



Università degli Studi di Bergamo

SCUOLA DI INGEGNERIA

Corso di Laurea Triennale in Ingegneria Informatica

Classe n. L-8 Ingegneria dell'informazione (D.M. 270/04)

Ambliopia

Relatore

Prof. Angelo Gargantini

Tesi di Laurea Triennale

Frangiamore Luca

Matricola 1074443

Anno Accademico 2022–2023

Indice

1	Ambliopia	3
1.1	Diagnosi	3
1.2	Cause	4
1.3	Sintomi	5
1.4	Trattamenti	5
2	3D4Amb	9
2.1	Funzionamento e strumenti	9
2.2	Obiettivi	13
3	Progetto	15
3.1	Tecnologie utilizzate	16
3.2	Struttura	19
3.3	Implementazione	20
3.4	Problemi	27
4	Conclusioni e Sviluppi futuri	31

Elenco delle figure

1.1	Pediatric Vision Scanner.Fonte: [1]	4
1.2	Trattamento con Patching.Fonte: [2]	5
1.3	Trattamento con lente di bangerter. Fonte: [3]	6
1.4	Trattamento innovativo con Luminopia.Fonte: [4]	7
1.5	Pattern Flicker.Fonte: [6]	7
2.1	Logo 3D4Amb.Fonte: [7]	9
2.2	Cardboard di cartone.Fonte: [8]	10
2.3	Differenziazione colori. Fonte: [7]	10
2.4	Differenziazione elementi su schermo.Fonte: [7]	11
2.5	occhiali anaglifici.Fonte: [9]	12
2.6	Anaglifo.Fonte: [9]	12
2.7	Differenziazione elementi su schermo.Fonte: [10]	13
3.1	Layout gamepad utilizzato.Fonte: [11]	15
3.2	Assi Giroscopio.Fonte: [12]	16
3.3	Schermate unity	17
3.4	Esempio di script realizzato con visual code	18
3.5	Esempio di trasparenza mediante un oggetto di gioco	19
3.6	Target con punteggio	20
3.7	Target in movimento	20
3.8	Scena prototipale	21
3.9	Codice usato	22
3.10	Mappatura tasti	23
3.11	Vista del menu nella finestra <i>Scene</i>	24
3.12	Vista del menu nella finestra <i>simulator</i>	24
3.13	Organizzazione script	26
3.14	Secondo livello	27
3.15	Esempio fly score	27
3.16	Albero low poly contro high poly.Fonte: [13]	28
3.17	Prova controller test, per tasto X	29

Introduzione

Nel corso della seguente trattazione, andrò a descrivere il processo che mi ha portato alla realizzazione di un gioco mobile all'interno del progetto 3D4Amb, condotto dall'Università di Bergamo e diretto dal professore Angelo Gargantini. L'obiettivo del progetto è quello di creare applicazioni innovative per la diagnosi e la cura dell'ambliopia, una patologia visiva che colpisce principalmente i bambini e che consiste in una riduzione dell'acuità visiva di un occhio, che non riesce a lavorare correttamente a causa di un problema nella trasmissione dei segnali visivi al cervello.

In particolare, il progetto 3D4Amb si prefigge di utilizzare la tecnologia 3D e gli strumenti di realtà virtuale per la creazione di applicazioni interattive che stimolino la visione dell'occhio pigro, sfruttando supporti fisici per differenziare le immagini dirette ai rispettivi occhi. L'obiettivo finale è quello di fornire una soluzione innovativa e personalizzata per la diagnosi e la cura dell'ambliopia, che possa essere utilizzata sia in ambito clinico che a casa.

Nel primo capitolo, approfondirò il problema dell'occhio pigro, descrivendone i sintomi e i possibili trattamenti, evidenziando l'importanza di una diagnosi precoce e della cura della patologia per prevenire conseguenze a lungo termine. Nel secondo capitolo, descriverò il progetto 3D4Amb, illustrando gli strumenti e i supporti utilizzati per la creazione delle applicazioni, evidenziando l'importanza dell'innovazione tecnologica per la diagnosi e la cura dell'ambliopia.

Nel terzo capitolo, mi concentrerò sulla descrizione dettagliata del flusso di lavoro che mi ha portato alla realizzazione del gioco mobile all'interno del progetto, evidenziando i problemi incontrati e le soluzioni adottate per superarli. Infine, nell'ultimo capitolo, le conclusioni, esprimerò le mie considerazioni sul progetto e le possibili migliorie future per sviluppare soluzioni ancora più efficaci per la diagnosi e la cura dell'ambliopia.

Capitolo 1

Ambliopia

L'ambliopia, anche conosciuta come "occhio pigro" è una patologia che si manifesta nei primi anni dello sviluppo, età compresa tra 0-6 anni, il quale colpisce circa il 4% della popolazione mondiale. La patologia riguarda la non corretta stimolazione dell'apparato visivo, in questo caso il cervello non riesce a ricreare l'immagine tridimensionale e quindi per evitare il fenomeno della visione doppia, anche detta **diplopia**, il cervello tenterà di sopprimere un'immagine. Nella maggior parte dei casi, l'occhio è anatomicamente perfetto, **ambliopia funzionale**, in casi peggiori possiamo riscontrare **ambliopia organica** in cui vi è una deviazione delle vie ottiche. La seguente patologia si può manifestare con più probabilità in maniera asimmetrica, **monolaterale**, la quale colpisce un solo occhio, o più raramente possiamo trovare la forma **bilaterale**, la quale colpisce entrambi gli occhi. Normalmente l'ambliopia non peggiora durante la vita adulta in quanto oramai lo sviluppo alterato della vista si è oramai instaurato, l'età in cui l'ambliopia si stabilizza in modo permanente va dagli 8 ai 15 anni.

1.1 Diagnosi

La diagnosi è un punto fondamentale per il trattamento dell'ambliopia, in quanto, una diagnosi tardiva può influire negativamente sulla corretta terapia, riducendo le possibilità di recupero. Il trattamento in genere viene personalizzato dal medico oculistico.

- Classica, in cui si cerca di stabilire l'acuità visiva e la visione binoculare del piccolo paziente. In caso di strabismo sarà necessaria anche una visita ortottica, la quale, è una branca dell'oftalmologia che si occupa della valutazione e riabilitazione di deficit muscolari e sensoriali.
- Pediatric Vision Scanner, Negli ultimi anni si sta sviluppando un nuovo metodo in grado di riconoscere l'ambliopia in età pediatrica con alta precisione, figura:1.1. Come pubblicato nel articolo Validation of the Pediatric Vision Scanner in a normal preschool population [14], in cui si prende in esame Pediatric Vision Scanner(PVS), il quale scatta una foto ad entrambi gli occhi per poi andare a misurare errore di rifrazione e disallineamento.

Il metodo in questione, descritto nell'articolo sopra citato, ha permesso di identificare correttamente tutti e 6 i bambini con ambliopia e/o strabismo nel campione di 300 bambini, garantendo così una precisione del 100%. Inoltre, il metodo po-



Figura 1.1: Pediatric Vision Scanner. Fonte: [1]

trebbe essere in grado di diagnosticare l'ambliopia già a partire dai 2 anni di età, suggerendo un potenziale miglioramento nella diagnosi precoce di questa patologia.

1.2 Cause

In generale come detto prima, l'occhio pigro riguarda il progressivo trascuramento dei segnali di uno dei due occhi. Questo processo di sviluppo è causato da un non corretto sviluppo delle vie nervose degli occhi, le quali vengono stimulate in modo non bilanciate, ciò è dovuto magari dalla presenza di una condizione oculare presente in uno dei due occhi. Alcune condizioni che possono insorgere sono:

- Astigmatismo, visione poco nitida e distorta in qualsivoglia direzione.
- Strabismo, deviazione degli assi visivi, impedisce il corretto coordinamento degli occhi.
- Cataratta, opacizzazione parziale o totale del cristallino, causa offuscamento e difficoltà nel mettere a fuoco le immagini.
- Ptosi palpebrale, una o entrambi le palpebre superiori sono abbassate più del normale.

Come detto prima, se il cervello non riesce a combinare le immagini proveniente dai due occhi, esso può decidere di trascurare un dei due segnali, prediligendo l'occhio ottimale, sviluppando quindi l'ambliopia.

1.3 Sintomi

Tra i problemi più comuni causati dall'ambliopia, oltre alla ridotta acuità visiva dell'occhio colpito, ci sono:

- Scarsa percezione della profondità: l'ambliopia può ridurre la capacità di percepire la profondità poiché la coordinazione visiva tra i due occhi viene compromessa.
- Difficoltà di visione in un occhio, l'ambliopia può causare la soppressione dell'occhio colpito dal cervello, il che può portare a problemi di visione come la riduzione della percezione del contrasto, della percezione dei colori e della visione notturna.
- Movimenti involontari dell'occhio, l'ambliopia può causare movimenti oculari anormali, come lo strabismo, che a loro volta possono causare diplopia (visione doppia).
- Sensibilità al movimento compromessa, i pazienti affetti da ambliopia possono avere difficoltà a distinguere il movimento e possono essere più sensibili al movimento nell'occhio non colpito.

È importante diagnosticare e trattare l'ambliopia il prima possibile, poiché il trattamento è più efficace durante l'infanzia e l'adolescenza. In generale, il trattamento dell'ambliopia prevede l'uso di occhiali, l'occlusione dell'occhio sano per stimolare la vista dell'occhio pigro e la terapia visiva.

1.4 Trattamenti

Subito dopo aver riconosciuto il disturbo bisogna procedere con la corretta terapia. Come prima cosa, bisogna correggere il difetto che ha portato l'inibizione dell'occhio pigro, successivamente si procede con le terapie.

- Patching [15], La terapia consiste nel coprire l'occhio dominante, da applicare per un periodo di tempo variabile, figura:1.2. Il trattamento è efficace, ma il recupero della vista impiega diversi mesi.



Figura 1.2: Trattamento con Patching. Fonte: [2]

- Lenti opacizzazione, usate per limitare la visione dell'occhio sano e stimolare l'occhio pigro a lavorare di più, figura:1.3. Un articolo che ha esaminato gli effetti delle lenti di Bangerter sulla funzione binoculare in pazienti con ambliopia è The effect of Bangerter filters on binocular function in observers with amblyopia [16].



Figura 1.3: Trattamento con lente di bangerter. Fonte: [3]

- Collirio, permette di offuscare la vista dell'occhio dominante, in modo da poter stimolare l'occhio più debole
- Luminopia, è un software approvato da **Food and Drug Administration(FDA)** [17] come annunciato dal CEO Scott Xiao, che permette ai pazienti di poter usufruire di 700 ore di serie o film, adattati tramite AI, figura:1.4. Questo sistema permette di rendere il trattamento dell'ambliopia piacevole e quindi sopportabile nel lungo periodo. I pazienti usufruiscono di questi contenuti mediante il vr, in modo da poter visionare i contenuti scelti.
- Esiste poi un altro tipo di tecnologia, I-BiT Plus [5]s, il quale permette al paziente di intrattenersi mediante giochi interattivi o video. Il suo funzionamento si basa sul mostrare la parte più "interessante" del gioco o DVD, come ostacoli o obiettivi, mentre nell'occhio pigro vengono mostrati oggetti di contorno, come i paesaggi. Per assicurare che il paziente continui a guardare le immagini in un posizione ottimale il sistema si serve di un Eye tracker, in modo da garantire il corretto posizionamento dell'immagine.
- Pattern Flicker [6], è costituito da una cupola emisferica nera contenente un ottagono di LED rossi e una console utilizzata per controllare i LED stessi, figura:1.5.



Figura 1.4: Trattamento innovativo con Luminopia.Fonte: [4]



Figura 1.5: Pattern Flicker.Fonte: [6]

L'utilizzo di queste luci consente di stimolare l'occhio e i nervi ottici del bambino attraverso giochi interattivi e stimolazione mirata. Lo sfondo nero aiuta a evitare distrazioni o stimoli indesiderati, consentendo al paziente di concentrarsi sull'ottagono di LED. Questi emettono luci monocromatiche rosse a bande alternate, un colore al quale la macula è particolarmente sensibile, per bombardare in modo mirato e selettivo i fotorecettori, migliorando l'efficienza visiva. Tuttavia, è importante notare che l'utilizzo di questo trattamento potrebbe non essere sempre consigliabile, poiché in alcuni pazienti affetti da epilessia potrebbe scatenare crisi.

Capitolo 2

3D4Amb

Il progetto è nato con l'intenzione di, creare uno strumento utile ed economico per il trattamento, misurazione dell'occhio pigro basato sul 3D.



Figura 2.1: Logo 3D4Amb.Fonte: [7]

2.1 Funzionamento e strumenti

Il funzionamento di base del sistema 3D4Amb consiste nel presentare al paziente due immagini correlate tra loro ma leggermente diverse, pre-elaborate tramite il software 3D4Amb. Questo permette di creare una percezione tridimensionale della scena, dando la sensazione che gli oggetti si trovino davanti ai propri occhi. Per realizzare le applicazioni video ludiche basate sulla differenziazione delle immagini, 3D4Amb utilizza diversi supporti tecnologici, il tutto a basso costo. Tuttavia, per poter utilizzare il sistema, è necessario garantire lo sdoppiamento delle immagini per i due occhi, in modo che ciascun occhio veda una sola immagine, ma insieme creino l'effetto tridimensionale. Per garantire il corretto sdoppiamento, sono state sperimentate diverse soluzioni. Una di queste è l'utilizzo di occhiali speciali dotati di lenti polarizzate, che permettono ai due occhi di vedere immagini diverse, in base alla polarizzazione delle lenti. Un'altra soluzione consiste nell'utilizzo di schermi speciali, che proiettano due immagini diverse in modo che ciascun occhio possa vederne una sola. Inoltre, per garantire la massima efficienza del sistema, è importante considerare anche la distanza tra gli occhi del paziente, in modo da regolare il sistema di visualizzazione in base alle caratteristiche anatomiche di ciascun individuo. In questo modo, è possibile ottenere una visione tridimensionale più nitida e confortevole per il paziente.

Cardboard

Il funzionamento del Google Cardboard, figura:2.2, si basa sulla creazione di un effetto stereoscopico [18], utilizzando uno smartphone, il quale viene inserito all'interno del dispositivo. Il cardboard utilizza due lenti che permettono di ingrandire l'immagine fino ad riempire il campo visivo [19].



Figura 2.2: Cardboard di cartone. Fonte: [8]

Questo dispositivo ci permette di poter usufruire della tecnologia 3D con un costo relativamente basso, che si aggira sui 10\$, esso ci permette di mostrare immagini completamente diverse hai due occhi.

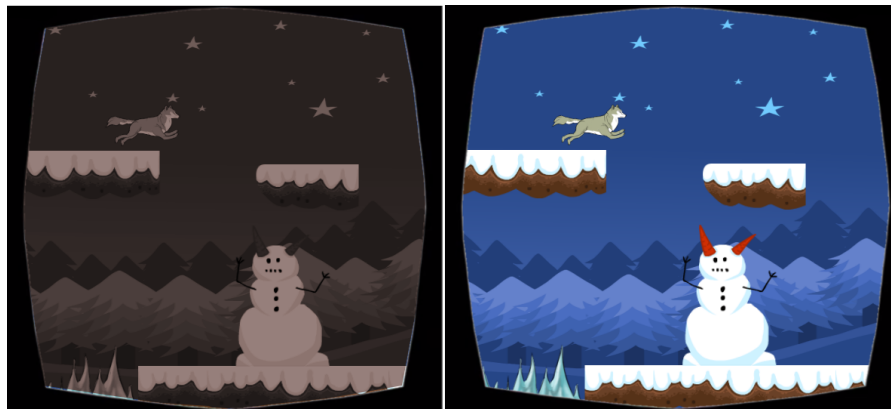


Figura 2.3: Differenziazione colori. Fonte: [7]

Come mostrato nella figura:2.3, le immagini mostrate differiscono per il colore, vi sono casi in cui si può andare ad omettere un elemento della scena, figura:2.4



Figura 2.4: Differenziazione elementi su schermo. Fonte: [7]

Il cardboard è uno strumento di realtà virtuale a basso costo che sfrutta il proprio smartphone come schermo e processore. Questo significa che, grazie al cardboard, è possibile sfruttare tutte le potenzialità del proprio smartphone per creare esperienze di realtà virtuale. Inoltre, il fatto che il telefono risieda all'interno del cardboard permette di dividere le immagini emesse dallo schermo del telefono in modo che siano percepite come immagini separate dai due occhi, creando un effetto di tridimensionalità e immersività per l'utente. Grazie a questa tecnologia, è possibile accedere a un'ampia gamma di applicazioni di realtà virtuale disponibili sullo store del proprio smartphone, rendendo l'esperienza di realtà virtuale accessibile a un pubblico sempre più vasto.

Occhiali Anaglifici

Gli occhiali anaglifici, come si può vedere nella figura 2.5, sono provvisti da due lenti in gelatina rossa e ciano, dette anche filtri, che vengono utilizzati per permettere all'occhio di percepire un'immagine composta da due immagini sovrapposte, ognuna delle quali filtrata da un colore diverso. In particolare, una delle due immagini è filtrata in rosso e l'altra in ciano, in modo che quando l'immagine viene vista attraverso le lenti degli occhiali, ciascun occhio vede solo l'immagine che gli corrisponde e il cervello le unisce per creare l'effetto tridimensionale. Questa tecnica, chiamata anaglifo, è stata utilizzata per la visione di immagini tridimensionali fin dagli anni '50, e ancora oggi viene utilizzata in alcune applicazioni di realtà virtuale e giochi in 3D.

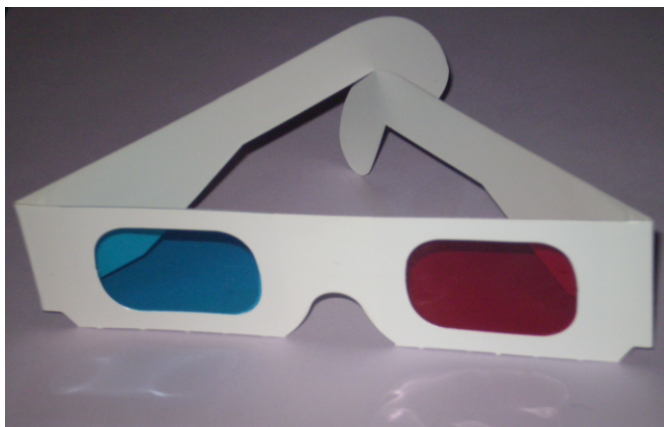


Figura 2.5: occhiali anaglifi.Fonte: [9]

Le immagini sono distanti tra loro (generalmente 5,7 cm), con colori complementari, figura:2.6, essi vengono usati per poter osservare solo l'immagine cromatica complementare, questo permette di ricreare l'effetto tridimensionale.



Figura 2.6: Anaglifo.Fonte: [9]

Active shutter 3D system

L'active shutter 3D system, invece, utilizza degli occhiali attivi, dotati di lenti liquide o cristalli liquidi che si alternano rapidamente tra uno stato di trasparenza e uno di opacità per bloccare la visione di uno degli occhi mentre l'altro vede l'immagine corrispondente, a una frequenza che può arrivare anche a 120Hz. In questo modo, le immagini vengono mostrate in rapida successione alternando i due occhi, creando la sensazione di visione tridimensionale. Questa tecnologia è diversa rispetto a quella del cardboard, in cui le immagini sono mostrate contemporaneamente davanti a entrambi gli occhi.



Figura 2.7: Differenziazione elementi su schermo.Fonte: [10]

2.2 Obiettivi

Utilizzando la tecnologia 3D è possibile trattare l'ambliopia, attraverso un approccio ludico, permettendo quindi di intrattenere maggiormente il paziente, spesso in età giovanile, aumentando quindi la probabilità di successo. I punti in cui 3D4Amb si focalizza sono:

- Basso costoso, il sistema si basa su tecnologie a basso costo.
- Facile da usare, non richiede nessun particolare capacità, permettendo anche ai pazienti più piccoli di poter usufruire del sistema senza l'assistenza dei genitori.
- Uso domestico, il sistema può essere utilizzato nel ambiente domestico, evitando quindi le visite frequenti all'ospedale
- Facilmente estendibile, 3D4Amb permette di usare le librerie software, agevolando quindi l'estensione del sistema, aggiungendo nuove applicazione o funzioni.

Capitolo 3

Progetto

L'obiettivo di questa tesi è quello di sviluppare uno strumento di terapia per l'ambliopia che sia in grado di intrattenere maggiormente il paziente. Per raggiungere questo obiettivo, ho implementato un videogioco utilizzando il cardboard per sdoppiare le immagini. Il genere scelto è il FPS (First Person Shooter), in modo da costringere il paziente a osservare e girare l'ambiente circostante in modo dettagliato, cosa che sarebbe stata più difficile da realizzare mediante altri generi. L'utilizzo del cardboard ha reso il gioco più immersivo e coinvolgente. Inoltre, ho scelto di utilizzare dei toni fanciulleschi per alleggerire il più possibile la terapia. L'intento ludico del gioco risiede nel fattore di progressione tra i vari livelli. Per l'interazione con l'ambiente, ho deciso di utilizzare un gamepad, come mostrato in figura 3.1, il quale mi permette di avere un buon numero di tasti e di conseguenza maggiori possibilità di interazione con il gioco.



Figura 3.1: Layout gamepad utilizzato. Fonte: [11]

3.1 Tecnologie utilizzate

Nell'era moderna, le tecnologie disponibili sono diverse, per la realizzazione del progetto mi sono avvalso delle seguenti tecnologie:

- :Cardboard,
- :Unity 3D, motore di gioco
- :Visual Code, editor di codice sorgente
- :Blender, creazione di elementi 3D

Cardboard, motivi della scelta

La motivazione che mi ha portato a scegliere il cardboard e non gli occhiali anaglifici, risiedono principalmente nella sua versatilità, visto che mi permette di poter realizzare un'applicazione con limitati limiti tecnici e con un budget ridotto, fatto cruciale, visto che si parla principalmente di un'applicazione idealmente indirizzata per pazienti giovanissimi, i quali potranno usufruire dei suoi benefici comodamente da casa. Una delle tecnologie che mi è stata particolarmente utile grazie al cardboard e allo smartphone, è il giroscopio. Questo componente ha permesso di creare un'esperienza di immersione più avanzata, in quanto ha reso possibile la percezione del movimento e dell'orientamento spaziale del dispositivo, trasferendo queste informazioni nel gioco. Grazie al giroscopio^{3.2}, il giocatore può muoversi all'interno dell'ambiente virtuale semplicemente inclinando la testa, senza dover utilizzare alcun altro dispositivo di controllo. Questo ha reso l'esperienza di gioco più fluida e naturale, aumentando l'immersione del paziente nell'ambiente virtuale e rendendo il gioco ancora più coinvolgente.

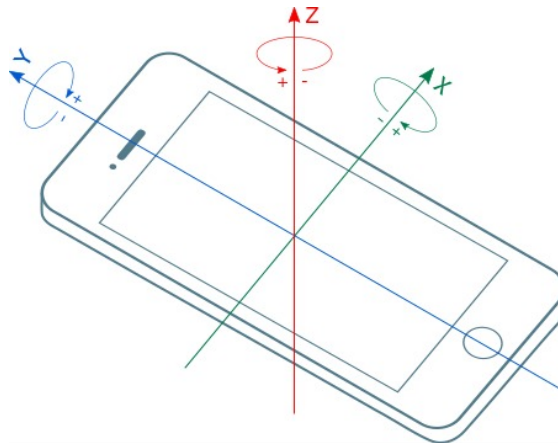


Figura 3.2: Assi Giroscopio. Fonte: [12]

Unity 3D

Unity 3D è un motore di gioco sviluppato dalla Unity Technologies, il quale permette di realizzare dei videogiochi tridimensionali e bidimensionali, la sua community e le varie guide messe a disposizione da unity permettono di cimentarsi in questo progetto superando il plateau d'apprendimento. Si tratta di uno strumento che permette di alleggerire il processo di produzione attraverso l'utilizzo di un interfaccia grafica ben organizzata, vi sono 4 schermate principali:

- Scene, Permette di spostare e/o modificare gli oggetti nella scena.
- Simulator, Permette di eseguire l'applicativo.
- Inspector, Il quale mette a disposizione delle impostazione di modifica.
- Hierarchy, Mostra i vari elementi nella scena.

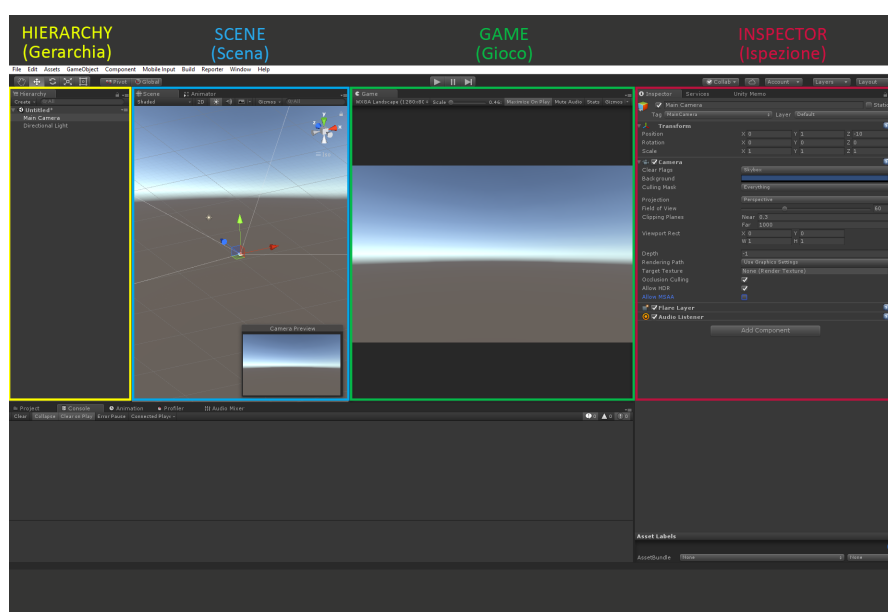


Figura 3.3: Schermate unity

Questo motore di gioco mi ha permesso di poter realizzare il gioco indipendentemente dalla piattaforma per poi andare a costruire la versione apposita, senza dovere effettuare nessuna modifica. Una funzione molto importante per questo progetto è stata la simulazione su dispositivo android, il quale mi ha permesso di poter provare l'applicativo direttamente sul mio telefono, e al contempo interagire e/o modificare la scena senza avere la necessita di dover buildare il tutto sul telefono. **L'Asset Store**, è una delle caratteristiche che ho trovato fondamentale, ovvero unity3D, mette a disposizione una serie di oggetti, anche pubblicati dall'utente, che permettono di tralasciare alcuni aspetti

in modo da potersi concentrare su altro. I materiali presenti sul Asset Store vanno dai semplici oggetti 3D, script a veri e propri pacchetti. Al interno del Asset Store è possibile quindi trovare tutti gli elementi utili per poter sviluppare un gioco da zero, volendo anche gratuitamente.

Visual code

Visual code è un editor di codice usato in correlazione con unity 3D per la realizzazione degli script di gioco, figura:3.4. Visual Studio Code è un editor di codice sorgente sviluppato da Microsoft che ha guadagnato rapidamente popolarità tra gli sviluppatori. È gratuito, multiplatforma e altamente personalizzabile grazie all'ampia gamma di estensioni disponibili attraverso il marketplace integrato. Una delle caratteristiche distintive di Visual Studio Code è la sua interfaccia utente pulita e intuitiva. L'editor utilizza una combinazione di barre degli strumenti, pannelli laterali e schede per fornire un accesso rapido alle funzionalità chiave. Un'altra caratteristica importante di Visual Studio Code è la sua capacità di supportare molteplici linguaggi di programmazione.

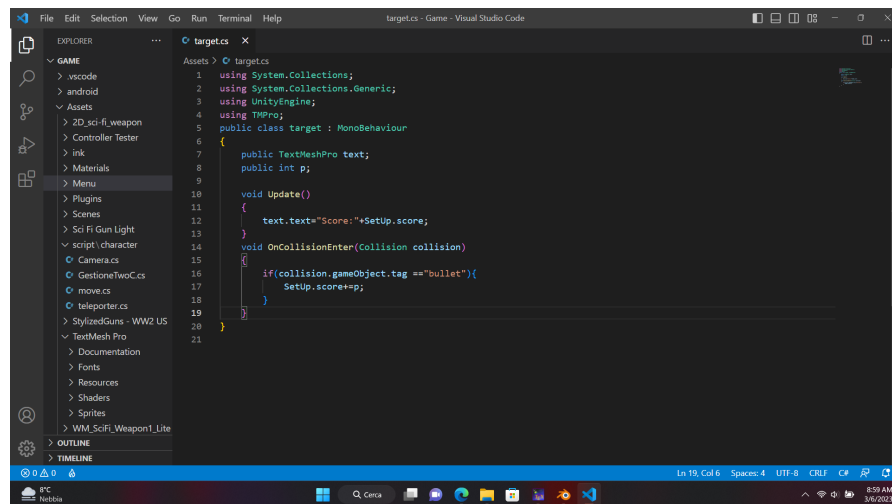


Figura 3.4: Esempio di script realizzato con visual code

Le estensioni utilizzate in questo progetto riguardano principalmente l'interazione con unity 3D:

- Unity code snippets [20], usato per introdurre l'auto completamento.
- Unity tools [21], installato per poter consultare la documentazione in modo facile e dinamico.

3.2 Struttura

La struttura dell'applicativo è molto lineare, bisogna raccogliere dei punti mediante l'interazione con determinati oggetti sparsi per la mappa, denominati *target*, una volta fatto ciò, si procede al livello successivo. Ogni livello è caratterizzata da un grado di trasparenza diverso, argomento che verrà trattato maggiormente nei paragrafi successivi. La struttura dei livelli è molto lineare, in diversi punti della mappa vi sono diversi obiettivi a cui bisogna sparare, ogni target permette al paziente di ottenere dei punti, dopo aver ottenuto un determinato punteggio si aprirà un portale che condurrà al livello successivo. Al interno dei livelli vi sono degli oggetti denominati *OcchioMalato*, i quali verranno mostrati trasparenti all'occhio sano e solidi a quello pigro, l'esempio è mostrato nella figura:3.5. Ho deciso di utilizzare il tag *OcchioMalato*, prevalentemente sui target, in modo da rendere l'ottenimento dei punti più difficile.



Figura 3.5: Esempio di trasparenza mediante un oggetto di gioco

Nella figura 3.5, possiamo notare come hai due occhi venga mostrato lo stesso oggetto ma con diversa trasparenza, in questo modo sollecitiamo l'occhio afflitto da ambliopia, lasciando anche una traccia all'occhio sano. Il livello di trasparenza varia ad ogni fase di gioco, esso aumenterà (gli oggetti diventeranno più trasparenti) man mano che si avanzerà di livello, in modo da poter stimolare efficacemente l'occhio con il passare dei livelli.

Target

Per questo progetto i target sono utilizzati per rendere l'esperienza più video ludica, dando al paziente un obiettivo da portare a termine, con difficoltà crescente, in modo da rendere il tutto più sopportabile per il lungo periodo. I target sono principalmente degli scudi medievali, figura:3.6, i quali sono sparsi all'interno dei vari livelli di gioco.



Figura 3.6: Target con punteggio

Essi si possono trovare in 2 forme:

- Scudo Classico, mostrato in figura:3.6, in questo caso la difficoltà risiede nel loro posizionamento.
- Scudo in Movimento, mostrato in figura:3.7, caratterizzato dal fatto che l'obiettivo è in movimento, costringendo quindi il paziente a seguirlo con lo sguardo, cosa che diventerà più difficile con il progredire dei livelli, sia per la velocità, che per il fattore trasparenza.

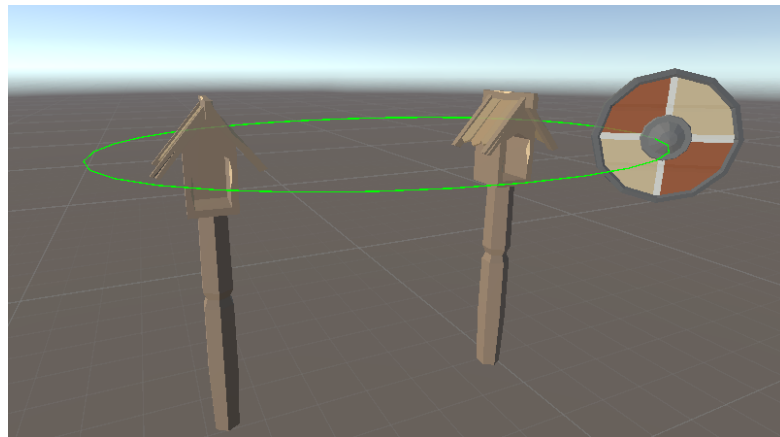


Figura 3.7: Target in movimento

3.3 Implementazione

Per l'implementazione ho deciso di utilizzare Unity 3D come motore di gioco, per l'interazione ho optato per il semplice gamepad, in modo da poter interagire efficacemente con il mondo di gioco. Il flusso di lavoro è stato diviso principalmente in due parti:

1. Fase di prototipazione, in questa fase sono andato a definire i requisiti di progetto e la loro fattibilità, dopo aver fatto ciò, sono andato a sviluppare le componenti principali, il tutto è stato fatto all'interno di un ambiente di testing.

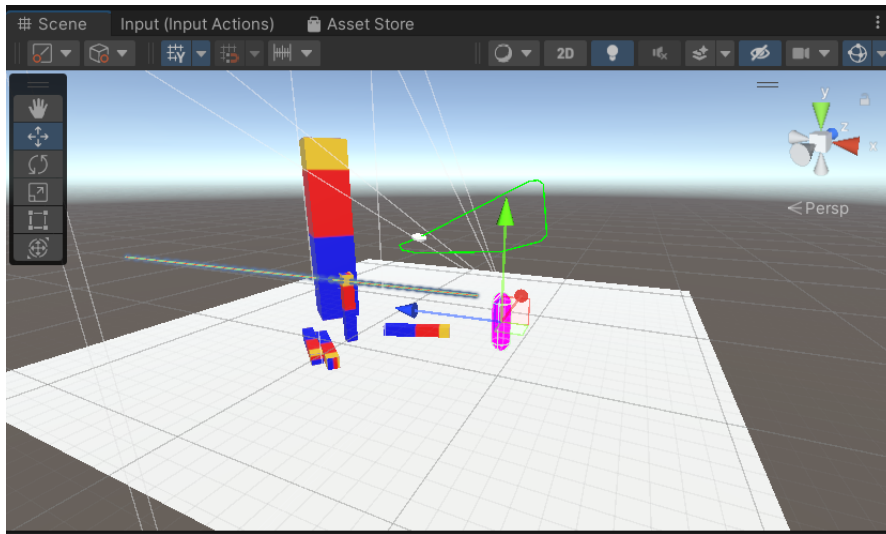


Figura 3.8: Scena prototipale

2. Fase di sviluppo, dopo aver definito i vari requisiti ed aver sviluppato le componenti fondamentali mi sono cimentato nella creazione effettiva dei vari livelli, in cui sono andato ad utilizzare le componenti sopra sviluppate.

Fase di prototipazione

Come detto in precedenza, in questa fase sono andato a definire le componenti principali del progetto:

- Visuale, in cui mi sono occupato gestire due visualizzazioni differenti.
- Mappatura dei tasti, in cui ho definito i tasti utilizzati e le azioni a esse collegate.
- Menu, in cui ho implementato i menu di scelta con le relative finestre.

Per la visuale, la preoccupazione più grande è stata ideare un sistema che permettesse di differenziare le immagini dirette nei rispettivi occhi, per fare ciò ho usato due telecamere, poste all'interno di un oggetto, in cui è presente lo script *Camera*, di cui discuterò più avanti. Le due telecamere sono state poste alla distanza interpupillare media, 62mm [22]. Come accennato prima, le telecamere mostrano oggetti diversi, ovvero, oltre alla loro distanza, utile per ricreare l'effetto tridimensionale, mi sono anche preoccupato di trovare uno stratagemma che mi permettesse di mostrare lo stesso oggetto in maniera differente, l'effetto trasparenza. L'effetto visivo desiderato è stato ricreato mediante la creazione dello script *SetUp*, che sfrutta la funzionalità di Unity3D di poter duplicare gli oggetti

identificati con il tag *OcchioMalato*. Lo script utilizza il metodo *Instantiate* per creare una copia dell'oggetto, a cui viene applicato un effetto di trasparenza modificando il materiale. Infine, la copia trasparente viene mostrata all'occhio sano, mentre l'originale non trasparente viene mostrato all'occhio pigro. Quest'ultima parte è gestita dallo script *GestioneTwoC*. Come detto in precedenza le due telecamere sono poste all'interno di un oggetto padre, il quale contiene lo script *Camera*. Questo script mi permette di utilizzare il giroscopio del telefono in modo da poter regolare l'angolazione della visuale. Per l'integrazione del giroscopio all'interno del sistema ho utilizzato la classe *Gyroscope*, il quale mi permette di ottenere diversi valori:

- *attitude*, Restituisce l'orientamento nello spazio del dispositivo.
- *enabled*, Imposta o recupera lo stato di abilitazione del giroscopio.
- *gravity*, Restituisce il vettore dell'accelerazione di gravità.
- *rotationRate*, Restituisce la velocità di rotazione misurata dal giroscopio.
- *rotationRateUnbiased*, Restituisce il tasso di rotazione elaborato.
- *updateInterval*, Imposta o recupera l'intervallo di misurazione del giroscopio in secondi.
- *userAcceleration*, Restituisce l'accelerazione che l'utente fornisce al dispositivo.

Nel mio caso sono andato ad usare *rotationRate3.9*

```
float x=Mathf.Clamp(Input.gyro.rotationRate.x,-limitRotate,limitRotate);
float y=Mathf.Clamp(Input.gyro.rotationRate.y,-limitRotate,limitRotate);
float z=Mathf.Clamp(Input.gyro.rotationRate.z,-limitRotate,limitRotate);
testaR.transform.localEulerAngles+=new Vector3(-x,0,z)*sensitivity*Time.deltaTime;
body.transform.Rotate(0, -y*sensitivity*Time.deltaTime,0);
```

Figura 3.9: Codice usato

Per evitare movimenti troppo bruschi, ho limitato la massima e minima velocità di rotazione, attraverso *limitRotate*. Nella figura3.9, riporto il comportamento della X

$$-limitRotate \leq X \leq limitRotate$$

Per la mappatura mi sono preoccupato di associare ogni tasto ad una specifica funzione, in modo da ricreare, una facile ergonomia, in modo tale da poter riconoscere i tasti principali senza l'utilizzo della vista, in quanto durante l'esperienza è richiesto l'uso del cardboard. Di seguito, figura:3.10, riporto il layout dei tasti con le funzioni associate. Ho raggruppato la presentazione in 2 macro gruppi:Azioni personaggio e Navigazione menu. Per il macro gruppo, azioni del personaggio, vado ad elencare tutte le possibilità date al paziente, per navigare e interagire con i vari livelli dati a disposizione.

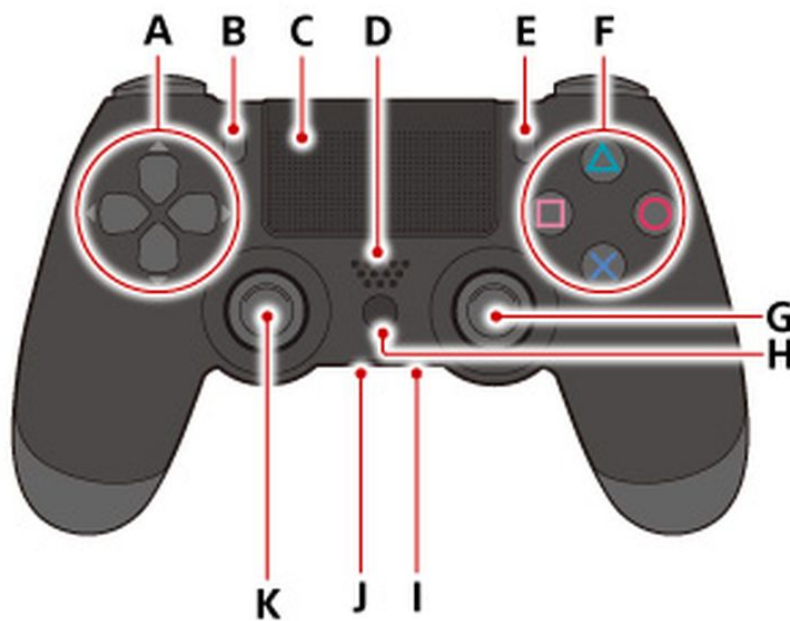


Figura 3.10: Mappatura tasti

- Movimento tridimensionale. *Stick G*
- Teletrasporto. *Stick K*
- Sparo. *Dorsale RT*
- Accucciarsi, Abbassarsi. *Tasto stick G*
- Voltarsi. *RB + Stick G*
- Sporgesi. *D Pad: A Tasti orizzontali*
- Riposizionamento telecamera. *Tasto triangolo*

Per il macro gruppo, navigazione dei menu, vado ad elencare l'insieme dei tasti attui ad utilizzare correttamente il menu.

- Apertura: *Tasto E*
- Chiusura: *B*
- Pagina precedente: *Tasto E*
- Navigazione opzioni. *D Pad Tasti verticali*

- Selezionare: *Tasto quadrato*

Come si può evincere, vi sono tasti non assegnati, questo implica la possibilità di espandere il gioco mediante l'implementazione non solo di nuovi livelli ma anche di nuove funzioni e azioni.

Nella creazione dei menu, mi sono concentrato a cercare un sistema che rispettasse il concetto di stereogramma [23]. Ho notato che i classici strumenti di Unity per la creazione dei menu non permettevano di rispettare il principio fondamentale dell'immagine stereoscopica. In questo tipo di immagini, lo stesso oggetto - in questo caso, il menu - deve essere osservato con una variazione di prospettiva per creare l'effetto tridimensionale. Per ovviare a ciò, sono andato a ricreare il menu nel mondo di gioco, in modo che venisse ripreso dalle due telecamere, con differente prospettiva, figura:3.11. In questo

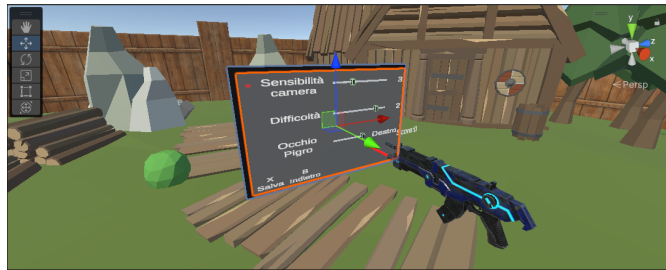


Figura 3.11: Vista del menu nella finestra *Scene*

modo l'effetto ottenuto, figura:3.12, rispetta il principio citato prima. Dopo aver progettato

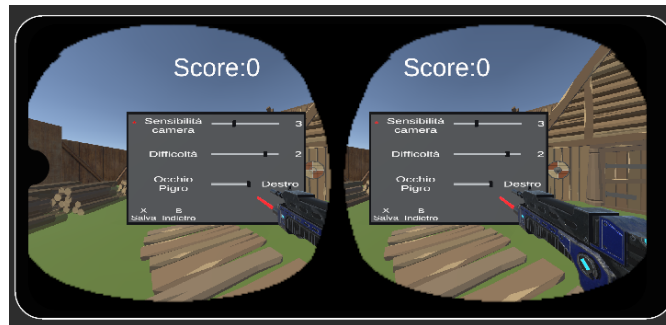


Figura 3.12: Vista del menu nella finestra *simulator*

tato e creato i componenti sopra citati, mi sono cimentato nella creazione dei vari livelli durante la **fase di sviluppo**. Il punto focale era quello di ricreare un ambiente in grado di stimolare sia l'occhio pigro che la curiosità dei pazienti. Ho cercato di rendere i singoli livelli intrattenenti, per mantenere l'interesse del giocatore.

Fase di sviluppo

Dopo aver completato la fase di progettazione e creazione dei vari componenti del gioco, mi sono concentrato sulla fase di sviluppo vera e propria. Durante questa fase, ho lavorato alla creazione dei singoli livelli del gioco, con l'obiettivo di rendere l'esperienza di gioco il più coinvolgente possibile. In questa sezione, descriverò il processo di sviluppo dei vari livelli. Per prima cosa, sono andato ad importare i vari Asset, edifici, oggetti, ecc, in modo da potermi focalizzare su altri aspetti che ho reputato più importanti. Gli asset che ho utilizzato sono:

- Rocce [24]
- Asset medievale [25], contenente edifici, alberi e oggetti medievali.
- Muri [26], contenente i vari muri che circondano ogni livello.
- Curva di Bézier [25], il quale mi ha permesso di ricreare i target in movimento.
- Controller Test [27], Asset utilizzato per la mappatura, questione approfondita più avanti.

Dopo aver selezionato i singoli asset, sono andato ad assemblare il personaggio giocante, formato principalmente dagli script:

- Camera, in cui vado a ruotare e/o muovere l'oggetto *testa*, contenente le due telecamere.
- GestioneTwo, dove mi sono concentrato sulla gestione delle due telecamere del gioco, ricreando l'effetto trasparenza e di profondità.
- Move, è uno script che ho sviluppato per gestire il movimento all'interno del gioco, questo script è stata fondamentale per garantire un'esperienza di gioco coinvolgente e immersiva.
- Shoot, riguarda l'implementazione di un'azione fondamentale del gioco, il quale permette ai giocatori di raccogliere punti sparando ai target disseminati nella mappa.
- Teleporter, è uno script che ho creato per gestire il teletrasporto all'interno del gioco. Questo elemento è stato implementato per garantire una seconda opzione di movimento, nel caso in cui il motion sickness [28] dovesse influenzare negativamente l'esperienza di gioco degli utenti.

Gli script sopra citati, vengo organizzati seguendo uno schema gerarchico, figura:3.13, essi compongono il personaggio, il componente più importante del gioco.

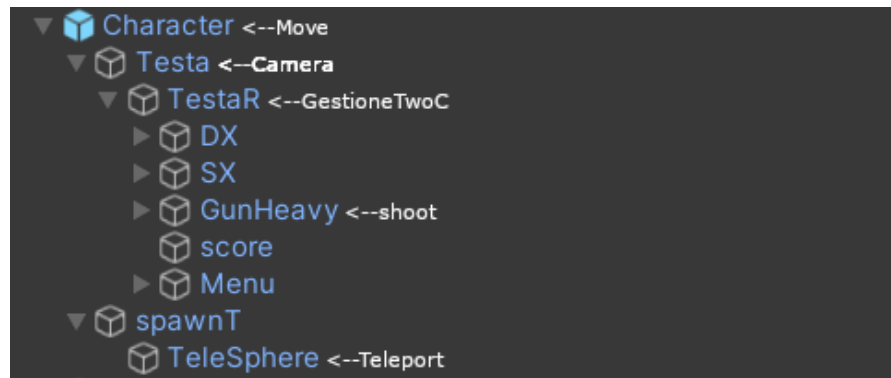


Figura 3.13: Organizzazione script

Oltre agli elementi sopra citati, sono contenuti all'interno del personaggio altri elementi, tra cui:

- DX e SX, le due telecamere usate per ricreare l'effetto stereoscopica.
- score, il quale visualizza il punteggio.
- Menu, implementa il sistema menu, descritto nel paragrafo precedente.
- spawnT, consente di impostare la posizione di origine dell'oggetto chiamato *TeleSphere*, il quale può essere spostato in modo che funga da punto di riferimento per il teletrasporto.

L'oggetto *GunHeavy* contiene lo script *shoot* come mostrato nella figura 3.13. Questo script gestisce l'azione di sparare utilizzando la classe Raycast di Unity 3D. La classe Raycast permette di emettere un raggio dalla posizione dell'oggetto in gioco e di determinare se colpisce altri oggetti nella scena. Questa funzione è stata utilizzata per permettere al giocatore di sparare ai vari target e guadagnare punti. Quando il Raycast colpisce i target, attiva una funzione all'interno degli stessi che permette di conteggiare i punti ottenuti dal giocatore, chiamata hit.

Dopo aver costruito il personaggio giocante, sono andato a creare il terreno di gioco, disseminandolo di rocce, edifici e oggetti di scena. Uno degli aspetti fondamentali è stato il posizionamento e la scelta degli oggetti "OcchioMalato", in quanto l'obiettivo principale è quello di andare a stimolare efficacemente l'occhio pigro, costringendo il paziente a prestare attenzione agli elementi del paesaggio. Legato a questo, ho disseminato ogni livello con un determinato numero di target fissi e/o in movimento, i quali sono essenziali per il proseguo del gioco e della terapia. Per quanto riguarda i target sono composti da un singolo script chiamato *target* nel caso degli statici, invece per quelli in movimento, mi sono servito di uno script aggiuntivo che ricrea le curve di Bézier [25]. **Le curve di Bézier** [29], sono utilizzate principalmente in ambito grafico e di progettazione assistita dal computer (CAD), in questo caso sono state usate per delineare il percorso che i vari

target devono seguire. Queste curve sono particolarmente utili per creare forme e traiettorie fluide e precise utilizzando un insieme di punti di controllo. Le curve sono state rappresentate in verde nella figura:3.14.

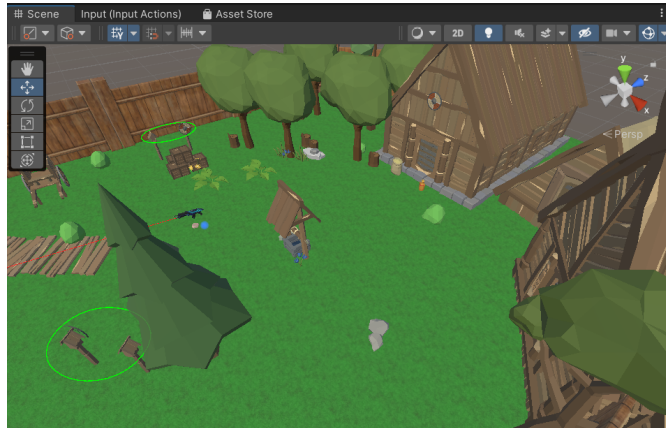


Figura 3.14: Secondo livello

Nello script del target, oltre a definire la funzione "hit" scatenata dal Raycast, è stata implementata anche la creazione dell'oggetto *fly score* attraverso il metodo *Instantiate*, che sostituisce il target colpito. All'interno di questo oggetto è stato implementato uno script che crea un effetto visivo in cui il punteggio del target colpito vola verso l'alto rimpicciolendosi gradualmente3.15. Inoltre, è stata utilizzata una funzione coseno per generare una leggera turbolenza, al fine di rendere l'effetto più dinamico.



Figura 3.15: Esempio fly score

3.4 Problemi

In questo paragrafo elencherò le limitazioni e le forzature che ho introdotto nel progetto per garantirne la stabilità. Come già accennato in precedenza3.1, l'utilizzo del Cardboard mi ha fornito un set di tecnologie molto utili, ma ha anche comportato una limitazione

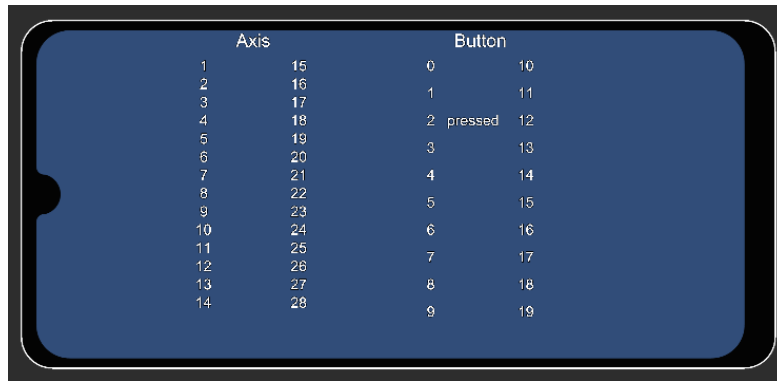
tecnica significativa. Lo smartphone utilizzato con il Cardboard non è in grado di elaborare immagini troppo complesse, il che mi ha costretto a scegliere uno stile low poly per gli asset gioco. Questa limitazione si può anche osservare dagli alberi scelti i quali non presentano foglie ma chiome stilizzate, figura:3.16.



Figura 3.16: Albero low poly contro high poly.Fonte: [13]

In relazione a ciò, parlerò della trasparenza, che è stata ampiamente discussa nei paragrafi precedenti e che comporta la duplicazione degli oggetti identificati come "OcchioMalato", modificandone il materiale. L'operazione sopra descritta implica che lo smartphone dovrà renderizzare e tenere in memoria nel peggior dei casi, il doppio degli elementi, questo implica che bisogna scegliere saggiamente a quali oggetti impostare il tag "OcchioMalato", per evitare di sovraccaricare il telefono. In fine per chiudere il discorso cardboard, andrò a parlare della limitazione che influisce maggiormente sull'immersività del gioco, ovvero la limitazione dei gradi di libertà dell'oggetto Testa, il quale è responsabile di orientare le due telecamere. Ho deciso di limitare la rotazione sull'asse z in quanto, dopo svariati test, mi sono accorto che essa causava problemi di mancata sincronizzazione tra il cardboard e il gioco.

Un altro problema, non correlato al cardboard, riguarda la mappatura dei comandi. Inizialmente, non avevo trovato la mappatura del gamepad utilizzato, il che significava che non avevo informazioni sui codici associati ad ogni tasto. Per risolvere il problema della mappatura dei comandi, ho utilizzato un asset chiamato "Controller test" [27], reperibile sull'Asset Store di Unity3D. Dopo averlo installato sul telefono, l'asset mi ha permesso di visualizzare i codici associati ai tasti del gamepad e di mapparli correttamente in Unity3D. In questo modo, ho risolto il problema della mappatura dei comandi, come mostrato nella figura 3.17.



The image shows a screenshot of a game controller configuration interface. It features a dark blue background with a white border. On the left side, there is a small icon of a game controller. The main area is divided into two columns: 'Axis' and 'Button'. The 'Axis' column lists numbers 1 through 14, and the 'Button' column lists numbers 0 through 19. The text 'pressed' is visible next to button 12.

Axis		Button	
1	15	0	10
2	16	1	11
3	17	2	12
4	18	3	13
5	19	4	14
6	20	5	15
7	21	6	16
8	22	7	17
9	23	8	18
10	24	9	19
11	25		
12	26		
13	27		
14	28		

Figura 3.17: Prova controller test, per tasto X

Grazie a questo strumento, sono stato in grado di associare i codici dei singoli tasti del gamepad alle corrispondenti azioni di gioco e movimenti. Dopo aver fatto ciò, ho configurato i tasti per eseguire le azioni e i movimenti desiderati, rendendo così possibile l'esperienza di gioco.

Capitolo 4

Conclusioni e Sviluppi futuri

In conclusione, la creazione di un'applicazione per il trattamento dell'ambliopia rappresenta un importante passo avanti nella lotta contro questa patologia visiva. Grazie alla combinazione di tecnologie alla portata di tutti, l'applicazione proposta si propone di aiutare i pazienti affetti da ambliopia a migliorare la loro acuità visiva e la loro qualità di vita. Tuttavia, come tutte le tecnologie sanitarie, anche questa applicazione necessita di essere testata ed evoluta nel tempo, al fine di garantirne la massima efficacia e sicurezza per i pazienti.

Uno sviluppo futuro potrebbe riguardare l'implementazione di un sistema multiplayer, il quale permetterebbe ai pazienti di sentirsi meno soli durante questo percorso, aumentando così l'esposizione al trattamento. Inoltre, considerando le limitazioni emerse durante lo sviluppo del progetto riguardanti l'utilizzo del cardboard, sarebbe interessante sviluppare questa applicazione per un visore che non abbia le limitazioni citate in precedenza. Ciò potrebbe comportare un ulteriore miglioramento dell'esperienza utente e dei risultati ottenuti dal trattamento. Inoltre, l'utilizzo delle VR controller gun potrebbe essere un altro interessante sviluppo, in quanto potrebbero aumentare il senso di immersione dell'utente all'interno dell'esperienza VR. Ciò potrebbe portare ad un ulteriore coinvolgimento dell'utente nel trattamento e potenzialmente migliorare i risultati a lungo termine.

In sintesi, l'applicazione per il trattamento dell'ambliopia rappresenta un'importante innovazione tecnologica nel campo della diagnostica e cura delle patologie visive, e ha il potenziale per aiutare molti pazienti a migliorare la loro acuità visiva.

Bibliografia

- [1] Universal Design Style. Pediatric vision scanner detects early childhood vision problems. <https://www.universaldesignstyle.com/pediatric-vision-scanner-detects-early-childhood-vision-problems/>. [Online; visited on 4-11-2022].
- [2] Poliambulatori San Gaetano. Ambliopia, l'importanza di una diagnosi precoce. <https://poliambulatorisangaetano.it/poliambulatorio/news/ambliopia-limportanza-diagnosi-precoce/>. [Online; visited on 6-11-2022].
- [3] Fresnel Prism. Bangerter occlusion foils. <https://www.fresnel-prism.com/product/professional-products/professional-bangerter-occlusion-foils/professional-foils/bangerter-occlusion-foils/>. [Online; visited on 9-11-2022].
- [4] ELISE REUTER. Startup testing video-based treatment for 'lazy eye' touts phase 3 results. <https://medcitynews.com/2021/09/startup-testing-video-based-treatment-for-lazy-eye-touts-phase-3-results/>. [Online; visited on 1-12-2022].
- [5] eye bit. How it works. <https://lazy-i-bit.co.uk/howitworks.html>. [Online; visited on 27-03-2023].
- [6] oculistanet. Stimolatore pattern-flicher mf17. <http://www.oculistanet.it/mf17/mf17.html>. [Online; visited on 29-03-2023].
- [7] 3d4amb. Game for the treatment. <https://3d4amb.unibg.it/>. [Online; visited on 27-12-2022].
- [8] airlapp. Google cardboard: cos'è e come funziona? <https://airlapp.com/blog/google-cardboard-cosa-e-come-funziona/>. [Online; visited on 27-03-2023].
- [9] 3d4amb. Anaglifo. <https://it.wikipedia.org/wiki/Anaglifo>. [Online; visited on 4-01-2023].
- [10] Brilens. Active shutter 3d technology. <https://www.brilens.com/news/active-shutter-3d-technology.html>. [Online; visited on 10-02-2023].
- [11] Alexia Simard. Ps4 controller. <https://dribbble.com/shots/3352609-Ps4-controller>. [Online; visited on 03-03-2023].

- [12] xiaomiplanet. Qual è la funzione di un giroscopio e come calibrarlo nel tuo smart-phone xiaomi? <https://it.xiaomiplanet.sk/come-calibrare-un-giroscopio/>. [Online; visited on 4-2-2023].
- [13] 3dstudio. What is low poly and high poly modeling? <https://3dstudio.co/low-and-high-poly-modeling/>. [Online; visited on 28-03-2023].
- [14] Shaival S. Shah, Jennifer J. Jimenez, Emily J. Rozema, Miki T. Nguyen, Melissa Preciado, and Ashish M. Mehta. Validation of the pediatric vision scanner in a normal preschool population. *Journal of AAPOS*, 25:216.e1–216.e4, 8 2021.
- [15] David K. Wallace, Elizabeth L. Lazar, Jonathan M. Holmes, Michael X. Repka, Susan A. Cotter, Angela M. Chen, Raymond T. Kraker, Roy W. Beck, Michael P. Clarke, Ingrid J. Lorenzana, David B. Petersen, Joan T. Roberts, and Donny W. Suh. A randomized trial of increasing patching for amblyopia. *Ophthalmology*, 120, 2013.
- [16] Zidong Chen, Jinrong Li, Benjamin Thompson, Daming Deng, Junpeng Yuan, Lily Chan, Robert F Hess, and Minbin Yu. The effect of bangerter filters on binocular function in observers with amblyopia. 56:139–149, 2015.
- [17] businesswire. Luminopia announces fda approval of digital therapeutic that uses tv shows to improve vision in children with lazy eye. <https://www.businesswire.com/news/home/20211020005896/en/Luminopia-Announces-FDA-Approval-of-Digital-Therapeutic-that-Uses-TV-Shows-to-Improve-Vision-in-Children-with-Lazy-Eye>. [Online; visited on 09-9-2022].
- [18] wikipedia. Stereoscopia. <https://it.wikipedia.org/wiki/Stereoscopia>. [Online; visited on 27-03-2023].
- [19] mrcardboard. Che cos'È pop! cardboard? <https://mrcardboard.eu/it/che-cos-e-google-cardboard-come-funziona-google-cardboard/>. [Online; visited on 27-03-2023].
- [20] Kleber Silva. Unity code snippets. <https://marketplace.visualstudio.com/items?itemName=kleber-swf.unity-code-snippets>. [Online; visited on 10-9-2022].
- [21] Avin Zarlez. Unity tools. <https://marketplace.visualstudio.com/items?itemName=Tobiah.unity-tools>. [Online; visited on 10-9-2022].
- [22] otticavasari. Distanza interpupillare: cos'è e come si misura. <https://www.otticavasari.it/magazine/distanza-interpupillare>. [Online; visited on 24-03-2023].
- [23] wikipedia. Stereogramma. <https://it.wikipedia.org/wiki/Stereogramma>. [Online; visited on 27-03-2023].

- [24] Broken Vector. Low poly rock pack. <https://assetstore.unity.com/packages/3d/environments/low-poly-rock-pack-57874>. [Online; visited on 21-02-2023].
- [25] Gigel. Rpg poly pack - lite. <https://assetstore.unity.com/packages/3d/environments/landscapes/rpg-poly-pack-lite-148410>. [Online; visited on 21-02-2023].
- [26] Jaroslav Grafskiy. Mobile wooden fences. <https://assetstore.unity.com/packages/3d/props/exterior/mobile-wooden-fences-54772>. [Online; visited on 02-03-2023].
- [27] Inc. Ladmer Technologies. Controller tester. <https://assetstore.unity.com/packages/tools/input-management/controller-tester-43621>. [Online; visited on 06-10-2022].
- [28] wikipedia. Virtual reality sickness. https://en.wikipedia.org/wiki/Virtual_reality_sickness. [Online; visited on 27-03-2023].
- [29] Sebastian Lague. Bézier path creator. <https://assetstore.unity.com/packages/tools/utilities/b-zier-path-creator-136082>. [Online; visited on 06-03-2023].