

Ringraziamo il Professor Angelo Gargantini per i preziosi insegnamenti, Silvia Bonfanti per il supporto nella stesura di questo elaborato, inoltre vogliamo ringraziare le nostre famiglie per il sostegno ed il grande aiuto.

Fabio e Matteo

Indice

I Descrizione del contesto	7
1 Introduzione	9
1.1 Il progetto 3D4Amb	9
1.2 Obiettivo	10
2 L'Ambliopia	11
2.1 Il disturbo	11
2.2 Trattamenti	13
2.2.1 Occlusione	13
2.2.2 Penalizzazione ottica	13
2.2.3 Penalizzazione farmacologica	13
2.2.4 Settorizzazione	13
2.3 L'idea di 3D4Amb	15
3 La tecnologia per il trattamento	17
3.1 Sistema operativo ed ambiente di sviluppo	17
3.1.1 Perchè Android	18
3.1.2 Perchè Android Studio	19
3.2 Google cardboard	20
II Panoramica sull'applicazione	23
4 Car Racing Cardboard: l'applicazione	25
4.1 Il principio del trattamento tramite il gioco	25

4.2	Il gioco	27
4.2.1	Obiettivo	27
4.2.2	Doppia Viewport	28
4.3	Guida all'utilizzo	29
III	Implementazione	35
5	Panoramica su software e hardware	37
5.1	Architettura di sistema	37
5.2	Vista di implementazione software	39
5.3	Scenari e casi d'uso	40
5.3.1	Sign Up	40
5.3.2	Log In utente comune	42
5.3.3	Log In medico	44
5.3.4	Update Info	46
6	Software design	49
6.1	Package Diagram	49
6.2	Component diagram	50
6.2.1	Il sistema di Car Racing Cardboard	50
6.2.2	L'applicazione Android	50
6.2.3	Dettaglio applicazione	51
6.2.4	Il server	52
6.2.5	Database	53
6.3	Sequence diagram	54
6.3.1	Funzionamento con log in	54
6.3.2	Funzionamento senza log in	56
6.3.3	Funzionamento sign up	56
7	Analisi degli Algoritmi	57
7.1	Penalizzazione progressiva delle immagini	57
7.2	Generazione casuale della scena	59

7.2.1	Nemici	59
7.2.2	Panorama	60
7.3	Cifratura password	61
7.4	Servizio web	62
7.5	Gestione dello scontro con il nemico	83
7.6	Game Thread	88
7.7	Gestione animazioni	93
7.8	Asynctask	95
8	Tool utilizzati	97
8.1	GitHub	97
8.2	Crashlytics	99
IV	Valutazioni finali	103
9	Social media	105
10	Conclusioni	107
10.1	Paper	109
10.2	Sviluppi futuri	114

Parte I

Descrizione del contesto

Capitolo 1

Introduzione

3D4Amb (1.1) è un progetto che punta a sviluppare un sistema basato sulla tecnologia 3D per la diagnosi e il trattamento dell’ambliopia. Nell’ambito di questo progetto è stata sviluppata un’applicazione Android il cui scopo è quello di effettuare il trattamento della malattia.

1.1 Il progetto 3D4Amb

Il progetto 3D4Amb 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.

1.2 Obiettivo

L’obiettivo di fondo era quello di un **upgrade** della tesi di laurea di primo livello, dove era stato fatto un restyling di un videogioco desktop, chiamato Space Invaders¹.

Nel 2013 provare il tipo di trattamento proposta da 3D4Amb avrebbe implicato l’acquisto di una scheda grafica 3D e di un monitor 3D.

Oggi con Car Racing Cardboard, è sufficiente acquistare un visore 3D come Google Cardboard (reperibile a circa 10\$ online), praticamente tutti possiedono uno smartphone, quindi la terapia “mnobile” proposta da 3D4Amb è in grado di raccogliere un bacino d’utenza molto più ampio.

Il fulcro di questa tesi è l’applicazione Car Racing Cardboard, nel corso di questo documento verrà analizzata a partire da un macro livello, focalizzandosi sull’idea che ha portato alla sua realizzazione, fino al livello di dettaglio implementativo degli algoritmi che la compongono.

¹<http://3d4amb.unibg.it/software/spaceinvaders.it.html>

Capitolo 2

L'Ambliopia

In questo secondo capitolo viene spiegato il l'ambito nel quale opera questa tesi (2.1), quali sono i tipi di trattamento al momento disponibili (2.2) e la nuova tipologia di trattamento proposta dal progetto 3D4Amb (2.3).

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 rielaburate 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).[7]

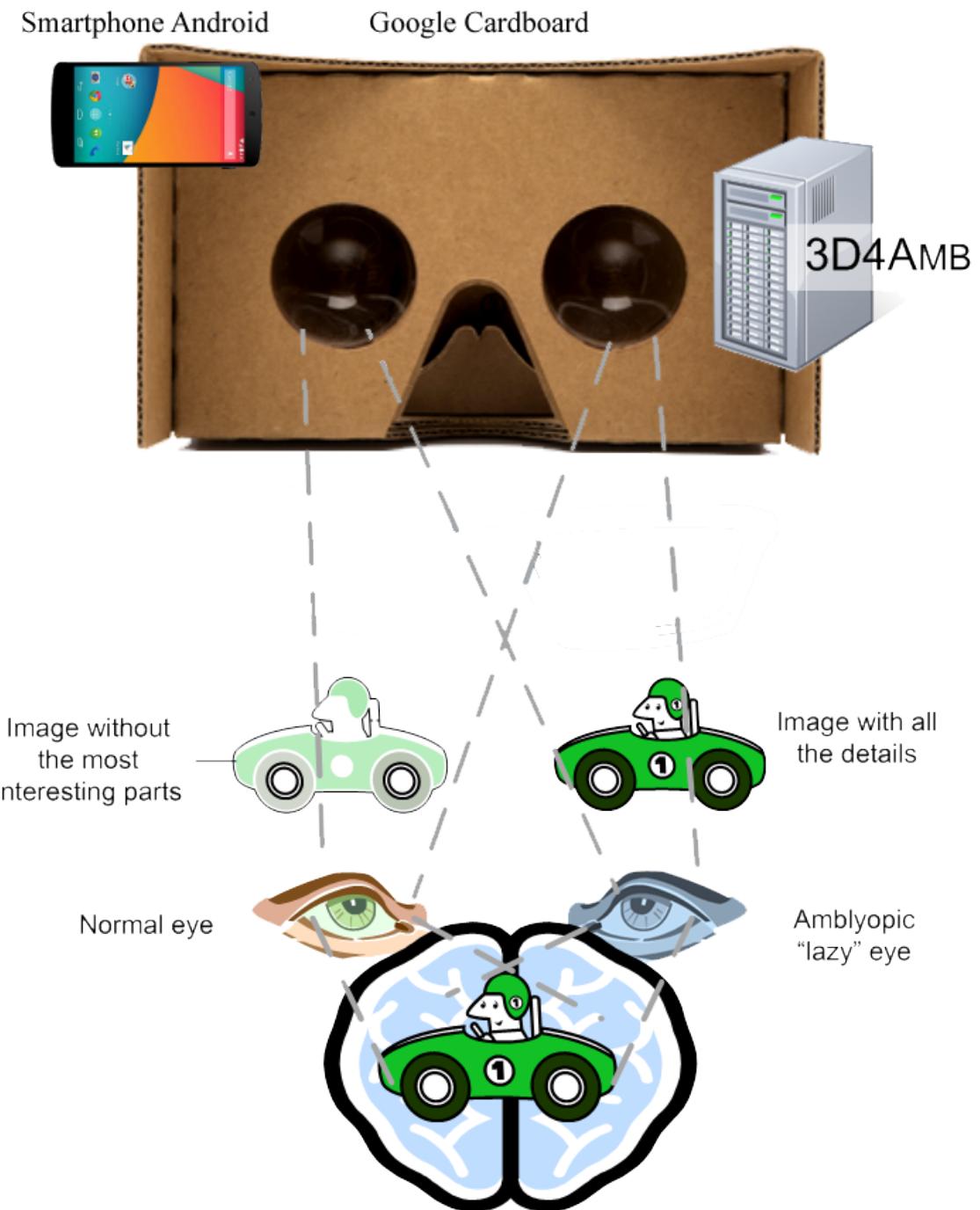


Figura 2.1: La tecnologia di 3D4Amb

Capitolo 3

La tecnologia per il trattamento

In questo terzo capitolo viene spiegata la scelta del sistema operativo dell'applicazione realizzata e l'ambiente di sviluppo utilizzato (3.1), inoltre il capitolo contiene la descrizione al visore 3D “ideale” con il quale interagisce l'applicazione (3.2).

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.

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

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.

3.1.1 Perchè Android

La scelta di sviluppare l’applicazione per il sistema operativo Android, anzichè 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.

Parlando di numeri, Android domina fra gli smartphone con una quota di mercato dell’ 84,7%. E’ il dato principale che emerge dall’ analisi della società di ricerca *Idc*

²(ICE_CREAM SANDWICH_MR1)

sul mercato dei cellulari. Il sistema operativo mobile di Google, a differenza di quello di Apple, è presente su decine di modelli di cellulari di marche diverse e domina il mercato lasciando alla “Mela” l’11,7% di share [2].

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 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).



Figura 3.1: Android Logo

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” [6]



Figura 3.2: Google Cardboard

Introdotto nel 2014 al “Google I/O developers conference for Android device”, è formato 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.

Parte II

Panoramica sull'applicazione

Capitolo 4

Car Racing Cardboard: l'applicazione

Questo capitolo contiene la descrizione del principio di trattamento del gioco (4.1), la spiegazione degli obiettivi del gioco (4.2), e una guida all'utilizzo dell'applicazione (4.3).

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 ampiope 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 gamerplayer 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 gamerplayer 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.

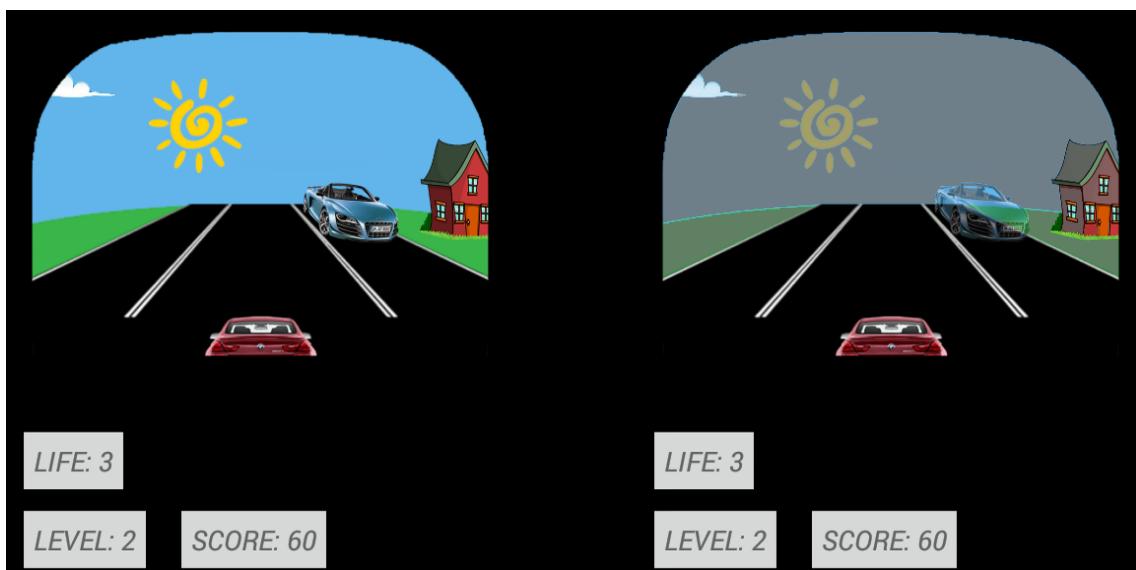


Figura 4.2: una scena del gioco

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 da la possibilità di scegliere l'occhio pigro:



Figura 4.3: scelta dell'occhio pigro

In questo modo si da all'applicazione 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 friure 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):

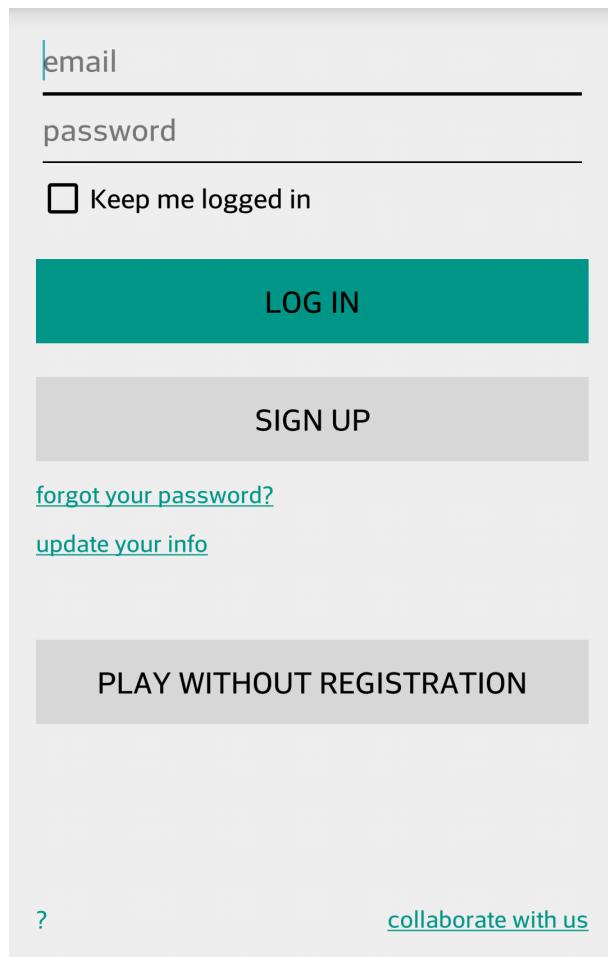


Figura 4.4: Schemata di avvio

Da questa schermata è possibile effettuare il log in, se l'utente è già registrato, 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, poiché non avrebbe senso raccogliere informazioni forvianti (Figura 4.5)

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

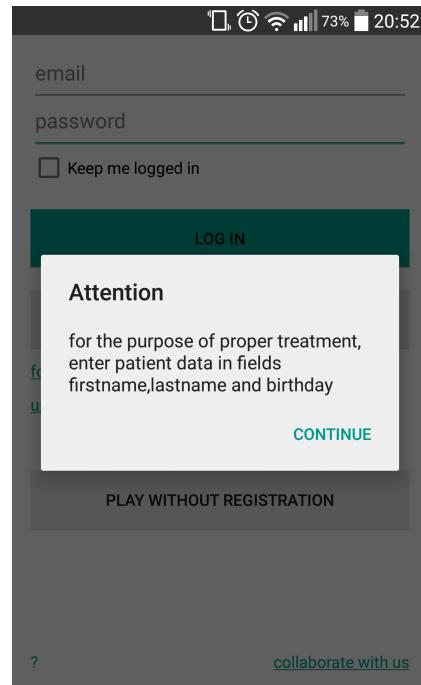


Figura 4.5: Pop up di warning

Cliccando su "CONTINUE" si entra nella vera schermata di registrazione (Figura 4.6):

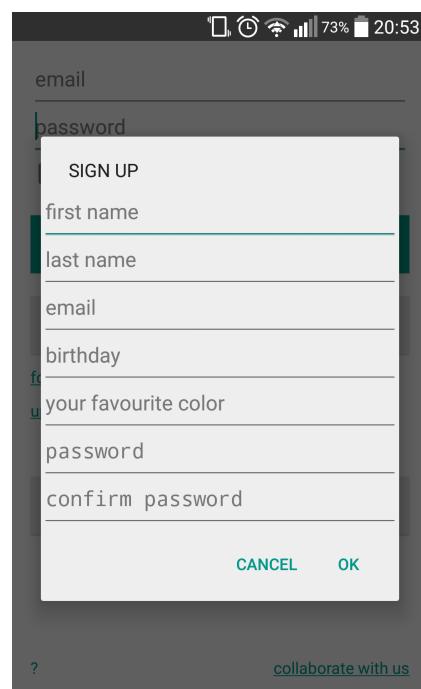


Figura 4.6: Registrazione nuovi utenti

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 nella quale un utente 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).

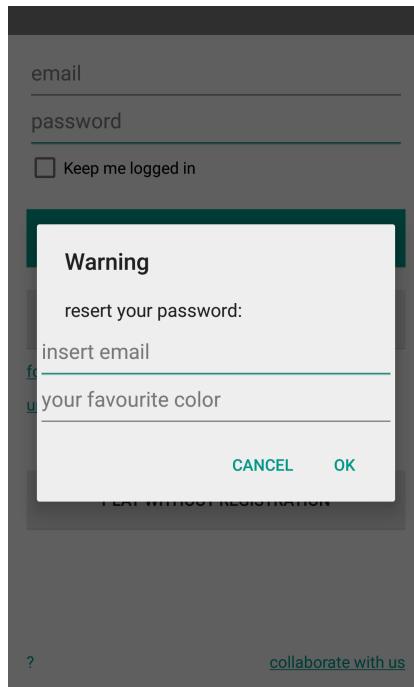


Figura 4.7: Pop up di reset della password

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(vedi anche sezione 9). 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(Figura 4.8).

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

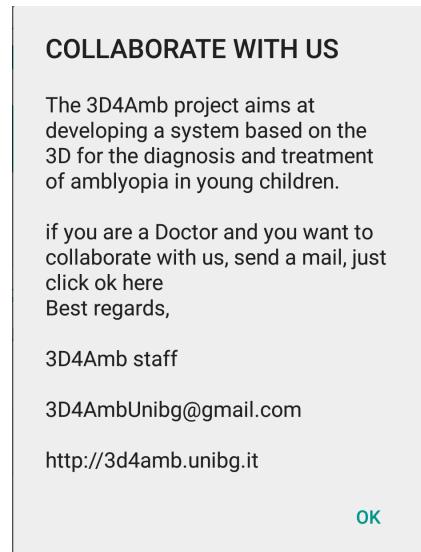


Figura 4.8: Pop up collaborate with us

Il motivo per il quale è stata fatta questa distinzione è il seguente: non sarebbe efficiente, ad esempio, per un dottore che ha tre pazienti, dover registrare tre account diversi, 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).

The image shows a registration form for doctors. At the top, it says "Insert here your patient data:". Below that, there are three input fields: "firstname", "lastname", and "birthday", each with a corresponding horizontal line for input. At the bottom center of the form is a grey "START" button.

Figura 4.9: Schermata per i dottori

In questo caso il dottore inserisce i dati del paziente (nome, cognome e data di

nascita), il sistema in autonomia verifica se si tratta di un primo inserimento oppure di un “log in”. Nel caso di un primo inserimento viene aggiunto un record alla tabella dei pazienti, questo record è collegato a quello del dottore, per consentire poi un eventuale estrazione dei dati di tutti i pazienti di quel dottore.

Giocare senza registrarsi: 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 registrato, questa possibilità è stata pensata per chi volesse provare l’applicazione senza per forza dover fornire i propri dati.

Avvio del gioco: dopo aver effettuato il log in o aver avviato l’applicazione con l’opzione senza registrazione, nel caso di un primo avvio viene visualizzato un video-tutorial per l’utilizzo del gioco, nel quale vengono descritti i comandi per utilizzare lo stesso, è possibile saltare questo video spuntando la checkbox in basso “don’t show me again”.

A questo punto bisogna inserire lo smarthphone nel visore 3D, ed inserire nel sistema l’occhio al quale si desidera somministrare il trattamento (Figura 4.3), e premere conferma.

Il gioco vero e proprio ha inizio ora, l’utente può muovere la sua auto a sinistra premendo il tasto “+” del controller e a destra premendo il tasto “-” (Figura 4.2). La difficoltà aumenta dinamicamente, per rendere via via più stimolante il trattamento. Per ogni sessione di gioco un utente ha tre vite, dopo di che, una volta arrivati al “game over”, l’applicazione chiede se si desidera ricevere via mail lo storico con i risultati (Figura 4.10).

Come accennato in precedenza, tale funzionalità, è accessibile solo dagli utenti registrati, un utente non registrato visualizzerà, una volta terminata la sua partita, la stessa schermata in Figura 4.10, con solamente l’opzione di riavvio del gioco.



Figura 4.10: Schermata finale

Parte III

Implementazione

Capitolo 5

Panoramica su software e hardware

In questo capitolo troviamo l'architettura di sistema con relativo diagramma ed una breve descrizione sulle scelte dell' hardware (5.1), una panoramica sul software sotto forma di vista di implementazione (5.2) infine una serie di scenari e casi d'uso (5.3).

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 Linux con database MySql

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.

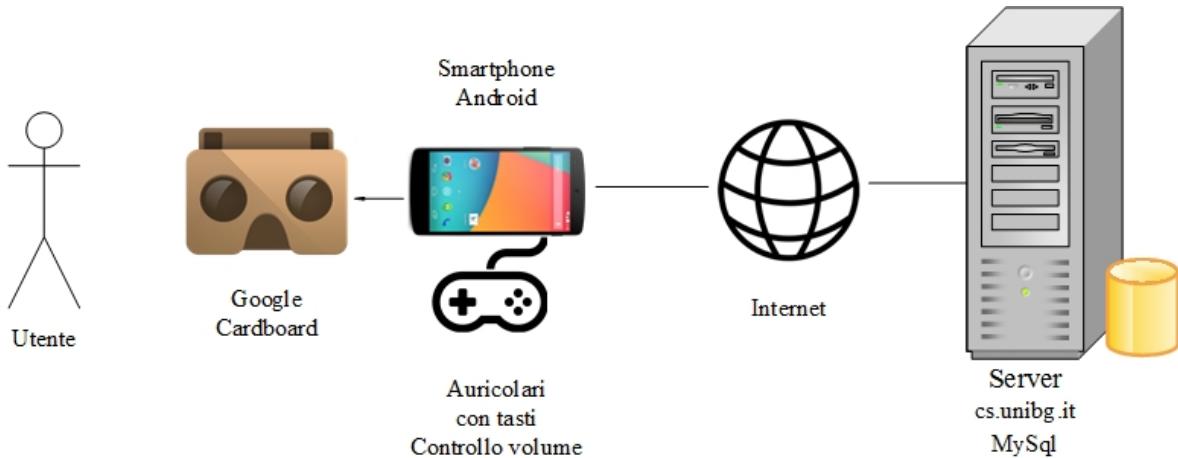


Figura 5.1: Architettura di sistema

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 del 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, è una macchina Linux ed 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 Vista di implementazione software

Questo grafico mostra dove sono installate le varie componenti software, in che linguaggio sono scritte e come interagiscono tra di loro (Figura 5.2).

Tutte le componenti lato client (Android), sono realizzate in java e xml, per interagire con il servizio web, passano attraverso una classe che gestisce le chiamate post, rappresentata nel grafico come Post Call Manager. Questa classe lancia una serie di chiamate post, grazie alle quali è possibile gestire: la verifica dei dati per il log in, l'inserimento di un nuovo utente tramite il processo di sign up, la modifica dei dati dell'utente e della propria password, infine la gestione dei dati di gioco. Tutte queste informazioni vengono memorizzate nel database MySql presente sul server cs.unibg.it. Tramite il servizio web, interamente realizzato in php viene anche gestito l'invio delle email contenenti: il riepilogo dei dati di gioco, le notifiche di conferma registrazione il cambiamento dei dati o la modifica della password.

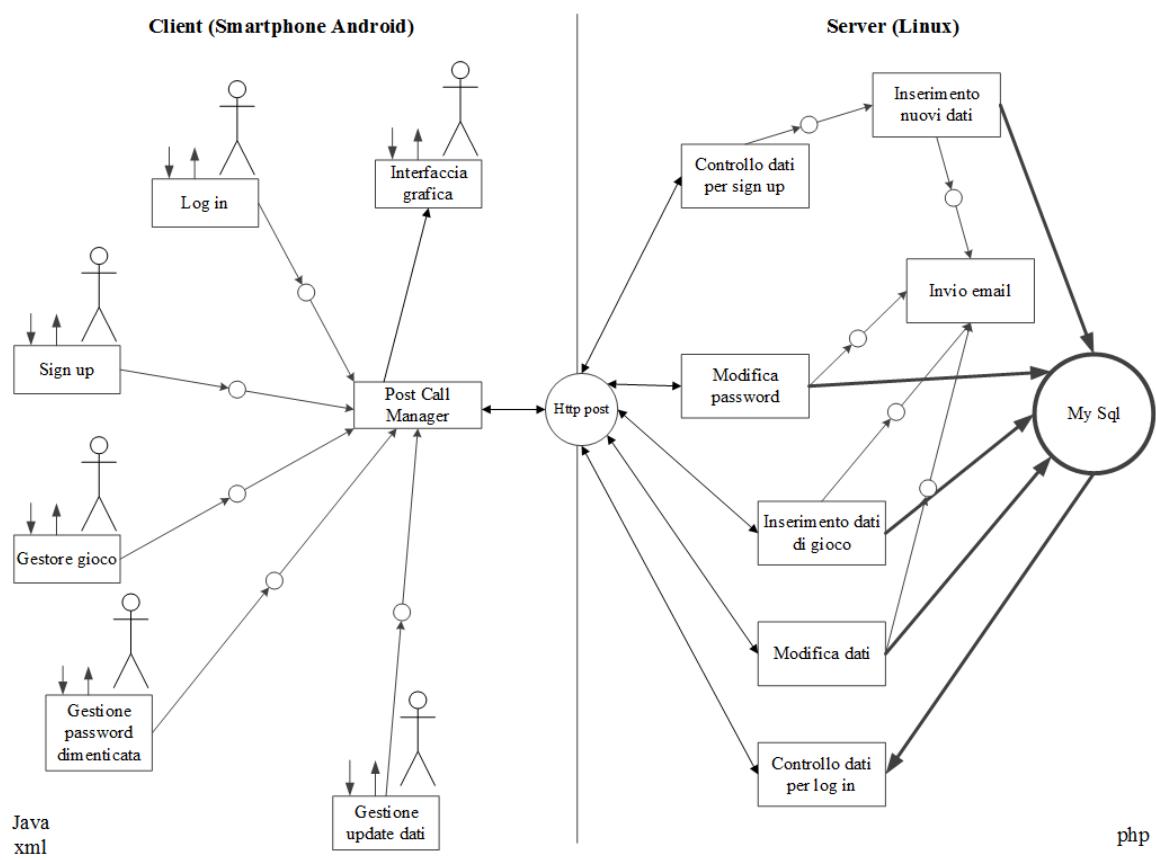


Figura 5.2: Vista d'implementazione del software

5.3 Scenari e casi d'uso

5.3.1 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.

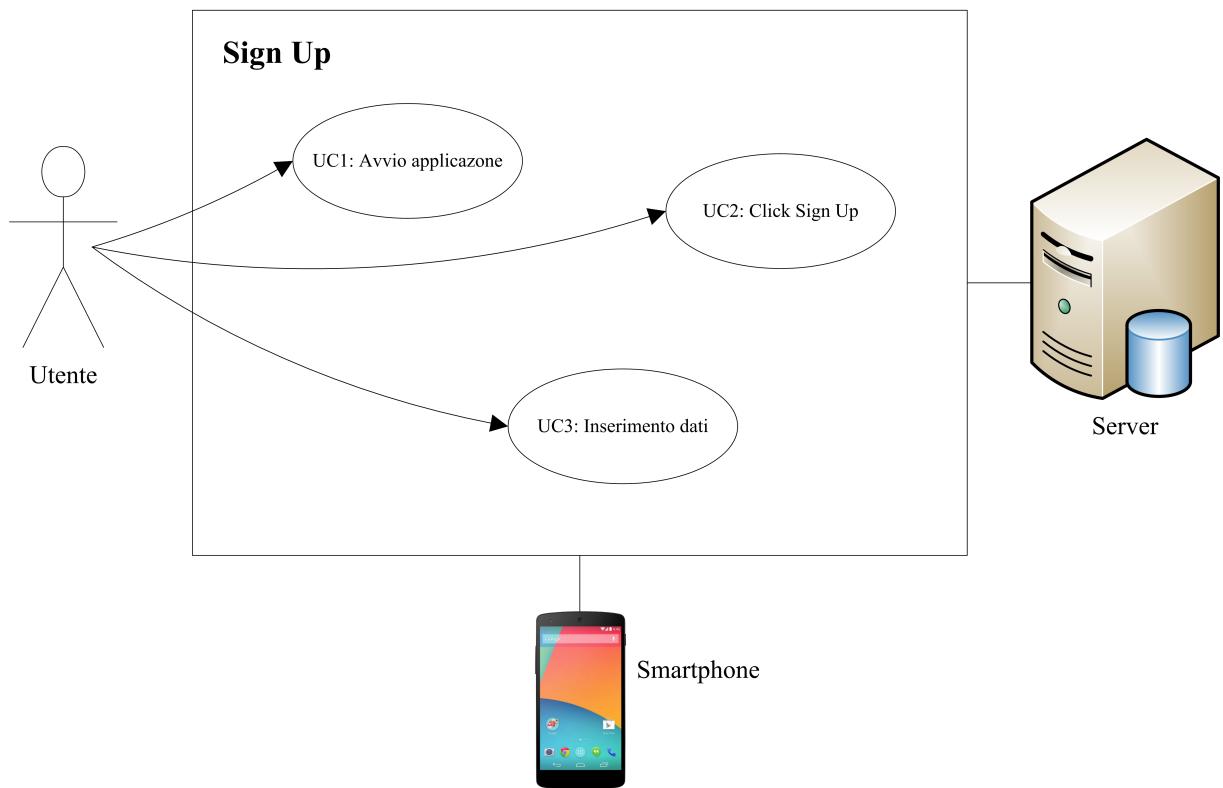


Figura 5.3: Use Case Sign Up

5.3.2 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.

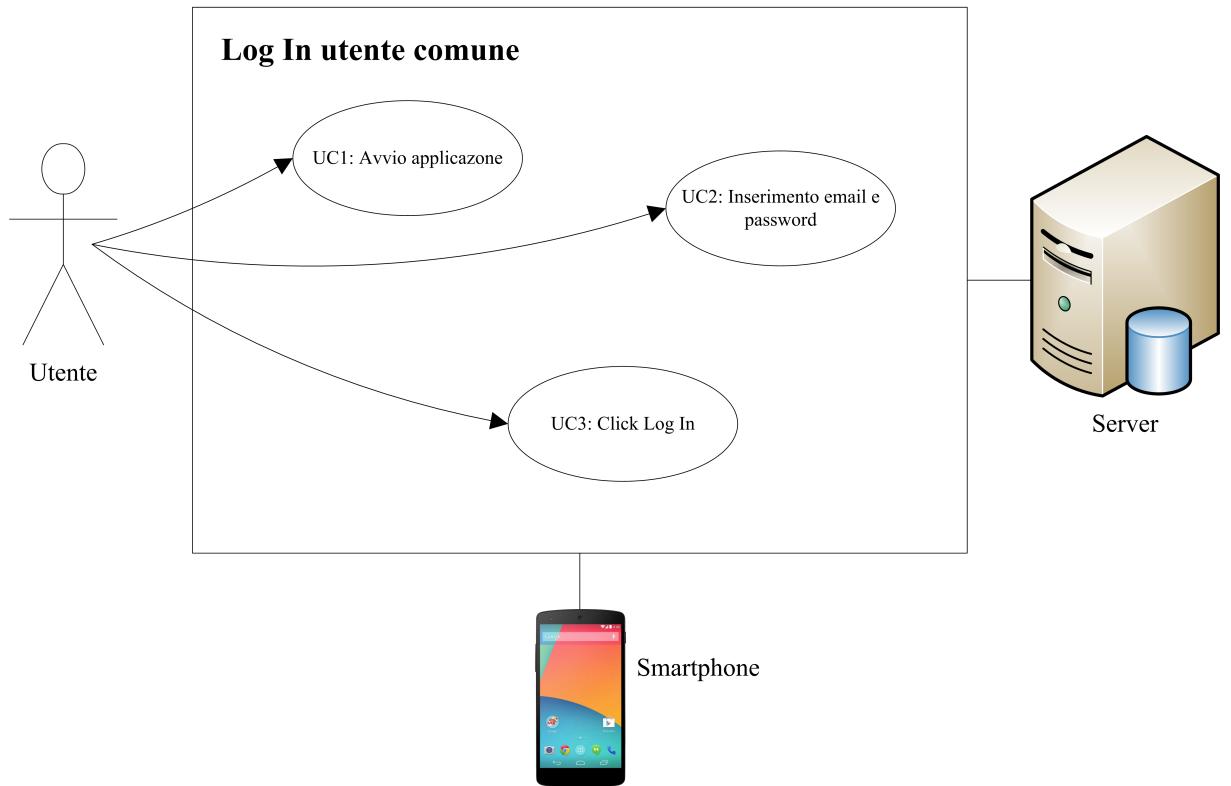


Figura 5.4: Use Case Log In utente comune

5.3.3 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 clicca sul pulsante di Log In.
4. 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.

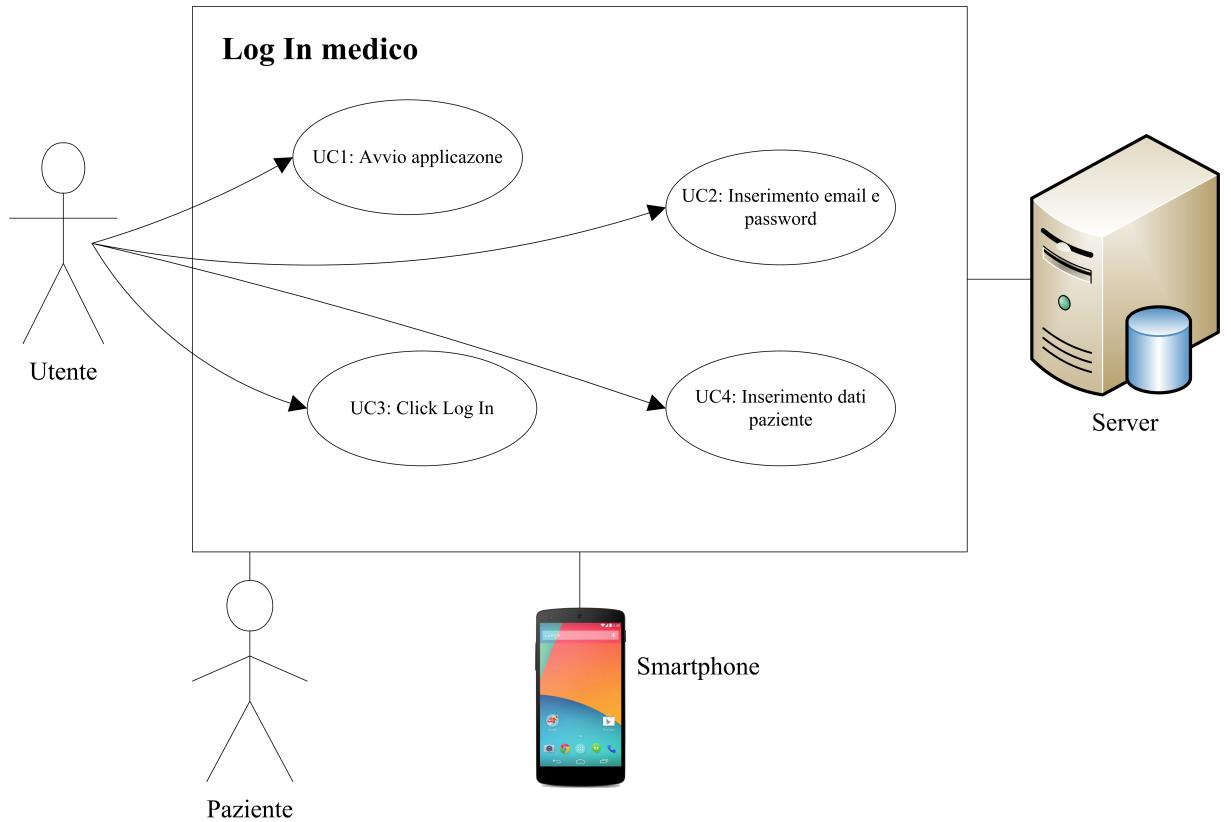


Figura 5.5: Use Case Log In medico

5.3.4 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.

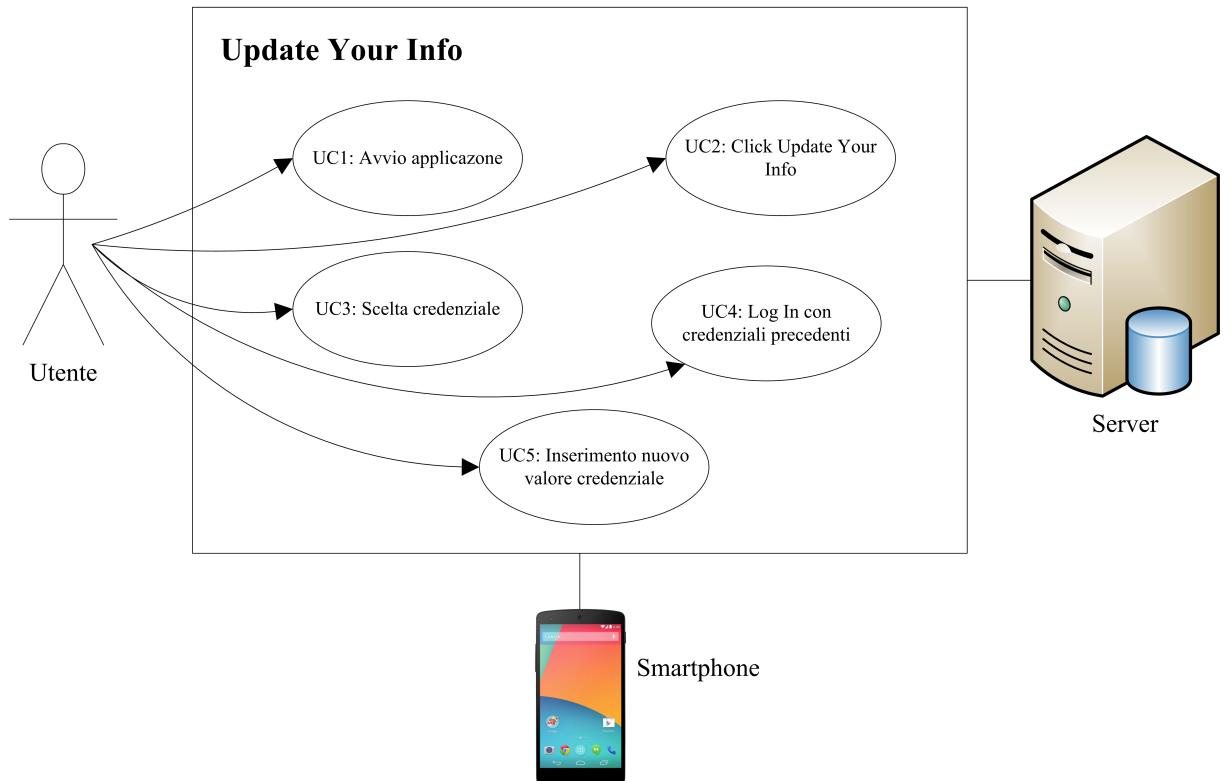


Figura 5.6: Use Case Update Info

Capitolo 6

Software design

In questo capitolo vengono riportati i classici diagrammi UML, i quali permettono tramite l'utilizzo di modelli visuali, di analizzare, descrivere, specificare e documentare un software, in sequenza di trovano package diagram (6.1), component diagram (6.2) e sequence diagram (6.3), i class diagram non sono stati riportati poichè data la mole di codice, sarebbero risultati illeggibili.

6.1 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.

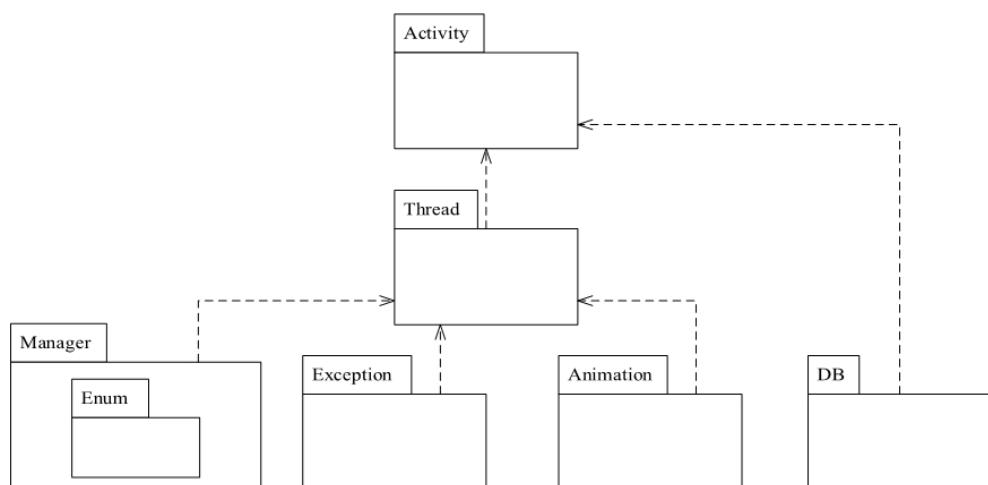


Figura 6.1: Package diagram

6.2 Component diagram

6.2.1 Il sistema di Car Racing Cardboard

Le due componenti principali del sistema Car Racing Cardboard sono il server cs.unibg.it e l'applicazione Android, 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 al 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 [4].

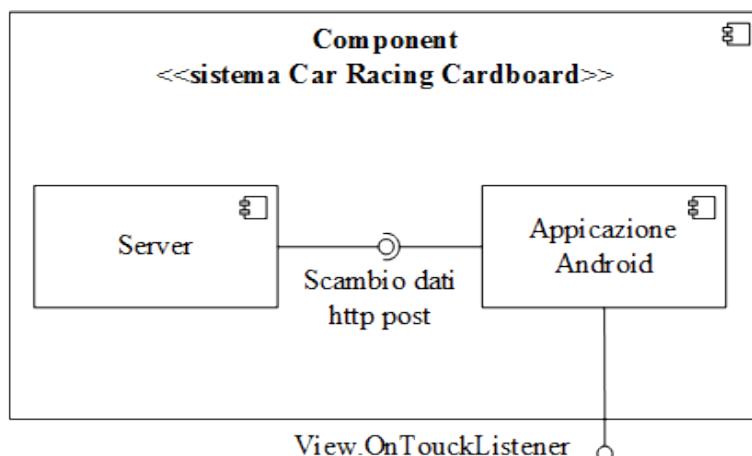


Figura 6.2: Component diagram sistema

6.2.2 L'applicazione Android

Entrando nel dettaglio della prima componente, 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, EditText, 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 (entrambe sfruttate in Car Racing Cardboard):

- 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.

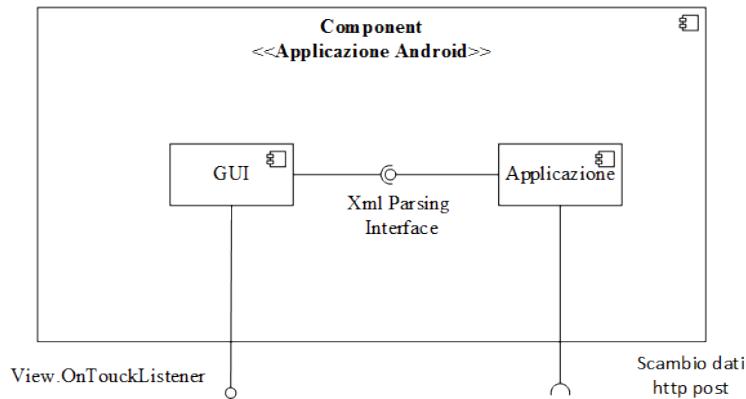


Figura 6.3: Component diagram applicazione Android

6.2.3 Dettaglio applicazione

L'applicazione per quanto riguarda la componente algoritmica (intesa come la parte che non riguarda la GUI), è composta dalle Activity, ossia le varie schermate o pagine di un'applicazione Android, le quali si occupano di tradurre gli input dell'utente in output implementati tramite codice java.

Sulle Activity si appoggiano i thread, utilizzati per gestire l'esecuzione del gioco nella sua ripetizione ciclica, infine gli asynctask, sono utilizzati per alleggerire l'esecuzione dei thread sulla User Interface (Figura 6.4).

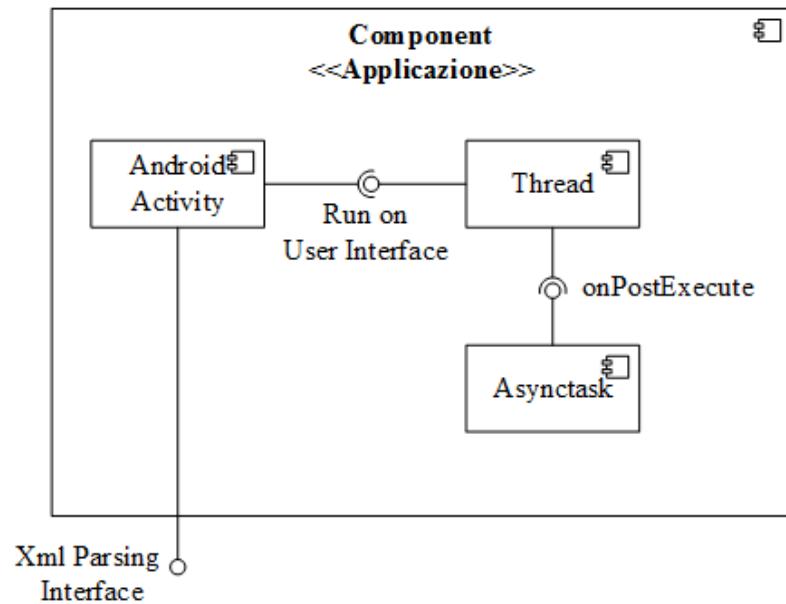


Figura 6.4: Component applicazione

6.2.4 Il server

```
//TODO: Descrizione server -testo inserito per capire l'impaginazione - scrivere almeno 2 righe testo testo
```

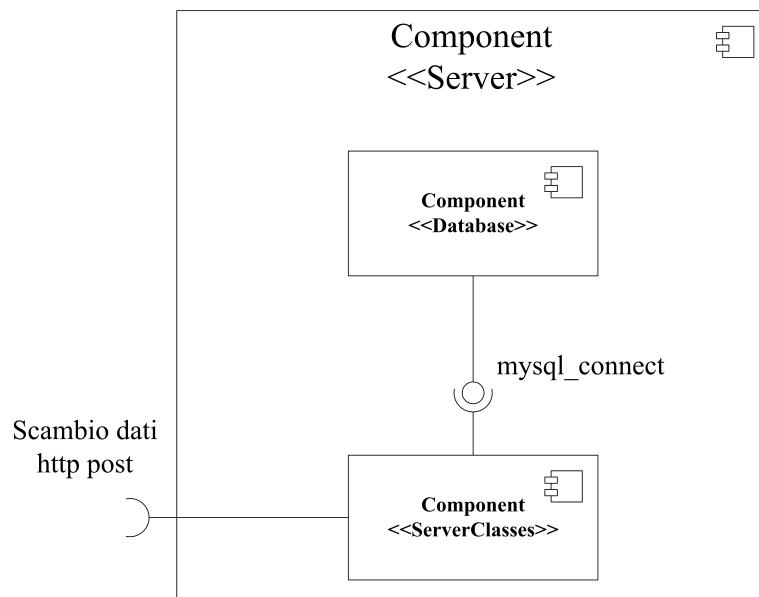


Figura 6.5: Component diagram Server

6.2.5 Database

//TODO: Descrizione DB - testo inserito per capire l'impaginazione - scrivere almeno 2 righe testo testo

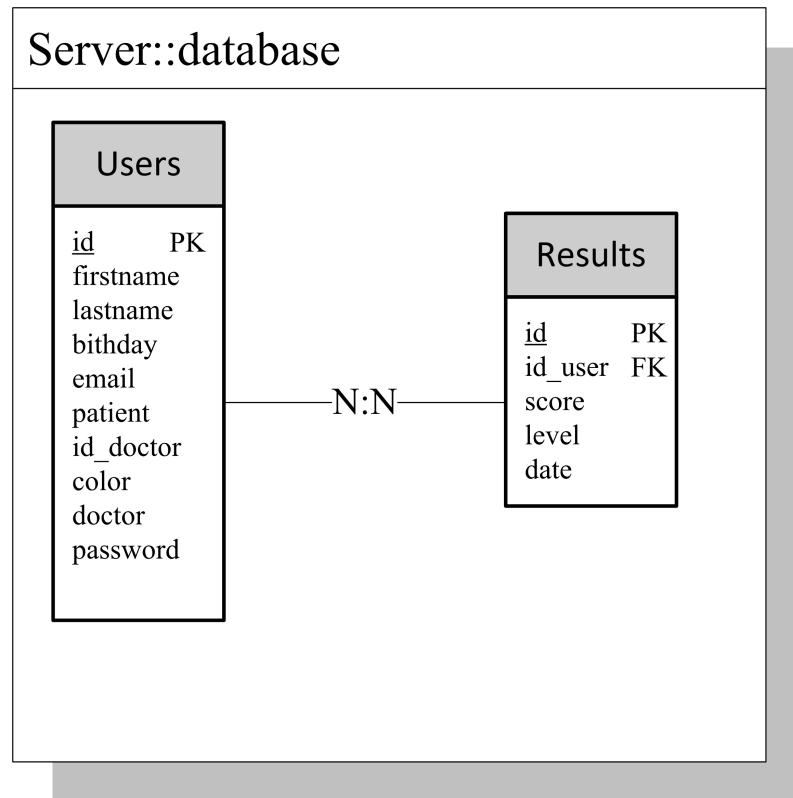


Figura 6.6: Component database

6.3 Sequence diagram

6.3.1 Funzionamento con log in

All'avvio dell'applicazione, se l'utente ha popolato correttamente i campi email e password, nell' interfaccia corrente viene visualizzata una “rotella” di attesa (dialog ring), mentre in background vengono scatenate una serie di chiamate:

- alla classe PostCall vengono passate la stringa contentente l'email e quella della password
- la classe PostCall effettua una chiamata post al server cs.unibg.it, più precisamente al servizio 3d4ambservice.php

Il servizio php verifica nel database se l'email e la password sono corrette, e manda un messaggio di risposta alla PostCall, la quale effettua un parsing della risposta e restituisce un messaggio all'interfaccia corrente.

A questo punto la dialog ring viene interrotta, e nello scenario di un' autenticazione corretta, viene lanciata l'activity TutorialActivity, nella quale è presente una breve spiegazione del gioco, gestita tramite un thread.

Terminata la presentazione della TutorialActivity (che può anche essere saltata spuntando un'apposita checkbox) viene lanciata la SplashActivity, la quale contiene un' animazione d'avvio. Viene avviata ora la SettingsActivity, nella quale viene chiesto qual è l'occhio pigro che si vuole trattare.

Confermata la scelta dell'occhio, si apre la MainActivity, contenente il gioco vero e proprio, i thread principali sono tre:

1. GameThread: che gestisce i nemici e l'auto dell'utente
2. TimerPanorama: per la gestione del panorama laterale
3. TimerSun: per la gestione del sole e delle nuvole sullo sfondo

Terminato il gioco (gameover) è possibile richiedere i risultati via mail oppure ricominciare a giocare, in ogni caso viene fatta una chiamata post per memorizzare i risultati sul server (Figura 6.7).

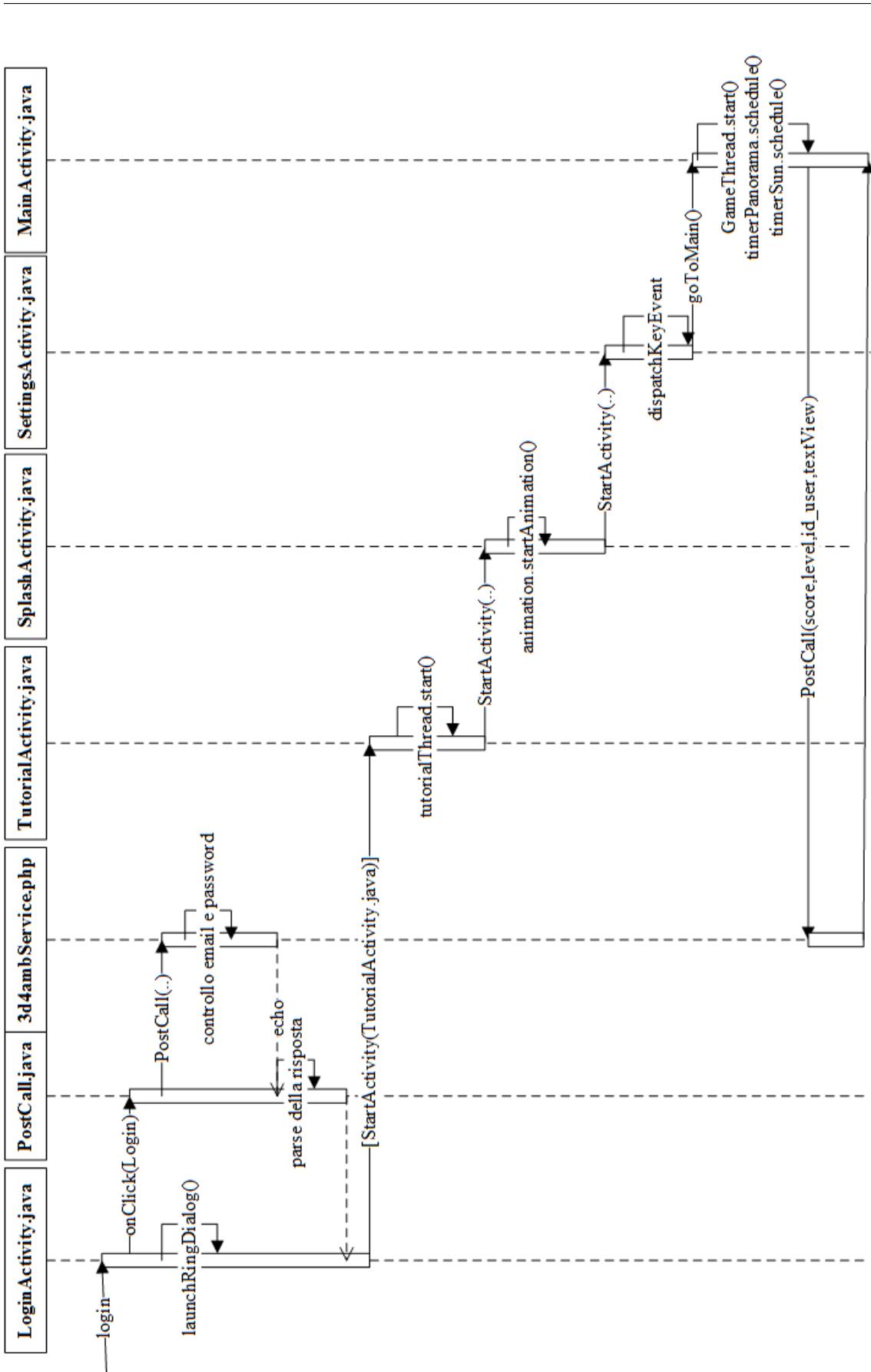


Figura 6.7: Sequence diagram funzionamento app con log in

6.3.2 Funzionamento senza log in

Il funzionamento in modalità “play without registration” è identico a quello del log in utente, infatti, in questo caso è stato definito un utente di default, al quale vengono associati tutti i risultati degli utenti che hanno giocato senza registrarsi.

6.3.3 Funzionamento sign up

Per quanto riguarda la registrazione di nuovi utenti, una volta premuto il bottone “SIGN UP”, si apre un pop up di warning, dopo di che ,se è stato premuto il tasto di conferma, come verrà poi approfondito nella sezione 7.4, l’utente deve fornire: nome, cognome, email, colore preferito, data di nascita e password.

Tutti i dati vengono inviati classe PostCall (con una chiamata analoga a quella per il log in), che a sua volta effettua una chiamata post al servizio 3d4ambservice.php. Se i dati inseriti sono corretti (l'email è esistente e non sono presenti record nel database con gli stessi attributi, inteso come stessa email, nome e cognome), il nuovo utente viene inserito (Figura 6.8).

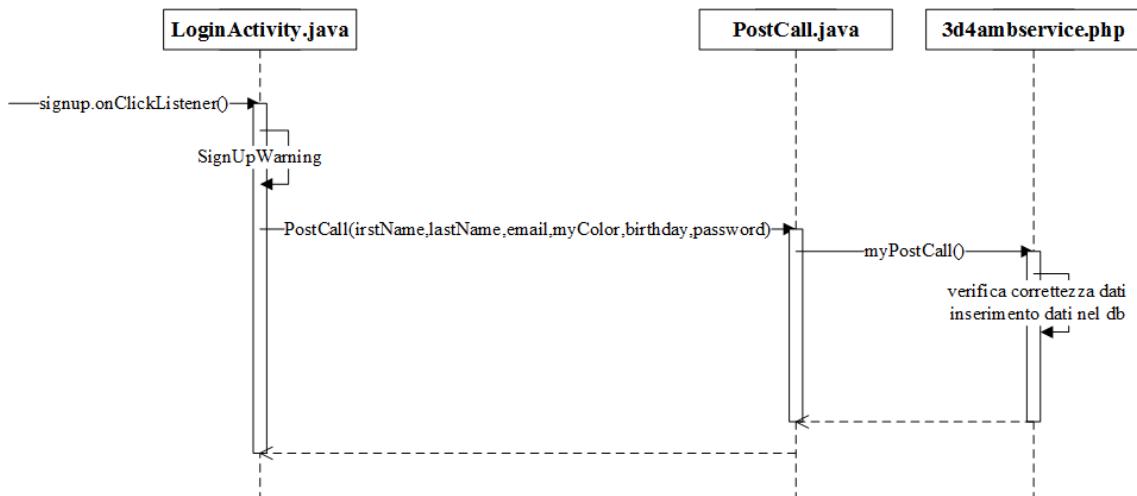


Figura 6.8: Sequence diagram funzionamento app registrazione utente (sign up)

Capitolo 7

Analisi degli Algoritmi

In questo capitolo vengono analizzati gli algoritmi che compongono l' applicazione. Gli algoritmi utilizzati sono: penalizzazione progressiva delle immagini (7.1), generazione casuale della scena (7.2),cifratura delle password degli utenti (7.3), servizio web (7.4), gestione degli scontri con i nemici (7.5), thread principale del gioco (7.6), gestione delle animazioni (7.7) ed i task asincroni per la gestione di una parte della grafica (7.8).

7.1 Penalizzazione progressiva delle immagini

La penalizzazione progressiva delle immagini avviene modificando la trasparenza tramite il parametro alpha value che può assumere valori tra 0 e 255. Un' immagine con alpha=0 è trasparente, mentre con alpha=255 conserva tutte le sue caratteristiche. Nell'applicazione sviluppata la trasparenza dell'immagine viene aumentata all'aumentare del livello e rimane costante al valore 50 dopo l'ottavo livello. Dall'ottavo livello la difficoltà del gioco viene aumentata 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}
```

7.2 Generazione casuale della scena

La scena è suddivisa in due soggetti principali: i nemici e il paesaggio.

7.2.1 Nemici

Nella scena ci sono tre corsie dalle quali possono arrivare i nemici (Figura 4.2). Inizialmente la corsia viene scelta con un'estrazione random, ma per evitare che venga scelta sempre la stessa corsia e il giocatore resti troppo a lungo fermo, viene fatto un controllo sulla scelta precedente. Un nemico non viene mai mandato più di due volte consecutive 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) selectedLane=1;
8         }
9     }
10    previousLane=selectedLane;
11    selectedCar=(int)(numberOfCar*Math.random())+1;
12 }
```

7.2.2 Panorama

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

```
1
2 public void randomPanorama() {
3     int pick = new Random().nextInt(Side.values().length);
4     selectedSide= Side.values()[pick];
5     idSubject= (int)(numberOfSubject*Math.random())+1;
6     if(previousSide==selectedSide){
7         countPrevious++;
8         if(countPrevious>3){
9             selectedSide++;
10            if(selectedSide>3) selectedSide=1;
11        }
12    }
13    previousSide=selectedSide;
14 }
```

7.3 Cifratura password

Per garantire la privacy degli utenti le password vengono memorizzate nel database utilizzando l'algoritmo di cifratura sha256.

Solitamente tale algoritmo viene utilizzato per generare un' "impronta digitale" di un documento. È stato utilizzato in questa applicazione poichè in java viene fornito un metodo per la cifratura in sha256.

In seguito è mostrato il codice implementato.

```
1 public static String calculateHash(String data) {  
2     String hash = new String(Hex.encodeHex(DigestUtils.  
3         sha256(data)));  
4     return hash;  
5 }
```

7.4 Servizio web

La comunicazione tra **client** (smarthphone) e **server** (cs.unibg.it) avviene tramite chiamate post (vedi anche sezione 6.2.1), le chiamate sono sincrone, ossia ad un messaggio inviato dal client corrisponde sempre una risposta dal servizio. Le chiamate partono sempre dal client, il server di suo non invia messaggi.

Chiamata post la chiamata viene effettuata al servizio php di Car Racing Cardboard (3d4ambService.php), all'interno del metodo *public void myPostCall(final TypeCall type, final Activity logInActivity)* tramite un thread.

```
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     }
}
```

Le chiamate che possono essere effettuate dal client sono varie e per ognuna è stata predisposta una lista di stringhe che possono essere passate all'oggetto htppost, ossia il preposto, all'interno del metodo myPostCall, a gestire la connessione con il servizio php.

```
1 HttpPost htppost = new HttpPost("http://3d4amb.unibg.it  
/3dcar/3d4ambService.php");
```

Dopo aver inizializzato l'oggetto htppost vengono inizializzate le stringhe sopra citate

```
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);
```

Infine in base al tipo di chiamata che deve essere effettuata, tali stringhe vengono valorizzate e assegnate tramite un apposito metodo all'oggetto htppost. In seguito è mostrato l'esempio della chiamata di log in.

```
1 if (type.equals(TypeCall.LOG_IN)) {  
2     nameValuePairsLogIn.add(new BasicNameValuePair("type  
", "log_in"));
```

```

3     nameValuePairsLogIn.add(new BasicNameValuePair("email", email));
4     nameValuePairsLogIn.add(new BasicNameValuePair("password", password));
5     httpPost.setEntity(new UrlEncodedFormEntity(
6         nameValuePairsLogIn));
    }

```

Gestione delle chiamate lato server: le chiamate effettuate lato client, gestite tramite l'arrayList `ArrayList<NameValuePair>(i)`, vengono ricevute lato server eseguendo una Post sul parametro type, in seguito alla connessione a MySql ed alla selezione del database (come mostrato nel codice)

```

1
2 $server = "localhost";
3 $user = "3d4amb";
4 $pass = "amb43d";
5 $db = "cardboard";
6
7 //Connect to mysql
8 $conn=mysql_connect($server,$user,$pass) or die("errore
9
10 mysql");
11 mysql_select_db($db) or die("sorry, can't connect to the
12 database.");
13 $type = $_POST['type'];

```

In seguito all'inizializzazione del parametro type, che specifica se è stata richiesta una procedura di sign up, log in,log in come dottore, report, reset della password

o update dei dati personali, viene eseguito uno 'switch' per intraprendere diverse azioni in base alla procedura scelta.

Log in

La chiamata Post per il log in avviene ogni volta che l'utente preme l'apposito pulsante (Figura 7.1):

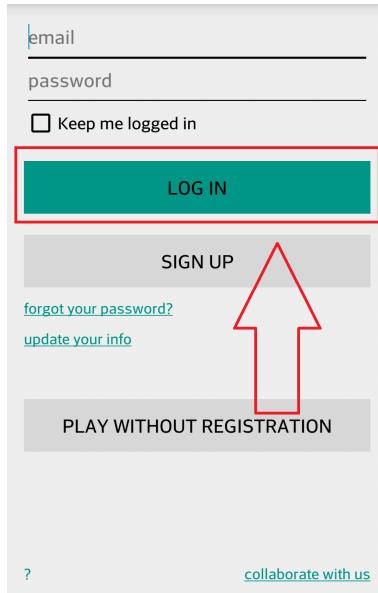


Figura 7.1

Per effettuare il log in vengono inviate al server email e password, che prima di essere inviata viene cifrata (vedi 7.3).

All' utente viene mostrata una finestra in attesa di una risposta dal server (Figura 7.2):

Se il servizio risponde in maniera positiva viene lanciata l'activity successiva (SplashActivity o TutorialActivity a seconda delle impostazioni utente), altrimenti viene mostrata una finestra contenente una descrizione dell'errore.

La procedura di Log In è trattata lato server nel seguente modo.

```
1
2 case "log_in":
3     $emailFound = false;
4     $email = $_POST['email'];
5     $password = $_POST['password'];
```

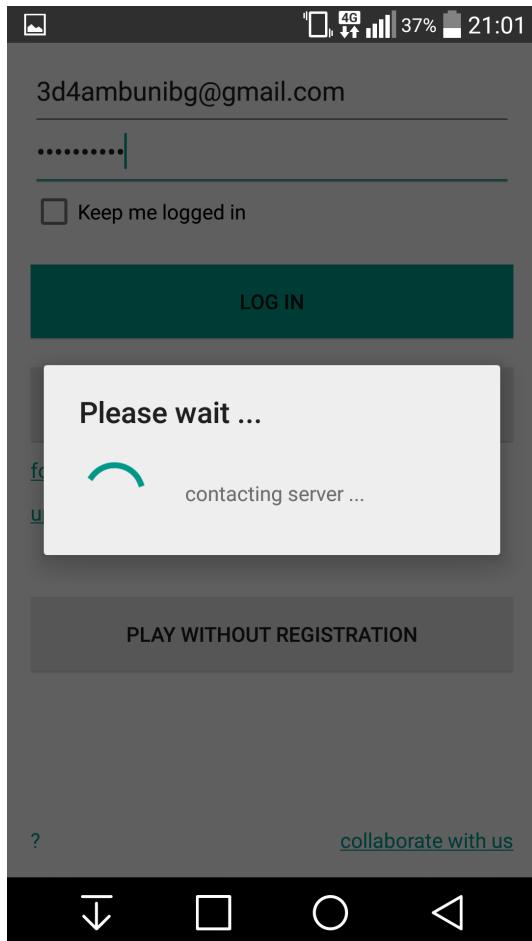


Figura 7.2: interfaccia di attesa

```
6 $sqlSelectAllFromUsers = "SELECT * FROM users";
7 $querySelectAllFromUsers = mysql_query(
8 $sqlSelectAllFromUsers);
9
9 while($row = mysql_fetch_array($querySelectAllFromUsers
10 )) {
11     if($email == $row['email'] && $email != NULL) {
12         $emailFound = true;
13         if($password == $row['password']) {
14             $statusConnection = true;
15             $id_user = $row['id'];
16             $doc = $row['doctor'];
```

```
16     }
17
18     else {
19         echo "password errata";
20     }
21 }
22
23 if ($emailFound == false) {
24     echo "emailNotFound";
25 }
26 if ($statusConnection == true) {
27     echo "connection/".$id_user."/". $doc;
28 }
29
30 break;
```

Come si può vedere sopra, viene eseguito un controllo sui campi email e password e viene restituita una stringa che informa sull'esito dell'operazione. Viene inoltre restituito un valore concatenato (\$doc, 0 or 1) per indicare se si tratta del Log In di un utente comune o di un dottore, il quale dovrà quindi aggiungere successivamente i dati del paziente.

Sign up

L'evento post call per la registrazione (sign up) avviene dopo che l'utente ha premuto il tasto "ok" (Figura 7.3)

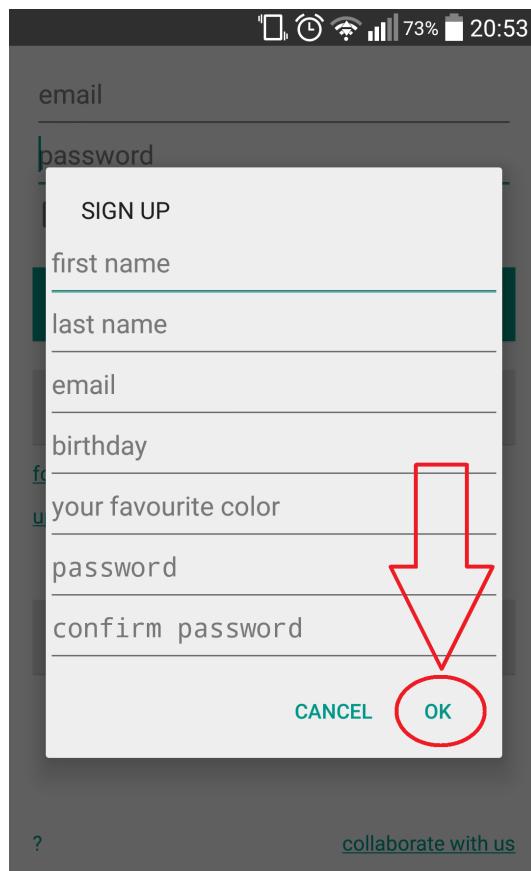


Figura 7.3

Al server vengono inviati i seguenti parametri: nome, cognome, email, colore preferito, data di nascita e password.

Se va tutto a buon fine il nuovo utente viene inserito nel database:

```
1  
2 case "sign_up":  
3  
4     $firstname = $_POST['first_name'];  
5     $lastname = $_POST['last_name'];  
6     $email = $_POST['email'];
```

```
7  $password = $_POST['password'];
8  $color = $_POST['color'];
9  $date = $_POST['date'];
10 $doctor = false;
11 $emailAlreadyChecked = false;
12 $patient = false;
13
14 // include SMTP Email Validation Class
15 require_once('smtp_validateEmail.php');
16
17 // instantiate the class
18 $SMTP_Validator = new SMTP_validateEmail();
19
20 // turn on debugging if you want to view the SMTP
21 // transaction
22 // $SMTP_Validator->debug = true;
23
24 // do the validation
25 $results = $SMTP_Validator->validate(array($email),
26                                         "3D4AmbUnibg@gmail");
27
28 if ($results[$email]) {
29     $sqlSelectAllFromUsers = "SELECT * FROM users";
30     $querySelectAllFromUsers = mysql_query(
31     $sqlSelectAllFromUsers);
32     while($row = mysql_fetch_array(
33         $querySelectAllFromUsers)) {
34         if($email == $row['email']) {
35             $emailAlreadyChecked = true;
36         }
37     }
38 }
```

```

35
36     if ($emailAlreadyChecked == false) {
37
38         //insert user into DB
39
40         $sqlInsertNewUser = "INSERT INTO users (firstname,
41             lastname,birthday,email,patient,color,password)
42             values ('$firstname','$lastname','$date','$email',
43             '$patient','$color','$password')";
44
45         $queryInsertNewUser = mysql_query($sqlInsertNewUser
46 );
47
48         //send mail registration
49
50         gmail($email,"REGISTRATION 3D4AMB","Hello dear ".
51             $email.", welcome to 3d4Amb!".'
52             <br> <br>'.
53             "The 3D4Amb project aims at developing a system
54             based on the 3D for the diagnosis and treatment of
55             amblyopia in young children.
56
57             Below you can find your authentication data:".'
58             <br>'.
59             "<br>".Firstname: ".$firstname.'<br>'.
60
61             "Lastname: ".$lastname.'<br>'.
62
63             "Email: ".$email.'<br>'.
64
65             "Birthday: ".$date.'<br>'.
66
67             "Favourite color: ".$color.'<br> <br>'.
68
69             "If you want change your info, you can go over '
70             update you info'.
71
72             Thanks for using our application, best regards.".'
73             <br> <br>'.
74             " 3d4amb Staff".'
75             <br> <br>'.
76             "Contact us on:
77             3D4AmbUnibg@gmail.com"
78
79             .'
80             <br> <br>'.
81             "Visit our site on: http://3d4amb.
82             unibg.it";
83
84
85             echo "emailChecked";

```

```
55     }
56
57     else {
58         echo "emailAlreadyExist";
59     }
60
61     else {
62         echo "emailNotChecked";
63     }
64
65 break;
```

Nella gestione della registrazione utente lato server vengono innanzitutto inizializzate le variabili contenenti i dati dell'utente catturati dalle textBox dell'applicazione. In seguito viene controllato che l'email inserita non sia già presente nel database. I controlli relativi all'inserimento di campi nulli o non validi vengono effettuati direttamente dall'applicazione a lato client. Se i valori inseriti dall'utente sono corretti e l'email non è già presente nel database si procede alla registrazione dell'utente nel database ed all'invio di un' email di benvenuto. Durante questa fase si è utilizzata la funzione 'gmail(\$to, \$subject, \$message)' appartenente alla classe 'SMTP_validateEmail', in cui viene eseguito un controllo approfondito sull'esistenza e sulla validità dell' email inserita dall'utente. In caso di esito positivo o negativo, viene sempre restituita una stringa a lato client che specifica l'esito dell' operazione. Se la procedura viene eseguita con successo, l'utente riceve una mail di conferma analoga a quella mostrata in figura 7.4

Hello dear ~~xxxxxxxxxxxxxx~~, welcome to 3d4Amb!

The 3D4Amb project aims at developing a system based on the 3D for the diagnosis and treatment of amblyopia in young children. Below you can find your authentication data:

Firstname: Pino
Lastname: Rosso
Email: ~~xxxxxxxxxxxxxx~~
Birthday: 1991-5-27
Favourite color: green

If you want change your info, you can go over 'update you info'. Thanks for using our application, best regards.

3d4amb Staff

Contact us on: 3D4AmbUnibg@gmail.com

Visit our site on: <http://3d4amb.unibg.it>

Figura 7.4: email di conferma registrazione

Reset

L'utente può resettare la password premendo sul tasto “forgot your password?”.

Viene richiesta l'email e il colore preferito; se i dati superano la verifica sul database, viene generata random una nuova password alfanumerica di otto caratteri, che potrà poi successivamente essere personalizzata (vedi 4.3). Verrà inoltre inviata una email come mostrato in figura 7.5

Hello dear ~~xxxxxxxxxxxxxx~~

Your password has been changed and the new password is: 5d31jg38. If you want customize your password, you can go over 'update you info'. Thanks for using our application, best regards.

3d4amb Staff

Contact us on: 3D4AmbUnibg@gmail.com

Visit our site on: <http://3d4amb.unibg.it>

Figura 7.5: email di reset della password

L'operazione di reset della password è gestita a livello server nel seguente modo:

```
1
2   case "reset":
3
4     $new_password = $_POST['new_password'];
5     $new_passwordCfr = $_POST['new_passwordCfr'];
6     $email = $_POST['email'];
7     $color = $_POST['color'];
```

```
8 $sqlSelectColor = "SELECT * FROM users WHERE email = '"
9     $email'";
10
11 $querySelectColor = mysql_query($sqlSelectColor);
12
13
14 while($row = mysql_fetch_array($querySelectColor)) {
15     if($row['color'] == $color) {
16         $id_user=$row['id'];
17
18         $sqlInsertNewPassword = "UPDATE users SET password
19             ='$new_passwordCfr' WHERE id='$id_user'";
20
21         $queryInsertNewPassword = mysql_query(
22             $sqlInsertNewPassword);
23
24
25         //send email password changed
26         gmail($email,"PASSWORD CHANGED SUCCESSFULLY","Hello
27             dear ".$email."!".'
28             <br> <br>'.
29             "Your password
30             has been changed and the new password is: ".
31             $new_password".
32
33             If you want customize your password, you can go
34             over 'update you info'.
35
36             Thanks for using our application, best regards.".'
37             br> <br>'." 3d4amb Staff".'
38             <br> <br>'."Contact us on:
39             3D4AmbUnibg@gmail.com"
40
41             .'<br> <br>'."Visit our site on: http://3d4amb.
42             unibg.it");
43
44             echo "colorTrue";
45
46     }
47
48     else {
49
50         echo "colorFalse";
51
52     }
53
54 }
```

```
    break;
```

Come si può vedere, vengono dapprima catturati i valori relativi alla nuova password (cifrata e non), il valore dell'email ed il valore relativo al colore preferito, necessario per la verifica dell' identità dell'utente. Sulla base dell' email indicata viene poi eseguito un controllo tra il valore di colore inserito e quello indicato in fase di Sign Up. Se i due valori sono congruenti l'operazione di ripristino della password viene eseguita con successo e l' utente riceverà un' email contenente la nuova password.

Doctor call

Nel caso in cui l' utente, effettua l'accesso come dottore, viene reindirizzato ad un' apposita pagina: DoctorActivity. Il dottore può inserire i dati dei pazienti, consentendogli così di gestirli tutti con il suo account su Car Racing Cardboard (vedi anche sezione 4.3).

L' utilizzo dell'applicazione da parte di un dottore prevede inoltre l' avvio del vero proprio gioco con l'identificativo univoco del paziente (id_user), per il monitoraggio delle prestazioni ed il salvataggio dei dati in modo ordinato nel database. La gestione lato server dell' avvio di una partita da parte di un dottore, dopo aver inserito i dati del paziente, è stata realizzata nel seguente modo.

```
1
2 case "doctorcall":
3
4     $first_name = $_POST['first_name'];
5     $last_name = $_POST['last_name'];
6     $id_doctor = $_POST['id_doctor'];
7     $birthday = $_POST['birthday'];
8     $patient = true;
9     $patientAlreadyExist = false;
10
11    $sqlSelectAllFromUsers1 = "SELECT * FROM users";
12    $querySelectAllFromUsers1 = mysql_query(
13        $sqlSelectAllFromUsers1);
14
15    while($row = mysql_fetch_array(
16        $querySelectAllFromUsers1)) {
17
18        if($first_name == $row['firstname'] &&
19            $last_name == $row['lastname'] &&
20            $birthday == $row['birthday'] &&
```

```
18     $patient == true) {
19
20         $patientAlreadyExist = true;
21
22         $id_user = $row['id'];
23
24     }
25
26
27     if($patientAlreadyExist == false) {
28
29         $sqlInsertDoctorPatient = "INSERT INTO users (
30             firstname , lastname , birthday , patient , id_doctor ) values
31             ('$first_name' , '$last_name' , '$birthday' ,
32             '$patient' , '$id_doctor') ";
33
34
35         $queryInsertDoctorPatient = mysql_query(
36             $sqlInsertDoctorPatient );
37
38
39         $sqlSelectAllFromUsers = "SELECT * FROM users";
40
41
42         $querySelectAllFromUsers = mysql_query(
43             $sqlSelectAllFromUsers );
44
45
46         while($row = mysql_fetch_array(
47             $querySelectAllFromUsers )) {
48
49             if($first_name == $row['firstname'] &&
50                 $last_name == $row['lastname'] &&
51                 $birthday == $row['birthday'] &&
52                 $row['patient'] == true) {
53
54                 $id_user = $row['id'];
55
56             }
57
58         }
59
60     }
61
62 }
```

```
44 echo $id_user;
45
46 break;
```

Prima di avviare una partita viene controllato se il paziente è già presente nel database, in tal caso verrà restituito come valore 'id_user' il valore pescato dal database. In caso si tratti di un nuovo paziente, si procede all'inserimento dei nuovi dati nel database e viene creato un nuovo 'id_user'.

Report

Il report generato al termine del gioco differenzia l' utilizzo del gioco da parte di un utente registrato e l' utilizzo del gioco senza registrazione. Nel caso in cui l' utente non sia registrato, alla fine della partita viene semplicemente visualizzata una finestra contenente i dati relativi ai punteggi raggiunti. Nel caso in cui l'utente è registrato, avviene in automatico il salvataggio e il monitoraggio dei risultati raggiunti (Figura 7.6).



Figura 7.6: Schermata finale del gioco

A livello di codice, nel thread del gioco (vedi anche sezione 7.6), viene lanciata una chiamata post, passando come parametri il punteggio ottenuto, il livello raggiunto e i dati dell' utente.

```
1 PostCall postCall= new PostCall(globalData.getScore().  
toString(),globalData.getLevel().toString(),id_user,t2  
,true);  
2 postCall.myPostCall(TypeCall.REPORT,activity);
```

Nel codice mostrato è possibile vedere nel dettaglio la gestione della fase di report lato server per quanto riguarda gli utenti registrati (siano essi pazienti di dottori o

singoli utenti).

```
1
2 case "report_and_email":
3     $score = $_POST['score'];
4     $level = $_POST['level'];
5     $id_user = $_POST['id_user'];
6     $isDoctor = false;
7
8     $sqlInsertReport = "INSERT INTO results(id_user,score,
9         level,date)
10        values ('$id_user','$score','$level',now())";
11
12     $queryInsertReport = mysql_query($sqlInsertReport);
13
14     $sqlSelectAllFromUsers2 = "SELECT * FROM users WHERE id
15         ='$id_user'";
16
17     $querySelectAllFromUsers2 = mysql_query(
18         $sqlSelectAllFromUsers2);
19
20     $sqlSelectAllFromResults = "SELECT * FROM results WHERE
21         id_user=' $id_user'";
22
23     $querySelectAllFromResults = mysql_query(
24         $sqlSelectAllFromResults);
25
26
27     while($row = mysql_fetch_array(
28         $querySelectAllFromUsers2)) {
29
30         $patient_firstname = $row['firstname'];
31         $patient_lastname = $row['lastname'];
32
33
34         if($row['patient'] == NULL) {
35
36             $message="Hello ".$row['email']."! ".'
37             <br> <br>'.
38
39             "These are your results: ".'
40             <br> <br>'.
```

```

23         while($row3 = mysql_fetch_array(
24             $querySelectAllFromResults)) {
25
26             $extract=$extract.
27
28             "SCORE: ".$row3['score'].".".'<br>'.
29             "LEVEL: ".$row3['level'].".".'<br>'.
30             "DATE: ".$row3['date'].".".'<br> <br>';
31
32             $message=$message.$extract.
33
34             "Carry on playing in order to improve your lazy eye
35             ! Thank you for using our app. Regards."..'<br> <br>'.
36             "3d4amb Staff"..'<br> <br>'."Contact us on: 3
37             D4AmbUnibg@gmail.com"..'<br> <br>'."Visit our site on:
38             http://3d4amb.unibg.it";
39
40             gmail($row['email'],"3D4AMB SCORE",$message);
41         }
42
43         else {
44
45             $id_doctor = $row['id_doctor'];
46
47             $sqlSelectDoctorFromUsers = "SELECT * FROM users
48             WHERE id='".$id_doctor."'";
49
50             $querySelectDoctorFromUsers = mysql_query(
51                 $sqlSelectDoctorFromUsers);
52
53             while($row1 = mysql_fetch_array(
54                 $querySelectDoctorFromUsers)) {
55
56                 gmail($row1['email'],"3D4AMB SCORE","Hello dear "
57                 . $row1['email'].'!'. '<br> <br>'."There are the results
58                 of your patient's game : "
59
60                 .'<br> <br>'."PATIENT'S FIRSTNAME: ".
61                 $patient_firstname."."
62
63                 .'<br> '. "PATIENT'S LASTNAME: ".$patient_lastname
64                 .""

```

```

42     . '<br> '. "SCORE: ". $score . "
43     ". '<br>' . "LEVEL: ". $level . ". ". '<br> <br>' . "
44     Continues to play to improve your results!
45     Thanks for using our application, best regards." .
46     '<br> <br>' . " 3d4amb Staff". '<br> <br>' . "Contact us on
47     : 3D4AmbUnibg@gmail.com"
48     . '<br> <br>' . "Visit our site on: http://3d4amb.
49     unibg.it");
50
51 }
52
53 echo "report_and_email";
54
55 break;

```

Come si può vedere, una volta finita la partita, i risultati ottenuti dall’ utente vengono inseriti nella tabella ’results’. In seguito viene individuato lo storico dell’ utente dal database e viene inviata una mail che notifica i risultati ottenuti fino all’ ultima partita effettuata. L’ obiettivo è quello di mantenere una cronologia che permetta all’ utente di osservare progressivamente i miglioramenti ottenuti.

7.5 Gestione dello scontro con il nemico

Nel momento in cui si verifica una collisione con un autoveicolo nemico, oltre al decremento del numero di vite a disposizione ed alla gestione di un eventuale Game Over, viene visualizzata a schermo un' esplosione animata nel punto di scontro. Questo effetto è stato progettato basandosi sulle potenzialità dei Thread e sfruttando le potenzialità della classe SurfaceView, messa a disposizione da Android. Essa fornisce una superficie di disegno pienamente controllabile che può essere utilizzata per effettuare delle sovrapposizioni grafiche con altre View. L' accesso alla superficie sottostante è fornito tramite l' interfaccia SurfaceHolder, la quale può essere richiamata usando il metodo getHolder(). Nella classe è necessario implementare i metodi 'surfaceCreated(SurfaceHolder)' e 'surfaceDestroyed(surfaceHolder)' che agiscono quando la superficie viene creata e distrutta. Lo scopo principale della classe SurfaceView è quello di fornire una superficie dove un thread secondario possa renderizzare a schermo. Per questo obiettivo, vengono utilizzate delle Callback sull' oggetto SurfaceHolder che agiscono da filo conduttore tra thread principale e secondario. Di seguito è riportata la classe AnimationExplosionView, che rappresenta l' estensione della classe SurfaceView all'interno dell'applicazione sviluppata.

```
1
2 public class AnimationExplosionView extends SurfaceView {
3     private SurfaceHolder holder;
4     private AnimationLoopThread gameLoopThread;
5     private Sprite spriteExplosion;
6
7     public AnimationExplosionView(Context context)
8     {
9         super(context);
10        gameLoopThread = new AnimationLoopThread(this);
11        setZOrderOnTop(true);
12        holder = getHolder();
```

```

42     }
43
44
45     private void createSprite()
46     {
47         Bitmap bmpExplosion = BitmapFactory.decodeResource(
48             getResources(),
49             R.drawable.explosion_anim);
50         spriteExplosion = new Sprite(this, bmpExplosion, 12);
51     }
52
53     public void onDrawAnimationExplosionView(Canvas canvas)
54     {
55         if (canvas != null)
56             spriteExplosion.onDrawExplosion(canvas);
57     }
58 }
```

Come si può osservare dal codice, nella classe vengono utilizzati oggetti di tipo Bitmap. Questi oggetti sono molto utili per la rappresentazione di immagini in formato raster. Inoltre viene istanziato anche un oggetto GameLoopThread, il quale ha il compito di richiamare ad intervalli regolari il metodo onDrawAnimationExplosionView(Canvas canvas). Il principio dell' animazione dell' esplosione è dato dallo scorrimento di un vettore diviso in colonne aventi la medesima dimensione. Ogni colonna contiene 'l' evoluzione' dell' animazione e lo scorrimento veloce da parte del thread simula l'effetto dell'esplosione.

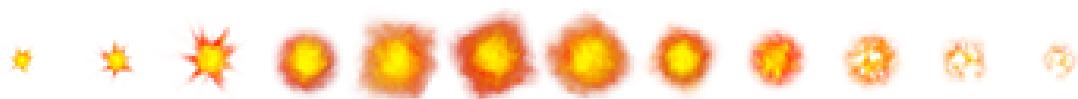


Figura 7.7: Vettore esplosione

In AnimationExplosionView viene inoltre istanziato l'oggetto spriteExplosion della classe Sprite. Il costruttore della classe Sprite accetta come parametri la SurfaceView , la Bitmap ed il numero di colonne in cui è divisa la bitmap da scorrere. Il metodo richiamato da GameLoopThread 'onDrawAnimationExplosion(Canvas canvas)' chiama a sua volta il metodo 'onDrawExplosion(Canvas canvas)' sull'oggetto spriteExplosion. Tale metodo va effettivamente a disegnare a schermo la nostra animazione.

```
1
2 public void onDrawExplosion(Canvas canvas)
3 {
4     updateExplosion();
5     if(canvas!=null)
6         canvas.drawColor(Color.TRANSPARENT, PorterDuff.Mode.
7             CLEAR);
8
9     int srcX = currentFrame * bmpWidth;
10    int srcY = 0;
11    Rect src = new Rect(srcX, 0, srcX + bmpWidth, srcY +
12        bmpHeight);
13    Rect dst = new Rect(xExplosion, yExplosion, xExplosion
14        + bmpWidth, yExplosion + bmpHeight);
15    if(canvas!=null)
16        canvas.drawBitmap(bmp, src, dst, null);
17}
```

La bitmap viene disegnata all'interno di un rettangolo avente le dimensioni dei singoli frame, nel punto specificato dai valori 'xExplosion' e 'yExplosion'. Il metodo updateExplosion() non fa altro che scorrere la nostra immagine sulla base del frame corrente e del numero di colonne specificato in fase di costruzione.

```
1
2     private void updateExplosion()
3     {
4         currentFrame = ++currentFrame % nColumns;
5     }
```

7.6 Game Thread

La parte principale dell'applicazione è stata sviluppata all'interno della classe GameThread. Essa estende la classe Thread ed effettua l' override di alcune funzioni, come ad esempio la funzione 'run()', che vedremo in seguito in dettaglio. Nel costruttore della classe GameThread vengono accettati ed in seguito istanziati tutti gli oggetti relativi alla composizione grafica del gioco, come ad esempio le immagini (di tipo imageView) relative agli autoveicoli, quelle relative al paesaggio, le textBox principali ed i layout (di tipo relativeLayout) riguardanti l' occhio destro e sinistro, con le relative dimensioni di larghezza ed altezza. Altro importante parametro accettato è l' enumerativo che identifica l'occhio scelto. Sono inoltre presenti l'activity da cui parte la chiamata a questo thread, l'id dell'utente ed il parametro della classe GlobalData, per tenere traccia dei parametri del gioco. Di seguito viene riportata la segnatura del costruttore precedentemente descritto:

```
1  
2 public GameThread(Activity activity, TextView text1,  
3 TextView text2,  
4 TextView tLevelLeft, TextView tLifeLeft, TextView  
5 tScoreLeft,  
6 TextView tLevelRight, TextView tLifeRight, TextView  
tScoreRight, ArrayList<ImageView> i1, ArrayList<  
ImageView> i2, ArrayList<ImageView> i3, ArrayList<  
ImageView> i4, ArrayList<ImageView> i5, ArrayList<  
ImageView> i6, ImageView target1, ImageView target2,  
ImageView target3,  
5 GlobalData globalData, Eye eye, RelativeLayout  
RLAnimationLeft, RelativeLayout RLAnimationRight,  
6 int width, int height, String id_user) { ... }
```

La parte principale della classe GameThread è contenuta all' interno del metodo run(). Le principali azioni vengono lanciate all' interno del metodo runOnUiThread(), il quale esegue le procedure specificate sul thread dell' interfaccia utente.

Nel thread viene controllato il numero di vite attuale dell' utente. Se il valore delle vite rimaste è pari a '0', vengono eseguite le procedure relative alla fase di GameOver. Se l'utente è registrato viene mostrato il report a schermo e viene inviata la mail contenente lo storico ed i progressi dell'utente, altrimenti verrà visualizzato solo il report. Nel caso in cui il valore del numero di vite sia maggiore di 0, vengono mantenuti aggiornati i valori di score, level e numero di vite, e viene settata la corsia sul quale immettere il nuovo nemico scelta dall'algoritmo random precedentemente esplicato. Si esegue successivamente un controllo sulla corsia scelta per predisporre le animazioni a destra, a sinistra o nella parte centrale dello schermo. Se ad esempio il nemico viene posto sulla corsia centrale (pick = 2), le azioni intraprese sono le seguenti:

```
1
2 if (pick == 2) {
3     if(gameManager.getIdEnemy().get(i).getNumberOfCar()==1)
4     {
5         animationEnemies.showImage(enemyLeftLane2.get(0));
6         animationEnemies.showImage(enemyRightLane2.get(0));
7         animationEnemies.animateFrontCarLane2(enemyLeftLane2.
8             get(0), enemyRightLane2.get(0),
9             displayWidth, displayHeight);
10    }
11    if(gameManager.getIdEnemy().get(i).getNumberOfCar()==2)
12    {
13        animationEnemies.showImage(enemyLeftLane2.get(1));
14        animationEnemies.showImage(enemyRightLane2.get(1));
15        animationEnemies.animateFrontCarLane2(enemyLeftLane2.
16            get(1), enemyRightLane2.get(1),
17            displayWidth, displayHeight);
18    }
19    if(gameManager.getIdEnemy().get(i).getNumberOfCar()==3)
20    {
```

```

16     animationEnemies.showImage(enemyLeftLane2.get(2));
17     animationEnemies.showImage(enemyRightLane2.get(2));
18     animationEnemies.animateFrontCarLane2(enemyLeftLane2.
19         get(2), enemyRightLane2.get(2),
20             displayWidth, displayHeight);
21 }
22 if(gameManager.getIdEnemy().get(i).getNumberOfCar()==4)
23 {
24     animationEnemies.showImage(enemyLeftLane2.get(3));
25     animationEnemies.showImage(enemyRightLane2.get(3));
26     animationEnemies.animateFrontCarLane2(enemyLeftLane2.
27         get(3), enemyRightLane2.get(3),
28             displayWidth, displayHeight);
29 }
30 onAnimationTimer();
31 animationTarget.animateTarget2(target2, displayWidth,
32     displayHeight);
33 animationEnemies = new AnimationEnemies();
34 }
```

L’ implementazione dell’animazione sull’ oggetto `animationEnemies` verrà descritta più in dettaglio nella sezione 7.7. Il metodo `onAnimationTimer()` implementa i listener contenenti i metodi `onAnimationRepeat`, `onAnimationStart` ed `onAnimationEnd`, necessari per la gestione dei vari eventi rispettivamente a inizio e fine animazione. Dato che non è possibile agire nel momento in cui le animazioni sono in corso, ma soltanto ad inizio o fine, la soluzione adottata per far scomparire all’attimo esatto le macchine evitate è stata quella di creare, oltre alle `imageView` degli autoveicoli, delle `imageView` coincidenti, con l’ obiettivo di fare da ‘target’. Le `imageView` che fanno da target hanno visibilità nulla. L’ animazione di quest’ ultime è rallentata rispetto alle `imageView` delle auto. In questo modo è possibile agire

sulla visibilità delle auto durante gli eventi lanciati e catturati dai listener riferiti ai target. Di seguito si riporta l' esempio di tale implementazione adottato nella corsia centrale:

```
1
2     @Override
3     public void onAnimationEnd(Animation animation)
4     {
5         globalData.setEnd2(true);
6         t1.setText("FINITA ANIMAZIONE 2");
7         if(globalData.isEnd2() && globalData.
8             getAbsolutePosition()==2) {
9             globalData.decreaseLife();
10            textLifeLeft.setText("LIFE: " + globalData.getLife().
11                toString());
12            textLifeRight.setText("LIFE: " + globalData.getLife()
13                .toString());
14            t2.setText("COLLISIONE SU 2");
15            animationExplosionViewLeft = new
16            AnimationExplosionView(activity.getApplicationContext
17            ());
18            animationExplosionViewRight = new
19            AnimationExplosionView(activity.getApplicationContext
20            ());
21            animationExplosionViewLeft.setX((float) (displayWidth
22            *0.135));
23            animationExplosionViewLeft.setY((float) (
24            displayHeight*0.25));
25            animationExplosionViewRight.setX((float) (
26            displayWidth*0.135));
27            animationExplosionViewRight.setY((float) (
28            displayHeight*0.25));
```

```

18     relativeLayoutAnimationLeft.addView(
19         animationExplosionViewLeft);
20     relativeLayoutAnimationRight.addView(
21         animationExplosionViewRight);
22     animationEnemies.hideImage(enemyLeftLane2.get(0));
23     animationEnemies.hideImage(enemyRightLane2.get(0));
24     animationEnemies.hideImage(enemyLeftLane2.get(1));
25     animationEnemies.hideImage(enemyRightLane2.get(1));
26     animationEnemies.hideImage(enemyLeftLane2.get(2));
27     animationEnemies.hideImage(enemyRightLane2.get(2));
28     animationEnemies.hideImage(enemyLeftLane2.get(3));
29     animationEnemies.hideImage(enemyRightLane2.get(3));
30 }
31 else {
32     globalData.increaseScore();
33     textScoreLeft.setText("SCORE: " + globalData.getScore()
34         .toString());
35     textScoreRight.setText("SCORE: " + globalData.
36         getScore().toString());
37 }
38 }
```

Come si può vedere, si agisce sugli oggetti `animationEnemies` sfruttando l' evento `onAnimationEnd(Animation animation)` generato dal target. In particolare, oltre alla gestione della scomparsa dei veicoli si effettua un controllo sull' eventuale collisione con un auto nemica. Se lo scontro ha luogo, viene lanciata l' animazione dell'esplosione.

7.7 Gestione animazioni

La gestione delle animazione relative alle auto nemiche ed al paesaggio, è stata realizzata estendendo la classe Animation messa a disposizione da Android. In seguito verrà presentato l' esempio relativo alle auto nemiche; per quanto riguarda il panorama, è stato applicato il medesimo principio.

Per ogni corsia, è stato necessario creare un 'AnimationSet', ovvero un gruppo di animazioni riproducibile simultaneamente:

```
1  
2 public AnimationSet animationSetLane1 = new AnimationSet(  
3     false);  
4 public AnimationSet animationSetLane2 = new AnimationSet(  
5     false);  
6 public AnimationSet animationSetLane3 = new AnimationSet(  
7     false);
```

Sono poi stati definiti tre metodi che implementano il comportamento della ImageView delle auto, a seconda che si trovino nella corsia di destra, di sinistra o in quella centrale. Vediamone un esempio di funzionamento nella corsia di sinistra:

```
1  
2 public void animateFrontCarLane1(final ImageView ivLeft,  
3     final ImageView ivRight, int displayWidth, int  
4     displayHeight)  
5 {  
6     TranslateAnimation TranslateAnimation1 = new  
7     TranslateAnimation(0, ((int) (Animation.  
8         RELATIVE_TO_SELF - (displayWidth * 0.04))),  
9     0, ((int) (Animation.RELATIVE_TO_SELF + (displayHeight *  
10        0.08))));
```

```
6     ScaleAnimation ScaleAnimation1 = new ScaleAnimation(1,
7         5.2f,
8         1, 5.2f, Animation.RELATIVE_TO_SELF, 0.9f, Animation.
9             RELATIVE_TO_SELF, 0.5f);
10            animationSetLane1.addAnimation(TranslateAnimation1);
11            animationSetLane1.addAnimation(ScaleAnimation1);
12            animationSetLane1.setDuration(4600);
13
14        ivLeft.startAnimation(animationSetLane1);
15        ivRight.startAnimation(animationSetLane1);
16    }
```

Come si può vedere dalla segnatura del metodo, vengono passati i due oggetti relativi alle imageView delle auto della corsia sinistra. In particolare 'ivLeft' si riferisce all' autoveicolo presente nel relativeLayout di sinistra, mentre 'ivRight' all' autoveicolo del relativeLayout destro. Per fare in modo che le animazioni avessero lo stesso comportamento su ogni dispositivo, si è deciso di applicare le traslazioni e la scala operando sui valori di larghezza ed altezza del display. Per le traslazioni vengono specificati come parametri i punti di partenza relativi alle coordinate 'x' ed 'y' dell'oggetto ed in seguito i punti di arrivo. Per la scala viene specificata la grandezza iniziale dell' imageView ed in seguito è indicato il valore moltiplicativo di scala sull'asse verticale ed orizzontale. Le singole animazioni vengono poi aggiunte all' animationSet tramite il metodo 'addAnimation'. Viene inoltre impostata la durata delle animazioni ed infine vengono avviate utilizzando il metodo 'startAnimation'.

7.8 Asynctask

Per ottimizzare l'utilizzo della 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 `publicProgress(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.[5].

In Car Racing Cardboard gli asynctask sono usati per gestire: il movimento del panorama, il sole e le nuvole sullo sfondo. Di seguito un esempio di utilizzo di asynctask, si tratta in questo caso del task asincrono che gestisce il sole.

¹User Interface

```
1  @Override
2
3  protected void onPostExecute(Void aVoid) {
4      super.onPostExecute(aVoid);
5      animationPanorama.animatePanoramaSun(panoramaLeftSky
6          .get(1), panoramaRightSky.get(1), displayWidth,
7          displayHeight);
8      animationPanorama = new AnimationPanorama();
9  }
```

Nel metodo doInBackground viene inizializzato l'oggetto relativo alla gestione del panorama (sole), mentre nel metodo onPostExecute viene lanciata la sua esecuzione (riga 4), in modo che questa operazione non gravi sul thread della User Interface.

Capitolo 8

Tool utilizzati

In questo capitolo vengono spiegati i tool utilizzati durante lo sviluppo dell'applicazione, il tool principale, Android Studio, è descritto nella sezione (3.1.2), il tool per la sincronizzazione del lavoro da remoto, GitHub (8.1) ed il tool per la gestione dei crash, crashliterals (8.2).

8.1 GitHub

Come repository del lavoro è stato utilizzato GitHub.

GitHub è un servizio web di hosting per lo sviluppo di progetti software, che usa il sistema di controllo di versione Git. Può essere utilizzato anche per la condivisione e la modifica di file di testo e documenti revisionabili.

GitHub offre diversi piani per repository privati sia a pagamento (utilizzato nel nostro caso, grazie all'account 3D4Amb), sia gratuiti, molto utilizzati per lo sviluppo di progetti open-source.

Grazie alla versione Desktop di GitHub, la gestione del lavoro e l'iterazione con il repository è resa ancora più semplificata. I comandi per l'aggiunta, la creazione e la clonazione di un repository direttamente accessibili dall'applicazione:

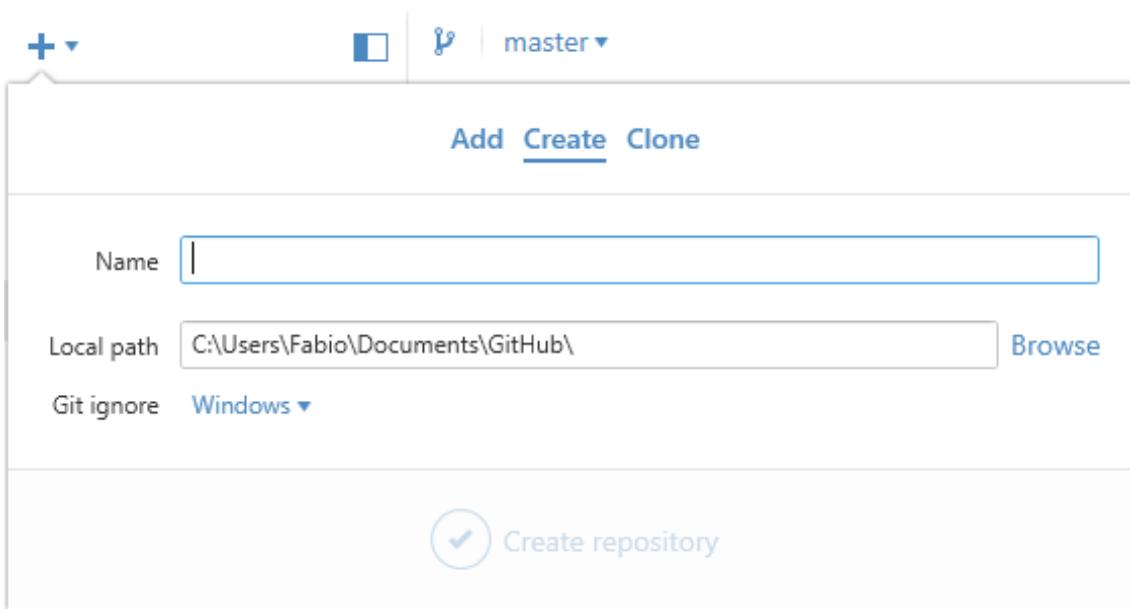


Figura 8.1: Pannello dei comandi GitHub Desktop

Inoltre è possibile gestire i vari progetti e documenti mantenendo uno storico delle modifiche chiaro e tracciabile:

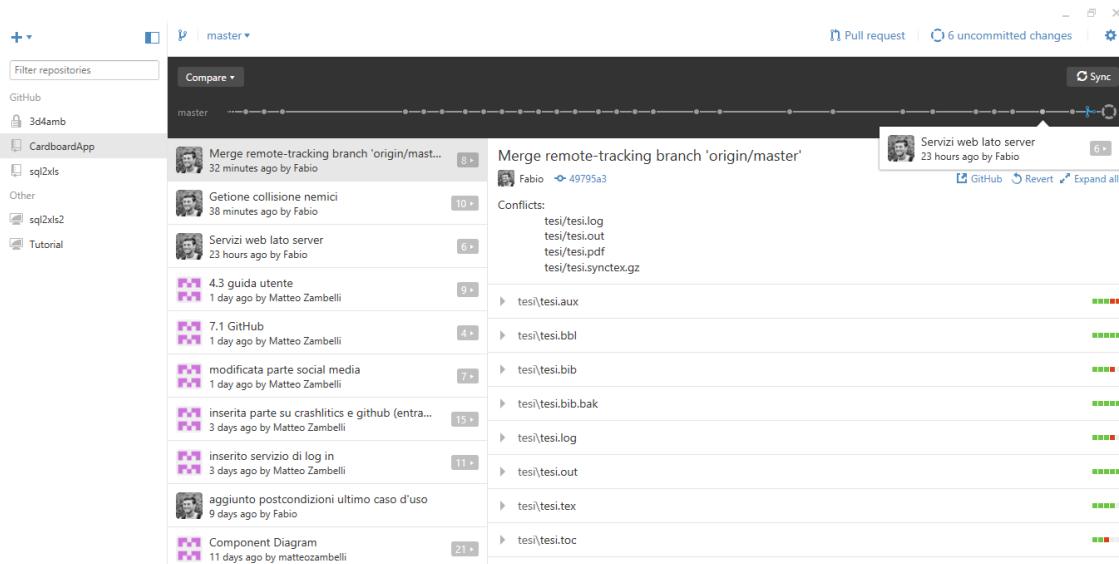


Figura 8.2: Storico GitHub Desktop

Tramite questo tool è quindi possibile creare un account, aggiungere un nuovo progetto ed inviare dati in pochi minuti. Se il progetto venisse inoltre reso open source, sarebbe possibile anche avere un'ampia comunità di sviluppatori che potrebbe osservare il progetto, biforcarlo ed dare un aiuto per migliorarlo.

8.2 Crashlytics

Per tenere traccia di eventuali problemi all'interno dell'applicazione è stato utilizzato Crashlytics¹.

Crashlytics è uno strumento per tenere traccia dei crash all'interno dell'applicazione, segnala in quale activity si è verificato il problema, genera statistiche di utilizzo degli utenti, e consente agli sviluppatori di gestire i bug (marcandoli come risolti o non risolti).

//TODO - aggiornare i dati prima di stampare la tesi (15/8/2015)

Da quando l'applicazione è stata rilasciata (11 Maggio 2015 per la versione 1.1) Si sono verificati in totale 21 crash, molti dei quali(7) dovuti all'assenza di risposta da parte del server.



Figura 8.3: utenti attivi per giorno

¹<https://try.crashlytics.com/>



Figura 8.4: nuovi utenti ogni giorno



Figura 8.5: utenti attivi per mese

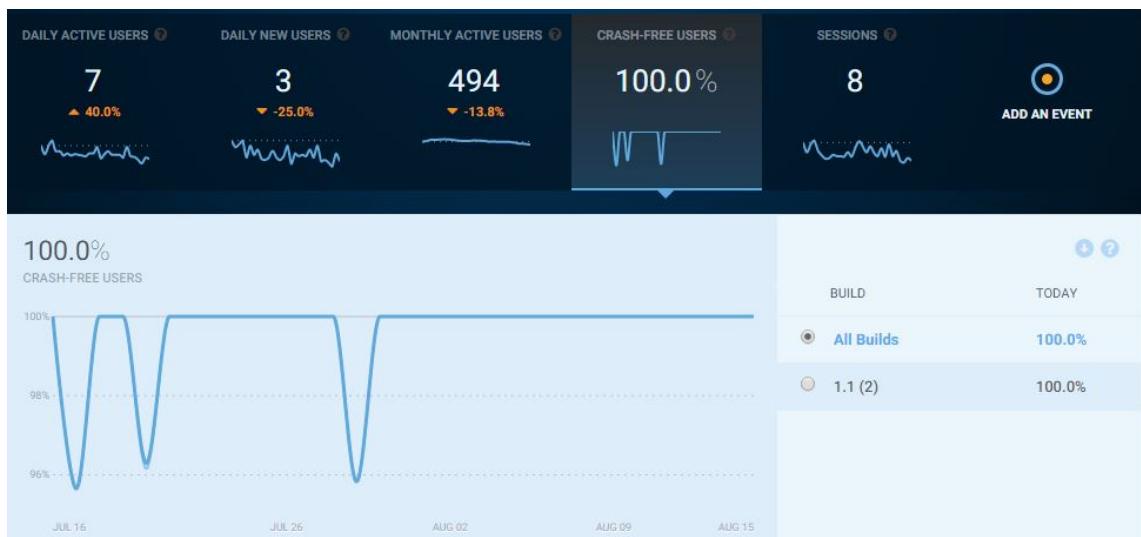


Figura 8.6: utenti che non hanno riscontrato crash

Parte IV

Valutazioni finali

Capitolo 9

Social media

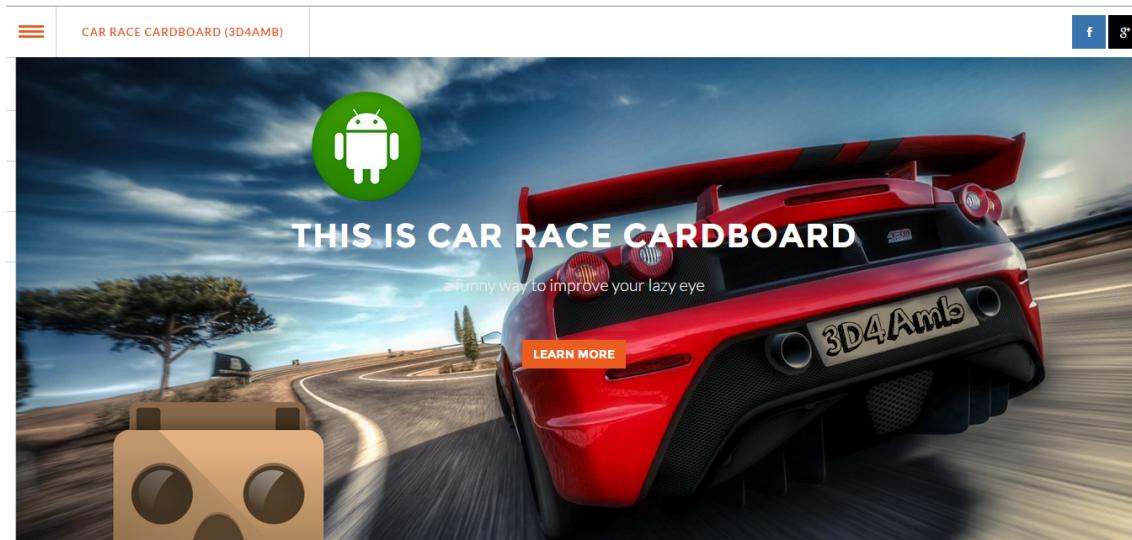


Figura 9.1: Car Racing Cardboard web site

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 (Figura 9.1).

Il sito contiene informazioni sulla strumentazione necessaria per l'utilizzo dell'applicazione, rimanda alla pagina di Google Play per il download e contiene anche una piccola guida utente.

¹<https://www.facebook.com/3D4Amb>

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

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

Risultati promozione sicuramente tale “pubblicizzazione” ha influito nel numero di download dell’applicazione, possiamo notare un notevole picco dopo la data di rilascio:

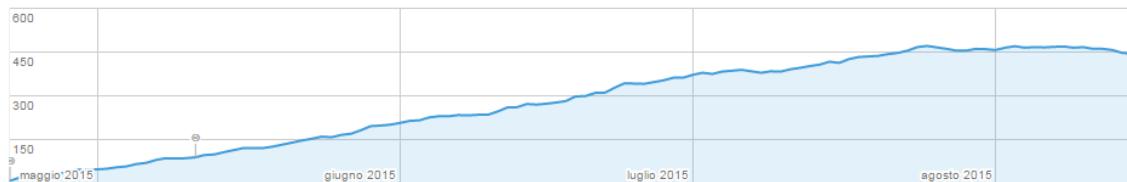


Figura 9.2: statistiche download da Google Play Store

Capitolo 10

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 Goggle 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[3].

Alla base di questo progetto (Car Racing Cardboard), si trovava l’idea di dare continuità al nostro lavoro di tesi triennale “Progetto e sviluppo di un videogioco basato sulla penalizzazione per il trattamento dell’ambliopia”, nel quale gli aspetti fondamentali del lavoro era una penalizzazione dinamica delle informazioni visive fornite al giocatore per rendere il trattamento dell’ambliopia equilibrato con le capacità del giocatore, infatti più l’utente si dimostra capace più viene penalizzato.

Quest’idea di avere una penalizzazione proporzionata alle abilità del giocatore è stata applicata non solo alla quantità di informazioni fornite agli occhi, ma anche alla difficoltà vera e propria del gioco. Infatti, ad ogni ondata di nemici evitati, che equivale ad aver superato un livello, la velocità dei nemici aumenta, in questo modo il gioco viene reso più dinamico, non resta identico fino al gameover, ed inoltre, in base alle prestazioni dell’utente, la difficoltà di gioco aumenta[7].

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



Figura 10.1: Classificazione contenuti

- Australian Classification Board (ACB) – Australia
- Classificação Indicativa (ClassInd) – Brazil
- Entertainment Software Rating Board (ESRB) – North America
- Pan-European Game Information (PEGI) – Europe
- Unterhaltungssoftware Selbstkontrolle (USK) – Germany
- IARC Generic – Rest of the world
- Google Play – South Korea

Questa certificazioni sul Google Play Store, insieme al numero di download, contribuiscono a dare credibilità alla nostra applicazione.

10.1 Paper

Riguardo l'applicazione Car Racing Cardboard è stato realizzato un paper dal titolo:
A low-cost virtual reality game for amblyopia rehabilitation, ed è stato sottoposto alla commissione della conferenza rehab 2015¹, ed è stato accettato:

1 Dear Angelo Gargantini ,
2
3 We have received the reviewers ' reports on your paper 25
entitled
4 A Low -cost Virtual Reality Game for Amblyopia
Rehabilitation and I am glad to inform you that your
paper has been classified as Accepted.
5 For your guidance , reviewers ' comments are appended below
. I hope these comments
6 can help you to improve the contribution .
7
8 Please note that the deadline for the camera ready is the
next September 13th , 2015
9 the process to register will be available next week in
the workshop web page
10 [<http://www.rehab-workshop.org>]
11
12 Thank you for your interest in the REHAB 2015 workshop .
13
14 Yours sincerely ,
15 REHAB 2015 co-chairs

Di seguito riportato il contenuto del nostro elaborato, redatto in collaborazione con il Professor Angelo Gargantini e l' Ing. Silvia Bonfanti:

¹<http://www.rehab-workshop.org/>

A Low-cost Virtual Reality Game for Amblyopia Rehabilitation

Angelo Gargantini, Fabio Terzi, Matteo Zambelli, Silvia Bonfanti

Department of Engineering, University of Bergamo

Viale Marconi 5 24044

Dalmine (BG), ITALY

{angelo.gargantini,silvia.bonfanti}@unibg.it, {f.terzi1,m.zambelli3}@studenti.unibg.it

ABSTRACT

The paper presents the design and development of a mobile application realizing a video game that aims at treating amblyopia by using a Google Cardboard. Google Cardboard is a low cost device able to reproduce virtual reality by means of a smartphone. The proposed video game engaged the patient in a car racing game and it displays the same image to the eyes, but with some differences that stimulate the lazy eye more than the normal eye.

1. INTRODUCTION

Amblyopia, otherwise known as ‘lazy eye’, is reduced visual acuity that results in poor or indistinct vision in one eye that is otherwise physically normal. It may exist even in the absence of any detectable organic disease. Amblyopia is generally associated with a squint or unequal lenses in the prescription spectacles. This low vision is not correctable (or only partially) by glasses or contact lenses. Amblyopia is caused by media opacity, strabismus, anisometropia, and significant refractive errors, such as high astigmatism, hyperopia, or myopia. This condition affects 2-3% of the population, which equates to conservatively around 10 million people under the age of 8 years worldwide [8]. If amblyopia is not diagnosed and treated in the first years of life, the lazy eye becomes weaker and the normal eye becomes dominant. The traditional way to treat amblyopia is carried out wearing a patch over the normal eye for several hours a day, through a treatment period of several months. This treatment has some drawbacks: it is unpopular, not well accepted by the young patients, and sometimes can disrupt the residual fusion between the eyes.

Our group has been involved in the use of computer technologies for the treatment of amblyopia for several years. The project 3D4AMB¹ exploits the stereoscopic 3D technology, that through glasses with LCD active shutters permits to show different images to the amblyotic eye and the nor-

mal eye. We developed some software both for amblyopia diagnosis [3] and treatment that uses this kind of 3D technology [6]. A form of treatment we have proposed, consists in watching video clips with 3D glasses that realize a virtual visual rebalancing [2]. In this work we plan to advance w.r.t. the existing treatments by using a much cheaper virtual reality device and to emphasize the activity to be performed by the patients. Indeed, while patching and vision rebalancing are classified as passive method, other treatments which require some activity on the part of the patients are classified as *active*. Active methods are intended to enhance treatment of amblyopia in a number of ways, including increased compliance and attention during the treatment periods (due to activities that are interesting for the patient) and the use of stimuli designed to activate and to encourage connectivity between certain cortical cell types. A good survey and assessment about active treatments and their efficiency can be found in [5].

The active treatment proposed by 3D4AMB consists in playing with interactive games or exercises, which will stream binocular images. In this settings, the child plays with a special video game which will exploit the binocular vision to send to the lazy eye all the details while the normal eye will see only a part of the game scene. To successfully complete the game or the exercise the patient must use the information shown to the lazy eye (and possibly fuse it with the information shown to the normal eye). In this way, the amblyotic eye is more stimulated and the fusion encouraged. The game application must continuously monitor the success rate of the game in order to adjust the difficulty based on the real capability of the player. It is well known that video games can be very useful for visual rehabilitation [1]. Classical examples of games found in literature, include PAC-MAN and simple car racing games [7]. In this work we propose a simple car racing game.

The classic use of a 3D system (like 3D glasses or the Google Cardboard) is to provide different images to the two eyes of the same scene with viewing angles slightly out of phase, that correspond to the different points of view of left and right eye. This vision produces an illusion of depth of the scene and is the heart of virtual reality. The primary principle of the system is that the images shown to the two eyes are different but related.

This principle can be used in practice also for the treatment of amblyopia by sending two different images to the two eyes: to the amblyopic eye we will show the most interesting part of the frame of the clip (or game), while to the not amblyopic eye (or good) we will show the least interest-

¹<http://3d4amb.unibg.it/>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Rehab2015

Copyright 2006 ACM 1-59593-256-9/06/0006 ...\$5.00.

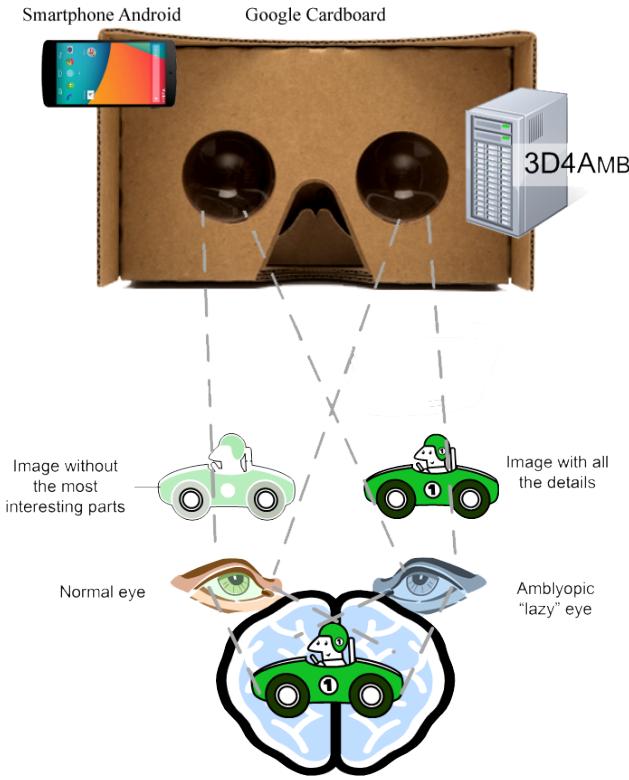


Figure 1: 3D4Amb system with Google Cardboard

ing part. The principle, in case of use of the Google Cardboard is depicted in Fig. 1. Since the patients are young children, we decided to implement the diagnosis and treatment modules in a form of videogame, in order to make the treatment fun and not boring. The final aim of the project is to give the patients a complete system for the treatment that can be used at home. In fact, a smartphone and an inexpensive Google Cardboard are enough to run the software presented in this paper.

In this paper, first in Sect. 2, we describe the Google Cardboards and how they work. Then In Sect. 3, we introduce our mobile application that realizes the video game for amblyopia treatment.

2. GOOGLE CARDBOARD

Google Cardboard is a virtual reality (VR) platform developed by Google. It was created by David Coz and Damien Henry, Google engineers at the Google Cultural Institute in Paris, in their 20% “Innovation Time Off”[4], and introduced at the Google I/O 2014 developers conference for Android devices. It consists of a fold-out cardboard with two lenses where the user must insert the smartphone (see Fig. 2). The user looks inside in order to see the images displayed by the phone. It permits a stereo vision by sending two different images to the two eyes. It works with different smartphones and can be easily adapted to be used by children. The system proposed in this paper also works with other types of VR viewers (e.g., Samsung Gear VR).

It is not an experience as the strap-on Oculus Rift headset, which requires a computer (and is still in development), or Samsungs Gear VR, which costs \$200 and only works with



Figure 2: Google Cardboard

the Galaxy Note 4. But it is an easy way to get a feel for what is possible with modern virtual reality, and beyond the low cost of the headset, most of the available apps are free.

3. CAR RACING MOBILE APPLICATION FOR AMBLYOPIA TREATMENT

The principle of using 3D for penalization of the normal eye in amblyopic children, as explained before, has been applied in the game development for the treatment of amblyopia. The game is called *Car Racing Cardboard (CRC)* which works for the Google Cardboard and it is freely available on the Google play store².

Required hardware. The user must have an Android Smartphone (with the CRC application installed), Google Cardboard, or any other 3D VR glasses and earphone with controller (+, -, and confirm) to play with the game.

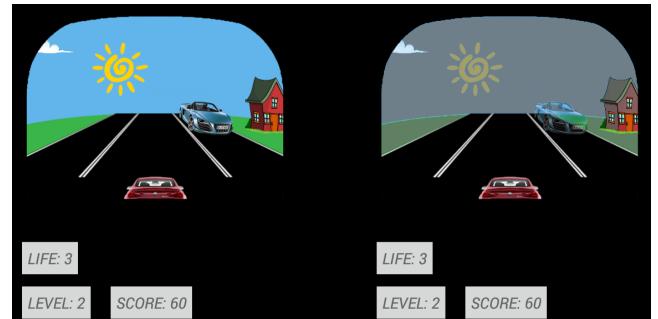


Figure 3: A simple game scene

Game principle. As shown in Fig. 3, the game scene shown to the patient is divided by CRC in two parts, one for the healthy eye (right eye in the figure) and one for the amblyopic eye (left eye in the figure). The CRC decides which images send to the eyes depending on the type of treatment suggested by the doctor. In any case the lazy eye of the child is stimulated to work and the healthy eye is not covered. This is a positive aspect since the children does not interrupt the merger between the eyes (it is a problem with

²<https://play.google.com/store/apps/details?id=it.unibg.p3d4amb.carracingcardboard>

the occlusion treatment, which can disrupt the residual fusion).

The brain of the patient has to combine the two images to view the complete frame successfully and perform simple operations like identifying the incoming cars and move the main car. There are a significant number of elements common to both images, to make sure that the patient can merge them. The final frame is a two-dimensional representation since the objective is not to stimulate the stereo vision of the patient (at least initially), but to make the eyes working in different way.

Game description. Before the beginning of game, the application allows the user to choose the lazy eye (left or right), in order to decide between two different views (penalise the right or left eye). The goal of the game consists in getting the highest score possible. The gamer moves the main car (in the bottom of the view) in order to avoid obstacles (i.e. incoming cars in the opposite direction), and if it does not hit any obstacle, then the score increases. When the score reaches some given thresholds (number of cars avoided), the level increases. When the level rises, the speed and the number of obstacles are increased as well so making the game more difficult. However, the most important aspect is given by the dynamic penalization of the scene shown to the healthy eye. In fact, when the level rises, the application increases the transparency of the panorama and the obstacles displayed for the not amblyopic eye, in order to train the lazy eye. The game ends after 3 collisions (after 3 lives lost). The gamer uses earphone to play the game. “+” and “-” buttons are used to choose the lazy eye at the beginning of the game and to move the main car in order to avoid obstacles. Instead confirm button is used to start the game or restart it after the end.

CRC improvement tracking service. The CRC game includes a web sever to which it sends data about the game activity. In this way, users can track their improvement. Once registration is complete (going on SIGN UP button (Fig. 4)), users can log in and the CRC system will trace game results. At the end of the game, users can ask to receive an email with a summary of the performance, in this way they can understand if they are getting better. Users can play also without registration, but the CRC system will not trace improvements.

Doctor interface. CRC provide also an interface to the doctors to check patient improvements. Doctors can send an email to the CRC Staff, and they will provide to them a special page (Fig. 5). This functionality allows CRC to reach younger users (1-4 years old), who are also the most affected by amblyopia.

Certifications. Until now CRC application has obtained the following certifications on Google Play Store (Fig. 6) :

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

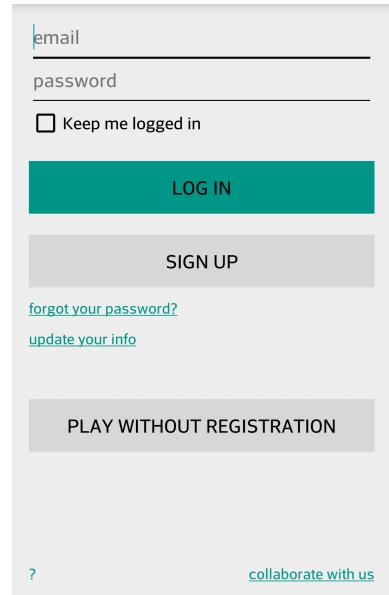


Figure 4: Log in activity

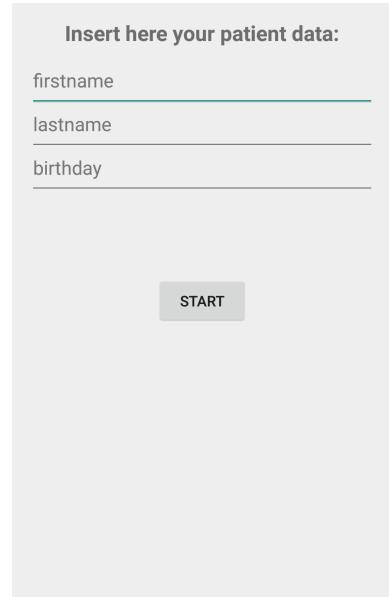


Figure 5: Doctor interface

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

4. CONCLUSIONS

The application's aim is to treat amblyopia or lazy eye, without requiring the use of an expansive device (nowadays all people have a smartphone and Google Cardboard is very cheap). By using the CRC application, people with lazy eye can improve their vision in the lazy eye. The policy for the



Figure 6: CRC Google Play Store certifications

treatment of amblyopia proposed by 3D4AMB, also tries to avoid the classical risks of the patch therapy (poor conformance and fusion disruption) and allows a interactive and supervised healing. However, at least initially, this therapy can be performed alongside with the classical occlusion. Although there are not clinical results available at the moment to support the effectiveness of the application, a series of experiments with children are currently carried on at the local hospital in order to check the validity and viability of the proposed approach.

5. REFERENCES

- [1] R. Achtman, C. Green, and D. Bavelier. Video games as a tool to train visual skills. *Restorative Neurology and Neuroscience*, 26(4-5):435–446, 2008.
- [2] A. Gargantini, M. Bana, and F. Fabiani. Using 3d for rebalancing the visual system of amblyopic children. In *Virtual Rehabilitation (ICVR), 2011 International Conference on*, pages 1 –7, Zurich, june 2011.
- [3] A. Gargantini, G. Facoetti, and A. Vitali. A random dot stereoacuity test based on 3D technology. In *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare - 2nd Patient Rehabilitation Research Techniques Workshop*, PervasiveHealth '14, pages 358–361, ICST, Brussels, Belgium, Belgium, May 2014. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [4] N. Statt. Facebook has oculus, google has cardboard. <http://www.cnet.com/news/facebook-has-oculus-google-has-cardboard/>, 2015. Accessed: 2015-07-10.
- [5] C. M. Suttle. Active treatments for amblyopia: a review of the methods and evidence base. *Clinical and Experimental Optometry*, 2010.
- [6] A. Vitali, G. Facoetti, and A. Gargantini. An environment for contrast-based treatment of amblyopia using 3D technology. In *International Conference on Virtual Rehabilitation 2013 - August 26-29, 2013 in Philadelphia, PA, U.S.A.*, 2013.
- [7] P. E. Waddingham, S. V. Cobb, R. M. Eastgate, and R. M. Gregson. Virtual reality for interactive binocular treatment of amblyopia. In *The Sixth International Conference on Disability, Virtual Reality and Associated Technologies*, 2006.
- [8] A. L. Webber and J. Wood. Amblyopia: prevalence, natural history, functional effects and treatment. *Clinical and Experimental Optometry*, 88(6):365–375, Nov. 2005.

10.2 Sviluppi futuri

Questo progetto non è fine a se stesso, ci sono vari spunti per sviluppi futuri, di seguito eccone alcuni:

- un famiglia di videogiochi mobile per il trattamento dell'ambliopia. Per come è stata pensata l'applicazione sarà sufficiente cambiare le immagini montate sugli "scheletri" che gestiscono la parte grafica del gioco, per quanto riguarda la gestione utenti e dati, le modifiche richieste saranno veramente minime.
In questo modo, sarà possibile rendere ancora più accattivante la rabilitazione, fornendo un ventaglio di soluzioni differenti, per catturare i gusti di tutti i bambini
- una collaborazione attiva con i centri di ipovisione per raccogliere una maggior quantità di dati per migliorare l'efficacia dell'applicazione.
Grazie ad una collaborazione di questo tipo, sarebbe molto più facile per gli sviluppatori correggere eventuali errori, sia di implementazione, ma anche di logica, ad esempio una penalizzazione troppo repentina o troppo poco.
- attuare i feedback provenienti dalla conferenza REHAB 2015, dove verrà presentata questa applicazione

Bibliografia

- [1] F. Collini. <http://android.devapp.it/>, Luglio 2011.
- [2] L. Ferro. Smartphone, ad android [...]. *Il Fatto Quotidiano*, 18 agosto 2014.
- [3] A. Gargantini, F. Terzi, M. Zambelli, and S. Bonfanti. A low-cost virtual reality game for amblyopia rehabilitation. 2015.
- [4] S. D. Modica. appunti a cura di salvatore di modica. *Università di Pisa*.
- [5] A. official site. <http://developer.android.com/>.
- [6] N. Statt. Facebook has oculus, google has cardboard. 2014.
- [7] F. Terzi and M. Zambelli. Progetto e sviluppo di un videogame basato sulla penalizzazione per il trattamento dell’ambliopia. 2013.