

POLITECNICO DI TORINO

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

Tesi di Laurea Magistrale

**Data Visualization in VR:  
prototipazione di uno strumento  
interattivo per la rappresentazione  
umanizzata dei dataset relativi  
alle persone.**



**Politecnico  
di Torino**

**Relatore**

Prof. Francesco STRADA

**Candidata**

Mariagrazia Pia GIUNTA

MARZO 2026



# Abstract

Questo lavoro presenta la prototipazione e la valutazione sperimentale di uno strumento di visualizzazione dati in realtà virtuale concepito per dataset relativi a persone, con l'obiettivo di esplorare un approccio "umanizzato" alla rappresentazione dell'informazione.

Il prototipo permette di mappare simultaneamente fino a dieci categorie diverse per ogni individuo, sfruttando la spazialità immersiva della VR (Virtual Reality) e la presenza di numerosi NPCs (Non-Player Characters) per mantenere leggibilità anche in presenza di grandi quantità di dati, che tendono altrimenti a de-umanizzare i soggetti rappresentati.

L'applicazione è stata sviluppata in Unity 6 adottando l'architettura DOTS/ECS per la gestione efficiente di moltitudini di NPCs, integrando sintesi vocale locale tramite PiperTTS, modelli tridimensionali realizzati con Blender e una colonna sonora composta da contenuti audio royalty-free; i dataset utilizzati, provenienti da repository online o costruiti ad hoc, descrivono scenari demografici e comportamentali eterogenei.

La valutazione dell'esperienza è stata progettata attraverso test di usabilità basati su protocollo Think-Aloud moderato, coinvolgendo sia utenti generici (studenti) sia profili più vicini ai contesti applicativi della visualizzazione (CRM, insegnanti, operatori marketing), con l'obiettivo di indagare la leggibilità del sistema di mapping a dieci dimensioni, l'usabilità dell'interfaccia e il potenziale dell'ambiente immersivo nel restituire ai dati la loro dimensione umana originaria.

I risultati dei suddetti test mostrano in che misura la combinazione di realtà virtuale, interazione e metafore antropomorfe possa costituire un contributo innovativo al campo della data visualization orientata alle persone.

# Ringraziamenti

*Ai miei genitori.  
Non ci sono chilometri o oceani che possano farmi sentire lontana da voi.*

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Contesto . . . . .	1
1.2	Motivazioni dietro al progetto . . . . .	2
1.3	Idea fondante e obiettivi . . . . .	3
1.4	Struttura del documento . . . . .	3
<b>2</b>	<b>Stato dell'arte</b>	<b>5</b>
2.1	Visualizzazione dati multidimensionale . . . . .	5
2.2	Visualizzazione dati interattiva . . . . .	5
2.3	Visualizzazione dati immersiva . . . . .	7
2.3.1	Trasposizione in VR . . . . .	8
2.3.2	Gamification . . . . .	9
2.3.3	Geospazialità . . . . .	9
2.3.4	Caso studio: DataGarden . . . . .	11
2.3.5	Teletrasporto . . . . .	13
<b>3</b>	<b>Concept</b>	<b>15</b>
3.1	Esperienza utente . . . . .	15
3.2	Analisi requisiti . . . . .	16
3.3	Scelte progettuali . . . . .	17
3.3.1	Scelte imposte dai requisiti . . . . .	17
3.3.2	Visualizzazione dati . . . . .	18

3.3.3	Navigazione e interazione . . . . .	20
3.3.4	Interfacce . . . . .	23
3.3.5	Stile . . . . .	26
3.4	Target utenti . . . . .	27
<b>4</b>	<b>Prototipazione</b>	<b>29</b>
4.1	Paradigma ECS e Unity DOTS . . . . .	29
4.1.1	Introduzione a ECS con DOTS . . . . .	29
4.1.2	Approccio scelto e strumenti usati . . . . .	30
4.1.3	Prestazioni . . . . .	30
4.2	Implementazione . . . . .	32
4.2.1	File di input . . . . .	32
4.2.2	File di setup . . . . .	34
4.2.3	Scripts . . . . .	42
4.2.4	Unity . . . . .	45
4.3	Versione 1.0 VS Versione 1.1 . . . . .	48
<b>5</b>	<b>User testing</b>	<b>51</b>
5.1	Campione selezionato . . . . .	51
5.2	Metodologia e strumenti . . . . .	51
5.3	Criteri di valutazione . . . . .	53
5.4	Risultati . . . . .	57
<b>6</b>	<b>Limiti e sviluppi futuri</b>	<b>69</b>
6.1	File di input . . . . .	70
6.2	Tutorial . . . . .	70
6.3	Legende . . . . .	70
6.4	Interazione . . . . .	71
6.4.1	TopView . . . . .	71
6.5	Funzionalità . . . . .	72

6.6 Estetica e ottimizzazione . . . . .	72
<b>7 Conclusioni</b>	<b>75</b>
<b>Sigle e acronimi</b>	<b>77</b>
<b>Indice delle tabelle</b>	<b>79</b>
<b>Indice delle figure</b>	<b>81</b>
<b>Bibliografia</b>	<b>85</b>



# Capitolo 1

## Introduzione

### 1.1 Contesto

Con il termine realtà virtuale si indica un insieme di tecnologie che permettono di collocare l'utente all'interno di un ambiente tridimensionale simulato digitalmente, nel quale è possibile muoversi e interagire in modo coerente con la scena.

A seconda del grado di immersione, si va da soluzioni non completamente immersive, fruite tramite schermo e periferiche di input tradizionali, a sistemi che mirano a sostituire in larga misura le percezioni dell'ambiente fisico con quelle generate artificialmente.

La realtà virtuale, seppur non ancora diffusa su larga scala, trova oggi numerose applicazioni in svariati ambiti, da quello videoludico a quello formativo, dal settore sanitario e psicoterapeutico alle installazioni museali interattive.

Con visualizzazione dei dati si intende la rappresentazione grafica di informazioni quantitative o qualitative mediante elementi visivi come grafici, mappe, diagrammi e altre forme di codifica visiva. L'obiettivo principale è rendere più immediata la comprensione di andamenti, relazioni e pattern che, in forma puramente numerica o testuale, sarebbero difficili da cogliere [1].

Negli ultimi decenni, la visualizzazione dati si è trasformata da insieme relativamente limitato di grafici statici a un campo estremamente articolato, che comprende strumenti interattivi, dashboard, animazioni e visualizzazioni integrate in applicazioni web e ambienti tridimensionali. Lo sviluppo di tecnologie hardware e software ha reso possibile lavorare su dataset sempre più grandi e complessi, sperimentando rappresentazioni che vanno oltre i grafici bidimensionali tradizionali e includono, tra le altre, visualizzazioni immersive e in realtà virtuale [2, 3].

## 1.2 Motivazioni dietro al progetto

Esistono due principali motivazioni alla base di questo progetto: la prima di natura applicativa, l'altra di carattere umanistico e psicologico.

Dal punto di vista applicativo, si volevano infatti esplorare nuovi approcci all'uso della realtà virtuale come strumento di visualizzazione dati. La realtà virtuale offre infatti una serie di caratteristiche intrinseche che la portano a distinguersi dagli strumenti di visualizzazione tradizionali: oltre alle componenti visive, condivise con le rappresentazioni bidimensionali dei dati, si elencano infatti immersività, sonorizzazione, navigazione e interazione nello spazio tridimensionale.

Si è quindi voluto indagare sull'adeguatezza di tali caratteristiche alla rappresentazione dei dati, sperimentando l'impiego dello spazio tridimensionale per distribuire le informazioni e progettando modalità di esplorazione che combinino canali visivi, interattivi e uditivi.

La motivazione principale del progetto nasce dall'esigenza di contrastare la tendenza alla deumanizzazione, che emerge quando i fenomeni umani vengono rappresentati mediante grandi numeri e statistiche aggregate. Quando le persone sono ridotte a conteggi, percentuali o barre su un grafico, esse rischiano infatti di perdere visibilità come individui, favorendo processi di *numbing* e distacco emotivo. Tali dinamiche sono state documentate nella letteratura psicologica, in particolare negli studi sulla *compassion fade* e sull'attenuazione della risposta empatica all'aumentare del numero di vittime [4, 5].



**Figura 1.1:** Andamento della valutazione soggettiva del salvataggio di vite in funzione del numero di vite a rischio [5, Fig. 3].

Come evidenzia la Figura 1.1, *saremo portati a non sentire molta differenza, né a valutarla, tra il salvare 87 vite e salvarne 88* [5, p. 4].

## 1.3 Idea fondante e obiettivi

Sulla base delle motivazioni precedentemente esposte, si è scelto di procedere con la progettazione e lo sviluppo prototipale di uno strumento di visualizzazione dati di carattere generico, parametrico e personalizzabile, che consentisse all'utente di effettuare una mappatura 1:1 tra dataset relativi alle persone e NPCs interattivi collocati all'interno di una stanza virtuale parametrica.

Nell'ottica di esplorare le possibilità di rappresentazione offerte dalla VR, il primo obiettivo del progetto è l'individuazione e implementazione di un ampio insieme di possibili mappature tra dati e ambiente virtuale, con l'intento di metterle a disposizione dell'utente come opzioni di personalizzazione della visualizzazione, mantenendo al contempo una buona leggibilità complessiva.

Il vero cuore del progetto consiste tuttavia nell'obiettivo di restituire ai dati sulle persone la loro dimensione umana, trasformando aggregati statistici in presenze individuali con cui l'utente può entrare in relazione all'interno dello spazio virtuale.

## 1.4 Struttura del documento

L'organizzazione del presente documento riflette il percorso seguito nello sviluppo di *inPersona*, dalla definizione del problema fino alla validazione sperimentale e alle prospettive future.

Nel Capitolo 2 viene presentato lo stato dell'arte, con una rassegna delle principali tecniche di visualizzazione di dati multidimensionali, interattive e immersive, e con l'analisi di alcuni casi studio rilevanti per il progetto.

Il Capitolo 3 introduce il concept di *inPersona*, descrivendo l'esperienza utente, i requisiti individuati, le scelte progettuali alla base del sistema e il target di riferimento. Il Capitolo 4 è dedicato alla fase di prototipazione: dopo una breve introduzione al paradigma ECS e a Unity DOTS, vengono descritte l'architettura implementativa, la pipeline di input e setup e le principali componenti software, includendo anche le differenze tra le versioni 1.0 e 1.1 del prototipo.

Il Capitolo 5 illustra il protocollo di user testing scelto, il campione coinvolto, i criteri di valutazione adottati e i risultati ottenuti.

Infine, il Capitolo 6 discute in modo critico i limiti emersi e propone possibili sviluppi futuri, mentre il Capitolo 7 raccoglie le conclusioni generali del lavoro, sintetizzando i contributi della tesi e delineando le direzioni di ricerca e sviluppo aperte.

# Capitolo 2

## Stato dell'arte

### 2.1 Visualizzazione dati multidimensionale

Per visualizzazione dati multidimensionale si intende la rappresentazione di insiemi di dati caratterizzati da più variabili, la cui raffigurazione va ben oltre le due dimensioni gestibili in modo diretto da un semplice grafico cartesiano.

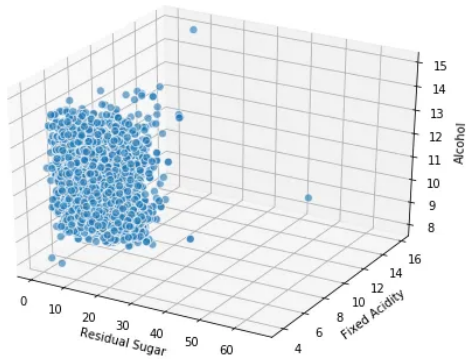
L'approccio alla visualizzazione dei dati multidimensionali offerto dalla rappresentazione statica tradizionale prevede l'uso sistematico di codifiche visive multiple. Vengono quindi impiegati colore, posizione (2D o 3D), forma, grandezza, trasparenza e sottografici (*facets*). Queste caratteristiche possono poi essere combinate in layout specifici per l'esplorazione di dataset fino a 5-6 dimensioni [6].

Dalle Figure 2.1 - 2.6 è evidente che, per quanto riguarda visualizzazioni 2D o 3D su supporti bidimensionali, come la carta o lo schermo, al crescere delle informazioni rappresentate aumenta anche il carico cognitivo a cui l'utente o il lettore vengono sottoposti.

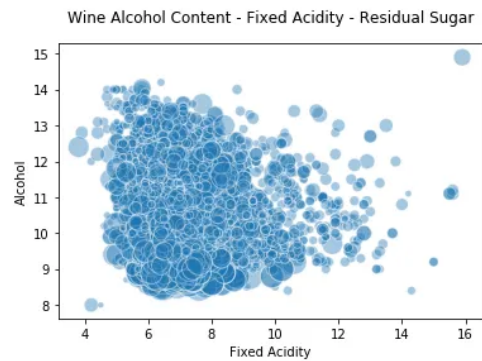
### 2.2 Visualizzazione dati interattiva

Oggi la visualizzazione dati non si limita alla semplice rappresentazione grafica, ma include un ampio spettro di interazioni che consentono agli utenti di esplorare, manipolare e interrogare i dati in modo dinamico.

La visualizzazione dati interattiva permette all'utente di selezionare, filtrare e manipolare i dati grazie all'uso di grafici dinamici, controlli di interazione e feedback in tempo reale. Questo tipo di visualizzazione consente agli utenti di esplorare i

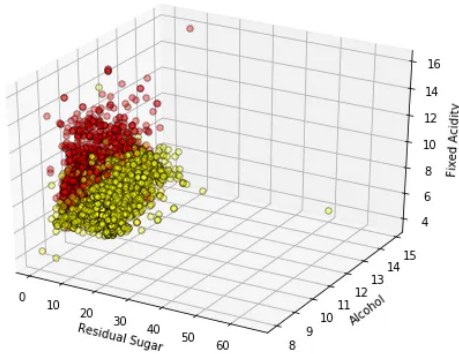


**Figura 2.1:** Esempio di uso della profondità per dati a tre dimensioni [6]



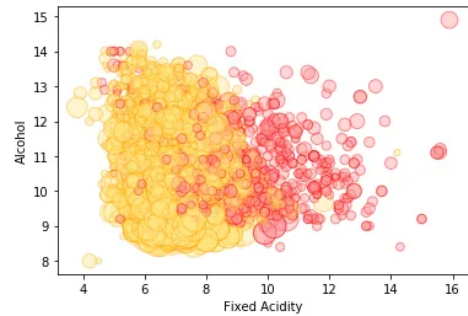
**Figura 2.2:** Esempio di uso della grandezza per dati a tre dimensioni [6]

Wine Residual Sugar - Alcohol Content - Acidity - Type



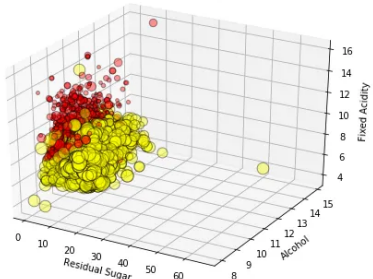
**Figura 2.3:** Esempio di uso combinato di profondità e colore per dati a quattro dimensioni [6]

Wine Alcohol Content - Fixed Acidity - Residual Sugar - Type



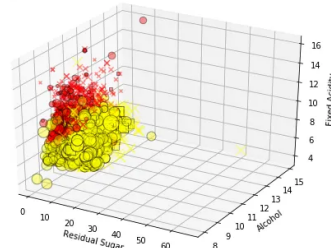
**Figura 2.4:** Esempio di uso combinato di grandezza e colore per dati a quattro dimensioni [6]

Wine Residual Sugar - Alcohol Content - Acidity - Total Sulfur Dioxide - Type



**Figura 2.5:** Esempio di uso combinato di colore, profondità e grandezza per dati a cinque dimensioni [6]

Wine Residual Sugar - Alcohol Content - Acidity - Total Sulfur Dioxide - Type - Quality



**Figura 2.6:** Esempio di uso combinato di colore, profondità, grandezza e forma per dati a sei dimensioni [6]

dati in modo più intuitivo e coinvolgente, facilitando l'identificazione di pattern nascosti e relazioni che potrebbero non essere evidenti nelle visualizzazioni statiche [7].

Le applicazioni di Business Intelligence, come Tableau o Microsoft Power BI, permettono all'utente di costruire dashboard personalizzate tramite cui è possibile passare da viste aggregate a dettagli puntuali, confrontare sottogruppi, interrogare scenari e aggiornare in tempo reale le rappresentazioni [8, 9].

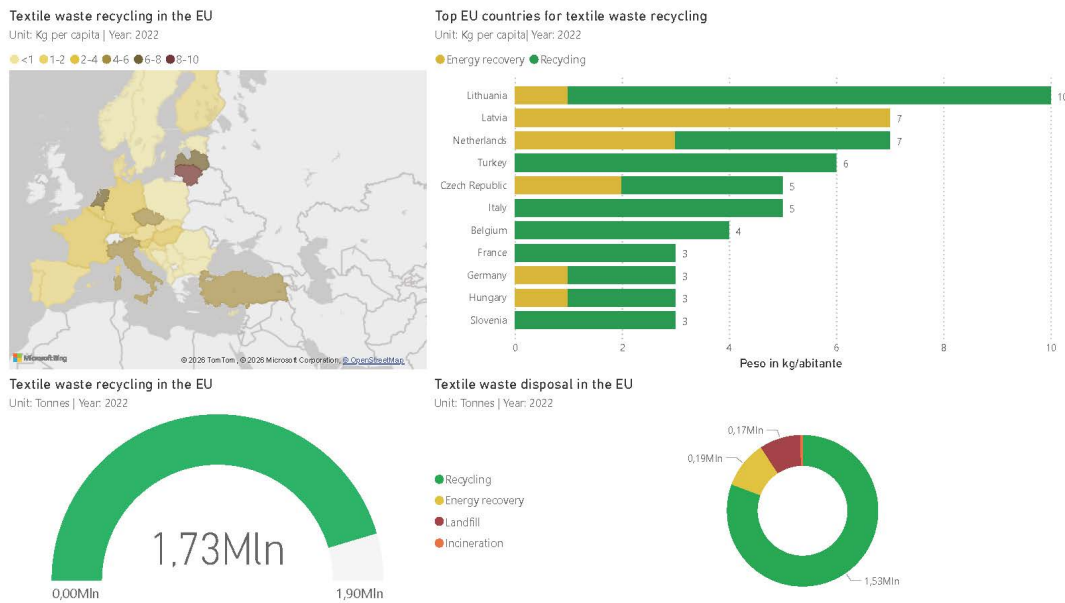


Figura 2.7: Esempio di dashboard interattiva in Microsoft Power BI.

L'*interactive visualization*, pur costituendo un'evoluzione rispetto alla visualizzazione statica, rimane tuttavia vincolata alla bidimensionalità dei supporti, aprendo dunque la strada alla ricerca di nuove forme di interazione e rappresentazione, tra cui le esperienze immersive.

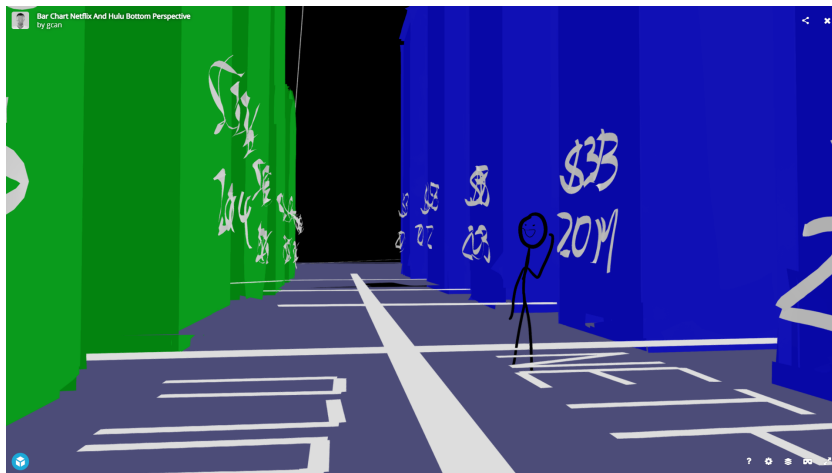
## 2.3 Visualizzazione dati immersiva

Le esperienze immersive in realtà virtuale estendono il paradigma della visualizzazione interattiva introducendo immersione spaziale, *embodiment* e nuove metafore visive. In questa sezione viene presentata una rassegna di alcuni approcci esistenti all'*immersive data visualization*.

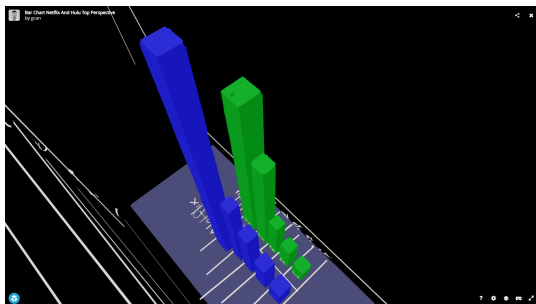
### 2.3.1 Trasposizione in VR

La soluzione più immediata consiste nella trasposizione diretta in VR dei grafici classici [10]. Il risultato sono ambienti tridimensionali popolati con istogrammi, grafici a barre, grafici a linee, heatmap o altri tipi di grafici che, pur mantenendo la grammatica visiva classica, sfruttano le funzionalità offerte dalla VR per permettere all'utente di immergersi, cambiare prospettiva, passando da FPS (First Person Shooter) a TPS (Third Person Shooter), e interagire con i dati all'interno di spazi navigabili tramite meccaniche di teletrasporto, manipolazione e zoom tipiche della VR.

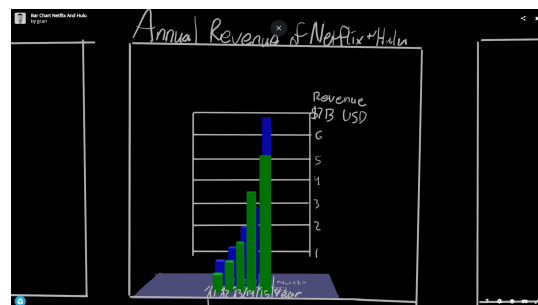
Le Figure 2.8 - 2.10 mostrano un prototipo di visualizzazione dati in VR che sfrutta la metafora dei grattacieli per i grafici a barre e integra la visualizzazione in FPS, TPS e 2D [10].



**Figura 2.8:** Modalità di navigazione in FPS [10].



**Figura 2.9:** Modalità di navigazione in TPS [10].



**Figura 2.10:** Modalità di visualizzazione in 2D [10].

Esempi di applicazioni esistenti che seguono questo approccio sono DataVizVR, mostrato in Figura 2.11, e LookVR, in Figura 2.12. Queste applicazioni portano la *visual analytics* in un ambiente VR in cui l'utente può caricare i propri dataset, mappare variabili su assi, colore e dimensione, manipolare le visualizzazioni con gestures e combinare operazioni di filtro, raggruppamento e ordinamento in uno spazio tridimensionale interattivo.

Questi approcci mirano principalmente a estendere in VR il repertorio già noto delle visualizzazioni 2D, offrendo un passaggio relativamente graduale per l'utente.

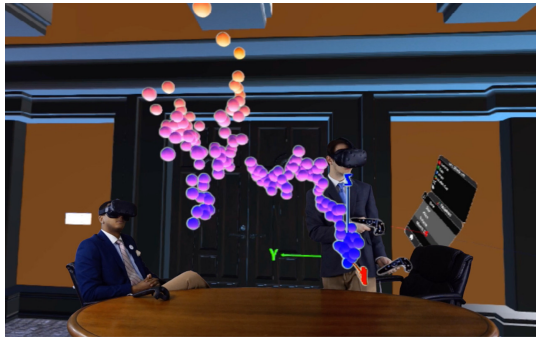


Figura 2.11: DataVizVR [11].

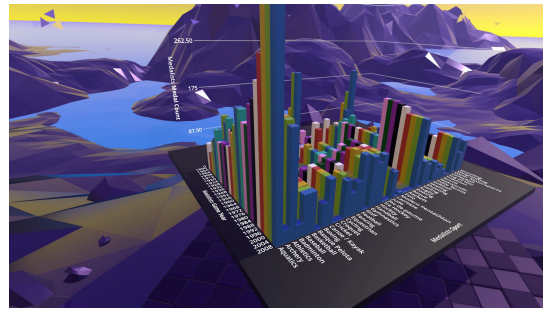


Figura 2.12: LookVR [12].

### 2.3.2 Gamification

Oltre alla semplice trasposizione dei grafici 2D in ambienti VR, LookVR sperimenta un approccio innovativo basato sulla gamification, trasformando le visualizzazioni in esperienze ludiche che sfruttano la componente fisica e motoria della VR.

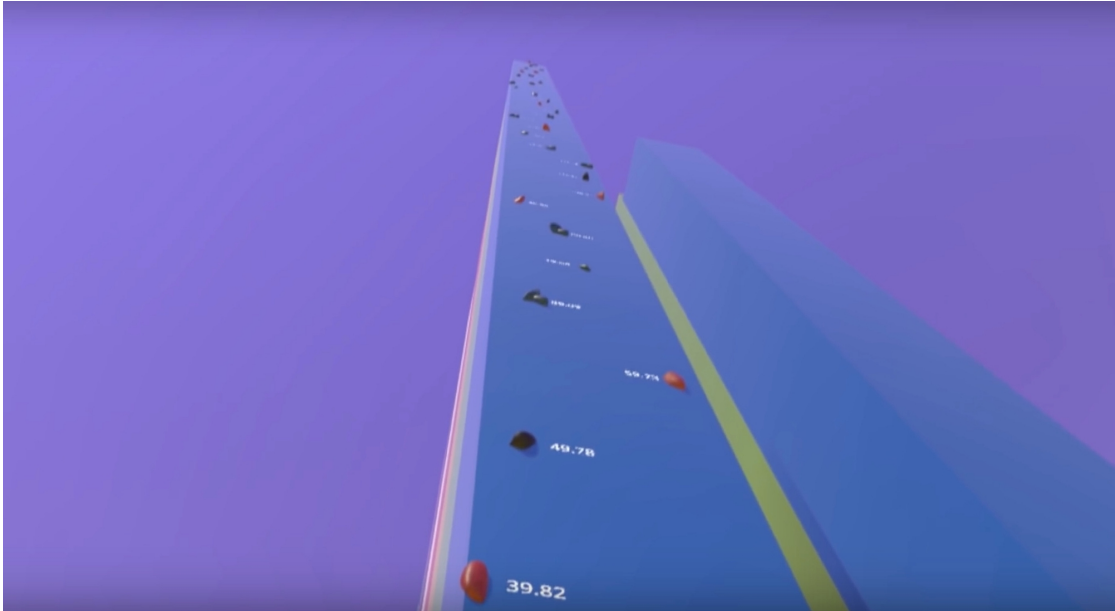
La metafora proposta nella demo è quella di una parete da arrampicata, mostrata in Figura 2.13, dove ogni appiglio rappresenta un dato. In questo esempio, l'atto fisico di scalare un muro diventa un meccanismo per ancorare percettivamente e mnemonicamente l'ordine di grandezza rappresentato.

La gamification, in questo senso, aumenta l'engagement e il tempo di esposizione ai dati, potenzialmente a costo di una minore precisione nella lettura di valori numerici.

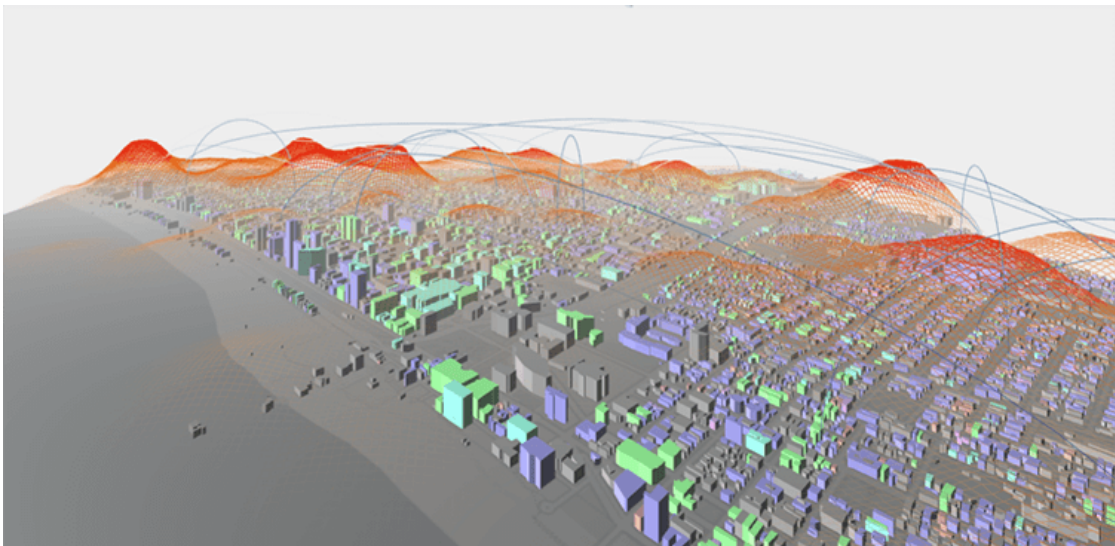
### 2.3.3 Geospazialità

Tra i diversi ambiti applicativi della visualizzazione immersiva, di particolare rilievo è quello dei dati geospaziali. La struttura di questo tipo di dati sembra infatti prestarsi in modo naturale a una mappatura in ambiente tridimensionale navigabile.

A scopo esemplificativo, in Figura 2.14 è riportata una delle demo progettate da BadVR [13, 14].



**Figura 2.13:** Un grafico a barre diventa una parete da arrampicata in LookVR [10, 12].



**Figura 2.14:** *Mirror City* di BadVR [13, 14].

### 2.3.4 Caso studio: DataGarden

In questa sezione viene presentato DataGarden, un progetto di visualizzazione dati in ambiente virtuale immersivo del 2023 [15]. In particolare, ne vengono descritte alcune funzionalità chiave e indicate le mancanze più evidenti.

#### Metafore rappresentative

Una linea di ricerca emergente esplora la costruzione di un linguaggio visivo più empatico e metaforico. In questo contesto si colloca DataGarden [15].

Questo sistema rappresenta i dati relativi a comunità di persone sotto forma di un giardino virtuale, dove ogni individuo è codificato come fiore o albero, e attributi come personalità, abitudini o opinioni determinano caratteristiche visive quali colore, numero di nuvole, tipo di vegetazione [15].



**Figura 2.15:** Ambiente virtuale di DataGarden [15, Fig. 6.b].

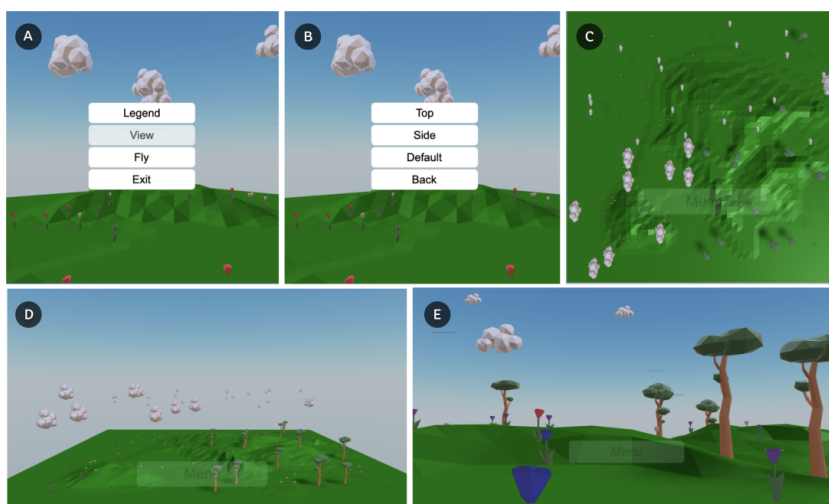
L'obiettivo non è l'ottimizzazione percettiva in senso classico, ma la creazione di rappresentazioni che favoriscano l'empatia verso le persone dietro i dati sfruttando metafore naturali e interazioni motorie come camminare, toccare e osservare da vicino.

In questo senso, DataGarden promuove un coinvolgimento emotivo e una lettura più narrativa del dataset.

## Cambio visuale

DataGarden offre un esempio di implementazione del passaggio da FPS a TPS mostrato nei prototipi delle Figure 2.8 - 2.9.

Una prima iterazione di DataGarden permetteva infatti all'utente di cambiare visuale scegliendo tra vista dall'alto, vista laterale e vista di default tramite un apposito menu, mostrato in Figura 2.16 [15].



**Figura 2.16:** Menu di selezione della visuale nella prima versione di DataGarden [15, Fig. 4.a - 4.e].

La seconda versione di DataGarden implementa invece la modalità *Fly*, mostrata in Figura 2.17 come alternativa alla modalità *Walk* [15].

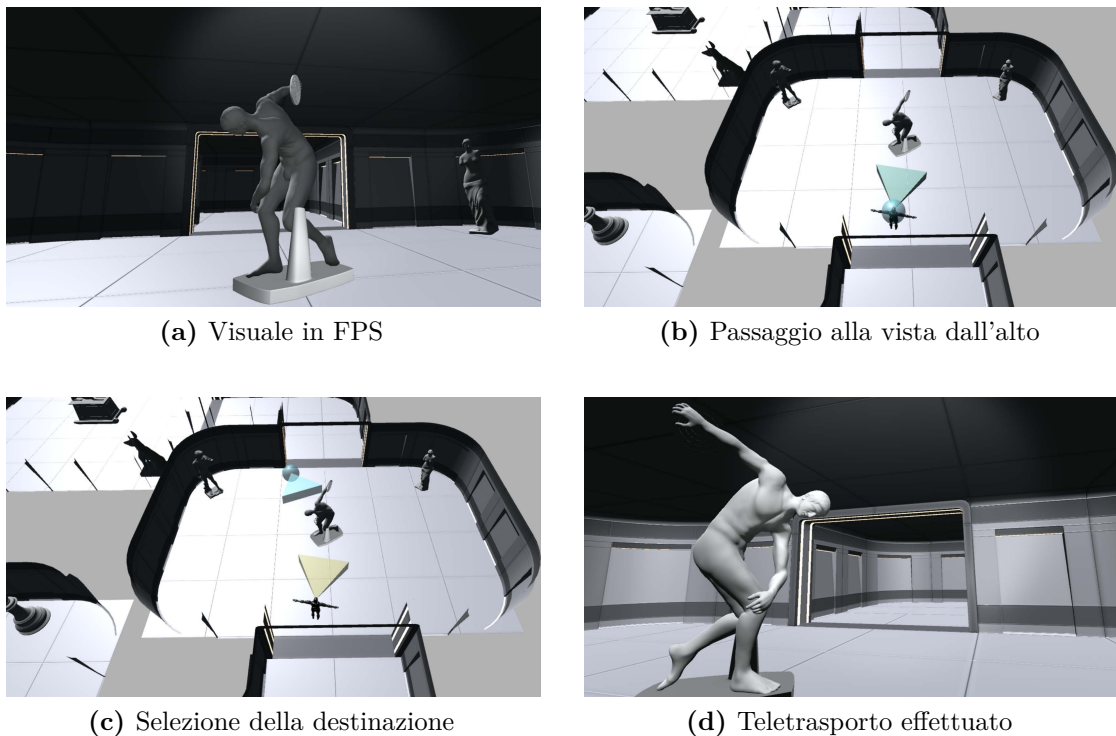


**Figura 2.17:** *Fly mode* nella seconda versione di DataGarden [15, Fig. 6.a].

### 2.3.5 Teletrasporto

È utile notare come spesso, nelle opere presenti in letteratura, il passaggio a una visuale dall'alto sia strettamente collegato alla funzionalità di *teleporting*.

Un esempio di implementazione di questa funzionalità è rappresentato dalla MOT (Map Overview Teleport). Si tratta, appunto, del passaggio temporaneo a una vista dall'alto, con conseguente selezione di una destinazione tramite cursore e teletrasporto nel punto indicato [16].

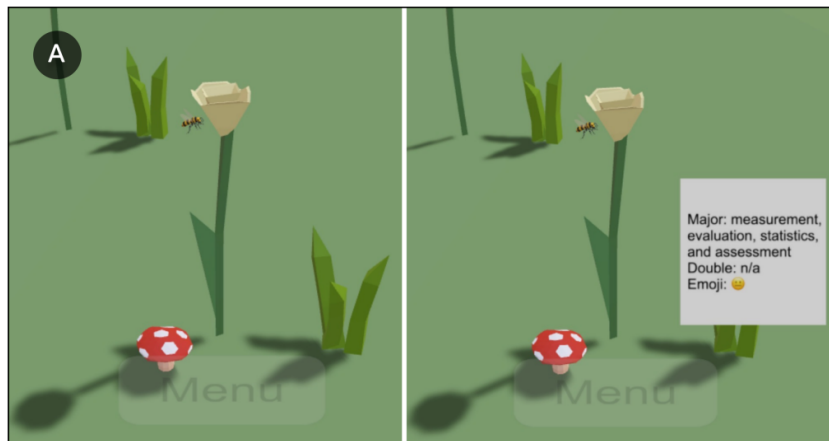


**Figura 2.18:** Le quattro fasi della MOT [16, Fig. 3.a - 3.d].

Pur offrendo la possibilità di cambiare vista e passare a un'inquadratura dall'alto dell'ambiente virtuale, DataGarden non implementa il *teleporting*.

#### Interazione con gli elementi della scena

Una delle funzionalità interattive di DataGarden consiste nella possibilità di ricevere informazioni dettagliate riguardo al fiore o all'albero che viene puntato dal controller. Le informazioni, come mostra la Figura 2.19, appaiono in forma scritta all'interno di un riquadro fluttuante in corrispondenza dell'elemento puntato [15].



**Figura 2.19:** Informazioni dettagliate mostrate in seguito all'hover in DataGarden [15, Fig. 8.a].

## Suono

Infine, sebbene il progetto DataGarden accenni al potenziale utilizzo del suono per arricchire l'esperienza sensoriale, tale componente viene discussa esclusivamente come possibile sviluppo futuro e non risulta ancora integrata nell'implementazione descritta nel lavoro originale [15].

# Capitolo 3

## Concept

In questa prima iterazione l'obiettivo principale dei test, descritti in modo approfondito nel Capitolo 5, è valutare la leggibilità della visualizzazione dei dati e il carico cognitivo associato alle codifiche adottate. Introdurre fin da subito il visore e i controller avrebbe aggiunto un ulteriore livello di complessità, legato alla familiarità limitata degli utenti con la VR, con conseguente rischio di compromettere la purezza delle misurazioni.

È per questo motivo che si è scelto di implementare un prototipo desktop, pensato specificamente come strumento per la fase di testing, al fine di raccogliere dati quanto più possibile isolati rispetto alle difficoltà di interazione tipiche della realtà virtuale, concentrando l'analisi sul contenuto dell'applicazione e sulle scelte di visualizzazione.

### 3.1 Esperienza utente

L'esperienza offerta dall'applicazione si può dividere in due macrofasi, di seguito definite come *fase dell'utente* e *fase del fruitore*:

- La user experience comincia quando l'utente carica il proprio dataset nell'applicazione e procede a impostare e personalizzare la visualizzazione dei dati in esso contenuto tramite le funzionalità dell'app, elencate nella Sezione 3.3.2 e spiegate in modo approfondito nella Sezione 4.2.2.
- Completata la fase di setup, l'utente diventa il fruitore della visualizzazione, ed è dunque libero di esplorare la stanza che ha configurato, navigando lo spazio immersivo e interagendo con gli NPC.

È importante notare che le figure di *utente* e *fruitore* non coincidono necessariamente. Nella Sezione 3.4 vengono proposte alcune possibili combinazioni di coppie utente-fruitore, costruite a partire dai target individuati per l'applicazione.

## 3.2 Analisi requisiti

A partire dall'idea progettuale descritta nella Sezione 1.3, sono emersi tre requisiti fondamentali, ritenuti tanto centrali per il concept al punto che si è deciso di incorporarli fin dalla fase prototipale:

- **Performance:** Il primo requisito riguarda la possibilità di gestire un elevato numero di NPCs mantenendo un carico computazionale adeguato.
- **Flessibilità:** In secondo luogo, lo strumento deve essere in grado di adattarsi a dataset eterogenei, evitando che la logica di visualizzazione sia rigidamente vincolata a un singolo scenario d'uso.
- **Formato di input:** Infine, si è reso necessario definire in modo esplicito il formato dei dati accettati dall'applicazione.

In particolare, per quanto riguarda il requisito sulla performance, studi recenti mostrano che frame rate elevati in VR hanno un impatto diretto sulla *simulator sickness* percepita dagli utenti. Nello specifico, pare che 120 fps rappresentino una soglia importante, oltre la quale i sintomi della *simulator sickness*, quali nausea e disorientamento, risultano significativamente ridotti rispetto a valori di frame rate inferiori [17].

Tuttavia, nella pratica progettuale corrente, la maggior parte dei visori commerciali opera in un intervallo compreso tra 72 e 90 Hz, con 90 Hz comunemente indicati come valore minimo confortevole e 72 Hz considerato ancora accettabile per esperienze a ritmo relativamente lento [18]. Nel contesto di questo lavoro, caratterizzato da scenari di esplorazione dati non particolarmente dinamici, si assume pertanto il range 72-90 fps come compromesso progettuale tra comfort dell'utente e vincoli hardware.

Il frame rate target del progetto è dunque stato fissato nell'intervallo compreso tra 72 e 90 fps.

## 3.3 Scelte progettuali

Per l'applicazione realizzata è stato scelto il nome *inPersona*, che richiama l'idea fondante del progetto, ovvero di restituire la dimensione umana ai dati.

Il prototipo desktop sviluppato rispetta i requisiti individuati nella sezione precedente e integra le codifiche visive progettate per la rappresentazione dei dataset. Nelle sezioni seguenti verrà distinta, ove rilevante, la parte di implementazione pensata specificamente per il prototipo desktop dalle soluzioni già concepite come VR-ready, evidenziando quali elementi richiedono un adattamento in fase di porting su visore.

### 3.3.1 Scelte imposte dai requisiti

Per garantire un carico computazionale basso pur gestendo un numero elevato di NPCs, si è scelto di adottare l'architettura ECS (Entity Component System) di Unity, nell'ambito del framework DOTS (Data-Oriented Technology Stack). Ulteriori giustificazioni di questa scelta vengono fornite nella Sezione 4.1, insieme a una breve spiegazione del funzionamento di DOTS.

Riguardo al requisito di flessibilità dello strumento, l'ambiente virtuale è stato progettato in modo parametrico: le dimensioni della stanza sono infatti determinate in base al massimo numero di NPCs richiesto dalla visualizzazione impostata dall'utente.

A questa impostazione si affiancano ulteriori scelte orientate all'adattabilità e alla personalizzazione dell'esperienza, come la possibilità per l'utente di caricare i propri asset (modelli 3D e suoni), e l'aggiornamento automatico delle indicazioni presenti nella stanza sulla base dei valori contenuti nel dataset caricato. Tali aspetti verranno descritti più nel dettaglio nella Sezione 4.2.2.

Infine, per quanto concerne la definizione del formato dei dataset accettato dall'applicazione, al fine dello sviluppo del prototipo, la scelta è ricaduta sui file CSV. Questi devono essere strutturati in modo che ogni riga rappresenti un singolo individuo, mentre ogni colonna corrisponda a un attributo associato a tale individuo. È dunque il caso di survey, questionari, record anagrafici, censimenti o elenchi analoghi.

Alla luce di quest'ultima scelta si può dunque definire l'approccio utilizzato nel progetto come una mappatura 1:1 tra le righe del dataset e gli NPCs presenti nella scena. Tale impostazione va ad avvalorare ulteriormente le necessità prestazionali discusse nella Sezione 3.2.

D'altronde, si può affermare che la mappatura 1:1 costituisca un elemento centrale del concept, in quanto deriva dalla motivazione di fondo dell'intero progetto, già descritta nella Sezione 1.3, di voler restituire la componente umana ai dati sulle persone.

### 3.3.2 Visualizzazione dati

Prima di descrivere in dettaglio le visualizzazioni implementate, è utile richiamare brevemente il confronto con DataGarden, presentato nello stato dell'arte.

Da DataGarden è stata ripresa l'idea di una mappatura 1:1 dei dati sulle caratteristiche delle entità presenti nell'ambiente e, di conseguenza, la costruzione di un linguaggio visivo coerente e riconoscibile, la cui interpretazione è supportata da legende dedicate.

La principale differenza risiede nella scelta delle metafore: mentre DataGarden rappresenta i dati attraverso elementi naturalistici, quali fiori, alberi e nuvole presenti in un giardino, *inPersona* mantiene la mappatura di più variabili su una singola entità, ma rinuncia alla metafora, adottando invece modelli umanoidi espliciti come rappresentazione dei dati relativi alle persone.



**Figura 3.1:** Ambiente virtuale di DataGarden [15]



**Figura 3.2:** Ambiente virtuale di *inPersona*

Venendo ora alle codifiche individuate per la visualizzazione dati in VR, *inPersona* permette di mappare i valori presenti nel dataset caricato sulle seguenti caratteristiche degli NPCs:

- **Colore**
- **Livello di riempimento**
- **Posizione nello spazio**, lungo l'asse x (sinistra-destra) e y (profondità) della stanza
- **Suono emesso**
- **Pitch del suono emesso**

- **Frase pronunciata**
- **Modello 3D**
- **Filtro**, rappresentato dalla stanza stessa, che, al variare del filtro, si popola con gli NPC's corrispondenti al valore selezionato.

Il sistema permette l'applicazione simultanea di tutte le mappature sopra elencate. Per la spiegazione dei setup disponibili e dei dati input ideali per ognuna delle funzionalità sopra elencate, si rimanda alla Sezione 4.2.2.



**Figura 3.3:** Esempio di NPC's caratterizzati da colori, modelli 3D e livelli di riempimento diversi.



**Figura 3.4:** Gli NPC's occupano posizioni diverse nella stanza in base ai valori scelti per gli assi x e y.



(a) Filtro applicato: "Degree Major = Computer Science"



(b) Filtro applicato: "Degree Major = Data Science"

**Figura 3.5:** Effetto del cambiamento del filtro.

### 3.3.3 Navigazione e interazione

#### Esplorazione

Per quanto riguarda la possibilità di esplorare i dati camminando all'interno dell'ambiente immersivo che li ospita, *inPersona* si colloca sullo stesso livello di DataGarden [15] e degli altri strumenti mostrati nello stato dell'arte, come DataVizVR [11] e LookVR [12].

In particolare, in *inPersona* la navigazione dell'utente nello spazio è funzionale alla fruizione dei dati mappati sul suono e/o sul pitch sonoro. Il suono emesso dai singoli NPCs è infatti udibile solo quando ci si trova in prossimità di essi.

Allo stesso modo, solo trovandosi vicino a un NPC l'utente sarà in grado di interagire con esso per ascoltare la frase che ha da dire, poiché anch'essa è udibile solo in prossimità del personaggio che la pronuncia.

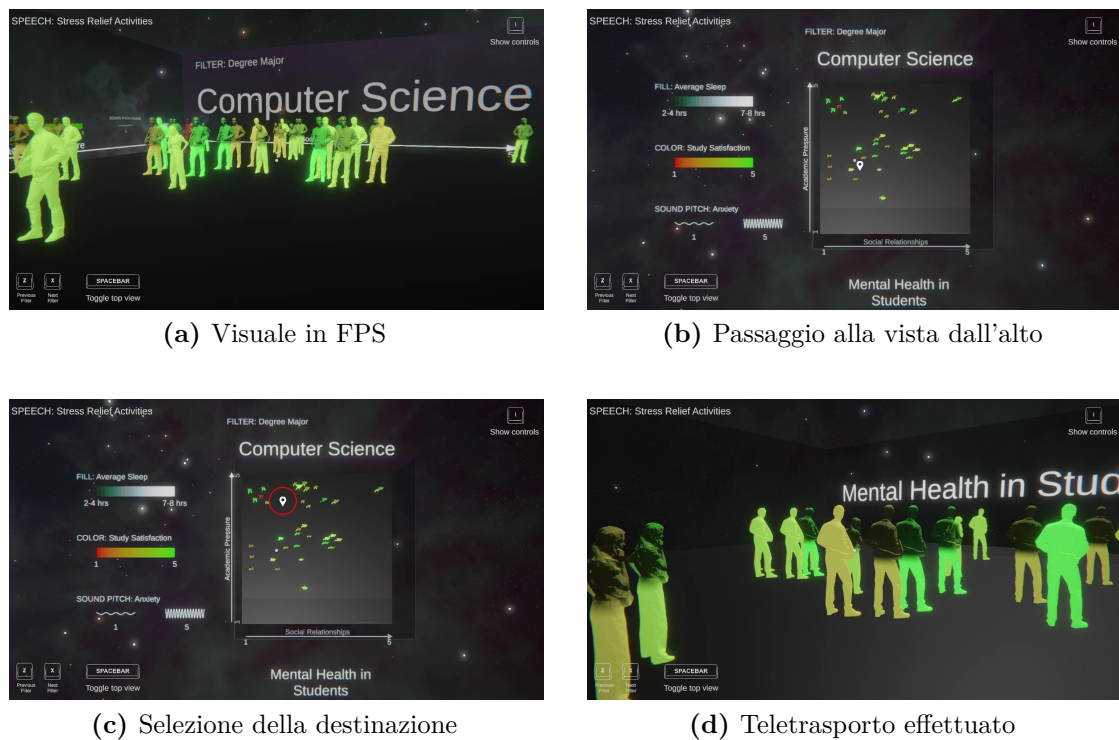
A differenza di quanto avviene in DataGarden [15], questa versione desktop di *inPersona* non implementa alcuna funzionalità di hover che consenta di visualizzare le informazioni associate ai singoli NPCs. Tale scelta è coerente con la decisione di non utilizzare il visore VR e riflette lo stesso principio metodologico: poiché l'obiettivo principale del progetto è indagare la comprensibilità e l'intuitività delle codifiche grafiche individuate, si è preferito evitare qualsiasi elemento interattivo che potesse compromettere la validità della misurazione, fornendo all'utente indicazioni o "risposte" esplicite durante l'esplorazione.

## Cambio visuale

La logica di cambio visuale implementata in *inPersona* combina le caratteristiche del prototipo in Figura 2.10 e le funzionalità della MOT [16].

Il risultato è il passaggio dalla *Room view* a una *Top View* interattiva, che permette all'utente di visualizzare la stanza come un grafico 2D tradizionale e di teletrasportarsi da un punto all'altro di essa.

Nel prototipo desktop realizzato, la selezione della destinazione avviene per mezzo di una classica interazione *punta e clicca*, durante la quale il cursore assume la forma di una *location pin*, come mostrato in Figura 3.6c.



**Figura 3.6:** Le quattro fasi della MOT [16] in *inPersona*.

## Comandi e feedback

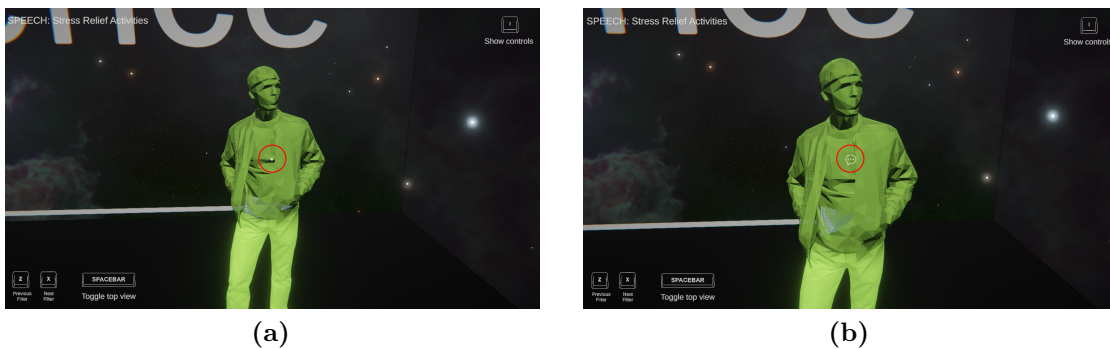
Per la realizzazione del prototipo desktop di *inPersona* è stata adottata la modalità di interazione tipica dei giochi in FPS. Questa scelta è comune nei prototipi dei progetti VR, poiché consente di simulare in modo efficace la libertà di movimento e di rotazione tipiche della realtà virtuale.

In questo contesto, il cursore rimane fisso al centro dello schermo e l'utente può orientare la visuale con il mouse e muoversi tramite i tasti WASD (o le frecce direzionali). Tutti i comandi del prototipo desktop di *inPersona* sono riportati nella Tabella 3.1.

Room View	
Camminare	WASD
Ruotare visuale	Movimento del mouse
Parlare	Click
Top View	
Spostare inquadratura	WASD
Zoom	Rotella del mouse
Scegliere destinazione	Movimento del mouse
Effettuare teletrasporto	Click
Room e Top View	
Mostrare finestra comandi	I
Valore filtro precedente	Z
Valore filtro successivo	X
Cambiare view	Barra spaziatrice

**Tabella 3.1:** Comandi

Inoltre, per quanto riguarda i feedback, nel prototipo è stato implementato il cambio cursore. Per suggerire all'utente la possibilità di *parlare* con un NPC, il cursore di default diventa infatti un fumetto, come mostrato in Figura 3.7.



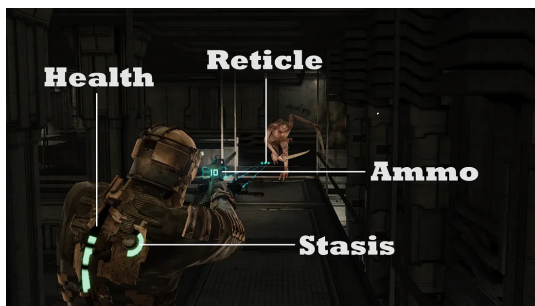
**Figura 3.7:** In prossimità di un NPC, il cursore diventa un fumetto per suggerire l'interazione di *speech* all'utente.

### 3.3.4 Interfacce

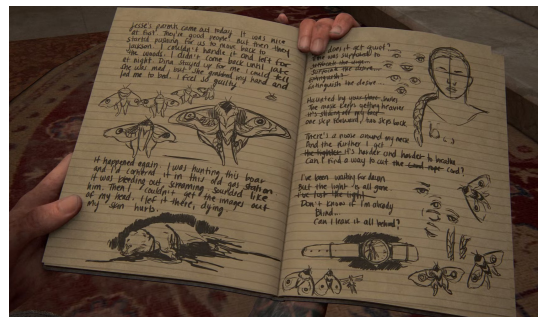
Per il posizionamento delle legende si è scelto di seguire un duplice approccio. La maggior parte delle informazioni è stata rappresentata tramite interfacce diegetiche, mentre l'indicazione circa la natura della frase pronunciata degli NPC e i comandi sono riportati in un'interfaccia non diegetica.

#### Interfacce diegetiche

Le interfacce diegetiche sono elementi informativi integrati nel mondo virtuale, ad esempio orologi, documenti o dispositivi tenuti in mano dal personaggio, che non interrompono l'esperienza immersiva [19].

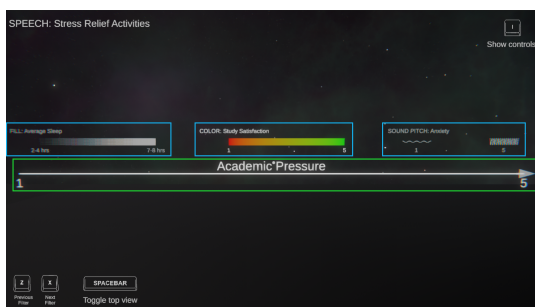


**Figura 3.8:** In *Dead Space*, salute, munizioni e altre informazioni sono indicate sul corpo del personaggio [20, 21].

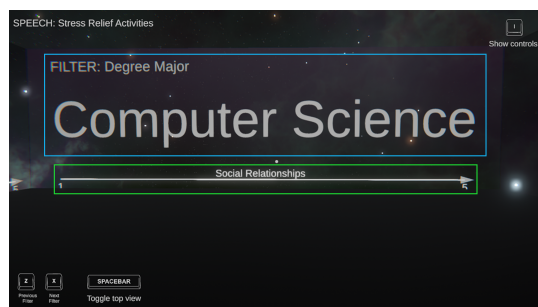


**Figura 3.9:** Diario con i progressi di gioco in *The Last of Us* [22].

In *inPersona*, le legende sono distribuite sui muri della stanza o sui pavimenti, come mostrato nelle Figure 3.10 - 3.14.



**Figura 3.10:** Legende di livello di riempimento, gradiente di colore, pitch sonoro (in azzurro) e asse y (in verde).



**Figura 3.11:** Legenda del Filtro (in azzurro) e dell'asse x (in verde).

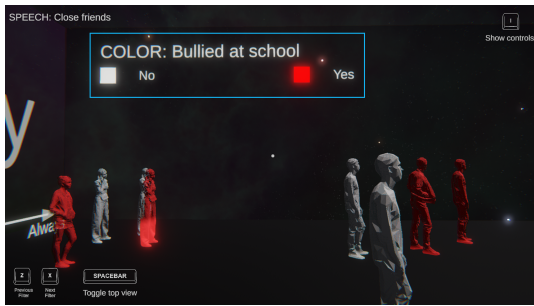


Figura 3.12: Legenda dei colori.

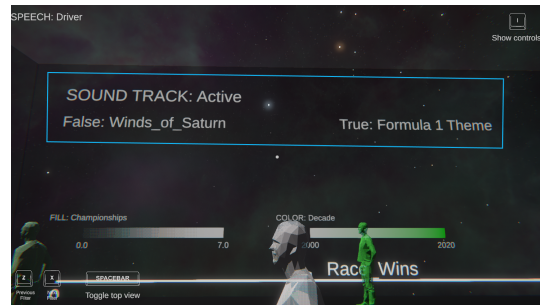


Figura 3.13: Legenda dei suoni.

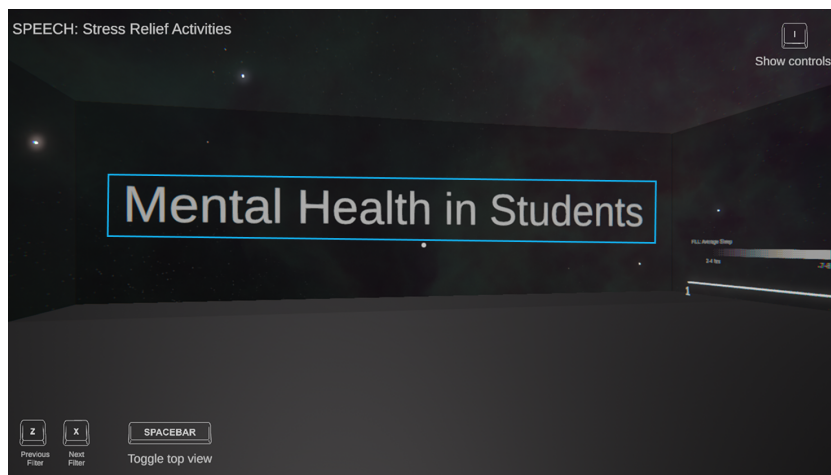


Figura 3.14: Titolo del dataset visualizzato.

### Interfacce non diegetiche

Le interfacce non diegetiche sono invece costituite da componenti 2D sovrapposti alla scena, come HUD (Heads-Up Display) e pannelli informativi, che forniscono informazioni senza appartenere al mondo diegetico [19].

Le interfacce non diegetiche, implementate esclusivamente per il prototipo desktop di *inPersona*, collocano l'informazione relativa alle frasi insieme ai controlli, all'interno di un'interfaccia non diegetica.

Questa scelta è motivata dal fatto che lo *speech* viene reso attraverso un feedback del cursore, elemento anch'esso non diegetico. Di conseguenza, è parso coerente mantenere le informazioni sul valore mappato tramite *speech* sempre visibili sullo schermo, in continuità con la modalità di interazione adottata.



Figura 3.15: L'HUD di *The Witcher 3: Wild Hunt* mostra una serie di informazioni, tra cui le *command hints* (in azzurro) [23].



Figura 3.16: UI di *inPersona*: comandi (in azzurro) e legenda dello *speech* (in verde).

### 3.3.5 Stile

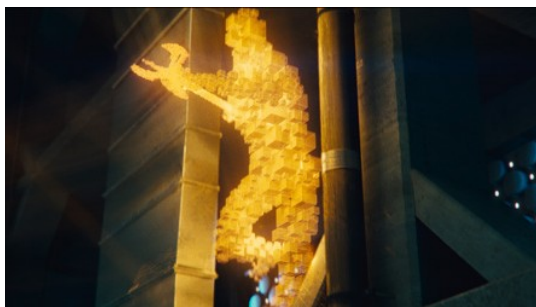
Per lo stile dei personaggi, dati i requisiti in termini di performance, già discussi nella Sezione 3.2, si è optato per un design *low poly*. Questa scelta è supportata da studi che dimostrano come personaggi stilizzati possono generare empatia e simpatia comparabili a quelle generate da personaggi realistici [24].

Come reference per l'estetica minimal dei personaggi è stato scelto *SUPERHOT VR* [25].



**Figura 3.17:** Personaggi *low poly* in *SUPERHOT VR* [25].

Per la rappresentazione delle entità presenti nella scena si è dunque scelto di utilizzare ologrammi umanoidi. Questa soluzione consente di giustificare narrativamente una delle mappature implementate, ovvero il livello di riempimento. Le references per gli "ologrammi che *spawnano* in una stanza" provengono dal film *The Hunger Games: Catching Fire* [26].



(a)



(b)

**Figura 3.18:** Ologrammi nemici durante una scena di allenamento in *The Hunger Games: Catching Fire* [26].

La presenza degli ologrammi ha guidato l'impostazione dello stile generale della stanza e dell'atmosfera. Si è infatti optato per la creazione di un ambiente etereo e rilassante, che non costituisca una fonte di distrazione per l'utente senza tuttavia risultare neutro o sterile.

La stanza appare dunque come uno spazio sospeso nella galassia, accompagnato da una traccia musicale costante e coerente con il tema scelto.

## 3.4 Target utenti

Fin dalla definizione dell'idea fondante, è apparso chiaro che il contesto ideale per *inPersona* sia quello museale, che rappresenta infatti il target primario dello strumento.

D'altro canto la dinamica utente-fruitore, a cui si accennava nella Sezione 3.1, può essere ritrovata anche in altri contesti, uno fra tutti quello scolastico.

Il target secondario è infatti costituito dai docenti delle scuole secondarie che scelgono di usare *inPersona* come supporto alla didattica, sia per l'insegnamento tradizionale, che per sensibilizzare i giovani su tematiche specifiche. Un esempio di quest'ultimo uso è mostrato nella Figura 3.19, che riporta la visualizzazione di un dataset sul bullismo.



**Figura 3.19:** Dataset "Bullying in schools" visualizzato su *inPersona*.

Tra i target secondari individuati per *inPersona* si collocano anche strutture ricettive e agenzie organizzatrici di viaggi o eventi. Lo strumento può infatti essere configurato come un visualizzatore di recensioni, con le valutazioni degli

utenti mappate sulle caratteristiche degli ologrammi ed eventuali commenti liberi direttamente pronunciati da essi.

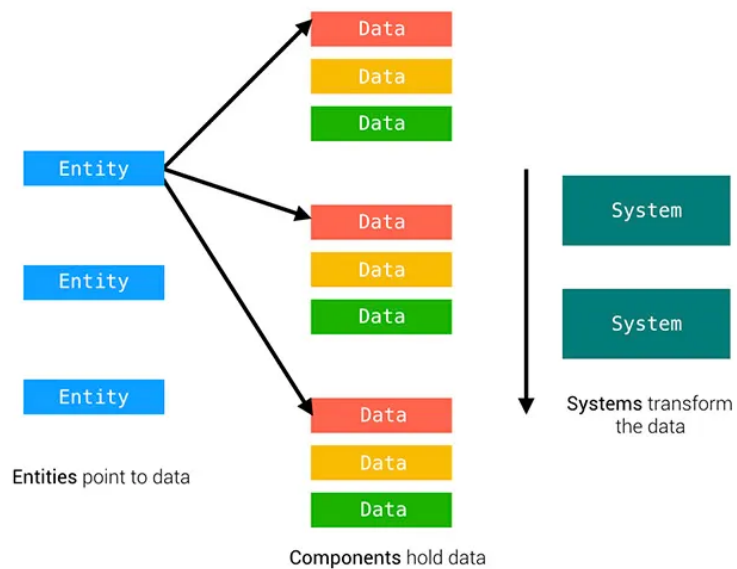
# Capitolo 4

## Prototipazione

### 4.1 Paradigma ECS e Unity DOTS

#### 4.1.1 Introduzione a ECS con DOTS

L'architettura ECS (Entity Component System) suddivide i dati, il comportamento e il rendering in elementi distinti.



**Figura 4.1:** Schema esplicativo dell'architettura ECS [27].

Invece del paradigma object-oriented tradizionale, in cui gli oggetti contengono sia dati che logica, si segue infatti un approccio data-oriented così strutturato:

- Le *Entities* fungono da semplici identificatori per gli oggetti di gioco.
- I *Components* contengono i dati puri (che nel caso di *inPersona* sono posizione, colore, livello di riempimento...).
- I *Systems* gestiscono la logica e l'elaborazione degli specifici componenti.

Tale separazione tra dati e comportamento ottimizza l'uso delle risorse hardware, migliorando le prestazioni tramite un uso più efficiente della memoria e aumentando la scalabilità per mezzo del processing parallelo multi-core [27, 28].

Unity DOTS (Data-Oriented Technology Stack) è il framework di Unity che permette di programmare seguendo il paradigma ECS [28]. Tra le risorse offerte da Unity DOTS, le seguenti sono di particolare rilevanza:

- **C# Job System:** è il framework di Unity che permette di eseguire codice in parallelo su più core della CPU, distribuendo i Jobs su diversi worker threads invece di eseguirli tutti sul main thread.
- **Burst Compiler:** è il compilatore di Unity che traduce il codice C# usato nei Jobs in codice a basso livello altamente ottimizzato.

### 4.1.2 Approccio scelto e strumenti usati

Come base da cui partire per lo sviluppo del prototipo [29], sono stati usati gli script mostrati nei corsi di Hugo Cardoso (Code Monkey) [30], di cui è stato adottato lo stesso approccio non puramente ECS. Infatti, laddove non è stato necessario sfruttare i vantaggi offerti dal paradigma data-oriented, si è deciso di implementare i comportamenti tramite MonoBehaviour tradizionali [31].

Gli strumenti adottati per l'implementazione di *inPersona* sono stati la versione 6000.1.2f1 di Unity e l'IDE Visual Studio Code con l'estensione di GitHub Copilot. L'intelligenza artificiale è infatti stata adoperata - in combinazione con la documentazione ufficiale, i tutorial già menzionati e il forum *Discussion* di Unity - come supporto alla scrittura del codice e alla risoluzione degli errori.

### 4.1.3 Prestazioni

Le Figure 4.2 - 4.3 mettono a confronto i dati raccolti dal Profiler di Unity durante due *run* di *inPersona*. A parità di scena, le prestazioni aumentano notevolmente

quando si utilizza il Job System per l'esecuzione parallela dello *Spawner System*, ovvero lo script che colloca le entità nella stanza.

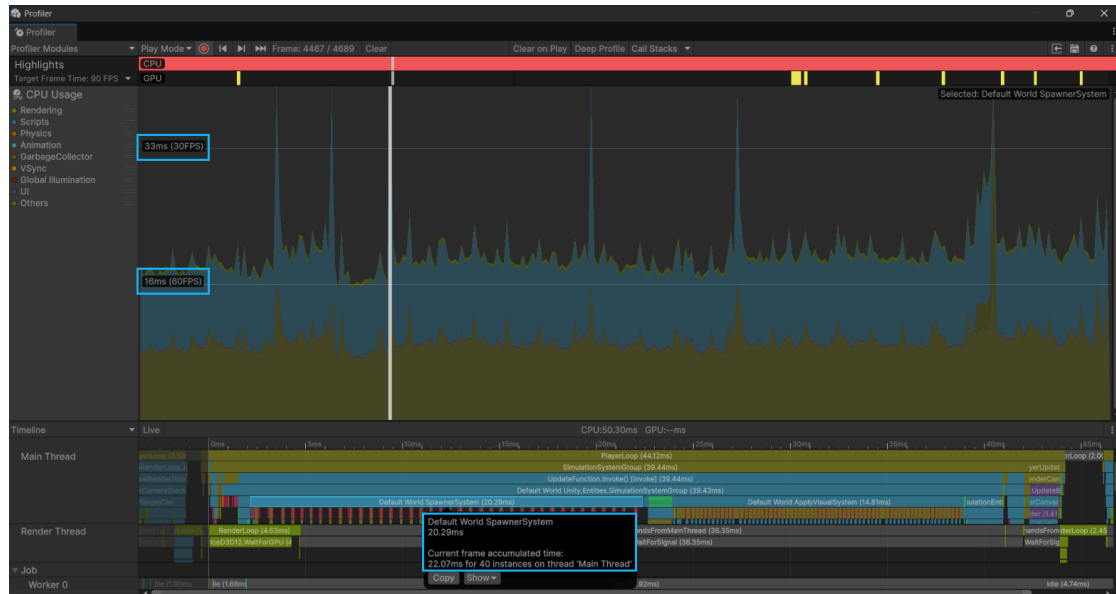


Figura 4.2: Senza Job System lo *Spawner System* occupa 22ms sul main thread.

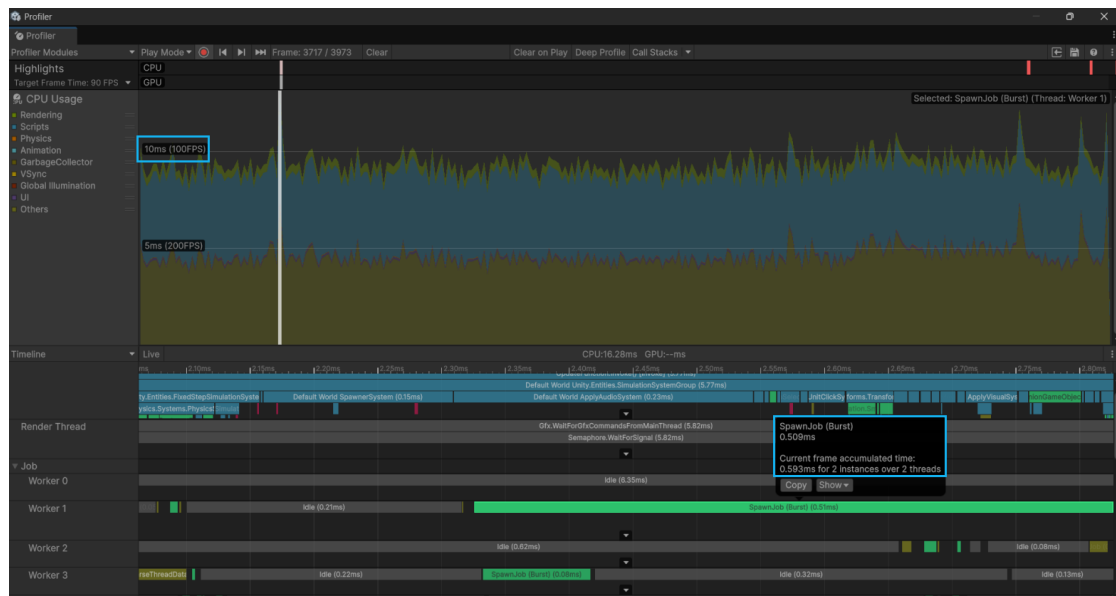


Figura 4.3: Con il Job System lo *spawnJob* viene parallelizzato su due worker threads e occupa in totale 0.593ms.

Si noti come, integrando l'uso dei Job System, viene raggiunto il range target di 72-90 fps fissato nella definizione dei requisiti effettuata nel Capitolo 3.

## 4.2 Implementazione

In questa sezione verranno trattati i file più importanti che compongono il progetto, proseguendo secondo il flusso tracciato in Figura 4.4, perciò da input a output.

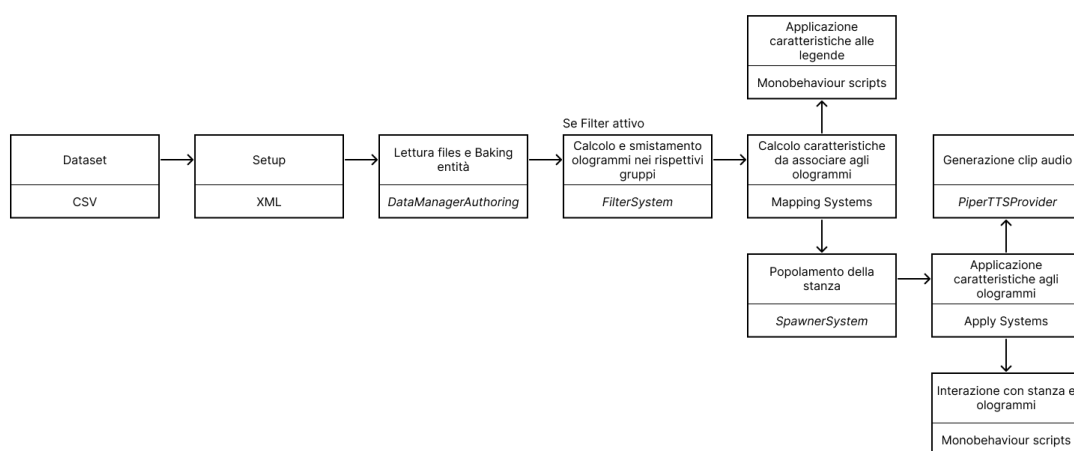


Figura 4.4: Flusso logico di *inPersona*.

Da qui in avanti, nella spiegazione dell'implementazione, quasi tutti i riferimenti saranno al caso d'uso dei test descritti nel Capitolo 5: si tratta del dataset sulla salute mentale degli studenti già mostrato nelle Figure del Capitolo 3.

### 4.2.1 File di input

Prima di trattare nel dettaglio gli script che compongono il processo di elaborazione dati e generazione delle visualizzazioni, è necessario illustrare il formato dei file che il prototipo accetta come input validi.

Questi corrispondono a file CSV in cui il separatore è la virgola, ogni riga rappresenta una persona diversa e ogni colonna un attributo individuale, come già anticipato

nei capitoli precedenti. Non sono dunque supportati formati con dati aggregati, come somme e percentuali.

Le Figure 4.5 - 4.6 illustrano un esempio di configurazione ideale del dataset rispetto ad una meno ottimale. Il prototipo gestisce meglio colonne con valori che rappresentano categorie, anziché *colonne-categoria* con valori booleani che ne indicano l'appartenenza.

Age	Gender	Close friends	Bullied at school	Physical fighting	Cyber bullied	Felt lonely	Weight
12	Male	I have more than two friends	Yes	2 or 3 times	No	I never feel lonely	Overweight
12	Female	I have two close friends	Yes	0 times	No	I feel lonely most of the time	Overweight
12	Male	I have two close friends	Yes	0 times	No	I rarely feel lonely	Overweight
13	Female	I don't have any close friend	No	0 times	No	I always feel lonely	Underweight
13	Female	I have more than two friends	No	0 times	No	I always feel lonely	Normal weight
13	Female	I have more than two friends	No	0 times	No	I feel lonely most of the time	Normal weight
13	Female	I have more than two friends	No	0 times	No	I never feel lonely	Normal weight
13	Female	I have more than two friends	No	0 times	No	I never feel lonely	Overweight
13	Male	I have more than two friends	No	0 times	No	I never feel lonely	Normal weight
13	Male	I have more than two friends	No	0 times	No	I never feel lonely	Obese
13	Male	I have more than two friends	No	0 times	No	I never feel lonely	Overweight
13	Male	I have more than two friends	No	1 time	No	I never feel lonely	Normal weight
13	Male	I have more than two friends	No	2 or 3 times	No	I never feel lonely	Normal weight
13	Male	I have more than two friends	No	4 or 5 times	No	I never feel lonely	Obese
13	Male	I have more than two friends	No	4 or 5 times	No	I never feel lonely	Overweight

**Figura 4.5:** Configurazione ideale: valori diversi in un'unica colonna.

Age	Gender	Close friends	Bullied at school	Physical fighting	Cyber bullied	Felt lonely	Were_underweight	Were_overweight	Were_obese
12	Male	I have more than two friends	Yes	2 or 3 times	No	I never feel lonely	No	Yes	No
12	Female	I have two close friends	Yes	0 times	No	I feel lonely most of the time	No	Yes	No
12	Male	I have two close friends	Yes	0 times	No	I rarely feel lonely	No	Yes	No
13	Female	I have more than two friends	No	0 times	No	I always feel lonely	No	No	No
13	Female	I have more than two friends	No	0 times	No	I feel lonely most of the time	No	No	No
13	Female	I have more than two friends	No	0 times	No	I never feel lonely	No	No	No
13	Female	I have more than two friends	No	0 times	No	I never feel lonely	No	Yes	No
13	Male	I have more than two friends	No	0 times	No	I never feel lonely	No	No	No
13	Male	I have more than two friends	No	1 time	No	I never feel lonely	No	No	No
13	Male	I have more than two friends	No	2 or 3 times	No	I never feel lonely	No	No	No
13	Male	I have more than two friends	No	0 times	No	I never feel lonely	No	Yes	No
13	Male	I have more than two friends	No	4 or 5 times	No	I never feel lonely	No	Yes	No
13	Male	I have more than two friends	No	6 or 7 times	No	I never feel lonely	No	Yes	No
13	Male	I have more than two friends	No	0 times	No	I never feel lonely	No	Yes	Yes
13	Male	I have more than two friends	No	4 or 5 times	No	I never feel lonely	No	Yes	Yes

**Figura 4.6:** Configurazione non ideale: valori booleani su colonne diverse.

Si noti che per quest'ultimo esempio è stato usato un dataset diverso rispetto al caso d'uso in esame.

Infine, poiché visualizzati direttamente nell'ambiente VR, gli *header* delle colonne del dataset originale (Figura 4.7) sono stati riformulati come frasi complete, ovvero prive di abbreviazioni, underscores o altri simboli.

Gender	Age	Degree Major	Average Sleep	Study Satisfaction	Academic Pressure	Social Relationships	Anxiety	Stress Relief Activities
Female	17	Computer Science	4-6 hrs	4	4	3	4	My main stress relief is Religious Activities.
Female	17	Computer Science	4-6 hrs	4	5	1	5	My main stress relief is Sleep.
Female	18	Data Science	4-6 hrs	1	5	2	5	My main stress relief is Creative Outlets.
Female	19	Computer Science	4-6 hrs	3	5	3	3	I reduce stress by Social Connections and Sleep.
Female	19	Computer Science	4-6 hrs	3	5	3	5	My main stress relief is Sleep.
Female	19	Computer Science	7-8 hrs	4	3	3	4	I reduce stress by Religious Activities, Social Connections, and Sleep.
Female	19	Data Science	2-4 hrs	2	4	2	3	My main stress relief is Religious Activities.
Female	19	Data Science	4-6 hrs	4	3	2	4	I reduce stress by Religious Activities, Creative Outlets, and Social Connections.
Female	19	Data Science	4-6 hrs	4	4	1	1	I reduce stress by Religious Activities, Social Connections, and Online Entertainment.
Female	19	Data Science	4-6 hrs	4	4	4	2	My main stress relief is Sports and Fitness.
Female	19	Data Science	4-6 hrs	4	5	2	4	I reduce stress by Religious Activities and Social Connections.
Female	19	Data Science	4-6 hrs	5	4	3	4	I reduce stress by Religious Activities and Creative Outlets.
Female	19	Data Science	7-8 hrs	5	2	4	2	My main stress relief is Religious Activities.
Female	19	Data Science	7-8 hrs	5	3	3	1	I reduce stress by Religious Activities, Social Connections, and Online Entertainment.
Female	19	Data Science	7-8 hrs	5	4	2	2	My main stress relief is Creative Outlets.
Female	20	Computer Science	4-6 hrs	3	5	3	5	My main stress relief is Online Entertainment.
Female	20	Computer Science	4-6 hrs	4	5	3	5	I reduce stress by Religious Activities, Social Connections, and Online Entertainment.
Female	20	Computer Science	4-6 hrs	5	5	1	5	My main stress relief is Online Entertainment.
Female	20	Data Science	2-4 hrs	5	4	1	2	My main stress relief is Social Connections.
Female	20	Data Science	4-6 hrs	3	3	2	3	My main stress relief is Religious Activities.
Female	20	Data Science	7-8 hrs	5	4	3	2	My main stress relief is Social Connections.
Female	20	Software Engineering	4-6 hrs	2	4	1	5	I reduce stress by Online Entertainment, Outdoor Activities, and Sleep.
Female	21	Information Technology	7-8 hrs	3	3	3	5	My main stress relief is Sleep.
Female	21	Information Technology	7-8 hrs	3	4	4	4	I reduce stress by Online Entertainment and Sleep.
Female	21	Software Engineering	4-6 hrs	3	5	2	4	I reduce stress by Religious Activities, Sports and Fitness, Social Connections, Online Entertainment, and Sleep.
Female	21	Software Engineering	4-6 hrs	5	4	2	5	I reduce stress by Sports and Fitness, Creative Outlets, Social Connections, Outdoor Activities, and Sleep.
Female	21	Software Engineering	7-8 hrs	4	5	2	4	My main stress relief is Sports and Fitness.
Female	22	Information Technology	2-4 hrs	2	5	4	3	I reduce stress by Creative Outlets, Social Connections, and Online Entertainment.
Female	22	Software Engineering	2-4 hrs	5	5	1	5	I reduce stress by Religious Activities and Sleep.
Female	22	Software Engineering	4-6 hrs	2	5	1	5	My main stress relief is Sleep.
Female	22	Software Engineering	4-6 hrs	4	4	2	4	I reduce stress by Creative Outlets, Social Connections, Online Entertainment, and Sleep.
Female	22	Software Engineering	4-6 hrs	5	5	2	4	I reduce stress by Religious Activities, Sports and Fitness, and Social Connections.

**Figura 4.7:** Caso d’uso: dataset sulla salute mentale degli studenti (visualizzato sottoforma di tabella).

## 4.2.2 File di setup

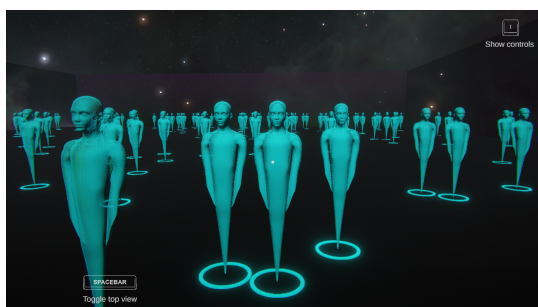
Dal momento che la fase di setup della visualizzazione non è stata oggetto di testing, si è deciso di non implementare una UI (User Interface) di configurazione: nel prototipo di *inPersona* il setup iniziale della scena avviene direttamente tramite scrittura sul file XML sottostante, andando di fatto a bypassare la UI.

Nel sopracitato file di setup, vengono definite le coppie colonna-mappatura tramite una serie di nodi `<column>`: alla stessa colonna possono essere associate più mappature, ma non viceversa.

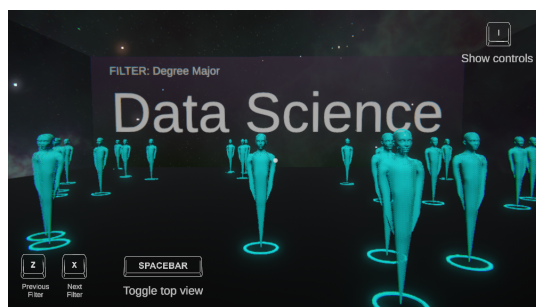
Nel seguito, alla spiegazione di ogni funzione viene affiancata, a titolo di esempio, la configurazione scelta nel caso d’uso.

## Mappature

- **Filtro:** suddivide gli ologrammi in gruppi in base ai valori distinti presenti nella colonna selezionata.



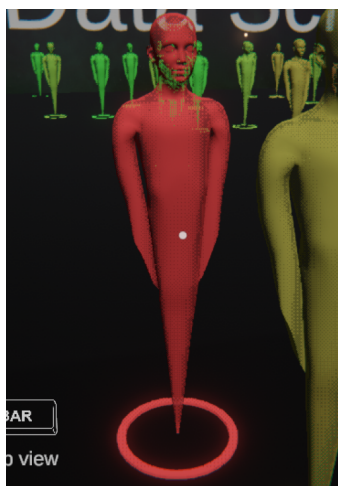
**Figura 4.8:** Filtro non attivo: tutti gli studenti sono presenti nella stanza contemporaneamente.



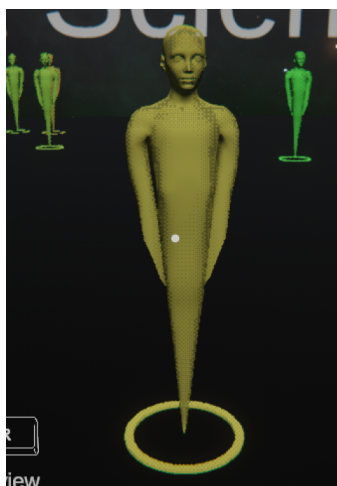
**Figura 4.9:** Filtro attivo: studenti divisi per corso di studi.

- **Colore:** mappa i valori contenuti nella colonna specificata sul colore degli ologrammi. Sono state sviluppate tre modalità:
  - **Gradiente:** specificando il colore di partenza e quello di arrivo, i valori della colonna vengono mappati sul gradiente tramite interpolazione lineare.
  - **Selezione personalizzata:** le coppie valore-colore vengono definite manualmente.
  - **Selezione casuale:** a ogni valore unico presente nella colonna viene associato un colore random.

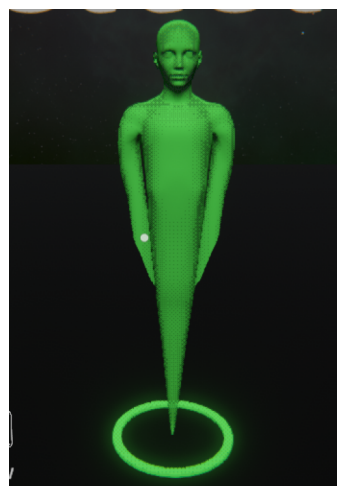
Nell'esempio proposto, è stato usato il gradiente di colore per visualizzare il grado di soddisfazione degli studenti per il proprio corso di studi.



**Figura 4.10:** Studente poco soddisfatto.



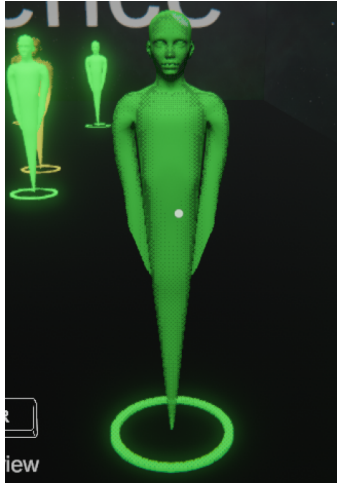
**Figura 4.11:** Studente abbastanza soddisfatto.



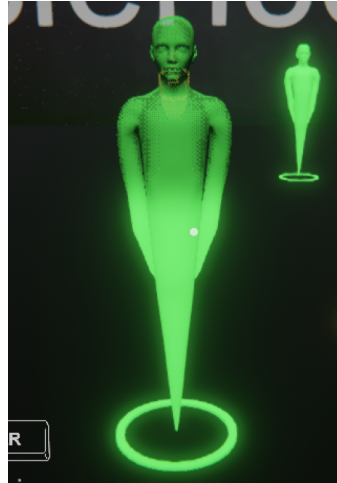
**Figura 4.12:** Studente molto soddisfatto.

- **Livello di riempimento:** mappa i valori della colonna selezionata sul livello di riempimento verticale (dal basso verso l'alto) degli ologrammi.

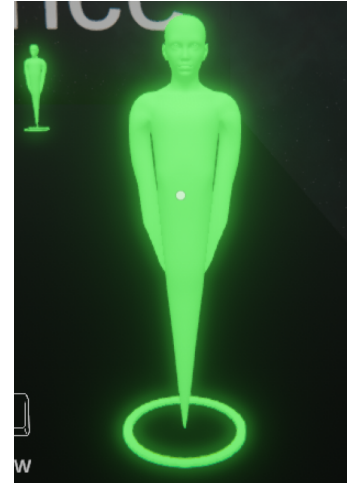
Questa funzione è stata usata per visualizzare la quantità di ore di sonno media di ogni studente.



**Figura 4.13:** Studente poco riposato.



**Figura 4.14:** Studente abbastanza riposato.



**Figura 4.15:** Studente riposato.

- **Suono emesso:** permette di definire le coppie suono-valore che associano le tracce audio (caricabili dall'utente) ai valori presenti nella colonna.

Nel caso d'uso proposto, questa opzione non è stata attivata poiché si è osservato che risulta difficile distinguere sia i suoni diversi che la variazione di pitch, di seguito descritta.

- **Pitch del suono emesso:** specificando il pitch di partenza e quello di arrivo, i valori della colonna vengono mappati tramite interpolazione lineare.

Se l'opzione *Suono* non è stata configurata, a tutti gli ologrammi viene assegnata la stessa traccia audio di default, con il livello di pitch risultante dalla mappatura.

Nel caso d'uso mostrato, si è scelto di mappare sul pitch il livello di ansia percepita dagli studenti.

- **Modello 3D:** assegna agli ologrammi i modelli 3D (caricabili dall'utente) sulla base di condizioni multi-colonna.

Nell'esempio mostrato finora, i modelli 3D sono stati assegnati in base al genere degli studenti.

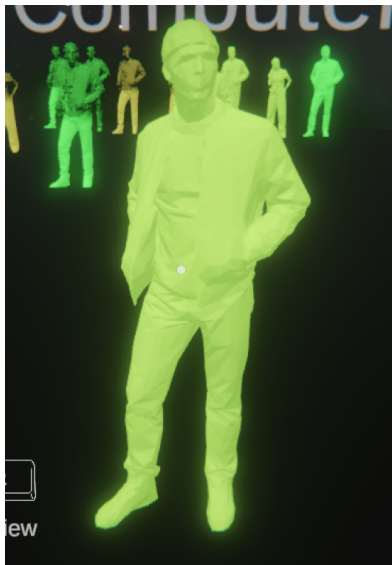


Figura 4.16: Studente.



Figura 4.17: Studentessa.

- **Posizione nello spazio:** assegna la posizione lungo una direzione nello spazio in base ai valori della colonna. È possibile attivare questa opzione su entrambi gli assi, associando una colonna a ciascuno, oppure attivarla su un solo asse e lasciare casuale la posizione lungo l'altro.

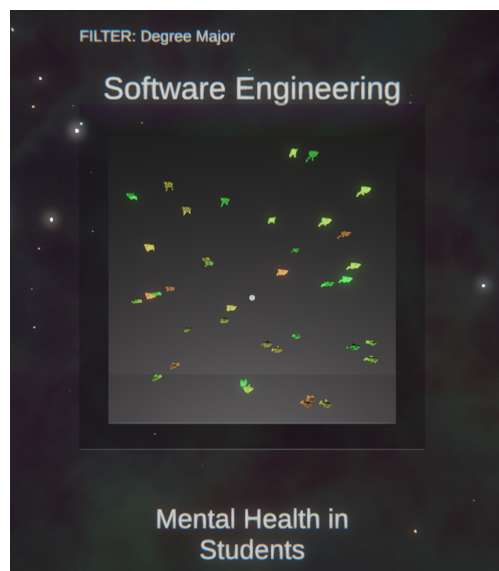
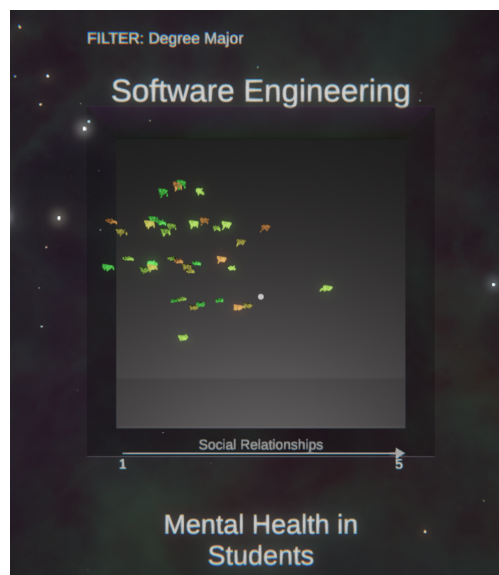
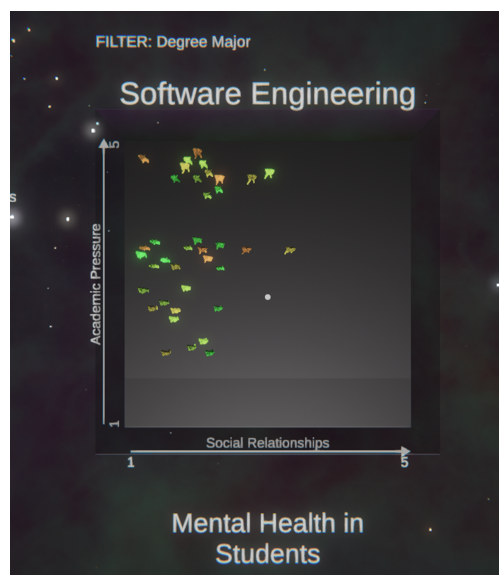


Figura 4.18: Posizione spaziale non attiva: ologrammi posizionati casualmente.



**Figura 4.19:** Posizione attiva su un solo asse: le relazioni sociali crescono lungo x.



**Figura 4.20:** Posizione attiva sui due assi: le relazioni sociali crescono lungo x e la pressione accademica lungo y.

- **Frase pronunciata:** associa a ogni ologramma la sintesi vocale del valore contenuto nella colonna selezionata.

Questa opzione permette anche di associare voci diverse a ologrammi diversi. Ciò avviene con la costruzione di condizioni, descritte nel paragrafo

*Funzionalità avanzate.*

Ad esempio, nel dataset considerato, le frasi pronunciate dagli ologrammi si riferiscono al modo in cui gli studenti sfogano lo stress, mentre i modelli vocali sono stati assegnati in base al genere.

## Funzionalità avanzate

- **Ordine personalizzato:** l'ordine di default per i valori numerici è quello crescente, mentre per le stringhe viene usato l'ordine alfabetico.

Questa opzione permette di definire un ordine personalizzato, particolarmente utile quando i valori contenuti nella colonna sono, ad esempio: *Mai, Raramente, A volte, Spesso, Sempre*.

Nell'esempio proposto, questa funzionalità è stata sfruttata per ordinare nel modo corretto le stringhe della colonna "Average Sleep".

```
<!-- FILL -->
<column index ="10" name ="Average Sleep">
  <dataType>string</dataType>
  <mapping>fill</mapping>
  <customOrder>2-4 hrs, 4-6 hrs, 7-8 hrs</customOrder>
</column>
```

**Figura 4.21:** Applicazione dell'ordine personalizzato.

- **Ordine inverso:** permette di invertire l'ordine di default.
- **Intervalli:** permette di definire degli intervalli personalizzati. Ad esempio, se una colonna contiene valori numerici da 0 a 100, si può associare un colore ai valori che appartengono all'intervallo 0-10, un altro colore ai valori nel range 11-18, un altro ancora per 19-26, e così via.
- **Condizioni:** permette di definire condizioni multi-colonna per l'assegnazione delle caratteristiche agli ologrammi. Le condizioni possono essere usate sia per associare (es.: suono, modello 3D) che per modificare una caratteristica (es.: *voice pitch override*).

```
<!-- 3D MODEL -->
<column index="1" name="Gender">
  <dataType>string</dataType>
  <mapping>model</mapping>
  <conditions>
    <condition modelIndex="0">
      <when column="Gender" equals="MALE"/>
      <when column="Age" lessThan="36"/>
    </condition>
    <condition modelIndex="1">
      <when column="Gender" equals="MALE"/>
      <when column="Age" greaterThan="35"/>
    </condition>
    <condition modelIndex="2">
      <when column="Gender" equals="FEMALE"/>
      <when column="Age" lessThan="36"/>
    </condition>
    <condition modelIndex="3">
      <when column="Gender" equals="FEMALE"/>
      <when column="Age" greaterThan="35"/>
    </condition>
  </conditions>
</column>
```

**Figura 4.22:** Definizione delle condizioni per l'assegnazione dei modelli 3D in base a genere ed età. *Questo codice non fa riferimento all'esempio sugli studenti finora discusso.*

Quanto descritto finora è schematizzato nelle Tabelle 4.1 - 4.2, che per ogni mappatura riportano anche le caratteristiche assunte dagli ologrammi quando questa è "spenta", l'indicazione sul tipo di dato più indicato e le eventuali configurazioni preimpostate (es.: tracce audio di default).

In Tabella 4.3 è invece riassunto il setup del caso d'uso.

Mappatura	Tipologia di dato indicata	Configurazione preimpostata	Valore di default
<b>Colore</b>			
Gradiente di colore	Range	Verde → Azzurro	Azzurro
Colore personalizzato	Categoria, Range	-	
Colore casuale	Categoria	-	
<b>Livello di riempimento</b>	Range	-	0%
<b>Posizione nello spazio</b>	Range	-	Casuale
<b>Suono emesso</b>	Categoria	Pulsazione sonora	Nessun suono
<b>Pitch del suono emesso</b>	Range, Booleano	Pulsazione sonora con pitch che varia da 0 a 2	1
<b>Frase pronunciata</b>	Testo	Voce femminile	Nessuna frase
<b>Modello 3D</b>	Categoria	-	Ologramma androgino
<b>Filtro</b>	Categoria, Range	-	Nessun filtro

Tabella 4.1: Mappature

Mappatura	Ordine personalizzato	Ordine inverso	Intervalli	Condizioni
<b>Colore</b>				
Gradiente di colore	✓	✓	×	×
Colore personalizzato	✓	✓	✓	×
Colore casuale	×	×	✓	×
<b>Livello di riempimento</b>	✓	✓	✓	×
<b>Posizione nello spazio</b>	✓	✓	×	×
<b>Suono emesso</b>	✓	✓	✓	×
<b>Pitch del suono emesso</b>	✓	✓	✓	×
<b>Frase pronunciata</b>	×	×	×	✓
<b>Modello 3D</b>	×	×	×	✓
<b>Filtro</b>	✓	✓	×	×

Tabella 4.2: Funzionalità avanzate

Colonna	Mappatura	Configurazione
Corso di studi	Filtro	-
Soddisfazione per il corso di studi	Colore	Gradiente rosso → verde
Ore di sonno in media	Livello di riempimento	Ordine: 2-4 hrs, 4-6 hrs, 7-8 hrs
	Suono emesso	-
Ansia	Pitch del suono emesso	1 → 3
Genere	Modello 3D	Maschio = Modello 3D ragazzo
		Femmina = Modello 3D ragazza
Relazioni sociali	Posizione nello spazio	Asse x
Pressione accademica	Posizione nello spazio	Asse y
Sollievo dallo stress	Frase pronunciata	Maschio = Voce maschile
		Femmina = Voce femminile

**Tabella 4.3:** Setup del dataset sulla salute mentale degli studenti

### 4.2.3 Scripts

#### Authoring

La parte di Authoring configura le *Entities* e i *Components* dell'architettura ECS: questi script definiscono i componenti sui `GameObject` e, tramite il Baker, li convertono in entità [28].

I tipi di dato impiegati per le entità sono esclusivamente *unmanaged* (float, int, `FixedString32Bytes`, ecc.), a differenza dei tipi *managed* (classi, stringhe, liste, ecc.). La distinzione è cruciale poiché il Burst Compiler supporta solo i tipi *unmanaged*: questi, risiedendo nello stack, garantiscono un accesso cache-friendly e permettono ottimizzazioni estreme. I dati di tipo *managed*, al contrario, richiedono di essere allocati nell'heap allocation e l'uso del Garbage Collector, il che li rende incompatibili con Burst.

Tra gli script di Authoring che compongono il prototipo, il più importante è il `DataManagerAuthoring`, poiché rappresenta il punto d'ingresso della pipeline.

In questo script, il Baker crea il componente `DataManager`, che contiene i flag di inizializzazione di tutti i sistemi di mapping. All'interno del Baker avvengono anche i *parsing* del CSV e dell'XML: i dati estratti dai file di input e setup vengono memorizzati in buffer composti da strutture appositamente definite. Viene inoltre predisposto un buffer per ogni mappatura: inizialmente vuoti, essi saranno popolati a runtime dai rispettivi sistemi di mapping.

## Mapping Systems

Questi sistemi vengono eseguiti una sola volta a runtime, motivo per cui non si è ritenuto necessario l'utilizzo dei Jobs.

I Mapping Systems controllano innanzitutto il buffer che contiene tutte le mappature da applicare: se non viene trovata la caratteristica di cui il sistema si occupa, viene assegnato il valore di default; altrimenti, si procede con l'elaborazione dei dati estraendo la colonna target dal buffer con i dati derivanti dalla lettura del CSV.

Nei sistemi che prevedono il calcolo interpolato della caratteristica, la colonna viene dunque ordinata seguendo la configurazione specificata durante la fase di input o, se assente, l'ordine di default.

Se il sistema prevede il calcolo della caratteristica, si procede con l'interpolazione lineare tra il valore minimo e quello massimo. Il calcolo è stato programmato in modo da supportare la presenza di eventuali intervalli, se impostati in fase di setup. Altrimenti, se la mappatura prevede l'assegnazione diretta o secondo condizioni multi-colonna, si procede alla memorizzazione del valore specificato.

A questo punto, la caratteristica calcolata viene scritta sul buffer di output corrispondente. Inoltre, per le caratteristiche che prevedono una legenda, viene effettuato il salvataggio dei metadati su un componente appositamente creato per la costruzione della UI.

Come ultima cosa, il sistema viene segnato come eseguito.

## Spawner System

Lo `SpawnerSystem` rappresenta un altro punto cruciale nel flusso architetturale di *inPersona*: è il sistema che popola la stanza sia all'inizio dell'esperienza che ogni qual volta il valore del filtro viene cambiato. Dato che, a differenza degli altri sistemi, viene eseguito molte volte, anche a distanza ravvicinata, l'implementazione prevede l'uso di un Job, lo `spawnJob`, come già anticipato nella sezione 4.1.3.

Per partire, lo `SpawnerSystem` ha bisogno della conferma che tutti i Mapping Systems abbiano completato l'esecuzione. Quando il rispettivo flag nel `DataManager` viene marcato, si procede con il controllo di un'eventuale richiesta di *respawn*: se questa è presente, il sistema distrugge tutte le entità precedentemente collocate nella scena e procede con il ripopolamento.

Innanzitutto, viene calcolata l'area in cui posizionare le entità. Questo avviene in base al gruppo più numeroso definito dall'opzione di *Filtro* o, se l'opzione di *Posizione nello spazio* è attiva, in base ai valori calcolati nel rispettivo System.

A questo punto, vengono selezionate le entità da collocare nella scena in base al valore corrente del filtro: questa operazione viene svolta in parallelo su più core grazie allo `spawnJob`, che assegna alle entità modello 3D, posizione e tutte le altre eventuali caratteristiche.

## UI MonoBehaviours

Tutti gli script che si occupano di compilare la UI condividono lo stesso pattern: aspettano che il mondo ECS sia pronto e che i dati siano inizializzati, quindi leggono i buffer per trovare la regola pertinente. Se la regola non esiste, si disattivano; altrimenti, popolano la UI.

La maggior parte di questi script viene eseguita una sola volta, fatta eccezione per quelli che gestiscono la legenda del *Filtro* e la comparsa delle info sui controlli (funzionalità necessaria per il prototipo).

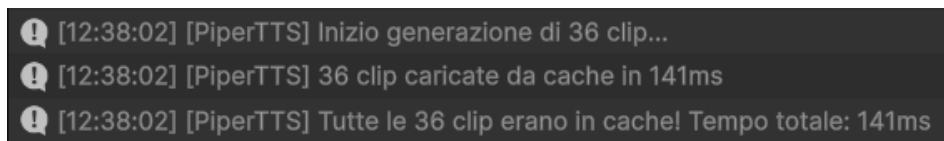
## Apply Systems

Gli Apply Systems costituiscono l'ultimo anello della pipeline ECS: ricevono le entità già *spawnate* dallo `SpawnerSystem`, ciascuna dotata dei componenti calcolati dai Mapping Systems, e li traducono in effetti concreti sul `GameObject` associato, scrivendo nei materiali URP, impostando le `AudioSource` e assegnando le clip vocali generate da Piper.

Questi sistemi sono fondamentali poiché i Mapping Systems e lo `SpawnerSystem` operano a livello di dati ECS puri (struct, buffer, componenti), che per produrre risultati visibili nella scena devono essere applicati alle risorse *managed* (materiali, `AudioClip`, controller `MonoBehaviour`), incompatibili con ECS.

## Text To Speech MonoBehaviour

`PiperTTSPProvider` è lo script `MonoBehaviour` che genera le clip audio, invocando `piper.exe` in background, gestendo cache su disco e convertendo i WAV in `AudioClip`.



```
[12:38:02] [PiperTTS] Inizio generazione di 36 clip...  
[12:38:02] [PiperTTS] 36 clip caricate da cache in 141ms  
[12:38:02] [PiperTTS] Tutte le 36 clip erano in cache! Tempo totale: 141ms
```

**Figura 4.23:** Messaggi in console generati dal TTS.

## Interaction MonoBehaviours

Infine, le interazioni a runtime sono state implementate tramite script MonoBehaviour. Tra queste si elencano:

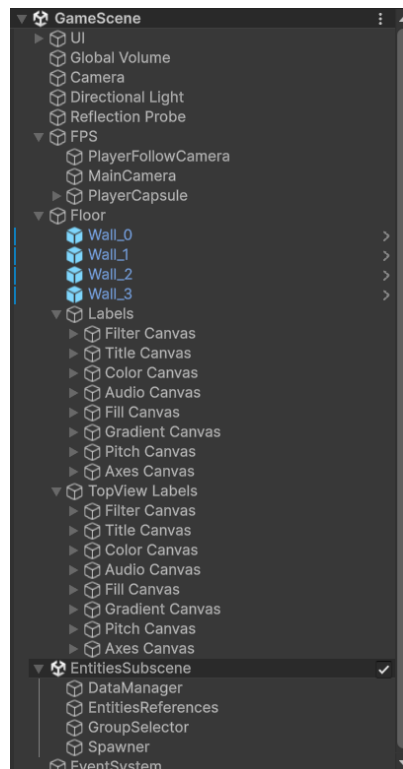
- Il cambio tra camera in prima persona e vista dall’alto con teletrasporto,
- Il feedback visivo del cursore che cambia aspetto quando punta un ologramma,
- La navigazione ciclica tra gruppi di dati tramite tastiera,
- Il click sulle unità per riprodurre la clip vocale (mettendo automaticamente in pausa la riproduzione della traccia relativa all’opzione *Suono*).

### 4.2.4 Unity

In questa sezione viene mostrato il setup del prototipo nell’ambiente Unity.

#### Scene Hierarchy

Gli elementi principali che compongono la scena sono riportati in Figura 4.24, mentre la Tabella 4.4 mostra l’assegnazione degli script ai GameObject.



**Figura 4.24:** Gerarchia della scena in Unity. Il GameObject *UI* contiene gli elementi della UI.

GameObject	Scripts
<b>GameScene</b>	
MainCamera	CursorRaycast
Floor	FloorScaler, CameraSwitcher, AssetProvider, PiperTTSPProvider, ShowInfo
FilterCanvas	FilterDisplay, GroupDisplay
TitleCanvas	TitleDisplay
ColorCanvas	ColorDisplay
AudioCanvas	AudioDisplay
FillCanvas	FillDisplay
GradientCanvas	GradientDisplay
PitchCanvas	PitchDisplay
AxesCanvas → xAxis	AxisDisplay
AxesCanvas → yAxis	AxisDisplay
<b>EntitiesSubscene</b>	
DataManager	DataManagerAuthoring
EntitiesReferences	EntitiesReferencesAuthoring
GroupSelector	GroupSelectorAuthoring
Spawner	SpawnerAuthoring
Unit	UnitAuthoring, UnitClickHandler, UnitAudioController

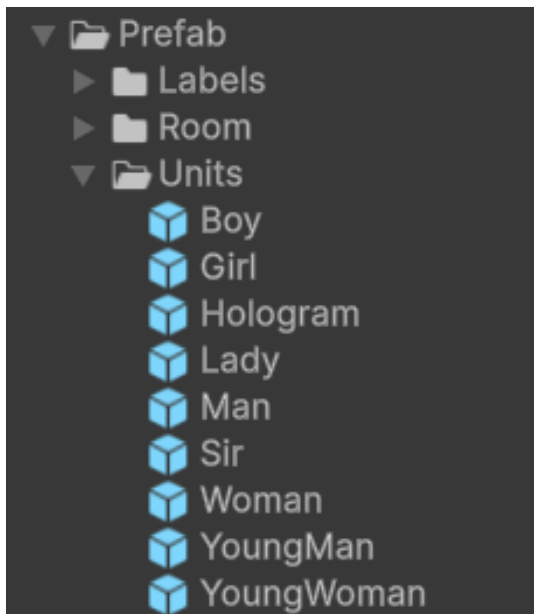
**Tabella 4.4:** Assegnazione scripts ai GameObject

## Assets

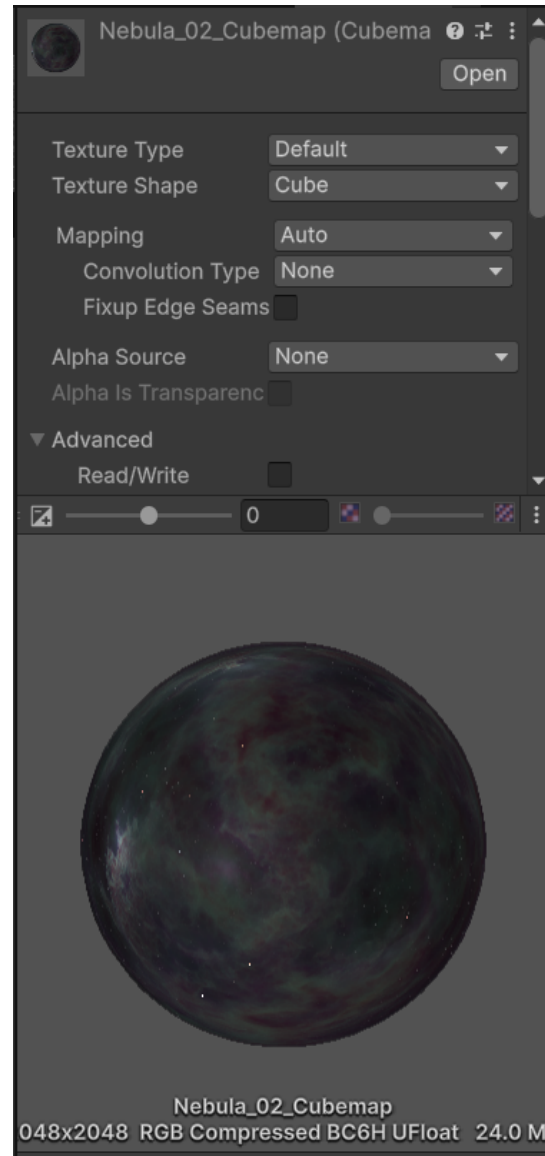
Gli assets usati per la realizzazione del prototipo sono di seguito elencati.

- **Modelli 3D:** l'ologramma androgino è stato modellato su Blender modificando un modello di BlenderKit [32], mentre tutti i modelli usati per le persone di varie età e genere (Fig. 4.25) sono stati esportati da BlenderKit senza subire modifiche.
- **Textures:** la galassia nello sfondo è stata ottenuta impostando come skybox una cubemap dell'Asset Store di Unity [33] (Fig. 4.26).
- **Suoni:** tutti i suoni usati sono privi di copyright e provengono da freesound [34].

- **Dataset:** i file CSV sono stati scaricati da kaggle [35].



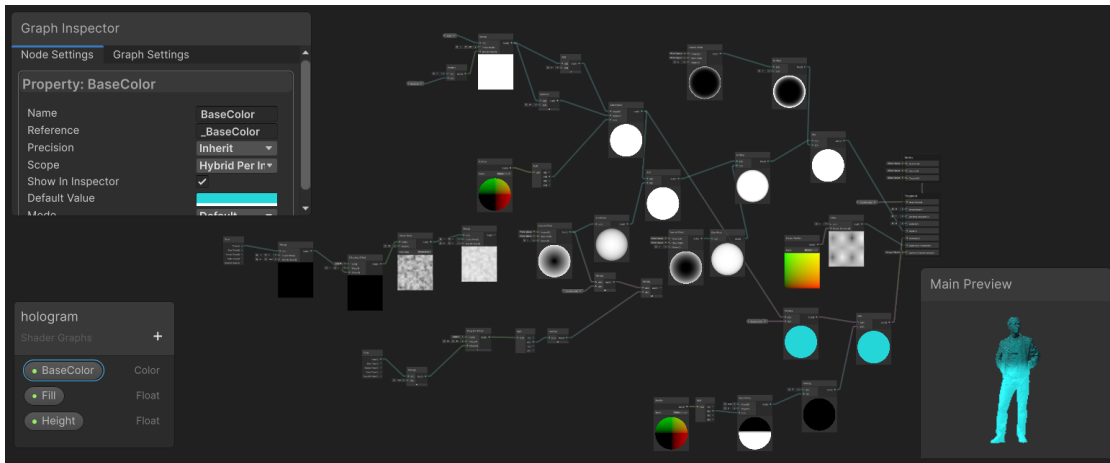
**Figura 4.25:** Elenco dei modelli 3D (prefabs) nel Project Manager di Unity.



**Figura 4.26:** Cubemap della galassia usata per lo sfondo.

## Shader

I materiali di ologrammi, muri, e pavimento sono stati costruiti da zero con lo Shader Graph di Unity (Fig.4.27).



**Figura 4.27:** Shader Graph del materiale degli ologrammi. I valori delle variabili *Fill* e *Color* vengono impostati dinamicamente via script dagli Apply System, come spiegato nella relativa sezione.

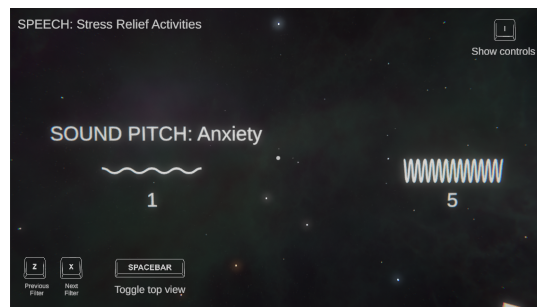
### 4.3 Versione 1.0 VS Versione 1.1

Sono state realizzate due versioni del prototipo: le motivazioni di questa scelta sono fornite nel Capitolo 5, mentre in questa sezione si mostrano le principali differenze tra le due versioni.

- **Legenda del pitch:** la tartaruga (pitch basso) e la lepre (pitch alto) sono state sostituite con la rappresentazione di una sinusoide coerente con il valore indicato.



**Figura 4.28:** Legenda del pitch: versione 1.0



**Figura 4.29:** Legenda del pitch: versione 1.1

- **Feedback cursore:** nella prima versione, per segnalare l'interazione con gli

ologrammi, il cursore incrementa la propria opacità, mentre nella seconda versione diventa un fumetto.

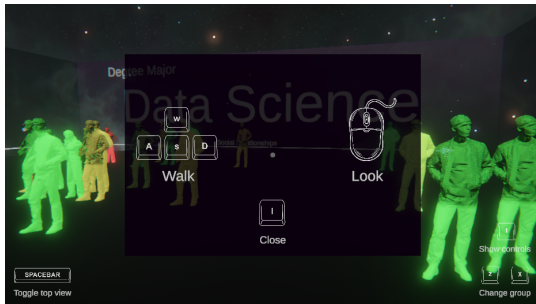


**Figura 4.30:** Feedback cursore: versione 1.0

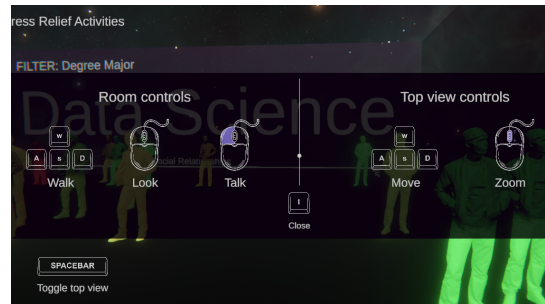


**Figura 4.31:** Feedback cursore: versione 1.1

- **Schermata controlli:** la versione 1.1 contiene informazioni sui controlli più complete rispetto alla 1.0, come mostrato in Figura 4.33.



**Figura 4.32:** Schermata controlli: versione 1.0



**Figura 4.33:** Schermata controlli: versione 1.1

- **Legende:** alle legende sono stati integrati i nomi delle mappature. È inoltre stata aggiunta l'indicazione, in alto a sinistra nella schermata, sulla frase pronunciata dagli ologrammi.
- **UI:** gli elementi della UI sono stati ingranditi e riposizionati.



## Capitolo 5

# User testing

I test utente di *inPersona* sono stati condotti con lo scopo di valutare la leggibilità e l'intuitività delle rappresentazioni implementate, e verificare se fosse stato raggiunto l'obiettivo fondante del progetto di restituire la dimensione umana ai dati sulle persone.

### 5.1 Campione selezionato

La Tabella 5.1 riporta l'elenco degli utenti selezionati per i test.

I primi sette test sono stati condotti principalmente su studenti universitari, al fine di validare i task iniziali e identificare problemi macroscopici senza coinvolgere immediatamente utenti professionali. Questa strategia si è rivelata efficace, poiché ha evidenziato la necessità delle modifiche elencate nella Sezione 4.3 e implementate nella versione 1.1 del prototipo.

Dopo questa iterazione, la release aggiornata è stata testata con altri sei partecipanti. Si è trattato di professionisti dei settori dell'istruzione e del marketing, individuati come ambiti target nella Sezione 3.4.

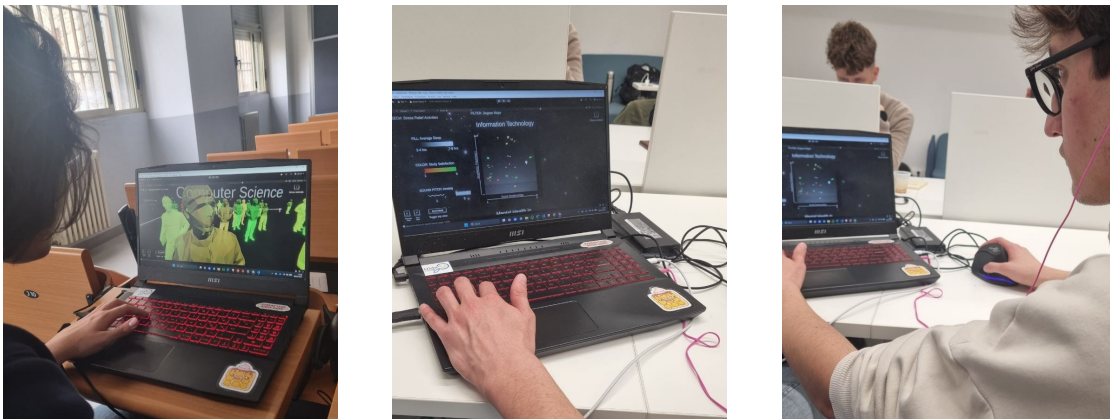
### 5.2 Metodologia e strumenti

L'approccio scelto per i test è stato il paradigma *think aloud* moderato: le sessioni di testing sono infatti state condotte da un moderatore che ha invitato i tester a "pensare ad alta voce", in modo da comprendere il loro flusso di pensieri durante l'utilizzo di *inPersona*. I test hanno avuto durata variabile tra 20 minuti e un'ora.

Genere	Età	Professione
<b>Beta test</b>		
Uomo	25	Professionista nel settore automotive
<b>Versione 1.0</b>		
Donna	23	Studentessa
Donna	26	Studentessa
Donna	25	Studentessa
Uomo	26	Studente
Uomo	28	Studente
Donna	45	Sales Support Specialist
Uomo	53	Ingegnere Informatico
<b>Versione 1.1</b>		
Donna	25	Junior Communication Specialist
Uomo	24	Sviluppatore web
Uomo	25	Studente
Uomo	25	Studente
Donna	44	Professoressa
Uomo	43	Customer Relationship Manager

**Tabella 5.1:** Partecipanti ai test di usabilità

Le sessioni di testing hanno richiesto un setup semplice: un pc, con mouse e auricolari collegati, su cui eseguire l'app da far testare all'utente. Il moderatore si è invece servito di un iPad per prendere appunti e di uno smartphone per scattare foto.



**Figura 5.1:** Setup delle sessioni di testing.

## 5.3 Criteri di valutazione

In questa sezione vengono illustrate le varie parti del test di usabilità di *inPersona*.

- **Introduzione:** il partecipante viene accolto e gli viene indicata la durata del test e il contesto del progetto. Si procede con la richiesta di consenso per appunti, foto e registrazione, e la descrizione della struttura del test.

In questa fase, il partecipante viene inoltre rassicurato sul focus della sessione, che è testare l'applicazione e non la performance dell'utente, e viene incoraggiato a pensare ad alta voce, fornendo anche degli esempi di *think aloud*.

- **Scenario:** all'utente viene proposto uno scenario in cui calarsi durante l'esecuzione dei task. Si tratta di una tecnica standard prevista da questo tipo di test. In particolare, per i test utente di *inPersona* è stato detto agli utenti di immaginare di trovarsi a un evento che prevedesse uno spazio espositivo (mostra, fiera, salone...).
- **Task:** all'utente viene chiesto di svolgere una serie di task di difficoltà crescente. I nove task progettati per i test utente di *inPersona*, riassunti anche in Tabella 5.2, sono di seguito descritti:

1. Dire a cosa serve l'app.
2. Individuare tutte le mappature attive.
3. Ottenere una vista d'insieme della stanza (*TopView*) e spostarsi velocemente da un punto all'altro di essa (teletrasporto).
4. Trovare il modo per determinare il corso di studi con gli studenti più soddisfatti.
5. Trovare il modo per determinare il corso di studi con gli studenti più riposati.
6. Trovare il modo per determinare un'eventuale relazione tra pressione accademica e relazioni sociali.
7. Trovare il modo per determinare un'eventuale relazione tra genere e pressione accademica.
8. Trovare il modo per determinare un'eventuale relazione tra ansia e pressione accademica.
9. Trovare il modo per determinare cosa fanno gli studenti che provano poca pressione accademica per ridurre lo stress.

I primi tre task hanno lo scopo di far prendere confidenza all'utente con l'ambiente virtuale, mentre i task 4-9 testano la fruibilità del sistema e la leggibilità delle mappature.

- **Questionario post-task:** ogni task riceve una valutazione sia dal moderatore che dall'utente. Il moderatore indica se il task è stato svolto con successo, con un piccolo inconveniente, con un grande inconveniente, o se è stato fallito. All'utente vengono invece poste alcune domande, a cui rispondere assegnando un punteggio da 1 a 7.
- **Questionario post-test SUS:** all'utente viene sottoposto il questionario SUS (System Usability Scale). Si tratta di una lista standard di dieci domande che permette di calcolare il punteggio di usabilità di un sistema.
- **Feedback e commenti liberi:** a questo punto, l'utente è invitato, se lo desidera, a esplorare l'applicazione in autonomia e fornire eventuali commenti in modo non strutturato.

Durante i test di *inPersona*, questa fase ha permesso agli utenti di esplorare altri dataset (precedentemente configurati per la visualizzazione) e modificare il setup delle mappature a proprio piacimento.

**Tabella 5.2:** Elenco dei task e delle valutazioni

Task	Valutazione	Descrizione
1	Successo	L'utente crede che <i>inPersona</i> sia uno strumento di visualizzazione per dati relativi alle persone
	Piccolo inconveniente	L'utente capisce che l'app ha a che fare con la visualizzazione dati, ma non riesce a capire il significato degli oggetti nella scena
	Grande inconveniente	L'utente pensa di essere davanti a una visualizzazione statica di dati o che l'app tratti in modo specifico solo il dataset attualmente visualizzato
	Fallimento	L'utente pensa che l'app abbia a che fare con tutt'altro che la visualizzazione dati
2	Successo	L'utente individua tutte le mappature attualmente attive nella stanza
	Piccolo inconveniente	L'utente individua tutte le mappature, ma ne confonde i significati, oppure individua mappature che non esistono
	Grande inconveniente	L'utente non individua tutte le mappature
	Fallimento	L'utente non capisce come interagire con la stanza
Domanda 1		Quanto è stato facile capire che i tasti z e x servono ad andare al valore precedente o successivo del filtro?
Domanda 2		Quanto reputi adeguato il funzionamento del cursore in relazione agli ologrammi cliccabili presenti nella stanza?

*Continua nella pagina successiva*

Tabella 5.2: Elenco dei task e delle valutazioni (continua)

Task	Valutazione	Descrizione
Domanda 3		Quanto è stato facile capire che il pitch dei suoni emessi dagli ologrammi rappresenta una mappatura?
Domanda 4		Quanto è stato facile associare le legende sulle pareti alle caratteristiche degli ologrammi?
3	Successo	L'utente usa la <i>TopView</i> con teletrasporto per spostarsi velocemente nella stanza
	Piccolo inconveniente	L'utente fa diversi tentativi prima di usare la <i>TopView</i> con teletrasporto
	Grande inconveniente	L'utente usa la <i>TopView</i> ma non per teletrasportarsi
	Fallimento	L'utente non usa la <i>TopView</i>
Domanda 5		Quanto è stato facile capire come attivare la <i>TopView</i> ?
Domanda 6		Quanto è stato facile capire come teletrasportarti da un punto all'altro?
Domanda 7		Quanto reputi adeguato il funzionamento del teletrasporto?
4	Successo	L'utente trova il modo per individuare il corso di studi con gli utenti più soddisfatti
	Piccolo inconveniente	L'utente capisce come fare a determinare il corso di studi desiderato, ma con qualche difficoltà o errore di interpretazione
	Grande inconveniente	L'utente capisce come fare a determinare il corso di studi desiderato, ma confonde le mappature
	Fallimento	L'utente non ha idea di come fare a determinare il corso di studi con gli studenti più soddisfatti
Domanda 8		Quanto è stato facile determinare il corso di studi con gli studenti più soddisfatti?
5	Successo	L'utente trova il modo per individuare il corso di studi con gli studenti più riposati
	Piccolo inconveniente	L'utente capisce come fare a determinare il corso di studi desiderato, ma con qualche difficoltà o errore di interpretazione
	Grande inconveniente	L'utente capisce come fare a determinare il corso di studi desiderato, ma confonde le mappature
	Fallimento	L'utente non ha idea di come fare a determinare il corso di studi con gli studenti più riposati
Domanda 9		Quanto è stato facile determinare il corso di studi con gli studenti più riposati?

*Continua nella pagina successiva*

**Tabella 5.2:** Elenco dei task e delle valutazioni (continua)

Task	Valutazione	Descrizione
6	Successo	L'utente trova il modo per identificare la relazione richiesta
	Piccolo inconveniente	L'utente capisce come fare a individuare la relazione, ma con qualche difficoltà o errore di interpretazione
	Grande inconveniente	L'utente capisce come fare a individuare la relazione, ma confonde le mappature
	Fallimento	L'utente non ha idea di come fare a determinare una relazione tra pressione accademica e relazioni sociali
Domanda 10		Quanto è stato facile individuare la relazione tra pressione accademica e relazioni sociali?
7	Successo	L'utente trova il modo di determinare se le ragazze sentono la pressione accademica più dei ragazzi
	Piccolo inconveniente	L'utente capisce come fare a determinare l'aspetto desiderato, ma con qualche difficoltà o errore di interpretazione
	Grande inconveniente	L'utente capisce come fare a determinare l'aspetto desiderato, ma confonde le mappature
	Fallimento	L'utente non ha idea di come fare a determinare chi senta di più la pressione accademica tra ragazze e ragazzi
Domanda 11		Quanto è stato facile determinare se c'è un genere che prova più pressione accademica?
Domanda 12		Quanto è stato facile individuare le eccezioni al trend?
8	Successo	L'utente trova il modo per determinare la relazione richiesta
	Piccolo inconveniente	L'utente capisce come fare a individuare la relazione, ma con qualche difficoltà o errore di interpretazione
	Grande inconveniente	L'utente capisce come fare a individuare la relazione, ma confonde le mappature
	Fallimento	L'utente non ha idea di come fare a individuare una relazione tra ansia e pressione accademica
Domanda 13		Quanto è stato facile individuare la relazione tra ansia e pressione accademica?
9	Successo	L'utente trova il modo di indagare quanto richiesto
	Piccolo inconveniente	L'utente capisce come fare a scoprirlo, ma facendo più tentativi del necessario o commettendo errori di interpretazione

*Continua nella pagina successiva*

**Tabella 5.2:** Elenco dei task e delle valutazioni (continua)

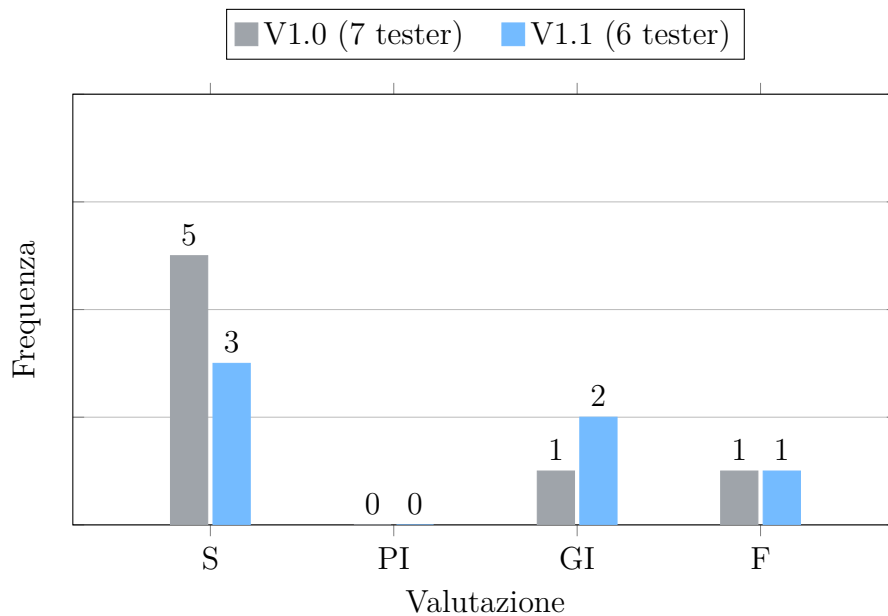
Task	Valutazione	Descrizione
	Grande inconveniente	L'utente capisce come fare a individuare la relazione, ma confonde le mappature
	Fallimento	L'utente non ha idea di come fare a trovare la risposta
Domanda 14		Quanto è stato facile scoprire cosa fanno gli studenti per alleviare lo stress?

## 5.4 Risultati

In questa sezione vengono presentati i risultati dei test utente. In primo luogo, si riportano i risultati relativi ai singoli task, mettendo a confronto le prestazioni ottenute con la versione 1.0 e con la versione 1.1 del prototipo. Vengono poi illustrati i risultati del questionario SUS e una sintesi dei feedback.

Nei grafici di seguito riportati, S sta per *Successo*, PI per *Piccolo Inconveniente*, GI per *Grande Inconveniente* e F per *Fallimento*.

### Task 1

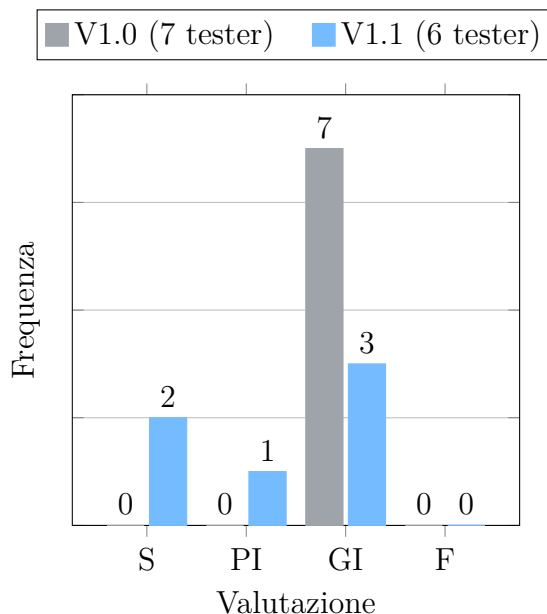
**Figura 5.2:** Esiti Task 1

Il Task 1 richiedeva all'utente di individuare lo scopo dell'applicazione: 9 utenti su 13 hanno compreso di essere di fronte a una visualizzazione di dati su persone, rappresentate dai personaggi nella stanza. Questi utenti hanno descritto gli elementi visivi usando parole come: grafici, barre, scale, indicatori, UI, filtri o unità di misura.

Tra gli utenti che non hanno avuto successo nel primo task, uno ha interpretato l'app come una simulazione in tempo reale di persone presenti a un evento, un altro come uno strumento per misurare la vista, mentre un terzo ha dichiarato di non comprenderne lo scopo e ha giudicato poco informativi gli elementi visivi.

Infine, un utente ha notato esplicitamente di ricevere "tante info tutte in una volta", suggerendo fin da subito un elevato carico cognitivo richiesto dall'applicazione.

## Task 2



	V1.0	V1.1
Domanda 1	2.7	3.5
Domanda 2	2.6	5.3
Domanda 3	4.1	4.5
Domanda 4	5.9	5.8

**Tabella 5.3:** Media risposte alle domande post-task 2.

**Figura 5.3:** Esiti Task 2

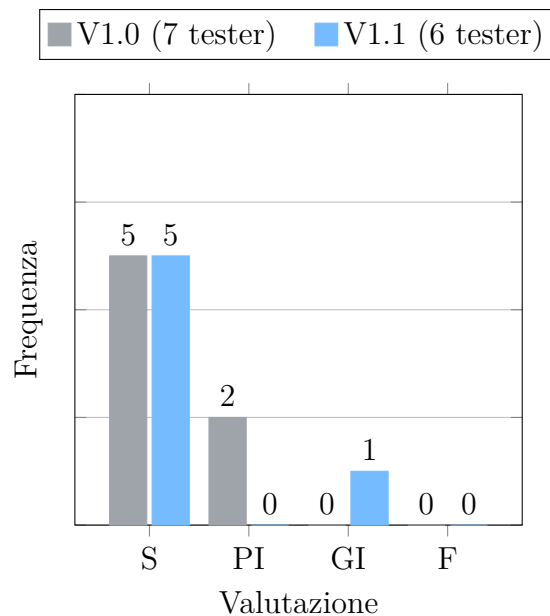
Il secondo task richiedeva di individuare le mappature attive: la maggior parte degli utenti ha subito iniziato a muoversi nell'ambiente virtuale.

In generale, le mappature più facili da individuare sono *Suono*, *Colore* e *Posizione nello spazio*, mentre il *Livello di riempimento* sembra essere la mappatura più difficile da interpretare nel modo corretto. Quasi nessun utente ha "scoperto" la funzionalità di cambio *Filtro*.

Per quanto riguarda il riconoscimento delle mappature di *Pitch* e *Frasi pronunciate*, si notano dei miglioramenti significativi in seguito alle modifiche introdotte con la versione 1.1; infatti, mentre nella versione 1.0 i commenti sul *Pitch* vertevano sull'inadeguatezza delle icone usate per la legenda, nella versione 1.1 questi si spostano sulla difficoltà a capire la fonte del suono nei cluster di ologrammi.

Similmente, mentre nei test della prima versione i commenti sull'interazione con gli ologrammi riguardavano la scarsa visibilità del feedback del cursore, nella seconda versione la risoluzione di questo problema permette di farne emergere un altro, rappresentato da un bug nei collider degli ologrammi che impedisce l'interazione se il cursore è puntato dal petto in su.

### Task 3



**Figura 5.4:** Esiti Task 3

	V1.0	V1.1
<b>Domanda 5</b>	5.6	6
<b>Domanda 6</b>	6.6	5.7
<b>Domanda 7</b>	6.6	7

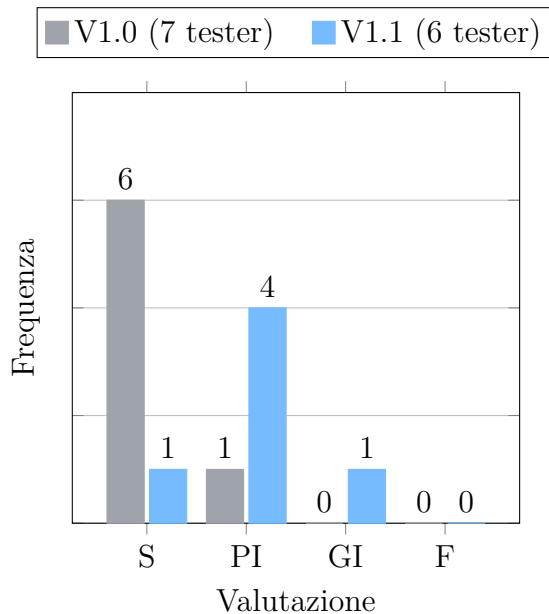
**Tabella 5.4:** Media risposte alle domande post-task 3.

Nel terzo task veniva chiesto all'utente di ottenere una vista d'insieme della situazione attuale della stanza e di spostarsi velocemente da un punto a un altro di essa. Il task si è ritenuto completato con successo qualora l'utente avesse usato la *Top View* e il teletrasporto in combinazione.

Fatta eccezione per un utente che non aveva visto l'indicazione sui comandi per passare in *Top View*, e un altro utente che ha capito come teletrasportarsi solo dopo qualche tentativo, non si segnalano particolari criticità nello svolgimento di

questo task, che risulta infatti uno dei più facili in base alle risposte alle domande post-task.

#### Task 4



	V1.0	V1.1
<b>Domanda 8</b>	6.3	5.7

**Tabella 5.5:** Media risposte alle domande post-task 4.

**Figura 5.5:** Esiti Task 4

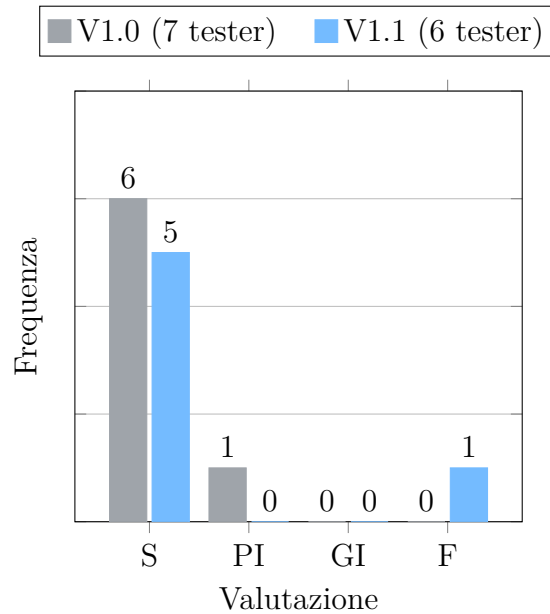
Con il quarto task si inizia a testare leggibilità e fruibilità dei dati. Viene chiesto agli utenti di determinare il corso di studi di cui gli studenti sono più soddisfatti. In questo caso, il successo non dipende tanto dalla risposta esatta, quanto dal fatto che il partecipante riesca a trovare e applicare un metodo per arrivare a una risposta.

La maggior parte degli utenti ha portato a termine questo task con successo o con piccoli inconvenienti. Molti di loro hanno scelto di scorrere i valori del *Filtro* osservando la stanza dalla *TopView*, mentre solo un utente ha deciso di rimanere in *RoomView*.

Una delle difficoltà principalmente individuate nello svolgimento di questo task riguarda la mancanza di un'indicazione sul numero totale di ologrammi di volta in volta presenti nella stanza: un utente ha infatti affermato che "*non si può dire con certezza*" quale sia il gruppo con gli studenti più soddisfatti del loro corso di studi, sottolineando proprio la mancanza di un'informazione numerica.

Alcuni utenti hanno invece confuso le mappature, scambiando l'informazione mappata sul *Colore* con quelle riportate sugli assi.

### Task 5



**Figura 5.6:** Esiti Task 5

	V1.0	V1.1
<b>Domanda 9</b>	5	4.7

**Tabella 5.6:** Media risposte alle domande post-task 5.

Il quinto task richiedeva invece di determinare il corso di studi con gli studenti più riposati.

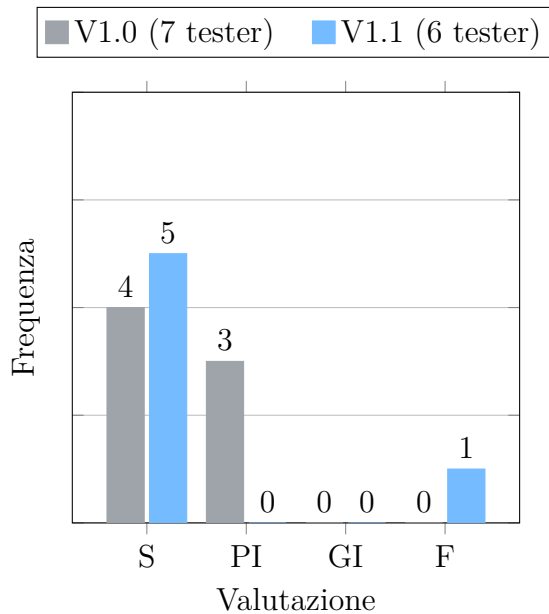
Durante lo svolgimento, alcuni utenti hanno continuato a usare il metodo già individuato al task precedente, servendosi talvolta dello strumento di zoom offerto dalla *TopView*, mentre altri hanno preferito tornare in *RoomView*.

La criticità nel completamento di questo task consiste nella scarsa leggibilità del *Livello di riempimento* dalla *TopView*, combinata all'impossibilità di una vista d'insieme dalla *RoomView*.

Inoltre, è emerso un problema di leggibilità legato all'estetica del prototipo: solo gli ologrammi con un colore abbastanza brillante emettono un effetto "glow". Questo fattore ha inciso a peggiorare la comprensibilità della visualizzazione.

Infine, due utenti hanno lamentato una scarsa manovrabilità dell'inquadratura in *topView*, suggerendo una realizzazione alternativa che implementi il *drag-and-drop*.

## Task 6



**Figura 5.7:** Esiti Task 6

	V1.0	V1.1
<b>Domanda 10</b>	5.43	4.8

**Tabella 5.7:** Media risposte alle domande post-task 6.

Il sesto task chiedeva di trovare un'eventuale relazione le grandezze mappate sui due assi.

Per lo svolgimento del Task 6, la maggior parte degli utenti ha proseguito con l'utilizzo della *TopView*; solo un utente ha optato per la *RoomView*, affermando di preferire il movimento nella stanza alla vista dall'alto.

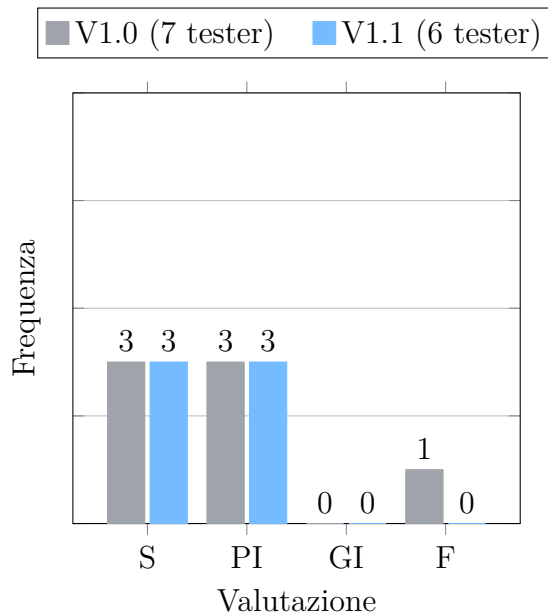
Un utente ha ammesso di essersi dimenticato degli assi "*dopo tutto questo tempo*", indicando ancora una volta l'elevato carico cognitivo richiesto dall'applicazione.

Solo un utente non è stato in grado di portare a termine il task, a causa della mancanza di valori e della scarsa intuitività della visualizzazione. Ha infatti suggerito l'aggiunta di grafici interattivi.

## Task 7

Il Task 7 richiedeva di determinare quale genere sentisse più pressione accademica, e di individuare eventuali eccezioni al trend.

Nello svolgimento di questo task, molti partecipanti alternano *TopView* e *RoomView*, percorrendo la stanza da cima a fondo e concentrandosi su specifiche zone o sui



**Figura 5.8:** Esiti Task 7

	V1.0	V1.1
<b>Domanda 11</b>	3.7	5.5
<b>Domanda 12</b>	4.3	6.3

**Tabella 5.8:** Media risposte alle domande post-task 7.

comportamenti agli estremi.

In generale, gli utenti hanno trovato questo task difficile, trovandosi infatti a cambiare più volte idea sulla risposta, e definendolo un processo lungo e influenzato da bias personali.

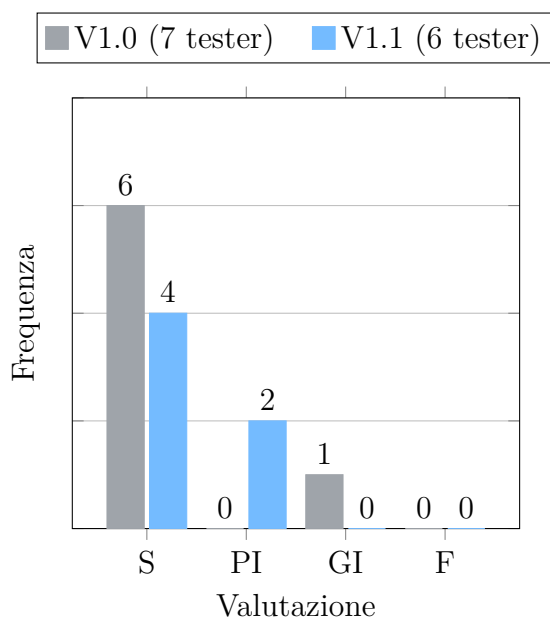
Molti utenti hanno infatti sentito il bisogno di strumenti aggiuntivi, rappresentati da numeri di supporto o funzionalità per raggruppare o isolare gli ologrammi. Un utente ha per esempio suggerito di far poggiare i vari modelli 3D su basi di forme diverse a seconda del genere, in modo che risultino leggibili anche dalla *TopView*.

## Task 8

La risoluzione di questo task prevedeva l'uso combinato del suono e dell'informazione mappata sull'asse y.

Nello svolgimento di questo task sono riemerse le criticità sul suono già individuate durante il Task 2: un utente si è domandato se i suoni sovrapposti provenissero dallo stesso ologramma, per poi capire che non è così; ha dunque suggerito di fare in modo che venga emesso solo il suono degli ologrammi attualmente puntati dal cursore.

Un utente in particolare, pur avendo trovato il modo per indagare sulla presenza di



**Figura 5.9:** Esiti Task 8

	V1.0	V1.1
<b>Domanda 13</b>	5.6	5.5

**Tabella 5.9:** Media risposte alle domande post-task 8.

una relazione, non è stato in grado di individuarne alcuna. Altri invece hanno trovato l'operazione richiesta alquanto lunga, dichiarandolo esplicitamente o limitandosi a sospirare.

D'altro canto, alcuni utenti hanno risposto alla domanda considerando anche l'eventuale mitigamento offerto dalle relazioni sociali, andando quindi oltre le richieste del task e valutando tre grandezze contemporaneamente.

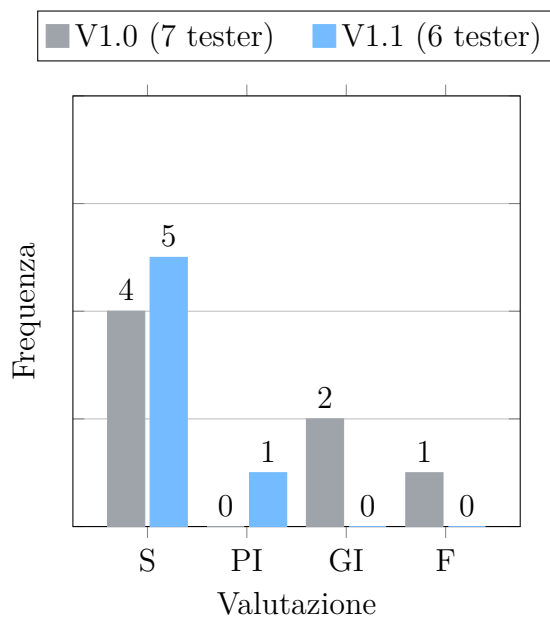
Infine, uno dei commenti più significativi arriva da un utente dislessica, la quale ha affermato che *"il suono aiuta più delle informazioni visive"*.

## Task 9

L'ultimo task prevedeva l'interazione con gli ologrammi combinata con la lettura della *Posizione nello spazio*.

Di particolare rilevanza sono state le parole usate dagli utenti prima di iniziare a svolgere il task; molti di loro hanno infatti pronunciato frasi come: *"andiamo a vedere da vicino e glielo chiediamo"*, *"mi metto a parlare con loro"* o *"te lo dicono loro"*, che evidenziano l'avvenuto riconoscimento degli ologrammi come individui "umani".

Un solo utente ha trovato questo processo "macchinoso".

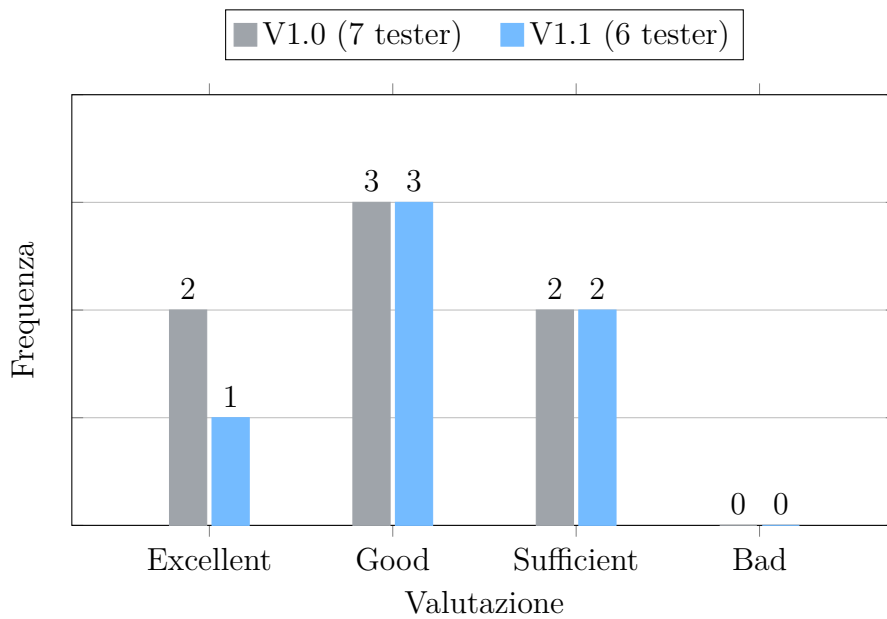


	V1.0	V1.1
<b>Domanda 14</b>	4.6	6.7

**Tabella 5.10:** Media risposte alle domande post-task 9.

**Figura 5.10:** Esiti Task 9

### Questionario SUS



**Figura 5.11:** Esiti questionario SUS

I questionari SUS riportano punteggi buoni, ma non ottimi: poche valutazioni

risultano eccellenti, e molte si aggirano tra la media e il sufficiente. Questi risultati costituiscono la base di partenza per il Capitolo 6, che tratta dei margini di miglioramento e degli sviluppi futuri volti a migliorare il prototipo per i prossimi test.

## Feedback

I commenti forniti dagli utenti durante la seconda parte del test, dedicata alla raccolta di feedback non strutturati, sono qui presentati e organizzati per argomento.

- **Riumanizzazione dei dati:** tutti gli utenti che hanno avuto modo di esprimersi in merito, hanno dichiarato che *inPersona* riesce a restituire la dimensione umana ai dati sulle persone: hanno infatti sostenuto che vedere persone al posto di barre e tabelle renda l'esperienza più empatica e meno noiosa rispetto ai grafici tradizionali.

Inoltre, l'app è stata spesso descritta come divertente, o comunque coinvolgente, anche da chi normalmente non ama l'analisi dati.

- **Leggibilità della visualizzazione:** il CRM che ha preso parte ai test ha fornito uno dei feedback più importanti, in quanto si tratta della conferma del raggiungimento di uno degli obiettivi delineati nel Capitolo 1, ovvero quello che riguarda la leggibilità.

L'utente, abituato a lavorare quotidianamente con dati e tabelle, ha spiegato che, solitamente, la visualizzazione di più di tre variabili diventa caotica, motivo per cui trova notevole che *inPersona* ne gestisca "addirittura 8" in modo ancora comprensibile, pur riconoscendo che l'applicazione richieda un elevato tempo di utilizzo.

- **Scenari d'uso:** molti utenti hanno dichiarato di vedere *inPersona* come un ottimo strumento di supporto alla didattica o, in generale, alle presentazioni, poiché permette di mostrare i dati in modo più impattante e meno noioso. Alcuni hanno nominato anche processi di recruitment e misurazione della soddisfazione in contesti aziendali.

Il commento più importante in questo contesto proviene dall'insegnante che ha preso parte al test, alla quale è stata mostrata anche la visualizzazione di un dataset sulle vittime di bullismo. L'utente ha quindi affermato che *inPersona* costituirebbe uno strumento utile per i docenti supplenti: "*Mi ritrovo davanti a 26 ragazzi che non conosco: avere una visione del genere prima di entrare in classe mi aiuterebbe a capire con chi ho a che fare senza doverlo scoprire da sola*", dice, facendo riferimento a eventuali situazioni familiari o altri disturbi presenti negli alunni.

- **Ruolo del suono:** alcuni utenti hanno apprezzato particolarmente il ruolo del suono nell'applicazione, facendo commenti come: “*Mi sono sentita dentro la stanza*”, pur segnalando che “*a volte si perde qualcosa nell'analisi dati*”. Diversi utenti hanno infatti chiesto supporti visivi alle mappature sonore e vocali, lamentato l'assenza di esse in *TopView* e suggerito la possibilità di far emettere il suono solo all'ologramma puntato dal cursore.

In questo contesto, il commento più significativo è arrivato da un utente con dislessia che, oltre a trovare nel suono un'alternativa molto valida alle visualizzazioni numeriche tradizionali, ha dichiarato di essere abituata a pensare per immagini, motivo per cui farebbe un uso personale di *inPersona* al fine di ottenere una lettura più visiva dei dati.

- **Supporti e funzionalità mancanti:** quando si è trattato di ottenere risposte precise e non solo qualitative delle distribuzioni, più utenti hanno sentito il bisogno di un'indicazione numerica sulle quantità totali e sub-totali delle varie categorie, e accusato la mancanza di funzionalità di filtraggio specifiche.

In sintesi, l'approccio qualitativo e percettivo è stato apprezzato, ma viene visto come complementare, non sostitutivo degli strumenti tradizionali.

- **Carico cognitivo:** si è notata una certa tendenza degli utenti a confondere tra loro le mappature visive (*Colore*, *Livello di riempimento* e *Posizione nello spazio*).

Al contrario, ciò non è avvenuto per la mappatura sonora: la maggior parte degli utenti, una volta individuato il significato del suono, l'ha infatti memorizzato per il resto dell'esperienza senza confonderlo, come dimostrato dal fatto che, nei task in cui il livello di ansia non era rilevante, molti di loro hanno rimosso l'auricolare, giudicando l'informazione non necessaria.

- **VR e AR:** quasi tutti gli utenti sono stati d'accordo nel dire che la VR aggiungerebbe valore all'esperienza in termini di immersività, sorpresa e impatto, soprattutto in contesti museali.

La realtà virtuale sarebbe quindi la “*ciliegina sulla torta*”, anche se molti hanno considerato la versione desktop già pienamente sfruttabile, se non addirittura più adatta in caso di presentazioni e di uso condiviso.

Due utenti hanno proposto la realtà aumentata come alternativa alla VR, giudicandola più interessante rispetto al visore. Credono infatti che quest'ultimo produca sulle persone un “*effetto wow che finisce subito*”.

- **Stile:** la maggior parte degli utenti ha apprezzato lo stile grafico, trovandolo piacevole, rilassante, non distraente e adatto a evidenziare le differenze tra i dati. Qualcuno lo ha percepito ancora prototipale o leggermente cupo.

- **Gamification:** una tester ha acconsentito alla creazione di un setup ex-novo. La scelta del dataset è ricaduta sulla lista di tutti i piloti di Formula 1 della storia.

La configurazione è stata impostata in modo che il colore mappasse la decade di attività, il riempimento i punti accumulati, il suono lo stato di attività e la posizione nello spazio le pole positions (asse y) e le gare vinte (asse x). Interagendo con gli ologrammi, filtrati per paese d'origine, questi pronunciavano il proprio nome.

L'analisi dati ha subito preso la piega di un'attività ludica, il cui scopo era individuare i piloti più famosi: *"Andiamo a Monaco da Leclerc", "Siamo in UK, quello pieno al 100% là in fondo è sicuramente Hamilton"*.

Questo episodio, oltre a dimostrare che il sistema è in grado di rappresentare i dati in modo efficace, dal momento che i piloti venivano effettivamente "indovinati", ha messo in evidenza un potenziale ludico non previsto in fase di progettazione, particolarmente rilevante in prospettiva di scenari gamificati.

## Capitolo 6

# Limiti e sviluppi futuri

Nel corso dello sviluppo e della valutazione del prototipo di *inPersona*, sono emersi diversi limiti che rappresentano importanti punti di riflessione per il proseguimento del progetto. In questo capitolo vengono dunque elencati tali limiti, fornendo ove possibile una strategia di miglioramento.

Un aspetto particolarmente significativo emerso durante le sessioni di test riguarda il confronto tra la versione desktop e quella in realtà virtuale. Sebbene la versione desktop fosse stata inizialmente concepita esclusivamente come prototipo preliminare per lo sviluppo del sistema e la verifica delle scelte concettuali, è stata comunque percepita come uno strumento efficace dagli utenti. In particolare, pur offrendo un livello di immersione inferiore rispetto alla realtà virtuale, la modalità desktop è stata considerata più versatile e immediata in diversi contesti di utilizzo.

Questo risultato suggerisce la possibilità di ripensare l'evoluzione del progetto non come una semplice transizione dalla versione desktop alla VR, ma piuttosto come lo sviluppo di uno strumento bivalente capace di operare su entrambe le piattaforme. In questa prospettiva, una futura versione 2.0 di *inPersona* potrebbe prevedere la realizzazione e il mantenimento parallelo di due build distinte: una destinata all'utilizzo su desktop e una progettata specificamente per i visori.

I prossimi passi di sviluppo dovrebbero quindi concentrarsi su tre direttrici principali: il miglioramento e la stabilizzazione della versione desktop 2.0, l'adattamento e l'ottimizzazione dell'esperienza per l'ambiente VR, e la conduzione di nuove sessioni di testing comparativo tra le due piattaforme. Solo in una fase successiva, corrispondente a una possibile versione 3.0 del sistema, sarà opportuno valutare un'eventuale passaggio alla realtà aumentata e l'introduzione di dinamiche più avanzate di interazione e coinvolgimento, rappresentate ad esempio da meccaniche di gamification.

## 6.1 File di input

Un primo limite dell'attuale implementazione di *inPersona* è costituito dalla rigidità del formato del file CSV accettato come input valido dall'applicazione.

Per superare questa criticità, una possibile strategia evolutiva consisterebbe nella modifica dell'architettura di parsing e mapping in modo da renderla compatibile con tipologie di dataset più eterogenee, come colonne contenenti percentuali o variabili booleane, senza richiedere all'utente una preventiva armonizzazione dei dati.

L'integrazione di una fase preliminare di "preparazione del dataset" nell'interfaccia utente equivarrebbe infatti allo sviluppo di un ulteriore strumento di data cleaning, funzionalità già ampiamente soddisfatta da soluzioni consolidate come Microsoft Power BI [9], che rappresenta un'opzione già robusta e matura per questo tipo di operazioni.

Nelle implementazioni future si potrebbe introdurre la possibilità di mantenere i nomi originali delle colonne nel dataset sorgente, consentendo invece all'utente di definire durante la fase di setup iniziale le etichette personalizzate da visualizzare nell'interfaccia di *inPersona*.

Un ulteriore miglioramento riguarda la possibilità di modificare i valori di default, riassunti nella tabella 4.1, che al momento sono hard-coded.

## 6.2 Tutorial

Durante le sessioni di test utente, alcuni partecipanti non hanno immediatamente compreso lo scopo di *inPersona*, hanno manifestato un senso di sovraccarico informativo o hanno incontrato difficoltà iniziali nell'orientamento all'interno dell'interfaccia.

Per superare queste criticità, si propone l'aggiunta di un tutorial guidato e ripetibile, accessibile dall'utente in qualsiasi momento. Infatti, se la necessità di tale supporto è già emersa nel contesto del prototipo desktop, risulta sicuramente ancora più imprescindibile per la versione VR.

## 6.3 Legende

I test hanno decretato che le legende realizzate per il prototipo di *inPersona* sono migliorabili sotto diversi aspetti. Una possibile linea guida progettuale consiste

nella quasi totale rimozione della UI, trasferendo le informazioni su controlli e mappature direttamente nell'ambiente virtuale (interfacce diegetiche), mantenendo sullo schermo solo il comando per l'apertura di un eventuale menu contestuale.

Alcuni utenti hanno suggerito di raggruppare tutte le legende su un'unica parete: questa ipotesi introduce tuttavia una serie di problematiche legate alla variabilità degli elementi da rappresentare. Ad esempio, la selezione personalizzata dei colori e dei suoni può generare elenchi di dimensione variabile da 2 a n elementi, con conseguente incremento di difficoltà nella gestione dei casi possibili.

In alternativa, si è pensato di rendere le legende più accessibili, aggiungendo un modo per poterle consultare in qualsiasi momento durante l'esplorazione della stanza. Nella versione desktop si tratterebbe di farle apparire a schermo in seguito alla pressione di un tasto, mentre in VR si potrebbe implementare un wrist menu.

Dal punto di vista della leggibilità, la legenda del *Livello di riempimento* necessita di qualche aggiustamento: si potrebbe valutare l'uso di una barra verticale oppure indicazioni esplicite che utilizzino le sagome degli ologrammi stessi come esempi visivi dei diversi livelli assumibili.

Infine, le legende dovrebbero evolvere verso un paradigma interattivo: le icone del *Pitch sonoro* potrebbero emettere un sample sonoro in seguito all'interazione (funzionalità intuitivamente attesa dagli utenti durante i test), fungendo inoltre da punto di accesso per le operazioni di filtraggio descritte nella sezione 6.4.

## 6.4 Interazione

Va innanzitutto eliminato il bug dei collider, che impedisce di interagire con gli ologrammi quando vengono puntati dal petto in su.

Per quanto riguarda il *Suono*, dato che distinguere la fonte sonora all'interno di un cluster si è rivelata un'operazione difficile, alcuni utenti hanno suggerito di aggiungere una logica che permetta di sentire solo il suono emesso dal singolo ologramma che viene puntato: si tratta un'opzione valida che andrebbe considerata in fase di sviluppo della versione 2.0 di *inPersona*.

### 6.4.1 TopView

Per risolvere sia il problema della navigazione sia quello relativo alla scarsa visibilità del riempimento, si suggerisce di implementare la funzionalità di *drag-and-drop* così come la implementano gli editor di Unity e Blender, che consiste nel premere un tasto e trascinare per cambiare inquadratura, oppure premere un tasto diverso

e trascinare per cambiare angolazione. Quest'interazione andrebbe poi adattata ai controller del visore.

Nella versione attuale del prototipo, la *TopView* è completamente sprovvista delle mappature sonore. Una volta implementata la logica che permette di sentire il suono emesso dagli ologrammi puntati, questa va aggiunta anche alla visuale dall'alto, insieme alla possibilità di cliccare sugli NPC per farli parlare.

Infine, un'indicazione che manca e che dovrebbe essere aggiunta è quella sulla posizione inizialmente occupata dall'utente nel momento in cui si sposta in *TopView*, di cui al momento non si ha informazione.

## 6.5 Funzionalità

La parte più significativa della versione 2.0 sarà la progettazione delle funzionalità relative a filtraggio, raggruppamento e informazioni numeriche, esplicitamente richieste da molti tester.

Per informazioni numeriche si intende, ad esempio, il numero totale di ologrammi che di volta in volta popola la stanza, così come i subtotali relativi alle mappature: la prima fase del lavoro futuro sarà caratterizzata dalla decisione su quali e quante di queste informazioni visualizzare, e su come visualizzarle (statiche o a discrezione dell'utente? A schermo o tramite popup/wirst menu?).

Riguardo invece alle operazioni di filtraggio, una base di partenza potrebbe essere la funzionalità *Display*, finora non discussa perché rimasta in stato di *work-in-progress* e impiegata principalmente a fini di debug.

Nella sua versione attuale, questa mappatura permette di trattare separatamente le colonne su cui viene attivata: se la modalità è impostata su "show", queste sono le uniche a essere mostrate, altrimenti vengono rimosse prima ancora di popolare i buffer ECS. In prospettiva, questo funzionamento dovrà essere adattato per funzionare a runtime in modo che gli ologrammi vengano in qualche modo oscurati e non rimossi del tutto.

## 6.6 Estetica e ottimizzazione

Un ultimo aspetto riguarda lo stile complessivo dell'applicazione, che nella versione 2.0 andrà consolidato verso un'identità visiva più riconoscibile, intervenendo in modo coerente su palette cromatica, tipografia e soluzioni grafiche, in parallelo a un sound design più caratterizzato.

Dal punto di vista prestazionale, sono inoltre possibili ulteriori ottimizzazioni, ad esempio l'introduzione di sistemi di LOD (Level of Detail) per gli ologrammi al fine di alleggerire il carico sulla GPU.



## Capitolo 7

# Conclusioni

L'obiettivo di questo lavoro era esplorare le potenzialità della realtà virtuale come strumento per la visualizzazione e l'interpretazione di dataset multidimensionali, indagando in particolare se e in che misura un approccio basato sull'*umanizzazione* dei dati potesse favorire la comprensione e il coinvolgimento degli utenti. In questa prospettiva è stato progettato e sviluppato *inPersona*, un prototipo che traduce le righe di un dataset in ologrammi umani, ciascuno caratterizzato da attributi visivi e sonori derivati direttamente dai valori contenuti nel file di input.

Nel corso della tesi sono stati affrontati tre aspetti principali: la definizione del concept e del modello di mappatura tra dati e rappresentazione audiovisiva, l'implementazione tecnica del sistema, e la valutazione preliminare dell'esperienza tramite test con utenti. I risultati ottenuti mostrano come l'approccio proposto sia percepito come efficace nel rendere più immediata la relazione con i dati sulle persone. Allo stesso tempo, i test hanno messo in luce criticità rilevanti in termini di onboarding, interazione, gestione dell'input e leggibilità di alcune componenti chiave dell'interfaccia.

Dal punto di vista dei contributi, il lavoro fornisce innanzitutto una proposta progettuale per la rappresentazione di individui-dato in ambiente immersivo, che integra in modo coerente componenti visive e sonore. Sul piano implementativo, la tesi documenta un'architettura modulare basata su ECS, in grado di gestire un numero consistente di entità e di collegare il caricamento di un file CSV a una popolazione dinamica della scena.

Un elemento emerso con forza dai test è la rilevanza inattesa della versione desktop, inizialmente concepita come semplice strumento di supporto allo sviluppo. Nonostante la minore immersione, gli utenti l'hanno percepita come sufficientemente efficace e, in alcuni casi, più versatile per l'esplorazione dei dati. Questo risultato

ha portato a ripensare l'evoluzione del progetto in termini di sviluppo parallelo: la roadmap proposta prevede una versione 2.0 di *inPersona* che includa sia una build desktop sia una build VR, da sviluppare e testare in modo coordinato. Guardando oltre la versione 2.0, una possibile evoluzione verso una versione 3.0 apre lo spazio per introdurre dinamiche più complesse, ad esempio in termini di gamification.

In conclusione, la tesi dimostra come la realtà virtuale e, più in generale, gli ambienti immersivi possano costituire un terreno fertile per sperimentare nuove forme di visualizzazione dei dati capaci di coinvolgere l'utente su un piano più personale. Pur nella sua natura prototipale, *inPersona* rappresenta un passo in questa direzione, offrendo spunti concreti sia per ulteriori ricerche accademiche sia per possibili applicazioni in contesti reali (formazione, comunicazione scientifica, supporto decisionale). Il lavoro svolto non esaurisce le domande aperte, ma definisce un quadro di riferimento e una base tecnica su cui costruire iterazioni future, con l'obiettivo di avvicinare sempre di più i dati alle persone e le persone ai dati.

# Sigle e acronimi

## **NPC**

Non-Playable Character

## **VR**

Virtual Reality

## **FPS**

First Person Shooter

## **TPS**

Third Person Shooter

## **ECS**

Entity Component System

## **fps**

frames per second

## **DOTS**

Data-Oriented Technology Stack

## **CSV**

Comma-Separated Values

## **HUD**

Heads-Up Display

**MOT**

Map Overview Teleport

**CPU**

Central Processing Unit

**IDE**

Integrated Development Environment

**XML**

Extensible Markup Language

**UI**

User Interface

**WAV**

Waveform Audio File Format

**TTS**

Text To Speech

**URP**

Universal Render Pipeline

**SUS**

System Usability Scale

**AR**

Augmented Reality

**CRM**

Customer Relationship Manager

**GPU**

Graphics Processing Unit

# Indice delle tabelle

3.1	Comandi . . . . .	22
4.1	Mappature . . . . .	41
4.2	Funzionalità avanzate . . . . .	41
4.3	Setup del dataset sulla salute mentale degli studenti . . . . .	42
4.4	Assegnazione scripts ai GameObject . . . . .	46
5.1	Partecipanti ai test di usabilità . . . . .	52
5.2	Elenco dei task e delle valutazioni . . . . .	54
5.3	Media risposte alle domande post-task 2. . . . .	58
5.4	Media risposte alle domande post-task 3. . . . .	59
5.5	Media risposte alle domande post-task 4. . . . .	60
5.6	Media risposte alle domande post-task 5. . . . .	61
5.7	Media risposte alle domande post-task 6. . . . .	62
5.8	Media risposte alle domande post-task 7. . . . .	63
5.9	Media risposte alle domande post-task 8. . . . .	64
5.10	Media risposte alle domande post-task 9. . . . .	65



# Indice delle figure

1.1	Andamento della valutazione soggettiva del salvataggio di vite in funzione del numero di vite a rischio [5, Fig. 3]. . . . .	2
2.1	Esempio di uso della profondità per dati a tre dimensioni [6] . . . . .	6
2.2	Esempio di uso della grandezza per dati a tre dimensioni [6] . . . . .	6
2.3	Esempio di uso combinato di profondità e colore per dati a quattro dimensioni [6] . . . . .	6
2.4	Esempio di uso combinato di grandezza e colore per dati a quattro dimensioni [6] . . . . .	6
2.5	Esempio di uso combinato di colore, profondità e grandezza per dati a cinque dimensioni [6] . . . . .	6
2.6	Esempio di uso combinato di colore, profondità, grandezza e forma per dati a sei dimensioni [6] . . . . .	6
2.7	Esempio di dashboard interattiva in Microsoft Power BI. . . . .	7
2.8	Modalità di navigazione in FPS [10]. . . . .	8
2.9	Modalità di navigazione in TPS [10]. . . . .	8
2.10	Modalità di visualizzazione in 2D [10]. . . . .	8
2.11	DataVizVR [11]. . . . .	9
2.12	LookVR [12]. . . . .	9
2.13	Un grafico a barre diventa una parete da arrampicata in LookVR [10, 12]. . . . .	10
2.14	<i>Mirror City</i> di BadVR [13, 14]. . . . .	10
2.15	Ambiente virtuale di DataGarden [15, Fig. 6.b]. . . . .	11

2.16 Menu di selezione della visuale nella prima versione di DataGarden [15, Fig. 4.a - 4.e]. . . . .	12
2.17 <i>Fly mode</i> nella seconda versione di DataGarden [15, Fig. 6.a]. . . . .	12
2.18 Le quattro fasi della MOT [16, Fig. 3.a - 3.d]. . . . .	13
2.19 Informazioni dettagliate mostrate in seguito all'hover in DataGarden [15, Fig. 8.a]. . . . .	14
3.1 Ambiente virtuale di DataGarden [15] . . . . .	18
3.2 Ambiente virtuale di <i>inPersona</i> . . . . .	18
3.3 Esempio di NPCs caratterizzati da colori, modelli 3D e livelli di riempimento diversi. . . . .	19
3.4 Gli NPCs occupano posizioni diverse nella stanza in base ai valori scelti per gli assi x e y. . . . .	19
3.5 Effetto del cambiamento del filtro. . . . .	20
3.6 Le quattro fasi della MOT [16] in <i>inPersona</i> . . . . .	21
3.7 In prossimità di un NPC, il cursore diventa un fumetto per suggerire l'interazione di <i>speech</i> all'utente. . . . .	22
3.8 In <i>Dead Space</i> , salute, munizioni e altre informazioni sono indicate sul corpo del personaggio [20, 21]. . . . .	23
3.9 Diario con i progressi di gioco in <i>The Last of Us</i> [22]. . . . .	23
3.10 Legende di livello di riempimento, gradiente di colore, pitch sonoro (in azzurro) e asse y (in verde). . . . .	23
3.11 Legenda del Filtro (in azzurro) e dell'asse x (in verde). . . . .	23
3.12 Legenda dei colori. . . . .	24
3.13 Legenda dei suoni. . . . .	24
3.14 Titolo del dataset visualizzato. . . . .	24
3.15 L'HUD di <i>The Witcher 3: Wild Hunt</i> mostra una serie di informazioni, tra cui le <i>command hints</i> (in azzurro) [23]. . . . .	25
3.16 UI di <i>inPersona</i> : comandi (in azzurro) e legenda dello <i>speech</i> (in verde). . . . .	25
3.17 Personaggi <i>low poly</i> in <i>SUPERHOT VR</i> [25]. . . . .	26
3.18 Ologrammi nemici durante una scena di allenamento in <i>The Hunger Games: Catching Fire</i> [26]. . . . .	26

3.19 Dataset "Bullying in schools" visualizzato su <i>inPersona</i> . . . . .	27
4.1 Schema esplicativo dell'architettura ECS [27]. . . . .	29
4.2 Senza Job System lo <i>Spawner System</i> occupa 22ms sul main thread. . . . .	31
4.3 Con il Job System lo <i>spawnJob</i> viene parallelizzato su due worker threads e occupa in totale 0.593ms. . . . .	31
4.4 Flusso logico di <i>inPersona</i> . . . . .	32
4.5 Configurazione ideale: valori diversi in un'unica colonna. . . . .	33
4.6 Configurazione non ideale: valori booleani su colonne diverse. . . . .	33
4.7 Caso d'uso: dataset sulla salute mentale degli studenti (visualizzato sottoforma di tabella). . . . .	34
4.8 Filtro non attivo: tutti gli studenti sono presenti nella stanza contemporaneamente. . . . .	35
4.9 Filtro attivo: studenti divisi per corso di studi. . . . .	35
4.10 Studente poco soddisfatto. . . . .	35
4.11 Studente abbastanza soddisfatto. . . . .	35
4.12 Studente molto soddisfatto. . . . .	35
4.13 Studente poco riposato. . . . .	36
4.14 Studente abbastanza riposato. . . . .	36
4.15 Studente riposato. . . . .	36
4.16 Studente. . . . .	37
4.17 Studentessa. . . . .	37
4.18 Posizione spaziale non attiva: ologrammi posizionati casualmente. . . . .	37
4.19 Posizione attiva su un solo asse: le relazioni sociali crescono lungo x. . . . .	38
4.20 Posizione attiva sui due assi: le relazioni sociali crescono lungo x e la pressione accademica lungo y. . . . .	38
4.21 Applicazione dell'ordine personalizzato. . . . .	39
4.22 Definizione delle condizioni per l'assegnazione dei modelli 3D in base a genere ed età. <i>Questo codice non fa riferimento all'esempio sugli studenti finora discusso.</i> . . . . .	40
4.23 Messaggi in console generati dal TTS. . . . .	44

4.24 Gerarchia della scena in Unity. Il GameObject <i>UI</i> contiene gli elementi della UI. . . . .	45
4.25 Elenco dei modelli 3D (prefabs) nel Project Manager di Unity. . . .	47
4.26 Cubemap della galassia usata per lo sfondo. . . . .	47
4.27 Shader Graph del materiale degli ologrammi. I valori delle variabili <i>Fill</i> e <i>Color</i> vengono impostati dinamicamente via script dagli Apply System, come spiegato nella relativa sezione. . . . .	48
4.28 Legenda del pitch: versione 1.0 . . . . .	48
4.29 Legenda del pitch: versione 1.1 . . . . .	48
4.30 Feedback cursore: versione 1.0 . . . . .	49
4.31 Feedback cursore: versione 1.1 . . . . .	49
4.32 Schermata controlli: versione 1.0 . . . . .	49
4.33 Schermata controlli: versione 1.1 . . . . .	49
5.1 Setup delle sessioni di testing. . . . .	52
5.2 Esiti Task 1 . . . . .	57
5.3 Esiti Task 2 . . . . .	58
5.4 Esiti Task 3 . . . . .	59
5.5 Esiti Task 4 . . . . .	60
5.6 Esiti Task 5 . . . . .	61
5.7 Esiti Task 6 . . . . .	62
5.8 Esiti Task 7 . . . . .	63
5.9 Esiti Task 8 . . . . .	64
5.10 Esiti Task 9 . . . . .	65
5.11 Esiti questionario SUS . . . . .	65

# Bibliografia

- [1] *What Is Data Visualization? Definition, Examples, And Learning Resources*. URL: <https://www.tableau.com/visualization/what-is-data-visualization> (cit. a p. 1).
- [2] *A Brief History of Data Visualization: From Maps to BI*. 2025. URL: <https://insightsoftware.com/blog/a-brief-history-of-data-visualization/#the-late-20th-century%3a-the-computer-age> (cit. a p. 1).
- [3] Li Q. «Overview of Data Visualization». In: *Embodying Data*. 2020, pp. 17–47. DOI: 10.1007/978-981-15-5069-0\_2 (cit. a p. 1).
- [4] *Measuring Nonuse Damages Using Contingent Valuation: An Experimental Evaluation of Accuracy*. 2<sup>a</sup> ed. RTI Press publication No. BK-0001-1009. Research Triangle Park, NC: RTI Press, 2010 (cit. a p. 2).
- [5] «Psychic Numbing and Mass Atrocity». In: *The Behavioral Foundations of Public Policy*. A cura di Eldar Shafir. Princeton, NJ: Princeton University Press, 2013, pp. 126–142 (cit. alle pp. 2, 3).
- [6] Dipanjan Sarkar. *The Art of Effective Visualization of Multi-dimensional Data*. 2018. URL: <https://medium.com/data-science/the-art-of-effective-visualization-of-multi-dimensional-data-6c7202990c57> (cit. alle pp. 5, 6).
- [7] Qiyue Zhang. «The Impact of Interactive Data Visualization on Decision-Making in Business Intelligence». In: *Advances in Economics Management and Political Sciences* 87.1 (2024), p. 167. DOI: 10.54254/2754-1169/87/20241056 (cit. a p. 7).
- [8] *Dashboards done right*. URL: <https://www.tableau.com/dashboard> (cit. a p. 7).
- [9] *Introduction to dashboards for Power BI designers*. URL: <https://learn.microsoft.com/it-it/power-bi/create-reports/service-dashboards> (cit. alle pp. 7, 70).

- 
- [10] *Uncharted Territory: Diving into Data Visualization in Virtual Reality*. URL: <https://studio.knightlab.com/results/exploring-data-visualization-in-vr/uncharted-territory-datavis-vr/> (cit. alle pp. 8, 10).
- [11] *DatavizVR Demo*. URL: [https://store.steampowered.com/app/551960/DatavizVR\\_Demo/](https://store.steampowered.com/app/551960/DatavizVR_Demo/) (cit. alle pp. 9, 20).
- [12] *LookVR*. URL: <https://store.steampowered.com/app/595490/LookVR/> (cit. alle pp. 9, 10, 20).
- [13] *Who's Offering Data Visualization VR?* 2018. URL: <https://datavizcatalogue.com/blog/whos-offering-data-visualization-vr/> (cit. a p. 10).
- [14] *BadVR*. URL: <https://devpost.com/software/badvr#updates> (cit. a p. 10).
- [15] Joy Kondo. «DataGarden: Exploring Our Community in a VR Data Visualization». In: 2023. URL: <https://api.semanticscholar.org/CorpusID:260068524> (cit. alle pp. 11–14, 18, 20).
- [16] Ariel Caputo, Federico Borin e Andrea Giachetti. «A Comparison of Navigation Techniques in a Virtual Museum Scenario». In: *Proceedings of the Eurographics Workshop on Graphics and Cultural Heritage (GCH 2019)*. Geneva, Switzerland: Eurographics Association, 2019 (cit. alle pp. 13, 21).
- [17] Jie Wang, Rui Shi, Wuchen Zheng, Wei Xie, Dingwen Kao e Hai-Ning Liang. «Effect of Frame Rate on User Experience, Performance, and Simulator Sickness in Virtual Reality». In: *IEEE Transactions on Visualization and Computer Graphics* (2023). DOI: 10.1109/TVCG.2023.3247057 (cit. a p. 16).
- [18] *VR Headset Refresh Rate Comparison: The Ultimate Guide to Smooth Immersion*. 2025. URL: <https://inairspace.com/blogs/learn-with-inair/vr-headset-refresh-rate-comparison-the-ultimate-guide-to-smooth-immersion> (cit. a p. 16).
- [19] Andrea Bottino. *Realtà Virtuale: 3D UI*. Corso di Laurea Magistrale in Ingegneria del Cinema e dei Mezzi di Comunicazione, Politecnico di Torino. Materiale didattico non pubblicato. 2024 (cit. alle pp. 23, 24).
- [20] Maximillian Navarro. *The Immersive Excellence of Dead Space's UI: A Masterclass in Integrated Design*. 2023. URL: <https://medium.com/@maximillianknararro/the-immersive-excellence-of-dead-spaces-ui-a-masterclass-in-integrated-design-4368492c1731> (cit. a p. 23).
- [21] *Dead Space*. Video game, EA Redwood Shores, Electronic Arts. 2008 (cit. a p. 23).
- [22] *The Last of Us Part II*. Video game, Sony Interactive Entertainment. 2020 (cit. a p. 23).

- [23] *The Witcher 3: Wild Hunt*. Video game, Project Red. 2015 (cit. a p. 25).
- [24] Jisu Park, Nicoletta Adamo-Villani e Robert W. Proctor. «Does a Character’s Visual Style Affect Audience Empathy and Sympathy?» In: *Human Interface and the Management of Information. Visual Information and Knowledge Management*. Vol. 11569. Lecture Notes in Computer Science. Springer, 2019, pp. 476–488. DOI: 10.1007/978-3-030-22660-2\_35 (cit. a p. 26).
- [25] *SUPERHOT VR*. Video game, SUPERHOT Team. 2016 (cit. a p. 26).
- [26] *The Hunger Games: Catching Fire*. Film. United States, 2013 (cit. a p. 26).
- [27] Username: Be Content. *Unlocking Performance and Scalability: A Deep Dive into Unity’s ECS and DOTS*. 2023. URL: <https://medium.com/@be.content23/unlocking-performance-and-scalability-a-deep-dive-into-unitys-ecs-and-dots-1c6cf6fb240c> (cit. alle pp. 29, 30).
- [28] Brian Will. *Introduction to the Data-Oriented Technology Stack for advanced Unity developers*. 2024. URL: <https://unity.com/resources/introduction-to-dots-ebook> (cit. alle pp. 30, 42).
- [29] *inPersona*. URL: <https://github.com/MariagraziaGiunta/inPersona.git> (cit. a p. 30).
- [30] Hugo Cardoso (Code Monkey). *Learn Unity DOTS*. URL: [https://unitycodemonkey.teachable.com/p/make-a-game-with-unity-dots?coupon\\_code=LEARN-DOTS](https://unitycodemonkey.teachable.com/p/make-a-game-with-unity-dots?coupon_code=LEARN-DOTS) (cit. a p. 30).
- [31] Hugo Cardoso (Code Monkey). *Learn Unity DOTS! (FREE Tutorial Course)*. URL: <https://youtu.be/1gSnTlUjs-s?t=8159> (cit. a p. 30).
- [32] *BlenderKit*. URL: <https://www.blenderkit.com/> (cit. a p. 46).
- [33] *Nebula Skyboxes*. URL: <https://assetstore.unity.com/packages/2d/textures-materials/sky/nebula-skyboxes-219924> (cit. a p. 46).
- [34] *Freesound*. URL: <https://freesound.org/> (cit. a p. 46).
- [35] *Kaggle*. URL: <https://www.kaggle.com/> (cit. a p. 47).