

Università degli Studi di Bergamo  
Facoltà di Ingegneria



# **Riconoscimento comandi vocali per Videogiochi CardBoard**

**Autore**

Coter Daniele Matr. 1010321

**Corso di Laurea**

Ingegneria Informatica

**Relatore**

Prof. Angelo Gargantini



# INDICE:

<b><u>1. INTRODUZIONE .....</u></b>	<b><u>5</u></b>
1.1. IL PROGETTO 3D4AMB .....	5
1.2. OBIETTIVO .....	6
<b><u>2. AMBLIOPIA.....</u></b>	<b><u>7</u></b>
2.1. GENERALITÀ.....	7
2.2. CAUSE .....	7
2.3. SINTOMI .....	8
2.4. TRATTAMENTO .....	9
2.4.1. OCCLUSIONE.....	9
2.4.2. PENALIZZAZIONE OTTICA.....	9
2.4.3. PENALIZZAZIONE FARMACOLOGICA .....	10
2.4.4. SETTORIZZAZIONE .....	10
2.5. L'IDEA DI 3D4AMB.....	10
<b><u>3. REALTÀ VIRTUALE .....</u></b>	<b><u>12</u></b>
3.1. CARATTERISTICHE .....	12
3.2. PERIFERICHE.....	13
3.2.1. VISORI .....	13
3.2.2. WIRED GLOVES.....	13
3.2.3. CYBERTUTA.....	13
<b><u>4. VISORI .....</u></b>	<b><u>14</u></b>
4.1. GOOGLE CARDBOARD .....	14
4.1.1. COME FUNZIONA GOOGLE CARDBOARD.....	15
4.2. SAMSUNG GEAR VR .....	16
4.3. CARL ZEISS VR ONE .....	17
4.4. ALTRI VISORI .....	17
<b><u>5. IL 3D .....</u></b>	<b><u>18</u></b>
5.1. ANAGLIFO VS. SIDE-BY-SIDE .....	20
5.2. IL FUTURO .....	21

---

<b>6.</b>	<b>APPLICAZIONI CARDBOARD .....</b>	<b>22</b>
6.1.	CARDBOARD .....	22
6.2.	VR ROLLERCASTER .....	23
6.3.	VOLVO REALITY .....	24
6.4.	VANGUARD V .....	24
<b>7.</b>	<b>CAR RACING CARDBOARD .....</b>	<b>25</b>
7.1.	TRATTAMENTO TRAMITE IL GIOCO .....	25
7.2.	IL GIOCO .....	26
7.2.1.	COMANDI .....	27
7.2.2.	OBIETTIVO .....	27
7.2.3.	VISUALIZZAZIONE .....	28
7.2.4.	REGISTRAZIONE ED INVIO RISULTATI .....	29
<b>8.</b>	<b>RICONOSCIMENTO VOCALE .....</b>	<b>30</b>
8.1.	RICONOSCIMENTO AUTOMATICO DEL PARLATO .....	30
8.2.	SPEECH RECOGNIZER .....	32
8.3.	SPHINX .....	32
<b>9.</b>	<b>IMPLEMENTAZIONE IN CAR RACING CARDBOARD .....</b>	<b>36</b>
9.1.	IMPORTAZIONE LIBRERIE .....	36
9.2.	IMPLEMENTAZIONE .....	37
9.3.	SCHERMATA SELEZIONE COMANDI .....	40
<b>10.</b>	<b>CONCLUSIONE E RINGRAZIAMENTI .....</b>	<b>41</b>
	<b>BIBLIOGRAFIA.....</b>	<b>43</b>



---

# 1.Introduzione

3D4Amb 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'implementazione per 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. Il progetto punta a sviluppare una tecnologia per consentire una facile diagnosi dell'ambliopia e il suo trattamento, per mezzo di giochi interattivi e attività di intrattenimento. La soluzione proposta mira ad eliminare i problemi del trattamento classico dell'occlusione, è adatta 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 sviluppare un sistema per la diagnosi e il trattamento dell'ambliopia, basato sulla visione binoculare in modo accessibile. Con il termine accessibile si intende: poco costoso, 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 e, allo stesso tempo, sottoporlo al trattamento per la cura della sua malattia.

## **1.2. Obiettivo**

L'obiettivo della tesi è il miglioramento dell'applicazione Android Car Racing Cardboard.

Nell'applicazione originale i comandi vengono dati attraverso i pulsanti presenti sulle auricolari in dotazione a tutti gli smartphone (UP, DOWN, PLAY).

Per rendere l'applicazione più semplice e comoda da utilizzare, si è pensato di implementare dei comandi vocali, in modo tale che il paziente possa comandare il gioco attraverso l'uso della sua voce, senza quindi bisogno di periferiche esterne.

---

## 2.Ambliopia

### 2.1. Generalità

L'ambliopia è una condizione caratterizzata dall'indebolimento della vista in un occhio. Il disturbo, noto anche come "occhio pigro", è la principale causa di disabilità visiva tra i bambini.

L'ambliopia deriva da un non corretto sviluppo visivo e neuronale. L'occhio pigro si manifesta durante i primi anni di vita e, a meno che non venga trattato con successo nella fase di sviluppo del bambino, può persistere fino all'età adulta. L'ambliopia, infatti, è la più comune causa di disabilità visiva monoculare (cioè ad un solo occhio) tra giovani e adulti di mezza età. Se trascurata, questa condizione può causare la perdita permanente della vista.

### 2.2. Cause

Ambliopia è il termine medico utilizzato per indicare una ridotta visione in uno degli occhi, poiché l'occhio e il cervello elaborano in modo diseguale od anomalo l'input visivo. Questa condizione si verifica quando le vie nervose tra il cervello e l'occhio non sono adeguatamente stimulate.

L'ambliopia può essere causata da qualsiasi condizione che sconvolge il normale sviluppo visivo o l'uso degli occhi, tra cui lo strabismo (quando gli occhi sono disallineati e non guardano nella stessa direzione) oppure la differenza nella qualità di visione tra gli occhi (per esempio, se un occhio è più miope, presbite o astigmatico dell'altro). Talvolta l'ambliopia è causata da altre patologie oculari, come la cataratta.

Il cervello e l'occhio lavorano insieme per analizzare ed elaborare le informazioni visive. La luce entra nell'occhio, dove la retina traduce l'immagine in segnali nervosi



che vengono inviati, grazie alle vie ottiche, al cervello. Quest'ultimo combina gli stimoli visivi provenienti da ciascun occhio in un'immagine tridimensionale.

I bambini devono imparare a vedere, o più specificamente, il loro cervello deve apprendere come interpretare i segnali nervosi che vengono inviati dagli occhi tramite le vie ottiche. Sono necessari circa 3-5 anni prima che i bambini possano vedere chiaramente come gli adulti e fino a 7 anni prima che il sistema visivo si sviluppi completamente. Se un deficit colpisce uno degli occhi durante la crescita, la qualità dei segnali diventa perturbata e questa, a sua volta, influenza l'interpretazione delle immagini. Ciò significa che il bambino può vedere meno chiaramente da un occhio e tende ad affidare la propria visione all'altro. Spesso, le strutture dell'occhio ambliope appaiono sane e funzionali, ma non sono utilizzate correttamente, in quanto il cervello sta favorendo l'altro occhio (detto dominante). Come risultato, il cervello fa sempre più affidamento sulla parte dominante ed inizia ad ignorare i segnali ricevuti dall'occhio ambliope.

## **2.3. Sintomi**

Il risultato finale di tutte le forme di ambliopia è un deficit dell'acuità visiva nell'occhio ambliope; l'entità di tale difetto può essere lieve o grave.

Per capire se un bambino ha un occhio pigro è talvolta necessario interpretare alcuni segnali; in genere, infatti, i bambini più piccoli non realizzano che qualcosa non va nella loro visione, oppure ne sono consapevoli ma non sono in grado di spiegare il loro disagio. I bambini più grandi possono lamentare l'incapacità di vedere bene con un occhio ed evidenziare problemi con la lettura, la scrittura ed il disegno. Talvolta, gli unici sintomi evidenti di un occhio pigro sono correlati ad una condizione di base, come ad esempio lo strabismo, la cataratta congenita o la ptosi palpebrale.

Segni e sintomi dell'ambliopia possono comprendere:

Difficoltà di visione in un occhio;

Movimento involontario di un occhio verso l'interno o verso l'esterno: in alcuni casi si può notare che un occhio guarda in una direzione diversa rispetto all'altro (questo potrebbe essere dovuto ad uno strabismo);

Bassa sensibilità al contrasto;

---

Bassa sensibilità al movimento;

Scarsa percezione della profondità: i bambini con un occhio pigro, di solito, hanno problemi a giudicare con precisione la distanza tra sé e gli oggetti. Questo può rendere alcune attività più difficili, come prendere una palla.

L'ambliopia è generalmente monolaterale, ma non è escluso che possa insorgere in entrambi gli occhi.

## **2.4. Trattamento**

Il trattamento precoce dell'ambliopia è fondamentale per ottenere buoni risultati, la correzione oculare avviene in diversi modi, che verranno descritti nei prossimi paragrafi:

### *2.4.1. 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 in proporzione all'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.

### *2.4.2. Penalizzazione Ottica*

Questo trattamento si attua 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 per l'occlusione parziale, viene attuata per coloro che necessitano di una occlusione “morbida” o come terapia di mantenimento

#### *2.4.3. Penalizzazione farmacologica*

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

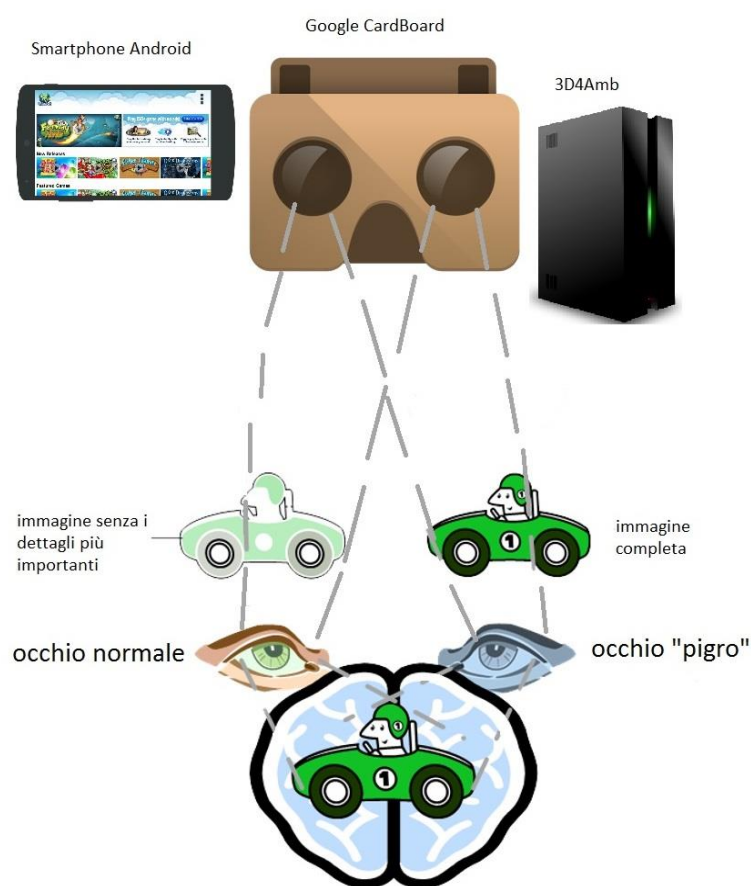
#### *2.4.4. Settorizzazione*

Consiste nella copertura di una 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.

### **2.5. 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 1) e una per l'occhio sinistro (occhio buono, Figura 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 senza escludere l'occhio sano; in questo modo il paziente continua il processo di fusione tra l'occhio destro e sinistro. 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.



**Figura 1 – il progetto di 3D4Amb**

## **3.Realtà Virtuale**

La realtà virtuale (in inglese virtual reality, abbreviato VR) è il termine utilizzato per indicare una realtà simulata.

### **3.1. Caratteristiche**

La realtà virtuale, per sua stessa definizione, simula la realtà effettiva. L'avanzamento delle tecnologie informatiche permette di navigare in ambientazioni fotorealistiche in tempo reale, interagendo con gli oggetti presenti in esse.

Anche se, a livello teorico, la realtà virtuale potrebbe essere costituita attraverso un sistema totalmente immersivo in cui tutti i sensi umani possono essere utilizzati (più specificamente realtà virtuale immersiva o RVI), attualmente il termine è applicato solitamente a qualsiasi tipo di simulazione virtuale creata attraverso l'uso del computer, dai videogiochi che vengono visualizzati su un normale schermo, alle applicazioni che richiedono l'uso degli appositi guanti muniti di sensori (wired gloves) e infine al World Wide Web.

---

## 3.2. Periferiche

La realtà virtuale immersiva (un ambiente costruito intorno all'utente) secondo il livello tecnologico attuale e secondo le previsioni possibili per il prossimo futuro potrà essere utilizzata dalla massa grazie ad alcune periferiche (in parte già utilizzate):

### 3.2.1. *Visori*

Un casco o dei semplici occhiali in cui gli schermi vicini agli occhi annullano il mondo reale dalla visuale dell'utente. Il visore può inoltre contenere dei sistemi per la rilevazione dei movimenti, in modo che girando la testa da un lato, ad esempio, si ottenga la stessa azione anche nell'ambiente virtuale.

auricolari - trasferiscono i suoni all'utente.

### 3.2.2. *Wired gloves*

I guanti rimpiazzano mouse, tastiera, joystick, trackball e gli altri sistemi manuali di input. Possono essere utilizzati per i movimenti, per impartire comandi, digitare su tastiere virtuali, ecc.

### 3.2.3. *Cybertuta*

Una tuta che avvolge il corpo. Può avere molteplici utilizzi: può simulare il tatto flettendo su sé stessa grazie al tessuto elastico, può realizzare una scansione tridimensionale del corpo dell'utente e trasferirla nell'ambiente virtuale.

La differenza con una realtà virtuale non immersiva consiste nel fatto che in quest'ultimo caso non si fa uso di caschi, ma l'utente si troverà semplicemente dinanzi ad un monitor, il quale fungerà da finestra sul mondo tridimensionale con cui l'utente potrà interagire attraverso joystick appositi. È chiaro che l'effetto che ne deriva è assolutamente diverso da quello che si ottiene con una realtà virtuale immersiva, in cui gli effetti che l'utente percepisce sono molto più coinvolgenti e capaci di distrarre del tutto dalla realtà.

## 4. Visori

In questo elaborato faremo particolare attenzione ai visori per la realtà virtuale, essendo utilizzati nell'applicazione Car Racing Cardboard.

Sul mercato, ad oggi, è presente una grande quantità di visori per tutte le fasce di prezzo.

### 4.1. Google CardBoard

A ideare e lanciare i visori VR low-cost è stata Google nella seconda metà del 2014. Per incentivare la nascita di start up attive nel settore della realtà virtuale e favorire gli sviluppatori a lavoro su applicazioni VR, il gigante di Mountain View ha voluto rilasciare un visore duttile e fai-da-te. Nel corso del Google I/O 2015, Big G ha presentato la seconda versione del suo visore VR low-cost: si tratta di un *wearable device* più raffinato nelle forme e compatibile con un maggior numero di smartphone.



Figura 2 – Google CardBoard

---

#### *4.1.1. Come Funziona Google CardBoard*

Per poter immergersi nella realtà virtuale low-cost di Google è necessario, prima di tutto, acquistare un kit per visore VR. Dal portale creato da Big G per il progetto cardboard è possibile scegliere i modelli più economici in cartone da autoassemblare, oppure modelli in plastica – più resistenti e con rifiniture migliori – già assemblati. Scelto il visore VR più adatto alle proprie necessità (e alle proprie finanze), si dovrà lanciare l'app cardboard e inserire lo smartphone compatibile con cardboard nella "tasca" apposita. Terminata la prima sincronizzazione, il telefonino sarà riconosciuto automaticamente dal visore per realtà virtuale e l'app cardboard partirà in automatico non appena il dispositivo mobile sarà alloggiato nella tasca.



**Figura 3 – Utilizzo Google CardBoard**



## 4.2. Samsung gear VR

Samsung Gear VR è visore per realtà virtuale sviluppata da Samsung Electronics in collaborazione con Oculus VR. A Differenza di Google Cardboard, questo visore è compatibile solo con Smartphone Samsung (Galaxy Note 5, Galaxy S6 / S6 Edge o Galaxy S7 / S7 edge). Lo Smartphone viene collegato al visore tramite micro – USB per sfruttare il sensore di rotazione presente nel Samsung gear VR. Questo sensore è più preciso e ben calibrato con una latenza inferiore rispetto a quello interno allo smartphone (Utilizzato ad esempio nel CardBoard di Google).

L'auricolare Gear VR include anche un touchpad e pulsante Indietro sul lato, così come un sensore di prossimità per rilevare quando l'auricolare è acceso. Il touchpad e il pulsante consentono agli utenti di interagire con la realtà virtuale, mentre i dispositivi di Google presentano solo un pulsante. Il prezzo di questo visore è molto più elevato del dispositivo Google si aggira sui 199 Euro.



Figura 4 – Samsung gear VR

---

### 4.3. Carl Zeiss VR One

Il Visore per realtà virtuale Carl Zeiss VR One è stato sviluppato dal famoso produttore di lenti e sistemi optoelettronici Carl Zeiss. Molto simile al visore Samsung sopra citato, ha però il pregio di essere compatibile con tutti i modelli di smartphone con display compresi nel range 4.7-5.2 pollici. Lo smartphone deve essere agiato su di un vassoio (adattato a seconda del modello) e poi inserito nel visore. Sul visore sono disponibili i connettori di ricarica e l'uscita audio, mentre i sensori di movimento vengono gestiti totalmente dallo smartphone. Il prezzo è decisamente più basso rispetto a gear VR della Samsung e si aggira sui 100 euro.



Figura 5 – Zeiss One VR

### 4.4. Altri visori

Se non si è disposti a spendere grosse cifre, ma si vuole avere qualcosa di più performante e comodo da usare di un visore di cartone, è possibile acquistare visori per realtà virtuale a prezzi bassi e di buona qualità.

Infatti è possibile trovare su molti siti di e-commerce (Ebay, Amazon, Aliexpress, ...) visori di buona fattura ad un prezzo modico (15-30 euro)

La maggior parte di questi visori si adattano a tutti i tipi di smartphone.

## 5. Il 3D

L'acronimo *3D* sta letteralmente per "tre dimensioni" e riguarda la visione stereoscopica.

La stereoscopia è una tecnica di realizzazione e visione di immagini, disegni, fotografie e filmati, atta a trasmettere un'illusione di tridimensionalità, analoga a quella generata dalla visione binoculare del sistema visivo umano.

La visione tridimensionale è ottenuta tramite l'utilizzo di due immagini parallele scostate orizzontalmente di pochi centimetri l'una dall'altra e osservabili tramite appositi occhialini che danno allo spettatore il senso di tridimensionalità. La scelta dello scostamento orizzontale delle due immagini influisce sulla distanza dell'oggetto percepita dallo spettatore.

1. Quando le due immagini, quella dedicata all'occhio sinistro e quella dedicata all'occhio destro, coincidono si vede l'oggetto esattamente posizionato sullo schermo. Questo è il caso che non sfrutta la tecnologia 3D ed è riconducibile alla visione su normali schermi televisivi.

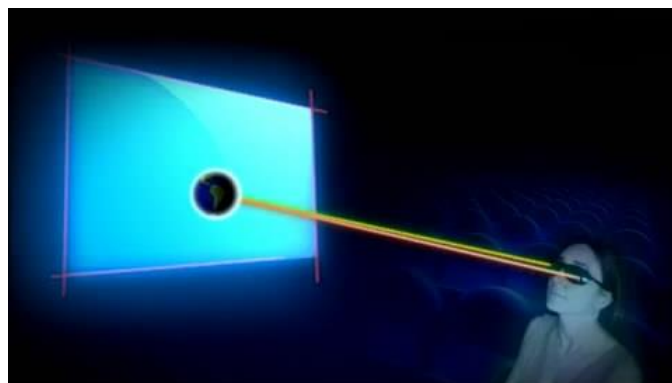
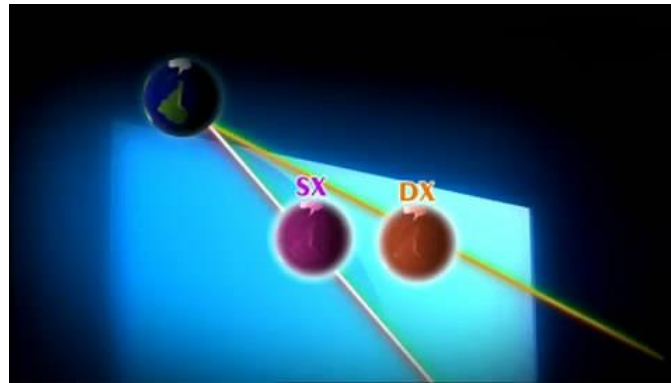


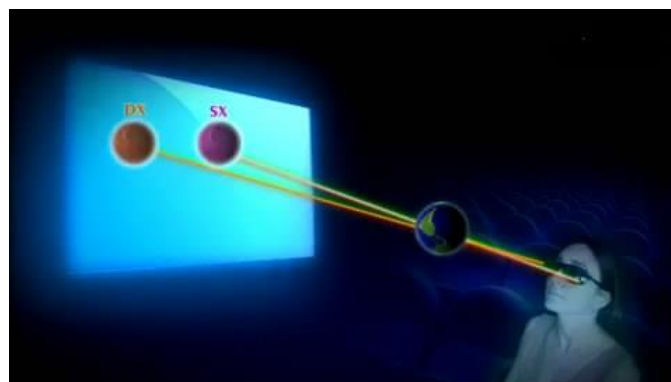
Figura 6 – assenza 3D

- 
2. Quando l'immagine per l'occhio destro e quella per l'occhio sinistro vengono distanziate rimanendo però ognuna nel rispettivo campo visivo sia dell'occhio destro che di quello sinistro, l'immagine viene percepita "in profondità" rispetto alla finestra definita dallo schermo.



**Figura 7 – profondità**

3. Quando l'immagine per l'occhio sinistro va a posizionarsi nel campo visivo dell'occhio destro e parallelamente quando l'immagine dell'occhio destro viene posta nel campo visivo dell'occhio sinistro, gli oggetti vengono visti più vicini dallo spettatore.



**Figura 8 - vicinanza**

## 5.1. Anaglifo VS. Side-by-side

Esistono due metodi per applicare la visione stereoscopica appena analizzata: l'**Anaglifo** e il **Side-By-Side**.

Il primo consiste in un'immagine o una serie di immagini che possono essere viste utilizzando occhialini con lenti di colore Rosso/Blu (oppure Rosso/Verde). Queste lenti assegnano una porzione ben specifica dello spettro a ciascun occhio, porzione stabilita in fase di preparazione dell'anaglifo.

Il rosso, il verde e il blu costituiscono la terna dello spazio colore RGB (Red, Green, Blue):

- R+G (Rosso e Verde) restituisce il Giallo
- R+B (Rosso e Blu) restituisce il Magenta
- G+B (Verde e Blu) restituisce il Ciano.

È possibile produrre anaglifi usando una delle tre combinazioni, ma la più utilizzata è Rosso (per l'occhio sinistro) e Ciano (per l'occhio destro) per le caratteristiche intrinseche dell'occhio.

La posizione del colore rosso dell'immagine modifica la percezione di profondità dell'anaglifo:

- Un rosso spostato verso sinistra dà la sensazione che l'oggetto rientri
- Un rosso spostato verso destra dà la sensazione che l'oggetto sporga

Il principale difetto dell'anaglifo consiste nella scarsa fedeltà cromatica offerta, soprattutto in presenza di colori saturi. I migliori risultati si ottengono con immagini in bianco e nero. La facilità con la quale tale stereoscopia è sfruttata, permette a qualsiasi TV o display di riprodurre un video anaglifico.



Figura 9 – immagine 3D anaglifica

---

Al contrario dell'anaglifo, il formato **Side-by-side** è più complesso da realizzare. Esso è caratterizzato da due video affiancati ripresi in angolazioni differenti, uno destinato all'occhio destro e uno per quello sinistro.

Questa tecnologia è quella usata per l'applicazione Car racing Cardboard e per tutte le applicazioni utilizzabili con i visori di realtà virtuale. Sullo Smartphone vengono visualizzate due immagini, una per l'occhio destro e uno per l'occhio sinistro. Grazie alle lenti presenti nei visori, si ha la sensazione di vedere un'immagine in 3D.

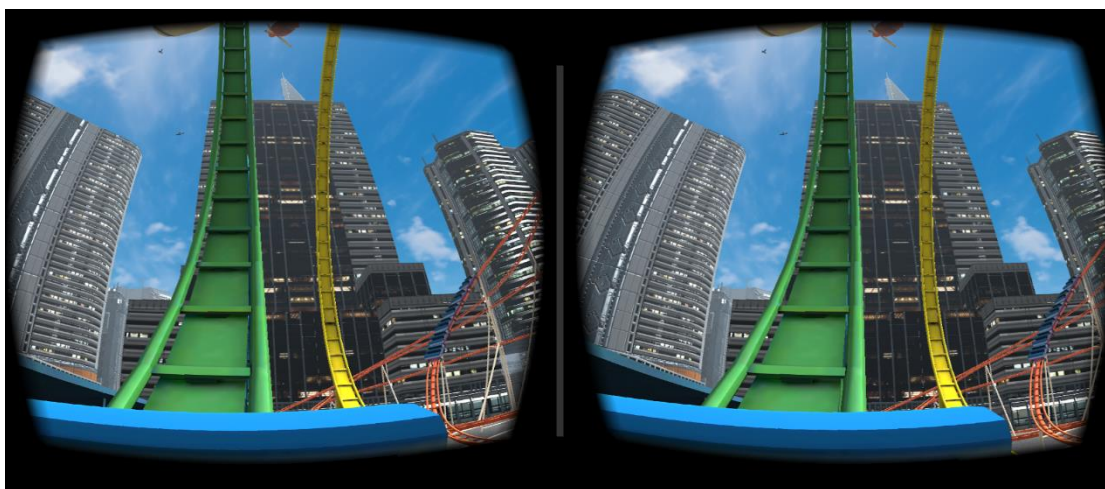


Figura 10 – side by side su smartphone

## 5.2. Il futuro

“Il 3D non ha senso fino a quando non si potrà avere senza occhiali” si è sempre detto, e sappiamo bene che per avere il 3D senza occhiali ad una risoluzione accettabile serve anche un pannello ad altissima risoluzione. Pannelli che ora, dopo l'arrivo del 4K consumer, iniziano ad essere disponibili: sfruttando un pannello 8K da 110”, Samsung ha comunque mostrato al CES 2015 il televisore 3D senza occhiali più grande al mondo, e dobbiamo dire che la qualità di visione, seppur vincolata come sempre a punti di visione ben precisi, è migliorata rispetto ai prototipi senza occhiali degli anni scorsi. Lo stesso filtro, applicato probabilmente ad un televisore da 55” 8K, potrebbe iniziare a dare quella qualità che i TV 3D “glassfree” non hanno mai raggiunto per la bassa risoluzione di visione.

## 6.Applicazioni CardBoard

Sul PlayStore di Google le applicazioni presenti da utilizzare con i visori di realtà virtuali non sono tantissime, ma crescono ogni giorno.

Di seguito verranno trattate le applicazioni migliori presenti ad oggi.

### 6.1. CardBoard

Una delle applicazioni che meglio si adatta ai visori di realtà virtuale è proprio Cardboard. L'applicazione sviluppata dai laboratori dell'azienda di Mountain View, infatti, offre tanti strumenti per impostare al meglio il proprio Google Cardboard. Con Earth sarà possibile visitare tutti i luoghi del mondo, attraversare strade, ponti e "passeggiare" nei vicoli delle maggiori capitali europee, mentre all'interno della sezione I miei video sarà possibile guardare da una prospettiva differente tutti i filmati presenti sullo smartphone.



Figura 11 – app Google CardBoard



---

Grazie alla realtà virtuale sarà molto più semplice scoprire informazioni sui monumenti e sui palazzi storici: all'interno di Cardboard è presente una sezione dedicata alle Visite guidate dove sarà possibile visitare la reggia di Versailles che per anni ha ospitato i re di Francia, mentre in Mostra si potranno ammirare le opere d'arte da qualsiasi angolazione. Infine in Photo Sphere si potranno ammirare le foto panoramiche sfruttando i vantaggi della realtà virtuale.

## **6.2. VR RollerCaster**

L'applicazione Android VR RollerCaster permette all'utente di salire virtualmente a bordo della carrozza di un trenino che partirà a tutta velocità sulle rotaie delle montagne russe.



**Figura 12 – app VR RollerCaster**



### 6.3. Volvo Reality

Negli ultimi mesi anche il settore dell'automotive sta scoprendo le funzionalità offerte dalla realtà virtuale, sviluppando delle applicazioni che offrono all'utente la possibilità di provare in anteprima i nuovi modelli. Volvo ha lanciato sul Google PlayStore Volvo Reality un'applicazione pensata per i visori VR che permette di salire a bordo della Volvo XC90 e di scoprire tutti gli optional offerti dalla casa svedese.



Figura 13 – app Volvo reality

### 6.4. Vanguard V

Vanguard V è uno sparatutto sviluppato dalla software house Proton Pulse e ambientato nello spazio. Il giocatore dovrà viaggiare nello spazio infinito portando a termini le proprie missioni per difendere la propria vita dall'attacco dei nemici.

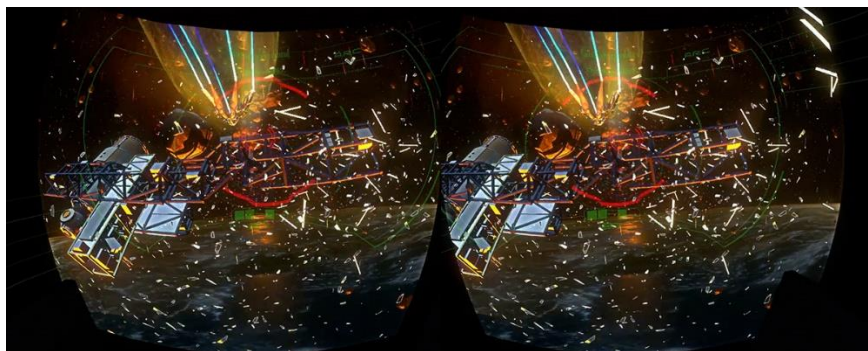


Figura 14 – app Vanguard V

---

## 7. Car Racing CardBoard

L'applicazione su cui verte questo elaborato è Car Racing Cardboard.

L'app in questione è stata realizzata come tesi da due studenti di ingegneria informatica presso l'Università di Bergamo (Fabio Terzi, Matteo Zambelli) avendo come relatore il prof Angelo Gargantini, ed ha come obiettivo il trattamento dell'ambliopia

### 7.1. Trattamento tramite il gioco

L'applicazione punta a sfruttare le potenzialità dei visori CardBoard per correggere il disturbo dovuto all'ambliopia. Fino ad ora, il trattamento standard utilizzato per la cura della malattia è stato l'occlusione visiva. Questa tecnica può però presentare alcuni limiti:

- 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;
- 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.

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 è stato realizzato dai due studenti il gioco interattivo Car racing CardBoard che punta ad una cura dell'occhio pigro in modo più coinvolgente.

Grazie all'accessibilità dei Google Cardboard, all'evoluzione del mondo Android ed all'intrattenimento volontario motivato dato dal gioco, è stato creato un metodo di cura innovativo.

## 7.2. Il Gioco

Il gioco consente all'utente di muovere un veicolo su tre corsie con l'obiettivo di evitare gli ostacoli che si presentano sulle tre corsie.

Gli ostacoli sono rappresentati da altri veicoli che procedono in senso di marcia contrario rispetto al veicolo comandato dall'utente.

Inizialmente il veicolo controllato dall'utente si trova in posizione centrale, è possibile poi spostare tale veicolo a destra e a sinistra.

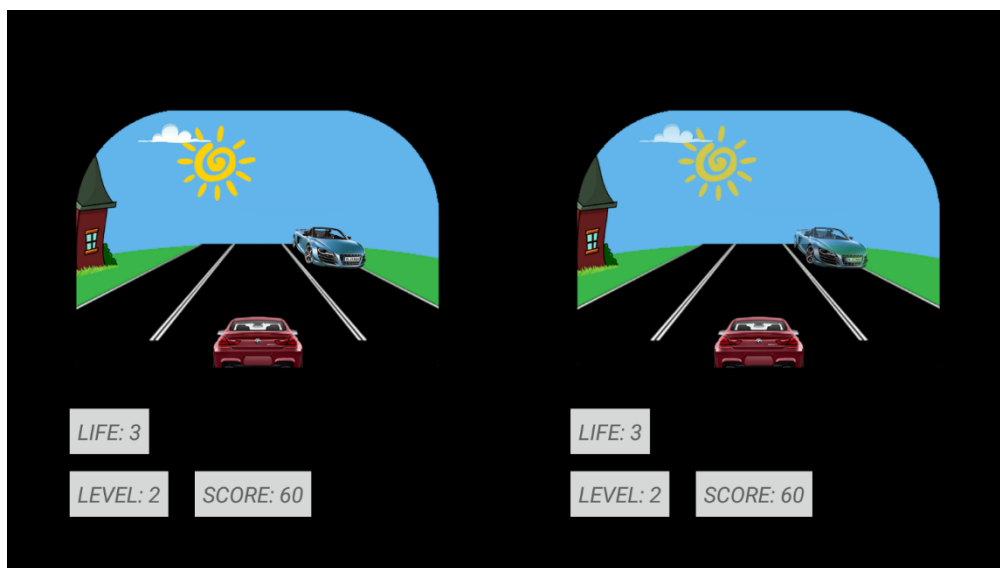
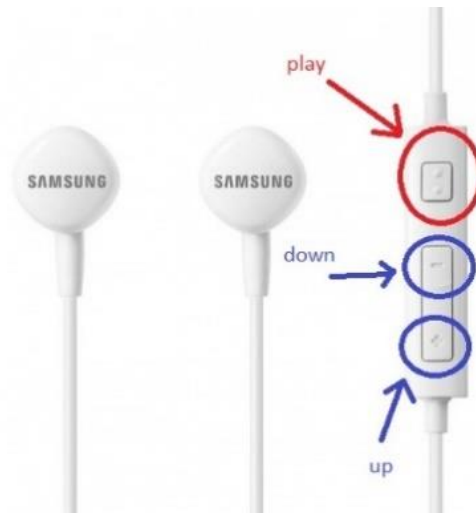


Figura 15 – Car Racing CardBoard

---

### 7.2.1. Comandi

Nell'applicazione originale, l'utente deve utilizzare i pulsanti presenti sugli auricolari (UP, DOWN e PLAY), o in alternativa un dispositivo Bluetooth collegato allo smartphone, per lo spostamento del veicolo.



**Figura 16 – pulsanti auricolari standard**

### 7.2.2. Obiettivo

L'obiettivo del gioco è ottenere il maggior punteggio possibile, il punteggio incrementa ogni volta che l'utente evita un ostacolo.

A disposizione dell'utente, nel momento della partenza del gioco, ci sono 3 vite le quali decremento quando il veicolo effettua una colluttazione con una macchina proveniente dal senso opposto. I 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.

### 7.2.3. Visualizzazione

L'applicazione è stata realizzata renderizzando sullo smartphone dell'utente una doppia Viewport. Questo sdoppiamento, grazie alla tecnologia dei visori Cardboard, permette all'utente una visione stereoscopica del gioco. Lo sdoppiamento è fondamentale per il principio di funzionamento dell'applicazione. Infatti, grazie a questo è possibile un trattamento diverso per l'occhio destro e l'occhio sinistra e, più specificamente l'occhio sano e l'occhio "pigro".

Prima dell'inizio del gioco è presente una schermata in cui poter scegliere l'occhio ambliope.



**Figura 17 – Schermata scelta occhio pigro**

In questo modo l'applicazione tratta l'occhio pigro in modo da svilupparne la visione. In particolare, l'applicazione penalizza 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 del punteggio e dei livelli raggiunti dall'utente.

---

#### 7.2.4. *Registrazione ed invio risultati*

La prima schermata dell'applicazione permette di effettuare il LOG IN, di registrarsi o di giocare senza registrazione.

Se un utente è registrato, una volta terminata la sessione di gioco è possibile richiedere di inviare via mail lo storico con i risultati o in alternativa riavviare il gioco.



**Figura 18 – schermata finale**

## 8. Riconoscimento Vocale

L'obiettivo principale di questo elaborato è implementare nell'applicazione un riconoscimento vocale per permettere all'utente di spostare il veicolo utilizzando solo la voce e quindi rendere l'applicazione hands-free.

### 8.1. Riconoscimento automatico del parlato

Il riconoscimento automatico del parlato (automatic speech recognition) è il processo con cui un computer traduce un segnale sonoro in un testo. Un'ampia varietà di tecniche è usata per il riconoscimento automatico del parlato. Il primo passo è il campionamento digitale del segnale acustico. In questa fase il segnale contenente l'informazione viene trasformato da analogico, formato in cui il segnale viene normalmente registrato o conservato, a digitale, formato più comodo per l'elaborazione. Il passo seguente è l'elaborazione del segnale digitalizzato. La maggior parte delle tecniche prevede in questa fase un'analisi spettrale del segnale. Il passo successivo è il riconoscimento dei fonemi, dei gruppi di fonemi e, finalmente, delle parole. Questa fase può essere realizzata da molti processi quali HMM (Hidden Markov Models), NN (reti neurali), sistemi esperti, e combinazione di più tecniche. I sistemi basati su modelli di Markov sono attualmente i sistemi il più comunemente usati, poiché sono quelli che offrono prestazioni migliori. In questo passo, la maggior parte dei sistemi di riconoscimento, usa informazioni specifiche sui suoni e sulle parole/frasi della lingua che si intende riconoscere (modello acustico e modello del linguaggio rispettivamente). La realizzazione ex novo di un sistema automatico per il riconoscimento del parlato è un compito laborioso, che può richiedere anni di lavoro. Nella prossima sezione analizzeremo le caratteristiche di due soluzioni software già esistenti che ho testato sull'app Car Racing CardBoard per il riconoscimento automatico.

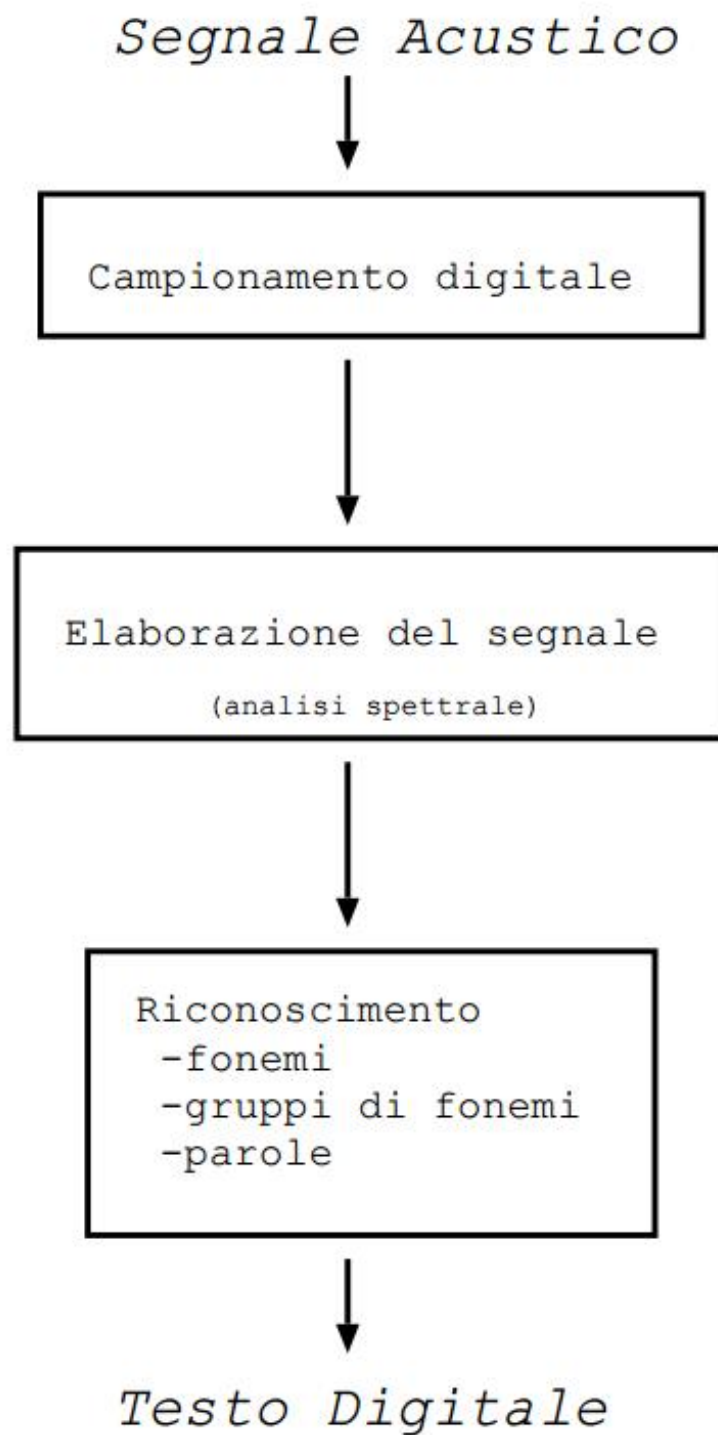


Figura 19 – schema generico riconoscitore vocale



## **8.2. Speech Recognizer**

La prima opzione che ho pensato di utilizzare per l'implementazione dei comandi vocali nell'applicazione Android è stata di utilizzare android speech recognizer.

Questa classe fornisce accesso al servizio di riconoscimento vocale. L'API in questione richiede lo streaming audio a server remoti per eseguire il riconoscimento vocale.

Quindi è necessario avere una connessione ad internet attiva sullo smartphone per poter usufruire di tale servizio, e questo può essere un limite.

Tale API non è destinata ad essere utilizzata per il riconoscimento continuo, in quanto consuma una notevole quantità di batteria e larghezza di banda.

Dopo aver implementato lo Speech Recognizer nell'applicazione Car Racing Cardboard ho effettuato alcuni test per stabilire se fosse l'opzione adatta alle mie esigenze.

Il riconoscimento delle parole è risultato ottimo sia per la lingua inglese che per quella italiana, e anche la reattività ai comandi non è stata da meno.

Il problema che ho notato è che, dopo 5-6 secondi di silenzio in cui non vengono richiamati dei comandi, lo speech recognizer va in errore di time out e da quel momento non riparte più e quindi l'applicazione non è più in grado di ascoltare i comandi vocali.

Dopo aver accertato, attraverso forum e siti internet, che questo è un bug presente nello speech recognizer sono giunto alla conclusione che questa opzione va scartata.

## **8.3. Sphinx**

Sphinx è un progetto sovvenzionato dal DARPA (Agenzia di ricerca militare statunitense) e sviluppato presso la Carnegie Mellon University. Al progetto partecipano anche la Sun Microsystems, Mitsubishi Electric Research Labs, la Hewlett Packard, la University of California at Santa Cruz e il Massachusetts Institute of Technology (MIT). Lo scopo principale del progetto è lo sviluppo di strumenti software riguardanti il trattamento del parlato. La licenza con cui questi software sono rilasciati è una derivazione della licenza BSD (una particolare licenza free): in pratica non ci sono restrizioni sull'uso commerciale o sulla ridistribuzione del software. Sphinx è un riconoscitore vocale scritto interamente in Java, lo rende totalmente cross-platform, cioè indipendente dal sistema operativo.

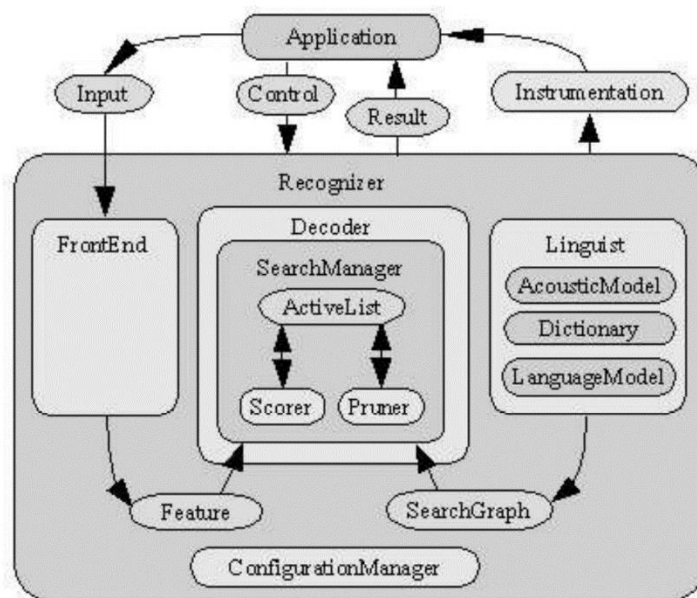
---

L'architettura di Sphinx è estremamente modulare (Figura 19), infatti Sphinx è pensato come uno strumento per la ricerca, ed è quindi progettato per permettere la sperimentazione di nuove soluzioni tecniche. Sphinx è corredato da un'ampia documentazione, che ne permette un apprendimento piuttosto rapido.

Un grosso limite di Sphinx è la mancanza di moduli specifici per l'italiano. Tra gli elementi fondamentali dell'architettura Sphinx c'è il modulo Linguist che contiene tre sotto-moduli:

- **Modello acustico:** Questo modulo contiene l'informazione a priori sulle caratteristiche fisiche dei suoni presenti in una specifica lingua.
- **Dizionario:** Questo modulo contiene l'informazione su quali sono le parole che appartengono ad una specifica lingua.
- **Modello del linguaggio:** Questo modulo contiene l'informazione su quale sia la probabilità che un certo numero di parole siano consecutive in una frase della specifica lingua.

Tra questi moduli il più laborioso da produrre è il modello acustico: infatti esistono software open-source che permettono lo sviluppo di un dizionario e di un modello del linguaggio in un tempo ragionevole. Il progetto Sphinx mette a disposizione un software specifico anche per la creazione di un modello acustico, ma il problema principale è la preparazione dei dati per far funzionare tale software. Occorre infatti preparare un numero notevole di file audio, contenenti frasi in italiano, e corrispondentemente dei file di testo con la trascrizione fonetica di tali frasi.



**Figura 19 – architettura di Sphinx**

Per testare sphinx ho scaricato il demo per Android Studio presente sul sito <http://cmusphinx.sourceforge.net/>

Il demo presenta 3 opzioni che si possono testare:

- Phones: permette di riconoscere i fonemi e di visualizzare un linguaggio fonetico.
- Digits: permettere di riconoscere le parole e di visualizzare.
- Forecast: riconoscitore ottimizzato per il meteo.

Dopo aver provato le opzioni presenti, ho intuito che la più adatta alle mie esigenze era senza dubbio la parte denominata Digits.

A questo punto ho creato una piccola app di prova attraverso Android Studio ottenuta adattando il codice del demo di Sphinx alle mie esigenze, quelle cioè di riconoscere una quantità molto limitata di vocaboli ed eseguire delle azioni nel momento in cui vengono riconosciute.

L'app di prova riconosce semplicemente l'input di tre comandi vocali (Left, Right, Okay) e a seconda del vocabolo visualizza un'immagine a schermo.

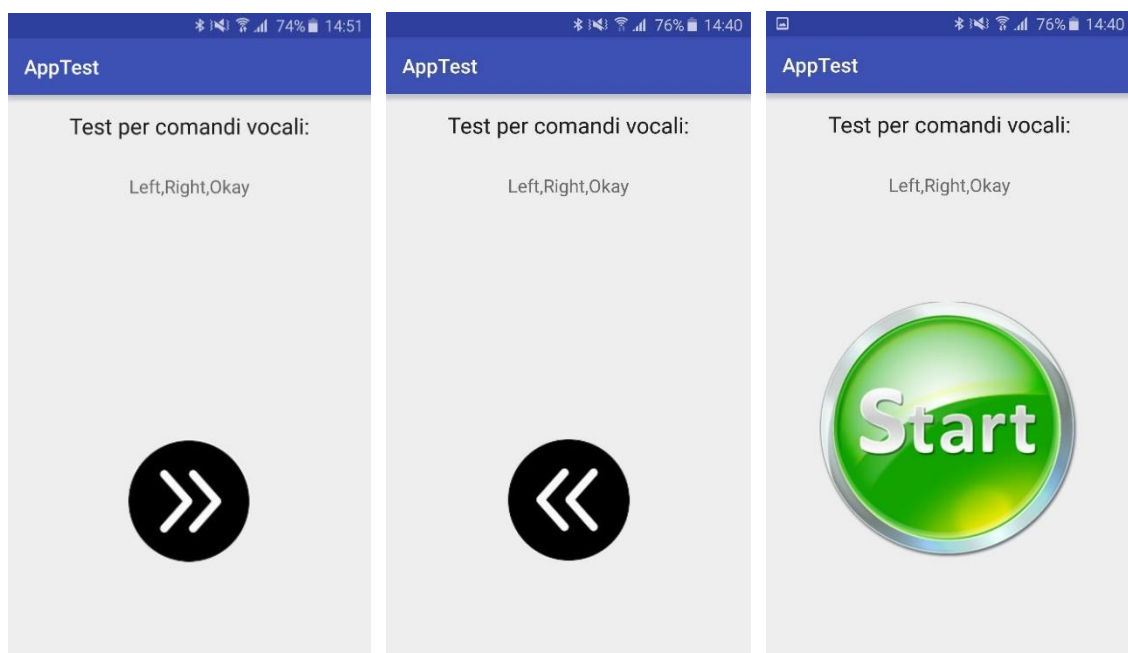
```

@Override
public void onResult(Hypothesis hypothesis) {
    if (hypothesis != null) {
        String text = hypothesis.getHypstr();
        if (text.contains("right"))
            immagine.setImageResource(R.drawable.dx);
        else if (text.contains("left"))
            immagine.setImageResource(R.drawable.sx);
        else if (text.contains("okay"))
            immagine.setImageResource(R.drawable.start);
    }
}
}

```

**Figura 20 – codice per app di prova**

Quindi se viene pronunciata una delle parole “right”, “left”, ”okay” si ottiene rispettivamente:



**Figura 21 – Schermate app di prova(right,left,okay)**

Dopo aver valutato come positiva questa prova, ho deciso di implementare il codice di Sphinx adattato nel gioco Car Racing CardBoard.

# 9.Implementazione in Car Racing CardBoard

## 9.1. Importazione librerie

Dopo aver aperto il progetto Car Racing CardBoard con Android Studio, prima di procedere con l'implementazione del codice, ho seguito il tutorial presente nel sito di CMU Sphinx per poter importare nel progetto tutto il necessario per poter utilizzare il riconoscimento Vocale.

- Importare la libreria `pocketsphinx-android-5prealpha-nolib.jar` nella cartella `app/libs`.
- Aggiungere al file `AndroidManifest.xml` le due seguenti linee di codice:

```
<uses-permission  
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
<uses-permission  
android:name="android.permission.RECORD_AUDIO" />
```

Per abilitare la scrittura su storage esterno e la registrazione audio

- Copiare il file `assets.xml` dal demo nella cartella `app`, ed aggiungere al file `build.gradle` il seguente codice per avviare `assets.xml` :

```
ant.importBuild 'assets.xml'  
preBuild.dependsOn(list, checksum)  
clean.dependsOn(clean_assets)
```

- Infine importare sempre dal demo di Sphinx il modello acustico (`en-us-ptm`) ed il dizionario (`cmudict-en-us.dict`).

---

## 9.2. Implementazione

Le Activity in cui è necessario introdurre i comandi vocali sono due, Settings e Main.

Nella schermata di Settings l'utente deve scegliere l'occhio pigro, nell'app originale questo viene fatto tramite i comandi UP, DOWN che permettono di scorrere tra occhio sinistro e occhio destro, e con il tasto PLAY che permette di confermare la scelta.

Con l'introduzione dei comandi vocali, la scelta può essere effettuata semplicemente pronunciando la parola "Left" o la parola "Right". Una volta effettuata la scelta è possibile confermarla e iniziare il gioco con la parola "Okay".

Nella Main Activity, ovvero nella schermata di gioco, l'obiettivo è comandare, attraverso 3 corsie, un veicolo.

Anche in questo caso, per spostarsi da una corsia all'altra, nell'app originale si utilizzano i tasti UP e DOWN.

Dopo aver aggiunto i comandi vocali, la vettura può essere mossa dall'utente attraverso la pronuncia dei vocali "Left" (se si vuole spostarla a sinistra) e Right (se si vuole spostarla a destra). Pronunciando invece la parola "home" il gioco viene riavviato.

```
import java.io.File;
import java.io.IOException;
import java.util.HashMap;

import edu.cmu.pocketsphinx.Assets;
import edu.cmu.pocketsphinx.Hypothesis;
import edu.cmu.pocketsphinx.RecognitionListener;
import edu.cmu.pocketsphinx.SpeechRecognizer;
import it.unibg.p3d4amb.carracingcarboard.Exception.MyException;
import it.unibg.p3d4amb.carracingcarboard.Manager.Enum.Eye;
import it.unibg.p3d4amb.carracingcarboard.R;

import static edu.cmu.pocketsphinx.SpeechRecognizerSetup.defaultSetup;

public class SettingsActivity extends ActionBarActivity implements RecognitionListener {

    private String id_user;
    private static final String DIGITS_SEARCH = "digits";
    private Eye eye= Eye.LEFT_EYE;
    private Eye lazyEye;
    private SpeechRecognizer recognizer;
    private HashMap<String, Integer> captions;
```

Figura 22 – SettingsActivity

In figura 22 si può vedere la prima parte della Activity Settings dove vengono importare le librerie di pocketSphinx e la classe SettingsActivity che implementa RecognitionListener per il riconoscimento vocale.

```
@Override
public void onActivityResult(Hypothesis hypothesis) {
    if (hypothesis != null) {
        String text = hypothesis.getHypstr();
        if (text.contains("left")) {
            textEyeLeft.setText("Left");
            textEyeRight.setText("Left");
            lazyEye = Eye.LEFT_EYE;
        }
        else if (text.contains("right")) {
            textEyeLeft.setText("Right");
            textEyeRight.setText("Right");
            lazyEye = Eye.RIGHT_EYE;
        }
        else if (text.contains("okay"))
            goToMain();
    }
}
```

**Figura 23 – OnResult SettingsActivity**

In quest'altra figura è possibile notare la funzione OnResult sempre della classe SettingsActivity. Viene eseguito un controllo su hypothesis che contiene i vocaboli che vengono percepiti mentre la funzione “ascolto” è attiva. Come prima cosa viene verificato che siano presenti dei vocaboli, ovvero che hypothesis non sia nullo. Se questo è verificato viene definita una Stringa text contenente i vocaboli presenti in hypothesis.

Se text contiene “left”, viene visualizzata la parola “Left” sulle textView denominate textEyeLeft e textEyeRight e viene impostato l'occhio sinistro come occhio pigro.

Se text contiene “right” sulle textView viene visualizzato “Right” e l'occhio destro viene impostato come l'occhio pigro.

Infine se text contiene “okay” viene lanciata l'activity Main, ovvero la schermata di gioco.

```

@Override
public void onResult(Hypothesis hypothesis) {
    if (hypothesis != null) {
        String text = hypothesis.getHypstr();
        if (text.contains("left")) {
            volumeUp();
            textcd.setText("Left");
            textcs.setText("Left");
        }
        else if (text.contains("right")) {
            volumeDown();
            textcd.setText("Right");
            textcs.setText("Right");
        }
        else if (text.contains("home"))
            playAgain();
    }
}

```

**Figura 24 – onResult MainActivity**

Qui invece viene visualizzato onResult ma per il Main Activity.

In questo caso quando text contiene la parola “Left” o “Right” viene lanciata la funzione VolumeUp (nel caso di “Left”) o la funzione VolumeDown (nel caso di “Right”).

Queste funzioni muovono rispettivamente a sinistra e a destra il veicolo comandato dall’utente.

Nel caso in cui text contenga “home” il gioco viene riavviato.

```

private void setupRecognizer(File assetsDir) throws IOException {
    // The recognizer can be configured to perform multiple searches
    // of different kind and switch between them

    recognizer = defaultSetup()
        .setAcousticModel(new File(assetsDir, "en-us-ptm"))
        .setDictionary(new File(assetsDir, "cmudict-en-us.dict"))

        // To disable logging of raw audio comment out this call (takes a lot of space on the device)
        //.setRawLogDir(assetsDir)

        // Threshold to tune for keyphrase to balance between false alarms and misses
        //.setKeywordThreshold(1e-45f)

        // Use context-independent phonetic search, context-dependent is too slow for mobile
        .setBoolean("-allphone_ci", true)

    .getRecognizer();
    recognizer.addListener(this);
}

```

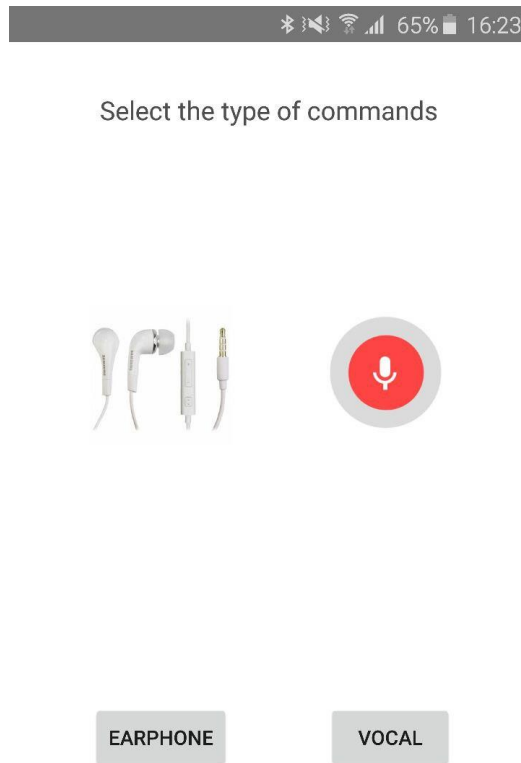
**Figura 25 – setupRecognizer**

In figura 25 è mostrato il Setup di default per il recognizer dove vengono settati il modello acustico e il dizionario di riferimento.



### 9.3. Schermata selezione comandi

Per poter permettere all'utente di scegliere quale tipo di comandi utilizzare, ho aggiunto una schermata iniziale che permette di farlo.



**Figura 26 – schermata scelta comandi**

Al momento dell'avvio della app viene visualizzata la seguente schermata (Figura 26) dove l'utente può scegliere che tipologia di comandi utilizzare.

A seconda del tasto che viene premuto (Earphone o Vocal) verrà abilitato una delle due tipologie di comandi.

---

## 10. Conclusione e Ringraziamenti

Lo scopo dell'applicazione Car Racing CardBoard è il trattamento dell'ampliopia. Gli aspetti fondamentali del lavoro erano di aggiungere una seconda tipologia di comandi per questa applicazione, con l'obiettivo di implementare i comandi vocali. Con l'inserimento dei comandi vocali è stato possibile rendere l'applicazione hands-free, quindi permettere all'utente di avere le mani libere e di comandare tutto tramite la voce.

Ora l'utente ha la possibilità di scegliere quale tipologia di comandi utilizzare nella schermata iniziale, quindi se l'originale utilizzando i tasti presenti sulle cuffie oppure i comandi vocali.

Per ora nel progetto Sphinx è disponibile solamente la versione inglese-americana del dizionario e del modello acustico, quindi è in grado di riconoscere solo vocaboli di questa lingua.

Qualora venisse aggiunto il supporto alla lingua italiana nel progetto sarà possibile implementarlo nell'applicazione in modo da permetterne l'uso anche con comandi espressi in lingua italiana.

Per la prova finale "RICONOSCIMENTO COMANDI VOCALI PER VIDEOGIOCHI CARDBOARD" ringrazio il relatore, prof. Angelo Gargantini e la Dott.ssa Silvia Bonfanti che mi hanno concesso l'opportunità di affrontare questi concetti.

*Coter Daniele*



---

# Bibliografia

Ambliopia (occhio pigro) <http://www.my-personaltrainer.it/salute-benessere/ambliopia.html> (MyPersonaTrainer) 2016

Come funziona CardBoard (Fastweb copyright CULTURE) 2014

3D4Amb (Gargantini, Bonfanti, Facoetti, Vitali, Zambelli, Terzi)

<http://3d4amb.unibg.it/index.it.html>

Google Cardboard, i migliori giochi e applicazioni (Fastweb copyright CULTURE) 2016

Integrazione di un sistema di riconoscimento automatico del parlato nella piattaforma Cyrano (Alessandro Mazzei) Marzo 2005

Progetto Sphinx <http://cmusphinx.sourceforge.net> 2015

Un Videogioco su realtà virtuale a basso costo per il trattamento dell'ampliobia (Fabio Terzi, Matteo Zambelli) 2014/2015

