SOMMARIO

1 Introduzione

Il lavoro svolto nel corso di questa tesi di Laurea Triennale in ingegneria informatica è stato quello di progettare, sviluppare ed implementare un'applicazione software sotto la direzione del prof. Angelo Gargantini e della dott.ssa Silvia Bonfanti.

L'applicazione prende il nome di StereoAcuityMultiTestApp.

L'obiettivo di questo progetto era quello di creare un'applicazione per dispositivi Android che vertesse alla misurazione dell' acuità visiva, utile per rilevare la presenza di patologie oculari, in particolar modo di ambliopia.

In primo luogo verranno mostrate la natura del problema, al fine di capire di che cosa si tratta la malattia che si sta cercando di identificare, e le soluzioni, ovvero terapie e tecniche, che sono state impiegate nel corso degli anni per diagnosticare e curare la patologia.

In secondo luogo si presenterà un'analisi delle tecnologie software che sono state utilizzate per realizzare questa applicazione, dando spazio anche agli strumenti già esistenti usati per altri progetti.

Successivamente si descriverà l'applicazione vera e propria: partendo dall'idea "su carta", si ripercorreranno i passi i quali hanno portato alla creazione degli algoritmi che permettono di eseguire il test di acuità visiva. Questi passi riguarderanno:

- la fase di definizione dei risultati voluti
- la fase di implementazione del codice
- la fase di testing degli algoritmi

Infine verrà presentata una serie di possibili migliorie per l'applicazione in questione.

2 Ambliopia

Per prima cosa si tratterà la malattia che il software sviluppato vuole identificare.

2.1 Acuità visiva

L'acuità visiva (o visus) è la capacità dell'occhio umano di vedere distintamente due oggetti. Generalmente viene misurate dall'ottico o dall'oculista per stabilire il grado di normalità del visus di ogni individuo (es: 10/10), valutando anche la nitidezza dell'immagine proiettata sulla retina, tramite vari test.

Essendo una delle abilità maggiormente tenute in considerazione, ogni diminuzione dell'acutezza visiva dai valori normali (es: 10/10) durante l'esame visivo, è spesso sintomo che richiede maggiori approfondimenti.

Proprio grazie ad essa si è in grado, vedendo due punti, non solo di percepirne l'esistenza, ma anche di capire che sono separati e distinti, individuandone i limiti e i confini nettamente.

Questa prorpietà dell'occhio umano si misura valutando il minimo valore dell'angolo sotto il quale si ha la visione distinta di due punti luminosi.

L'angolo minimo che viene preso in considerazione è 1', ovvero un sessantesimo di grado. Questo valore permette di far corrispondere le immagini a due recettori retinici non contigui (che distano nell'occhio umano 5 millimetri) e, quindi, di osservare le due immagini e di identificarle come distinte.

2.2 Fisiologia ed eziologia dell'ambliopia

Per definizione, l'ambliopia (dal greco ἀμβλυωπία, composto di ἀμβλύς, «ottuso, fiacco, debole», e ὤψ, «vista»), comunemente chiamata "occhio pigro", si riferisce alla riduzione unilaterale o, meno comunemente, bilaterale, dell'acuità visiva meglio corretta.

Essa non è direttamente attribuita ad un'anomalia strutturale dell'occhio o delle vie visive posteriori, ma bensì correlata ad un danno visivo cerebrale, in assenza di causa organica.

Si ritiene che derivi dalla degradazione dell'immagine retinica associata a un'esperienza visiva anormale durante il periodo di sviluppo del sistema visivo nell'infanzia e nella prima infanzia (Kanonidou, 2011).

Essa dipende da un'alterata trasmissione del segnale nervoso tra occhio e cervello: quest'ultimo andrà a privilegiare un occhio a causa della ridotta acuità visiva dell'altro. Più semplicemente, consiste in una riduzione della capacità visiva di uno o due occhi (che devono differire di almeno 3\10 l'uno dall'altro, o avere un visus inferiore ai 3\10).

Le sue cause primarie sono lo strabismo (perdita dell'allineamento oculare), l' anisometropia (perdita di messa a fuoco) o errori refrattivi bilaterali elevati e privazione dello stimolo, come la cataratta congenita, che impedisce allo stimolo luminoso di raggiungere la retina. Questi problemi portano a una riduzione della chiarezza dell'immagine, interrompendo così la visione binoculare.

Come ci ricordano Birch, Kelly e Wang nel loro recente articolo pubblicato a settembre 2021, l'ambliopia è la causa più comune di disabilità visiva monoculare nei bambini, con una prevalenza del 2-3%. Non solo l'acuità visiva è ridotta in un occhio, ma la visione binoculare è compromessa, possono essere presenti deficit oculari simili, la coordinazione occhio-mano e la lettura possono essere compromesse e la percezione di sé può essere ridotta.

2.3 Diagnosi

Diagnosticare precocemente l'ambliopia è la chiave del successo dei successivi trattamenti. Come sostiene l'USPSTF (Preventive Services Task Force), supportata dall'American Academy of Family Physicians, è ottimale sottoporre tutti i bambini di età compresa tra i tre (anche fin dopo i sei mesi, se possibile) e i cinque anni a uno screening per l'ambliopia o per verificare la presenza di eventuali fattori di rischio presenti, in modo tale che, non appena ne viene identificato uno, si possa sottoporre il bambino ad un esame oftalmologico completo.

Con la diagnosi e il trattamento precoci, i bambini con ambliopia possono migliorare significativamente il problema. In uno studio longitudinale randomizzato basato sulla popolazione, lo screening precoce ripetuto ha comportato una riduzione del 60% della

prevalenza di ambliopia e un miglioramento dell'esito dell'acuità visiva all'età di 7 anni rispetto alla sorveglianza solo fino all'ingresso a scuola. Inoltre, l'ambliopia può essere trattata in modo più efficace se trattata precocemente; lo stesso studio ha riscontrato una prevalenza inferiore del 70% di ambliopia residua dopo il trattamento quando la terapia è stata iniziata prima dei 3 anni (Birch et al., 2021).

Tuttavia, se viene trascurata e non trattata durante l'infanzia, purtroppo diminuisce permanentemente la vista. Pertanto, la prevenzione, l'individuazione e il trattamento dipendono in gran parte dai genitori che, come dimostrato da Alatawi et al. nello studio effettuato nel 2021, spesso sono poco consapevoli e ignorano il problema. È quindi fondamentale una sensibilizzazione maggiore per accrescere il livello di consapevolezza, che, successivamente, garantirà una maggior partecipazione agli screening.

2.3.1 Fotoscreening

La prima metodologia utile a identificare l'ambliopia consiste nel fotoscreening. Essa viene utilizzata dai sei mesi fino a quando il bambino è in grado di leggere un grafico per il test della vista. Questo metodo consiste nell'utilizzo di una macchina fotografica per registrare le immagini dei riflessi pupillari, durante la fissazione di un bersaglio visivo e i rilessi rossi in risposta a uno stimolo luminoso, per poi confrontarle e valutare eventuali opacità oculari, disallineamento e asimmetrie. Le immagini vengono analizzate direttamente dal software o dallo specialista.

Figura 1: interfaccia di un'applicazione per fare fotoscreening

I dispositivi automatizzati forniscono un approccio semplice e più efficace allo screening, mirando a rilevare l'errore di rifrazione e i fattori di rischio dello strabismo per l'ambliopia, piuttosto che rilevare direttamente l'ambliopia. Tuttavia, la prevalenza dei fattori di rischio è del 21% rispetto

a una prevalenza di ambliopia del 2,5%, ovvero 8 o 9 bambini su 10 che non superano lo screening automatizzato perché hanno un fattore di rischio sono falsi positivi e non hanno l'ambliopia (Birch et al., 2021). In un recente studio su 10.000 bambini di 3 anni visti nella rete Pediatric Research in Office Settings (PROS), i pediatri hanno riferito di non aver nemmeno tentato lo screening per il 62% dei bambini. I membri di PROS hanno riferito che l'alto tasso di falsi positivi (sovraccarico) di screening automatizzati era la principale barriera (Birch et al., 2021).

2.3.2 Test dell'acuità visiva

Nel momento in cui il bambino, all'età di 3\4 anni, è in grado di collaborare tramite l'utilizzo della parola, gli possono essere sottoposti diversi test grafici, considerati il gold standard, come la mappa oculare di Snellen (test che coinvolge un grafico con 11 righe di lettere; la prima riga è composta da 1 lettera molto grande e ogni riga sottostante ha un numero crescente di lettere che diminuiscono di dimensione). Anche per i bambini prealfabetizzati ci sono diverse opzioni, come mostrato in figura 1: le figure Allen (test che coinvolge 4 flash card che contengono 7 figure schematiche; le figure sono identificate da varie distanze), i grafici E cadenti (test che coinvolge la lettera E presentata con le braccia che puntano in direzioni diverse; le lettere diminuiscono di dimensione dall'alto

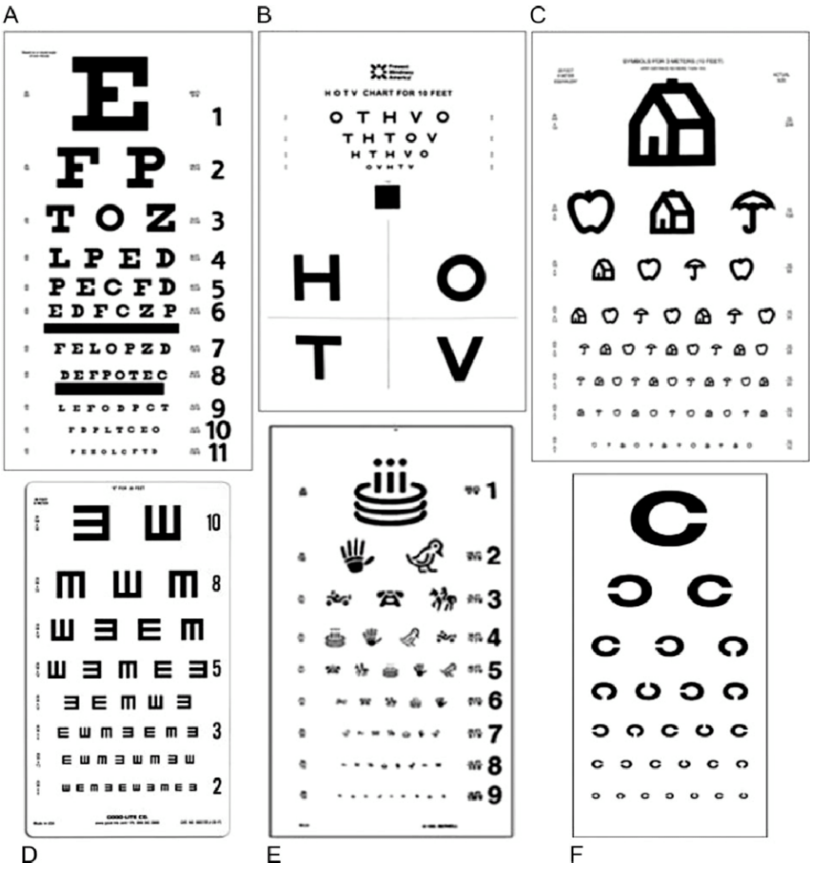


Figura 2 Esempi di grafici dell'acuità visiva. (A) Snellen, (B) HOTV, (C) Lea, (D) Tumbling "E", (E) Allen e (F) Landolt C.

verso il basso del grafico), i grafici HOTV (test che comporta l'identificazione delle lettere H, O, T e V; le lettere diminuiscono di dimensione dall'alto verso il basso del grafico) e i simboli Lea (test che prevede l'abbinamento dei simboli delle carte ai simboli sul muro; i simboli diminuiscono di dimensione dall'alto verso il basso del grafico). I bambini di età inferiore ai 5 anni dovrebbero essere in grado di identificare la maggior parte delle cifre sulla linea 20/40 con entrambi gli occhi individualmente. Coloro che non sono in grado di vedere la linea 20/40 con entrambi gli occhi devono essere indirizzati a un esame oftalmologico secondo le linee guida AAPOS. Dopo i 5 anni, la maggior parte dei bambini dovrebbe essere in grado di vedere la maggior parte delle cifre sulla linea 20/30 di un diagramma oculare, e coloro che non sono in grado di farlo dovrebbero essere indirizzati. Lo screening dovrebbe essere ripetuto ogni 1 o 2 anni durante gli anni della scuola elementare.

La Preventative Services Task Force statunitense afferma che i bambini tra i 3 e i 5 anni devono essere sottoposti a screening visivo almeno una volta per contribuire a promuovere la salute visiva. I test di acuità visiva sono in grado di rilevare errori refrattivi, mentre il fotoscreening è prevalentemente in grado di individuare i fattori di rischio di ambliopia con rapidità.

È raccomandato un uso concomitante di questi due test di screening, al fine di andare a identificare un maggior numero di bambini che hanno bisogno di un intervento oculistico, contribuendo a prevenire la perdita della vista.

Oltre alle metodologie di screening appena descritte, esistono oggigiorno delle nuove alternative, che vanno a colpire direttamente l'ambliopia piuttosto che i fattori di rischio. Tra di esse troviamo: la birifrangenza retinica, lo screening dell'acuità visiva del nistagmo otocinetico e l'intelligenza artificiale (Birch et al., 2021).

2.3.3 Screening binoculare della birifrangenza retinica

Recentemente, la scansione della birifrangenza retinica è stata sviluppata come metodo per rilevare l'ambliopia e lo strabismo. La scansione della birifrangenza retinica sfrutta l'architettura unica della fovea umana, con la sua organizzazione radiale delle fibre di Henle. Poiché lo strato di fibre di Henle è un tessuto birifrangente, un punto di luce

polarizzata scansionato in un anello a 100 Hz durante la fissazione foveale centrale e costante provocherà il raddoppio della frequenza e 200 Hz sarà la frequenza del segnale dominante nella luce di ritorno. Se la fissazione è instabile o eccentrica a causa dell'ambliopia, ci sarà poca o nessuna birifrangenza e, di conseguenza, segnale a 200 Hz scarso o nullo nella luce di ritorno (Birch et al., 2021).

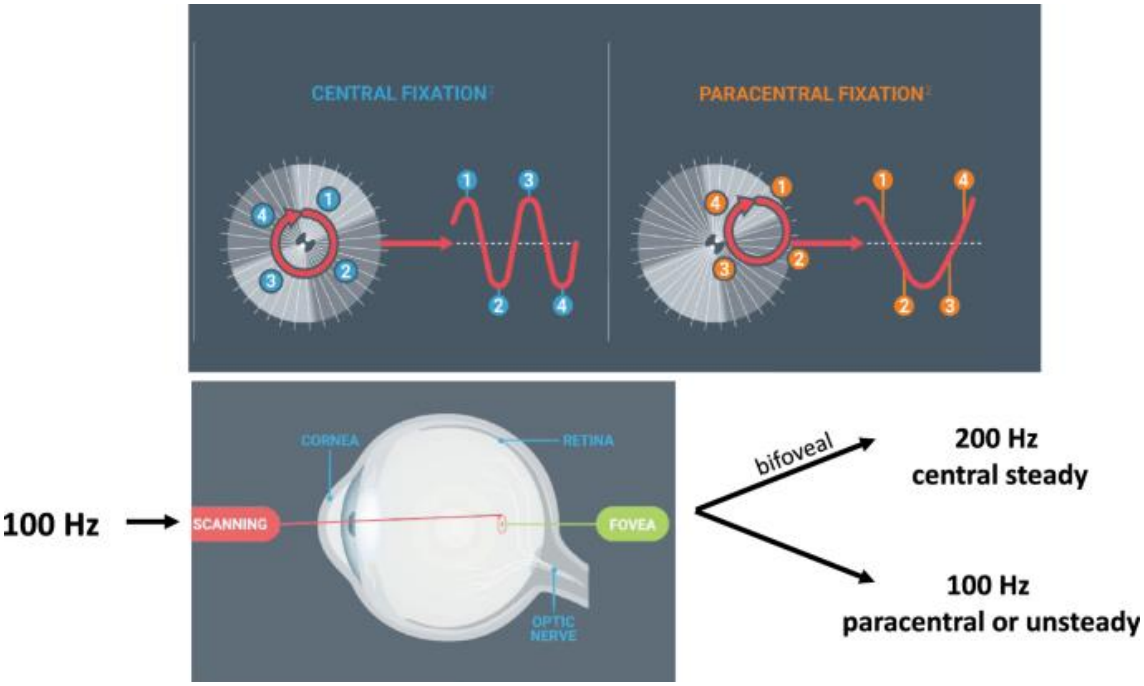


Figura 3: funzionamento screening binoculare

2.3.4 Screening dell'acuità visiva del nistagmo optocinetico

Poiché il gold standard per lo screening dell'ambliopia è il test dell'acuità visiva, c'è un grande interesse nello sviluppo di una misurazione obiettiva dell'acuità visiva nei bambini piccoli. L'obiettivo dell'acuità visiva Screener è attualmente in fase di sviluppo per soddisfare questa esigenza. A differenza dei tamburi optocinetici del nistagmo (OKN) comunemente usati in clinica, questo test utilizza nuovi ottotipi a scomparsa per indurre movimenti oculari OKN involontari (Birch et al., 2021).

2.4 Nuovi dispositivi

Un ruolo fondamentale e molto promettente è quello dell'intelligenza artificiale (AI) per migliorare lo screening della vista in età prescolare, ma inizia solo ora a essere studiato (Birch et al., 2021).

L'intelligenza artificiale può essere utilizzata per adattare la progressione degli stimoli del test alle prestazioni di ciascun bambino per ridurre al minimo il tempo di test e gli algoritmi e l'euristica parametrizzati dall'intelligenza artificiale possono ottimizzare l'elaborazione dei dati sensoriali, di imaging e dei movimenti oculari (Birch et al., 2021).

Nel complesso, l'obiettivo è addestrare i modelli di rete neurale e valutare la loro accuratezza diagnostica per l'ambliopia e lo strabismo rispetto a un esame completo standard di riferimento.

Questo è stato il concetto cardine che ha guidato l'intero sviluppo dell'applicazione per questo progetto: creare un software che potesse diagnosticare la presenza o meno di ambliopia in un paziente attraverso un test di breve durata ed adattivo. L'algoritmo che verrà presentato più avanti (sezione 8.1), infatti, regola la difficoltà del test in funzione delle risposte che il paziente fornisce.

3 Trattamenti

Le percentuali di successo del trattamento dell'ambliopia diminuiscono con l'aumentare dell'età. I vari trattamenti consistono nell'eliminazione di ogni ostacolo nella visione, nella correzione degli errori di rifrazione e nell'uso forzato dell'occhio ambliope. Nei bambini con ambliopia da ostruzione la priorità è quella di rimuovere l'ostruzione dell'asse visivo (ad esempio la cataratta) (DeSantis, 2014).

La correzione ottica nei bambini con ambliopia anisometropica o refrattiva è solitamente ottenuta con l'uso di occhiali. Recenti studi hanno dimostrato che il trattamento dell'errore di rifrazione da solo può migliorare sostanzialmente l'acuità visiva negli occhi con ambliopia. Quando la sola correzione ottica non porta a una visione normale in un occhio ambliopico, spesso è necessaria l'occlusione dell'occhio sano.

Dunque, dopo aver valutato le cause scatenanti del disturbo, il medico oculista può intervenire sull'occhio ambliope, attuando un processo di ''penalizzazione''. La penalizzazione consiste in una stimolazione dell'occhio pigro, occludendo quello sano, attraverso differenti metodologie, di seguito riportate.

3.1 Patching

Il metodo occlusivo è in assoluto quello più diffuso ed utilizzato. Esso consiste nel coprire, con bendaggi, cerotti o lenti oscurate, l'occhio sano, in modo tale da consentire una stimolazione di quello ambliope.

La dottoressa DeSantis, nel suo articolo pubblicato nel 2014, sostiene che il numero di ore a cui il soggetto è sottoposto a "terapia con cerotto" non è direttamente proporzionale: 2 ore di patch giornaliere possono essere efficaci quanto 6 ore al giorno.

Figura 4: bambino con occhio coperto dal cerotto di patching

L’uso di questa metodologia rimane il gold standard di trattamento per ora e ha diversi vantaggi: la semplicità di utilizzo, l’economicità del prodotto e l’efficacia del trattamento.

Tuttavia sono degni di nota anche alcuni svantaggi, tra cui è importante citare la scarsa tollerabilità del cerotto da parte del bambino, e che quindi porta ad una richiesta di supervisione e collaborazione continua da parte del genitore.

3.2 Somministrazione di atropina

L’atropina è un alcaloide tropanico e farmaco anticolinergico, che, nel caso dell’ambliopia, viene utilizzato sotto forma di soluzione oftalmica all’1% (collirio) per ridurre l’efficienza dell’occhio sano, stimolando quindi l’attività di quello pigro. Questo aiuta lo sviluppo della parte del sistema nervoso deputato a elaborare la visione. L'atropina può anche essere utilizzata in combinazione con la penalizzazione ottica.

Figura 5: paziente durante la somministrazione di antropina

È importante sottolineare che, nonostante sia stata dimostrata l’efficacia di questo trattamento da diversi studi, esso può comportare anche degli effetti avversi: sono stati segnalati casi di riduzione transitoria dell'acuità visiva nell'occhio non ambliopico derivante dall'uso di atropina, casi di fotosensibilità, in particolare in aree di elevata esposizione al sole. Inoltre, gli effetti avversi sistemici possono includere secchezza delle fauci, tachicardia, febbre e delirio (DeSantis, 2014).

3.3 Chirurgia refrattiva

Il ruolo della chirurgia refrattiva nel trattamento dell'ambliopia anisometropica rimane controverso. Alcuni studi hanno dimostrato che la chirurgia cheratorefrattiva è una procedura sicura nei bambini con ambliopia anisotropa che non sono conformi alla correzione refrattiva. Attualmente, la chirurgia cheratorefrattiva nei bambini è un uso off-

label di un dispositivo approvato dalla FDA. Le procedure refrattive potrebbero avere un ruolo futuro nella gestione dell'ambliopia in alcuni bambini che falliscono il trattamento convenzionale.

3.4 Agopuntura

Figura 6: paziente durante l'agopuntura

L' agopuntura, una componente essenziale della medicina tradizionale cinese e una componente importante della medicina complementare e alternativa, è efficace nel trattamento di condizioni come emicrania, lombalgia, e ipertensione, e tali disturbi oculari come miopia e ambliopia.

In uno studio randomizzato e controllato, è stato riscontrato che l'agopuntura è efficace quanto il cerotto nel trattamento dell'ambliopia anisometropica per i bambini più grandi (Lam et. Al., 2011).

È stato svolto nel 2011, da Lam e colleghi, un ulteriore studio crossover, le cui conclusioni indicano che l'agopuntura potrebbe produrre effetti aggiuntivi alla correzione refrattiva per l'ambliopia anisometropica non trattata e residua nei bambini più piccoli. Gli effetti del trattamento sono stati sostenibili nell'arco di 1 anno. Inoltre, l'agopuntura è essenzialmente sicura e ha una buona soddisfazione e accettazione dei genitori. Pertanto, l'agopuntura ha un buon potenziale per diventare una modalità terapeutica complementare per l'ambliopia nelle normali cure mediche (Lam et. Al., 2011).

Il meccanismo d'azione dell'agopuntura nel trattamento dell'ambliopia è sconosciuto e sono necessarie ulteriori indagini (DeSantis, 2014).

3.5 Terapia della vista

La terapia della vista è stata anche promossa da alcuni nel trattamento dell'ambliopia, come terapia primaria o come aggiunta o alternativa al cerotto. La terapia della vista può assumere varie forme, a seconda dell'oculista, inclusi esercizi muscolari, esercizi di inseguimento o tracciamento e occhiali con o senza lenti bifocali o prismi (DeSantis, 2014).

Sebbene la terapia della vista sia in uso da molti anni da alcuni professionisti della cura degli occhi, a tutt'oggi vi sono studi di coorte insufficienti o studi clinici randomizzati per supportare l'uso di queste tecniche (DeSantis, 2014).

3.6 Metodo I-BiT

Interactive Binocular Treatment (I-BiT) è un metodo di cura per l'ambliopia sviluppato dalla University of Notthingham.

Sebbene il bendaggio dell'occhio dominante sia il metodo conosciuto più efficace di terapia dell'ambliopia, presenta alcune limitazioni, specialmente tra i bambini con minore propensione alla collaborazione. Inoltre, il miglioramento visivo mediante l'applicazione di patch richiede diversi mesi e, per tutto questo tempo, il bambino ed i suoi genitori si devono impegnare nel far seguire la cura al minore.

Figura 7: macchinario per il metodo I-BiT

I-BiT funziona guardando determinate immagini attraverso occhiali con otturatore con filtri di aumento e riduzione per occhi ambliopi e non ambliopi per dissociare due occhi l'uno dall'altro. Sebbene gli occhiali con otturatore inducano una maggiore stimolazione foveale della retina, sono costosi e la maggior parte delle famiglie non può permetterseli. Il meccanismo si basa sulla presentazione di diverse condizioni dell'immagine a ciascun occhio del bambino ambliopico mentre entrambi gli occhi sono aperti. A questo proposito

- All’occhio ambliopico vengono mostrati dettagli fondamentali dell’immagine
- All’occhio sano vengono mostrate informazioni secondarie della scena
- Entrambi gli occhi vengono mostrati ad entrambi gli occhi

L’occhio malato, dunque, è messo sotto particolare sforzo dal paziente che deve cercare di progredire il più possibile nel gioco, fino ad una possibile vittoria.

I vantaggi del metodo I-BiT sono molteplici, sia per il paziente che per il medico:

- È un metodo più efficace nei bambini di età superiore a 8 anni in quanto attiva le cellule dormienti della corteccia visiva del paziente
- Non è richiesto il bendaggio dell’occhio sano. Il paziente deve solo indossare l’apparecchiatura montante i due schermi sui quali vengono proiettate le immagini che verranno viste separatamente.
- I bambini sono più propensi a collaborare in quanto devono semplicemente “giocare” con l’applicazione
- Si ha un miglioramento visivo del paziente a partire da dopo solo sei settimane (contro i mesi richiesti dal metodo di patching)
- Il medico può vedere su uno schermo collegato all’apparecchiatura le immagini che vengono mostrate al paziente co una serie di parametri clinici che gli permettono di modificare i parametri del sistema tramite un pannello di controllo.

L’unico contro, di fatti, è il costo delle apparecchiature: ciò ne impedisce la diffusione e l’acquisto da parte delle famiglie con bambini con occhio pigro, costringendoli a recarsi in strutture specializzate per seguire la terapia (invece che farla da casa).

L'implementazione di un sistema di realtà virtuale mediante l'applicazione di filtri anaglifici (rosso/verde) comprende i vantaggi dell'I-BiT con un costo inferiore. Questo metodo, infatti, sfrutta occhiali speciali con filtri rosso/verde che hanno un costo decisamente inferiore rispetto alle apparecchiature del metodo I-BiT, ma anche una minore stimolazione foveale della retina.

4 3D4amb

Il progetto 3D4amb, sviluppato dall'Università degli Studi di Bergamo, mira allo sviluppo di sistemi basati per diagnosticare e trattare l'ambliopia nei bambini. Sfrutta le tecnologie 3D per mostrare immagini differenti ai due occhi, quello ambliopico (o pigro) e quello sano.

Figura 8: logo del progetto 3D4amb

Molte applicazioni sono nate da questo progetto con lo scopo di visitare i giovani pazienti attraverso delle attività software interattive: in questo modo si è riusciti a sostituire parzialmente il metodo del patching, ma il più grande vantaggio ottenuto è la semplicità dei metodi di diagnosi rispetto alle complessità dei metodi classici.

4.1 Filosofia del progetto

L'obiettivo principale di questo progetto di ricerca è quello di sviluppare un sistema per la diagnosi ed il trattamento dell'ambliopia con le seguenti caratteristiche:

- Economico: non sono richieste attrezzature particolarmente costose per avviare le applicazioni del progetto in quanto bastano comuni pc, tablet o smartphone con una potenza di calcolo nemmeno troppo elevata
- Facile da usare: i software devono essere di facile utilizzo ed intuitivi, in modo che sia i bambini che i rispettivi genitori (o, più in generale, del personale non specializzato) possano eseguire le attività senza una particolare skill.
- Utilizzabile anche in casa: il trattamento può essere eseguito anche autonomamente nelle proprie mura domestiche, data l'assenza di macchinari

medici particolari. Questo, inoltre, fa si che il paziente non debba recarsi in ospedale o in centri specializzati.

- Altamente espansibile: l'implementazione di nuovi software da aggiungere al progetto deve essere il più facile possibile. Per questo motivo vi sono degli standard e delle librerie che possono essere usate per sviluppare nuove applicazioni.
- Divertente: data la giovane età dei pazienti, si deve cercare di rendere le applicazioni il più interattive possibili, con lo scopo di ottenere la collaborazione dei bambini. Questa si guadagna tramite la natura videoludica delle applicazioni del progetto.

4.2 Principio base

L'obiettivo delle applicazioni del progetto è quello di consentire la visione binoculare del paziente sfruttando le tecnologie 3D e restando aderenti alle caratteristiche elencate sopra.

Un sistema 3D fornisce ai due occhi due immagini diverse della stessa scena con angoli di visione leggermente sfalsati che altro non sono che i diversi punti di vista dell'occhio destro e dell'occhio sinistro. In questo modo si produce un'illusione della profondità reale della scena (è lo stesso meccanismo alla base della realtà virtuale). Le applicazioni non creano una realtà virtuale, ma sfruttano le capacità del sistema 3D di inviare agli occhi due immagini diverse .

All'occhio ambliopico e a quello normale vengono mostrate due immagini diverse: al primo si mostrano le immagini più interessanti e più ricche di dettagli, mentre a quello sano vengono mostrate immagini private delle parti d'interesse.

Per raggiungere tale scopo esistono svariate tecniche.

4.2.1 Tecnica di separazione visiva anaglifica

Un anaglifo è un'immagine stereoscopica che, se osservata mediante appositi occhiali dotati di due filtri di colore complementare l'uno rispetto all'altro, fornisce un'illusione

di tridimensionalità. SI tratta di una sovrapposizione di due immagini filtrate con due colori specifici (solitamente rosso e ciano) le quali creano un'immagine illusoriamente tridimensionale se guardata attraverso quegli occhiali.

Figura 9: occhiali anaglifi

Gli occhialini (normalmente con montatura in cartoncino) devono montare un paio di filtri dello stesso colore di quelli utilizzati nel processo di creazione dell'immagine anaglifica. La lente rossa, posizionata in corrispondenza dell'occhio sinistro, permette di vedere la parte dell'immagine con filtraggio ad essa complementare, mentre la lente ciano (oppure blu o verde), permette di vedere la parte con filtraggio rosso. In questo modo ad ogni occhio giunge soltanto l'immagine che esso deve vedere, mentre l'altra immagine viene filtrata dalla lente colorata

4.2.2 Tecnica con occhiali 3D

Un metodo simile a quello con anaglifi è ottenuto usando degli appositi occhiali 3D a lenti non colorate (come gli occhiali 3D di NVIDIA).

Con questi occhiali, il contenuto che deve essere mostrato dal paziente (gioco o immagine) è diviso dal sistema 3D4Amb in due parti, una per l'occhio destro (l'occhio ambliopico nella figura) e una per l'occhio sinistro (l'occhio buono nella figura) e sarà il software a decidere cosa inviare ad entrambi gli occhi a seconda del tipo di trattamento suggerito dal medico. In questo modo si obbliga il cervello del paziente ad unire le due immagini al fine di poter eseguire correttamente le attività interattive. Ciò viene reso possibile dal fatto che le immagini hanno un numero significativo di elementi comuni e, se messe assieme, formano un'immagine bidimensionale.

Da notale come l'occhio ambliope è stimolato a lavorare, ma quello sano non viene rattoppato.

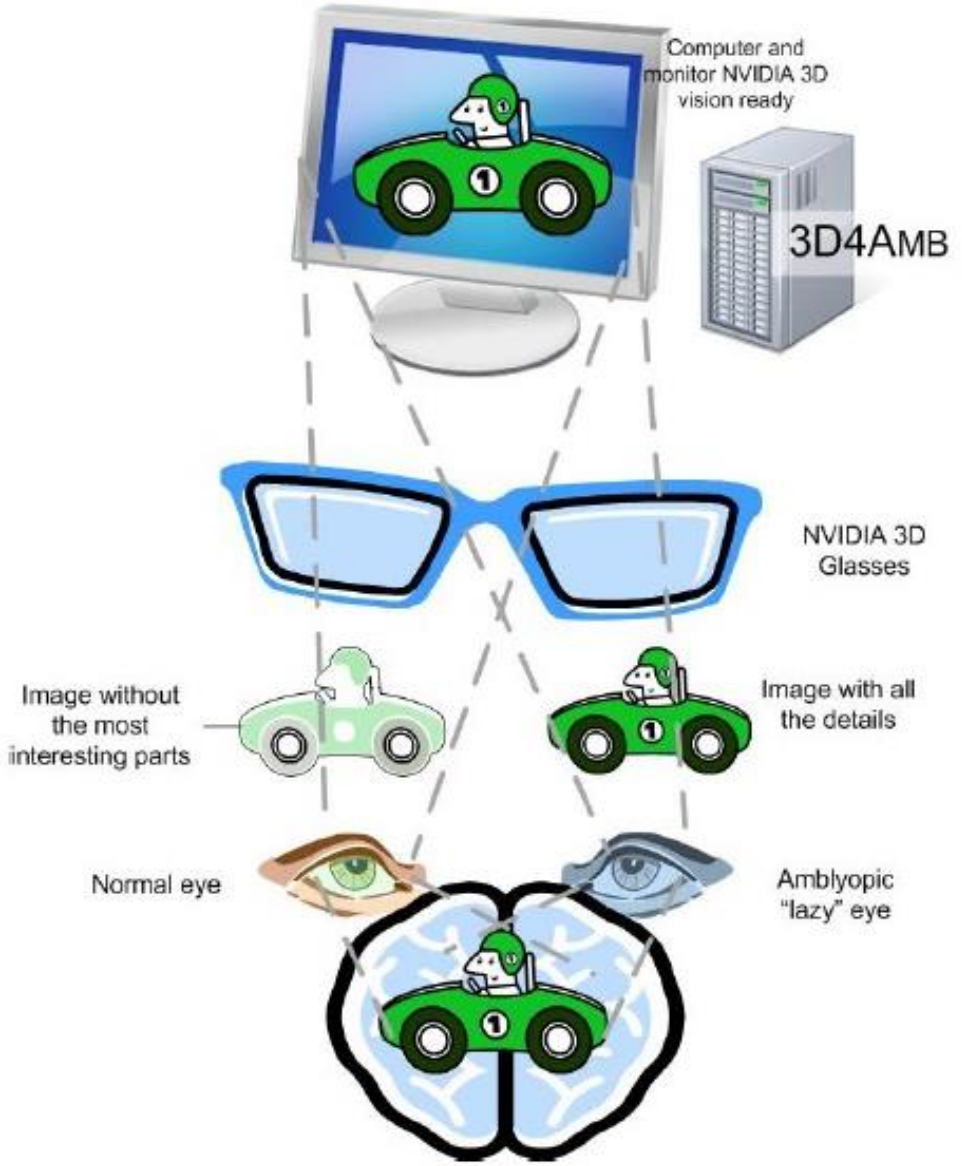


Figura 10: separazione visiva con occhiali 3D NVIDIA

4.2.3 Tecnica con dispositivi VR

Con il termine realtà virtuale si identificano vari modi di simulare delle situazioni reali mediante l'utilizzo di computer e l'ausilio di interfacce appositamente sviluppate.

Negli ultimi anni sono stati creati dispositivi per la realtà virtuale di ogni genere, dai più sofisticati come l'Oculus Rift ed il Paystation VR che necessitano il collegamento a potenti calcolatori per funzionare (computer o console), a quelli più semplici come il

Google Cardboard, un visore che sfrutta uno smartphone sia come schermo sul quale vedere le immagini che come calcolatore.

Questi visori sullo schermo del cellulare fan si di mostrare la stessa scena duplicata, una per occhio. Questo fa si che si renda necessario modificare solo una delle due immagini, invece che separare artificialmente un'immagine sovrapposta.

Il risultato è una semplificazione a livello dello sviluppo software, in quanto viene reso più semplice agire efficacemente sul campo visivo del singolo occhio.

Figura 11: cardboard

5 Tecnologie utilizzate

Prima di andare a descrivere lo sviluppo dell'applicazione e del relativo codice, è bene presentare le varie tecnologie che sono state utilizzate per la realizzazione di questo progetto.

5.1 Android Studio

Android Studio è un ambiente di sviluppo per creare le applicazioni sul sistema operativo Android. Integra un editor IDE (Integrated Developement Enviroment) con layout visuale e per essere utilizzato nel modo corretto è necessario conoscere il linguaggio Java.

Figura 12: logo Android Studio

Tra professionisti e amatori, è considerato il tool ufficiale per la realizzazione di applicazioni per il sistema operativo omonimo. Il download, assolutamente senza costi, è disponibile per diverse piattaforme e per un po' tutti i principali sistemi operativi in circolazione, come Windows, Linux e macOS. Con la sua uscita, Google ha voluto sostituire il precedente Eclipse con i suoi Android Development Tools, mettendo a disposizione degli sviluppatori un ambiente di sviluppo sensibilmente migliorato, sia dal punto di vista delle funzionalità che dell'efficienza nel sviluppare gli applicativi. Per altro con una maggiore facilità di utilizzo, grazie ad una interfaccia più pulita e ottimizzata anche per gli utenti meno esperti.

Di fatto, Android Studio è lo strumento leader per sviluppare software destinato al sistema operativo mobile che porta lo stesso nome. Deriva da IntelliJ di JetBrains, un IDE per il

mondo Java particolarmente sensibile alle necessità degli sviluppatori. Al suo interno troviamo però anche Gradle, uno strumento pensato per tutte le esigenze legate alla build automation, particolarmente flessibile e arricchito da tutte quelle feature dei predecessori come Apache Ant e Maven. Ciò si traduce nella facoltà di creare APK (estensione per indicare un file Android Package variante del formato .JAR e usato per la distribuzione e l'installazione di componenti in dotazione sulla piattaforma per dispositivi mobili Android) varianti e multipli, così come nella presenza di diversi elementi per la cattura di prestazioni, compatibilità di versione, usabilità e problematiche di vario genere.

Non manca naturalmente ampio supporto ai servizi Google, trattandosi di un tool di sviluppo realizzato direttamente dai tecnici dell'azienda, così come un ambiente di layout con editing a tema abbastanza produttivo. Android Studio supporta anche Google Cloud Platform, cosa che facilita l'integrazione con Google Cloud Messaging e App Engine, mentre a chiudere il pacchetto troviamo il Software ProGuard e pp signing.

Questo IDE ha la peculiarità di suddividere il progetto che si sta sviluppando in più moduli, ognuno dei quali svolge un ruolo diverso.

Nel modulo "app" in particolare sono contenute le tre componenti fondamentali che permettono, una volta creato un nuovo "project", di avere a disposizione un progetto semplice, ma funzionante:

- La cartella con il codice Java
- La cartella "res", contenente le risorse realizzate in XML
- Un file AndroidManifest.xml che contiene le informazioni basilari sull'app, necessarie al sistema per far girare qualsiasi porzione di codice sulla stessa, quali nome del package Java, descrizione delle componenti e dei processi che gireranno in esse, dichiarazione dei permessi dell'app e di quelli necessari ad altre app per interagire con essa, elenco delle librerie necessarie e dichiarazione del livello minimo di API Android richieste dall'app

Successivamente al modulo, in Android Studio vi è la sezione Gradle Scripts, luogo virtuale in cui trovano casa i file di build che userà Gradle per trasformare il nostro progetto in una applicazione perfettamente funzionante.

Il Gradle è un sistema di build responsabile della compilazione del codice, del test e dell'esecuzione dell'app sul dispositivo. Entrando nel dettaglio, i file di build a disposizione sono due, uno per tutto il progetto ed uno per il solo modulo app. Di fondamentale importanza è che, non appena apportata una modifica al file di build, si deve selezionare il pulsante "Sync Project with Gradle files", pena il fallimento nel tentativo di far girare l'applicazione su un dispositivo Android.

5.2 Eclipse

Eclipse, come Android Studio, è un ambiente di sviluppo integrato.

Figura 13: logo Eclipse

Si tratta di un IDE multi-linguaggio e multi-piattaforma ed è un software disponibile gratuitamente per svariati sistemi operativi (HP-UX e AIX oltre ai classici Windows, macOS e Linux), usato per la produzione di software.

Eclipse è scritto principalmente in Java e contiene uno spazio di lavoro di base che permette lo sviluppo di applicazioni Java. A questo si unisce un sistema di plug in (ovvero di componenti software ideate per uno scopo specifico) estensibile sul quale si incentra la piattaforma di sviluppo: di fatti, tutta la piattaforma è un insieme di plug in (versione base compresa). Questi sono scaricabili dall'Eclipse Marketplace, un luogo dove il programmatore può scegliere tra migliaia di componenti software capaci di adattare l'ambiente secondo le necessità dell'applicativo che sta sviluppando.

Nel caso specifico di questo progetto, Eclipse è stato usato per sviluppare e testare l'algoritmo che regola l'esecuzione del test di acuità visiva.

5.3 GitHub

GitHub è un famoso servizio di hosting (allocazione su un server web di un sito o di un'applicazione web rendendolo così accessibile ai suoi utenti) per progetti software.

Figura 14: logo GitHub

Il sito è principalmente utilizzato dagli sviluppatori, che caricano il codice sorgente dei loro programmi e lo rendono scaricabile dagli utenti. Questi ultimi possono interagire con lo sviluppatore tramite un sistema di issue tracking, pull request e commenti che permette di migliorare il codice del repository risolvendo bug o aggiungendo funzionalità. Inoltre Github elabora dettagliate pagine che riassumono come gli sviluppatori lavorano sulle varie versioni dei repository.

6 Il progetto: StereoAcuityMultiTestApp

6.1 Analisi preliminare

Come buona prassi nello sviluppo di un qualsiasi progetto applicativo, è bene effettuare prima un'analisi di ciò che si desidera che tale applicazione svolga.

6.1.1 Obiettivo

L'obiettivo di StereoAcuityMultiTestApp è quello di poter valutare l'acuità visiva di una persona mediante un test interattivo messo in forma di gioco. Trattandosi contemporaneamente di un test medico e di un'attività videoludica, di fondamentale importanza è stato trovare una soluzione che fosse facile ed intuitiva per il paziente da utilizzare, ma al tempo stesso intrattenente, al fine di aumentare la collaborazione nell'esecuzione del test. Infatti, essendo i pazienti in giovane età, si è dovuto pensare ad un'attività che potesse coinvolgere il bambino nell'esecuzione del test senza farlo annoiare, al fine di aumentarne la propensione a collaborare.

6.1.2 Requisiti funzionali

Nell'ingegneria del software, i requisiti funzionali sono delle caratteristiche che definiscono una funzione di uno o più componenti di un sistema , definendone la tipologia di input ed output ed il comportamento, le cui esecuzioni si danno a partire da una richiesta dell'utente o del software stesso.

Essendo che il test fornito dall'applicazione viene svolto da diversi utenti, è necessario che siano messe a disposizione le seguenti funzioni:

- Il login per il dottore
- La possibilità di selezionare i settaggi del test che si andrà ad effettuare, in particolare quelle di impostare la difficoltà di un livello massimo e di scegliere l'immagine con la quale si decide di effettuare il test
- Selezionare il paziente a cui verrà sottoposto il test (con la possibilità di registrarne di nuovi)

- Mostrare i risultati del test appena concluso
- Salvare i risultati del test nel database del dispositivo per poterli consultare successivamente
- La possibilità per un paziente di effettuare il test anche se il dottore non effettua il login: in questo caso l'applicazione funzionerà allo stesso modo, con l'unica differenza che il nome del paziente deve essere definito prima di effettuare l'accesso nell'applicazione. Esso, non essendo associato a nessun dottore, non verrà salvato in una particolare lista ed i risultati del suo test saranno scritti in un file di default.

  Questa particolare funzionalità è stata richiesta esplicitamente dai medici che, una volta eseguito il test, non volevano che i risultati venissero inviati al database online dell'applicazione.

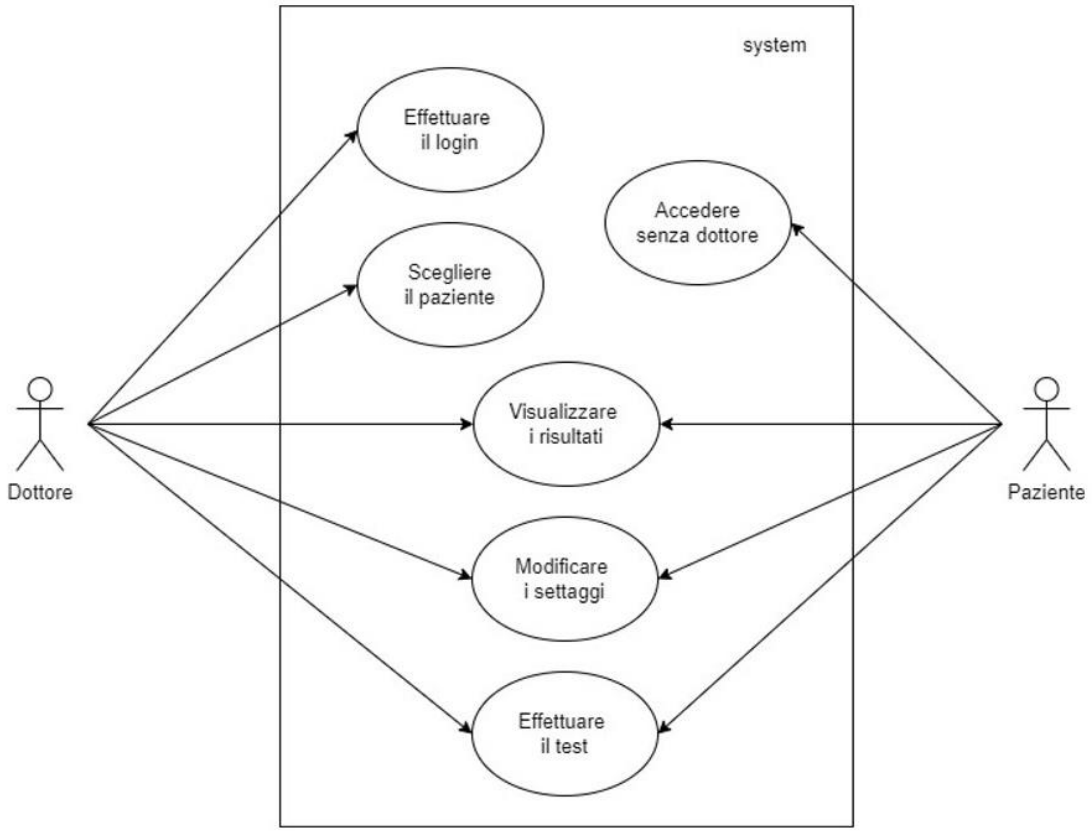


Figura 15: diagramma dei casi d'uso del software

6.2 Configurazione del progetto

Per creare un progetto in Android Studio occorre seguire due passi:

- Scegliere la tipologia di applicazione ed il relativo contesto applicativo, ovvero per quali dispositivi tale applicazione è destinata (phone e tablet, wear OS, Android Auto o Android Things)
- Configurare il progetto in tutti i suoi aspetti: nome dell'applicazione, nome del package, linguaggio di programmazione (Java o Kotlin) e la versione minima indicativa delle API da andare a supportare.

Nel caso particolare di questa applicazione, è stato scelto il contesto "Phone and Tablet" che prevede come minimo SDK Android 5.0 Lollipop (API 21). Quest'ultimo, secondo Android Studio, permette all'applicazione di girare su circa il 98% dei dispositivi che hanno installato l'omonimo sistema operativo.

6.3 Setup del progetto

Nonostante le operazioni svolte al passo 6.2 hanno portato alla creazione di un progetto funzionate, per lo sviluppo di StereoAcuityMultiTestApp si sono dovute svolgere due operazioni aggiuntive:

- L'importazione della libreria "3d4amb.sat.lib" (vedi 6.3.1)
- L'importazione di tutte le immagini che verranno usate per eseguire un test (vedi 6.3.2)

6.3.1 3d4amb.sat.lib

3d4amb.sat.lib è una libreria sviluppata precedentemente per il progetto 3D4amb ed è stata importata in questo progetto Andorid tramite il file Gradle dell'applicazione. Essa mette a disposizione una serie di classi per la gestione delle immagini di test. Di particolare importanza sono le classi:

- SATTEST: classe che contiene i metodi per la creazione dell'immagine anaglifa. Di particolare importanza è il metodo GetPointFromShape che restituisce una amtrice di PointType grazie alla quale si riesce ad effettuare la costruzione della bitmap e a mostrare a schermo ciascuna immagine.

- ImageShape: classe presente nel folder "shapes" che permette di gestire tutte le figure ed i relativi set di immagini

- RandomDotImage e StripesImage: classi del folder "background" che estendono la classe "PointImage" usate come alternative dal metodo GetPointFromShape per dire se le immagini dovranno essere rappresentate secondo RanDots Anaglifi o mediante stripes. Il secondo caso viene evidenziato perché è stato utilizzato al fine di capire la logica di creazione delle immagini da parte del metodo sopra citato.

6.3.2 I set di immagini

Nonostante nella libreria siano presenti i sei set di immagini, queste non vengono lette da Android Studio. Si è resa quindi necessaria un'operazione di importazione di tutte le immagini nella cartella "res\drawable" dell'applicazione.

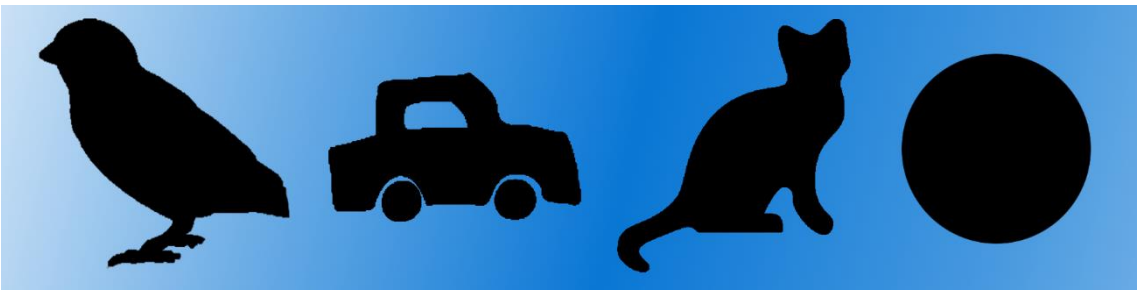


Figura 16: imageset lang

Figura 17: imageset lea

Figura 18: imageset lea con contorno

Figura 19: imageset letters

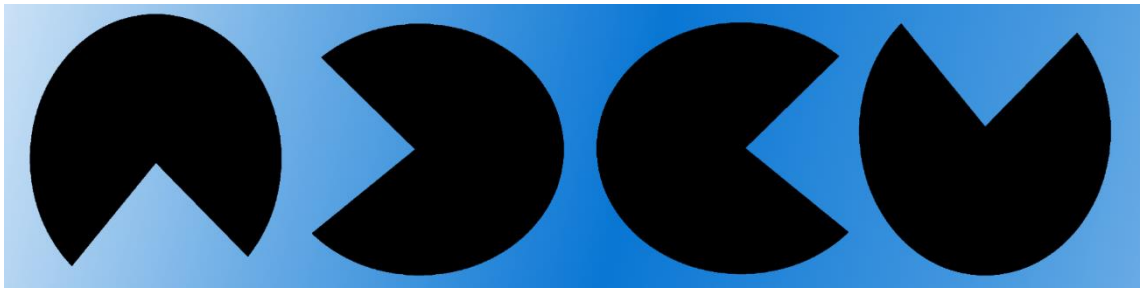


Figura 20: imageset pacman

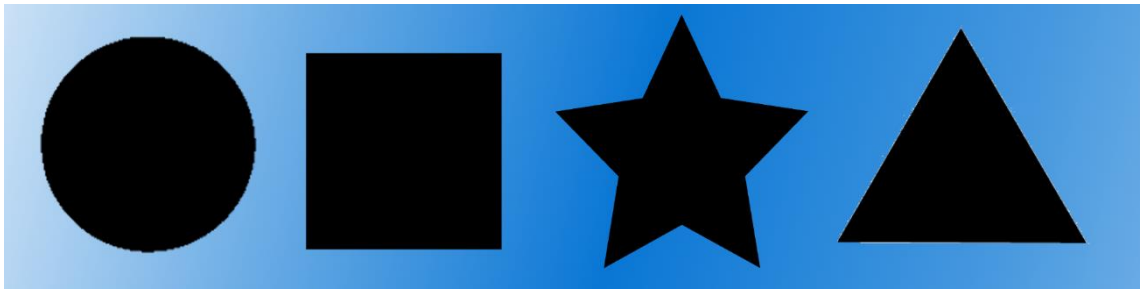


Figura 21: imageset tno

7 Applicazione software

Dopo aver parlato dei tool che sono stati utilizzati e aver mostrato le fasi che hanno permesso di fare il setup del progetto, si passa a descrivere l'applicazione vera e propria.

7.1 Caratteristiche generali

L'applicazione è stata sviluppata con l'IDE Android Studio per dispositivi mobili che supportano il sistema operativo Android tramite il linguaggio Java.

L'applicazione consente a diversi utenti registrati di poter svolgere un test per l'acuità visiva mediante randots anaglifi. Il test è suddiviso in passi: ad ognuno di essi vengono mostrate all'utente sei immagini, tre delle quali contengono al loro interno una shape (scelta tra quelle disponibili nei vari set) con un relativo valore di disparità, mentre le restanti tre non contengono nulla. L'obiettivo del test è quello di individuare il maggior numero di immagini contenenti una shape, evitando di selezionare le immagini che non ne contengono nessuna. In base a come il paziente esegue del test, esso può incrementare o diminuire la difficoltà. Al termine, i risultati sono mostrati nella schermata di "results".

L'applicazione mette a disposizione due modalità di test:

- Una "permissiva", nella quale si registrano tutti i valori di disparità "visti"
- Una "severa", nella quale si registra un certo valore di disparità come "visto" solo se la risposta a quel passo del test è classificata come "RIGHT"

Inoltre, vi è anche la possibilità di effettuare una fase di training nella quale si va a mostrare il funzionamento del test.

Il funzionamento dell'Activity di test verrà presentato in modo più completo più avanti (sezione 8.1).

7.2 Architettura software

L'applicazione risiede nel package principale "com.example.stereoacuitymultitestapp" all'interno del quale si trovano suddivise tutte le classi java in due "gruppi": la serie dei

file .java delle varie Activites ed il package .utils il quale contiene le classi di supporto al progetto.

7.2.1 Le Activities

Un'applicazione Android consiste in un "assemblaggio" di componenti Activity indipendenti o accoppiate in modo lasco. Ad ogni Activity è associato un file .xml che risiede nella cartella res\layout del progetto e corrispondono ai contenitori delle View, luogo in cui è possibile visualizzare l'Activity corrispondente.

In Android Studio un Activity è un componente software incaricato di visualizzare la "User interface" e di gestire l'interazione con l'utente. Tipicamente corrisponde alla schermata visualizzata in un determinato momento sul dispositivo.

In StereoAcuityMultiTestApp esse sono:

- LoginActivity: è l'activity con cui viene avviata l'applicazione. Da questa è possibile accedere all'applicazione secondo due modalità:

  1) Come dottore tramite una propria mail ed una propria password: per fare ciò bisogna riempire gli appositi campi e premere il pulsante di "sing in"

  2) Come paziente: in questo caso si preme il pulsante "Try without registration", si inseriscono il proprio nome ed il proprio cognome e si può eseguire l'accesso anche non essendo dei dottori.

  Mell'Activity è inoltre presente il pulsante di "sign up": questo rimanda al sito "https://se4med.unibg.it/home/#login" nel quale un nuovo dottore può registrarsi oppure un dottore già registrato può controllare i suoi dati personali ed i progressi dei suoi pazienti.

- NoRegPatientActivity: questa Activity viene creata solo quando si preme sul pulsante "Try without registration" di LoginActivity. In essa viene solo chiesto di inserire nome e cognome del paziente che andrà ad effettuare il test. Confermati i due campi si preme sul pulsante "Enter without registration" e si ha accesso alle funzioni dell'applicazione nella loro totalità.

- MainActivity: è l'Activity che rappresenta il menù dell'applicazione e viene avviata una volta che si effettua il login nell'applicazione. In essa sono presenti quattro "Buttons" che rimandano ad altre attività nell'applicazione.

- PatientActivity: è l'Activity in cui si sceglie il paziente che effettuerà il test e deve venire avviata almeno una volta dopo che si effettua il login nell'applicazione. Infatti, se viene effettuato il login, il test richiede la scelta di un paziente prima di poter essere avviato. Se la lista dei pazienti è vuota (perché, per esempio, si è appena effettuata la registrazione), si fornisce la possibilità di creare un nuovo paziente. Se non si effettua il login come dottore, viene impedito all'utente di accedere a PatientActivity in quanto la definizione del nome e del cognome del paziente è stata fatta in NoRegPatientActivity.

- CreatePatient: è l'Activity che viene lanciata quando si preme sull'icona "+" presente nella PatientActivity. Come il nome lascia intendere, in questa schermata si possono creare un nuovo paziente che verrà aggiunto alla lista del dottore che ha effettuato il login nell'applicazione. In particolare viene richiesto solo di inserire nome e cognome, ma l'algoritmo assegnerà a quel paziente un ID numerico univoco.

- Settings: in questa Activity viene data la possibilità di modificare i parametri del test, quali il colore delle lenti, la distanza dallo schermo, la difficoltà minima del test (che corrisponde al valore di disparità massimo) e la severità del test (sezione 7.1).

- ChooseImageActivity: effettuato il login e selezionato un paziente, dalla MainActivity, premendo il pulsante "Test" si viene mandati in questa Activity. Di fatti, l'applicazione, permette all'utente di scegliere tra 24 immagini (sezione 6.3.2) quella che più lo aggrada per l'esecuzione del test.

- TestActivity: questa Activity è il nucleo dell'applicazione, nella quale viene sviluppato tutto il test tramite randots anaglifi creati in real time. La sessione si suddivide in due parti

    - Una parte di "demo" che viene avviata non appena l'activity è creata. Questa modalità è stata pensata per mostrare all'utente la logica del programma: delle

sei immagini a video, solo tre di esse conterranno l'immagine selezionata in ChooseImageActivity, mentre le altre tre saranno vuote. Nella "demo" va sottolineato che le immagini vengono mostrate per come sono, ovvero senza la creazione di BitMap. L'obiettivo di questa modalità, infatti, riguarda puramente il far capire nel modo più chiaro possibile al paziente il compito che dovrà svolgere

- Premuto il pulsante di "START TEST" la "demo" viene interrotta e si inizia il test vero e proprio. La sessione inizia sempre con tre valori di disparità calcolabili a priori: un'immagine avrà disparità massima (selezionata nei "Settings"), una avrà disparità minima (sempre 1) e la terza immagine avrà come disparità il valore medio tra le precedenti due (se il numero non è intero si arrotonda per eccesso). Il test, poi, evolverà in funzione delle risposte del paziente che dovrà guardare le sei immagini e cercare di riconoscere in tre di esse l'immagine scelta in ChooseImageActivity. Visionate le sei immagini, potrà premere il pulsante "SUBMIT" per confermare la risposta o "SKIP" per passare a vedere sei nuove immagini

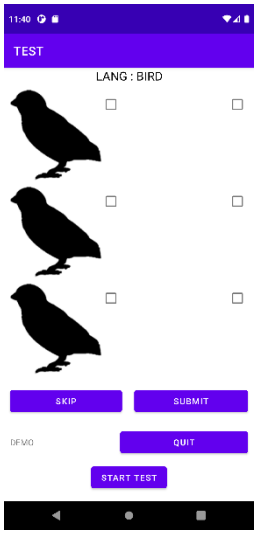


Figura 23: "demo" mode

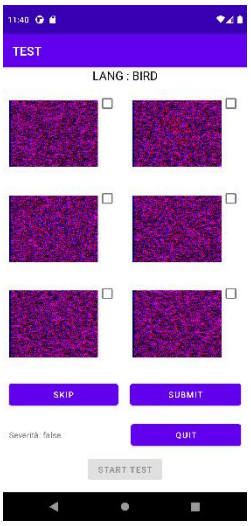


Figura 22: "test" mode

- TestResultActivity: una volta che l'algoritmo di test decide che il paziente ha terminato l'esecuzione del test, lancia questa Activity nel quale verranno mostrati i risultati sotto forma di elenco, più una serie di caratteristiche che riassumono brevemente l'esecuzione del test.

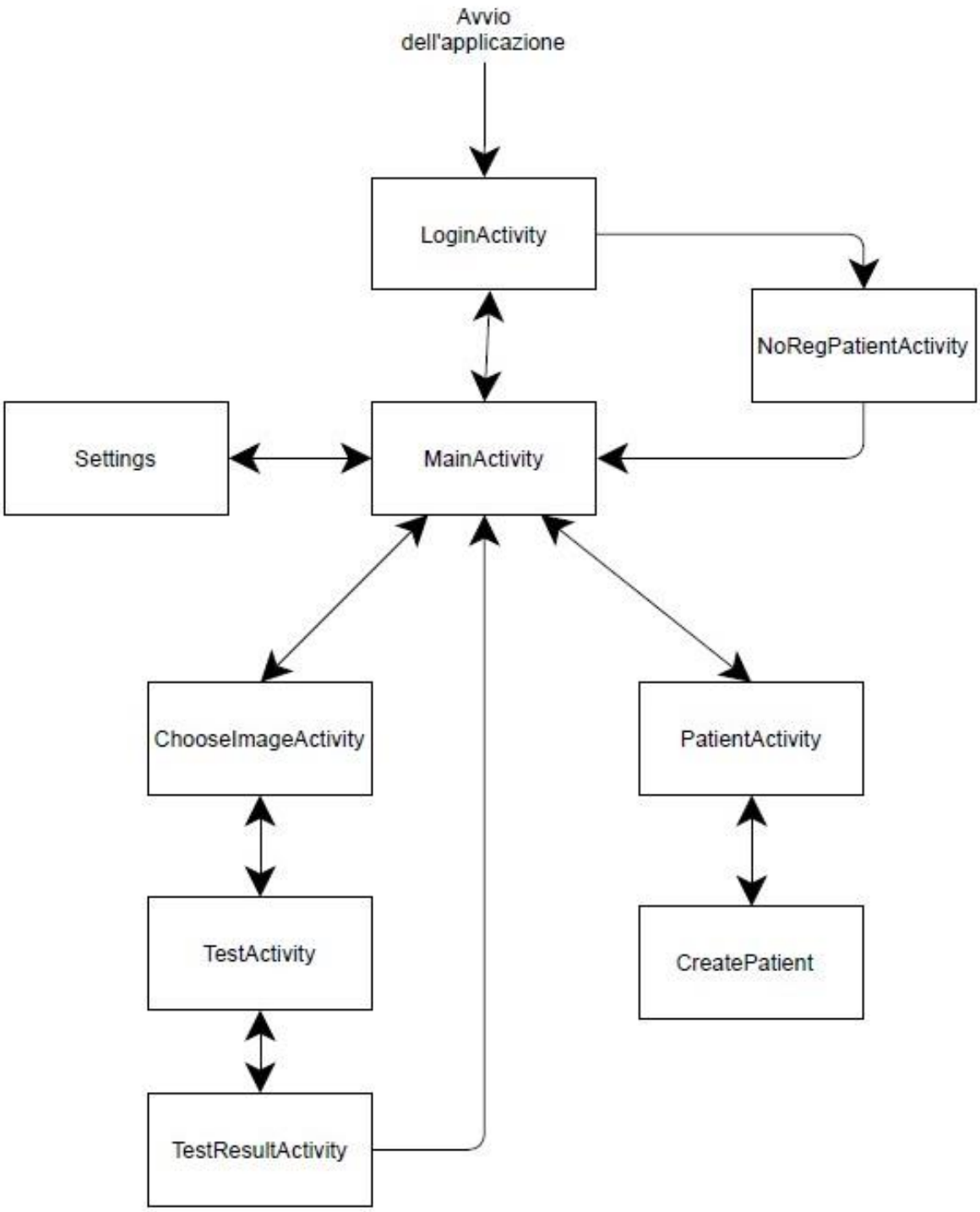


Figura 24: diagramma delle attività dell'applicazione

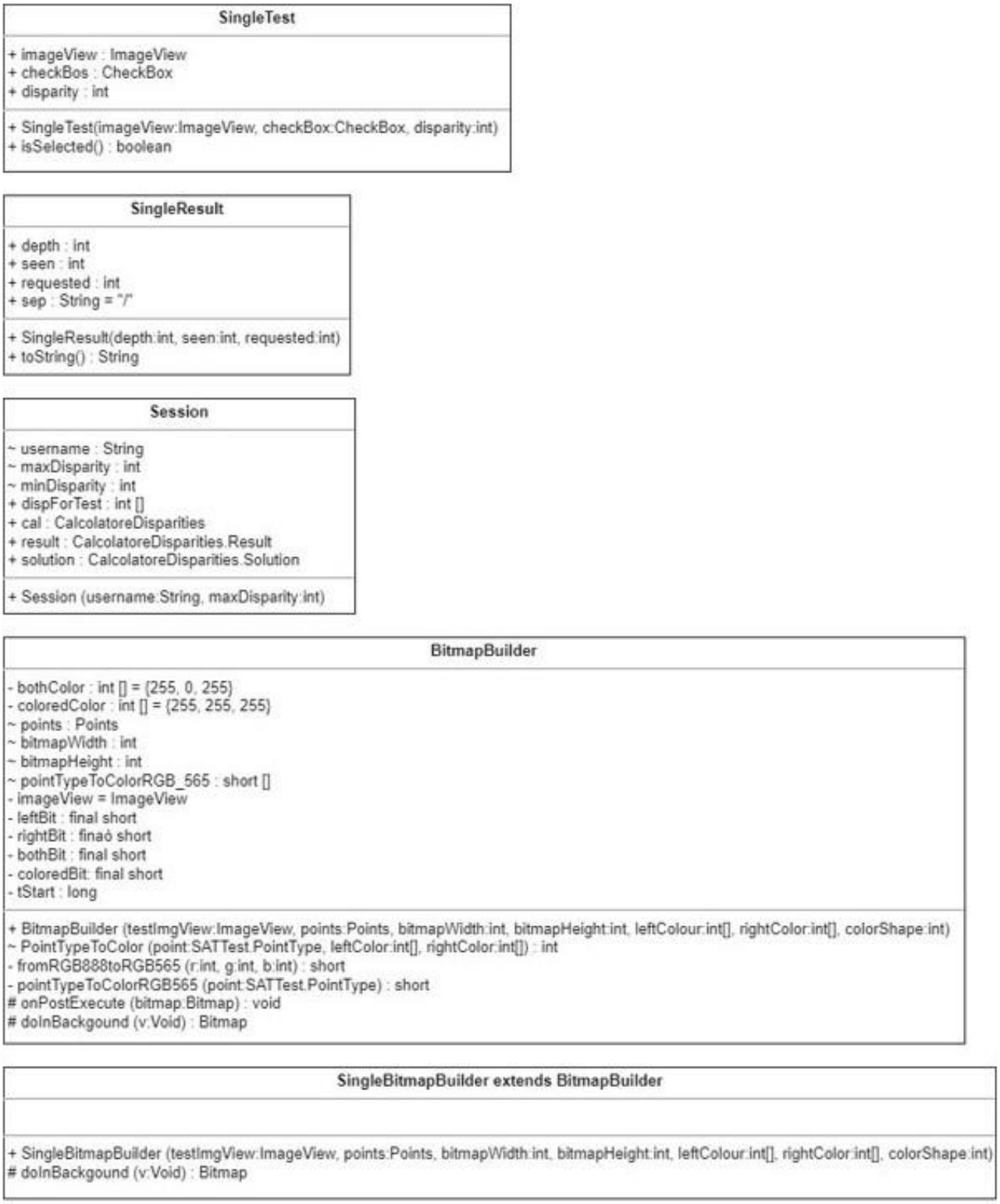
7.2.2 Il package utils

Questo package contiene una serie di classi il cui ruolo è di fondamentale importanza per lo svolgimento ed il funzionamento generale dell'applicazione.

Quelle che verranno elencate di seguito sono le più rilevanti ai fini del test

- DefaultValues contiene una serie di stringhe utilizzate per il salvataggio di valori nelle SharedPreferences del progetto e le costanti di default usate come standard dalle impostazioni
- CalcolatoreDisparities raggruppa in sé tutti i metodi che permettono il corretto funzionamento dell'algoritmo. Il suo funzionamento verrà descritto in seguito (sezione 8.1).
- Session ha in se l'oggetto CalcolatoreDisparities ed è quindi quella classe che dà inizio al test.
- BitmapBuilder permette la costruzione delle Bitmap contenenti le immagini di randots anaglifi tramite il costruttore e contiene al suo interno una serie di metodi per la manipolazione dei colori. A rigore, però, nel codice la costruzione viene fatta dalla classe SingleBitmapBuilder o BitmapParallelBuilder, due classi che estendono BitmapBuilder: la scelta su quale costruttore usare viene fatta in funzione del numero di processori della macchina. Per semplicità implementativa viene sempre usata SingleBitmapBuilder
- SingleResult è una semplice classe in cui si va ad istanziare un solo metodo "toString" il quale verrà richiamato nel momento in cui si dovranno mostrare a display i risultati per ogni valore di disparità nell'Activity TestResultsActivity.
- SingleTest è anch'essa piuttosto semplice. L'idea dietro questa classe è quella di raggruppare in un'unica istanza tre oggetti diverse: un intero rappresentate il valore di disparità, una ImageView ed una CheckBox. In questo modo nell'Activity TestActivity si avrà un vettore di sei SingleTest e da ogni elemento di quest'ultimo si potrà accedere ai relativi campi descritti sopra.

Di seguito viene riportato il diagramma delle classi presentate.


SingleTest
+ imageView : ImageView
+ checkBos : CheckBox
+ disparity : int
+ SingleTest(imageView:ImageView, checkBox:CheckBox, disparity:int)
+ isSelected() : boolean
SingleResult
+ depth : int
+ seen : int
+ requested : int
+ sep : String = "/"
+ SingleResult(depth:int, seen:int, requested:int)
+ toString() : String
Session
~ username : String
~ maxDisparity : int
~ minDisparity : int
+ dispForTest : int []
+ cal : CalcolatoreDisparities
+ result : CalcolatoreDisparities.Result
+ solution : CalcolatoreDisparities.Solution
+ Session (username:String, maxDisparity:int)
BitmapBuilder
- bothColor : int [] = {255, 0, 255}
- coloredColor : int [] = {255, 255, 255}
~ points : Points
~ bitmapWidth : int
~ bitmapHeight : int
~ pointTypeToColorRGB_565 : short []
- imageView = ImageView
- leftBit : final short
- rightBit : finaò short
- bothBit : final short
- coloredBit: final short
- tStart : long
+ BitmapBuilder (testImgView:ImageView, points:Points, bitmapWidth:int, bitmapHeight:int, leftColour:int[], rightColor:int[], colorShape:int)
~ PointTypeToColor (point:SATTest.PointType, leftColor:int[], rightColor:int[]) : int
- fromRGB888toRGB565 (r:int, g:int, b:int) : short
- pointTypeToColorRGB565 (point:SATTest.PointType) : short
onPostExecute (bitmap:Bitmap) : void
doInBackgound (v:Void) : Bitmap
SingleBitmapBuilder extends BitmapBuilder
+ SingleBitmapBuilder (testImgView:ImageView, points:Points, bitmapWidth:int, bitmapHeight:int, leftColour:int[], rightColor:int[], colorShape:int)
# doInBackgound (v:Void) : Bitmap

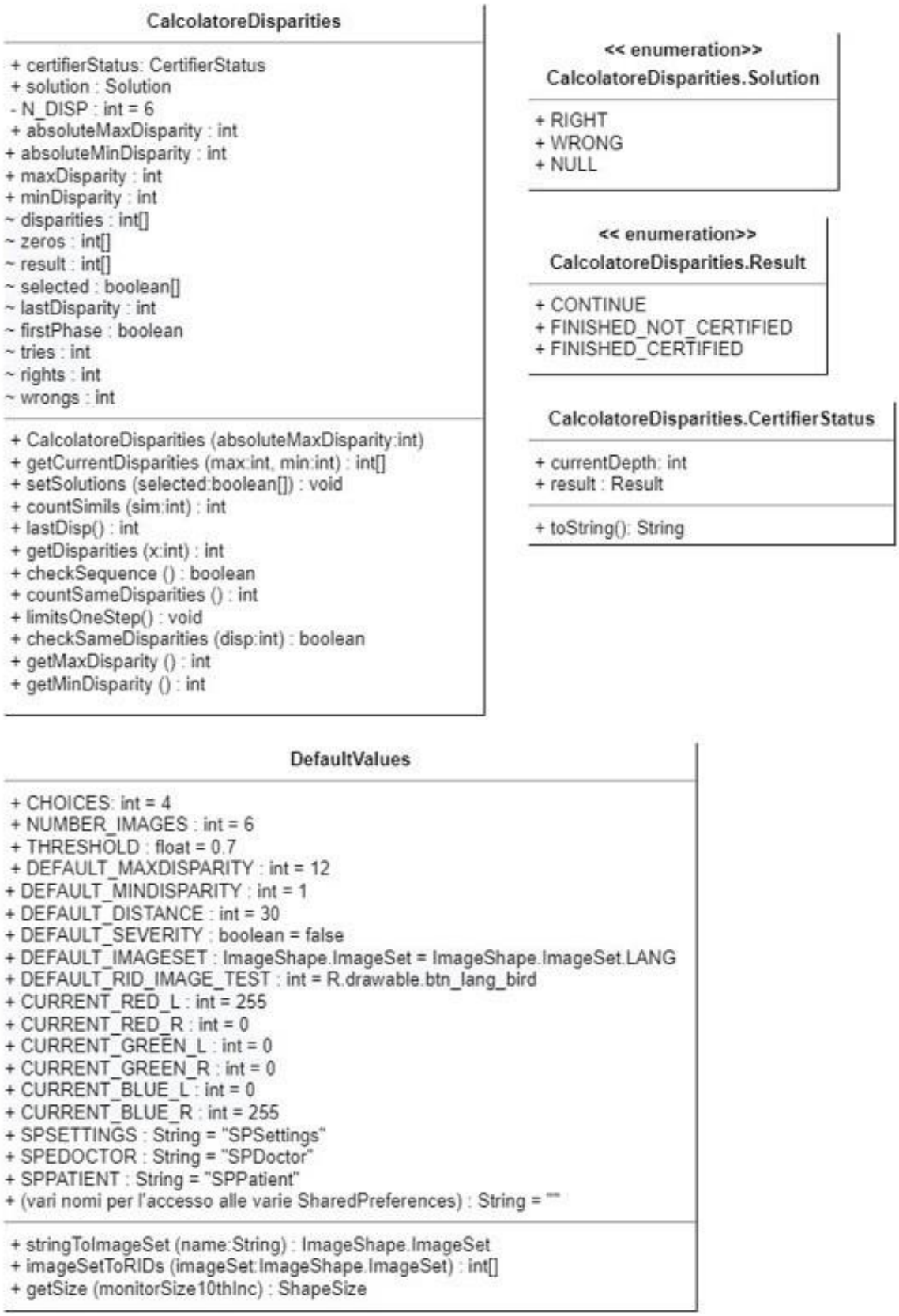


Figura 25: diagramma delle classi nel package .utils descritte

8 Algoritmi fondamentali

In questo capitolo verranno presentati i due algoritmi relativi alla sessione di test: quello che permette l'evoluzione del test stesso e quello di salvataggio dei risultati.

8.1 CalcolatoreDisparities

Si tratta di una classe contenuta nel package .utils nella quale sono stati implementati i metodi che permettono l'avanzamento del test nella TestActivity.

Per prima cosa si è pensato al funzionamento dell'algoritmo di test "su carta" e solo in seguito si è passati alla scrittura e al testing di tutti i metodi in Eclipse.

Solo quando la fase di testing ha portato un livello sufficientemente elevato di fiducia nell'algoritmo si è deciso, con le opportune modifiche, di importarlo nel progetto in Android Studio.

Nella TestActivity viene creato un oggetto "Session" che riceve in input il nome del paziente e la disparità massima con cui si intende eseguire il test. Questo secondo parametro viene usato per creare un oggetto CalcolatoreDisparities che determinerà, ad ogni iterazione del test, i valori di disparità da mostrare nelle sei immagini dell'Activity. Il costruttore di questa classe non fa altro che inizializzare il valore massimo di disparità uguale al valore passato come parametro ed il valore minimo di disparità pari a 1.

Per mostrare alcuni esempi del calcolo di disparità si considererà una disparità massima di 12.

Quando nella TestActivity si sta eseguendo il test in versione non "demo", viene chiamato il metodo "test()", il quale chiama a sua volta il metodo "getCurrentDisparities()" dell'oggetto CalcolatoreDisparities: con questo si vanno a settare i sei valori di disparità che si useranno in un particolare passo del test. Questo metodo restituisce un vettore di sei interi con le seguenti caratteristiche:

- La metà dei valori sono settati a 0
- L'altra metà viene calcolata in funzione delle disparità massima e minima valutate in quella precisa iterazione del test

È di fondamentale importanza evidenziare come, grazie all'ausilio del metodo "random()" della classe "Math" di java, si riesce a distribuire casualmente i sei valori all'interno del vettore. Questo evita che, per esempio, i valori di disparità diversi da 0 siano sempre i primi o gli ultimi tre valori del vettore.

Ciascun valore di disparità calcolato è assegnato in ordine a ciascuno dei sei oggetti "SingleTest". In questo solo nelle tre "ImageView" corrispondenti a "SingleTest" che hanno un valore diverso da 0 per disparità si potrà vedere l'immagine di test; negli altri tre casi non sarà riconoscibile nessuna immagine.

A questo punto il paziente deve osservare tutte le immagini e selezionare quelle che crede contengano l'immagine di test. Una volta fatto ciò deve premere sul pulsante "Submit" (o "Skip" se non riesce a visualizzare nessuna immagine): a questo punto viene richiamato nell'oggetto CalcolatoreDisparities il metodo "setSolutions(boolean[] selected)".

Questo è il metodo che valuta l'esecuzione del test in funzione di un vettore di booleani passato come parametro e del vettore di disparità calcolato precedentemente.

Come prima cosa l'algoritmo calcola:

- Quante volte le disparità diverse da 0 e uguali a 0 sono state cliccate
- Il più piccolo valore di disparità che viene visto, che chiameremo disparità ultima

Questi tre parametri saranno fondamentali per controllare l'esecuzione del test.

Il primo controllo che viene fatto sull'esecuzione dell'algoritmo riguarda la sequenza delle risposte date: esso viene eseguito dal metodo "checkSequence()" il quale controlla che disparità diverse da 0 siano state selezionate in sequenza. Questo viene effettuato tramite un ciclo for che è interrotto se l'algoritmo riconosce che non è stato selezionato un valore di disparità maggiore della disparità ultima. Ad esempio, se il vettore di disparità contiene i valori 12 - 7 – 1 si possono verificare i seguenti casi:

- Se vengono visti i valori 12 oppure 12 -7 oppure 12 -7 -1, indipendentemente dal numero di disparità a 0 viste, il controllo è superato
- Negli altri casi (7 oppure 7 – 1 oppure 12 – 1,…) il controllo non è superato e si identifica la soluzione di quel passo del test come "WRONG", ovvero errata.

Successivamente si passa a controllare i valori calcolati dall'algoritmo presenti nel vettore di disparità in quel momento del test. Anche in questo caso si possono identificare due situazioni:

- Se i valori massimo e mediano coincidono il test viene dichiarato finito
- Se quei due valori sono diversi si prosegue con il test e si calcolano nuovi valori da inserire nel vettore di disparità.

Superato il controllo di sequenza delle risposte, si passa a controllare in che "fase" del test ci si trova in quel momento. Vengono definite due fasi:

- La prima si ha quando tutti i valori di disparità, eccetto gli 0, calcolati sono diversi tra loro
- La seconda, invece, si ha quando nel vettore di disparità sono presenti due o tre valori di disparità uguali.

Se si è nella prima fase si possono verificare tre situazioni in funzione del numero di disparità a 0 e diverse da 0 viste:

- Se il numero di zeri visti è minore o uguale a 1 e il numero di disparità viste è maggiore o uguale a 1, allora la soluzione del test viene etichettata come "RIGHT", cioè corretta.
- Se il numero di disparità viste è 0, la soluzione viene indicata come "NULL", un valore di default che identifica il fatto di non aver riconosciuto nessuna immagine
- Se il numero di disparità a 0 viste è maggiore di 1 allora la soluzione viene settata a "WRONG", in quanto, indipendentemente dalle immagini viste, si è dichiarato di vedere delle immagini anche dove queste non sono presenti. Ciò è stato fatto per evitare il caso in cui tutte le immagini vengano selezionate e che quindi si dichiari che in tutte le "ImageView" è stata vista l'immagine di test: come è stato spiegato in precedenza solo in tre di queste ultime sarà presente un'immagine e l'obiettivo è di riconoscere in quali delle sei è presente l'immagine.

Comunque si classifichi la soluzione, in questa prima fase lo stato del test viene settato a "CONTINUE" e vengono calcolati nuovi valori di disparità in funzione di ciò che è stato

selezionato. In questo modo si passa all'iterazione successiva del test con nuovi valori di massimo e di minimo.

Se, invece, si è nella seconda fase, bisogna fare un controllo su quanti valori di disparità uguali sono presenti nel vettore.

Se tale valore è 2 (ad esempio 7 -7 – 6), la prima cosa che si esegue è un ulteriore controllo per vedere se entrambe e sole le disparità di ugual valore sono state viste. Successivamente si possono verificare tre casi:

- Se sono state viste solo le due disparità uguali e il numero di disparità a 0 viste è minore o uguale a 1, la soluzione viene settata a "RIGHT" e il test è concluso. Infatti si può affermare che oltre quel valore di disparità il paziente non riconosce più l'immagine di test. Questo è supportato dal fatto che quando il vettore di disparità contiene due valori uguali, il terzo è, per costruzione, un'unità in meno di tale valore. Se il paziente, eseguendo il test correttamente, vede le due disparità uguali, ma non quella diversa, allora si può concludere il test.

- Se vengono viste tutte e tre le disparità diverse da 0 ed un numero minore o uguale a 1 di disparità a 0, allora la soluzione del test è "RIGHT" e il suo stato viene settato a "CONTINUE". Infatti, dato che il paziente ha riconosciuto l'immagine con disparità leggermente inferiore a quelle di egual valore, gli si propone un nuovo test in cui il valore di disparità massimo sarà uguale al precedente valore minimo ed il nuovo valore minimo di disparità sarà uguale a 1 (con il vettore presentato precedentemente il nuovo test avrebbe valori 6 – 4 – 1).

- In tutti gli altri casi (numero di disparità a 0 viste maggiore di 1 oppure numero di disparità diverse da 0 viste minore o uguale a 1 oppure controllo sulla visione delle immagini con uguale disparità terminato con insuccesso) si classifica la soluzione come "WRONG", ma bisogna distinguere due casi:

    - Se la disparità mediana del vettore sia uguale al massimo valore di disparità del test (ad esempio 12 – 12 – 11), allora il test si considera finito in quanto non è più possibile calcolare vettori di disparità che rendano il test più semplice diversi da quest'ultimo

- Altrimenti lo stato del test risulta “CONTINUE” e si calcola un nuovo valore di disparità con valore massimo uguale al massimo possibile per quel test (12 in questi esempi) e valore minimo uguale al valore massimo del precedente vettore di disparità, ovvero ai due valori di disparità uguali presenti nel vettore (dato il vettore di esempio 7 – 7 – 6 si avrebbe dunque 12 – 10 – 7).

Se il numero di disparità diverse da 0 uguali presenti nel vettore è 3 bisogna fare un appunto: per come è stato costruito l’algoritmo, in questo ramo d’esecuzione si entra solo nel particolare caso in cui tali valori sono tutti a 1: si avrà quindi un vettore di disparità composto da tre 1 e tre 0.

Per prima cosa si controlla nuovamente se tutte le disparità a 1 sono state viste. La soluzione del test è “RIGHT” solo se tutte e tre le disparità a 1 sono viste ed il numero di disparità a 0 viste è minore o uguale a 1, altrimenti la soluzione diventa “WRONG”.

Però, essendo che 1 è il valore di disparità più piccolo del test e viene usato per indicare il test più difficile in assoluto, si forniscono al paziente tre tentativi per eseguire correttamente il test due volte. Da questa osservazione si deducono le seguenti possibilità:

- Se l’utente sbaglia il test, ma ha ancora dei tentativi, lo stato viene messo a “CONTINUE” e gli si presentano altre sei immagini con ancora tre valori di disparità a 1 e tre valori a 0.
- Se l’utente sbaglia il test e ha terminato i tentativi, lo stato viene messo a “CONTINUE”, ma il vettore di disparità viene calcolato con un nuovo massimo uguale al massimo del test ed il minimo uguale a 1.
- Se il test è svolto correttamente per la seconda volta, il test finisce.
- Se è stato svolto correttamente per la prima volta, lo stato del test è settato a “CONTINUE” e si propone un nuovo test con lo stesso vettore di disparità con tre 0 e tre 1.

Date queste situazioni si può evincere che l’utente dovrà effettuare due o tre tentativi con questo particolare vettore di disparità il quale verrà modificato solo nel caso in cui si risponda in modo errato a due (non necessariamente consecutivi) di questi test.

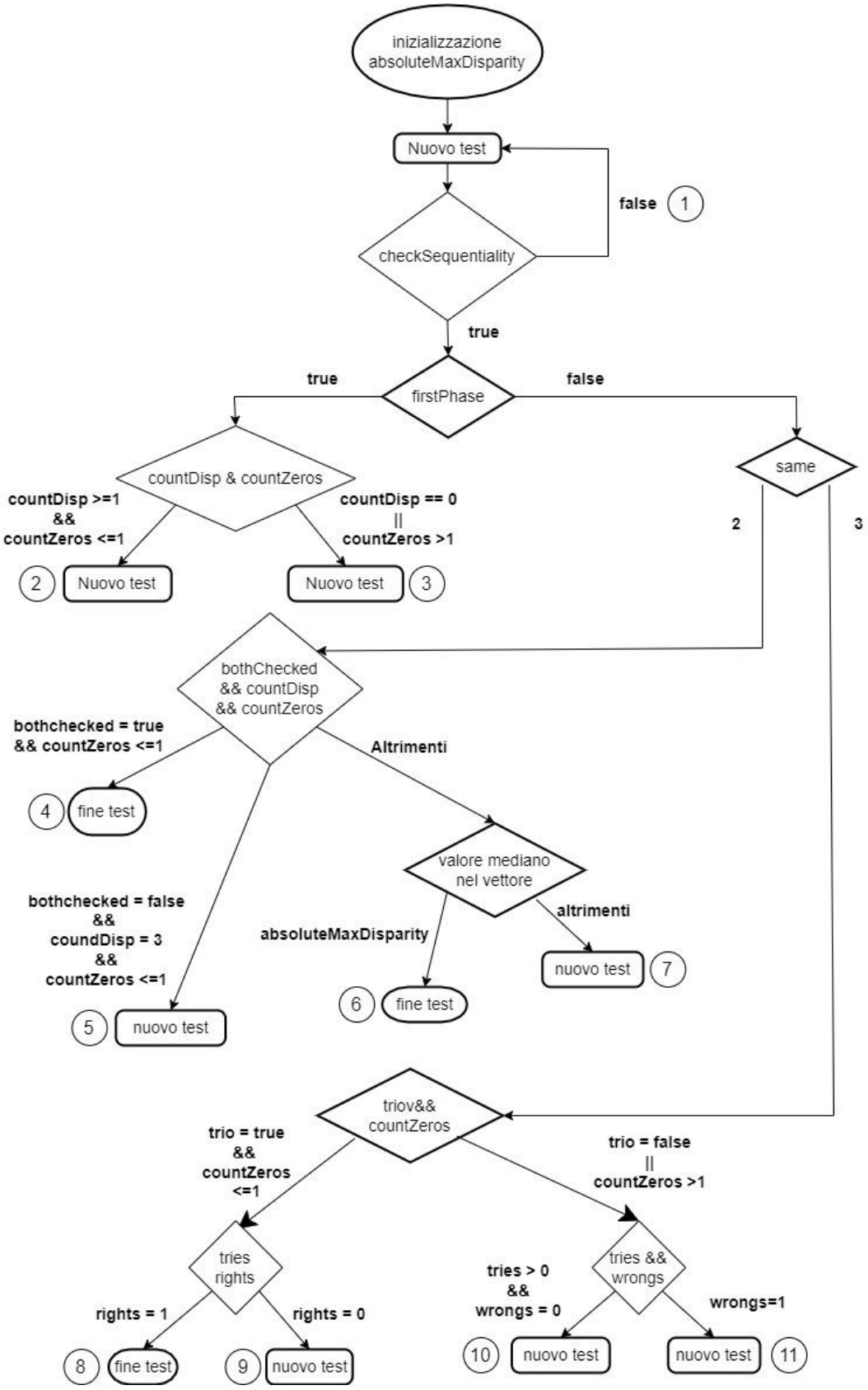


Figura 26: diagramma del funzionamento di CalcolatoreDisparities

Per questioni di leggibilità dello schema, si riporta di seguito una legenda che spiega:

- Il significato dei blocchi usati
- Il contenuto delle variabili scritte nei blocchi romboidali
- I vari scenari rappresentati da un numero a fianco di alcuni blocchi

8.1.1 I blocchi usati

Per questo diagramma si sono usati i seguenti blocchi con i relativi significati:

- L'ellisse rappresenta l'inizializzazione del valore massimo di disparità di test.
- Il rombo rappresenta la valutazione dell'espressione scritta al suo interno: il risultato farà proseguire l'algoritmo secondo un certo ramo.
- I rettangoli con angoli smussati rappresentano la creazione di un nuovo test con nuovi valori di disparità in funzione dello scenario in cui ci si trova (sezione 8.1.3).
- Le ellissi "appiattite" lungo l'asse di simmetria principale servono per indicare la fine del test.

8.1.2 Le variabili

Le variabili usate nei blocchi a forma di rombo hanno i nomi uguali a quelle presenti nella classe CalcolatoreDisparities:

- "checkSequentiality" è un booleano il quale controlla che la selezione delle checkbox sia fatta secondo un criterio di sequenzialità dei valori di disparità.
- "firstPhase" è un booleano che analizza i valori presenti nel vettore e valuta in che momento del test ci si trova: se con valori tutti diversi o con due o tre valori uguali di disparità.
- "countDisp" e "countZeros" sono 2 variabili intere che servono per contare rispettivamente il numero di disparità e di zeri cliccati (avendo fatto prima un controllo sulla sequenzialità le disparità cliccate seguono tale criterio).

- “same” è una variabile booleana che conta quanti valori di disparità uguali ci sono nel vettore.
- “bothChecked” è un booleano che valuta se sono state cliccate solo le checkbox corrispondenti ai 2 valori di disparità uguali quando “same” = 2.
- “trio” è l'equivalente di botchecked, ma considera 3 valori uguali quando “same” = 3 (quindi valuta se tutte le disparità a 1 sono state viste).

8.1.3 Gli scenari

L’avanzamento o la conclusione del test può avvenire in diversi modi, sia in funzione delle risposte date dall’utente che dalla posizione nel diagramma durante quella specifica iterazione del test:

1. Nuovo test con “massimo=absoluteMaxDisparity” e “minimo=valore mediano nel vettore”.
2. Nuovo test con “nuovo massimo” e “nuovo minimo” calcolati in funzione delle checkbox cliccate.
3. Nuovo test con “massimo=absoluteMaxDisparity” e “minimo=valore mediano del vettore”.
4. Fine test
5. Nuovo test con “nuovo massimo=minimo del vettore vecchio” e “nuovo minimo=1”.
6. Fine del test.
7. nuovo test con “nuovo massimo=absoluteMaxDisparity” e “nuovo minimo=valore corrispondente ai due valori di disparità uguali nel vettore”.
8. Fine test
9. Nuovo test con tre disparità a 0 e tre a 1 (aumento “rights” e diminuisco “tries”).
10. Nuovo test con tre disparità a 0 e tre a 1 (aumento “wrongs” e diminuisco “tries”).

11. Nuovo test con “massimo=absoluteMaxDisparity” e “minimo=1”.

8.2 Il salvataggio dei risultati

In quest’applicazione un risultato è un oggetto della classe “SingleResult”. Esso è composto da tre interi:

- Uno per indicare il valore di disparità del risultato
- Uno per il numero di volte che una disparità viene chiesta dall’algoritmo
- Uno per il numero di volte che una disparità viene valutata come “vista”

Il risultato assocerà una certa disparità ad una frazione e, affinchè un valore di disparità venga certificato, tale rapporto deve essere superiore ad una determinata soglia.

Durante la creazione di TestActivity vengono creati due vettori, “seen” e “requested”,il cui compito è, rispettivamente, quello di tenere traccia del numero di volte che una disparità viene classificata come vista e del numero di volte che una disparità viene chiesta dall’algoritmo, andando ad incrementare di uno la i-esima cella del vettore corrispondente al valore di disparità i.

I vettori hanno la stessa dimensione, pari al valore di disparità massima del test più uno: l’indice “i” tiene traccia di tutti i valori di disparità da 0 fino al massimo.

I due vettori vengono aggiornati in occorrenze diverse:

- “requested” ad ogni iterazione del test. In particolare incrementa di uno le celle la cui posizione corrisponde al valore di disparità calcolato dall’algoritmo e aumenta di tre la cella di posizione zero.
- “seen” viene aggiornato in funzione del tipo di test e delle risposte che vengono date dal paziente:
    - Se il test è “permissivo” i valori delle celle del vettore in questione vengono modificati in base alle disparità presenti nelle immagini che il paziente dichiara di vedere.

- Se il test è “severo” le modifiche al vettore vengono apportate solo nel caso in cui la soluzione del test sia classificata come “RIGHT”.

Terminata l’attività di test, prima di passare alla TestResultActivity, vengono eseguite le seguenti operazioni:

- Si crea un JSONArray nel quale si mettono le stringhe contenenti i risultati per tutte le disparità da 0 fino al valore massimo di test.
- Si certifica il minimo valore di disparità come risultato del test: questo viene fatto andando a calcolare in un ciclo for (da zero fino al valore di massima disparità di test) i rapporti “seen” su “requested” per ogni valore di disparità. Quando un risultato di questa divisione fornisce un numero maggiore o uguale di 0.7, vuol dire che quel valore di disparità è il più piccolo che è stato visto almeno tre quarti delle volte che è stato presentato nelle immagini. Tale valore non è casuale: infatti, ragionando su carta, si è pensato che quando il paziente raggiunge la fase con vettore di disparità costituito da tre 0 e tre 1 nel test “severo” nel minor numero di passi possibili (quindi dopo solo un’iterazione del test) e porta a compimento il test eseguendo correttamente due tentativi e sbagliandone uno, il numero di volte che il valore 1 è richiesto è dieci (uno alla prima iterazione e tre volte per tre tentativi) ed il numero di volte che la cella di posizione 1 del vettore “seen” è aggiornata è sette (uno alla prima iterazione e sei volte nei soli due tentativi corretti).
- Dato il valore massimo di disparità certificato, si computa l’angolo di acuità visiva, sfruttando le dimensioni del dispositivo ed il parametro di distanza posto nei settings prima di eseguire il test.

Successivamente, tramite meccanismi di “Intent” e “SharedPreferences” (due modi per passare dati tra Attività in Android Studio), si passa alla TestResultActivity.

Nell’intestazione della pagina vengono presentati i seguenti parametri:

- Nome, cognome e ID del paziente che ha effettuato il test
- Il tipo di test eseguito
- Il set di immagini e l’immagine scelta per il test

- L’angolo finale calcolato per il corrispondente valore massimo di disparità certificato
- La data e l’ora in cui il test viene eseguito

Scorrendo nella pagina, poi, si trova una lista dei risultati delle singole disparità. Ciascuna riga mostra una frazione per un certo valore di disparità: da notare come queste frazioni rappresentino sempre un numero compreso tra 0 (disparità mai vista) e 1 (disparità vista tutte le volte che l’algoritmo l’ha richiesta). Nel caso particolare in cui una disparità non è mai stata chiesta il risultato viene espresso come 0/0 e non è d’interesse per capire il valore più piccolo di disparità visto. Inoltre, questo elenco è formato da un numero di righe pari al massimo valore di disparità inserito nei settings più uno: di fatti, il valore 0 viene utilizzato per calcolare il numero di volte in cui l’utente afferma di vedere la figura in un’immagine vuota.

Infine, questi risultati vengono salvati in un file interno del dispositivo. Dato che il file di un utente dipende dal suo ID, anche in questo frangente si possono distinguere due situazioni:

- Se il paziente ha fatto l’accesso all’applicazione senza che un dottore abbia effettuato il login, il suo ID di default viene settato a 0. Ciò vuol dire che tutti i pazienti che svolgono il test in questa modalità troveranno i propri risultati salvati nel medesimo file di storage.
- Se, invece, il dottore ha eseguito il login nell’applicazione e scelto un paziente tra quelli della lista, l’ID di quest’ultimo sarà univoco per il relativo dottore. Questo fa si che ogni paziente abbia un suo file dedicato nel quale vengono ricapitolati tutti i risultati di tutti i suoi test.

9 Manuale per l'utente

Come si è evidenziato più volte in questa trattazione, l'utente principale di StereoAcuityMultiTestApp è il paziente che dovrà sottoporsi all'attività di test per diagnosticare la presenza o meno di ambliopia. Il suo accesso all' applicazione può essere mediato o meno dalla presenza di un medico:

- Nel primo caso ad effettuare il login è il dottore tramite una sua mail ed una sua password e sarà suo compito selezionare il paziente in questione prima di permettergli di eseguire il test.
- Nel secondo caso è il paziente stesso che, inserendo i propri nome e cognome, effettuerà l'accesso nell'applicazione.

Una volta definito il paziente è possibile, prima di iniziare il test, modificare i parametri di quest'ultimo tramite la schermata di "Settings". Di default questi sono :

- Un valore di distanza pari a 30 cm.
- Un valore di disparità massima di 12.
- La severità del test off (quindi di default si eseguirà un test "permissivo".
- La lente destra dell'anaglifo impostata sul colore blu.
- La lente sinistra dell'anaglifo impostata sul colore rosso.

Definiti anche i settaggi, si passa alla scelta dell'immagine con cui il paziente vorrà eseguire il test. Di default si ha l'immagine "bird" del set "Lang".

A questo punto si può passare alla sezione relativa all'esecuzione del test. Per prima cosa si svolge una piccola attività di "demo" nella quale l'utente deve capire il funzionamento del test: selezionare le "CheckBox" relative alle "ImageView" nelle quali si vede l'immagine scelta.

Compreso il lavoro da svolgere si può dar via all'attività di test vera e propria. Per questa fase è richiesto al paziente di indossare gli anaglifi: senza di questi, infatti, sarebbe impossibile distinguere la presenza o l'assenza dell'immagine di test nelle sei "ImageView" che vengono costruite appositamente per l'attività di test.

Il test prosegue in funzione delle risposte che il paziente fornisce, basate sulla capacità o meno di distinguere le “ImageView” contenenti l’immagine di test da quelle che ne sono sprovviste. Si è calcolato che il numero minimo di iterazioni del test è tre ed in questo caso il test certifica l’assenza di ambliopia nel paziente.

Terminata l’attività, l’utente verrà mandato nella sezione dei risultati relativi al test appena compiuto. Qui potrà vedere il più basso valore di disparità che è stato certificato dall’algoritmo ed i singoli risultati per ogni valore di disparità da 0 fino al massimo inserito nei settings. I risultati, poi, saranno salvati in uno specifico file per permettere al dottore o al paziente di consultarli successivamente.

Visionati i risultati, si torna al menù principale dell’applicazione e da qui è possibile fare il logout dell’applicazione o eseguire un nuovo test.

10 Conclusioni e possibili sviluppi futuri

Il principale scopo dell'applicazione sviluppata è quello di fornire un'ulteriore metodologia di trattamento per l'ambliopia.

In particolar modo, l'utilizzo di un videogioco dovrebbe rendere maggiormente appetibile la terapia per il paziente, superando così alcuni degli ostacoli spesso incontrati dai metodi che prevedono l'impiego dell'atropina o del patching: di fatti si fa leva sull'adrenalina che si sviluppa in un bambino nel momento in cui gli viene chiesto di vincere ad un determinato gioco interattivo. In questo modo si cerca di traslare tutta la sua attenzione solo sull'attività di test.

Con questa applicazione, inoltre, si sono rispettati a pieno i requisiti del progetto 3D4amb di semplicità del software, intuitività nell'utilizzo, basso costo e divertimento per l'utente.

Il modo in cui è stato implementato l'algoritmo di test permette, inoltre, di avere sempre un test adattivo in funzione delle risposte che il paziente fornisce: questo si traduce in una riduzione elevata dei tempi d'esecuzione del test.

Personalmente credo che l'informatica abbia grandissime potenzialità nelle svariate terapie e nel supporto ai pazienti, e questa mia app ne è solo una piccola dimostrazione: spero che questo mio lavoro contribuisca alla sensibilizzazione del pubblico nei confronti delle possibilità che queste tecnologie offrono, oltre che della prevenzione e cura della malattia. Colgo l'occasione per sottolineare l'importanza della pratica che ho sin qui trattato: come per molte altre malattie, la prevenzione non sarà mai sottolineata abbastanza.

11 Bibliografia e sitografia

- Cline D., Hofstetter H.W., Griffin J.R., Dictionary of visual space, 4ª edizione, Boston, Butterworth-Heinemann, 1996, p.820
- McConaghy, J. R., & McGuirk, R. (2019) Amblyopia: Detection and treatment. American Family Physician, 100(12), 745-750.
- Alatawi, A., Alali, N., Alamrani, A., Hashem, F., Alhemaidi, S., Alreshidi, S., & Albalawi, H. (2021). Amblyopia and Routine Eye Exam in Children: Parent's Perspective. Children, 8(10),1–12.
- Kanonidou, E. (2011) Amblyopia: a mini review of the literature. Int Ophthalmol 31, 249–256
- Birch, E.E., Kelly, K.R. & Wang, J. (2021) Recent Advances in Screening and Treatment for Amblyopia. Ophthalmol Ther .
- Dennis S.C. Lam, Jianhao Zhao, Li Jia Chen, Yunxiu Wang, Chongren Zheng, Qiaoer Lin, Srinivas K. Rao, Dorothy S.P. Fan, Mingzhi Zhang, Ping Chung Leung, Robert Ritch (2011) Adjunctive Effect of Acupuncture to Refractive Correction on Anisometropic Amblyopia: One-Year Results of a Randomized Crossover Trial, Ophthalmology, Volume 118, Issue 8, 1501-1511.
- Diana DeSantis, (2014) Amblyopia, Pediatric Clinics of North America, Volume 61, Issue 3, 505-518.

https://www.proquest.com/scholarly-journals/amblyopia-detection-treatment/docview/2454213493/se-2?accountid=31774

https://doi-org.humanitas.clas.cineca.it/10.3390/children8100935

https://doi-org.humanitas.clas.cineca.it/10.1007/s10792-011-9434-z

https://www.humanitas.it/malattie/ambliopia-occhio-pigro/

https://doi-org.humanitas.clas.cineca.it/10.1007/s40123-021-00394-7

https://www.uspreventiveservicestaskforce.org/Page/Document/UpdateSummaryFinal/visual-impairment-in-children-ages-1-5-screening

https://doi.org/10.1016/j.ophtha.2011.01.017.

https://www.sciencedirect.com/science/article/pii/S0031395514000212

12 Ringraziamenti

Per la prova finale della Laurea Triennale in ingegneria informatica “GAMIFICATION DI UN TEST INTERATTIVO PER L’ACUTEZZA DELLA VISIONE BINOCULARE” ringrazio il relatore, prof. Gargantini Angelo, ed il correlatore, dott.ssa Bonfanti Silvia, sia per l’opportunità di approfondire gli argomenti trattati che per la disponibilità e per l’attenzione con le quali mi hanno accompagnato nel corso di tutto questo progetto. Ringrazio inoltre il progetto 3D4amb per la gentile concessione di materiale, fondamentale per la progettazione e per lo sviluppo del suddetto elaborato.