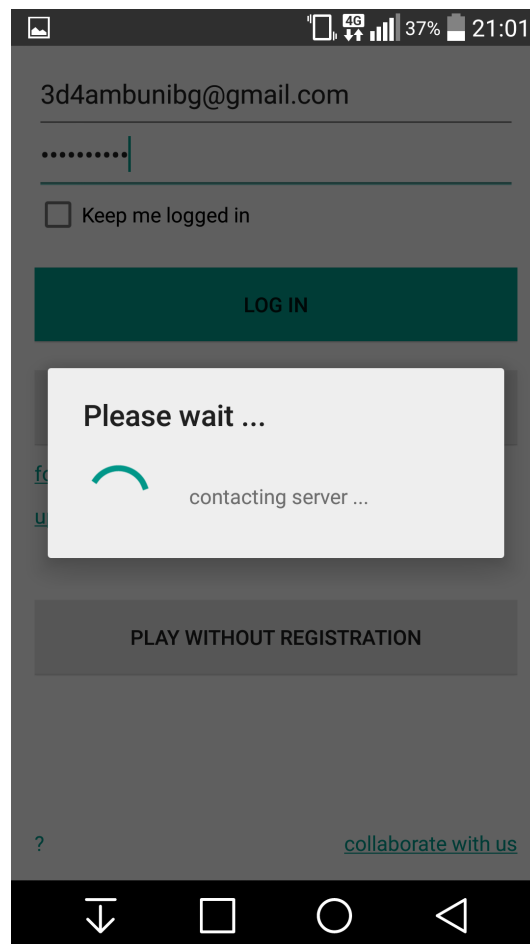


Figura 6.2: interfaccia di attesa

Se il servizio risponde in maniera positiva viene lanciata l'activity successiva (SplashActivity o TutorialActivity a seconda delle impostazioni utente).

Reset

//TODO

Doctor call

//TODO

Report

//TODO

Sign up

//TODO

6.5 Thread

6.6 AsyncTask

Per ottimizzare l'utilizzo di memoria del device da parte dell'applicazione sono stati utilizzate i task asincroni (AsyncTask) per la gestione di tutta la parte grafica dinamica.

In molti framework di sviluppo con interfacce grafiche i task lunghi non devono essere eseguiti nel thread principale dell'applicazione. Infatti il thread principale è quello che si occupa di gestire gli eventi che arrivano dall'interfaccia (per esempio il click di un pulsante), se questo thread è già occupato a eseguire qualcos'altro l'effetto che si ottiene è quello di un ritardo di esecuzione (lag)[1].

I quattro passi: un AsyncTask in android si divide in quattro "step"

1. `onPreExecute()`, invocato sul thread della UI¹ prima che il task venga eseguito. Qui viene solitamente messo il setup del task
2. `doInBackground(Params...)`, invocato nel background del thread, subito dopo `onPreExecute()`. Questo step è usato per eseguire computazioni in background che possono durare molto tempo. Il risultato della computazione deve essere tornato in questo step, e possono essere pubblicati grazie a `publishProgress(Progress...)`. I valori sono passati all' UI thread nello step `onProgressUpdate(Progress...)`.
3. `onProgressUpdate(Progress...)`, invocato nell' UI thread dopo aver chiamato `publishProgress(Progress...)`. Questo metodo è usato per mostrare ogni progresso nella UI mentre in background la computazione è ancora in esecuzione
4. `onPostExecute(Result)`, invocato dopo che la computazione in background è finita.[4].

¹User Interface

Capitolo 7

Toll utilizzati

7.1 GitHub

7.2 Crashlitics

Capitolo 8

Social media

La promozione dell'applicazione è stata effettuata con i mezzi che ad oggi sono più in voga, gli strumenti utilizzati sono stati Facebook¹ e Google+², è stato anche realizzato un sito internet³.

¹<https://www.facebook.com/3D4Amb>

²<https://plus.google.com/u/1/110383688193490676162/posts>

³<http://3d4amb.unibg.it/3dcar/cardboard%20site/>

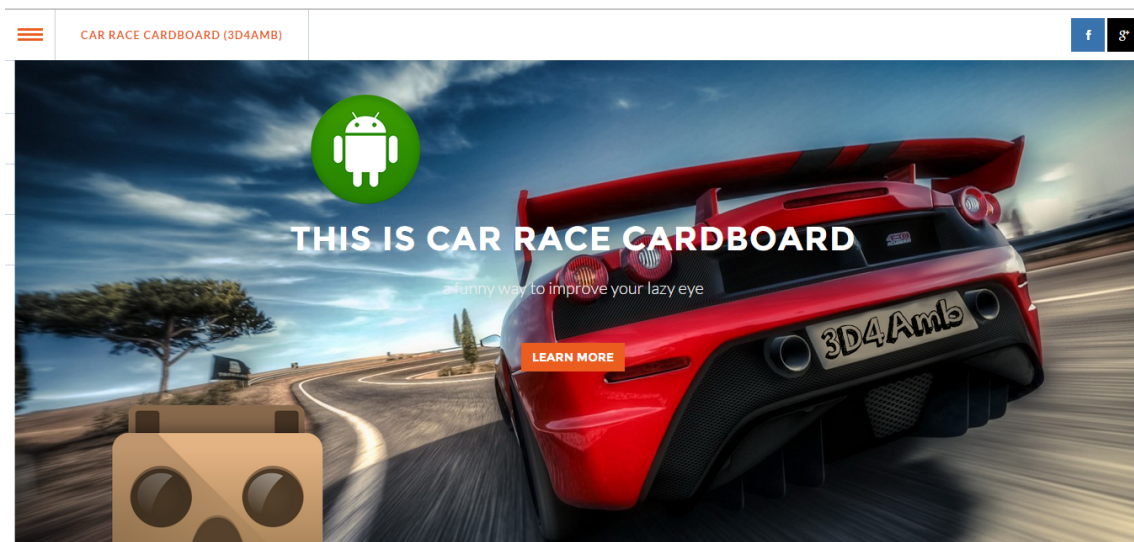


Figura 8.1: Car Racing Cardboard web site

Capitolo 9

Conclusioni

Lo scopo di questa applicazione è il trattamento dell'ambliopia, non è necessario avere a disposizione device costoso, al giorno d'oggi tutte le persone hanno uno smartphone e Goggles Cardboard è molto economico. Usando l'applicazione Car Racing Cardboard, le persone che soffrono di ambliopia possono migliorare il loro occhio pigro.

La tipologia di trattamento proposta da 3D4Amb, cerca di evitare i rischi della terapia occlusiva, ma può anche essere usata in parallelo con la terapia occlusiva[2].

Certificazione contenuti: L'applicazione ha ricevuto delle certificazioni da Google Play Store:

- Australian Classification Board (ACB) – Australia
- Classificação Indicativa (ClassInd) – Brazil
- Entertainment Software Rating Board (ESRB) – North America
- Pan-European Game Information (PEGI) – Europe



Figura 9.1: Classificazione contenuti

-
- Unterhaltungssoftware Selbstkontrolle (USK) – Germany
 - IARC Generic – Rest of the world
 - Google Play –South Korea

9.1 Paper

Bibliografia

- [1] F. Collini. <http://android.devapp.it/>, Luglio 2011.
- [2] A. Gargantini, F. Terzi, M. Zambelli, and S. Bonfanti. A low-cost virtual reality game for amblyopia rehabilitation. 2015.
- [3] S. D. Modica. appunti a cura di salvatore di modica. *Università di Pisa*.
- [4] A. official site. <http://developer.android.com/>.
- [5] N. Statt. Facebook has oculus, google has cardboard. 2014.
- [6] F. Terzi and M. Zambelli. Progetto e sviluppo di un videogame basato sulla penalizzazione per il trattamento dell'ambliopia. 2013.