POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria del Cinema e dei Mezzi di Comunicazione

Tesi di Laurea Magistrale

Progettazione e sviluppo di un ambiente ludico in realtà virtuale immersiva per lo studio di processi di incorporazione di arti superiori artificiali

Relatori

Prof. Andrea BOTTINO

Prof. Fabrizio LAMBERTI

Prof. Giacinto BARRESI

Candidato

Andrea LUCARONI

04 2023

Sommario

I soggetti che, per via di un'amputazione o per deformazione congenita, utilizzano un arto protesico possono incontrare difficoltà di adattamento tali da portali successivamente all'abbandono del dispositivo. Questo accade per via del dolore, come nei casi di Phantom Limb Pain, o per via del basso livello di embodiment (ovvero, l'incorporazione di un elemento all'interno del proprio schema corporeo e mentale) e quindi di mancata integrazione con l'arto artificiale. L'obiettivo che si pone questo studio è quello di valutare la validitá di una strategia atta a migliorare la sensazione di embodiment dei soggetti verso il proprio arto superiore. La strategia è sviluppata nell'ottica di migliorare le condizioni di vita degli utilizzatori di protesi elettromeccaniche, migliorando l'incorporazione del dispositivo prostetico nello schema corporeo. La presente ricerca si pone all'interno di un ciclo di studi condotti dall'Istituto Italiano di Tecnologia (IIT) di Genova e finalizzati a valutare il contributo dell'utilizzo di metodologie di biofeedback e di tecnologie di Realtà Aumentata per migliorare la sensazione di embodiment. Sulla scorta dei risultati dei precedenti lavori, si è scelto di utilizzare la Realtà Virtuale e, con essa, l'hand tracking come interfaccia, considerandolo il metodo più disintermediato possibile per intervenire sul mondo virtuale. L'esperienza inoltre prevede una struttura ludica, con degli obiettivi da perseguire e un punteggio ottenuto come ricompensa. La ricerca è mirata a verificare se il livello di engagement provato durante un'esperienza ludica immersiva possa influenzare positivamente il grado di embodiment percepito da una persona. Per questo motivo si è realizzata un' applicazione, fruibile attraverso un visore Meta Quest 2, in grado di offrire diversi livelli di sfida e raccogliere dati di utilizzo utili per valutare il coinvolgimento del giocatore e l'ergonomia di utilizzo. Il design dell'applicazione ha come target specifico gli utilizzatori di protesi della mano e, pertanto, è stato progettato un obiettivo di gioco tale da coinvolgere specificamente l'arto superiore degli utenti. Per rendere semplice la comprensione e l'apprendimento delle meccaniche di gioco, l'applicazione prevede la presa di oggetti posti di fronte all'utente. Il grado di sfida del gioco è rappresentato dalla necessità di ricordare una sequenza mostrata e dal dover prendere gli oggetti nel corretto ordine, con un conseguente assegnamento di un punteggio. Sono state strutturate due configurazioni, con l'obiettivo di

discriminare una versione sfidante dal punto di vista cognitivo da una non sfidante. Una configurazione infatti prevede che i soggetti memorizzino una sequenza di elementi prima che essa scompaia, mentre nell'altra versione non sarà richiesto lo stesso compito, ma la sequenza resterà sempre visibile. La lunghezza della sequenza sarà la stessa in entrambe le configurazioni, differendo quindi soltanto nell'impegno cognitivo di memoria richiesto, così da verificare l'influenza della sfida sull' embodiment risultante. L'applicazione è stata sperimentata nella sede dell'IIT di Genova, valutando l'esperienza personale di volontari non affetti da impedimenti agli arti superiori. Si è deciso di ricorrere a soggetti sani per effettuare una verifica preliminare della validitá teorica dello studio e stabilire il grado di usabilità dell'applicazione. Per valutare il livello di embodiment si è ricorso a questionari qualitativi e test di Rubber Hand Illusion, somministrati subito prima e dopo ciascuna seduta di sperimentazione.

Indice

Elenco delle figure V

1 Stato dell'arte 3
- 1.1 Definire l' Embodiment . . . 3
- 1.2 Phantom Limb Pain . . . 5
- 1.3 Studi Letteraturra- biofeedback study for prosthesis embodiment . . 5
- 1.4 Riabilitazione in ambienti immersivi . . . 7
- 1.5 Lower Limb Embodiment . . . 9
- 1.6 Studi in letteratura - VR gamified set up for PLP treatment . . . 10

2 Design dell'esperienza 14
- 2.1 Hannes . . . 14
- 2.2 Premesse di design . . . 17
- 2.3 GDD . . . 18

3 Sviluppo dell'applicazione 26
- 3.1 Creazione degli Asset e Integrazione . . . 26
- 3.2 Diagrama di esecuzione . . . 34
- 3.3 Estensibilità . . . 44
- 3.4 Interazione . . . 45

4 Analisi dei Dati 47
- 4.1 Risultati Questionari . . . 49
- 4.2 Analisi dei Risultati . . . 50
- 4.3 Feedback Playtesting . . . 52
- 4.4 Raccolta dati di Gioco . . . 53

Bibliografia 57

Elenco delle figure

1.1 Set Up Sperimentale 7
1.2 VR Set Up 11
1.3 Ambienti virtuali 12

2.1 Anatomia dito Hannes 15
2.2 Tipologie di presa Hannes 16
2.3 Modello 3D Low-Poly delle protesi Hannes 17
2.4 Asset Stanza 22
2.5 Asset Modellati 23
2.6 Elementi di UI fisici 24

3.1 fonte: Documentazione Oculus Developer 27
3.2 Composizione collider 30
3.3 Mano riggata all'armatura fornita da Oculus 31
3.4 Weight Paint dito indice 32
3.5 Diagramma esecuzione flusso di gioco 34

4.1 Test della Rubber Hand Illusion 48
4.2 Medie del calcolo del drift nel test Rubber Hand Illusion 49
4.3 Questionario NASA-TLX 50
4.4 Point Cloud dati di posizione 54
4.5 Heat Map dati di posizione 54

Introduzione

Le moderne protesi di arti superiori hanno visto un sostanziale miglioramento tecnologico e di facilità di utilizzo, permettendo a persone che hanno subito la perdita di una mano o dell'intero avambraccio di guadagnare autonomia nell' utilizzo quotidiano. L'integrazione dell' utilizzatore con l'arto protesico non è tuttavia scevra di complicazioni e in taluni casi può non raggiungere un livello soddisfacente, fino a portare all'abbandono del dispositivo. Tra le principali problematiche in cui i soggetti amputati possono incorreresi segnalano: un basso livello di embodiment verso l'arto e il cosiddetto dolore dell' arto fantasma (Phantom Limb Pain o PLP), due aspetti tra di loro correlati. Con il termine embodiment si intende l'integrazione o incarnazione di un oggetto esterno nello schema corporeo e mentale di un individuo. Da recenti studi emerge infatti come da un maggiore senso di embodiment, misurato tramite test di deriva propriocettiva e questionari qualitativi, derivi una diminuzione di dolore dell'arto fantasma. Partendo da questa premessa, sono state portate avanti ricerche finalizzate ad aumentare l'integrazione degli utilizzatori con le loro protesi, dalle quali è emersa la necessità di sviluppare strumenti atti ad aiutare gli utilizzatori di protesi a migliorare le prestazioni e l'integrazione dei loro arti prostetici. L' emergere di dispositivi consumer per realtà virtuale e realtà aumentata ha permesso di creare soluzioni immersive personalizzate, disponibili da remoto per gli utenti e altamente interattive. Lo studio e gli esperimenti di questa tesi sono realizzati in collaborazione con l' IIT (Istituto Italiano di Tecnologia) di Genova, in particolare con la supervisione del professor Giacinto Barresi. Il lavoro si situa all'interno di un percorso di ricerche portate avanti dal medesimo team, con la collaborazione di altri tesisti e dottorandi. Ci si riferisce in particolare alla realizzazione di due setting immersivi per migliorare il senso di embodiment di utilizzatori di protesi Hannes, anch'essa realizzata nei laboratori dell' IIT di Genova insieme ad Inail. L'obiettivo che si pone questo studio è quello di valutare la validità di una strategia atta a migliorare la sensazione di embodiment dei soggetti utilizzatori di protesi Hannes verso il proprio arto superiore. L' idea della ricerca è di creare un' esperienza ludica, che preveda un task sfidante e che deve che deve essere completato mediante l'utilizzo diretto della mano artificiale. Sebbene questo sia il target definito in fase iniziale, i dati derivanti dall' utilizzo sono stati raccolti

da dieci utenti dotati di entrambe le mani organiche e senza altri impedimenti agli arti superiori. La presente tesi è suddivisa come segue:

1. nel primo capitolo si è analizzata la definizione di embodiment e delle problematiche affrontate dagli utilizzatori di protesi prima e dopo l'utilizzo di un arto elettromeccanico; inoltre è stato analizzato lo stato dell' arte degli studi in ambito riabilitativo e di miglioramento della condizione delle persone amputate, con particolare attenzione alle soluzioni che prevedono l'utilizzo di tecnologie immersive;

2. nel secondo capitolo è stato analizzato il design dell' applicazione partendo dall' anatomia della protesi Hannes e il conseguente studio del modello 3D e arrivando al bilanciamento della difficoltà e dei parametri dell' applicazione;

3. nel terzo capitolo è stato trattato lo sviluppo dell' applicazione, visualizzando con maggior dettaglio i singoli elementi costitutivi della stessa e la loro estendibilitá;

4. nel quarto capitolo, infine, sono stati analizzati i dati raccolti durante le sessioni di playtesting realizzate da soggetti sani, dai quali sono state tratte le conclusioni sull'efficacia degli strumenti sperimentati in questa ricerca.

Capitolo 1

Stato dell'arte

1.1 Definire l' Embodiment

Lo sviluppo di protesi ha permesso alle persone amputate di recuperare parzialmente le funzionalitá dei loro arti mancanti, e l'avanzamento della ricerca ingegneristica nel campo ha portato alla produzione di modelli che sempre di piú si avvicinano al corrispettivo organico da replicare. Questo porterebbe a pensare che l'integrazione di un dispositivo medicale esterno con il corpo sia linearmente proporzionale al quantitativo di funzionalità offerte e alla somiglianza estetica con un arto naturale. Tuttavia questa relazione risulta essere piú sfaccettata di cosí, comprendendo ulteriori fattori, non sempre quantificabili numericamente e univocamente. Per comprendere e valutare il processo di integrazione è fondamentale definire il concetto di "Sense of Embodiment" (a cui ci riferiremo semplicemente come Embodiment), che descrive la misura in cui un corpo esterno entra a far parte dello schema corporeo di un individuo. La definizione fornita risulta essere comprensiva di molte sfumature che potrebbero talvolta rendere il suo utilizzo impreciso. Per questo motivo ci avvaliamo di una piú precisa distinzione presente in letteratura [1]. Kilteni et al. definiscono l' embodiment come una sensazione derivante da tre differenti aspetti che concorrono a definirla: Sense of Location , Sense of Agency , Sense of Body Ownership. In dettaglio si riportano le definizione del sopracitato studio:

1. Sense of Location: Con sensazione di posizione/locazione si intende un determinato volume di spazio dove un individuo si sente localizzato. Normalmente questa sensazione di locazione coincide con lo spazio del corpo, nel senso che un individuo percepisce una sensazione di posizione all'interno dei confini del proprio corpo. (Lenggenhager, Mouthon, Blanke, 2009). Tuttavia la definizione si riferisce piú genericamente ad un volume di spazio, per includere anche i fenomeni di Out of the Body Experiences (OBE), cioè momenti in cui la percezione di localizzazione del nostro corpo è situata all'esterno [2].

Il senso di posizione non deve essere confuso con il senso di presenza, poiché mentre con il primo si intende il senso di immedesimazione all' interno di un corpo fisico o di un avatar, con "senso di presenza" , invece , intendiamo ad esempio l'essere situati in un ambiente fisicamente intelligibile [1].

2. Sense of Agency: traducibile come "Sensazione d'agente" si riferisce alla sensazione di avere "controllo motorio globale, includendo la sensazione soggettiva di azione, controllo, intenzione, selezione di moto e l'esperienza conscia di intenzione" (Blanke Metzinger, 2009, p. 7). La definizione si applica anche per l'embodiment di dispositivi esterni sotto il controllo dell'utilizzatore.

3. Sense of Body Ownership: Il senso di proprietà del corpo si è proposto di definirlo come la combinazione di influenze "dal basso" e "dall'alto". Più precisamente le influenze dal basso si riferiscono alle informazioni sensoriali che arrivano al nostro cervello partendo dagli organi sensoriali, mentre con influenze dall'alto si intende il processo cognitivo che potrebbe modulare il processamento degli stimoli sensoriali. Tsakiris, 2010; Tsakiris Haggard, 2005)

Sebbene queste tre sottodefinizioni dell'embodiment ci aiutino a definire meglio i confini entro i quali si può utilizzare il termine e entro i quali lavorare, non è tuttavia certo il contributo delle singole parti alla misura finale del senso di embodiment. Tuttavia lo stesso studio [1] ci fornisce delle sintetiche linee guida per migliorare le tre diverse sensazioni, con particolare attenzione allo sviluppo di applicazioni immersive in realtà virtuale. In particolare:

1. Per migliorare il senso di posizione/ locazione è fondamentale utilizzare la prospettiva egocentrica, quindi in prima persona, piuttosto che ricorrere ad una prospettiva esterna, di piú difficile immedesimazione;

2. Lo sviluppo di senso di agenzia dipende strettamente dalla sincronicità della correlazione visuomotoria. Perció é fondamentale avere una latenza molto ridotta in termini di tempo e una correlazione tra le animazioni nel mondo virtuale e i movimenti degli input dell'utente.

3. Lo sviluppo di senso di proprietà può essere migliorato agendo sui due diversi contributi che lo definiscono. Per migliorare le influenze "dal basso" si puó agire su una migliore correlazione sensoriale sincrona tra i movimenti reali dell'utente e la simulazione del mondo virtuale, ricorrendo a dispositivi di feedback aptico o visuo propriocettivo. I contributi dall'alto sono influenzati invece dalla somiglianza morfologica tra corpo o arto reale con il suo modello virtuale che lo rappresenta.

1.2 Abbandono delle protesi e Phantom Limb Pain

Le persone che durante la loro vita sperimentano l' amputazione di un arto, possono incorrere in quella che viene definita la sensazione dell'arto fantasma (Phantom Limb), ovvero un fenomeno in cui viene mantenuta la percezione dell' arto mancante. [2] Nel 65-70% dei casi le persone che sperimentano la sensazione dell' arto fantasma presentano altre sensazioni che sfociano in un dolore persistente e debilitante, che prende il nome di "dolore dell' arto fantasma" (Phantom Limb Pain abbreviato con PLP). [2] La sintomatologia non è unica per tutti gli individui, ma tende a variare di caso in caso, da crampi a bruciori a scosse, con frequenza di manifestazione altrettanto variabile nel tempo. Vi sono in letteratura diversi studi circa l' origine di questo fenomeno, che pare non derivare dalle terminazioni nervose periferiche bensì avere origine dal sistema nervoso centrale. In particolare il fattore che ha determinato questa conclusione è dato dalla mancata risposta del corpo al dolore successiva alla somministrazione di anestetici. Infatti in alcuni pazienti che registravano il PLP, esso risultava persistente nonostante l'utilizzo dei farmaci [3] . Inoltre la presenza di PLP in individui con mancanza congenita di arti fornirebbe un valido supporto alla tesi appena citata [4, 5], in quanto essi non hanno mai posseduto l' arto il cui dolore viene associato. In seguito ad ulteriori esperimenti è emersa la teoria secondo la quale l' amputazione di un arto possa avere come conseguenza la rimappatura corticale a livello somatosensoriale e motorio[2]. Con rimappatura corticale ci si riferisce alla riorganizzazione della mappa corticale in risposta ad uno stimolo[6]. Con mappa corticale invece si intende l' insieme di connessioni di ciascuna parte del corpo con una corrispondente area del cervello. Inoltre, i comandi motori mandati ad un arto amputato non ricevono i feedback visuali, uditivi, propriocettivi e tattili che il nostro cervello si aspetterebbe [7, 8], creando una discrepanza o "rumore" nella rappresentazione cerebrale che ipoteticamente potrebbe condurre ad una sintomatologia dolorosa[2].

1.3 Studi in Letteratura- biofeedback study for prosthesis embodiment

Lo studio portato avanti in questo progetto di tesi pone le fondamenta in studi precedenti che ognuno a suo modo dimostra l' utilità di varie configurazioni sperimentali e l'utilizzo di strumentazioni informatiche per indurre lo sviluppo di embodiment in persone amputate. In particolare lo studio analizzato per primo [9] è stato svolto anch'esso dall' Istituto Italiano di Tecnologia a Genova, che insieme allo studio [sezione 1.5] si collocano come predecessori della presente ricerca. L' obiettivo che si pone è infatti il medesimo: migliorare la sensazione di embodiment verso

la protesi di mano, in particolare lavorando con persone utilizzatrici del modello Hannes, prodotto da Inail e IIT. Per raggiungere lo scopo il gruppo di ricerca si è avvalso di un ambiente di Realtá Mista, quindi un ambiente virtuale posto all' interno dello spazio fisico reale, e di tecniche di biofeedback quali la frequenza del respiro (definita come RR : Respiratory Rate) dei partecipanti. L' idea nasce dalle evidenze di uno studio del 2018 di Xu et al. [10] circa l'effetto di tecniche di meditazione e di "mindfulness" sui risultati di esperimenti di Rubber Hand Illusion. Infatti da essi si conclude che le pratiche di meditazione possono portare a differenti esperienze di embodiment quando i partecipanti intraprendono allenamenti interocettivi, suggerendo la possibilitá di rendere la nostra rappresentazione del corpo piú malleabile attraverso la meditazione e altre tecniche di mindfulness. Da ciò nasce la domanda di ricerca volta a verificare se l' uso delle suddette tecniche possa trarre benefici anche sull' embodiment di un arto prostetico, considerando gli effetti verificatisi con l'illusione di un arto artificiale (nell'esperimento di rubber hand illusion). L' esperimento utilizza un ambiente particolare definito SAR (Spatial Augmented Reality), in quanto presenta elementi virtuali (dei modelli 3D animati di una mano protesica Hannes, una scatola di legno e una sfera con un effetto visuale all' interno) visualizzati all' interno dello spazio fisico tramite uno schermo in questo caso. Differenziandosi dai piú comuni esempi di realtà aumentata fruibili attraverso visori. Verrá chiesto ai partecipanti praticare delle respirazioni controllate per raggiungere una frequenza specifica, quindi esercizi di Biofeedback. Da qui nasce la sigla del paradigma dell' esperimento SARB (SAR + Respiratory Biofeeback). L' utente viene posto davanti allo schermo rappresentante una mano virtuale e una barra sulla quale rotolerá la sfera, le sue mani saranno coperte da un telo nero che ne escluderà la visione e verranno poste una a sinistra della schermo e una sotto, facendola coincidere con il modello digitale dell'arto. L' immagine seguente mostra una fotografia della disposizione sperimentale e una schermata dell' ambiente virtuale.

Sulla mano sinistra sono applicate delle strumentazioni di raccolta dei dati biometrici quali il Blood Volume Pressure—BVP, che unitamente al microfono delle cuffie controlla il Respiratory Rate, e il GSR (Galvanic Skin Response) che restituisce i dati della conduttivitá elettrica della pelle, indicatore quest'ultimo del potenziale embodiment dell‘ utente. Sotto la mano destra, invece , è posto uno smartphone, utilizzato per il feedback aptico all'arto reale che contribuisce a favorire la gamification dell'esperienza. Vi sono due configurazioni sperimentali, una che richiede un ritmo lento di respirazione e una con frequenza respiratoria normale non controllata. L´obiettivo dell' utente, nel primo caso sarà quello di mantenere un ritmo di respirazione lento e pressoché costante, calibrando i ritmi di respirazione prima di iniziare. La riuscita o meno di questa sfida renderà la mano visibile oppure trasparente. Una sfera percorrerà inoltre il percorso sinistra-destra sulla barra nel tempo di un minuto. Nel setting sperimentale, se la mano sarà visibile all' arrivo

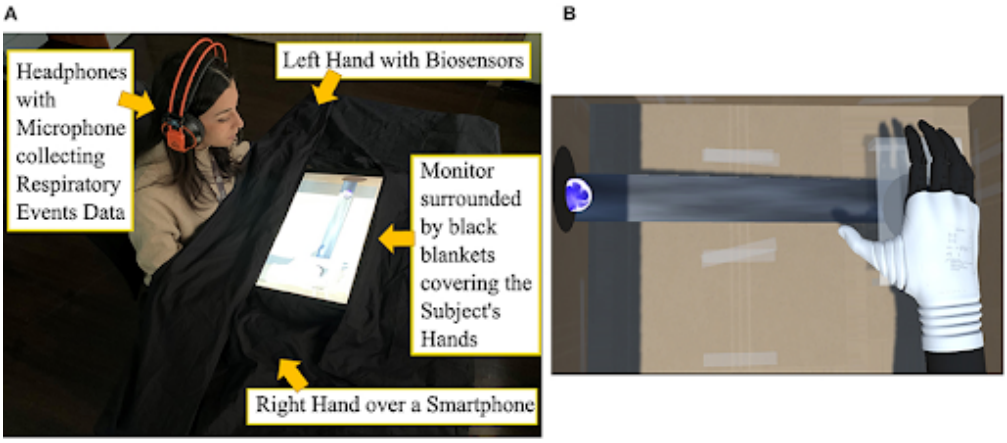


Figura 1.1: Set Up Sperimentale

della sfera, quindi nel caso in cui le condizioni di respirazione siano soddisfatte, lo smartphone vibrerá dando un riscontro immediato all'utente coinvolto. La ricerca ha prodotto due risultati:

1. Dai risultati dei questionari sottoposti ai partecipanti e dai dati dei test di Proprioceptive Drift, si conclude che il biofeedback fornito da respirazione lenta (rispetto a quella normale non controllata) puó migliorare il processo di incarnazione di un arto esterno all'interno del nostro schema corporeo. Infatti dai pareri dei partecipanti è emersa la sensazione di ownership dell' arto virtuale e la sensazione che la vibrazione fosse provocata dalla sfera rotante, sviluppando quindi anche un senso di presenza.

2. La configurazione sperimentale è risultata essere valida, e quindi una strada percorribile per successivi esperimenti di ricerca da remoto, attuabili quindi con strumentazioni accessibili dai partecipanti (schermo, microfono).

1.4 Riabilitazione in ambienti ludici immersivi di Realtá Virtuale

Come analizzato nello studio precedente, le ricerche in campo medico e di riabilitazione negli ultimi anni hanno sempre di piú utilizzato principi di Gamification e applicazioni di realtá virtuale e realtà aumentata per migliorare l'esperienza ed i risultati dei loro esperimenti. É importante quindi definire cosa si intende oggi per gamification. Essa é infatti una metodologia di sviluppo che utilizza strumenti derivanti dalla cultura ludica e videoludica per raggiungere obiettivi non prettamente di intrattenimento [11]. In tal modo si vuole fare sì che si generi motivazione intrinseca

nell'utente a compiere determinate azioni. Il vantaggio di utilizzare queste tecniche risiede nella possibilitá di aumentare il numero di partecipanti che interagiscono con un determinato contenuto, quindi applicabile anche a finalitá di marketing, e che l'esperienza vissuta sia significativa e divertente, che coincide esattamente con lo scopo di un qualsiasi gioco. Tra gli strumenti tra i piú comuni che vengono utilizzati si trova l'introduzione di un punteggio, di un obiettivo e di elementi di gratificazione quali badge e/o altri generi di ricompensa che vengono conferiti al giocatore al termine delle azioni compiute. Quest'ultima caratteristica offre una possibilitá di sfruttare questi principi in campo medico e in particolare riabilitativo, dove gli esercizi e le sessioni di terapia possono risultare lunghe, ripetitive e talvolta anche dolorose. Si è riscontrato infatti come il campo della riabilitazione fisica in ambito medico risulti essere il dominio con la piú alta frequenza di utilizzo di approcci di gamification per migliorare i risultati del trattamento [12]. All' interno di questa categoria, la riabilitazione della mano e degli arti superiori in generale sono le aree dove questo approccio è piú utilizzato. Quanto descritto in precedenza in merito all'utilizzo di metodologie di gamification applicate al campo riabilitativo e di ricerca è applicabile anche all'utilizzo di tecnologie di realtà estesa. In uno studio del 2019, che ha lo scopo di analizzare l' evoluzione delle applicazioni di realtà virtuale come campo di ricerca, si nota come la frequenza di pubblicazione di studi con temi principali connessi a questo ambito sia cresciuta costantemente negli anni. [13] Inoltre gli ambiti di applicazione si sono spostati progressivamente dal campo dell' informatica e della psicologia all' area della riabilitazione e della salute in generale [13, 14]. Non vi è nelle ricerche una nomenclatura unificata per definire la "VR" (virtual reality), si trovano infatti uno spettro di termini associati come "virtual environments" e "virtual gaming", suggerendo un interesse per le tecnologie immersive in senso ampio, comprendendo anche i dispositivi per realtá aumentata. Esistono diverse motivazioni che possono motivare il rinnovato interesse in questo campo. Da una parte vi è la commercializzazione su larga scala di dispositivi di realtà virtuale ed aumentata, come i visori di Meta (Rift, Quest), HTC (Vive) e Microsoft (Hololens) e applicazioni direttamente scaricabili sul proprio smartphone. Vi è stato un aumentato inoltre nell' offerta delle librerie di applicazioni e strumenti per facilitare la programmazione e prototipazione rapida di soluzioni VR. Infatti questi fattori permettono di avere un' ampia letteratura per quanto riguarda il linguaggio di interazione immersivo e di sviluppare soluzioni ad hoc, potendo prototipare applicazioni piú facilmente. Inoltre i dispositivi sopra citati presentano caratteristiche che ben si relazionano con gli obiettivi di ricerche mediche e riabilitative. Infatti i visori per realtà virtuale, ad esempio, forniscono la possibilitá di raccogliere informazioni circa la posizione e rotazione di vari giunti del corpo dell' utilizzatore. Per il presente studio si è deciso pertanto di avvalersi di tecniche proprie del linguaggio dei videogiochi, avendo dimostrato l'utilità riscontrata negli studi precedenti. Il nostro approccio non si può tuttavia definire

propriamente di Gamification, poiché l'applicazione realizzata sarà a tutti gli effetti un gioco, con lo scopo ultimo dell'utente è quello di divertirsi durante la fruizione. Nonostante quindi le differenze nella nomenclatura dei due casi, lo sfruttamento dei suddetti strumenti rimane invariato, e il loro utilizzo è il punto focale della presente ricerca.

1.5 Studi in letteratura - Lower Limb embodiment

Un altro lavoro che ha fornito importanti evidenze per creare il design del presente esperimento è quanto realizzato in un tesi di una studentessa del Politecnico di Torino Rebecca Lo Tauro [15] , del relatore Carlo Ferraresi e dei dottori Giacinto Barresi e Federico Tessari, per conto dell' istituto Italiano di Tecnologia. Nel suddetto progetto infatti si analizzano ulteriori strategie di feedback, rispetto allo studio in (sezione 1.3), per contribuire allo sviluppo di embodiment verso un arto inferiore virtuale. Anche in questo caso l'esperimento è pensato nell' ottica di una generalizzazione dei risultati verso embodimenti di arti protesici, infatti vengono utilizzati modelli 3D di sezioni di gambe artificiali. Le tecniche di feedback analizzate e messe a confronto nel lavoro di tesi sono le seguenti:

1. Stimolazione aptica, applicata al quadricipite degli utente, soluzione giá verificatasi utile nello studio precedente per aumentare la sensazione di presenza in ambiente virtuale

2. Feedback visivo, in quanto ai partecipanti verrá chiesto di muovere la gamba per toccare un muro che in alcune configurazioni si romperá al contatto

3. Feedback sonoro, anch'esso innescato dal contatto della gamba virtuale con il modello del muro

4. Feedback inibitorio, generato da elementi di "disturbo" durante alcune configurazioni, che l' utente deve evitare arrestando il movimento.

L'esperimento ha una disposizione spaziale degli elementi simile al caso dell' embodiment di arto superiore, infatti il partecipante restera seduto durante l' esperienza con uno schermo davanti a se, e un telo nero a coprire gli arti, rimanendo quindi in ambito SAR (Spatial Augmented Reality). Nello schermo sono rappresentati due modelli tridimensionali degli arti, uno organico, con possibilità discrezionale di rappresentazione maschile o femminile, e uno artificiale , riproducendo un prodotto in particolare, ovvero l' arto protesico "Hybrid Knee". Anche in questo caso si utilizza un attuatore vibrante, comandato da una scheda Arduino, per fornire un feedback aptico, ma in questo caso è presente anche un braccialetto REMO, uno

strumento per rilevare segnali EMG durante la contrazione del quadricipite. I dati EMG sono appunto segnali elettromiografici che attraversano le fibre muscolari sia durante contrazioni volontarie che involontarie e la loro modulazione da parte del cervello determina il movimento muscolare. Il sensore che li rivela, in questo caso avrá un valore di soglia di riferimento, oltre il quale la gamba virtuale sullo schermo effettuerá un' estensione. Nel lavoro sono state analizzate 4 configurazioni diverse, in modo da verificare i contributi dei singoli elementi discriminatori nelle varie casistiche. In particolare esse sono:

1. Test con muro non distruttibile
2. Test con muro distruttibile al contatto
3. Test con bolla da evitare e muro non distruttibile
4. Test con bolla e muro distruttibile

L' inserimento della bolla rappresenta un elemento di gamification, in quanto una sfida nei confronti dell' utente , che sarà posto nella condizione di difficoltà mentre effettua il compito di calciare il muro. La difficoltà richiederá una scelta che si esplica nella corretta modulazione degli stimoli muscolari. Nonostante il numero contenuto di partecipanti, dagli esperimenti effettuati sono emersi alcuni aspetti generali che possono essere usati come punto di partenza per ricerche future. Infatti sono stati sottoposti questionari relativi a ownership e agency e effettuati test di deriva propriocettiva come nello studio analizzato in precedenza (sezione 1.3). Dai questionari si evince come le configurazioni numero 2 e 4, dove vengono introdotti elementi di feedback all' utente, come i due feedback relativi alla rottura del muro e quelli relativi alla bolla da evitare. In particolare quando essi sono stati posti insieme si sono raggiunti risultati sperimentali piú soddisfacenti in termini di embodiment percepito. Questo perché, citando lo studio: "la condizione di feedback visivo di rottura del muro. È evidente che [questo] stimoli meglio l'attività e il coinvolgimento della persona" [15]. Nei risultati del test di deriva propriocettiva è emersa la stessa tendenza positiva dei questionari nelle due configurazioni con piú feedback , suggerendo l' efficacia della tesi di ricerca. Infine la risposta tattile fornita dagli attuatori posti sotto la gamba del partecipante, hanno rappresentato un elemento positivo per l' immedesimazione da parte dell'utente.

1.6 Studi in letteratura - VR gamified set up for PLP treatment

In letteratura , come già stabilito precedentemente, ci sono numerosi esempi di applicazioni immersive utilizzate per la riabilitazione. Si riporta in questa sezione

uno studio in particolare, dove aspetti di Gamification e gli ambienti interattivi ed immersivi sono utilizzati per il trattamento sanitario di Phantom Limb Pain [16], che si ipotizza essere correlato al concetto di embodiment. Per raggiungere l' obiettivo di attenuazione del dolore si è ipotizzato che il senso di presenza e l'offerta di giochi proposti potesse funzionare come incentivo per l' utente alla partecipazione e quindi all' attivazione muscolare. L' intento di Ambron E, et alt. è stato quello di realizzare una soluzione a basso costo, che potesse essere replicata in misura relativamente facile a casa degli utenti interessati. Infatti nella configurazione sperimentali vengono impiegati un visore di realtá virtuale consumer (Oculus Rift DK2) , che consente di avere i dati di posizione e rotazione della testa e presenta un ambiente virtuale senza distrazioni ai partecipanti, e delle unitá di misurazione inerziale (Inertial Measurements Units o IMU) poste sopra le cosce degli utenti e sotto il le loro ginocchia. Queste unità forniscono 9 gradi di libertà di movimento, quindi numerose informazioni di movimento degli arti inferiori, che sono state utilizzate per governare due gambe prostetiche virtuali poste nell' ambiente immersivo creato ad hoc. I dati degli IMU arrivano a una scheda Arduino e poi processati nel motore di gioco attivo sul pc e collegato al visore. Di seguito è riportata un' immagine raffigurante la configurazione, con il visore posto già sopra la testa dell'utente e le unità IMU indossate con fasce blu.

Figura 1.2: VR Set Up

Per quanto riguarda l' esperienza viene offerta la possibilità di giocare a 3 giochi differenti e la possibilità di navigare sul web, tutto questo usando come interfaccia le proprie gambe, che controlleranno un raggio puntatore per selezionare e muovere elementi.

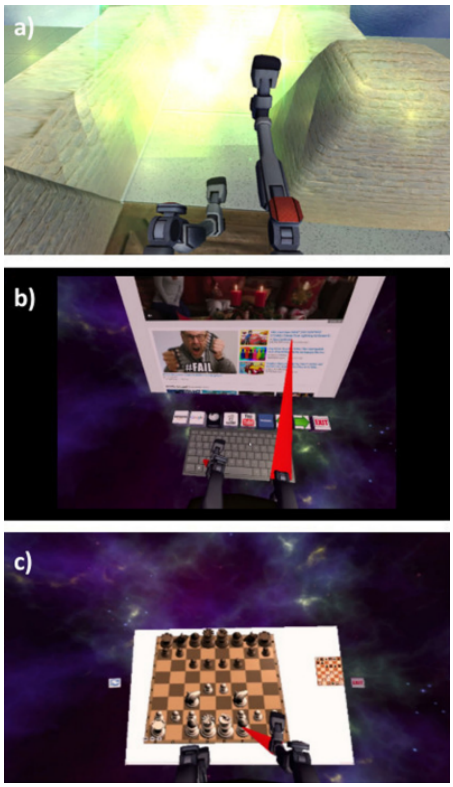


Figura 1.3: Ambienti virtuali

Ai partecipanti infatti , per giocare, era richiesto di ruotare i fianchi, piegare il ginocchio e altri movimenti coordinati, potendo però effettuare una pausa in qualsiasi momento in caso di stanchezza. Sebbene lo studio sia stato testato solamente su due pazienti amputati, i risultati sono stati molto promettenti. Dall'analisi dei questionari somministrati dopo l'esperienza - in cui si chiedevano informazioni circa l' engagement, l' ergonomia d' uso e il dolore dell'arto fantasma provato prima, durante e dopo aver giocato- il valore di PLP è risultato diminuire completamente nel primo utente durante la prima e la seconda sessione di test, mentre nel secondo utente il dolore è diminuito di una media del 97.3% in entrambe le prove. Nelle sei sessioni successive il dolore è rimasto al valore 0 su 10 per cinque volte. Indicativo è inoltre il mantenimento di buoni risultati di percezione di PLP nel tempo, con un decadimento del 22% tra la prima e la seconda sessione e del 67% per il secondo partecipante tra la prima e la quarta esperienza. Sempre quest' ultimo

ha dichiarato anche un miglioramento della agency con il proprio arto protesico, non ancora utilizzato prima di provare l' esperienza, che l'ha convinto ad iniziarne l'adozione. L' utilizzo di elementi ludici ha permesso ai partecipanti di migliorare la loro condizione di salute, giocando alle esperienze proposte per sessioni di un' ora senza pause (nonostante fossero previste liberamente), dimostrando quindi un' intrinseca motivazione di adattarsi all' ambiente virtuale e all' interfaccia di interazione. Gli autori dell' articolo suggeriscono come miglioramento un miglior parallelismo tra la rappresentazione virtuale degli arti con la controparte reale e migliore interazione per quanto riguarda la precisione degli strumenti (gli strumenti inerziali sono soggetti ad errori di deriva quando sottoposti a movimenti bruschi).

Capitolo 2

Design dell'esperienza

Il presente lavoro di tesi si pone l'obiettivo di verificare il trasferimento di embodiment ad un arto fisico partendo da quello provato verso un arto virtuale durante lo svolgimento di un task in ambiente immersivo. In particolare l'utente giocherà una partita ad un gioco a punti realizzato in VR comandando l'avatar mediante le proprie mani. Le configurazioni sperimentali vogliono verificare la differenza nella sensazione di incarnazione provata verso i propri arti superiori al variare del livello di sfida nel gioco. La prima fase di playtesting é stata effettuata su individui "sani", cioè senza impedimenti ed amputazioni degli arti superiori, in modo da avere un riscontro preliminare di usabilitá e di validitá della domanda di ricerca. Il target finale di questo studio e del ciclo di ricerche portate avanti dall'IIT di Genova sono gli utilizzatori di protesi di mano, al fine di migliorare le loro condizioni di vita e di utilizzo degli arti artificiali. Infatti , avendo l'IIT sviluppato un modello di protesi elettromeccanica (Hannes), anche le rappresentazioni virtuali vogliono replicare l'aspetto ed ergonomia di questo particolare modello, cercando di massimizzare l'immedesimazione degli utenti a disposizione.

2.1 Modello Hannes

Come stabilito dagli altri studi in letteratura, al fine di migliorare il Sense of Body Ownership dell' utente, è stato deciso di utilizzare un modello 3D che si avvicinasse il più possibile alla protesi dei fruitori dell'esperienza. L' arto a cui lo studio si riferisce è chiamato Hannes,e si tratta di un dispositivo realizzato nel corso di una progettazione di tre anni (Gennaio 2014 - Dicembre 2017) svolta congiuntamente da IIT e INAIL [17]. Si tratta di una protesi esterna controllata mioelettricamente e alimentata da una batteria, pensata per persone amputate mono o bilateralmente a livello transradiale. Il dispositivo si presenta come una mano poliarticolata che permette di ottenere il 90% delle funzionalità di una mano organica senza

impedimenti [18]. Un aspetto particolarmente importante di questo modello è il meccanismo differenziale di Dynamic Adaptive Grasp (DAG), che permette alla mano di adattarsi dinamicamente all' oggetto che si sta afferrando e ad ogni altra pressione esterna che vada a stimolare le dita. Il dispositivo si compone di diversi elementi:

- La mano Hannes
- Un modulo attivo (provvisto di un motore per imprimere una forza) per la flessione-estensione del polso
- Un modulo attivo per la pronazione-supinazione del polso
- Uno scompartimento che contiene i sensori elettromiografici superficiali
- Un contenitore della batteria e di componenti elettronici.

Le dita meccaniche riproducono gli stessi gradi di libertà delle dita reali, ad eccezione del movimento del giunto distale interfalangeo, in quanto la parte piú prossimale del dito è composta di un unico componente.

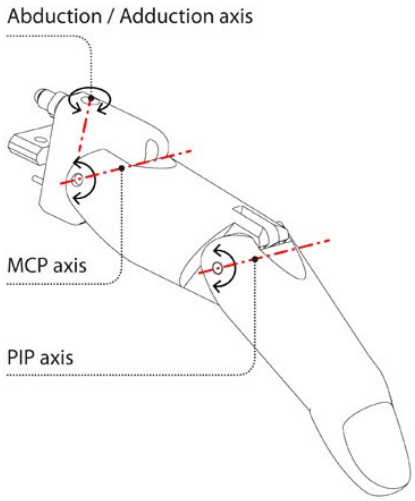


Figura 2.1: Anatomia dito Hannes

L' apertura e chiusura della mano sono comandate da due elettrodi elettromiografici (EMG) posti relativamente sui muscoli estensori e flessori dell' avambraccio. Quando l' ampiezza del segnale miografico supera una soglia di ampiezza, vengono azionati i motori per flessione/estensione delle dita sul giunto alla base, per la flessione/estensione del giunto prossimale interfalangeo e per la abduzione-adduzione

del pollice. La protesi Hannes é in grado di effettuare diverse tipologie di afferramento tramite tre posizioni discrete del pollice, realizzate attraverso un meccanismo personalizzato di stantuffi a molle. Infatti il pollice si puó allineare su tre posizioni passive ugualmente distanziate, variabili meccanicamente tra una l' altra, che permettono tre tipi di prese utili per oggetti di forma differente:

1. Afferro "Pizzicato", con il pollice piú vicino al dito indice per afferrare oggetti di piccole dimensioni, come una matita, che ha un diametro contenuto.
2. Afferro laterale, con il pollice nella posizione piú esterna
3. Afferro di potenza o "Palmare", il pollice è la posizione mediana, permettendo di trasportare oggetti di grandi dimensioni.

Figura 2.2: Tipologie di presa Hannes

Il sistema di controllo prostetico è gestito interamente da due controllori PID annidati (quello esterno controlla la corrente in entrata ai motori, mentre quello piú interno ne regola la velocità). Il controllo risultante è proporzionale ai riferimenti di velocità dei segnali mandati dal cervello verso la mano, variabili quindi a seconda dell' attivitá muscolare. Questa struttura permette quindi di impostare i parametri di controllo (guadagno e soglia per l' ampiezza dei due segnali EMG) per ogni utente. In fase iniziale i partecipanti provano i valori che garantiscono una velocità e una forza di afferramento più performanti, utilizzando un' interfaccia grafica dedicata. Per quanto riguarda l'adattamento grafico virtuale dell'applicazione si è scelto di realizzare un modello che permettesse di avere la stessa struttura del modello reale, con gli stessi giunti delle dita, le stesse proporzioni delle varie dimensionalitá e che visivamente restituisse una rappresentazione aderente alla realtá. Partendo da un modello CAD fotorealistico di Hannes è stato progettato un nuovo modello con una quantitá limitata di poligoni in modo che potesse essere processato piú facilmente e garantire una frequenza di aggiornamento dei frame dell' applicazione piú elevata possibile. Il modello di partenza offre un livello di dettaglio molto elevato, eccedendo quello richiesto per poter avere una rappresentazione realistica nei movimenti e nelle sembianze, infatti contiene dettagli anche sulle componenti interne al dispositivo. Le due versioni risultanti hanno una differenza considerevole nel numero di vertici e di triangoli:

- Il modello CAD presentava 228.761 vertici e 442.557 triangoli
- Il nuovo modello creato presenta 3.639 vertici e 7.191 triangoli

Come aspetto grafico si è mantenuto lo stile usato per la composizione dell'ambiente virtuale (come analizzato in Capitolo 2.3) , quindi lo stile Low-Poly: spigoli molto pronunciati e numero ridotto di poligoni delle mesh. La scelta è motivata dal rischio di rovinare il senso di presenza utilizzando modelli che non combaciassero tra di loro a livello estetico.

Figura 2.3: Modello 3D Low-Poly delle protesi Hannes

2.2 Premesse sul design dell'applicazione

È necessario all'inizio considerare quali siano gli obiettivi e i vincoli che hanno diretto la definizione del design dell'applicazione. L'obiettivo è il raggiungimento dell'embodiment dell'arto superiore perciò è necessario che l'esperimento coinvolga le mani dell'utente, e che l'interazione avvenga nel modo piú diretto e disintermediato possibile. Con questa condizione in mente si è scelto il medium della realtá virtuale, perché permette maggior livello di impersonificazione con l'avatar virtuale. Infatti utilizzando un visore VR è possibile comandare le braccia (ove presenti) e le mani di un avatar direttamente, che andranno a seguire i nostri movimenti. Per lo stesso motivo si è scelta la visuale egocentrica, quindi un prospettiva in prima persona,la scelta migliore per far sentire l'utente nel corpo del mondo immersivo. Si é optato inoltre per l 'utilizzo della tecnologia dell'Hand Tracking, rispetto alla scelta dei controller proprietari forniti dai costruttori di headset per la Realtá Virtuale. Il motivo di questa decisione é anche in questo caso la ricerca di immedesimazione del giocatore con il corrispettivo virtuale, che potrà in questo modo comandare ogni singolo giunto della mano 3D compiendo lo stesso movimento con la propria mano. Sebbene questa tecnologia non possa offrire un feedback tattile diretto

alla mano, a differenza di alcuni controller, offre in ogni caso una flessibilità nei movimenti molto maggiore. Per raggiungere l'embodiment si é ipotizzato che il coinvolgimento in un task sfidante, quindi in un'esperienza ludica, potesse portare alla maggior sensazione di inserimento dell'arto nel sistema corporeo. Perció è stato deciso di formulare un ambiente e delle dinamiche da gioco , nel quale viene chiesto all'utente di svolgere determinate azioni in modo corretto, con delle regolamentazioni e un conseguente punteggio per classificare la sua performance. Le suddette azioni devono essere semplici, realizzabili facilmente e non stressanti per chi partecipa al gioco, in modo da poterle reiterare nel tempo senza accusare dolori e stanchezza. Per venire incontro a queste condizioni è stato deciso di "richiedere" all'utente di afferrare degli oggetti, in movimento lento sopra un tavolo di fronte a lui. La dinamica di afferramento vuole imitare la pratica quotidiana di afferrare gli oggetti, infatti tramite l'hand tracking l'utente stringerà la mano per effettuare una presa nel gioco. Il compito richiesto é appunto quello di afferrare gli oggetti che si muovono davanti a sé e porli in un carrello posto di fianco, in un ordine determinato. Il task presentato all'utente è quindi duplice: memorizzare una sequenza di oggetti mostrati e afferrare gli oggetti in movimento ponendoli nel carrello nell'ordine indicato. Considerando il target di riferimento, caratterizzato da un impedimento fisico, e non quindi determinato da fasce di etá specifiche o da precedenti esperienze con altre tipologie di videogiochi, si è deciso di realizzare un'esperienza di piú semplice comprensione, con un punteggio solamente crescente e una finestra di tempo limitata entro cui giocare. Inoltre é stata tenuta da conto una variabilitá elevata nel grado di familiaritá degli utenti con tecnologie immersive e nella tolleranza alla nausea da movimento (Motion Sickness). La motion sickness è una sensazione di nausea e stordimento conseguente alla fruizione di un contenuto multimediale che prevede movimento incongruente a quello del mondo reale. Essa esiste anche per mezzi di comunicazione bidimensionali , prodotti quindi su schermo singolo, ma è molto piú frequente nella realtá virtuale, dove non si hanno riferimenti visivi con il mondo esterno, e la simulazione virtuale puó muoversi ad un ritmo completamente differente rispetto al corpo fisico. Per questa ragione nello sviluppo dell'applicazione si é esclusa la possibilità di muovere il corpo del nostro avatar dalla sua posizione iniziale, ad esclusione delle mani e della testa, che possono traslare e ruotare. Una volta definite le regole del gioco, sono state create 4 configurazioni sperimentali, così da ottenere diversi livelli di sfida, quindi di engagement nel gioco, e valutare eventuali differenze significative tra un caso e l'altro.

2.3 Game Design Document

Di seguito si riportano le sezioni del Game Design Document (GDD) dell' esperienza sviluppata per gli utenti. Il GDD è il documento che schematizza e unifica

le caratteristiche e le linee guida progettuali di un gioco. Di seguito si riportano le sezioni del Game Design Document (GDD) dell' esperienza sviluppata per gli utenti. Il GDD è il documento che schematizza e unifica le caratteristiche e le linee guida progettuali di un gioco.

Obiettivo:

- Totalizzare il maggior numero di punti afferrando oggetti in un tempo limite.

  Sulla parete di fronte all' utente vengono rappresentate le sequenze di oggetti da afferrare per guadagnare punti.

Meccaniche:

- Afferrare gli oggetti con le proprie mani
- Depositare oggetti nel carrello posto alla nostra destra
- Guardare la parete per memorizzare la sequenza prima che essa venga coperta

  La sequenza viene mostrata di fronte all' utente, leggermente spostata verso sinistra, contrapposta al carrello posto sulla destra.

Regole:

- Sará attiva solamente una mano virtuale, scelta a seconda della posizione della protesi dell´utente
- Il numero di oggetti in una sequenza è scelto a seconda del vincolo sperimentale, viene impostata prima di iniziare una partita, e rimarrá costante per tutta la durata
- Configurazione 1: Viene mostrata una sequenza di oggetti al termine quando l'utente é pronto schiaccia un bottone e un pannello scende dal soffitto fino a coprire la sequenza alla nostra vista
- Configurazione 2: Viene mostrata una sequenza di oggetti, e l'utente potrá riguardare la sequenza a piacimento
- Se deposito un oggetto nel carrello, ed esso appariva sullo schermo esso sará convalidato e mi sará richiesto il successivo nella serie

- Più oggetti corretti vengono inseriti in successione, senza sbagliare, più saranno i punti conferiti una volta completata la serie.
  In particolare:
  Moltiplicatore = numero di oggetti posizionati correttamente in fila Punteggio al termine di una sequenza = 3 punti + (1* Moltiplicatore)
- Ogni trial dura 30 secondi
- Vengono effettuati 10 trial
- Se metto un oggetto che al momento non è richiesto il moltiplicatore si azzera e quindi al termine della sequenza si otterranno meno punti, rispetto alla condizione senza errori.
- Gli oggetti compaiono solamente sulle tre piattaforme di fronte a sé
- Si puó afferrare solamente un oggetto alla volta

Limiti:

- Solo gli oggetti generati sopra le rampe di fronte a noi sono interagibili
- Il carrello ha grandezza limitata, ed è valida solo la consegna all'interno,nella zona colorata di blu
- Non è previsto movimento del giocatore, inteso come traslazione del corpo. Tutte le azioni possono essere realizzate intorno alla nostra posizione iniziale. Da seduti o da in piedi
- Gli oggetti vengono generati randomicamente, alcuni oggetti, se non richiesti in quel momento, non serviranno a niente

Procedure:

- Controllare periodicamente lo schermo che ci mostra la sequenza da ricordare
- Il giocatore controlla le rampe, non considerando gli oggetti sbagliati fino a che non capita quello corretto

Risorse:

- Oggetti afferrabili. Essi sono limitati, infatti una volta caduti nel buco al centro delle rampe, bisognerà aspettare che capitino nuovamente per poterli afferrare. Sono generati randomicamente, su una delle rampe, scelta anch'essa randomicamente tra le tre disponibili.

- Tempo. Avremo solo 5 minuti a disposizione, al termine dei quali non sará piú possibile ottenere punti.

Conflitto:

- Completare le sequenze che vengono richieste sulla parete di fronte sé

Sfida:

- Rimanere concentrati sugli oggetti sul tavolo e su quelli richiesti sul computer e afferrare gli oggetti in tempo, prima che cadano.

Risultato:

- Punti totalizzati durante la partita

Livelli:
Vengono proposte due configurazioni differenti, con lunghezza della sequenza da ricordare uguale, ma con un task di memoria differente:

- Sequenza di 5 oggetti in entrambe le configurazioni
- La configurazione 1 prevede che la sequenza venga mostrata, e poi quando l' utente é pronto essa verrá coperta
- La configurazione 2 non prevede la copertura, e i soggetti potranno guardare la sequenza per tutta la durata dell'esperienza

Stile grafico:
Lo stile scelto per la grafica é lo stile cosiddetto Low Poly, con il quale nome ci si riferisce allo stile di modellazione 3D per cui si utilizzano un numero di poligoni ridotti, ottenendo come risultato angoli molto accentuati tra una faccia e l'altra della mesh poligonale, che risulta essere quindi molto spigolosa. La scelta è motivata da una duplice ragione. Innanzitutto per alleggerire il carico computazionale della macchina su cui sta girando l' applicazione, infatti nel caso in cui il gioco volesse essere distribuito agli utenti per poter giocare a casa, é necessario che esso possa essere attivato su dispositivi stand-alone, con meno potenza computazionale di un computer dedicato. In secondo luogo l'intenzione era di creare un ambiente esteticamente piacevole e confortevole, ma al contempo non distraesse troppo il partecipante dalla sequenza da ricordare e dagli oggetti che compaiono e si muovono

di fronte a lui.

ASSET UTILIZZATI
Modelli 3D:

La stanza in cui si svolge la scena utilizza alcuni elementi che si trovano all' interno di un pacchetto presente sull' Unity Asset Store, sito web che comprende vari strumenti per aiutare gli sviluppatori nella progettazione di un gioco utilizzando il motore di gioco Unity. L' asset in questione si chiama : "Simple Space Interiors - Cartoon Assets" realizzato dall' azienda "Synty Studios". L' ambiente, come si evince dal nome, comprende varie componenti di laboratori tecnologici e parti di stazioni spaziali. L' idea sottostante alla scelta é stata quella di porre l' utente in un ambiente di sperimentazione come un laboratorio, e che potesse essere distaccato da un ambiente di vita quotidiana, ma piú inusuale e di gioco.

Figura 2.4: Asset Stanza

In particolare dell' asset scelto sono stati utilizzati i muri, gli armadi con le piante, la console dei comandi e le finestre di fianco all' avatar. Gli altri oggetti della scena sono stati realizzati ad hoc, cercando di ricalcare lo stile di quanto fornito dall' asset, affinché non vi sia un distacco eccessivo, il quale creerebbe una perdita di immersione.

Figura 2.5: Asset Modellati

SFX:
Per la componente audio sono stati scelti diversi Sound Effects (SFX), brevi clip che vengono azionate in particolari momenti del flusso di gioco, per migliorare il senso di immersivitá e presenza. In particolare sono stati utilizzati i seguenti effetti sonori:

- Per la selezione degli oggetti è stato inserito un effetto, che rinforza la sensazione di successo dell' utente quando egli afferra un oggetto. La clip audio é presente su Freesound.org con il nome "Button Click / Selection", realizzata da aphom000.

- Per la caduta degli oggetti nel carrello é stato utilizzato un altro effetto, sempre scaricato su Freesound.org. Esso serve a far capire che l' oggetto é caduto correttamente all' interno del cesto, anche se non stiamo osservando. L' effetto si chiama "soccer stomp 01", realizzato da Volivieri.

- Dopo ogni sequenza completata viene azionato un suono, che sta ad indicare il successo nel completamento. Il suono é extradiegetico, in quanto non é nessun oggetto dell' ambientazione virtuale a riprodurlo. L' effetto é chiamato "Approval-high-pitched-bell-synth"ed è realizzato da Gabfitzgerald, sulla stessa piattaforma dei precedenti.

UI/UX:
Gli elementi di UI scelti per l' applicazione sono integrati all' interno della scena, come elementi interni all'ambiente. Questo significa che anche gli elementi testuali sono posti su piattaforme fisiche, volendo mimare l' estetica di un tabellone reale. Vi sono infatti delle scritte poste sulla parete di fronte all' utente che indicano il punteggio attuale e il modificatore del punteggio. Alla fine dell' esperienza,

comparirà un tabellone che mostra un sommario delle statistiche della partita appena terminata. Le interazioni con la UI sono introdotte anch'esse come elementi fisici dell' ambiente. Sono presenti due bottoni da premere con le proprie mani: il primo bottone serve per iniziare la partita ed è posto da subito davanti all' utente sopra al tavolo, il secondo serve per azionare la funzionalitá di replay dei movimenti della mano, e si trova sulla piattaforma del sommario delle statistiche, quindi comparirá durante la fase finale dell' esperienza. Le statistiche che sono raccolte e mostrate all' utente sono: il punteggio raggiunto, il numero di oggetti afferrati, messi in relazione al numero di tentativi di presa effettuati, il numero di volte che si é controllata la sezione con la sequenza da ricordare, e i movimenti della la mano. Quando infatti si attiva il replay , compare un nuovo modello 3D dell'arto prostetico della scena, osservabile quindi dall' angolazione desiderata, che ripercorre le azioni dell' utente.

Figura 2.6: Elementi di UI fisici

Ulteriori elementi di interazione sono forniti dal colore degli oggetti posti nel carrello, infatti essi diventano verdi quando coincidono con quelli richiesti in quel momento, mentre si coloreranno di rosso, per poi scomparire, quelli sbagliati. Questa interfaccia visuale vuole aiutare il giocatore nel comprendere passo passo la

correttezza delle sue scelte.

Tecnologie Utilizzate:
La composizione dell'ambiente dell'esperienza e la logica del sistema interattivo sono state entrambe realizzate all' interno del motore di gioco Unity. La versione utilizzata per il prototipo finale del gioco è la versione stabile 2021.3.13f1 per PC con installato Windows 11. All'interno del software, sono stati installati dei pacchetti per poter ampliare le funzionalità di sviluppo e poter utilizzare a pieno l' hardware a disposizione, in questo caso il visore per realtà virtuale Meta Quest 2. Per ciò che concerne lo sviluppo degli elementi di gioco, si è utilizzato il pacchetto scaricato dallo Unity Asset Store chiamato "Oculus Integration" nella sua versione 44.0. Esso permette di ottenere le informazioni di movimento del visore e della posizione delle mani, sia attraverso l' utilizzo dei "Touch controller" di Meta, sia tramite riconoscimento delle mani. L'asset contiene al suo interno un sotto-pacchetto chiamato Interaction SDK che offre componenti per gestire le interazioni, infatti all'interno si trovano ad esempio delle porzioni di codice che gestiscono l' afferramento degli oggetti e la gestione delle pose delle mani. Nel capitolo sulla programmazione dell' esperienza, verranno coperti con maggior dettaglio i suddetti componenti. La rilevazione delle pose delle mani avviene all' interno del visore Meta Quest 2, grazie ad una rete neurale e le sue 4 telecamere monocromatiche poste agli angoli dell' headset. La rete neurale profonda individua la posizione dei giunti e li utilizza per ricostruire una posa a 26 gradi di libertá della mano e delle dita [19]. Questi valori sono applicati ad una mesh, che è possibile scegliere in fase di progettazione tramite l' SDK di Oculus Integration. La realizzazione dei modelli 3D della protesi Hannes, del tavolo , le rampe e gli altri oggetti di scena è avvenuta tramite il software di grafica Blender, nella sua versione 3.3.0.

Capitolo 3

Sviluppo dell'applicazione

In questo capitolo viene analizzata la struttura dell'applicazione, andando a trattare le fasi di sviluppo e principali righe di codice. In particolare il capitolo è diviso in differenti aree:

- Creazione degli asset
- Diagramma di esecuzione
- Analisi delle Interazioni
- Scena di gioco
- Classi C# e parametri
- Salvataggio dei dati

3.1 Creazione degli Asset e Integrazione

Oggetti di scena

Gli asset creati ad hoc, come anticipato nel capitolo precedente, sono stati modellati nel software di grafica tridimensionale Blender, cercando di mantenere un numero di vertici e facce più limitato possibile. Il tavolo è stato generato partendo da un cubo, aggiungendo geometria attraverso estrusioni e edge loops. Per ottenere una forma simmetrica è stato utilizzato il modificatore "Mirror" che permette di ottenere una copia speculare di quanto già modellato, quindi ottenere una seconda metá perfettamente uguale.
Di seguito si riporta il numero di vertici e triangoli dei principali oggetti di scena,

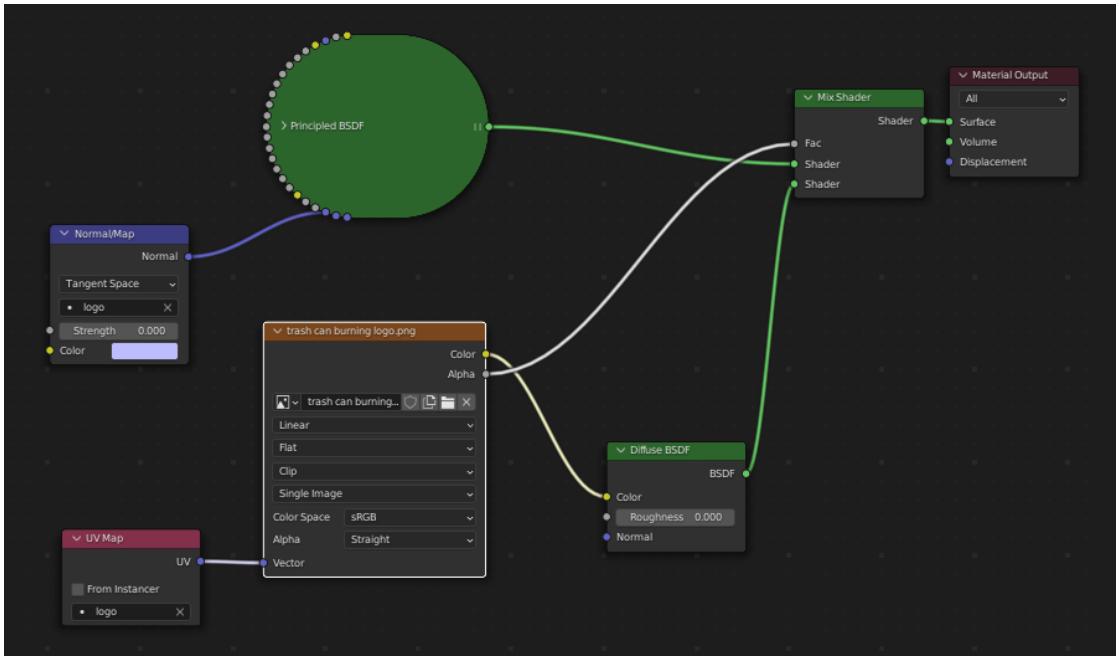
tra cui il bicchiere di carta, la mela e la scatola, i quali richiedono saranno creati anche diverse volte all'interno della scena.

Il numero di vertici, e di triangoli soprattutto si è cercato di tenerlo il piú basso possibile in modo tale da poter creare una versione per il visore Meta Quest 2 Stand Alone, che ha come valori consigliati tra i 750k e 1m triangoli totali nella scena.

Figura 3.1: fonte: Documentazione Oculus Developer

Platform	Triangle Count
Quest 1	350k-500k
Quest 2	750k-1.0m

La creazione dei modelli del bicchiere di carta e della mela è avvenuta in modo simile, partendo da cerchio, quindi un path di vertici bidimensionale, si è arrivati ad avere il modello 3D tramite diverse estrusioni. Per i materiali di tutti gli oggetti , ad esclusione di quelli contenenti le texture, si è ricorso allo shader "Principle BSDF" di Blender che è possibile trasferire nel software Unity senza perditá di impostazioni. Il trasferimento degli oggetti da Blender a Unity è avvenuto trasferendo oggetti con codifica .fbx, cosí da includere i materiali all'interno del file esportato. Per i materiali contenenti Texture si è creato uno shader differente nel Node Graph, che attraverso il nodo "Mix Shader" va a comporre i contributi di shader differenti, in questo caso: un BSDF e una texture bidimensionale con informazioni di trasparenza. Il suddetto materiale è usato per il logo nella parte posteriore del tavolo e nel modello della scatola di cereali.



Per quanto riguarda le rampe presenti nel gioco sono state modellate appositamente, creando due mesh separate, una per la base che rimane fissa nell' oggetto reale, e una per il tappeto rotante posizionato attorno. Invece di creare un' animazione del nastro, che avrebbe richiesto piú tempo per essere realizzata e piú potenza computazionale, si è deciso di creare uno shader da applicare come materiale dell' oggetto, shader che poi verrá condiviso per tutte e tre le rampe presenti nel gioco. Esso è stato realizzato da codice, utilizzando il linguaggio per programmare gli shader utilizzato da Unity, cioè Cg, derivante dal High Level Shading Language (HLSL) sviluppato da Microsoft.

```
struct Interpolators
            {
                float2 uv : TEXCOORD1;
                float3 normal : TEXCOORD0;
                float4 vertex : SV_POSITION;


            };

            sampler2D _MainTex;
            float4 _MainTex_ST;

```

```
13              Interpolators vert(MeshData v) {
14                  Interpolators o;
15                  o.vertex = UnityObjectToClipPos(v.vertex); // local
    space to clip space
16                  o.normal = UnityObjectToWorldNormal(v.normals);
17                  o.uv = v.uv0; //(v.uv0 + _Offset) * _Scale;
18                  return o;
19              }
20
21              float4 frag(Interpolators i) : SV_Target{
22
23                  float t= frac((i.uv.y - _Time.y * _SPEED) * 10) + .5
    ;
24                  float waves = t;
25                  float4 c = _ColorB;
26                  return c * (waves);
27              }
28
29                  ENDCG
```

In questo *snapshot* del codice viene indicato com'è realizzato il movimento del nastro. Nella funzione "frag" vengono definiti i valori di colore, essi ritornano come valore un vettore quadrimensionale di float, che in fase di rasterizzazione indicherá il colore da assegnare ai pixel. In questo script in particolare si prendono le componenti nella mappatura UV della Mesh del nastro e vengono assegnati dei valori tra 0 e 1 dalla funzione "frac". Essa restituisce la la parte decimale del numero che gli viene inserito come argomento, e in questo caso gli viene data la componente longitudinale della mappa UV, quindi il valore y. Esso, normalmente compreso tra 0 e 1, viene moltiplicato per 10, in modo tale da ottenere un effetto di replicazione dei valori della mappa UV. Questo accade perché la funzione frac restituirà sempre una rampa lineare di valori decimali, andando da 0.1 a 1.0, ma con l'ausilio della moltiplicazione la larghezza di questa rampa risulta accorciata e ripetuta 10 volte, con ogni iterazione identica alle altre. Una volta realizzate queste rampe di colore sulla superficie, viene sommato un valore variabile nel tempo che andrà a spostare le rampe lungo il senso di rotazione del rullo. Questi valori numerici float, sono moltiplicati per un array quadridimensionale di float che rappresenta le informazioni di colore. Il risultato finale appare come delle scale di colore che partono dal nero fino ad arrivare al colore moltiplicatore. Queste scale nel tempo si sposteranno di posizione lungo la componente longitudinale.
Al'interno di Unity si è ricorso ai Collider di tipo primitivo, cioè "Box Collider", "Sphere Collider", piuttosto che utilizzare il componente "Mesh Collider" che calcola i confini dell'oggetto a partire dalla mesh per calcolare le interazioni con la fisica del motore di gioco. La scelta è motivata dal peso computazionale a carico della CPU, che risulta essere maggiore nel caso dell'ultimo componente, dovendo essa effettuare

il controllo di collisione su tutte le facce della mesh associata. Negli oggetti con forme più complesse come nel caso del tavolo, del bicchiere e del martello si è ricorso alla composizione dei diversi componenti primitivi prima descritti.

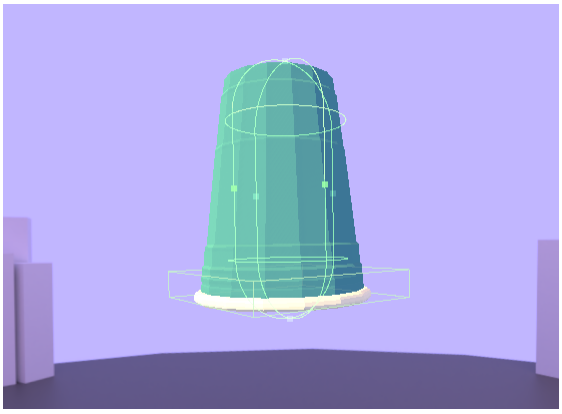


Figura 3.2: Composizione collider

Nella parte superiore del tavolo infatti vi è un incavo che servirá per far capire all'utente che gli oggetti che cadono all' interno verranno distrutti e non saranno piú afferrabili. In Unity piú Box Collider, quindi a forma di parallelepipedo, sono stati composti insieme e distribuiti in modo tale da avere dei confini definiti per il calcolo della fisica degli oggetti.

Mani

La modellazione delle mani dell'avatar, come già specificato nei capitoli precedenti, sono state modellate seguendo il riferimento della protesi reale, per quanto riguarda sia i colori che le proporzioni di essa. Sono state modellate le varie parti che compongono la protesi, ci: soffietto unito al pollice, palmo e dorso della mano, nocche, parte iniziale delle 4 dita, parte prossimale delle 4 dita; tuttavia le diverse mesh sono state tenute separate tra di loro. Questo procedimento ha facilitato la fase di rigging dell'armatura e l'assegnazione dei materiali alle relative facce. Sono presenti due versioni: una per mano sinistra e una per mano destra. All'inizio dell'esperienza sarà richiesto all'utente di scegliere con quale delle due si trova piú a suo agio. Una volta modellate le mani è stato necessario stabilire un procedimento per sostituire la mesh delle mani fornite dal SDK di Oculus con la nuova mesh creata appositamente. Per fare ciò si é optato per inserire nel progetto Blender delle mani le due mani fornite da Meta e quindi scollegare da esse la propria armatura. Una volta fatto ciò si è trattato di disporre il modello precedentemente creato nella posizione di riposo dell'armatura presa da Unity e imparentarla al modello della

mano protesica. Questo procedimento si è rivelato necessario in quanto intercorre una stretta relazione tra il sistema di Hand Tracking del visore di Meta e l'armatura inserita nel componente OVRRig della suite di Oculus Integration. In particolare il visore riconosce la posizione dei giunti della mano e successivamente l'applicazione Unity trasla le ossa dell'armatura inserita sulla posizione nello spazio 3D fornita dall'hardware. Il riconoscimento viene effettuato secondo relazioni reciproche tra i giunti, come rotazioni relative, e per questo motivo è fondamentale mantenere una gerarchia e posizionamento delle ossa dell'armatura identiche a quelle che vengono fornite nel pacchetto di Meta.

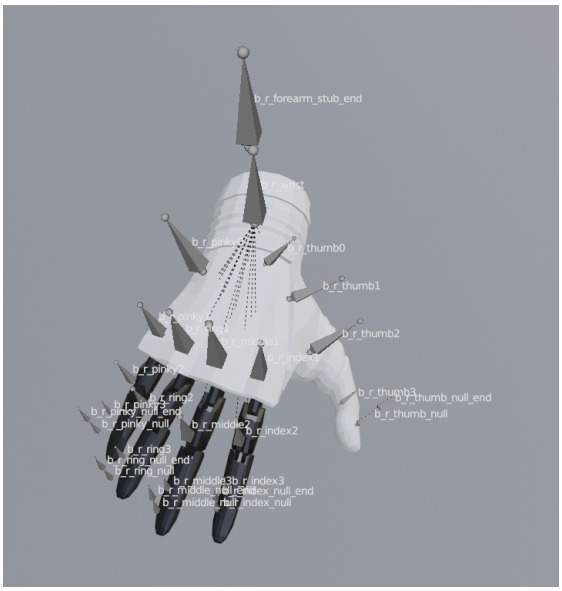


Figura 3.3: Mano riggata all'armatura fornita da Oculus

Le mesh sono state modellate separatamente in modo da avere un rigging che fornissime una simulazione del movimento reale. Infatti, essendo le parti della protesi reale meccaniche, esse risultano essere un insieme di componenti metalliche unite da corde e motori per farle muovere in sequenza. Per questo durante la fase di rigging, ad ogni osso dell'armatura corrispondente alle dita è stata assegnata una mesh specifica, con Weight Paint a 1. In questo modo infatti quando sarà mosso un osso della mano, la mesh corrispondente replicherà completamente la traslazione o rotazione solamente al componente con peso 1. Nel caso delle dita il peso è impostato pari a 1 per tutti i vertici che compongono la mesh, perciò non vi sarà deformazione, tutti i vertici del componente si muoveranno della stessa quantità.

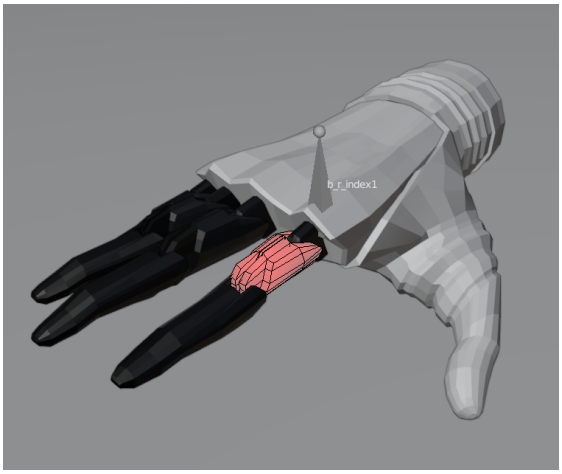


Figura 3.4: Weight Paint dito indice

Per alcuni componenti si è optato per un un Weight Paint differente, in particolare per la mesh corrispondente alle ossa metacarpali, alle nocche, e al pollice. Sia la mesh dei metacarpi sia le nocche hanno weight paint pari a 0 per tutte le ossa, non sará quindi previsto nessun tipo di deformazione o rotazione. Esse sono tuttavia imparentate con l'armatura, perciò trasleranno di posizione insieme alle altre ossa quando tutta l'armatura viene spostata o ruotata. Per il pollice invece si è optato per un rigging di deformazione, che permette quindi a varie sottosezioni della mesh di seguire differenti ossa appartenenti al pollice. La decisione è motivata dalla conformazione della protesi, infatti la mano, ad esclusione delle quattro dita dall'indice al mignolo, è coperta da una copertura in silicone per proteggere la componentistica elettromeccanica interna da polvere e altri elementi che potrebbero inficiarne l'uso. Per questo motivo dall'esterno la protesi appare come un arto organico, con angoli smussati, non dando a vedere i singoli giunti e le diverse componenti meccaniche. Il pollice virtuale deve perciò emulare al meglio questo comportamento e apparire come un componente unico snodato. Una volta modellato e riggato il modello della mano è stato inserito in Unity per essere governato secondo le informazioni dell'Hand Tracking. Il file .fbx della mano viene importato nell'engine e poi inserito nella scena, come elemento figlio del componente "Right Hand Anchor" oppure "Left Hand Anchor". è necessario aggiungere all'oggetto due componenti forniti sempre da Oculus Interaction SDK, ovvero "OVRSkeleton" e "OVRHand". Il primo si occupa di collegare le ossa dell'armatura inserita con gli script che gestiscono il movimento dei giunti, il secondo invece serve per conferire le proprietà della classe Hand al componente, per essere riconosciuto dagli altri componenti. A questo punto è stato possibile controllare la mano utilizzando le proprie mani ponendole di fronte al visore.

Audio

Le tracce sono state selezionate dalla libreria online FreeSound.org, all'occorrenza editate, come avvenuto per il rumore associato alle rampe, e poi caricate nella cartella di Unity. Infatti per ottenere un suono continuo che simulasse il lavoro degli ingranaggi per fare ruotare il nastro soprastante è stato necessario cercare due zone simili all'interno della traccia e tagliarla in modo da poter rendere la riproduzione ciclica. Per quanto riguarda gli altri suoni non è stato necessario effettuare lo stesso tipo di operazione in quanto vengono riprodotti una singola volta per ciascun evento associato.

Skybox

L'ambiente di gioco è delimitato dai muri di una stanza, un laboratorio in particolare, e per le zone esterne si è scelto di utilizzare una texture da applicare ad uno shader di tipo Skybox. Questo componente di Unity crea la superficie interna di un cubo immaginario, che verrà renderizzato dietro agli oggetti della scena. Fornendo una texture adeguata è quindi possibile ottenere una rappresentazione dell'orizzonte e dell'ambientazione personalizzata. Questa tecnica, restituisce una rappresentazione meno realistica rispetto all'utilizzo di componenti 2D e 3D disposti separatamente nell'ambiente, che forniscono invece informazioni visive di parallasse. Tuttavia, siccome lo spettatore non potrà traslare nella stanza e al di fuori di essa, si è ritenuto che l'utilizzo esclusivo di uno Skybox potesse essere sufficiente a garantire immersione nell'utente. La texture è stata realizzata in Blender, creando una sfera di grandi dimensioni e ponendo una camera al centro di essa. Lungo l'equatore di questa sfera sono stati disposti dei palazzi e al di sopra di essi delle nuvole bianche, modellati sempre seguendo lo stile Low-Poly dell'ambiente interattivo. Infine è stato posto un cubo, frontale alla camera e a ridosso della sfera, volendo rappresentare il sole, infatti gli è stato assegnato un materiale di tipo emissivo. Questo tipo di materiali in Blender sono in grado di emettere luce da tutte le facce del solido a cui viene applicato, svolgendo la funzione di colore e di illuminazione circostante. Il colore scelto per il cubo è arancione, così da illuminare anche i palazzi circostanti come accade con la luce del tramonto. Il render della texture è realizzato alla risoluzione di 2048 x 2048 pixel con forma equirettangolare, in proiezione cilindrica quindi. In Unity successivamente si è creato un componente "Directional Light" all'esterno del palazzo, per rappresentare la luce del sole che sta tramontando ed entra nelle stanze dell'edificio. La luce è stata pertanto posizionata sulla retta che congiunge lo spettatore e il sole rappresentato nella texture, così da fornire una luce realistica.

3.2 Diagrama di esecuzione

Di seguito si riporta un diagramma rappresentante la logica di funzionamento del sistema, in risposta alle interazioni dell'utente. In verde sono mostrati i componenti con uno script che regola un comportamento proprio, in bianco sono indicati gli eventi invocati da altre funzioni , in giallo i controlli che gestiscono le fasi del gioco, in blu i componenti che servono a salvare i dati e in rosso l'evento di fine della sessione.

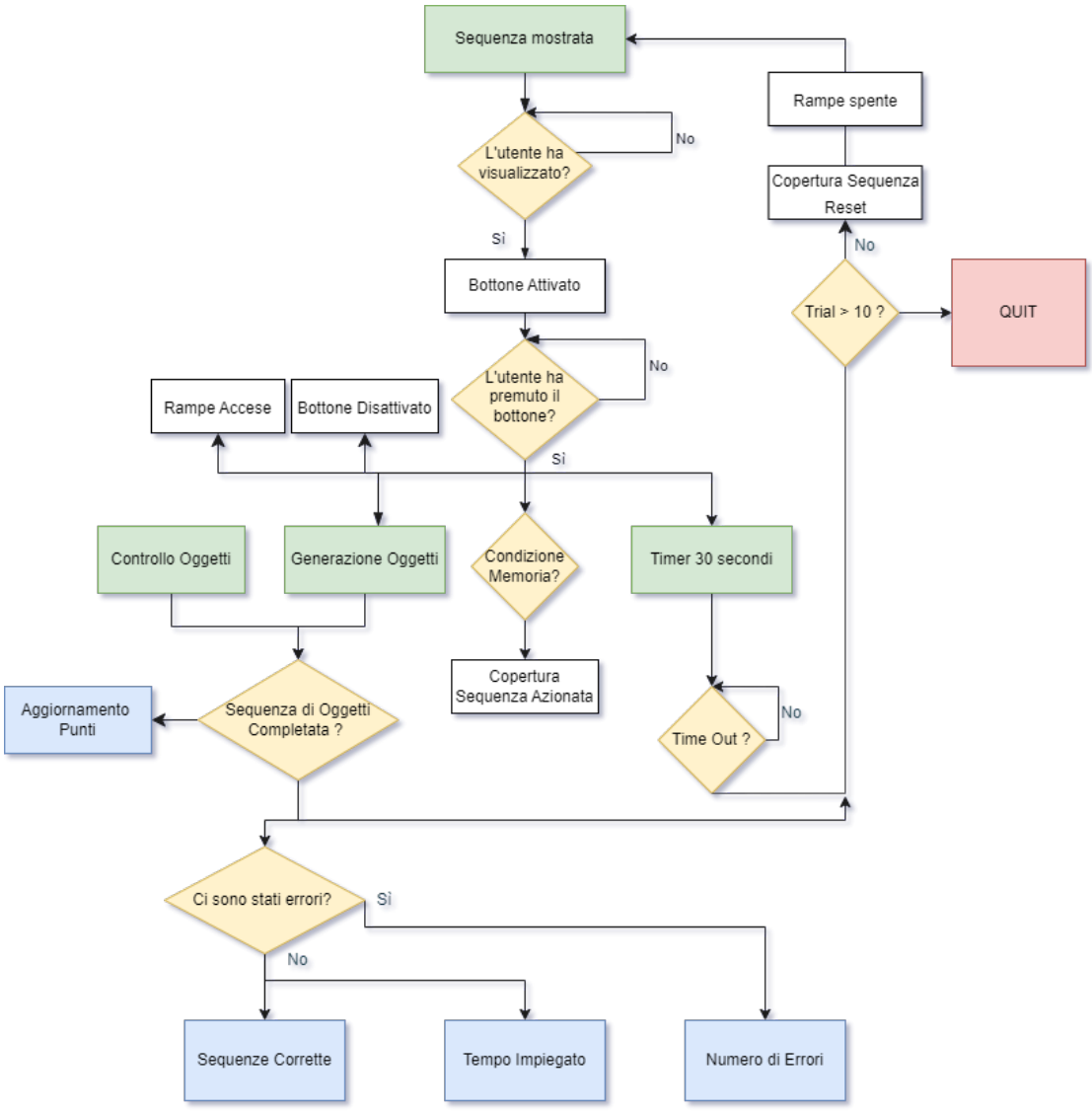


Figura 3.5: Diagramma esecuzione flusso di gioco

In particolare le classe più importanti del sistema di gioco sono: "TargetChanger.cs" che regola il susseguirsi delle sequenze e istanzia i le icone degli oggetti sulla lavagna; "ObjectGenerationVariant.cs" che gestisce la generazione degli oggetti da afferrare sulle 3 rampe di fronte all'utente, "ObjectChecker.cs" che effettua un controllo sugli oggetti depositati all'interno del carrello ; "GrabGameManager.cs" che gestisce la lunghezza della sequenza, e serve a comunicare agli altri script quando il gioco è iniziato e quando è finito. Il salvataggio dei dati è gestito da due diverse classi: "PointsManager.cs" determina quanti punti assegnare all'utente e li mostra all'utente durante il gioco, mentre "GameStatistics.cs" gestisce la scrittura dei file con i dati raccolti. Di seguito riporto alcune delle righe di codice più importanti degli script descritti.

```
1     IEnumerator TargetChangerRoutine()
2     {
3         Debug.Log("Coroutine Started");
4         // Considerare attesa iniziale se tolto il button
5         //yield return new WaitForSeconds(1);
6         manager._status = GrabGameManager.GameStatus.On;
7
8         while (manager._status == GrabGameManager.GameStatus.On)
9         {
10
11            clearBoard();
12
13            //Scelta e instanziazione dei target sulla lavagna
14            for (int i = 0; i < targetNumber; i++)
15            {
16                var t = targets[Random.Range(0, targets.Count)];
17                t.transform.position = new Vector3(0, 0, .6f);
18                objs[i] = Instantiate(t, targetPlacheholders[i]);
19                objs[i].transform.localScale = objs[i].transform.
    localScale * targetScale;
20                manager.targets[i] = t.GetComponent<ObjectUnit>().
    Type.ToString();
21            }
22
23            //Aggiornamento dei target nel controllore posto nel
    carrello
24            checker.NewTargetSet();
25
26            //Aspetto che il giocatore sia pronto per iniziare il
    prossimo livello
27            for (int i = 0; i < timeForMemorize; i++)
28            {
29                Debug.Log("Waiting for player");
```

```
30                  if (playerReadyFlag == true)
31                  {
32                      playerReadyFlag = false;
33                      break;
34                  }
35                  yield return new WaitForSeconds(.5f);
36
37              };
38
39              //Attivazione del telo che copre la sequenza
40              if (manager.planeActive)
41              {
42                  yield return new WaitForSeconds(planeScript.showTime)
    ;
43                  planeScript.StartScrolling();
44              }
45
46              //Controllo sul completamento della sequenza. Change
    Target Wait é il tempo del trial, -3 perché ci sono 3 secondi di
    countdown finale
47              for (int i = 0; i < (changeTargetWait - 3) * 2 ; i ++)
48              {
49                  if (sequenceCompletedFlag == true) break;
50                  yield return new WaitForSeconds(.5f);
51
52              };
53
54              //Segnalazione del TimeOut se non si é ancora completata
55              if (sequenceCompletedFlag == false)
56              {
57                  for (int i = 0; i < 3; i++)
58                  {
59                      var mymat = board.GetComponent<MeshRenderer>().
    material;
60                      mymat.EnableKeyword("_EMISSION");
61                      if (boardAudioSource && timeOutClip)
    boardAudioSource.PlayOneShot(timeOutClip);
62                      yield return new WaitForSeconds(.7f);
63                      mymat.DisableKeyword("_EMISSION");
64                      yield return new WaitForSeconds(.3f);
65                      if (sequenceCompletedFlag == true) break;
66                  }
67              };
68              if(sequenceCompletedFlag == false)
69              {
70                  FinishedNotCompleted.Invoke();
71              }
72              //Aggiornamento della sequenza nel controllore
73              checker.SequenceTargetReset();
```

```
74              sequenceCompletedFlag = false;
75
76              //Verifico che il livello sia finito
77              if (exitFlag)
78              {
79                  exitFlag = false;
80                  clearBoard();
81                  yield break;
82              }
83          }
84      }
```

La funzione di tipo IEnumerator chiamata "TargetChanger()" è chiamata appena la scena di gioco viene aperta, facendo quindi partire una coroutine. Al suo interno vi è un ciclo while che permette di ripetere le procedure seguenti fintanto che lo stato del gioco è attivo. Viene effettuata una "pulitura" della lavagna, ovvero vengono cancellati i target precedenti, in caso siano presenti, e successivamente se ne generano di nuovi in numero pari a "targetNumber" , definito nella classe "GrabGameManager" (il manager di gioco). I target vengono notificati al "Checker", la classe incaricata di verificare la sequenza composta dagli utenti, in questo modo si aggiornano i due componenti principali sulla successione corrente di oggetti. A questo punto partirá un conto alla rovescia che permette all'utente di ricordare la sequenza, vi è un tempo limite ma è impostato su 120 secondi, lasciando abbondante tempo nel caso in cui il soggetto lo reputi necessario. Un flag permette di capire quando il bottone è stato premuto e, quando il giocatore è pronto, il sistema esce dal ciclo e prosegue con le istruzioni successive. Se nel manager di gioco si è selezionata la seconda configurazione, dove è richiesto un task di memoria, adesso un pannello inizierà a scendere fino a coprire la lavagna per non permettere la vista della sequenza. Un nuovo contatore viene istanziato e incrementato fino allo scadere del tempo changerTargetWait, mostrando la sequenza per il tempo prefissato. Se un flag di sequenza completata viene attivato si esce dal ciclo for prematuramente e si prosegue nella coroutine. Se invece il flag rimane sul valore falso allora l'indice continuerá ad aspettare, facendo lampeggiare i contorni della lavagna quando mancano 3 secondi allo scadere. In quest'ultimo caso un evento di completamento fallito viene invocato, indicando che il trial è finito ma non è stata completata la sequenza. Il Checker viene resettato, scartando la sequenza incompleta realizzata fino a quel momento dal giocatore, e la coroutine rinizierá generando una nuova successione di oggetti.

```
1 IEnumerator Spawner()
2     {
3         int next = -1;
4
```

```
5           while (spawning)
6           {
7               List<int> pickedObjects = new List<int>();
8               List<int> pickedSpawn = new List<int>();
9               int variant = 0;
10              for (int n = 0; n < spawnPoints.Length; n++)
11              {
12                  if (next == -1)
13                  {
14                      variant = PickUnique(pickedObjects , objectList.
    Length);
15                  }
16                  else {
17                      variant = next;
18                      next = -1;
19                  }
20                  pickedObjects.Add(variant);
21                  var spawnIndex = PickUnique(pickedSpawn , spawnPoints
    .Length);
22                  pickedSpawn.Add(spawnIndex);
23                  var coordinate = spawnPoints[spawnIndex].position;
24                  //var coordinate = spawnPoints[n].position;
25                  var obj = Instantiate(objectList[variant], coordinate
    , objectList[variant].transform.rotation);
26              }
27
28              next = PickUnique(pickedObjects , objectList.Length);
29
30              yield return new WaitForSeconds(timeBetweenObjects);
31
32
33
34          }
35      }
```

Il segmento di codice riportato è responsabile della generazione degli oggetti sulle rampe. La variabile "SpawnPoints" è un array che può contenere GameObject, nell'editor si è provveduto a riempire il contenitore con 3 GameObject vuoti , quindi contenenti solo informazioni di posizione, corrispondenti ai punti di generazione soprastanti le rampe. Anche in questo caso si è ricorso ad una Coroutine in modo da poter temporizzare meglio la proposizione di oggetti. L'IEnumerator viene azionata tramite iscrizione della funzione "StartGenerating" all'evento di pressione del bottone da parte dell'utente. Al"interno" di essa vi è un ciclo while per garantire la continua creazione di oggetti, fintanto che una variabile "spawning" è posta ad un valore positivo. A quel punto viene creata "pickedObjects", una lista di interi con il compito di tenere traccia degli elementi che si stanno generando per evitare che siano presentati elementi uguali sulle 3 rampe. Durante le fasi di playtesting

la riproposizione di oggetti uguali si è rivelata un problema, in quanto l'utente doveva attendere per diversi secondi che gli capitasse l'elemento cercato. Ad ogni iterazione del ciclo while viene scelto un numero casuale compreso nel dominio [0, numero tipologia di oggetti - numero oggetti già scelti). Una volta scelto un intero casuale, esso viene confrontato con quelli già inseriti nella lista, iterando nella lista precedentemente creata. Se l'intero scelto è minore di quello inserito, esso sarà una scelta valida e verrá inserito in "pickedObjects", altrimenti l'intero sará aumentato di valore e verrá effettuato un controllo sul numero successivo nella lista. Questo algoritmo permette allo stesso tempo di non scegliere oggetti già scelti e garantisce che tutte le possibilità possono essere scelte senza dover reiterare il ciclo svariate volte. Viene inoltre salvato in una variabile il quarto oggetto mancante, cioè che non è stato scelto tra quelli possibili. Esso verrá istanziato per primo nella generazione successiva di oggetti, in modo tale da poter presentare all'utente l'oggetto che cerca almeno una volta ogni due cicli. A questo punto i 3 oggetti con l'indice corrispondente all'intero scelto vengono istanziati sopra le 3 rispettive rampe e la coroutine resterá in pausa per un tempo prefissato nella variabile "timeBetweenObjects".

```
1     public class ObjectUnit : MonoBehaviour
2 {
3   public enum Types
4     {
5         Inactive,
6         Cube,
7         Capsule,
8         Cylinder,
9         Extinguisher,
10        Hammer,
11        Apple,
12        Cup,
13        Cereal_Box,
14        Cone
15    };
16
17    public Types Type;
18
19    public int GetType()
20    {
21        return (int)Type;
22    }
23 }
```

Ogni oggetto che l'utente può afferrare al fine di completare una sequenza ha assegnato al proprio GameObject uno script "ObjectUnit.cs" che ne determina il

tipo tra quelli disponibili. Questa caratteristica sarà utile al verificarne la natura una volta selezionato.

```
1     void OnTriggerEnter(Collider other)
2     {
3         if (other.gameObject.layer != 3 || other.GetComponent<
      SpawnObjMovement>().Type == 0) return;    //object to check needs
      to have layer "ObjectInteractable"
4
5         //Check sull'oggetto rilasciato
6         var verdict = SequenceInputCheck(other);
7         ////////////////Oggetto Giusto
8         if (verdict == true)
9         {
10            collided = other.gameObject;
11            sequence.Add(collided);
12            ////////////////// Points check and update
13            if (sequenceCounter == sequenceLength)
14            {
15                collided.GetComponent<SpawnObjMovement>().Type = 0;
16                sequenceCounter = 0;
17                manager.targetGen.SequenceComplete();
18                _onScore.Invoke();
19                ptManager.IncrementPoints();
20            }
21            else { _onCorrectObject.Invoke();}
22            other.gameObject.GetComponent<SpawnObjMovement>().Type =
      0;
23            ////////////// Positive Feedback Color
24            var mymat = collided.GetComponent<Renderer>().materials;
25            foreach (var m in mymat)
26            {
27                m.SetColor("_Color", Color.green);
28            }
29           }
30        ////////Oggetto sbagliato
31        else
32        {
33            ////////////////// Points check and update
34            other.gameObject.GetComponent<SpawnObjMovement>().Type =
      0;
35            ptManager.DecrementPoints();
36            if (manager.useLife) manager.LoseLife();
37
38            ////////////// Negative Feedback Color
39            _onWrongObject.Invoke();
```

```
40              var mymat = other.gameObject.GetComponent<MeshRenderer>()
    .materials;
41              foreach (var m in mymat)
42              {
43                  m.SetColor("_Color", Color.red);
44              }
45              collided = other.gameObject;
46              objectSend = ObjectSend(other);
47              StartCoroutine(objectSend);
48          }
49      }
50      public bool SequenceInputCheck(Collider other)
51      {
52          var _isCorrect = false;
53          if (other.gameObject.GetComponent<SpawnObjMovement>().Type.
    ToString() == targets[sequenceCounter])
54          {
55              sequenceCounter++;
56              return _isCorrect = true;
57          }
58          return _isCorrect;
59      }
```

Nella porzione di codice di questa classe viene analizzata la logica di controllo sull'oggetto che entra in collisione con il collider posto sul fondo del carrello. La funzione è onTriggerEnter è chiamata quando i GameObject che hanno un componente Collider urteranno l'elemento Collider del carrello, peró esso deve essere di tipo Trigger, cioè che non parteciperà attivamente al calcolo della fisica dell'urto ma verificherà solo l'avvenimento. Quando una collisione avviene, lo script controlla che il GameObject scontrato sia collocato su un layer apposito degli oggetti e che non sia ancora stato controllato in precedenza. Se le condizioni sono verificate viene chiamata la funzione "SequenceInputCheck", la quale verificherá che il tipo dell'oggetto depositato sia uguale al tipo richiesto in quel momento. Al verificarsi anche di questa condizione si incrementerá un contatore che indica lo stato di avanzamento attuale della sequenza e di conseguenza indicherà il nuovo oggetto richiesto. Vi sono a questo punto due ulteriori condizioni. Se la lunghezza della sequenza realizzata dall'utente è pari a quella richiesta , quindi 5 in questo caso, allora si invocherà un evento "_onScore", che indica la fine del trial corrente. Se invece la sequenza è ancora minore, verrà preso il componente "MeshRenderer" dell'oggetto e cambiato il colore con il verde, ad indicare correttezza della scelta. Nel caso in cui il tipo dell'oggetto inserito non combaci con quello richiesto al momento, verrà invocato un evento "_onWrongObject", che servirá per notificare agli altri componenti di non contare il trial tra quelli corretti.

```
1     public void toSaveTrialStats()
2     {
3         var total = TrialTimeStat.Length;
4         Dictionary<int, float> statistics = new Dictionary<int, float
      >();
5         Dictionary<int, float> errors = new Dictionary<int, float>();
6         // Add some sample data to the dictionary
7         var averageTime = 0f;
8         for(int i=0; i < total; i++)
9         {
10            statistics.Add(i +1, TrialTimeStat[i]);
11            if(TrialTimeStat[i] > 0)
12            {
13                averageTime += TrialTimeStat[i];
14            }
15            errors.Add(i + 1, errorPerTrial[i]);
16        }
17        averageTime /= TrialSuccessStat;
18        var successRatio = ((float)TrialSuccessStat / (float)total) *
       100f;
19        // Convert the dictionary to a CSV string
20        StringBuilder sb = new StringBuilder();
21        sb.AppendLine(string.Format("{0}", "Statistiche per Trial:"))
      ;
22
23        foreach (KeyValuePair<int, float> kvp in statistics)
24        {
25            sb.AppendLine(string.Format("{0},{1}", kvp.Key, kvp.Value
      .ToString("0.00").Replace(",", ".")));
26        }
27        foreach (KeyValuePair<int, float> ept in errors)
28        {
29            sb.AppendLine(string.Format("{0},{1}", ept.Key, ept.Value
      .ToString()));
30        }
31        sb.AppendLine(string.Format("", "")); // Separatore
32        sb.AppendLine(string.Format("{0},{1}", "Total Successfull
      Trials:", TrialSuccessStat.ToString()));
33        sb.AppendLine(string.Format("","")); // Separatore
34        sb.AppendLine(string.Format("{0},{1}", "Success Ratio:",
      successRatio.ToString("0.00").Replace(",", ".") + "%"));
35        sb.AppendLine(string.Format("","")); // Separatore
36        sb.AppendLine(string.Format("{0},{1}", "Average Time Per
      Trial:", averageTime.ToString("0.00").Replace(",", ".") + "seconds
      "));
37        sb.AppendLine(string.Format("","")); // Separatore
38        sb.AppendLine(string.Format("{0},{1}", "Total errors:",
      TotalError.ToString()));
39        string csv = sb.ToString();
```

```
40
41          // Save the CSV data to a file
42          //string path = Application.persistentDataPath + "/statistics
    .csv";
43          string path = filepath + "/statistics.csv";
44          File.WriteAllText(path, csv);
45          Debug.Log("Trial Statistics saved on disk");
46      }
```

Nello script riportato sopra si indica come l'applicazione salva i dati di gioco della partita appena conclusa. Per integrare facilmente i dati numerici risultanti si è deciso di creare un file csv contenente una tabella riportante sulle righe il numero del trial e sulle colonne il tempo in secondi impiegato dall'utente nel concluderlo. Viene perciò creato un dizionario all'inizio della funzione "toSaveTrialStats", e all'interno vengono salvate le coppie chiave-valore: numero del trial , tempo impiegato. Il dizionario viene successivamente iterato per tutti sui valori e tramite la classe "AppendLine" della classe "StringBuilder" vengono scritte le righe e colonne sul file csv per tutte le coppie chiave-valore. In fondo alla tabella vengono aggiunti inoltre 4 valori statistici complessivi, su righe differenti dello stesso file: Il numero totale di successi, il rateo di successi , il tempo medio per ciascun trial corretto e infine il numero totale di errori. Per quanto riguarda il percorso del file in cui vengono salvati i dati si è scelta una stringa di caratteri composta da una parte variabile e una invariabile. Infatti i file sono salvati sempre all'interno del disco fisso del computer, ma per ogni sessione viene creata una cartella chiamata come nome e cognome del giocatore, dentro di essa una cartella con l'orario del trial e ancora all'interno una cartella indicante la condizione sperimentale dell'esperimento effettuato.

```
1       void FixedUpdate()
2       {
3           if (handComponent.IsTracked)
4           {
5               isTracked = true;
6               _frameTimeSkip ++;
7               if (!_isInitialized)
8               {
9               _currentArmature = _currentHand.transform.Find("Bones");
10              InitializeReferences(_currentArmature); }
11          }
12          else
13          { isTracked = false; }
14          if(_frameTimeSkip >= framToBeSkipped && isTracked) {
15              _frameTimeSkip = 0;
16              if (record)
17          {
```

```
18                  _frameCounter ++;
19                  StoreHandPosition(_currentArmature, _frameCounter);
20              }
21              }
22      }
```

Qui è stata riportata una breve sezione del codice scritto per immagazzinare le informazioni di movimento della mano virtuale comandata dal giocatore durante i trial. Durante ogni FixedUpdate infatti si effettuano dei controlli per stabilire se si vuole registrare le informazioni di posizione della mano. Viene infatti verificato che la mano sia tracciata correttamente dal visore, per non considerare valori inesatti, e si controlla inoltre che tutte le ossa siano state inizializzate correttamente. Se le condizioni si verificano si salvano i dati di posizione del polso della mano in una lista, aumentando un contatore che tiene conto di quanti frame sono stati registrati e quindi dell'indice della lista appena aggiornata. Considerando l'impatto sulle prestazioni che il salvataggio potesse avere su qualche dispositivo si è scelto di aggiungere la possibilità di saltare qualche frame impostando la quantità di essi da non considerare tra un salvataggio e il successivo (frameToBeSkipped). Se questo valore viene mantenuto a 1, si salveranno i valori di ogni frame.

3.3 Estensibilità

Durante le fasi di sviluppo dell'applicazione si sono create varie versioni, le quali venivano di volta in volta modificate e affinate per venire incontro ai vincoli di sperimentazione e di ricerca. Per questo motivo si è cercato di rendere l'applicazione più estendibile possibile, così da non richiedere eccessivo lavoro nel caso in cui determinate funzionalità richiedessero variazioni. La lunghezza della sequenza in fase di prototipazione iniziale era variabile, perciò è stato progettato un parametro associato allo script manager di gioco (GrabGameManager) per determinare quanto deve essere lunga. Vi sono all'interno della scena di gioco Unity cinque preset differenti, con le relative disposizioni dei marcatori sulla lavagna, dove cioè vengono posizionate le istanze degli oggetti appartenenti alla sequenza attuale. Vi è di conseguenza uno script dedicato a regolare l'alternarsi di questi preset: ActiveBoardSwitch.cs . Esso infatti contiene un metodo per attivare la disposizione con 3,4,5,7 oppure 9 marcatori, andando a disattivare le altre, per non renderle visibili ed interagibili. Lo sviluppo dell'applicazione prevede anche la possibilità di attivare una successione di livelli, differenziati tra di loro dalla presenza di un telo a copertura della sequenza o meno e dalla lunghezza della serie di oggetti. Questa successione permette infatti di consegnare agli utenti un' esperienza piú variegata e godibile nel tempo, con una difficoltà variabile e avendo tempo di sperimentare le meccaniche di interazioni con livelli molto facili all'inizio. Vi è anche in questo un caso uno script che regola il comportamento (LevelManager.cs) dei livelli, infatti

al suo interno è possibile modificare le lunghezze delle sequenze devi fari livelli, attivare o disattivare il telo e determinare il numero di punti necessari al passaggio tra uno stato e il successivo. La difficoltá della realizzazione dei compiti richiesti puó essere modulata inoltre tramite altri 3 parametri: il tempo massimo da poter dedicare a ciascuno trial prima del Time Out; la velocitá di movimento degli oggetti, e quindi della velocità di scorrimento delle rampe; l'introduzione del concetto di numero di vite il quale esaurimento farebbe concludere la partita. I primi due parametri sono stati impostati su un valore definito in seguito a varie prove di utilizzo dell'applicazione, in modo da permettere agli utenti di completare i compiti richiesti, ma mantenendo comunque un livello di sfida che potesse ingaggiarlo nel gioco. Sará comunque opportuno in futuro cercare i valori ottimali in caso in cui l'applicazione verrà utilizzata da soggetti utilizzatori di protesi, invece che da sperimentatori sani. Si prevede infatti che sarà richiesto un tempo maggiore e una velocità di movimento degli oggetti inferiore. L'inserimento delle vite nel gioco invece rappresenta una funzionalità che é stata rimossa in fase di stesura del protocollo sperimentale, ma che essendo già implementata potrà essere utilizzata per creare una versione sfidante e che si presta ad essere rigiocata piú volte.

3.4 Interazione

Durante le fasi di sviluppo dell'applicazione si sono create varie versioni, le quali venivano di volta in volta modificate e affinate per venire incontro ai vincoli di sperimentazione e di ricerca. Per questo motivo si è cercato di rendere l'applicazione più estendibile possibile, così da non richiedere eccessivo lavoro nel caso in cui determinate funzionalità richiedessero variazioni. La lunghezza della sequenza in fase di prototipazione iniziale era variabile, perciò è stato progettato un parametro associato allo script manager di gioco (GrabGameManager) per determinare quanto deve essere lunga. Vi sono all'interno della scena di gioco Unity cinque preset differenti, con le relative disposizioni dei marcatori sulla lavagna, dove cioè vengono posizionate le istanze degli oggetti appartenenti alla sequenza attuale. Vi è di conseguenza uno script dedicato a regolare l'alternarsi di questi preset: ActiveBoardSwitch.cs . Esso infatti contiene un metodo per attivare la disposizione con 3,4,5,7 oppure 9 marcatori, andando a disattivare le altre, per non renderle visibili ed interagibili. Lo sviluppo dell'applicazione prevede anche la possibilità di attivare una successione di livelli, differenziati tra di loro dalla presenza di un telo a copertura della sequenza o meno e dalla lunghezza della serie di oggetti. Questa successione permette infatti di consegnare agli utenti un' esperienza piú variegata e godibile nel tempo, con una difficoltà variabile e avendo tempo di sperimentare le meccaniche di interazioni con livelli molto facili all'inizio. Vi è anche in questo un caso uno script che regola il comportamento (LevelManager.cs) dei livelli, infatti

al suo interno è possibile modificare le lunghezze delle sequenze devi fari livelli, attivare o disattivare il telo e determinare il numero di punti necessari al passaggio tra uno stato e il successivo. La difficoltá della realizzazione dei compiti richiesti puó essere modulata inoltre tramite altri 3 parametri: il tempo massimo da poter dedicare a ciascuno trial prima del Time Out; la velocitá di movimento degli oggetti, e quindi della velocità di scorrimento delle rampe; l'introduzione del concetto di numero di vite il quale esaurimento farebbe concludere la partita. I primi due parametri sono stati impostati su un valore definito in seguito a varie prove di utilizzo dell'applicazione, in modo da permettere agli utenti di completare i compiti richiesti, ma mantenendo comunque un livello di sfida che potesse ingaggiarlo nel gioco. Sará comunque opportuno in futuro cercare i valori ottimali in caso in cui l'applicazione verrà utilizzata da soggetti utilizzatori di protesi, invece che da sperimentatori sani. Si prevede infatti che sarà richiesto un tempo maggiore e una velocità di movimento degli oggetti inferiore. L'inserimento delle vite nel gioco invece rappresenta una funzionalità che é stata rimossa in fase di stesura del protocollo sperimentale, ma che essendo già implementata potrà essere utilizzata per creare una versione sfidante e che si presta ad essere rigiocata piú volte.

Capitolo 4

Analisi dei Dati

Si è scelto di testare l'applicazione in via preliminare su un campione di soggetti non amputati, in modo tale da verificare il grado di usabilitá dell'esperienza e fornire una validazione delle ipotesi della ricerca. Al fine di valutare quanto il livello di sfida in un gioco immersivo influenzi il grado di embodiment percepito è stato necessario raccogliere informazioni quantitative e qualitative circa questo sentimento. L'embodiment provato è misurato tramite il test della Rubber Hand Illusion, mediante il quale si ottiene un valore numerico corrispondente alla posizione percepita del proprio dito indice prima e dopo una stimolazione sensoriale. In particolare viene effettuata una pressione sincrona su una mano artificiale posta di fronte al partecipante e sopra la sua mano reale, coperta però da un telo che ne esclude la visione. La disposizione necessità una configurazione simile a quella realizzata per questo esperimento, in figura:

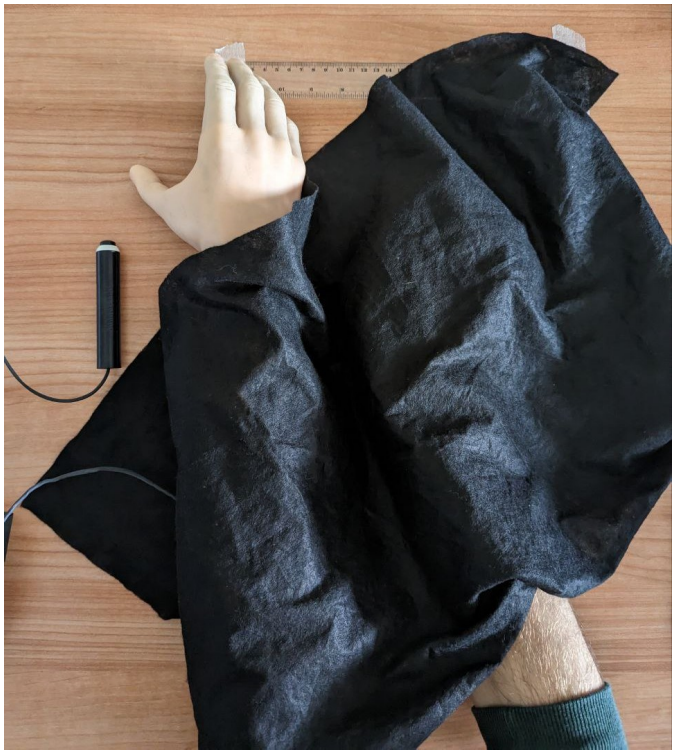


Figura 4.1: Test della Rubber Hand Illusion

In particolare in questa configurazione si è utilizzato un sistema composto da una scheda Arduino Uno, un bottone posto in cima ad un'asta cilindrica e un motorino vibrante. L'interconnessione di questi componenti permette di esercitare una pressione sulla mano artificiale, premendo così il bottone e, in modo sincrono, azionare l'attuatore vibrante sulla mano reale. Il motorino veniva posizionato sull'avambraccio dei soggetti testati, nella posizione dove verrebbe posizionato nel caso di soggetti con mano amputata. La stimolazione è ripetuta per 5 minuti, al termine dei quali viene chiesto di indicare la posizione del proprio indice tenendo gli occhi chiusi. Ottenuti i dati della posizione percepita prima e dopo la stimolazione sincrona, e il dato reale di dove è situato l'indice, otteniamo tre misure con le quali calcolare il drift percettivo. La misura del drift ci indica a livello oggettivo(perché la persona non vede la propria mano) il livello di embodiment percepito verso il proprio arto . Questo perché l'embodiment sviluppato viene generalizzato per la mano protesica, quindi un drift elevato, positivo verso la mano artificiale, ci indica che la sensazione di embodiment è elevata. Dalle sessioni di RHI abbiamo tratto i risultati mostrati nel grafico seguente.

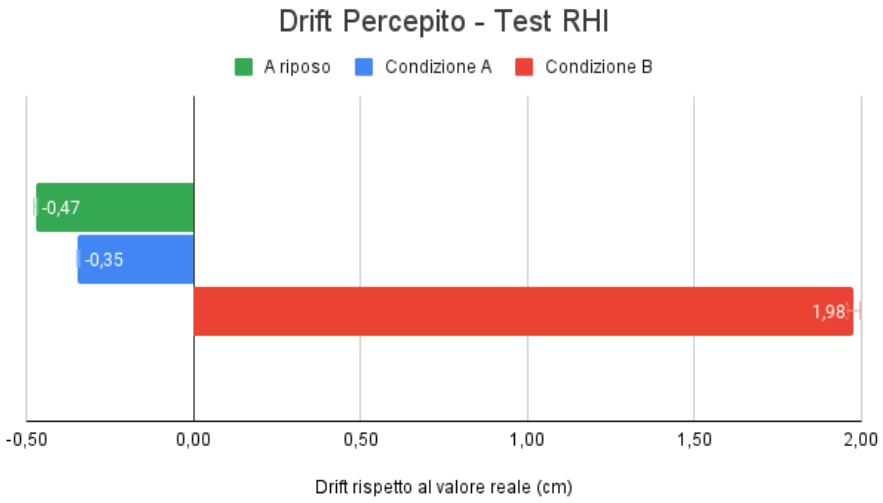


Figura 4.2: Medie del calcolo del drift nel test Rubber Hand Illusion

Il test RHI ha mostrato una differenza significativa (W=7 e p=0,01) tra la deriva verso sinistra della condizione senza compito di memoria (M=-0,11 cm e SD=2,27 cm) e la deriva verso destra della condizione con compito di memoria (M=1,99 cm e SD=2,64 cm). I valori indicati con W e p derivano dal Wilcoxon signed-rank test. Questo test non parametrico (poichè le misure sono su scala ordinale o non seguono distribuzione normale) serve per confrontare dati in due condizioni, secondo un disegno sperimentale *within-subject*, in cui ogni soggetto è sottoposto a tutte le condizioni in ordine randomizzato.

4.1 Risultati Questionari

Il primo questionario al quale abbiamo sottoposto i partecipanti è il Nasa-TLX (Task Load Index) sviluppato dallo Human Performance Group del centro NASA per calcolare l'indice di carico di lavoro. Esso contiene 6 domande in scala da 1 a 20 che vertono sul carico mentale fisico e le sensazioni come frustrazione durante una determinata esperienza con un servizio. Successivamente abbiamo richiesto la compilazione di un ulteriore questionario per raccogliere informazioni sull'esperienza nell'ambiente virtuale. Questo form prevedeva domande create appositamente per discutere del task appena effettuato. I risultati di queste domande sono stati utili per comprendere informazioni sull'usabilità generale dell'applicazione e valutare il grado di immersione percepito. Infine un questionario con domande specifiche riguardanti la sessione di Rubber Hand Illusion é stato somministrato una volta terminato il suddetto test e registrata la distanza per il calcolo del drift.

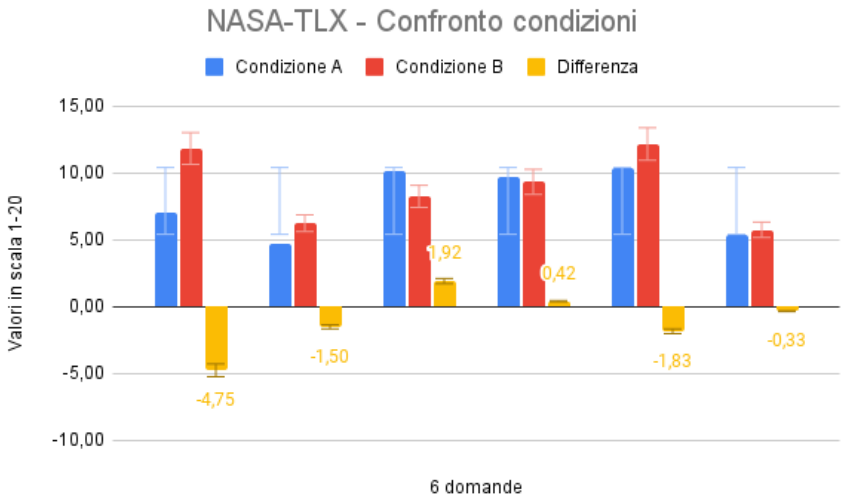


Figura 4.3: Questionario NASA-TLX

Nel questionario NASA-TLX sono state riscontrate differenze statisticamente significative (con p<0,05) solo per la domanda 1 sul carico di lavoro cognitivo (W=5,5 e p=0,01) con M=7,08 e SD=4,46 nella condizione senza compito di memoria, M=12,42 e SD=4,25 nella condizione con compito di memoria. Inoltre, una sola scala del questionario ad hoc ha evidenziato differenze significative (con p<0,05): l'item 8 (W=31,5 e p=0,04), basato su un elemento SUS, osserva che sono state percepite più incongruenze nella condizione senza compito di memoria (M=1,91 con p=0,76) rispetto alla condizione con compito di memoria (M=1,42 e SD=0,64).

4.2 Analisi dei Risultati

I risultati mostrano un livello di richiesta mentale più elevato per il compito di memoria, come previsto. Indice del fatto che le condizioni sono state differenziate in modo corretto per raccogliere evidenze sperimentali dell' influenza dei task cognitivi sulla formazione di embodiment. Il bias verso sinistra nella deriva del test della Rubber Hand Illusion, per quanto riguarda la condizione senza compito di memoria, potrebbe indicare la richiesta di risorse cognitive durante il processo di embodiment. Quindi nel condizione con memoria asisteremmo ad un'interferenza nel processo di formazione di embodiment che ne compromette un´efficace sviluppo. Dal questionario NASA-TLX si puó constatare che l'applicazione sia poco impattante dal punto di vista delle richieste di sforzo fisico, con valore medio di risposta alla q2 M= 5,5 e SD=3,61, calcolati su entrambe le condizioni. La differenza nella media delle

risposte per le due condizioni varia di 1.90, quindi in maniera relativamente ridotta, infatti si ha M=4,75 per la condizione A e M=6,25 per la condizione B. Dallo stesso questionario emerge la mancanza di frustrazione media delle persone che hanno partecipato all'esperienza, con media complessiva M=5,50 e SD=4,42. Ulteriori informazioni sull'esperienza virtuale sono state estrapolate dai dati del quesionario ad hoc, che ci permettono di trarre conclusioni circa l'usabilità dell'applicazione.

Risposte Questionario Virtual Session					
q	Domande Questionario	Condizione A		Condizione B	
		M	SD	M	SD
1	The session time was too long	1,75*	0,48	1,67*	0,98
2	After this session I feel quite stressed	2,00*	0,96	1,67*	1,37
3	I think that I would like to use this system frequently	3,42	0,66	3,58	1,21
4	I found the system unnecessarily complex	1,42*	0,39	1,17*	0,95
5	I thought the system was easy to use	3,67	0,94	3,83	1,41
6	I think that I would need the support of a technical person to be able to use this system	1,58*	0,72	1,75*	1,00
7	I found the various functions in this system were well integrated	4,08	0,94	4,17	1,31
8	I thought there was too much inconsistency in this system	2,00*	0,66	1,42*	0,95
9	I would imagine that most people would learn to use this system very quickly	4,00	0,78	1,59	4,50
10	I found the system very cumbersome to use	2,25	1,16	2,58	1,19
11	I felt very confident using the system	3,58	0,79	3,92	1,35
12	I needed to learn a lot of things before I could get going with this system	1,73*	0,57	1,82*	0,94
13	I think I was getting skilled in this task	3,64	0,79	4,18	1,50
14	It was difficult to memorize the sequence	2,64	1,31	3,18	1,36
15	It was easy to grab the objects	3,00	1,37	3,64	1,48
16	It was easy to drop the objects	3,55	1,43	3,82	1,60
17	It was difficult to follow the pace of the task	1,91	0,89	2,45	1,35
18	I felt located into the virtual environment at the point that I was ignoring the surroundings	3,82	0,94	4,09	1,75
19	I perceived the virtual hand as part of my body	4,09*	0,98	3,64	1,45
20	The objects were easy to recognise	4,82*	0,39	4,82*	1,50
21	I experienced fatigue at my right arm	2,00*	0,98	1,73*	1,13
22	The setting should have been more playful	1,73*	0,79	1,82*	0,93

In particolare il gioco è risultato essere di breve durata con risposte di media

M=1.69 e SD=0,73 alla domanda “La sessione risultava essere troppo lunga?”. Sintomo che i giocatori avrebbero potenzialmente giocato piú a lungo. Inoltre i giocatori hanno definito l’esperienza come poco stressante e non complessa in modo non necessario, rispondendo complessivamente nelle due condizioni alla q2 con M=1,77 e SD=1,11 e alla q4 con M=1,42 e SD=0,81. Non risultavano differenze sostanziali per le due configurazioni quindi i valori complessivi rispecchiano una condizione generale del gioco. Per quanto riguarda la necessitá di avere un tecnico per spiegare le funzionalitá si puó affermare quanto detto per le domande precedenti, con risposte alla domanda q6 con media M=1,69 e SD=0,79. Alla domanda q8, che si riferisce all’inconsistenza nel gioco, si sono ottenuti risultati positivi sul fronte dell’usabilitá, con media M=1,69 e SD=0,79, ma occorre tener conto la differenza delle condizioni con una differenza di punteggio medio pari a 0,58. È interessante notare la differenza tra le condizioni, considerando quella senza compito di memoria, che fa percepire più incongruenze. Il dato potrebbe significare che i soggetti, lasciati più liberi di agire, senza dover ricordare sequenze di oggetti, abbiano prestato piú attenzione ai dettagli dello scenario. In particolare piú attenzione ad imprecisioni che il tracciamento delle mani puó registrare in codizioni di scarsa illuminazione o occlusioni. Il sistema risulta essere poco faticoso per l’arto utilizzato per l’interazione, con M=1,83 e SD=0,99, rispecchiando quando emerso dal NASA-TLX per la richiesta di sforzo fisico. Infine i valori della q22, nella quale viene domandato se fosse chiesto se il sistema dovesse essere piú divertente e presentare piú elementi ludici, gli utenti hanno risposto univocamente alle condizioni con M=1,83 e SD=0,76.

4.3 Feedback Playtesting

Durante la fase di sperimentazione sono stati raccolti pareri e risolti problemi che non erano stati previsti in fase di design e migliorate alcune caratteristiche. La prima caratteristica a subire un aggiornamento è stata la generazione degli oggetti sulle rampe. Durante le prime partite degli partecipanti si è notato che alcuni giocatori che perdevano degli oggetti, non riuscendo ad afferrarli in tempo, dovevano aspettare anche più di un ciclo per poter ritentare la presa dello specifico oggetto. Per questo si è scelto di cambiare l’algoritmo di creazione dei suddetti oggetti. Infatti, se anche prima venivano generati 3 diversi oggetti sulle 3 rampe di fronte all’utente, dopo la modifica, al ciclo successivo si è deciso di creare l’oggetto che non è stato possibile afferrare al ciclo prima. In questo modo si aumenta la probabilità che ogni oggetto si presenti sulle rampe e si diminuisce la frustrazione di dover aspettare senza svolgere alcun compito. è stato raccolto un ulteriore riscontro sulle funzionalità dell’applicazione. Questa volta si è voluto correggere un problema di usabilitá, infatti si è notato che alcuni utenti hanno riscontrato problemi nel

depositare gli oggetti di fianco a loro, incorrendo in ritardi nel tracciamento e/o tracciamenti erronei. Questo comportamento si presentava nei casi di persone che tendevano ad appoggiare gli oggetti all'interno del carrello, invece di lasciarli cadere da un'altezza maggiore. Questo comportamento poteva portare ad occlusioni visive tra le videocamere inserite nel visore e la mano degli utilizzatori, dovendo posizionare la mano molto in basso, a volte al di sotto della sedia su cui erano situati. Sebbene il problema si verificasse solo con alcuni sperimentatori si è deciso di trovare una soluzione di sviluppo a questa problematica. Il carrello dove era richiesto di posizionare gli oggetti è stato rialzato, sollevando i bordi in altezza. In questo modo si è ridotto il numero delle problematiche di trcciamente, spingengo gli utenti a mantenere la loro mano piú in alto.

4.4 Raccolta dati di Gioco

Ulteriori analisi possono essere portate avanti a partire dai dati di gioco raccolti al termine delle partite di ogni partecipante. In quel momento viene infatti creato un file contenente il tempo impiegato per concludere ogni trial, la percentuale di trial completati con successo e il numero di errori totale e per tentativo. In aggiunta ad esso i dati di posizione della mano possono essere visualizzati in modi differenti a seconda della specifica richiesta. Infatti al termine della partita si puó scegliere di attivare la modalità *replay* quando si è ancora nel gioco, essa attiva una mano virtuale che va a ripercorrere i movimenti esatti compiuti dall'utente nella precedente sessione. Questa visualizzazione puó essere utile in caso in cui si voglia permettere ai partecipanti di avere una rappresentazione diretta delle proprie dinamiche di movimento. Utilizzando i dati numerici è possibile invece ottenere visualizzazioni differenti, come ad esempio Point Clouds oppure Heatmap della sezione bidimensionale di un piano dello spazio 3D. Il grafico della Point Cloud appare come nella seguente figura, che mostra punti in corrispondenza delle coordinate dello spazio reale occupate dalla mano in ogni frame. La HeatMap rappresenta sempre un range di punti nello spazio, ma contiene anche informazioni di densitá. Infatti il colore di ogni punto appare più caldo, contiene cioè più componente rossa nella rappresentazione RGB, più sono i punti presenti nelle vicinanze. Il risultato finale indica le zone nelle quali la mano del partecipante sono passate più frequentemente. I dati raccolti possono rappresentare un'importante risorsa per future interpretazioni dei risultati.

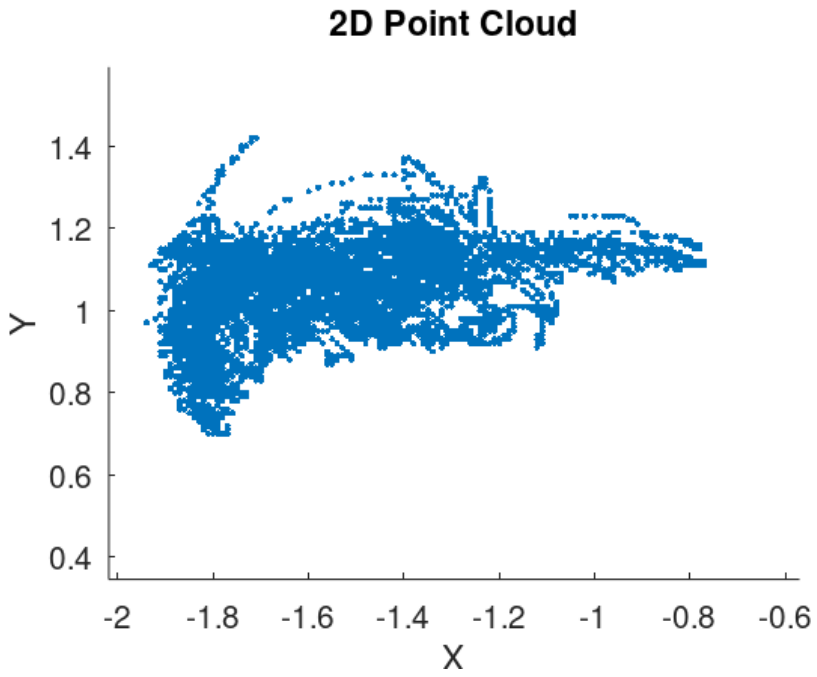


Figura 4.4: Point Cloud dati di posizione

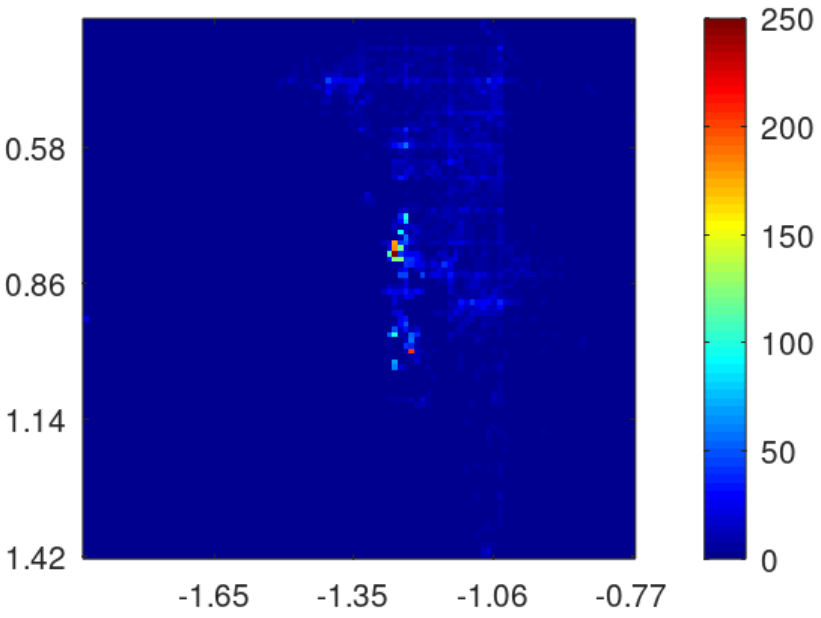


Figura 4.5: Heat Map dati di posizione

Conclusioni

I risultati ottenuti dai test di Rubber Hand Illusion e dai questionari sono risultati soddisfacenti al fine di questo studio, sono state infatti formulate nuove ipotesi sugli effetti che condizionano i processi di formazione dell'embodiment. L'esperienza realizzata prevedeva due configurazioni, e una di esse è risultata essere considerevolmente meno efficace in termini di embodiment percepito, secondo i dati raccolti immediatamente dopo la fruizione. La differenza tra le due impostazioni risulta essere sufficientemente marcata per trarre conclusioni sulle complicazioni che richieste di compiti cognitivi possono introdurre nel processo di formazione di embodiment. Si ipotizza quindi che, perchè si generi una sensazione di embodiment, sono richieste delle risorse cognitive, risorse che possono essere condivise con altri processi mentali, in questo caso quelli responsabili della memorizzazione a breve termine di una sequenza. Alla fase di sperimentazione hanno partecipato tredici soggetti non amputati, che hanno provato entrambe le condizioni sperimentali in ordine randomizzato. Ai fini dei risultati è stato possibile trarre delle evidenze sperimentali dal campione analizzato, tuttavia ripetere il test su un numero maggiore di individui potrebbe produrre ulteriori risultati interessanti. Si potrebbe infatti verificare come si comporti la deriva nel test della RHI considerando un numero maggiore di individui. In termini di benessere per i soggetti utilizzatori di protesi lo studio ha contribuito a indagare le metodologie per migliorare l'integrazione dei dispositivi protesici utilizzati, individuando fattori che potrebbero impedirne la sua riuscita. Le diverse condizioni create all'interno dell'ambiente sono risultate sufficientemente differenziate. L'applicazione risulta quindi essere un valido ambiente di sperimentazione per ulteriori configurazioni differenti in futuro. L'esperienza ha inoltre ricevuto dei valori positivi in termini di usabilitá generale dai questionari compilati dai partecipanti alle sperimentazioni. In particolare il gioco è risultato essere di facile comprensione ed apprendimento, non eccessivamente oneroso nell'impegno fisico richiesto e la durata è risultata essere piú breve rispetto a quanto potrebbe. I dati raccolti forniscono prospettive interessanti in ottica di sviluppi futuri dell'applicazione e in ottica di ricerche ulteriori. Infatti puó essere interessante ampliare la ricerca sui processi che favoriscono la formazione di embodiment, indagando l'influenza di altri elementi di gioco o propri

delle esperienze virtuali in generale. A questo proposito si potrebbe analizzare in dettaglio il contributo dell'impegno fisico richiesto o della durata complessiva dell'esperienza. Considerando il grado di usabilitá ottenuto dal gioco, si potrebbe pensare ad un eventuale utilizzo dello stesso come *exergame*, ovvero come gioco da utilizzare durante le sedute di riabilitazione motoria.

Bibliografia

[1] Raphaela Kilteni Konstantina Groten e Mel. Slater. «The Sense of Embodiment in Virtual Reality. Presence Teleoperators amp Virtual Environments.» In: (2012) (cit. alle pp. 3, 4).

[2] Ambron E, Miller A, Kuchenbecker KJ, Buxbaum LJ e Coslett HB. «ImmersiveLow-Cost Virtual Reality Treatmentfor Phantom Limb Pain: Evidence from Two Cases». In: (2018) (cit. alle pp. 3, 5).

[3] Nyström B e Hagbarth KE. «Microelectrode recordings from transected nerves in amputees with phantom limb pain». In: (1981) (cit. a p. 5).

[4] Melzack R, Israel R, Lacroix R e Schultz G. «Phantom limbs in people with congenital limb deficiency or amputation in early childhood». In: *Brain* 120(9) (set. 2021), pp. 1603–1620 (cit. a p. 5).

[5] Brugger P, Kollias SS, Müri RM, Crelier G, Hepp-Reymond MC e Regard M. «Beyond re-membering: phantom sensations of congenitally absent limbs». In: *ProcNatl Acad Sci USA* 97(11) (2000), pp. 6167–72 (cit. a p. 5).

[6] Wikipedia contributors. *MS Windows NT Kernel Description*. 2021 (cit. a p. 5).

[7] Robina Bekrater-Bodmann, Iris Reinhard, Martina Diers, Xaver Fuchs e Herta Flor. «Relationship of prosthesis ownership and phantom limb pain: results of a survey in 2383 limb amputees». In: *PAIN* 162(2) (feb. 2021), pp. 630–640 (cit. a p. 5).

[8] Flor H, Diers M e Andoh J. «The neural basis of phantom limb pain». In: *Trends CognSci* 17(7) (2013), pp. 307–8 (cit. a p. 5).

[9] Barresi G et al. «Exploring the Embodiment of a Virtual Hand in a Spatially Augmented Respiratory Biofeedback Setting». In: *Neurorobot.* (2021) (cit. a p. 5).

[10] Xu A, Cullen B, Penner C, Zimmerman C, Kerr C e Schmalzl L. «Comparing embodiment experiences in expert meditators and nonmeditators using the rubber hand illusion». In: *Conscious Cogn* 65 (2018), pp. 325–33 (cit. a p. 6).

[11] Gamification Staff. *Cos'è la Gamification*. 2010. URL: https://www.gamification.it/gamification/introduzione-alla-gamification/ (visitato il 26/03/2023) (cit. a p. 7).

[12] Tuah N M, Ahmedy F, Gani A e Yong L N. «A Survey on Gamification for Health Rehabilitation Care: Applications, Opportunities, and Open Challenges». In: *Information* 12 (2021), p. 91 (cit. a p. 8).

[13] Keshner EA, Weiss PT, Geifman D e Raban D. «Tracking the evolution of virtual reality applications to rehabilitation as a field of study». In: *J Neuroeng Rehabil* 16(1) (Jun 21), p. 76 (cit. a p. 8).

[14] Yeung AWK et al. «Virtual and Augmented Reality Applications in Medicine: Analysis of the Scientific Literature». In: *J Med Internet Res* 23(2) (Feb 21) (cit. a p. 8).

[15] Rebecca Lo Tauro e Federico Tessari Carlo Ferraresi Giacinto Barresi. «Studio di fattibilità sulle strategie di feedback e controllo in un setting interattivo per l'embodiment di un arto inferiore virtuale.= Feasibility study on feedback and control strategies in interactive settings for embodying a virtual lower limb». Tesi di dott. Politecnico di Torino, Corso di laurea magistrale in Ingegneria Biomedica, 2022 (cit. alle pp. 9, 10).

[16] Ambron E, Miller A, Kuchenbecker KJ, Buxbaum LJ e Coslett HB. «Immersive Low-Cost Virtual Reality Treatment for Phantom Limb Pain: Evidence from Two Cases». In: *Front. Neurol.* 9 (2018), p. 67 (cit. a p. 11).

[17] Caserta Giulia, Carlo A Frigo, Nicolò Boccardo, Giacinto Barresi e Marco Freddolini. «Evaluation and improvement of the embodiment of an upper limb prosthesis: the Hannes system». Tesi di dott. Politecnico di Milano, Master of Science in Biomedical Engineering, 2020 (cit. a p. 14).

[18] Candidate: Dario Di Domenico Academic Supervisor: Prof. Marcello Chiaberge Company Supervisors: Eng. Nicolò Boccardo Eng. Andrea Marinelli. «Hannes Prosthesis Control Based on Regression Machine Learning Algorithms». Tesi di dott. Politecnico di Torino, Corso di laurea magistrale in Ingegneria Meccatronica, 2020 (cit. a p. 15).

[19] Shangchen Han, Beibei Liu, Tsz Ho Yu, Randi Cabezas, Peizhao Zhang, Peter Vajda, Eldad Isaac e Robert Wang. *Using deep neural networks for accurate hand-tracking on Oculus Quest*. 2019. URL: https://ai.facebook.com/blog/hand-tracking-deep-neural-networks/ (visitato il 26/03/2023) (cit. a p. 25).