

# POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria Informatica



**Politecnico  
di Torino**

Tesi Magistrale

*Sense of Agency* ed ERP correlati su VR  
ritardata

Relatori

Prof. Andrea BOTTINO

Prof. Francesco STRADA

Candidato

**Fabio CIELO**

Anno accademico 2022/2023



# Sommario

Negli ultimi anni la realtà virtuale è diventata una tecnologia sempre più utilizzata in molteplici ambiti. La tendenza a produrre visori ad un prezzo sempre più accessibile, insieme alla grande libertà di personalizzazione e immersività che un ambiente virtuale creato ad hoc offre, sono i principali motivi per cui la realtà virtuale trova applicazione non solo nell'ambito videoludico, ma anche nel campo dell'istruzione e della ricerca medica.

Il presente elaborato illustra la progettazione, lo sviluppo e l'analisi di un sistema di realtà virtuale immersivo mirato allo studio del *Sense of Agency* e dei *potenziali evento-correlato* legati ad esso. La caratteristica principale su cui si basa lo studio è l'introduzione di un *delay* tra il movimento delle mani effettuato dal soggetto sperimentale e il corrispettivo movimento delle mani virtuali visibili all'interno dell'ambiente di realtà virtuale. La raccolta dei dati tramite una elettroencefalografia e la successiva analisi è stata effettuata in collaborazione con il dipartimento di neuroscienze dell'Università di Torino. Tramite l'analisi di dati ricavati da questionari post-sperimentali somministrati ai soggetti si è voluto inoltre indagare la variazione della sensazione di *embodiment* provato dai soggetti in relazione al *delay* impostato nell'ambiente virtuale.

# Ringraziamenti

Finalmente ci siamo!

Dopo tanti anni sono giunto al termine di questo percorso universitario. È vero, ci sono stati momenti in cui ho faticato a passare alcuni esami, momenti dove avrei voluto dire «*Bello, ma anche basta.*». Ma alla fine eccoci qua. Il merito però non è tutto mio, anzi! Senza le persone che mi sono state accanto, che mi hanno sopportato e supportato incondizionatamente, ~~probabilmente~~ sicuramente non ce l'avrei fatta.

Vorrei quindi dedicare questa tesi alle persone più importanti della mia vita. Ai miei genitori, **Silvia** e **Beppe**, e mio fratello, **Paolo**, ai quali (anche se non glielo dico mai) voglio un mondo di bene! A **Giulia**, che mi è stata accanto, accompagnandomi e soprattutto spronandomi, in questi abbondanti anni di università, facendomi andare avanti nei momenti in cui mi arenavo un po'.

Un grazie a tutti i miei amici e parenti per aver alleggerito i week end, soprattutto quelli durante le sessioni esame, ed esserci stati lungo il cammino.

Grazie di cuore,

*Fabio*



# Indice

|   |      |
|---|------|
| <b>Elenco delle tabelle</b>                         | VII  |
| <b>Elenco delle figure</b>                          | VIII |
| <b>1 Introduzione</b>                               | 1    |
| 1.1 Il mercato del virtuale . . . . .               | 2    |
| 1.2 VR e Neuroscienze . . . . .                     | 3    |
| 1.3 Sense of Agency . . . . .                       | 6    |
| 1.4 Potenziale evento-correlato . . . . .           | 7    |
| 1.4.1 Readiness Potential . . . . .                 | 8    |
| <b>2 Stato dell'arte</b>                            | 10   |
| 2.1 Sperimentazioni attuali . . . . .               | 10   |
| 2.1.1 Sperimentazioni sul sense of agency . . . . . | 11   |
| 2.2 Il progetto . . . . .                           | 11   |
| <b>3 Implementazione</b>                            | 13   |
| 3.1 Ambiente Virtuale . . . . .                     | 14   |
| 3.1.1 Scelta dell' HMD . . . . .                    | 14   |
| 3.1.2 Scelta del motore grafico . . . . .           | 15   |
| 3.1.3 Struttura ambiente virtuale . . . . .         | 16   |
| 3.1.4 Tracking delle mani . . . . .                 | 16   |
| 3.1.5 Server e scambio dati . . . . .               | 18   |
| 3.1.6 Delay . . . . .                               | 21   |
| 3.1.7 Specchiamento movimento locale . . . . .      | 30   |

|          |   |           |
|----------|---|-----------|
| 3.1.8    | Specchiamento movimento soggetto . . . . .      | 33        |
| 3.1.9    | Cambio punto di vista . . . . .                 | 33        |
| 3.2      | Sistema di LOG . . . . .                        | 34        |
| 3.2.1    | LOG generati dal visore . . . . .               | 34        |
| 3.3      | Software Gestione VR . . . . .                  | 39        |
| <b>4</b> | <b>Sperimentazione</b>                          | <b>42</b> |
| 4.1      | Sistema lettura EEG . . . . .                   | 42        |
| 4.2      | Pre-sperimentazioni . . . . .                   | 44        |
| 4.2.1    | Pulsante . . . . .                              | 45        |
| 4.3      | Protocollo sperimentale . . . . .               | 46        |
| 4.3.1    | Raccolta dati . . . . .                         | 47        |
| <b>5</b> | <b>Risultati</b>                                | <b>53</b> |
| 5.1      | Campione sperimentale . . . . .                 | 53        |
| 5.2      | Risultati oggettivi . . . . .                   | 55        |
| 5.2.1    | Metodi di Analisi . . . . .                     | 55        |
| 5.2.2    | Risultati . . . . .                             | 55        |
| 5.2.3    | Discussione . . . . .                           | 56        |
| 5.2.4    | Limiti dello studio e sviluppi futuri . . . . . | 58        |
| 5.3      | Risultati soggettivi . . . . .                  | 58        |
| 5.3.1    | Embodiment e delay . . . . .                    | 60        |
| 5.4      | Dati oggettivi e soggettivi . . . . .           | 63        |
| <b>6</b> | <b>Conclusioni</b>                              | <b>66</b> |
|          | <b>Bibliografia</b>                             | <b>67</b> |

# Elenco delle tabelle

|     |   |    |
|-----|---|----|
| 3.1 | Azioni gestite dal server . . . . .           | 20 |
| 3.2 | File generati dal server . . . . .            | 35 |
| 3.3 | Codici azioni usati nei file di LOG . . . . . | 38 |
| 5.1 | Risultati soggettivi esperienza VR . . . . .  | 59 |
| 5.2 | Test T. calcolato su domande QB . . . . .     | 61 |
| 5.3 | Indici di correlazione di Pearson . . . . .   | 65 |



# Elenco delle figure

|      |   |    |
|------|---|----|
| 1.1  | View-Master della Sawyer's Inc. insieme a dei dischetti . . . . . | 2  |
| 1.2  | Mattel Power Glove per il Nintendo NES (1989) . . . . .           | 2  |
| 1.3  | Ricavi mercato XR (consumer e enterprise) . . . . .               | 3  |
| 1.4  | Dispositivi AR e VR spediti a livello mondiale . . . . .          | 4  |
| 1.5  | Onda RP . . . . .   | 8  |
| 3.1  | Schema interazione componenti . . . . .                           | 13 |
| 3.2  | Meta Quest 2 . . . . .  | 14 |
| 3.3  | Logo Unreal Engine . . . . .                                      | 16 |
| 3.4  | Logo Unity . . . . .  | 16 |
| 3.5  | Ambiente virtuale . . . . .                                       | 17 |
| 3.6  | Controller Meta Quest 2 . . . . .                                 | 17 |
| 3.7  | Buffer - Fase preliminare . . . . .                               | 23 |
| 3.8  | Buffer - Passo 1 . . . . .  | 23 |
| 3.9  | Buffer - Passo 2 . . . . .  | 24 |
| 3.10 | Buffer - Passo 3 . . . . .  | 24 |
| 3.11 | Buffer - Passo 4 . . . . .  | 25 |
| 3.12 | Buffer - Passo 5 . . . . .  | 25 |
| 3.13 | Buffer - Passo 6 . . . . .  | 26 |
| 3.14 | Buffer - Stato finale . . . . .                                   | 26 |
| 3.15 | VirtualHand sistema di tracking . . . . .                         | 29 |
| 3.16 | VirtualHand buffer . . . . .                                      | 30 |
| 3.17 | Schermata IP . . . . .  | 40 |
| 3.18 | Schermata Nuovo Blocco . . . . .                                  | 40 |

|      |  |    |
|------|--|----|
| 3.19 | Schermata Nuova Sessione . . . . .   | 40 |
| 3.20 | Schermata Nuovo Blocco . . . . .   | 40 |
| 3.21 | Schermata di configurazione della stanza . . . . .                         | 41 |
| 3.22 | Schermata di attivazione della sessione . . . . .                          | 41 |
| 4.1  | Schema completo del sistema . . . . .                                      | 43 |
| 4.2  | Esempio tracciato EEG . . . . .  | 44 |
| 4.4  | Modello 3D del pulsante . . . . .  | 46 |
| 4.5  | Protocollo Sperimentale . . . . .  | 47 |
| 4.3  | Schema esecuzione sessione sperimentale . . . . .                          | 52 |
| 5.1  | Composizione dei soggetti partecipanti . . . . .                           | 54 |
| 5.2  | Distribuzione dell'età dei partecipanti (in percentuale) . . . . .         | 54 |
| 5.3  | Impiego dei partecipanti . . . . .   | 54 |
| 5.4  | Ambito di studio/lavoro dei partecipanti . . . . .                         | 54 |
| 5.5  | Ore giornaliere trascorse utilizzando un dispositivo elettronico . . . . . | 54 |
| 5.6  | Numero di volte utilizzo di un visore di realtà virtuale . . . . .         | 54 |
| 5.7  | Risultati RP . . . . .   | 56 |
| 5.8  | Distribuzione embodiment 0ms . . . . .                                     | 60 |
| 5.9  | Andamento embodiment con delay . . . . .                                   | 62 |
| 5.10 | Domande embodiment con delay . . . . .                                     | 63 |
| 5.11 | Andamento dati oggettivi e soggettivi . . . . .                            | 64 |



# Capitolo 1

## Introduzione

Quando si pensa alla Realtà Virtuale (VR) è probabile che la si immagini come una tecnologia recente, sviluppata nel corso degli ultimi 10 anni. In realtà, il primo brevetto di un dispositivo non elettronico che creava un ambiente tri-dimensionale a partire da una immagine 2D venne depositato nel 1939 a New York con il nome di *View-Master*, inventato dal William Gruber e commercializzato dalla *Sawyer's Inc.* (Fig. 1.1). Considerando la tecnologia in termini più simili a come la conosciamo oggi, l'idea della VR nasce già alla fine degli anni 60. Il primo *Head Mounted Display*<sup>1</sup> stereoscopico venne creato nel 1968. Nei decenni successivi sono stati ideati e realizzati sistemi di tracking e dispositivi di input ad hoc, come il *Mattel PowerGlove* per il *Nintendo NES* presentato nell'ottobre del 1989 (Fig. 1.2).

L'entusiasmo per questa nuova tecnologia calò però negli anni 90, principalmente a causa della mancanza di interfacce di visualizzazione ad alta risoluzione ed elaboratori con una adeguata potenza di calcolo.

Si dovette aspettare fino al secondo decennio degli anni 2000 per vedere una rinascita della VR. Complice la tecnologia che ormai aveva raggiunto un livello tale da poter sopperire ai problemi citati precedentemente, a partire dal 2010 vengono presentati visori e periferiche di livello *consumer*. Nel 2012 viene annunciato l'Oculus Rift (diventato oggi il Meta Quest) e nel 2014 il Playstation VR, destinati entrambi ad un utilizzo prettamente videoludico.

---

<sup>1</sup>HMD: letteralmente *schermo montato sulla testa* è un dispositivo che si indossa sulla testa e che presenta due schermi posti davanti agli occhi



**Figura 1.1:** View-Master della Sawyer's Inc. insieme a dei dischetti



**Figura 1.2:** Mattel Power Glove per il Nintendo NES (1989)

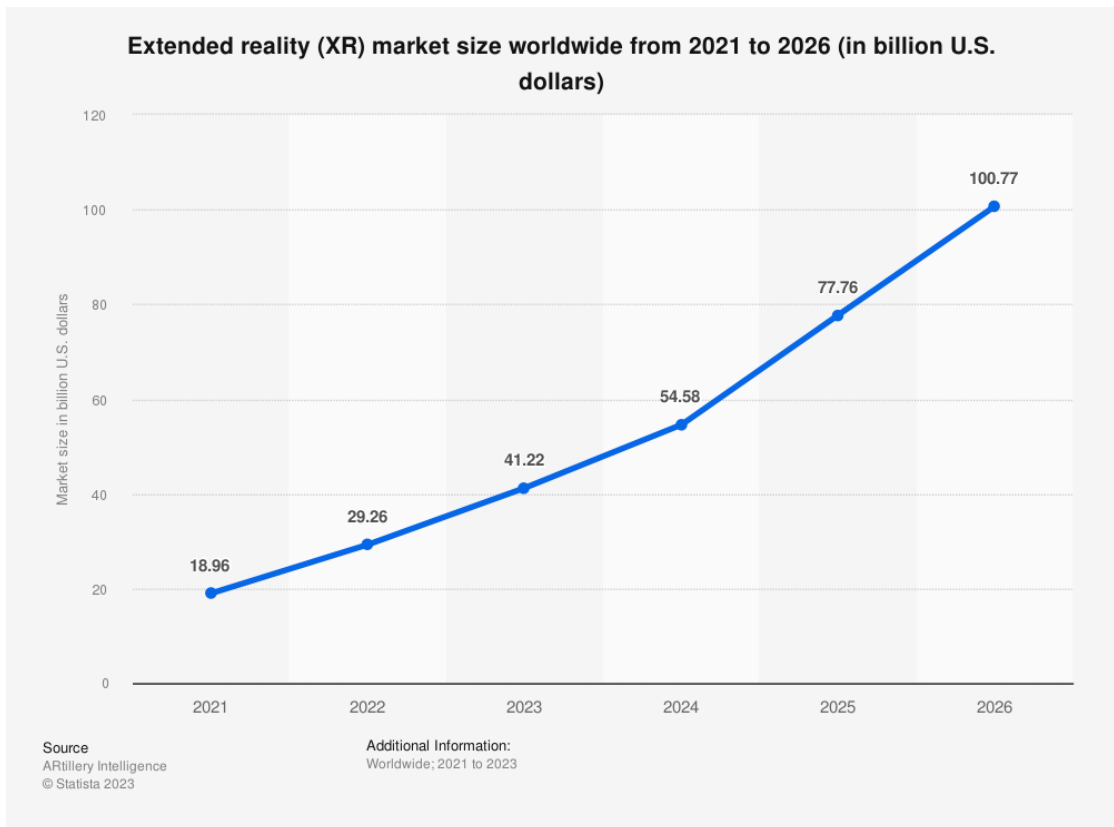
## 1.1 Il mercato del virtuale

Ad oggi sono molte le aziende che stanno investendo grandi capitali sulla realtà virtuale e la realtà aumentata (spesso indicate insieme con il termine *XR Reality*) e la motivazione è semplice: le previsioni di crescita del mercato del virtuale è più che promettente. Come mostrato nel grafico (Fig. 1.3), i ricavi derivanti dal mercato XR sono previsti in crescita anche nei prossimi anni, con la previsione che raggiungano i 100 miliardi di dollari U.S. entro il 2026 [1].

Il maggior interesse da parte di grandi aziende come *Meta*, *Microsoft*, *Sony* e, da Luglio 2023, *Apple* per il *metaverso*<sup>2</sup> ha portato allo sviluppo di HMD sempre più evoluti e alla ricerca di periferiche accessorie che permettano di aumentare il senso di presenza dell'utente all'interno di tali applicazioni. Prodotti tecnologicamente più avanzati con prezzi accessibili al grande pubblico ha fatto sì che le vendite lato *consumer* di visori VR (e AR) siano progressivamente aumentate, con la previsione

---

<sup>2</sup>Termine utilizzato per indicare uno o più ambienti virtuali all'interno dei quali gli utenti possono interagire tra di loro



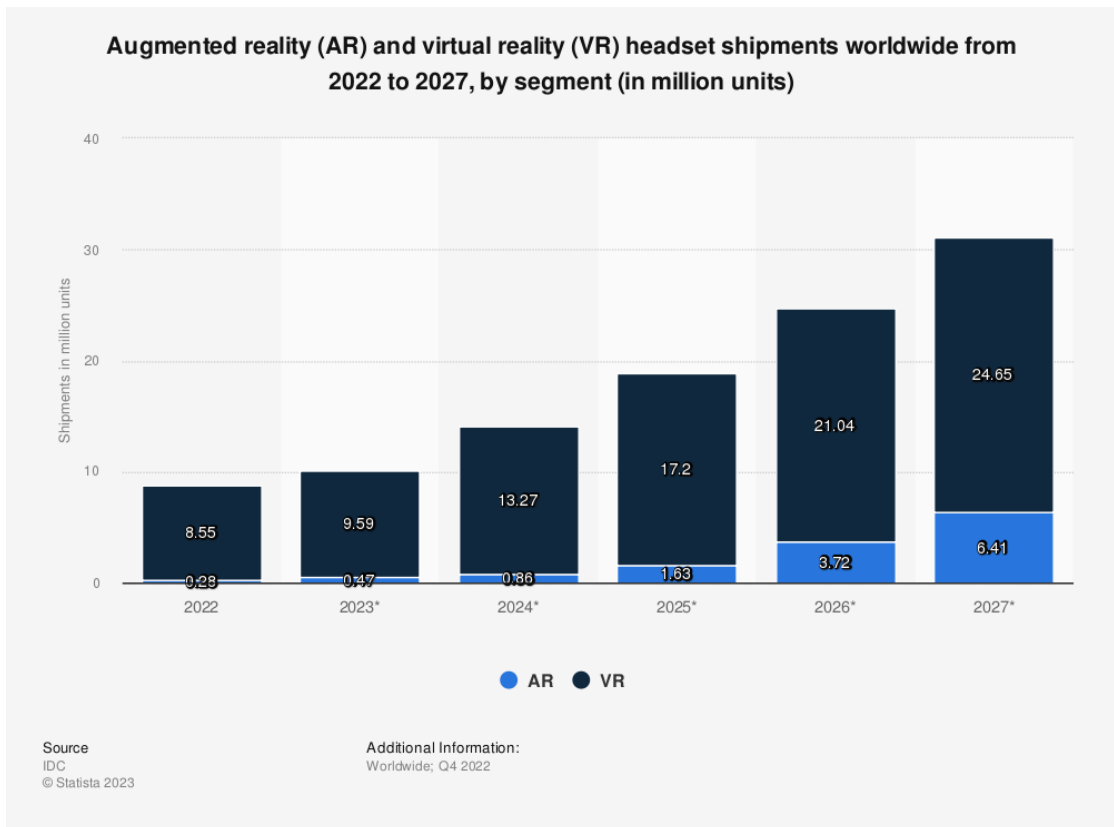
**Figura 1.3:** Ricavi mercato XR (consumer e enterprise)

[2] che la crescita continui anche nei prossimi anni (Fig. 1.4), ampliando sempre di più il bacino di utenza disponibile.

## 1.2 VR e Neuroscienze

Nell'ambito delle neuroscienze, sono molteplici i casi d'uso nei quali la VR ha portato ad ottenere risultati significativamente migliori rispetto all'utilizzo delle sole tecniche tradizionali. Secondo quanto riportato in *Virtual Reality in the Neurosciences: Current Practice and Future Directions* [3], la VR è attivamente studiata in neurochirurgia e neuroscienze e parte integrante dell'assistenza al paziente e della formazione chirurgica. Riporto di seguito alcuni casi d'uso reali.

**Pazienti con dolore acuto e cronico** La VR usata in concomitanza ad una terapia analgesica standard ha diminuito la valutazione soggettiva del dolore



**Figura 1.4:** Dispositivi AR e VR spediti a livello mondiale

peggiore e il tempo trascorso a pensare al dolore nei pazienti sottoposti a tale terapia. I pazienti che soffrivano di emicrania cronica immersi in una area di attesa virtuale costruita appositamente hanno riportato una diminuzione significativa della valutazione al dolore rispetto ad una area di attesa che riproduceva un ambiente ospedaliero.

**Trattamento patologie neuropsichiatriche** La VR viene usata per il trattamento di patologie come schizofrenia, disturbo da stress post-traumatico, disturbo ossessivo compulsivo, disturbo dello spettro autistico. Le metodologie

utilizzate sono tipicamente la desensitization<sup>3</sup> e il flooding<sup>4</sup>.

**Riabilitazione** Nella riabilitazione di malattie neurocognitive la VR viene utilizzata per la sua capacità di poter creare un ambiente immersivo e interattivo con feedback in tempo reale. Nel caso del trattamento della negligenza spaziale<sup>5</sup>, prima della VR venivano usati degli speciali occhiali con dei prismi per poter creare una discrepanza tra la posizione percepita e quella reale. Con l'avvento della VR questa tecnica viene replicata in digitale con il vantaggio di avere una maggiore efficacia e una maggiore libertà di modifica.

Come per tutti i contesti, scientifici e non, ci sono anche delle controindicazioni da tenere in considerazione quando si progettano degli esperimenti di questa tipologia. Una prima controindicazione è legata all'utilizzo del sistema dal punto di vista dei pazienti: le sessioni devono avere una durata limitata per evitare l'insorgere di problematiche come il *Motion Sickness*<sup>6</sup>. Spostandosi invece dal punto di vista di chi organizza la sperimentazione, come riportato in *Virtual Reality for Research in Social Neuroscience* [4], un punto da tenere in considerazione è legato alla progettazione. L'implementazione di uno studio che utilizza la VR richiede diverse figure specializzate in molteplici campi. Questi sono tipicamente la computer-grafica 3D, la programmazione, l'integrazione di sensori fisiologici specifici e l'analisi finale dei dati ottenuti dall'esperienza. Come conseguenza, il budget necessario a poter realizzare uno studio può salire molto rapidamente.

---

<sup>3</sup>Desensitization: i pazienti vengono esposti gradualmente a oggetti e scenari che provocano un senso di paura. L'obiettivo è quello di ottenere una tolleranza allo stimolo al punto di non provare più paura

<sup>4</sup>Flooding: il paziente viene esposto ad oggetti e/o scenari che provocano una risposta di paura ma in un ambiente controllato

<sup>5</sup>Negligenza spaziale: I pazienti sono inconsapevoli di eventi che hanno luogo nella parte dello spazio e del corpo controlaterale al lato della lesione (controlaterale)

<sup>6</sup>Il *Motion Sickness* si avverte quando il movimento che vediamo è diverso da quello avvertito dall'orecchio interno causando nausea, mal di testa e vomito



## 1.3 Sense of Agency

Il *sense of agency* (SoA) rappresenta l'esperienza di collegare le intenzioni risultati esterni, ovvero è la sensazione di avere il controllo sulle proprie azioni e le loro conseguenze. Il SoA viene acquisito già dall'infanzia, in alcuni casi già dai due mesi di vita, legato ad azioni come il sorriso o il succhiare il pollice della mano. Avanzando nell'età siamo in grado di espandere il SoA anche agli oggetti o eventi esterni. Un esempio di questo fenomeno si ha durante la guida di un autoveicolo: se giriamo il volante percepiamo che l'auto gira anche se la rotazione è controllata da elementi al di fuori del nostro controllo. Il SoA contiene quindi due livelli: il primo è legato all'esperienza di muovere il proprio corpo e ricevere un feedback da esso, il secondo è legato all'esperienza di controllare eventi esterni e di ricevere feedback dall'ambiente.

A prescindere dal livello che stiamo sperimentando, sono necessarie tre condizioni affinché si verifichi il SoA:

1. Si vuole produrre un risultato attraverso l'azione che stiamo per compiere
2. Il movimento è volontario
3. Il risultato atteso è ottenuto

Ottenere un risultato come conseguenza di una azione è cruciale nell'ambito delle *Human Computer Interface (HCI)*: se l'interazione con il sistema avvenuta tramite una tipologia di input genera il risultato atteso genera un SoA allora avremo la sensazione di star controllando il sistema.

Il SoA non trova un'utilità pratica solamente nel campo delle HCI. Riporto di seguito alcuni esempi tratti da *What Is the Sense of Agency and Why Does it Matter?* [5] per fornire al lettore altri campi di applicazione del SoA.

**Schizofrenia e altri disturbi** La schizofrenia è il classico disturbo del SoA e per questo è stata oggetto molti più studi rispetto ad altre tipologie di disturbi. Gli individui affetti da schizofrenia hanno problemi di elaborazione dell'agency, ha la sensazione che le sue azioni non siano sotto il suo controllo.

**Interazione uomo-computer (HCI)** Il già citato ambito delle HCI. Nell'interazione tra uomo e computer l'esperienza utente è al centro della progettazione

delle varie interfacce. Il SoA entra in gioco proprio in questo frangente. Secondo Ben Shneiderman (informatico americano che ha condotto importanti studi nel campo dell'interazione uomo-macchina) gli utenti desiderano fortemente la sensazione di essere al comando del sistema e che il sistema risponda alle loro azioni. Il design delle interfacce otterrà certamente dei benefici dalla ricerca sul SoA.

**Libertà e responsabilità** L'attribuzione della responsabilità è forse la funzione chiave del SoA. Se le persone possono essere ritenute responsabili di ciò che fanno, allora il comportamento può essere legittimato attraverso punizioni o premi. Nell'attribuire o meno la responsabilità il SoA trova un'importanza rilevante ed è probabile che la ricerca in questo ambito porti ad applicazioni pratiche nel campo giuridico.

## 1.4 Potenziale evento-correlato

Un potenziale evento-correlato (in inglese *event-related potential - ERP*) misura il cambiamento di tensione all'interno del cervello a seguito (o in alcuni casi anche in precedenza) di specifici stimoli sensoriali, come visivi, uditivi o spunti di preparazione all'attività motoria.

Questo tipo di potenziale è possibile ricavarlo attraverso una misura elettroencefalografica (*EEG*) dell'attività neurale, dalla quale vengono estratti i segmenti temporali di interesse. L'EEG, per come funziona, registra la generale attività elettrica del cervello, generata da migliaia di processi simultanei. Per questo motivo, per ottenere una forma d'onda ERP visibile, occorre effettuare più prove sperimentali delle quali effettuare una media dei risultati. Mediando i valori letti tramite EEG è possibile ridurre l'attività elettrica casuale ottenendo la forma d'onda rilevante, ovvero l'ERP di interesse.

Esistono molteplici componenti dell'ERP, ciascuna associabile a stimolazioni differenti. Ad esempio il *Visual Evoked Potential* è un potenziale che rappresenta la risposta celebrale (uguale per ogni soggetto) ad uno stimolo visivo. Nel caso specifico della sperimentazione per la quale questo progetto è stato sviluppato, lo stimolo ERP di interesse è quello generato in concomitanza con la preparazione di un movimento motorio, chiamato *Readiness Potential* (fig. 1.5).

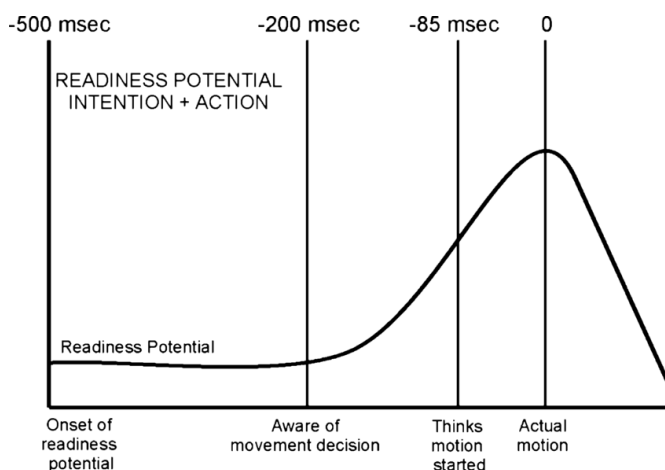


Figura 1.5: Onda RP

### 1.4.1 Readiness Potential

Il *Readiness Potential (RP)* (traducibile come potenziale di prontezza) è un segnale cerebrale legato al movimento volontario. L'RP è visibile attraverso l'analisi dell'EEG effettuato durante lo svolgimento di compiti che comportano dei movimenti spontanei o auto-impostati. Allineando i dati mediati ottenuti tramite EEG con l'inizio del movimento è possibile visualizzare un potenziale elettrico lento e negativo che precede l'inizio del movimento.

L'RP risulta complicato da distinguere a livello sperimentale rispetto ad altri potenziali a onde lente, la sostanziale differenza è rappresentata dal contesto sperimentale in cui vengono generati. Secondo quanto riportato nell'articolo *What Is the Readiness Potential?*[6], alcuni studi hanno riportato un RP in corrispondenza di compiti nei quali non era presente alcuna attività motoria, mettendo di fatto in discussione la relazione tra RP e attività motoria. Un ulteriore fattore negativo è dato dal fatto che il segnale RP è un ordine di grandezza più debole del rumore di lettura EEG, il che rende necessario effettuare una media di più sessioni sperimentali per poter rivelare l'RP. Questo porta a domandarsi se l'RP mediato rappresenti una reale media di valori presenti in ogni singola prova o se sia un artefatto generato dalla media stessa. Un ulteriore fattore da tenere in considerazione è che in linea teorica potrebbero essere presenti altre forme d'onda che, opportunamente distribuite, potrebbero generare un tracciato mediato identico a quello del RP.

Le considerazioni effettuate poc'anzi hanno lo scopo di far capire al lettore quanto il campo del *readiness potential* sia ancora per la maggior parte inesplorato e di quante domande non si conoscano ancora delle risposte definitive.

# Capitolo 2

## Stato dell'arte

### 2.1 Sperimentazioni attuali

La VR immersiva è diventato uno strumento che ha preso sempre più piede non solo, come illustrato nell'introduzione di questo elaborato, nel campo della riabilitazione o della terapia, ma anche come strumento attivo nelle sperimentazioni. Secondo quanto riportato in *Neural Applications Using Immersive Virtual Reality: A Review on EEG Studies* [7], l'uso sempre maggiore della VR è dovuto alla grande capacità di poter creare ambienti complessi e immersivi, con un alto grado di libertà ma mantenendo costi minimi. Sono diversi gli studi che dimostrano che l'utilizzo della VR porta a risultati più efficaci rispetto ai tradizionali schermi 2D: ne è un esempio lo studio condotto da Tian et al., nel quale è stato utilizzato un ambiente immersivo VR insieme all'EEG per confrontare l'evocazione di emozioni in ambienti 2D e 3D. Si è osservato che l'eccitazione emotiva è risultata maggiore quando fornita nell'ambiente 3D.

L'analisi EEG in combinazione con visori VR, oltre che per sperimentazioni prettamente legate all'ambito medico-scientifico, possono trovare anche applicazione nel campo ingegneristico. Ne è un esempio lo studio *Predicting head rotation using EEG to enhance streaming of images to a Virtual Reality headset* [8], all'interno del quale sono state effettuate sperimentazioni per ottimizzare lo streaming su internet di contenuti multimediali.

### 2.1.1 Sperimentazioni sul sense of agency

Nella letteratura accademica sono presenti diversi studi effettuati sul SoA con l'impiego della VR. Lo studio *Sensorimotor experience in virtual reality enhances sense of agency associated with an avatar* [9] si poneva come obiettivo quello di verificare se il movimento effettuato all'interno di un ambiente virtuale immersivo possa modificare il SoA. In particolare, lo studio si è concentrato sul SoA generato attraverso un avatar virtuale. Si è osservato che il livello di SoA suscitato dall'avatar è paragonabile a quello delle azioni umane reali al di fuori della VR. Un altro esempio è lo studio *Movement Augmentation in Virtual Reality: Impact on Sense of Agency Measured by Subjective Responses and Electroencephalography*[10] nel quale ci si è concentrati sull'analisi del SoA in un ambiente virtuale immersivo nei casi in cui è utilizzato il *movimento aumentato*<sup>1</sup>. Siccome nell'aumento del movimento aumentato c'è una discrepanza tra il movimento reale e quello virtuale, lo studio si è concentrato sul capire come il SoA varia e come possa essere aumentato per migliorare le applicazioni VR di riabilitazione. In questa sperimentazione il SoA è stato valutato sia tramite questionari soggettivi che con una analisi EEG. I risultati ottenuti mostrano ad un certo livello di movimento aumentato può indurre ad un SoA leggermente più elevato.

In conclusione, lo studio del SoA è un argomento molto attivo in neuroscienze proprio per le sue possibili applicazioni in diversi ambiti.

## 2.2 Il progetto

Il presente progetto di tesi effettuato in collaborazione con il *ContExp Lab* dell'Università di Torino, si prefigge come obiettivo quello di analizzare la relazione che sussiste tra il *sense of agency* e il *readiness potential* all'interno di un ambiente virtuale immersivo. Si è scelto di utilizzare la VR per poter avere una libertà di sperimentazione estremamente maggiore. Il mondo virtuale si può plasmare a proprio piacimento: in particolare andando ad aggiungere del delay o modificando la risposta dell'avatar virtuale rispetto ai reali movimenti dell'utente.

---

<sup>1</sup>L'aumento del movimento è utilizzato in applicazioni VR progettate per scopi specifici, come riabilitazione o esercizio fisico

Un secondo obiettivo è quello di analizzare come la sensazione di embodiment<sup>2</sup> provata dall'utente possa variare in concomitanza alla variazione di delay sperimentato all'interno dell'ambiente virtuale. Sarebbe interessante potersi basare sul *readiness potential* per ottenere una valutazione oggettiva dell'*embodiment*. Questa metodologia permetterebbe di distaccarsi dai questionari soggettivi sottoposti agli utenti dopo una sperimentazione VR, ottenendo un dato oggettivo e facilmente confrontabile con altre sperimentazioni.

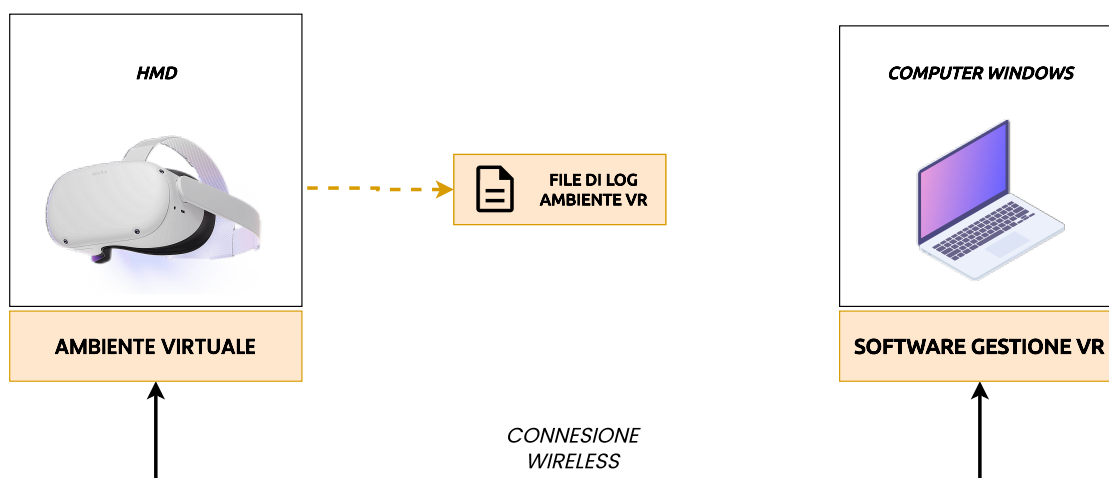
---

<sup>2</sup>L'*embodiment* si può tradurre in italiano come *incarnazione* e in questo ambito sta ad indicare la sensazione per l'utente di controllare effettivamente il corpo virtuale, di entrare a far parte ed immedesimarsi all'interno del mondo virtuale.

## Capitolo 3

# Implementazione

Il sistema sviluppato si compone di due applicativi che lavorano su due dispositivi differenti: l'**ambiente virtuale** e il **software di gestione VR**. Come visibile dallo schema (Figura 3.1) le due parti comunicano tra di loro utilizzando un canale di comunicazione wireless. Inoltre è presente anche un terzo componente, il **sistema di LOG**, gestito dall'ambiente virtuale.



**Figura 3.1:** Schema interazione componenti

Riassumendo, le componenti macroscopiche che compongono il sistema e che andremo ad analizzare nelle pagine seguenti sono:

- L' Ambiente Virtuale



- Il Software Gestione VR
- Il Sistema di LOG

## 3.1 Ambiente Virtuale

L'ambiente virtuale è l'applicativo di Virtual Reality che viene eseguito sul visore indossato dal soggetto. Ha come compiti la gestione della sessione di test e la raccolta dei dati all'interno sul mondo virtuale.

### 3.1.1 Scelta dell' HMD

L'HMD utilizzato è il Meta Quest 2 (Figura 3.2) ed è di tipo unthetered, ovvero è un dispositivo che non necessita di una connessione cablata con un PC perchè possiede la potenza computazionale necessaria per poter eseguire applicazioni VR in maniera autonoma. I motivi dietro a questa scelta sono principalmente due e solo legati all'ambiente reale nel quale vengono eseguite le sessioni sperimentali e alle prestazioni dei dispositivi in gioco.



**Figura 3.2:** Meta Quest 2

Avere un dispositivo unthetered permette di avere una maggiore libertà di gestione degli spazi. Anche se il soggetto rimane seduto per l'intera durata della sessione sperimentale bisogna considerare il fatto che il PC che si occupa della lettura dei dati EEG e della gestione dell'ambiente VR potrebbe non essere posizionato

vicino al soggetto. Non essendo vincolati ad un cavo di collegamento tra PC e HMD gli operatori possono posizionare più agevolmente i vari dispositivi nell'ambiente.

Il secondo motivo è legato alle prestazioni computazionali dei dispositivi in gioco. I PC utilizzati possiedono GPU inadatte all'esecuzione di applicazioni VR e quindi sarebbe stato necessario reperire dispositivi più performanti per eseguire l'ambiente virtuale. Usare un dispositivo standalone permette di essere indipendenti dagli altri dispositivi elettronici utilizzati.

Lo svantaggio principale rispetto ad un visore tethered è quello di raggiungere un livello di fotorealismo minore in quanto la potenza di calcolo del visore è minore rispetto a quella di un computer di fascia alta. Siccome però il fotorealismo non era una delle richieste vincolanti del progetto e gli svantaggi erano maggiori dei vantaggi si è confermata la scelta di un visore untethered.

### 3.1.2 Scelta del motore grafico

Un motore grafico permette di sviluppare videogiochi e altre applicazioni interattive come la visualizzazione di progetti architettonici, simulatori virtuali e animazioni 3D. Esistono diverse soluzioni di motori grafici, adatti a produrre svariate tipologie di contenuti interattivi. Nell'ambito dello sviluppo di applicazioni VR e più in generale di tutte le applicazioni interattive 3D i due maggiori competitor sono Unity e Unreal Engine.

L'Unreal Engine (Figura 3.3) è stato fondato nel 1998 dalla EpicGames ed è rinomato per la sua capacità di ottenere una resa grafica molto realistica.

Unity (Figura 3.4) è stato pubblicato nel 2005 ed è famoso per la sua maggiore facilità di utilizzo rispetto ad UnrealEngine per i neofiti, alla capacità di poter creare anche progetti 2D e la possibilità di creare progetti su un'ampia gamma di sistemi operativi.

Ambedue i motori sono valide alternative per lo sviluppo di progetti VR, permettono di raggiungere alti standard di qualità e forniscono gli elementi di base per poter interfacciarsi con gli HMD e poter iniziare subito a sviluppare il proprio progetto.

Un aspetto che non è stato sottovalutato nella decisione finale è stato quello relativo alla community di sviluppatori che lavora con il motore grafico. La presenza di una vasta community permette di avere a disposizione molti strumenti



**Figura 3.3:** Logo Unreal Engine



**Figura 3.4:** Logo Unity

di apprendimento facilmente reperibili con il vantaggio di facilitare e velocizzare lo sviluppo. La scelta finale è ricaduta su Unity.

### **3.1.3 Struttura ambiente virtuale**

La struttura dell'ambiente virtuale è in verità molto semplice. L'ambiente è costituito da una stanza all'interno della quale sono presenti un tavolo e un pulsante, oltre ad altri elementi decorativi volti ad aumentare l'immersione del soggetto (Figura 3.5). Tramite il software di gestione è possibile configurare la stanza virtuale in modo che rispecchi più fedelmente possibile gli elementi della stanza reale all'interno della quale sono svolte le sperimentazioni. È infatti possibile impostare la dimensione del tavolo e la posizione del pulsante per farla combaciare con la posizione del pulsante reale. Nel momento in cui la sessione sperimentale inizia, il soggetto vedrà comparire sul tavolo una scritta che sta ad indicare che può iniziare a compiere le azioni pre definite.

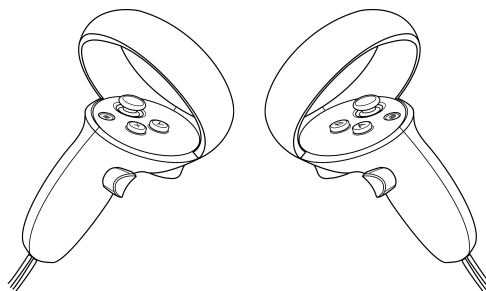
### **3.1.4 Tracking delle mani**

Per l'ottenimento della posizione delle mani l'HMD mette a disposizione due possibili strade: l'utilizzo dei controller del HMD o il tracking diretto delle mani.



**Figura 3.5:** Ambiente virtuale

I **controller** (Figura 3.6) sono dei dispositivi che contengono al loro interno della sensoristica che comunica direttamente con l'HMD e che permette all'HMD di tracciarli correttamente. Visto che i controller sono indossati dal soggetto, indirettamente forniscono la posizione delle mani. Questo sistema è molto preciso ma rende meno agevole l'esecuzione della sessione da parte del soggetto, in quanto c'è l'ingombro dei controller nel momento in cui si va a premere il pulsante.



**Figura 3.6:** Controller Meta Quest 2

Il **tracking diretto** invece permette di aumentare il senso di presenza del

soggetto, inquanto si avrà la sensazione di vedere le proprie mani davanti a se. Inoltre, permette di premere il pulsante più agevolmente dato che non c'è nessun oggetto aggiuntivo da gestire. Nel caso del Meta Quest, secondo quanto riportato sulla documentazione ufficiale di *Meta* [11], il tracking delle mani funziona tramite l'uso delle fotocamere *inside-out* dei visori *Meta Quest*. Il visore rileva la posizione e l'orientamento delle mani e la disposizione delle dita. Una volta effettuato il rilevamento, gli algoritmi di visione artificiale consentono di monitorare il movimento e l'orientamento delle mani.

Dato che il progetto richiede una immedesimazione da parte del soggetto maggiore possibile e dati i movimenti che il soggetto deve compiere, si è optato per l'utilizzo del tracking diretto delle mani. Unity mette a disposizione dei programmatori un pacchetto software, chiamato *XRHands* [12], che si occupa di interfacciarsi con il sistema di tracciamento dell'HMD, fornendo gli strumenti necessari al suo utilizzo.

### 3.1.5 Server e scambio dati

L'applicazione riceve le informazioni necessarie dal software di gestione. Il paradigma di comunicazione adottato è di tipo client-server con protocollo *TCP/IP*, dove l'applicazione VR ricopre il ruolo di server mentre il software di gestione ricopre il ruolo di client.

All'avvio dell'applicazione, il componente responsabile della gestione delle richieste di rete, si mette in ascolto sull'indirizzo IP del HMD e sulla porta *8080*. I messaggi ricevuti dal server devono essere strutturati in modo da contenere il comando che rappresenta l'azione da eseguire e i dati necessari in formato *JSON*. Nell'esempio seguente, il client richiede al server di modificare la posizione del pulsante, spostandolo di 50cm in orizzontale e di 80cm in verticale.

```
1 BUTTON_POSITION##{"horizontal":50,"vertical":80}
```

Nel messaggio ricevuto è possibile notare il comando **BUTTON\_POSITION** e i dati necessari alla corretta esecuzione della richiesta, separati dal blocco `##`. In questo esempio il server modificherà la posizione del pulsante all'interno del mondo virtuale e successivamente ritornerà al client un *JSON* contenente un booleano, per indicare il successo o meno della richiesta, e un messaggio opzionale (vedi 3.1).

**Listing 3.1:** Struttura risposta

```
1 struct ResponseData
2 {
3     public bool result;
4     public string message;
5 }
6
7 //Esempio di risposta
8 {
9     "result": true,
10    "message": "Button position updated successfully"
11 }
```

Nel caso di una richiesta contenente dati aggiuntivi la struttura della risposta varia leggermente (vedi 3.2). Viene aggiunto una campo data di tipo array. Un esempio di richiesta che produce una risposta di questo è la **GET\_SESSIONS\_LIST**, che ritorna un array di sessioni.

**Listing 3.2:** Struttura risposta con dati aggiuntivi

```
1 struct ResponseDataWithLPayload<T>
2 {
3     public bool result;
4     public string message;
5     public T[] data;
6 }
7
8 //Esempio di risposta
9 {
10    "result": true,
11    "message": "Sessions list retrieved",
12    "data":[
13        {
14            "Name": "Nome sessione",
15            "Description": "Descrizione sessione",
16            "Identifier": "Identificativo sessione",
17            "CreatedAt": "Data creazione sessione"
18        },
19        ...
20    ]
```

21 }

Nella Tabella 3.1 vengono illustrate tutte le possibili azioni che il server è in grado di gestire.

| CODICE AZIONE        | DATI RICHIESTI                                  | DESCRIZIONE   |
|----------------------|---|---|
| TEST_CONNECTION      | -   | Verifica se il server è operativo   |
| START_SESSION        | Dati sessione                                   | Inizia una nuova sessione con i parametri specificati nella richiesta                             |
| FORCE_END_SESSION    | -   | Forza la terminazione della sessione attiva, anche se non sono stati eseguiti tutti i blocchi     |
| GET_SESSIONS_LIST    | -   | Ritorna la lista delle sessioni presenti sulla memoria interna del visore                         |
| BUTTON_POSITION      | Spostamento orizzontale<br>Spostamento vertical | Modifica la posizione del pulsante sul tavolo   |
| MOVE_BUTTON          | Direzione                                       | Trasla il pulsante nelle direzioni destra (+X), sinistra (-X), avanti (+Z) o indietro (-Z)        |
| TABLE_DIMENSIONS     | Altezza<br>Larghezza<br>Profondità              | Specifica le dimensioni del tavolo in modo da farlo combaciare con quello reale                   |
| TABLE_POSITION       | Direzione spostamento                           | Trasla il tavolo avanti (+Z) o indietro (-Z)  |
| CAMERA_USER_POSITION | Direzione spostamento                           | Trasla la camera virtuale del soggetto in alto (+Y) o in basso (-Y)                               |
| DELETE_SESSION       | Identificativo sessione                         | Elimina dalla memoria del visore la cartella contenente i dati relativi alla sessione specificata |
| DOWNLOAD_SESSION     | Identificativo sessione                         | Comprime la cartella della sessione specificata in un file ZIP e la ritorna al client             |

**Tabella 3.1:** Azioni gestite dal server

Il server si occupa anche della creazione dei file contenenti i dati necessari per l'analisi della sessione che vengono salvati nella memoria interna dell'HMD.

L'argomento verrà trattato più nel dettaglio nella sezione 3.2.

### 3.1.6 Delay

L'introduzione di un *delay* è uno dei requisiti fondamentali del progetto. Per comprendere appieno come viene applicato bisogna prima capire come sono applicate le traslazioni agli oggetti che rappresentano le mani del soggetto.

L'HMD, tramite le camere esterne, effettua il tracciamento della posizione e rotazione delle mani reali (che chiameremo **RealHand**) del soggetto. Tramite la libreria *XRHands* già citata precedentemente, *Unity* recupera questi dati di *tracking* calcolati dal HMD e li applica ai modelli 3D che rappresentano la mani del soggetto nel mondo virtuale (che chiameremo **VirtualHand**). Questo è il flusso standard di una applicazione VR che sfrutta il tracking delle mani.

In questo progetto è richiesta l'introduzione di un *delay* tra i valori letti dal *tracking* e quelli applicati alla **VirtualHand**. Questo significa che se l'operatore specifica un ritardo di 200ms, la **VirtualHand** deve traslare e ruotare come la **RealHand** ma con un ritardo di 200ms.

A questo punto è necessario fare una considerazione: il *delay* specificato dall'operatore va inteso come una indicazione sull'ordine di grandezza desiderato e non un valore che deve essere rispettato con la massima precisione. Questo perchè esiste un *delay* intrinseco introdotto dal visore stesso durante la fase di *tracking*. Secondo un caso studio effettuato sul *Meta Quest 2* per valutare l'accuratezza dei sistemi di *hand-tracking*, questo delay intrinseco ha un valore che in media è pari a circa 45 millisecondi[13].

#### Buffer

La soluzione adottata per la gestione del delay prevede l'utilizzo di un buffer circolare, nel quale vengono salvati i valori più recenti di posizione e rotazione letti nel *frame* corrente e ritornati i valori con il *delay*. Siccome ogni cella rappresenta le informazioni relative ad un *frame*, il valore del *delay* (che viene definito dall'operatore in millisecondi) deve essere convertito in un valore che indica il numero di *frame* necessari a raggiungere quel ritardo.



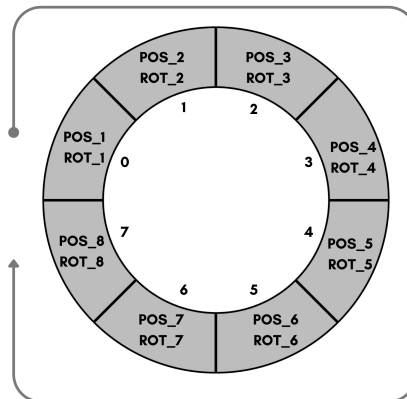
Osservando li *framerate* del dispositivo durante l'esecuzione dell'applicazione si è potuto constatare che il *framerate* è stabile con valori che oscillano tra i 70 e i 72 *fps*. Per la conversione del *delay* si è preso come valore di riferimento 71 *fps*. Essendo 71 il numero di frame mediamente renderizzati nell'arco di 1 secondo, la formula utilizzata per la conversione è:

$$DELAY_{frame} = (DELAY_{ms}/1000) \times 71$$

Il valore ottenuto è un numero decimale ma che occorre arrotondare ad un valore intero in quanto rappresenta la dimensione di un buffer. Questa conversione introduce uno scostamento dal valore di delay originariamente specificato ma che, alla luce delle considerazioni effettuate precedentemente, risulta essere trascurabile ai fini della sperimentazione.

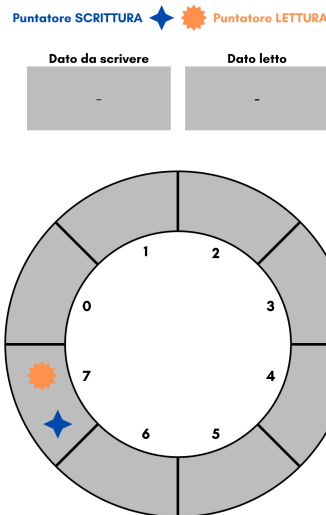
Supponiamo ora, per semplicità, che il valore del *delay* già convertito abbia un valore di 8 *frame*: di conseguenza il buffer circolare avrà una dimensione pari a 8 celle. Di seguito sono illustrati i diversi passaggi necessari alla corretta gestione del buffer.

**Fase preliminare** Come prima cosa il buffer viene riempito leggendo i valori di posizione e rotazione dai primi 8 *frame* renderizzati. Una volta completata questa operazione preliminare avremo all'interno della coda (cella 7) il valore più recente, ovvero il valore del *frame* numero 8, l'ultimo renderizzato. Nella testa (cella 0) avremo invece il valore più vecchio, cioè il primo *frame* renderizzato a partire dal momento in cui il delay viene attivato (*frame* numero 1).



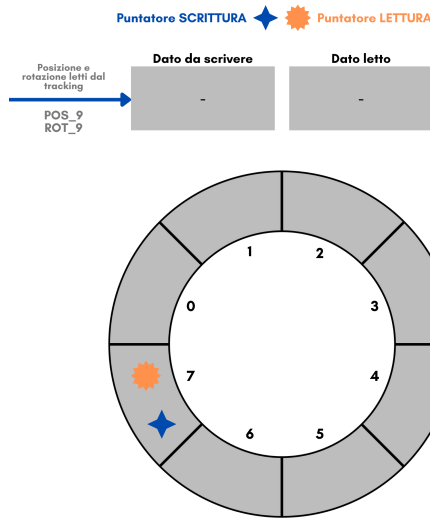
**Figura 3.7:** Buffer - Fase preliminare

**Passo 1** I valori di posizione e rotazione letti dal visore vengono ricevuti dalla funzione che gestisce il buffer. In questo esempio supponiamo di ricevere i dati relativi al frame numero 9.



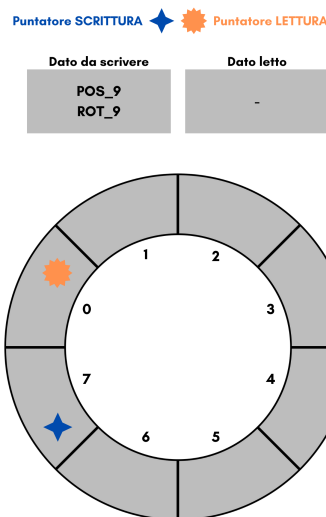
**Figura 3.8:** Buffer - Passo 1

**Passo 2** Il puntatore di lettura del buffer viene spostato alla cella successiva



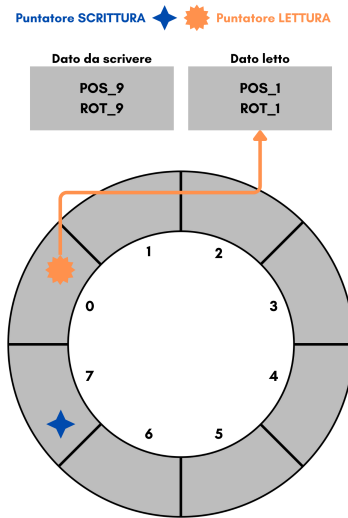
**Figura 3.9:** Buffer - Passo 2

**Passo 3** Viene letto il valore della cella e salvato in una variabile temporanea. In questo esempio i valori letti rappresentano i valori di posizione e rotazione relativi al primo *frame* letto (indicato in questo esempio come *frame* 1).



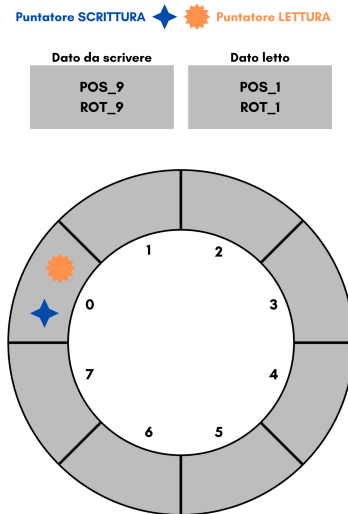
**Figura 3.10:** Buffer - Passo 3

**Passo 4** Il puntatore di scrittura viene posizionato nella cella successiva (cella 0)



**Figura 3.11:** Buffer - Passo 4

**Passo 5** I valori ricevuti nel passo 1 vengono scritti all'interno della cella. In questo esempio rappresentano i valori del *frame* numero 9.



**Figura 3.12:** Buffer - Passo 5

**Passo 6** I valori letti del *frame* 1 vengono ritornati dal buffer.

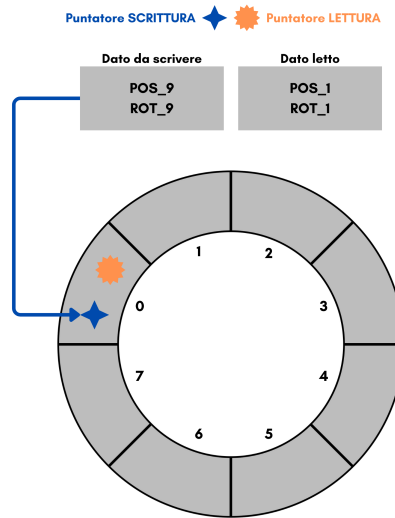


Figura 3.13: Buffer - Passo 6

**Stato finale** Al termine del *frame* numero 9 il buffer si troverà nella situazione indicata dalla figura seguente: il valore più recente è stato inserito nel buffer in sostituzione del valore più vecchio.

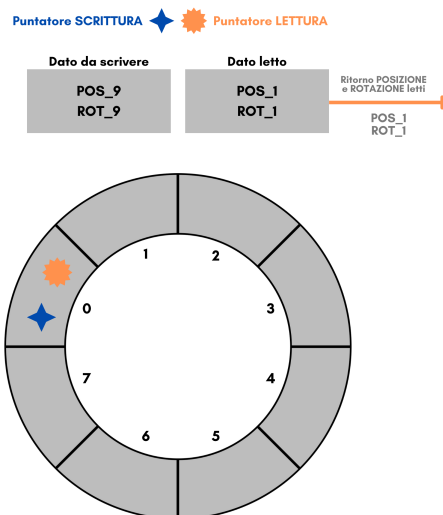


Figura 3.14: Buffer - Stato finale

Scendendo più nel dettaglio, il buffer è gestito tramite uno *ScriptableObject*

di nome *InputBufferSO*. Uno *ScriptableObject* [14] è un contenitore di dati che si può usare per salvare grandi quantità di dati, indipendentemente dalle istanze delle classi, evitando copie di valori e ottenendo una sola istanza di tali valori. A partire da questa classe è possibile creare un buffer per ogni oggetto al quale si desidera applicare un delay sulla rotazione e posizione. Nel progetto sono stati creati due buffer: uno per la mano sinistra e uno per la mano destra. Dall'oggetto al quale è associato il buffer sarà solamente necessario richiamare il metodo ***ReadThenAddValue*** il quale eseguirà tutti i passaggi mostrati precedentemente.

**Listing 3.3:** Codice buffer

```
1 public struct InputData
2 {
3     public Vector3 position;
4     public Quaternion rotation;
5     public bool isEmpty;
6 }
7
8 private List<InputData> _buffer;
9
10 // ...
11
12 public InputData ReadThenAddValue(InputData input)
13 {
14
15     // Se il delay è impostato a 0 viene ritornato il valore
16     appena letto
17     if(delay == 0)
18     {
19         return input;
20     }
21
22     // Se il delay varia e la dimensione del buffer risulta
23     // essere maggiore del delay, il buffer viene resettato
24     if (_buffer.Count > delay)
25     {
26         Clear();
27         return input;
28     }
29 }
```

```
28
29     // Se la dimensione del buffer è minore del delay
30     // si continua con la fare di riempimento preliminare
31     if (_buffer.Count < delay)
32     {
33         _buffer.Add(input);
34         return _emptyValue;
35     }
36
37     //Posizionamento del puntatore di lettura alla cella
38     //successiva
39     _readCounter++;
40     if (_readCounter >= delay)
41         _readCounter = 0;
42
43     //Lettura della cella
44     InputData readedData = _buffer[_readCounter];
45
46     //Posizionamento del putatore di scrittura alla cella
47     //successiva
48     _writeCounter++;
49     if (_writeCounter >= delay)
50         _writeCounter = 0;
51
52     //Scrittura del dato
53     _buffer[_writeCounter] = input;
54
55     //Il valore letto viene ritornato
56     return readedData;
57 }
58 }
```

Il codice mostrato in 3.3 rappresenta solo una parte della classe *InputBufferSO*, all'interno della quale sono definite altre proprietà e metodi necessari al corretto funzionamento del sistema. Tutto il codice è disponibile e liberamente consultabile sulla piattaforma GitHub[15].

## Applicazione del delay

Una volta gestito il buffer, è stato necessario capire come applicare effettivamente i dati letti dal buffer alle mani virtuali. L'idea alla base è stata quella di avere nell'ambiente due oggetti **VirtualHand**, dei quali solamente uno visibile al soggetto. Una **VirtualHand** è quella gestita dal pacchetto *XRHands* di *Unity* e quindi avrà posizione e rotazione sempre concordi a quanto fornito dal sistema di tracciamento. L'altra **VirtualHand** invece è quella alla quale vengono applicate le posizioni e rotazioni lette dal buffer. Ricapitolando, una **VirtualHand** rappresenta i dati di tracciamento e fornisce i dati di posizione e rotazione al buffer mentre l'altra **VirtualHand** rappresenta posizione e rotazione letti dal buffer.

Le immagini seguenti mettono in evidenza i comportamenti dei due oggetti **VirtualHand** presenti all'interno della scena, ipotizzando per semplicità un *delay* impostato pari a 2 *frame*. La **VirtualHand** di colore **rosa** (Fig. 3.16), che rappresenta i dati del buffer, segue con un ritardo di 2 *frame* le posizioni della **VirtualHand** di colore **grigio** (Fig. 3.15). La prima riceve i dati dal buffer del *delay*, la seconda riceve i dati dal sistema di tracciamento e li fornisce al buffer.

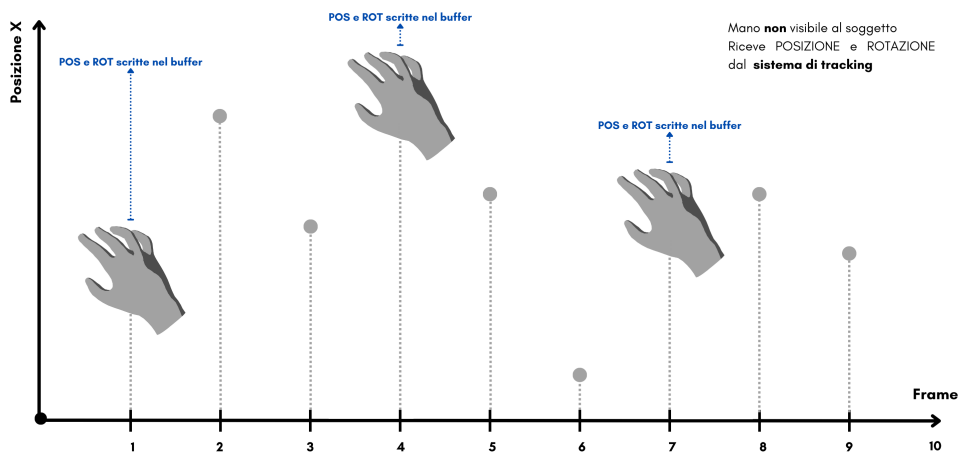


Figura 3.15: VirtualHand sistema di tracking



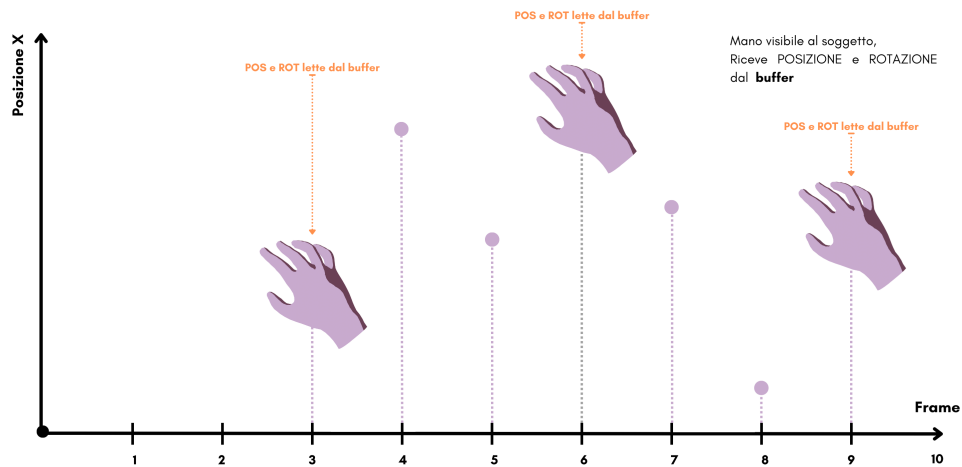


Figura 3.16: VirtualHand buffer

Questo sistema permette di essere indipendenti dalla versione o dal pacchetto che gestisce il tracciamento, in quanto posizione e rotazione vengono letti dopo che sono state applicate agli oggetti gestiti dal sistema di tracciamento. Se in un futuro venisse aggiornata o cambiasse la libreria utilizzata questa soluzione non ne verrebbe influenzata.

### 3.1.7 Specchiamento movimento locale

Con specchiamento movimento locale si intende l'inversione delle traslazioni della **VirtualHand** rispetto alla traslazione della **RealHand**. Se ad esempio la **RealHand** trasla verso destra, la **VirtualHand** deve traslare verso sinistra.

Un primo approccio è stato quello di specchiare semplicemente la posizione della **VirtualHand** nell'ambiente virtuale, ovvero se la **RealHand** si trova in posizione  $(10, 4, 0)$  la **VirtualHand** si troverà in posizione  $(-10, -4, 0)$ . Il problema di questa soluzione è che dipende dall'origine del mondo che, nel caso di questo progetto, è impostata al livello del pavimento. Questo comporta che nel momento in cui si specchia la *coordinata Y*, la **VirtualHand** risulta posizionata al di sotto del pavimento.

Per ovviare a questo problema si è dovuto cambiare approccio. Anzichè specchiare la posizione globale nell'oggetto nel mondo andando a invertire i valori delle coordinate spaziali, viene calcolato un vettore spostamento della **RealHand**. Il

vettore spostamento è invertito e applicato alla **VirtualHand**. In questo modo si è indipendenti dall'origine delle coordinate.

Per calcolare un vettore spostamento è però necessario avere due punti nello spazio: il punto di partenza e quello di arrivo. E' stata creata una routine che a intervalli regolari di tempo (es. 100ms) legge i dati all'interno del buffer ottenendo il valore della posizione più recente e salvando il valore letto per il prossimo ciclo di lettura. Leggendo i dati all'interno del buffer teniamo in considerazione anche un eventuale delay applicato all'ambiente virtuale. Una volta letto il valore della posizione più recente (*posizioneAttuale*) viene calcolato il vettore spostamento utilizzando il valore appena letto e l'ultimo valore letto (salvato dal ciclo di lettura precedente, *posizionePrecedente*).

$$\text{vettoreSpostamento} = \text{posizioneAttuale} - \text{posizionePrecedente}$$

Ottenuto il vettore *spostamento*, la **VirtualHand** può essere tralata lungo quella direzione. La traslazione deve però avvenire alla stessa velocità con la quale il soggetto ha traslato la **RealHand**. Per questo motivo nella traslazione si calcola la velocità come spazio percorso fratto il tempo impiegato. Lo spazio è rappresentato dalla *magnitudine* del vettore spostamento e il tempo è il tempo trascorso tra un campionamento e quello successivo.

Nello spezzone di codice seguente 3.4 è possibile vedere in maniera leggermente semplificata, come la logica spiegata viene applicata nel mondo virtuale.

**Listing 3.4:** Calcolo vettore spostamento

```

1 public struct InputData
2 {
3     public Vector3 position;
4     public Quaternion rotation;
5     public bool isEmpty;
6 }
7
8 // .....
9
10 private void ApplyTranslation()
11 {
12     transform.Translate(

```

```
13     bufferData.LastMovement.normalized * (
14         Time.deltaTime *
15         (bufferData.LastMovement.magnitude /
16         PositionSampleSeconds)
17     ), Space.World);
18 }
19 // .....
20
21 // Viene letto dal buffer il valore più recente ,
22 // calcola il vettore spostamento e mantiene il valore
23 // per utilizzarlo nel calcolo nella prossima lettura
24 IEnumerator DiscreteBufferManager()
25 {
26     InputData newPosition;
27     InputData oldPosition = new()
28     {
29         isEmpty = true ,
30     };
31
32     while (true)
33     {
34         newPosition = new()
35         {
36             isEmpty = false ,
37             position = new Vector3(
38                 bufferData.position.x,
39                 bufferData.position.y,
40                 bufferData.position.z) ,
41             rotation = bufferData.rotation
42         };
43
44         if (!oldPosition.isEmpty)
45         {
46             bufferData.LastMovement = newPosition.position -
oldPosition.position;
47             bufferData.LastRotation = newPosition.rotation;
48         }
49
50         oldPosition = new()
```

```
51     {
52         isEmpty = false ,
53         position = new Vector3(
54             newPosition.position.x,
55             newPosition.position.y,
56             newPosition.position.z) ,
57         rotation = newPosition.rotation
58     };
59
60     yield return new WaitForSeconds(PositionSampleSeconds) ;
61 }
62 }
```

### 3.1.8 Specchiamento movimento soggetto

Il concetto è analogo a quanto illustrato per lo specchiamento locale. La differenza è che il vettore spostamento viene calcolato, ad esempio, sulla **RealHand** sinistra ma applicato alla **VirtualHand** destra senza l'inversione del vettore. In questo modo quando il soggetto trasla la mano sinistra verso l'alto, vedrà nell'ambiente virtuale la mano destra che traslerà verso l'alto e viceversa.

L'implementazione di questa soluzione si basa su quanto fatto per lo specchiamento locale, con la differenza che anziché fare riferimento ai dati del buffer della mano si utilizzano i dati letti dal buffer dell'altra mano.

### 3.1.9 Cambio punto di vista

Questa funzionalità permette di osservare l'ambiente virtuale e i propri movimenti da un punto di vista differente. Sono stati impostati due punti di vista: il primo è stato posizionato in angolo nella parte della stanza, il secondo è stato posizionato frontalmente rispetto alla posizione del soggetto nel mondo virtuale. L'implementazione è molto semplice: le varie camere disposte nell'ambiente vengono attivate o disabilitate in base alla configurazione desiderata. Per evitare l'insorgere di *motion sickness* alle camere secondarie viene applicata la rotazione della testa del soggetto fornita dal visore.

Le modifiche al comportamento dell'avatar sono cumulabili tra di loro: è quindi possibile, ad esempio, attivare lo *specchimento del movimento soggetto* e lo *specchiamento del movimento locale* in contemporanea. In questo modo se il soggetto alza il braccio destro vedrà nel mondo virtuale abbassarsi il braccio sinistro.

## 3.2 Sistema di LOG

A supporto di una analisi post sperimentazione più completa possibile è stata implementata una raccolta di dati effettuata dall'applicativo VR che vengono salvati all'interno di file nominati di LOG. Questi dati potranno essere eventualmente utilizzati dall'operatore durante la fase di analisi in parallelo ai dati raccolti dal sistema di lettura EEG.

### 3.2.1 LOG generati dal visore

L'applicativo eseguito sul visore genera i dati relativi alla sessione distribuiti su cinque file differenti. All'inizio di ogni sessione viene creata una cartella nominata con l'**Identifier** della sessione all'interno della quale vengono salvate le informazioni suddivise in diversi file (vedi 3.5) Nell'istante in cui il *client* richiede i dati di una specifica sessione, la relativa cartella viene compressa in un file *ZIP* e ritornata al *client*.

**Listing 3.5:** Struttura cartello server

```
1 SESSIONS_DATA/  
2 |-- session_20230528_122437/  
3 |   |-- general.log  
4 |   |-- leftHand.log  
5 |   |-- rightHand.log  
6 |   |-- session_data.json  
7 |   '-- world.log  
8 |-- session_20230811_224924/  
9 |   |-- general.log  
10 |  |-- leftHand.log  
11 |  |-- rightHand.log  
12 |  |-- session_data.json  
13 |  '-- world.log
```

```

14 '-- session_20230811_232132/
15   |-- general.log
16   |-- leftHand.log
17   |-- rightHand.log
18   |-- session_data.json
19   '-- world.log

```

Nella Tabella 3.2 è possibile avere una panoramica su quali informazioni sono contenute in ciascun *file* e il formato di *file* adottato.

| NOME FILE    | TIPOLOGIA   | CONTENUTO  |
|--------------|-------------|--|
| session_data | File JSON   | Contiene i dati in formato JSON della sessione   |
| general      | File di LOG | Contiene informazioni temporali su quando è iniziata la sessione e quando è iniziata la registrazione della posizione delle mani virtuali del soggetto |
| world        | File di LOG | Contiene informazioni temporali su quando il pulsante virtuale è premuto / rilasciato e su quando il target è stato raggiunto                          |
| leftHand     | File di LOG | Contiene la posizione reale e quella ritardata della mano sinistra del soggetto  |
| rightHand    | File di LOG | Contiene la posizione reale e quella ritardata della mano destra del soggetto  |

**Tabella 3.2:** File generati dal server

## Session Data

All'interno del *file* **session\_data.json** vengono salvate tutte le informazioni della sessione in formato *JSON*, come mostrato nell'esempio 3.6.

**Listing 3.6:** Dati sessione in formato JSON

```

1 {
2   "Identifier": "session_20230811_224924",
3   "Name": "Test_1",
4   "Description": "Sessione di test n. 1",
5   "CreatedAt": "2023-08-11T22:49:24.1782909+02:00",
6   "SessionBlocksList": [
7     {

```

```
8         "id ": "20230811_224716_block ",
9         "numberOfTry ": 2,
10        "restTime ": 5,
11        "delay ": 0,
12        "target ": 0
13    },
14    {
15        "id ": "20230811_224725_block ",
16        "numberOfTry ": 3,
17        "restTime ": 7,
18        "delay ": 200,
19        "target ": 1
20    }
21 ]
22 }
```

Le informazioni relative ad una sessione ricevute dal server sono le seguenti:

- **Identifier:** stringa univoca generata dal client nel momento in cui la sessione è definita
- **Name:** nome della sessione specificato dall'operatore
- **Description:** descrizione della sessione specificata dall'operatore
- **CreatedAt:** data di esecuzione della sessione
- **SessionBlocksList:** è un array che contiene i blocchi da eseguire della sessione sperimentale. Ogni elemento di questo array rappresenta un blocco della sessione. Ogni blocco ha definite le seguenti informazioni
  - **numberOfTry:** rappresenta il numero di try dell'azione che il soggetto deve eseguire
  - **restTime:** rappresenta il numero di secondi di attesa tra un try e quello successivo
  - **delay:** rappresenta i millisecondi di delay da applicare al movimento delle mani del soggetto

- **target**: indica se il target che il soggetto deve raggiungere deve comparire sempre nella stessa posizione oppure se la posizione deve essere randomizzata

## Struttura LOG

Per tutti gli altri *file* di LOG la struttura è comune. Ogni informazione salvata è legata ad un preciso istante di tempo, definito come il numero di millisecondi trascorsi a partire da un riferimento temporale. Il riferimento temporale viene stabilito nel momento in cui la sessione è avviata sul server.

La struttura adottata per i file di *LOG* è mostrata nell'esempio 3.7

**Listing 3.7:** Struttura generica file LOG

```

1 ## Reference DATE and TIME: 2023-08-11 23-21-37-364 +02
2 ## SESSION ID: id_della_sessione
3
4 delta [ms] - informazione 1 - informazione 2 - ...
5 4,15 - inf1 - inf2 - ...
6 2050,321 - inf1 - inf2 - ...
7 86373,07 - inf1 - inf2 - ...
8 86373,334 - inf1 - inf2 - ...

```

La testa del file, costituito dalle prime due righe, specifica l'istante di tempo preso come riferimento e la sessione di riferimento.

La linea successiva specifica come sono strutturati e a cosa si riferiscono i dati contenuti nel file mentre le restanti linee contengono i dati veri a propri. L'informazione presente in tutti i file LOG è il delta, che rappresenta il tempo trascorso a partire dalla data di riferimento espresso in millisecondi. Le altre informazioni variano da file a file e, ad esclusione dei LOG delle mani, contengono un codice che specifica a quale azione fanno riferimento i dati.

Nella Tabella 3.3 sono indicati il codice, l'azione che rappresenta e in quale file è presente.

Per quanto riguarda i log delle mani, l'informazione che viene salvata è la posizione della mano reale e di quella virtuale (vedi esempio 3.8), ovvero la posizione della **VirtualHand** che rappresenta le informazioni del *tracking* e quella della



| CODICE LOG              | DESCRIZIONE  | FILE    |
|-------------------------|--|---------|
| SESSION_START           | La sessione è avviata nell'ambiente virtuale             | general |
| SESSION_START_RECORDING | E' avviata la registrazione della posizione delle mani   | general |
| SESSION_STOP_RECORDING  | E' terminata la registrazione della posizione delle mani | general |
| SESSION_END             | Le sessione viene terminata                              | general |
| BUTTON_PRESSED          | Il soggetto ha premuto il pulsante                       | world   |
| BUTTON_RELEASED         | Il soggetto ha rilasciato il pulsante                    | world   |
| TARGET_REACHED          | Il soggetto ha raggiunto e toccato il target             | world   |

**Tabella 3.3:** Codici azioni usati nei file di LOG

**VirtualHand** che rappresenta le informazioni del buffer. Se il *delay* è pari a zero, le due posizioni coincidono.

**Listing 3.8:** Esempio dati LOG mano sinistra

```

1 ## Reference DATE and TIME: 2023-10-26 15-23-25-854 +02
2 ## SESSION ID: session_20231026_152324
3 ## LEFT HAND
4
5 delta [ms] - realPosition (x,y,z) - userPosition (x,y,z)
6 ...
7 1412600,916 - (-0.110, -0.461, 0.255) - (-0.107, -0.481, 0.286)
8 1412712,528 - (-0.126, -0.481, 0.261) - (-0.107, -0.481, 0.286)
9 1412824,098 - (-0.126, -0.481, 0.261) - (-0.114, -0.464, 0.277)
10 1412934,875 - (-0.074, -0.456, 0.285) - (-0.114, -0.464, 0.277)
11 1413046,234 - (-0.109, -0.464, 0.264) - (-0.121, -0.468, 0.256)
12 1413157,818 - (-0.110, -0.464, 0.273) - (-0.110, -0.461, 0.255)
13 1413269,484 - (-0.113, -0.483, 0.271) - (-0.126, -0.481, 0.261)
14 1413381,581 - (-0.113, -0.483, 0.271) - (-0.091, -0.443, 0.269)
15 1413492,058 - (-0.115, -0.454, 0.250) - (-0.074, -0.456, 0.285)
16 1413603,484 - (-0.122, -0.476, 0.255) - (-0.109, -0.464, 0.264)
17 1413715,05 - (-0.122, -0.476, 0.255) - (-0.110, -0.464, 0.273)
18 ...

```

### 3.3 Software Gestione VR

Per la configurazione delle sessioni e la gestione dei dati salvati sul visore è stato implementato un applicativo *Windows Form*. Tramite questo software di gestione è possibile recuperare l'elenco delle sessioni salvate sul server, scaricarne i dati, eliminare quelle desiderate e, soprattutto, creare nuove sessioni.

Di seguito è proposta una panoramica sulle diverse schermate che compongono l'applicativo, insieme ad una descrizione delle funzionalità presenti.

**Schermata IP** Permette l'inserimento dell'indirizzo IP locale del server, ovvero del HMD (Fig. 3.17).

**Schermata Main** Mostra la lista delle sessioni attualmente salvate sul server. Da qui è possibile procedere con il download dei file, l'eliminazione o la creazione di una nuova sessione (Fig. 3.18).

**Schermata Nuova Sessione** In questa schermata l'operatore può specificare i dati relativi alla nuova sessione (titolo e descrizione) e gestire i blocchi. Tramite i pulsanti presenti in questa schermata è possibile suddividere i blocchi in blocchi più piccoli, randomizzare l'ordine dei blocchi, eliminare un blocco oppure aggiungere un nuovo blocco (Fig. 3.19).

**Schermata Nuovo Blocco** Permette la creazione di un nuovo blocco della sessione. E' possibile specificare il numero di *try* presenti nel blocco, i secondi di pausa tra un *try* e quello successivo, il *delay* delle mani virtuali e il metodo di generazione del *target*, che può essere randomico oppure fisso in una posizione (Fig. 3.20).

**Schermata di configurazione della stanza** Da questa schermata è possibile andare a configurare la stanza virtuale in modo da far combaciare la posizione del bottone e la dimensione del tavolo con gli elementi reali della stanza. È inoltre possibile modificare l'altezza della camera (Fig. 3.21).

**Schermata di attivazione della sessione** Da questa schermata è possibile avviare l'esecuzione del blocco correntemente in esecuzione. All'attivazione viene abilitato il bottone e il log delle mani sul server (Fig. 3.22).

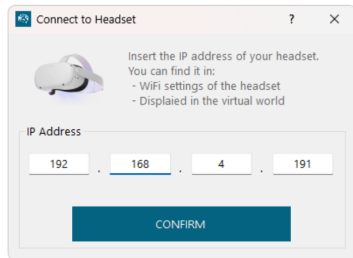


Figura 3.17: Schermata IP

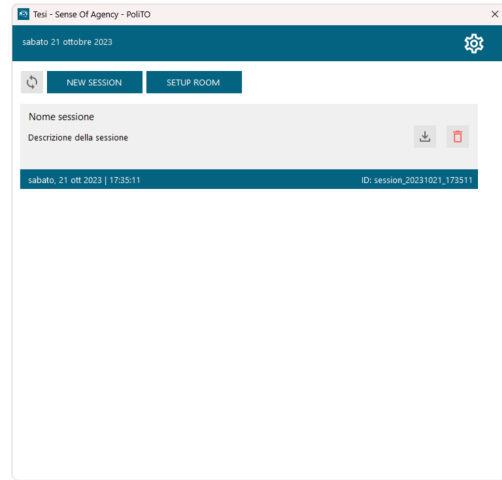


Figura 3.18: Schermata Nuovo Blocco

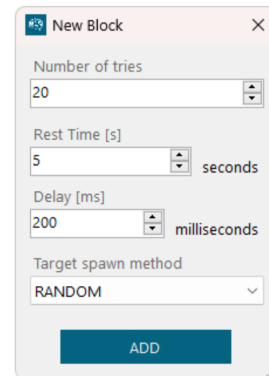
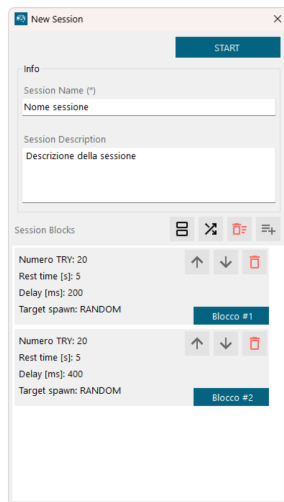
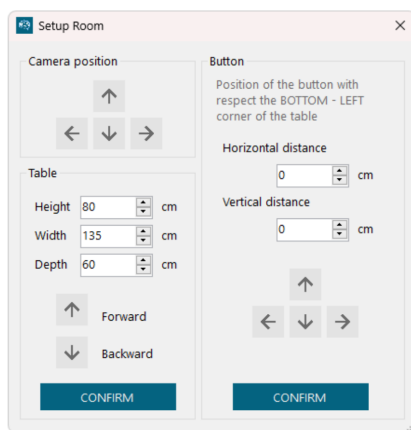
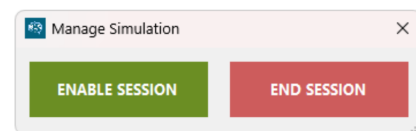


Figura 3.19: Schermata Nuova Sessione    Figura 3.20: Schermata Nuovo Blocco



**Figura 3.21:** Schermata di configurazione della stanza



**Figura 3.22:** Schermata di attivazione della sessione

# Capitolo 4

## Sperimentazione

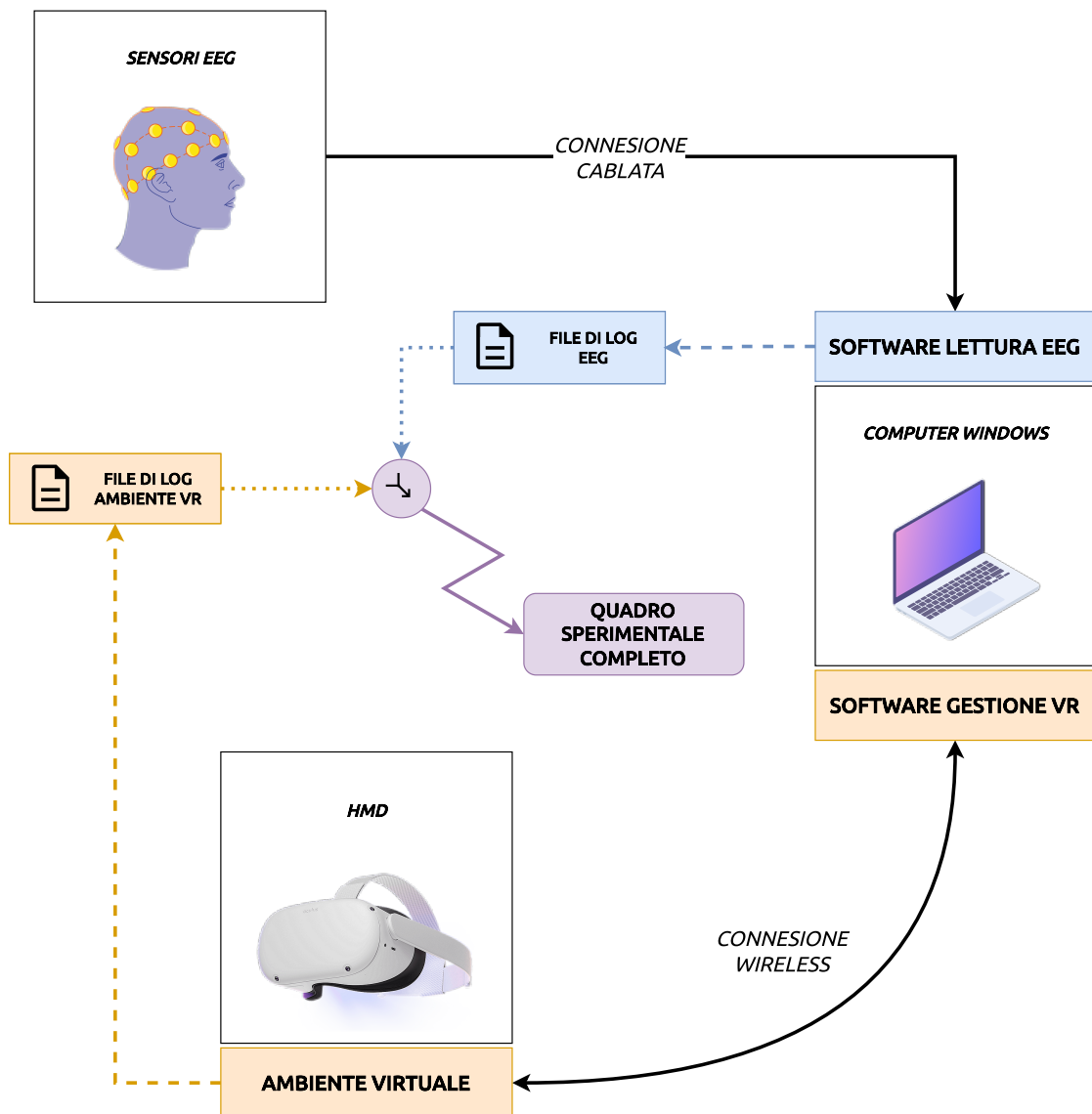
Durante la fase di sperimentazione, oltre all'applicativo VR e al software di gestione, entra in gioco un terzo componente: il sistema di lettura **EEG**. Lo schema generale della struttura del sistema si amplia, andando ad includere anche quest'ultimo componente (Fig. 4.1).

### 4.1 Sistema lettura EEG

La lettura EEG non fa parte degli elementi sviluppati per questo progetto ma occorre spiegarne il funzionamento generale per poter comprendere il funzionamento di tutto il sistema.

Il sistema di lettura EEG è composto da due parti: il blocco sensori e l'applicativo software di lettura. I sensori EEG vengono applicati sulla testa del soggetto tramite una speciale cuffia. Una volta posizionati omogeneamente sul capo viene applicato un gel che facilita la lettura del segnale elettrico generato dal cervello. Ogni sensore ha un cavo che è collegato ad un dispositivo intermedio che mette in collegamento i sensori, che producono segnali analogici, con il software di lettura, che legge dei segnali digitali. L'applicativo software utilizzato è nello specifico **Galileo NT Line - EEG.NET** [16] e si occupa della registrazione e della visualizzazione dei segnali ricevuti.

La registrazione EEG viene effettuata per canali, ovvero ogni sensore viene registrato dal software Galileo su un canale separato (vedi Fig. 4.2). Oltre ai



**Figura 4.1:** Schema completo del sistema

canali dei sensori è presente un canale aggiuntivo che nella sperimentazione viene utilizzato per avere dei punti di riferimento sul tracciato registrato. Questo canale è fondamentale per poter allineare i dati generati dall'ambiente VR con quelli generati del software di lettura EEG.

Questi riferimenti temporali vengono inviati al software tramite un pulsante meccanico costruito appositamente per questa sperimentazione. Il funzionamento

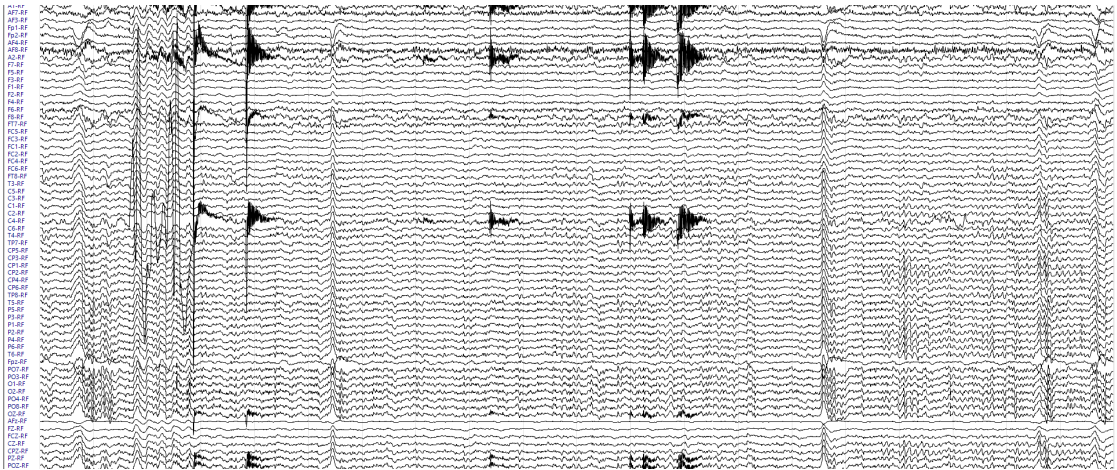


Figura 4.2: Esempio traccito EEG

del pulsante verrà approfondito poco più avanti, nella sottosezione 4.2.1.

## 4.2 Pre-sperimentazioni

Per definire il protocollo sperimentale si sono effettuate delle pre-sperimentazioni volte a capire come strutturare una sessione virtuale e come posizionare gli elementi necessari sia all'interno dell'ambiente reale sia all'interno dell'ambiente virtuale.

### Struttura classica

A prescindere dal fatto che si utilizzi o meno la Virtual Reality, il protocollo sperimentale prevede solitamente una suddivisione della sessione in blocchi. All'interno di ogni blocco il soggetto sperimentale (da qui in avanti indicato solo come soggetto) deve effettuare una specifica azione, come ad esempio dover toccare un oggetto con la mano, spostare un oggetto da una posizione ad un'altra, ecc. L'azione, che viene ripetuta più volte, è chiamata *trial*. Al termine di ogni *trial* il soggetto deve ritornare nella posizione di riposo che consiste nel tenere premuto un pulsante. Tra un *trial* e l'altro deve trascorrere un certo intervallo di tempo (ad esempio 10 secondi) necessario per poter avere una separazione più netta nelle registrazioni EEG. Ogni blocco solitamente viene variato andando a cambiare l'azione da compiere, il numero di *trial* o i tempi di attesa tra un *trial* e quello successivo.

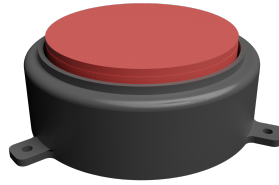
## Struttura VR

Con l'aggiunta del visore VR si è deciso di mantenere la struttura del protocollo sperimentale "classica" descritta precedentemente. Il soggetto nel momento in cui indossa il visore non ha però più modo di vedere il bottone reale da premere, nè tantomeno l'oggetto da toccare per svolgere l'azione. Si è deciso quindi di introdurre all'interno dell'ambiente virtuale un pulsante uguale a quello presente nell'ambiente reale e un *target* virtuale che compare nel momento in cui il soggetto può iniziare il *trial* e scompare una volta toccato. Con riferimento alla Figura 4.3, la sessione sperimentale virtuale risulta così strutturata: partendo dalla posizione di riposo (1), allo scadere del tempo di attesa, il soggetto dovrà sollevare la mano dal pulsante virtuale (2) (e di conseguenza dall'analogo reale) e raggiungere il *target* visualizzato nell'ambiente virtuale (3). Una volta raggiunto e toccato, il *target* virtuale scomparirà (4) e l'utente dovrà ritornare nella posizione di riposo (5), dalla quale potrà eseguire il movimento del *trial* successivo dopo il tempo di attesa.

### 4.2.1 Pulsante

Per questa sperimentazione è stato creato un pulsante reale ad hoc (Figura 4.4) più grande, in termini di superficie abilitata alla pressione, per agevolare il soggetto. Il pulsante, una volta collegato al software di lettura di EEG, permette di creare un segnale su un canale dedicato del software. Il segnale viene generato sia quando il pulsante reale viene premuto, sia quando viene rilasciato. Tramite questo canale dedicato l'operatore che si occuperà dell'analisi dei dati saprà in che istanti di tempo il soggetto è nella posizione di riposo e in quali istanti invece sta effettuando l'azione.





**Figura 4.4:** Modello 3D del pulsante

### 4.3 Protocollo sperimentale

Una volta analizzate le sessioni pre-sperimentali si è potuto definire un protocollo sperimentale definitivo che è stato applicato ad ogni sessione sperimentale effettuata all'interno di questo studio.

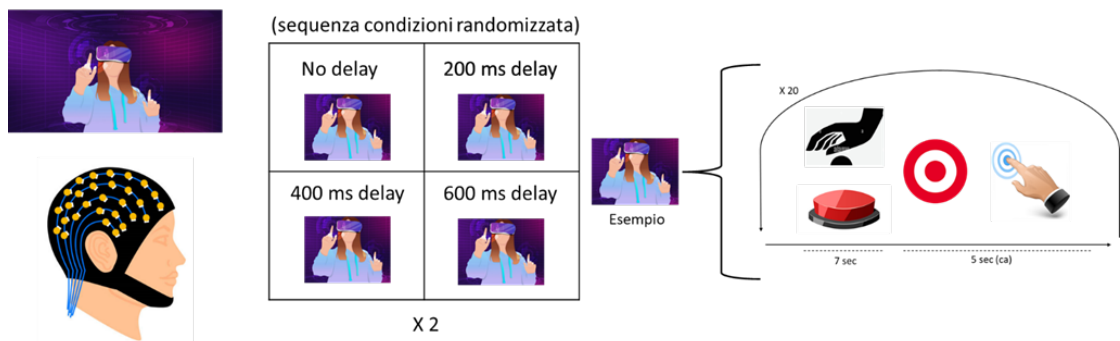
Lo studio prevede il coinvolgimento di circa 30 partecipanti reclutati presso l'*Università di Torino* e composti da volontari sani suddivisi equamente tra maschi e femmine. I partecipanti saranno reclutati dagli sperimentatori coinvolti nel progetto; il giorno dello studio ai partecipanti sarà registrata l'attività cerebrale tramite l'EEG e verrà chiesto loro di indossare un visore di realtà virtuale e di muovere mano e braccio dominanti partendo da un pulsante presente sia nell'ambiente virtuale che nella postazione sperimentale reale per raggiungere, di volta in volta, dei bersagli che appariranno nell'ambientazione virtuale.

Il protocollo, in breve, sarà così strutturato:

1. dopo aver posato la mano in modo da premere il pulsante, passeranno 7 secondi prima dell'apparizione del bersaglio che il soggetto dovrà andare a toccare
2. una volta toccato il bersaglio, questo scomparirà e il soggetto dovrà posare nuovamente la mano sul pulsante

Questo ciclo di movimenti si ripeterà per 40 volte per ciascuna delle quattro condizioni differenti; ogni condizione sarà suddivisa in due blocchi, composti da 20 *trial* ciascuno (i.e. 40 *trial* per condizione e 160 *trial* totali). Le condizioni differiranno una dall'altra nel tempo di *delay* inserito nell'ambientazione virtuale tra

il movimento visto nell'ambiente virtuale e il movimento reale eseguito dal soggetto; le condizioni saranno rappresentate da 0, 200, 400, 600 millisecondi di *delay* tra il movimento reale del partecipante ed il movimento della mano nell'ambiente virtuale. Le quattro diverse condizioni verranno bilanciate e randomizzate tra i partecipanti in modo da non avere la stessa sequenza di condizioni per partecipante (es. 1) 200, 2) 400, 3) 0, 4) 0, 5) 200, 6) 600, 7) 400, 8) 600). Questo bilanciamento tra partecipanti verrà effettuato per evitare effetti di affaticamento e di apprendimento durante le diverse condizioni.



**Figura 4.5:** Protocollo Sperimentale

La prima fase dello studio sarà sempre composta da alcuni minuti di introduzione al protocollo sperimentale e seguiti poi dal montaggio della cuffia EEG. Successivamente, verrà fatto indossare il visore VR, saranno settati i parametri spaziali in modo da coincidere con l'ambiente reale e, infine, prenderà avvio il protocollo sperimentale. Ipotizzando un tempo di raggiungimento del bersaglio e di ritorno al pulsante di circa 5 secondi si stima una durata complessiva del compito sperimentale di circa 30 minuti; aggiungendo i minuti necessari per il montaggio della cuffia EEG e la sistemazione dell'ambiente virtuale, nonché i minuti di pausa necessari per i cambiamenti di condizione, si possono stimare 75-90 minuti massimi di paradigma sperimentale.

### 4.3.1 Raccolta dati

Al termine della sessione sperimentale verrà proposto al soggetto un questionario da compilare. L'obiettivo è quello di ottenere dei dati che permettano una analisi

dell'esperienza VR su diversi aree. Quelle di interesse per questa sperimentazione sono:

**Presenza** Rappresenta la sensazione autentica di trovarsi in un mondo diverso da quello in cui ci si trova fisicamente, quindi di essere presenti nel mondo virtuale

**Immersione** Rappresenta quanto l'utente è in grado di immergersi nel mondo virtuale attraverso l'hardware fornito

**Flow** Utile per capire se l'utente è stato immerso totalmente nell'attività proposta all'interno del mondo virtuale

**Conseguenze sperimentali** Necessario per capire se l'utente ha sofferto di disturbi fisici durante l'utilizzo dell'applicazione VR

**Embodiment** Rappresenta la sensazione per l'utente di controllare effettivamente il corpo virtuale, di entrare a far parte ed immedesimarsi all'interno del mondo virtuale

Per la creazione del questionario si è deciso di appoggiarsi a questionari che sono uno standard o che puntano ad esserlo nel prossimo futuro. Utilizzare dei questionari standard, e quindi questionari utilizzati all'interno di altre sperimentazioni di realtà virtuale immersiva, offre la possibilità di effettuare dei confronti dei risultati in maniera semplice ed efficace, in quanto i risultati calcolati sono basati sulla stessa tipologia di domande.

Il primo questionario che si è preso in considerazione deriva dallo studio *Proposition and Validation of a Questionnaire to Measure the User Experience in Immersive Virtual Environments*[17] e copre gli aspetti legati a *Presence*, *Immersion*, *Flow* e *Experience consequences*. Il secondo questionario utilizzato deriva dallo studio *Avatar Embodiment. A Standardized Questionnaire*[18] che va a coprire invece l'aspetto dell'*Embodiment*. Ad ogni domanda il soggetto potrà rispondere con un valore compreso tra 1 e 7, dove 1 rappresenta la valutazione *totalmente in disaccordo* mentre 7 rappresenta la valutazione *totalmente d'accordo*.

La valutazione dell'*embodiment* richiede un passaggio in più, in quanto alcune domande verranno proposte al soggetto sperimentale al termine di ogni blocco

di azioni. Questa suddivisione del questionario è necessario per poter avere dei dati sull'*embodiment* percepito dal soggetto per ogni valore di delay applicato all'ambiente virtuale. Il *delay* varia di blocco in blocco ed è quindi necessario chiedere una valutazione al soggetto tramite un breve questionario al termine di ogni blocco.

Riporto di seguito le domande presenti all'interno del questionario.

### **Presenza**

- L'ambiente virtuale era sensibile alle azioni che ho avviato
- L'ambiente virtuale era sensibile alle azioni che ho avviato
- La qualità del display visivo mi ha distratto dall'esecuzione di attività assegnate
- I dispositivi (tracciamento delle mani) che controllavano il mio movimento nell'ambiente virtuale sembravano naturali
- Sono stato in grado di esaminare attivamente l'ambiente virtuale usando la visione
- Le mie interazioni con l'ambiente virtuale sembravano naturali
- Ho potuto concentrarmi sui compiti assegnati piuttosto che sul pulsante in se
- Ho correttamente localizzato i suoni prodotti dall'ambiente virtuale

### **Immersione**

- Sono talmente coinvolto nell'ambiente virtuale che è come se fossi all'interno del ambiente virtuale piuttosto che stare a guardare uno schermo
- Mi sentivo fisicamente in forma nell'ambiente virtuale
- Mi sono sentito stimolato dall'ambiente virtuale
- Ero talmente coinvolto nell'ambiente virtuale da perdere la cognizione del tempo
- Ero talmente coinvolto nell'ambiente virtuale che non mi rendevo conto di ciò che accadeva intorno a me

## **Flow**

- Sentivo di poter controllare perfettamente le mie azioni
- Ad ogni passo, sapevo cosa fare
- Sentivo di controllare la situazione
- Il tempo sembrava fluire in modo diverso del solito
- Il tempo sembrava accelerare
- Stavo perdendo il senso del tempo
- Non ero preoccupato del giudizio degli altri
- Non mi preoccupavo di ciò che gli altri pensavano di me
- Sentivo di vivere un momento emozionante
- Questa esperienza mi ha dato un grande senso di benessere
- Quando parlo dell'esperienza nell'ambiente virtuale, provo emozioni che vorrei condividere

## **Conseguenze sperimentali**

- Ho sentito un aumento nella sudorazione durante la mia interazione con l'ambiente virtuale
- Ho sofferto di "pienezza della testa" durante l'interazione con l'ambiente virtuale
- Ho sentito un aumento della salivazione durante l'interazione con l'ambiente virtuale
- Ho sofferto di mal di testa durante la mia interazione con l'ambiente virtuale
- Ho sofferto di vertigini ad occhi aperti durante l'interazione con l'ambiente virtuale
- Ho sofferto di affaticamento durante la mia interazione con l'ambiente virtuale

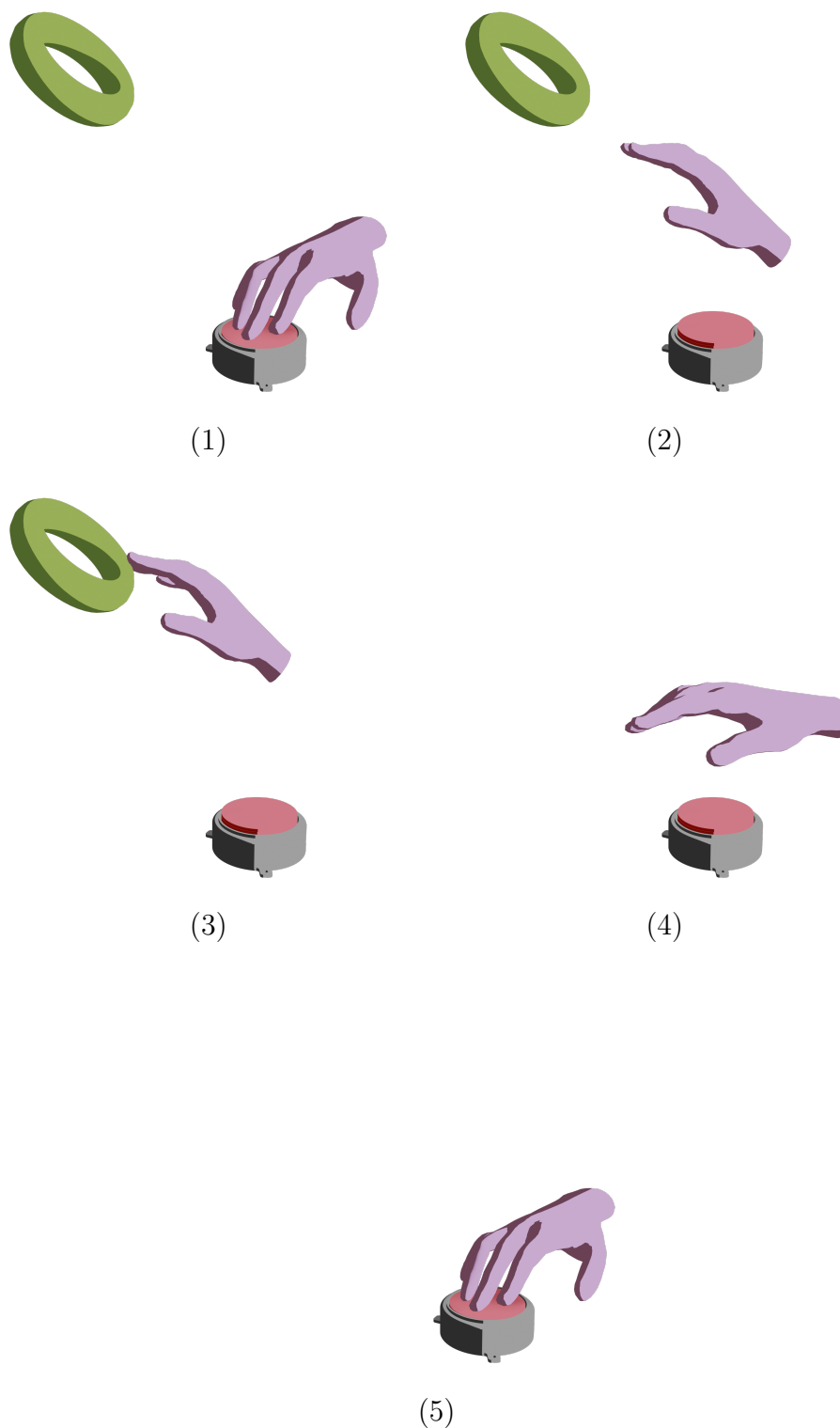
- Ho sofferto di nausea durante la mia interazione con l'ambiente virtuale
- Ho sofferto di vertigini durante l'interazione con l'ambiente virtuale

### **Embodiment finale**

- Sembrava che il mio (reale) corpo si stesse trasformando in un corpo "avatar"
- Mi sentivo come se il mio corpo fosse situato dove ho visto il corpo virtuale
- Mi sentivo come se stessi indossando abiti diversi da quando sono venuto in laboratorio
- Mi sentivo come se il mio corpo fosse cambiato
- Mi sentivo come se il mio (reale) corpo stesse andando verso il corpo virtuale o come se il corpo virtuale stesse andando alla deriva verso il mio (reale) corpo

### **Embodiment per blocco**

- Mi sono sentito fuori dal mio corpo
- Ho avuto la sensazione che i movimenti del corpo virtuale influenzassero i miei movimenti
- Mi sentivo come se il corpo virtuale fosse il mio corpo
- Mi sembrava di poter controllare il corpo virtuale come se fosse il mio corpo



**Figura 4.3:** Schema esecuzione sessione sperimentale

# Capitolo 5

## Risultati

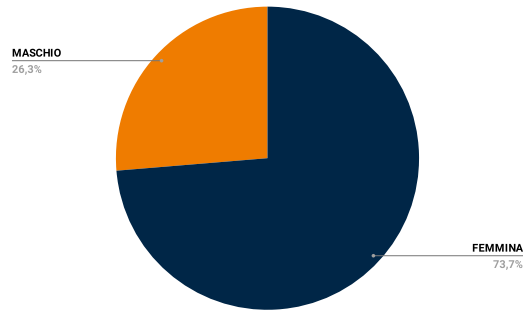
L'analisi dei dati raccolti durante la sperimentazione è stata effettuata su due aree: l'area **oggettiva**, formata dai dati EEG, e l'area **soggettiva**, composta dai dati raccolti tramite i questionari.

Oltre a mostrare i risultati derivanti dalle due aree si è cercato di trovare un punto di incontro tra l'*oggettività* e la *soggettività*, in modo da poter valutare un valore soggettivo come l'*embodiment* attraverso un valore oggettivo ricavato dall'EEG.

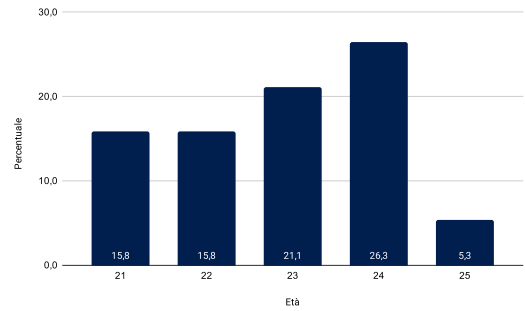
### 5.1 Campione sperimentale

Al termine della sperimentazione i soggetti che hanno partecipato sono stati **19**. Dai dati ricavati dai questionari sottoposti durante la sperimentazione si evince che il bacino di soggetti è composto dal **26.3%** di soggetti maschili e dal **73.7%** di soggetti femminili (fig. 5.1). L'età media si attesta intorno ai **23.5 anni**, distribuita come mostrato nel grafico visibile nella figura 5.2. La maggior parte dei soggetti appartiene alla categoria degli studenti frequentati un corso di medicina / psicologia (fig. 5.4).

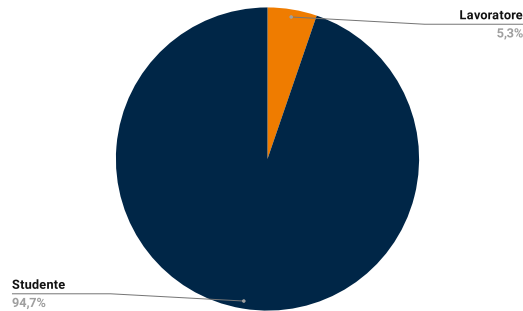




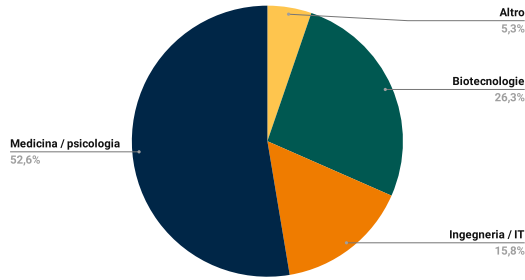
**Figura 5.1:** Composizione dei soggetti partecipanti



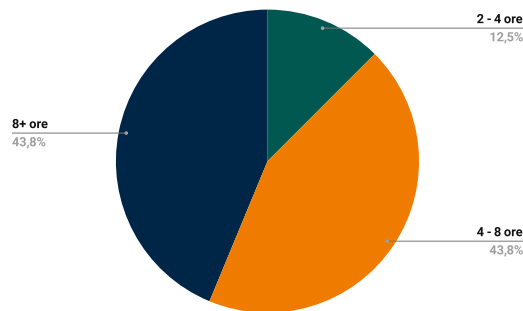
**Figura 5.2:** Distribuzione dell'età dei partecipanti (in percentuale)



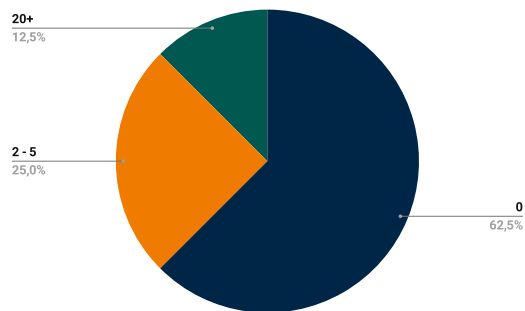
**Figura 5.3:** Impiego dei partecipanti



**Figura 5.4:** Ambito di studio/lavoro dei partecipanti



**Figura 5.5:** Ore giornaliere trascorse utilizzando un dispositivo elettronico



**Figura 5.6:** Numero di volte utilizzo di un visore di realtà virtuale

I dati sul numero di ore schermo (fig. 5.5) e sul numero di utilizzi di un visore (fig. 5.6) mostrano come ad oggi solo una minima parte del campione trascorre meno di 4 ore al giorno utilizzando un dispositivo elettronico. La maggioranza non ha mai utilizzato un visore, dato che mostra come la realtà virtuale sia ancora una tecnologia poco diffusa in ambito *consumer* e che ci indica che il campione sperimentale era formato da soggetti poco esperti nell'utilizzo di tale dispositivi.

## 5.2 Risultati oggettivi

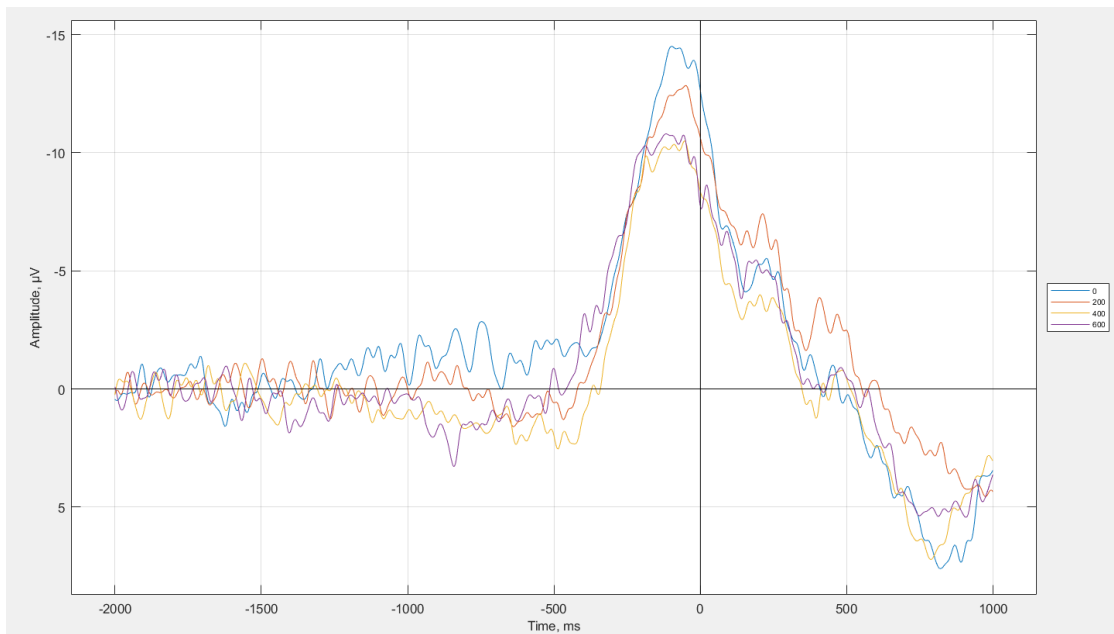
L'analisi dei dati ricavati tramite EEG è stata effettuata dal dottorando Campaci Francesco del dipartimento di neuroscienze dell'Università di Torino. Riporto di seguito in modo letterale la sua analisi.

### 5.2.1 Metodi di Analisi

I dati elettroencefalografici sono stati analizzati con *EEGLAB2022.1* su *Matlab R2023a*. Sono state estratte le varie epoche per l'*averaging* finale all'interno del range temporale -2000ms / +1000ms prima e dopo il *trigger* di movimento. Sono stati applicati i filtri *high-pass filter* 0.1Hz e *low-pass filter* 30Hz e i dati sono stati sottoposti a un processo di *re-referencing*, prendendo come referenza l'elettrodo OZ. I risultati sono relativi agli elettrodi AFZ, C1, C2, CZ, F1, F2, FCZ, FZ; elettrodi posti a livello della corteccia motoria.

### 5.2.2 Risultati

Nel grafico (fig. 5.7) sono riportati i *readiness potential* (da qui, RP) associati ognuno a una determinata condizione sperimentale (0, 200ms, 400ms, 600ms di delay fra movimento effettivo e movimento dell'avatar nell'ambientazione virtuale). Dal grafico è possibile notare una chiara e palese tendenza legata all'ampiezza del potenziale di preparazione motoria in associazione al tempo di delay associato. Nello specifico, è possibile osservare una correlazione negativa tra tempo di delay e ampiezza del potenziale; infatti, maggiore è il tempo di delay, inferiore risulta essere l'ampiezza del potenziale. L'ampiezza del potenziale sembra poi stabilizzarsi quando

**Figura 5.7:** Risultati RP

il livello di delay è sufficientemente elevato da rappresentare una incongruenza chiara per il soggetto (a partire dai 400ms).

### 5.2.3 Discussione

Il risultato è certamente interessante e, nonostante la scarsità campionaria ( $N = 11$ ), rappresenta un ottimo dato pilota per completare lo studio sperimentale. Il RP è un potenziale associato all'attività cerebrale coinvolta nella preparazione motoria; infatti, esso anticipa il movimento ed è pertanto un potenziale movimento-relato[19]. È un potenziale che ha fatto da protagonista in numerose ricerche di neuroscienze; tuttavia, il suo ruolo e le sue caratteristiche non sono tuttora comprese chiaramente. Ad esempio, originariamente inteso come una misura oggettiva della preparazione motoria, molti studi sperimentali hanno poi mostrato come la sua forma d'onda sia influenzata da molteplici fattori, come l'affaticamento soggettivo o la complessità generale dell'azione. È utile qui, ai fini di una discussione sui risultati ottenuti dalla raccolta e analisi dati, considerare alcuni dei principali fattori che possono influenzare la forma d'onda del RP. Il potenziale, per prima cosa, è stato trovato essere maggiore nei soggetti con un'elevata intenzione e coinvolgimento

nel movimento [20]; infatti, quando il movimento compiuto diventa noioso e poco coinvolgente per il soggetto, l'ampiezza del potenziale tende a decrescere. Come accennato sopra, inoltre, la forza richiesta per l'esecuzione del movimento è un secondo importante fattore; tendenzialmente, maggiore è la forza necessaria per il movimento maggiore è l'ampiezza del RP [21][22]. Un'altra situazione di movimento in cui l'ampiezza del RP è maggiore si ha durante l'apprendimento di nuovi schemi motori [23]. Infatti, quando viene dato da eseguire un nuovo compito motorio, come ad esempio suonare uno strumento musicale per la prima volta, l'ampiezza del potenziale aumenta; invece, una volta divenuti esperti nell'esecuzione dello schema motorio, l'ampiezza diminuisce. È possibile ora fare un collegamento tra RP e *Sense of Agency* (SoA). Alcuni studi hanno indagato la relazione focalizzandosi in particolare modo sulle conseguenze dell'azione nell'ampiezza del potenziale. In uno studio, per esempio, i ricercatori chiesero ai partecipanti di premere un pulsante per avviare un suono; questo avveniva in due condizioni sperimentali: una condizione di normale previsione e una condizione di bassa previsione. Nella condizione di normale previsione i suoni erano effettivamente avviati dalla pressione del pulsante; in questo modo, si veniva a creare una contingenza tra la pressione del pulsante e il suono nonché una relazione causa-effetto nella mente dei partecipanti. Invece, nella condizione di bassa previsione la causalità tra pressione del tasto e suono veniva resa dubbia in quanto un terzo dei suoni veniva presentato prima della pressione del pulsante. Le analisi si concentrarono poi sul potenziale evento-relato osservato prima e dopo l'avvio del suono. I risultati della registrazione dei potenziali di preparazione motoria indicarono un RP più precoce nella condizione di normale previsione rispetto a quella di bassa previsione, anche se le differenze non risultarono significative data la piccola grandezza campione [24]. Gli stessi autori indagarono il collegamento tra l'affidabilità della conseguenza di un'azione e il RP modificando la probabilità di avere un determinato *outcome* data una certa azione. Similmente allo studio precedente, anche in questo lavoro vi era una condizione di *outcome* costante, in cui l'azione volontaria dei partecipanti azionava sempre il suono, e una condizione di *outcome* incostante, in cui l'azione avviava il suono solamente nella metà dei *trial*. In questo modo i ricercatori trovarono RP più ampi nella condizione costante rispetto a quella di *outcome* incostante [25]. Questi risultati suggeriscono un legame tra l'affidabilità percepita di osservare una certa conseguenza data

una certa azione e le caratteristiche del RP; in particolare, un'elevata contingenza azione-effetto sembra corrispondere un'elevata attività neurale precedente all'azione, e viceversa. I risultati del nostro studio sembrano portare a conclusioni molto simili: maggiore la congruenza tra il movimento effettivo del soggetto e il risultato del movimento stesso, sotto forma di feedback visivo (movimento della mano virtuale) e feedback di esecuzione dell'azione intesa (raggiungimento del bersaglio), maggiore l'ampiezza del RP; al contrario, minore congruenza comporta una minor ampiezza. È interessante notare come questo effetto sembri essere progressivo e lineare e non invece un effetto tutto-nulla legato a congruenza/incongruenza tra intenzione motoria e feedback di movimento effettivo, anche se l'effetto sembra poi stabilizzarsi raggiunta una certa soglia di delay.

#### **5.2.4 Limiti dello studio e sviluppi futuri**

La conclusione di questi dati preliminari è plausibile e in linea con la letteratura sul tema; tuttavia, alcuni limiti di questo studio pilota sono comunque da considerare. In particolare, il campione di soli undici soggetti non permette di trarre conclusioni di tipo statistico e si scontra con la variabilità inter-soggetto intrinseca del RP, variabilità osservata sia a livello temporale che nell'ampiezza del potenziale [26]. Per di più, non tutte le componenti del RP risultano essere sempre visibili in tutti i soggetti e, in alcuni casi, il potenziale risulta addirittura essere assente, probabilmente per via di alcune peculiarità anatomiche [21]. Queste caratteristiche del RP rendono la scarsa numerosità campionaria un limite su cui sarà necessario lavorare.

I risultati di questo studio pilota rimangono promettenti e spingono verso ulteriori approfondimenti; in particolare, lo svolgimento dello studio con una grandezza campionaria elevata permetterà di trarre conclusioni più certe e scevre di variabili spurie.

### **5.3 Risultati soggettivi**

I risultati soggettivi sono stati ricavati dalle risposte fornite dai partecipanti alla sperimentazione tramite il questionario somministrato al termine di ogni

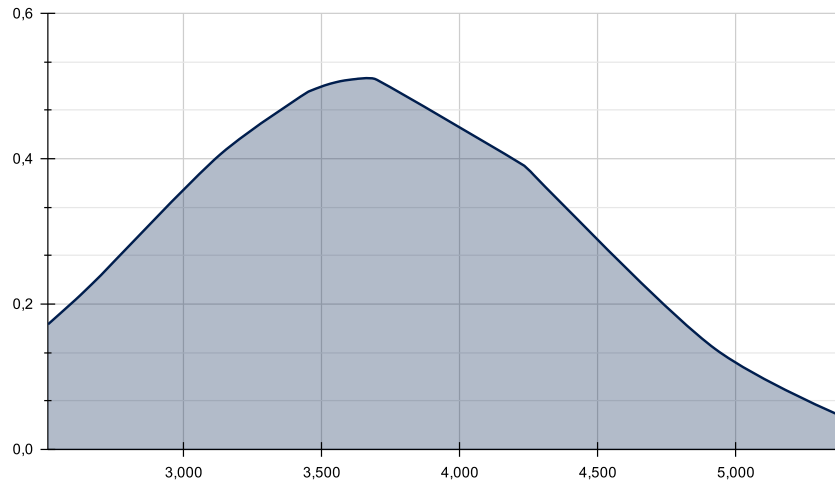
blocco sperimentale (da qui **QB**) e il questionario somministrato al termine della sperimentazione (da qui **QF**). Dal QF sono stati estrapolati i dati relativi a presenza, immersione, *flow*, conseguenze sperimentali ed *embodiment* (globale). Dal QB sono stati estrapolati i dati relativi all'*embodiment* influenzato dal *delay*.

Nell'analisi dei dati il primo passo è stato quello di calcolare un valore generale per ogni ambito considerato in modo da ottenere una panoramica su come l'ambiente virtuale è stato recepito dai soggetti sperimentali. Per i valori di presenza, immersione, *flow* e conseguenze sperimentali il valore è stato calcolato effettuando una media delle risposte per ogni soggetto, ottenendo un valore generico per il soggetto, e poi come una media delle medie per ottenere il valore globale. Calcolo diverso per quanto riguarda l'*embodiment*: siccome i dati provengono da due questionari, il valore finale è stato ottenuto utilizzando le risposte del QF e le risposte del QB relative allo 0 *delay*. Si è calcolata la media delle risposte del QF per ogni soggetto e la media delle risposte del QB per ogni soggetto relative allo 0 *delay*. Dopodichè si sono combinate le medie calcolate secondo lo schema proposto in *Avatar Embodiment. A Standardized Questionnaire*[18], adattando la formula in modo da non considerare le domande rimosse perchè ritenute non pertinenti con la sperimentazione. Si è così ottenuto un valore dell'*embodiment* per ciascun soggetto, distribuito come mostrato nella figura 5.8. Infine, per ottenere il valore globale di *embodiment* è stata calcolata la media dell'*embodiment* di ciascun soggetto. Nella tabella 5.1 sono riportati i valori globali per i vari ambiti.

|                          |            |
|--------------------------|------------|
| PRESENZA                 | <b>4.7</b> |
| IMMERSIONE               | <b>4.1</b> |
| FLOW                     | <b>4.7</b> |
| CONSEGUENZE SPERIMENTALI | <b>2.2</b> |
| EMBODIMENT               | <b>3.7</b> |

**Tabella 5.1:** Risultati soggettivi esperienza VR

Com'è possibile notare, i valori ottenuti sono piuttosto buoni principalmente per quanto riguarda presenza, *flow* e conseguenze sperimentali (per queste ultime un valore minore significa minori effetti collaterali sviluppati dai soggetti). Minore invece è stata la sensazione di *embodiment*: la causa principale è sicuramente da attribuire alla mancanza di un avatar virtuale completo. Osservare nel mondo



**Figura 5.8:** Distribuzione embodiment 0ms

virtualmente solamente i modelli delle mani anzichè un corpo virtuale completo ha sicuramente influito negativamente sulla sensazione di *embodiment* provata. Una seconda possibile causa, suggerita da uno dei soggetti partecipanti, potrebbe essere dovuta ai diversi elementi fatti indossare: la cuffia EEG con i suoi vari elementi di supporto potrebbero aver "*distratto*" i soggetti dal mondo virtuale.

### 5.3.1 Embodiment e delay

Il questionario QB è stato somministrato per ottenere delle informazioni sull'*embodiment* in relazione al *delay* impostato nell'ambiente virtuale. Tenendo presente che sono stati eseguiti 2 blocchi per ogni condizione di *delay* e per ciascuno dei 19 soggetti, in totale sono stati analizzati 38 campioni di dati per ogni condizione di *delay*. Riporto di seguito le domande proposte con il questionario QB per permettere una comprensione dei successivi dati più agevole:

**QB - #1** Mi sentivo fuori dal mio corpo

**QB - #2** Ho avuto la sensazione che i movimenti del corpo virtuale influenzassero i miei movimenti

**QB - #3** Mi sentivo come se il corpo virtuale fosse il mio corpo

**QB - #4** Mi sembrava di poter controllare il corpo virtuale come se fosse il mio corpo

Nell'analisi di questi dati il primo passaggio è stato effettuare un **TEST T**<sup>1</sup> sui campioni per appurare che la differenza tra i campioni non derivi dal caso. Il test è stato effettuato per ogni domanda del QB, confrontando la condizione 0 con quelle a 200, 400 e 600. Nella tabelle seguenti (tabella 5.2) sono esposti i risultati dei *test t*.

| TEST T. |          |         |          |
|---------|----------|---------|----------|
| QB - #1 |          | QB - #2 |          |
| Delay   | <i>p</i> | Delay   | <i>p</i> |
| 200     | 0.20     | 200     | 0.83     |
| 400     | 0.34     | 400     | 0.027    |
| 600     | 0.55     | 600     | 0.000089 |

| QB - #3 |            | QB - #4 |             |
|---------|------------|---------|-------------|
| Delay   | <i>p</i>   | Delay   | <i>p</i>    |
| 200     | 0.13       | 200     | 0.45        |
| 400     | 0.000068   | 400     | 0.000016    |
| 600     | 0.00000013 | 600     | 0.000000012 |

**Tabella 5.2:** Test T. calcolato su domande QB

Escludendo la domanda *QB - #1*, per la quale la probabilità *p* che la variazione dei valori tra i diversi *delay* sia dovuta al caso risulta essere molto alta, si ha in generale che per i valori di *delay* di 400 e 600 il valore *p* è molto piccolo. Considerando invece il *delay* di 200ms, per tutte le domande effettuate il valore *p* risulta essere molto elevato rispetto agli altri valori di *delay*. Questo potrebbe essere dovuto al fatto che il soggetto non si sia accorto del *delay* aggiuntivo introdotto, a differenza dei *delay* più importanti di 400 e 600ms, considerando quindi il caso 0 e 200 come analoghi in termini di esperienza virtuale.

Appurato che le variazioni tra i *delay* non siano dovute al caso, si è proseguito con il calcolare l'*embodiment* globale per ciascun valore di *delay*, in modo analogo

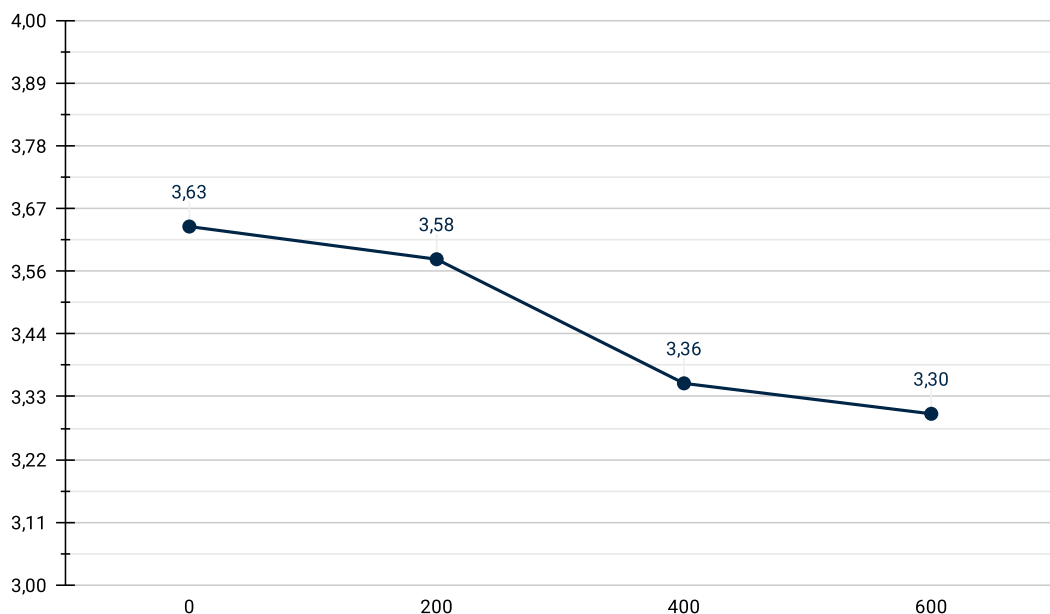
---

<sup>1</sup>Il *test t* è un tipo di analisi statistica utilizzata per confrontare le medie di due gruppi e determinare se le differenze tra loro siano più probabilmente dovute al caso.



a quanto fatto per l'*embodiment* globale a *delay* 0. Si può vedere (fig. 5.9) come la sensazione di *embodiment* vada a diminuire all'aumentare del *delay*, in modo più significativo tra il caso 200 e il caso 400 e in modo minore tra 400 e 600. Questo comportamento risulta essere analogo a quanto osservato durante l'analisi dei dati EEG:

"L'ampiezza del potenziale sembra poi stabilizzarsi quando il livello di *delay* è sufficientemente elevato da rappresentare una incongruenza chiara per il soggetto (a partire dai 400ms)"



**Figura 5.9:** Andamento *embodiment* con *delay*

Per capire quali sono i valori che influiscono sull'*embodiment* occorre analizzare l'andamento delle domande del QB rispetto al *delay* (fig. 5.10). I dati relativi a *QB - #3* e *QB - #4* hanno un andamento negativo, inversamente proporzionale all'aumento del *delay*. Più il valore di *delay* impostato aumenta, minore è la sensazione di avere il controllo sul proprio corpo, con una diminuzione maggiore tra il caso 400 e il caso 600. Andamento inverso si invece ha per la *QB - #2*: in questo caso la sensazione aumenta all'aumentare del *delay*, con un andamento più

accentuato tra il caso 400 e il 600 e rimanendo pressoché costante tra il caso 0 e il caso 200 (ulteriore fattore a favore che il soggetto non si accorgesse del *delay* impostato a 200ms). Questo comportamento è stato osservato anche durante le sperimentazioni: la maggior parte dei soggetti aveva la tendenza a rallentare il movimento reale per compensare quello virtuale affetto da *delay*, andando quindi ad influenzare il movimento reale rispetto a quello virtuale.

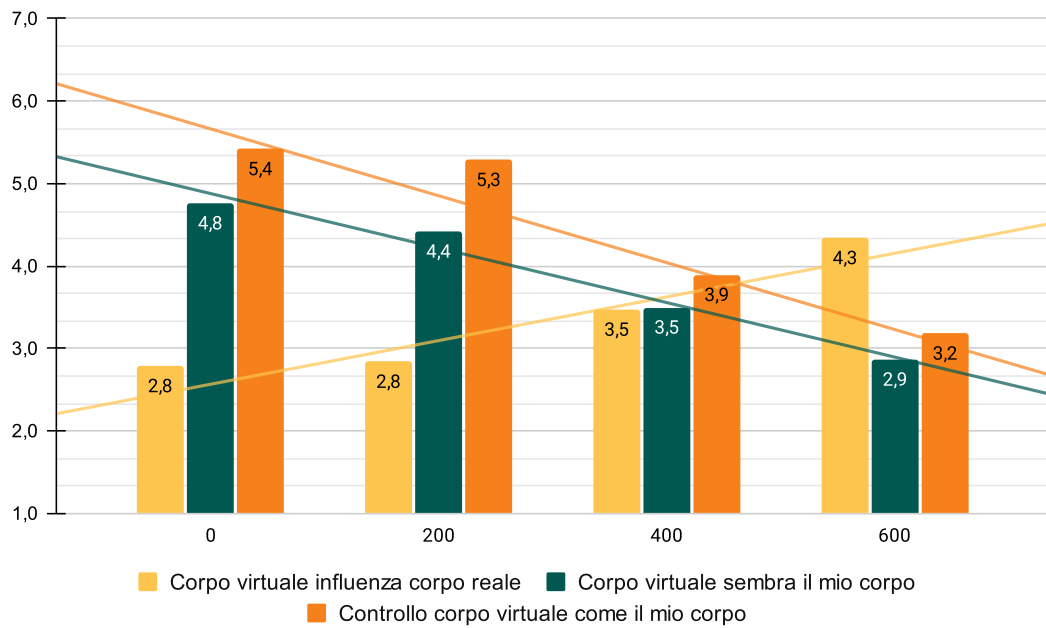


Figura 5.10: Domande embodiment con delay

## 5.4 Dati oggettivi e soggettivi

L'ultimo passaggio di questa analisi è stato quello di cercare una possibile correlazione tra i dati soggettivi e quelli oggettivi, in modo da poter ottenere dati oggettivi relativi al livello di *embodiment* sperimentato. Le caratteristiche dell'onda RP che sono state valutate sono:

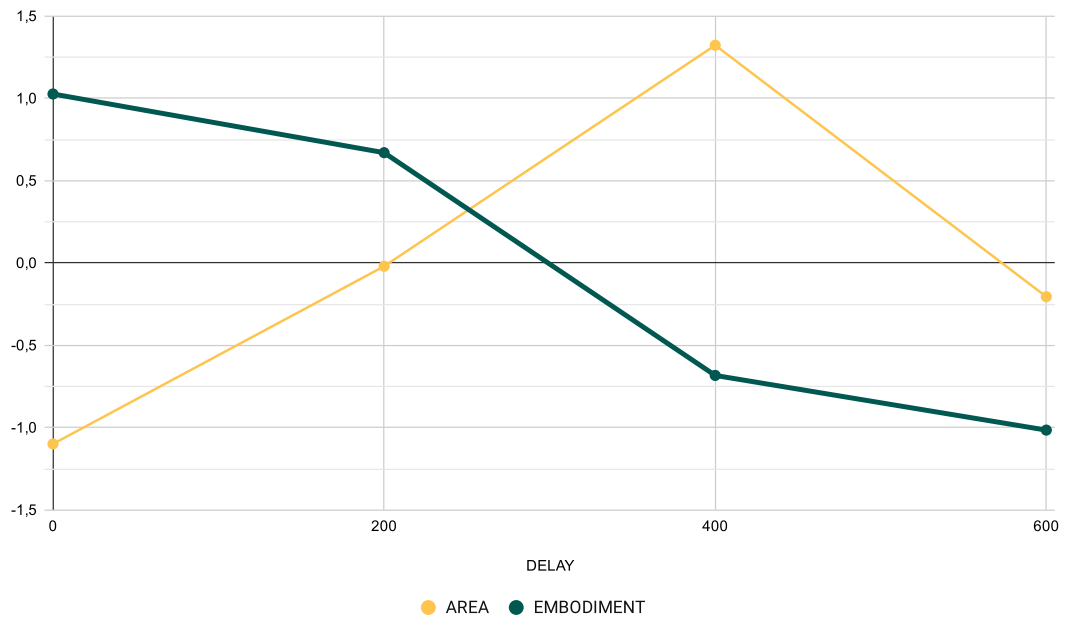
**T\_MAX\_AMPL (ms)** istante di tempo in cui l'onda RP raggiunge il valore massimo

**max\_AMPL(microV)** il valore massimo di ampiezza raggiunto dall'onda

**AREA (ms\*microV)** area sottesa alla curva

**MEAN\_AMPL (microV)** ampiezza media della curva

Nel grafico in figura 5.11 sono mostrati l'andamento dei valori normalizzati relativi all'*embodiment* e all'**area della curva RP**, indicata dai colleghi di psicologia come il parametro più rilevante tra quelli sopracitati.



**Figura 5.11:** Andamento dati oggettivi e soggettivi

Ad una prima osservazione sembrerebbe esserci una relazione negativa tra l'area della curva RP e l'*embodiment*. Questa correlazione negativa sembrerebbe essere valida per i valori di *delay* 0, 200 e 400 ma non per il valore di *delay* 600. Per capire se l'andamento sia effettivamente valido si è calcolata la correlazione tra l'*embodiment* e l'area partendo dai valori di ciascun soggetto per ogni domanda QB. Nelle quattro tabelle seguenti (tabella 5.3) sono presentati gli *indici di correlazione di Pearson* ottenuti per ciascuna domanda e per ogni valore di *delay*.

I valori più interessanti e sui quali vorrei far concentrare il lettore sono quelli relativi allo *delay* 0 e alle domande **QB - #2** e **QB - Media** e sul *delay* 200

| Indici correlazione di Pearson |         |            |         |
|--------------------------------|---------|------------|---------|
| 0 DELAY                        |         | 200 DELAY  |         |
|                                | Area RP |            | Area RP |
| QB - #1                        | -0.24   | QB - #1    | 0.058   |
| QB - #2                        | 0.42    | QB - #2    | 0.12    |
| QB - #3                        | -0.13   | QB - #3    | -0.43   |
| QB - #4                        | 0.13    | QB - #4    | -0.32   |
| QB - Media                     | 0.34    | QB - Media | -0.25   |

| 400 DELAY  |         | 600 DELAY  |         |
|------------|---------|------------|---------|
|            | Area RP |            | Area RP |
| QB - #1    | -0.068  | QB - #1    | 0.18    |
| QB - #2    | -0.099  | QB - #2    | 0.076   |
| QB - #3    | 0.18    | QB - #3    | 0.22    |
| QB - #4    | 0.16    | QB - #4    | 0.076   |
| QB - Media | 0.13    | QB - Media | 0.17    |

Tabella 5.3: Indici di correlazione di Pearson

per la domanda **QB - #3**. Per tutte e tre esiste una *correlazione moderata* (negativa nell'ultimo caso), il che significa che potrebbe essere possibile basarsi sul valore oggettivo dell'area della curva RP per ricavare informazioni soggettive sull'*embodiment*, ma solo per valori di *delay* 0 e 200. Al momento però non è possibile definire nessun tipo di soglia o conversione tra dato oggettivo e soggettivo, ma con un numero più alto si potrebbe confermare e consolidare queste previsioni.

## Capitolo 6

# Conclusioni

Alla luce dei risultati ottenuti si può affermare, in maniera preliminare, che sia il *readiness potential* sia la sensazione di *embodiment* sono influenzati dall'aggiunta di un *delay* virtuale. Inoltre, la presenza di una correlazione moderata tra il valore oggettivo (area della curva RP) e alcuni aspetti dell'*embodiment* fa ben sperare sull'ottenere un livello di *embodiment* a partire dai parametri del RP, anche se al momento tale dato non è stato possibile definirlo con precisione.

La ricerca iniziata con questo progetto di tesi è lontano dall'essere definita conclusa: il numero di soggetti sperimentali risulta ancora troppo basso per poter ottenere dei risultati sicuri e alcuni aspetti possono sicuramente essere migliorati.

A livello di applicazione VR, le richieste (ovvero la gestione del *delay* ed altre configurazioni non utilizzate) sono state realizzate con successo. Alcuni aspetti dell'ambiente virtuale richiederebbero delle modifiche, per fare un esempio l'aggiunta di un avatar virtuale completo permetterebbe di aumentare il senso di presenza ed *embodiment* dei soggetti sperimentali all'interno dell'ambiente virtuale.

Ciononostante, gli obiettivi prefissati sono stati raggiunti e i risultati ricavati dallo studio pilota serviranno come punto di partenza per una seconda fase della sperimentazione che avrà come primo obiettivo la raccolta dati su un numero maggiore di soggetti.

# Bibliografia

- [1] ARtillery Intelligence. *Extended reality (XR) market size worldwide from 2021 to 2026 (in billion U.S. dollars) [Graph]*. Mar. 2023. URL: <https://www.statista.com/statistics/591181/global-augmented-virtual-reality-market-size/> (cit. a p. 2).
- [2] Thomas Alsop. *AR/VR headset shipments by segment 2027 [Graph]*. Lug. 2023. URL: <https://www.statista.com/statistics/1396559/ar-vr-headset-shipments-by-segment/> (cit. a p. 3).
- [3] Hayden Scott, Connor Griffin, William Coggins, Brooke Elberson, Mohamed Abdeldayem, Tuhin Virmani, Linda J Larson-Prior e Erika Petersen. «Virtual reality in the neurosciences: Current practice and future directions». en. In: *Front. Surg.* 8 (2021), p. 807195 (cit. a p. 3).
- [4] Thomas D. Parsons, Andrea Gaggioli e Giuseppe Riva. «Virtual Reality for Research in Social Neuroscience». In: *Brain Sciences* 7.4 (2017). ISSN: 2076-3425. DOI: 10.3390/brainsci7040042. URL: <https://www.mdpi.com/2076-3425/7/4/42> (cit. a p. 5).
- [5] James W. Moore. «What Is the Sense of Agency and Why Does it Matter?» In: *Frontiers in Psychology* 7 (2016). ISSN: 1664-1078. DOI: 10.3389/fpsyg.2016.01272. URL: <https://www.frontiersin.org/articles/10.3389/fpsyg.2016.01272> (cit. a p. 6).
- [6] Aaron Schurger, Pengbo “Ben” Hu, Joanna Pak e Adina L. Roskies. «What is the readiness potential?» In: *Trends in Cognitive Sciences* 25.7 (lug. 2021), pp. 558–570. DOI: 10.1016/j.tics.2021.04.001 (cit. a p. 8).

- [7] Jin Woo Choi, Haram Kwon, Jaehoon Choi, Netiwit Kaongoen, Chaeun Hwang, Minuk Kim, Byung Hyung Kim e Sungho Jo. «Neural Applications Using Immersive Virtual Reality: A Review on EEG Studies». In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 31 (2023), pp. 1645–1658. DOI: 10.1109/TNSRE.2023.3254551 (cit. a p. 10).
- [8] Anne-Marie Brouwer, Jasper Van Der Waa e Hans Stokking. *Predicting head rotation using EEG to enhance streaming of images to a Virtual Reality headset*. 2018 (cit. a p. 10).
- [9] Gaiqing Kong, Kang He e Kunlin Wei. «Sensorimotor experience in virtual reality enhances sense of agency associated with an avatar». In: *Consciousness and Cognition* 52 (2017), pp. 115–124. ISSN: 1053-8100. DOI: <https://doi.org/10.1016/j.concog.2017.04.018>. URL: <https://www.sciencedirect.com/science/article/pii/S1053810016303051> (cit. a p. 11).
- [10] Liu Wang, Mengjie Huang, Chengxuan Qin, Yiqi Wang e Rui Yang. «Movement Augmentation in Virtual Reality: Impact on Sense of Agency Measured by Subjective Responses and Electroencephalography». In: *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 2022, pp. 832–833. DOI: 10.1109/VRW55335.2022.00267 (cit. a p. 11).
- [11] URL: <https://www.meta.com/it-it/help/quest/articles/headsets-and-accessories/controllers-and-hand-tracking/hand-tracking-quest-2/> (cit. a p. 18).
- [12] URL: <https://docs.unity3d.com/Packages/com.unity.xr.hands@1.3/manual/index.html> (cit. a p. 18).
- [13] Diar Abdulkarim, Massimiliano Di Luca, Poppy Aves, Mohamed Maaroufi, Sang-Hoon Yeo, R. Chris Miall, Peter Holland e Joeseeph M. Galea. «A methodological framework to assess the accuracy of virtual reality hand-tracking systems: A case study with the Meta Quest 2». In: *Behavior Research Methods* (2023). DOI: 10.3758/s13428-022-02051-8 (cit. a p. 21).
- [14] Unity Technologies. *ScriptableObject*. URL: <https://docs.unity3d.com/2023.2/Documentation/Manual/class-ScriptableObject.html> (cit. a p. 27).

- 
- [15] Fabio Cielo. *Fabio-sky/tesi-SOA-polito: Applicazione Ambiente VR per il progetto di tesi "sense of agency and ERP correlates on delayed VR"*. URL: <https://github.com/fabio-sky/tesi-soa-polito> (cit. a p. 28).
- [16] URL: <http://www.ebneuro.com/en/routine-eeg/galileo-nt-line-eeg-net-routine> (cit. a p. 42).
- [17] Katy Tcha-Tokey, Olivier Christmann, Emilie Loup-Escande e Simon Richir. «Proposition and Validation of a Questionnaire to Measure the User Experience in Immersive Virtual Environments». In: *International Journal of Virtual Reality* 16.1 (2016), pp. 33–48. URL: <https://hal.science/hal-01404497> (cit. a p. 48).
- [18] Tabitha C. Peck e Mar Gonzalez-Franco. «Avatar Embodiment. A Standardized Questionnaire». In: *Frontiers in Virtual Reality* 1 (2021). ISSN: 2673-4192. DOI: 10.3389/frvir.2020.575943. URL: <https://www.frontiersin.org/articles/10.3389/frvir.2020.575943> (cit. alle pp. 48, 59).
- [19] Marjan Jahanshahi e Mark Hallett, cur. *The Bereitschaftspotential: Movement-Related Cortical Potentials*. Kluwer Plenum, 2003 (cit. a p. 56).
- [20] Hans H. Kornhuber e Luder Deecke. «Hirnpotentialänderungen bei Willkurbewegungen und Passiven Bewegungen des menschen: Bereitschaftspotential und reafferente potentiale». In: *Pflugers Archiv für die Gesamte Physiologie des Menschen und der Tiere* 284.1 (1965), pp. 1–17. DOI: 10.1007/bf00412364 (cit. a p. 57).
- [21] James G. Colebatch. «Bereitschaftspotential and movement-related potentials: Origin, significance, and application in disorders of human movement». In: *Movement Disorders* 22.5 (2007), pp. 601–610. DOI: 10.1002/mds.21323 (cit. alle pp. 57, 58).
- [22] Wilfried Lang. «Surface recordings of the bereitschaftspotential in Normals». In: *The Bereitschaftspotential* (2003), pp. 19–34. DOI: 10.1007/978-1-4615-0189-3\_2 (cit. a p. 57).
- [23] David J. Wright, Paul Holmes, Francesco Di Russo, Michela Loporto e Dave Smith. «Reduced motor cortex activity during movement preparation following



- a period of motor skill practice». In: *PLoS ONE* 7.12 (2012). DOI: 10.1371/journal.pone.0051886 (cit. a p. 57).
- [24] Rin Minohara, Wen Wen, Shunsuke Hamasaki, Takaki Maeda, Hiroshi Yamakawa, Satoshi Shibuya, Yukari Ohki, Atsushi Yamashita e Hajime Asama. «Readiness potential reflects the predictive aspect of sense of agency». In: *The Abstracts of the international conference on advanced mechatronics: toward evolutionary fusion of IT and mechatronics: ICAM 2015.6.0* (2015), pp. 353–354. DOI: 10.1299/jsmeicam.2015.6.353 (cit. a p. 57).
- [25] Wen Wen, Rin Minohara, Shunsuke Hamasaki, Takaki Maeda, Qi An, Yusuke Tamura, Hiroshi Yamakawa, Atsushi Yamashita e Hajime Asama. «The readiness potential reflects the reliability of action consequence». In: *Scientific Reports* 8.1 (2018). DOI: 10.1038/s41598-018-30410-z (cit. a p. 57).
- [26] Luder Deecke, Berta Grozinger e H. H. Kornhuber. «Voluntary finger movement in man: Cerebral potentials and theory». In: *Biological Cybernetics* 23.2 (1976), pp. 99–119. DOI: 10.1007/bf00336013 (cit. a p. 58).