

# POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria del Cinema e dei  
Mezzi di Comunicazione



Tesi di Laurea Magistrale

## La Realtà Virtuale Immersiva per la prototipazione di postazioni di lavoro industriali: il caso applicativo dei Low Cost Automation

Relatore

Prof. Andrea SANNA

Candidato

Nicola GRAVINA

Anno Accademico 2021/2022

## Abstract

Le tecnologie di realtà virtuale (VR) hanno cambiato il modo di comunicare e di lavorare delle persone, offrendo la possibilità di creare ambienti virtuali esplorabili unendo tre componenti, strettamente correlate tra loro: interazione, immersione, presenza. Il grado di immersione dipende fortemente dalla tecnologia adottata per la simulazione virtuale: i dispositivi che offrono il maggior livello di coinvolgimento sono gli Head-Mounted Display (HMD), visori che nascondono il mondo reale all'utente e che contengono sistemi per la rilevazione dei movimenti. In questo modo i movimenti dell'utente all'interno di uno spazio fisico vuoto vengono mappati in coordinate virtuali.

La VR ha notevolmente contribuito alla trasformazione digitale delle aziende, il cui obiettivo finale è quello di aumentare la produttività e la redditività, in accordo con i principi su cui si basa l'Industria 4.0, introdotti nel 2011 dal governo tedesco alla fiera di Hannover. Questo nuovo paradigma industriale influenza direttamente l'intero ciclo di vita del prodotto e modificando l'ambiente di lavoro. La VR consente di abbreviare i tempi necessari per progettare, testare e controllare la qualità dei prodotti, dall'inizio del processo di sviluppo fino alla loro commercializzazione. In passato, diversi studi hanno permesso di comprendere quanto la VR semplifichi attività industriali quali formazione e allenamento, giungendo alla conclusione che le prestazioni complessive che questi sistemi offrono risultano migliori rispetto ai metodi tradizionali grazie alla maggiore interattività e alla visualizzazione stereoscopica, partendo da un set di modelli 3D, è possibile costruire intere catene di lavorazione in digitale, adattarle alle esigenze dell'operatore ed interagire con esse: questo approccio prende il nome di Smart Product Design.

Questa tesi è finalizzata alla ricerca e sviluppo di un'applicazione IVR che semplifichi e velocizzi la progettazione e l'arredamento di postazioni di lavoro, al fine di valutarne l'efficienza, la produttività e l'ergonomia, utilizzando il motore grafico open source Unity3D. Nello specifico, l'applicativo affronta la sfida della prototipazione virtuale di strutture per la Low Cost Automation (LCA), ovvero un insieme di apparati meccanici per la movimentazione di pezzi che, sfruttando in maniera creativa forze quali gravità o attrito, permettono di minimizzare gli sforzi fisici degli operatori.

Il sistema proposto permette all'utente di assemblare un LCA attingendo ad una libreria predefinita di oggetti e offrendo un sistema di aggancio magnetico tra componenti per semplificare e velocizzare la prototipazione. Tramite un visore HTC Vive Pro, l'operatore può visualizzare un ambiente virtuale in modalità immersiva e interagire con gli oggetti virtuali tramite i controller HTC Vive. Per spostare un oggetto è sufficiente afferrarlo, attivando con un tasto l'apposita funzionalità,

e spostando la mano nello spazio virtuale, l'oggetto ne segue il movimento; per ingrandirlo o rimpicciolirlo è sufficiente spostare i controller, allontanando o avvicinando le mani.

Per valutare l'applicativo in termini di usabilità ed efficienza sono stati svolti dei test utente, simulando la creazione di un LCA a partire da un modello virtuale. Sono stati raccolti sia dati oggettivi (tempo di esecuzione, numero di errori commessi) che soggettivi (tramite la compilazione di un questionario). L'analisi dei risultati ha permesso di valutare il sistema di proposto e di identificarne limiti e possibili sviluppi futuri.



# Indice

<b>Glossario</b>	VI
<b>Acronimi</b>	VIII
<b>Elenco delle figure</b>	X
<b>Elenco delle tabelle</b>	XIII
<b>1 Introduzione</b>	1
1.1 Realtà Virtuale e Industria 4.0 . . . . .	1
1.2 Obiettivo della tesi . . . . .	6
1.3 Low Cost Automation . . . . .	6
1.4 CIM 4.0 e Stellantis . . . . .	8
1.5 Organizzazione dei capitoli . . . . .	8
<b>2 VR: Concetti generali</b>	9
2.1 Le origini . . . . .	9
2.1.1 Hype Cycle di Gartner . . . . .	14
2.1.2 Il futuro della VR . . . . .	15
2.2 Reality-Virtuality Continuum . . . . .	16
2.3 Architetture input . . . . .	17
2.4 Architetture output . . . . .	21
2.5 Metafore d'interazione . . . . .	24
<b>3 VR e I4.0: Stato dell'arte</b>	29
3.1 VR per la formazione e l'assemblaggio . . . . .	29
3.2 VR per la prototipazione . . . . .	35
3.2.1 Ikea VR Experience . . . . .	36
3.2.2 Trattazioni accademiche . . . . .	37
3.2.3 Applicazioni industriali . . . . .	39

<b>4</b>	<b>Tecnologie</b>	<b>43</b>
4.1	Hardware . . . . .	45
4.1.1	HTC Vive Pro . . . . .	45
4.1.2	HTC Vive Controller . . . . .	47
4.2	Software . . . . .	48
4.2.1	Unity3D . . . . .	48
4.2.2	Visual Studio . . . . .	53
4.2.3	SteamVR Unity plug-in . . . . .	53
<b>5</b>	<b>Sviluppo e design</b>	<b>59</b>
5.1	Impostazioni generali . . . . .	59
5.2	Controllo del sistema . . . . .	61
5.3	Classi di oggetti . . . . .	64
5.3.1	Oggetti singoli . . . . .	65
5.3.2	Oggetti composti . . . . .	66
5.3.3	Strutture complesse . . . . .	71
5.4	Salvataggio e arresto del sistema . . . . .	75
<b>6</b>	<b>Risultati e analisi</b>	<b>77</b>
6.1	Test . . . . .	78
6.1.1	System Usability Survey . . . . .	81
6.1.2	NASA-Task Load Index . . . . .	82
6.2	Risultati . . . . .	83
6.2.1	Risultati oggettivi . . . . .	83
6.2.2	Risultati soggettivi . . . . .	84
<b>7</b>	<b>Conclusione e sviluppi futuri</b>	<b>89</b>
<b>A</b>	<b>Questionario</b>	<b>91</b>

# Glossario

**Stereoscopia** Percezione del volume e della tridimensionalità del mondo circostante, delle distanze relative degli oggetti e della rispettiva collocazione spaziale, e si ha in conseguenza della visione binoculare. L'occhio destro e l'occhio sinistro forniscono una differente visione prospettica bidimensionale del mondo: il sistema visivo combina le due prospettive e forma un'immagine detta stereoscopica. La percezione di profondità è rinforzata da una serie di indizi (depth cues) nelle immagini, quali ad esempio occlusioni e ombre. La visione binoculare coinvolge due processi: parallasse e stereopsi.

**Parallasse** Processo in cui gli occhi catturano una stessa scena da due posizioni differenti. Quando l'occhio fissa un punto F di un oggetto, l'angolo tra l'asse della visuale e le linee (pupille->F) si chiama angolo di convergenza e dipende dalla distanza dell'oggetto e dall'IDP.

**Stereopsi** Processo in cui il cervello unisce le due immagini ed elabora le informazioni di profondità, in funzione dell'angolo di convergenza e dell'IDP.

**Inter-Pupillary Distance IDP** Distanza tra le pupille, varia per ogni persona tra i 53-73 mm.

**Field Of View FOV** Caratteristica degli esseri viventi definita come la porzione di spazio che un occhio, restando immobile, può percepire intorno. Estendendo tale definizione ai display visuali, esso corrisponde all'angolo solido mostrato della visuale dell'utente, mostrato dal visore. Il FOV di un essere umano è circa 170° orizzontali e 120° verticali con un solo occhio, 220° orizzontali e 120° verticali per due occhi. In quest'ultimo caso, nell'area di sovrapposizione (circa 120° orizzontali) avviene il processo di ricostruzione tridimensionale. Questo parametro può variare in base ai dispositivi utilizzati e, qualora l'area di sovrapposizione per entrambi gli occhi fosse inferiore a 30°, risulterebbe difficile percepire la stereoscopia.

**Degrees Of Freedom DOF** Libert  di movimento di un corpo rigido nello spazio 3D. In particolare, un corpo   libero di cambiare posizione (traslazione) e orientamento (rotazione) attorno a tre assi perpendicolari  $X, Y, Z$ . In ambito VR, il DOF permette una classificazione dei dispositivi di input.

**Rendering** Processo di generazione di un'immagine sintetica mediante programmi di grafica computerizzata tridimensionale che partono dalla descrizione matematica di una scena, interpretata da algoritmi.

**Frame Rate** Frequenza con cui le immagini vengono visualizzate ed   rappresentato dal numero di frame per secondo (FPS). Normalmente il frame rate non dipende dal tipo di display utilizzato bens  dall'abilit  di rendering grafico dell'hardware e del software e dalla complessit  del sistema VR. I visori moderni (Oculus Rift, HTC Vive) lavorano a una frequenza standard di 90 Hz. Mantenere un parametro alto permette di evitare effetti indesiderati come la nausea dovuta al movimento (motion sickness) e di rendere l'esperienza virtuale il pi  fluida possibile agli occhi dell'utente.

**Refresh Rate** Frequenza con cui le immagini vengono ridisegnate su un display. Il refresh rate include anche la ripetizione della stessa immagine, a differenza del frame rate che invece indica soltanto il numero di nuove immagini al secondo trasmesse da una sorgente.

**Level Of Detail LOD** Livello di dettaglio   un parametro che ha il compito di definire con precisione il livello di approfondimento delle informazioni di varia natura che sono contenute all'interno del modello.

**Texture** Immagine 2D, in formato raster, applicata su una o pi  facce di un modello 3D. In particolare l'immagine viene vista come una griglia in cui ad ogni punto (pixel) viene associato un colore specifico.

**Precisione (*Accuracy*)** Differenza tra la posizione dell'oggetto e quella misurata dal sensore.

**Rumore (*Jitter*)** Variazione istantanea del valore misurato dal sensore quando l'oggetto   stazionario.

**Deriva (*Drift*)** Aumento dell'errore fisso introdotto nella misura dal sensore, in funzione del tempo.

**Latenza (*Latency*)** Ritardo di risposta del sensore nel misurare una variazione di posizione o orientamento dell'oggetto.

# Acronimi

**2D** *Bi-dimensional*, Bidimensionale

**3D** *Three-dimensional*, Tridimensionale

**AR** *Augmented Reality*, Realtà aumentata

**API** *Application Programming Interface*

**AV** *Augmented Reality*, Realtà aumentata

**CIM 4.0** *Competence Industry Manufacturing 4.0*

**CPU** *Central Processing Unit*, Unità centrale di elaborazione

**CAD** *Computer-Aided Drafting*, Disegno tecnico assistito dall'elaboratore

**CAVE** *Cave Automatic Virtual Environment*

**DOF** *Degrees Of Freedom*, Gradi di libertà

**FOV** *Field Of View*, Campo visivo

**FPS** *Frame Rate*, Frequenza dei fotogrammi

**GPU** *Graphics Processing Unit*, Unità di elaborazione grafica

**GUI** *Graphical User Interface*, Interfaccia grafica

**HMD** *Head-Mounted Display*, Visore con schermo montato sulla testa

**IDP** *Inter-Pupillary Distance*, Distanza interpupillare

**I4.0** *Industry 4.0*, Industria 4.0

**IoT** *Internet of Things*

**IVR** *Immersive Virtual Reality*, Realtà virtuale immersiva

**LCA** *Low Cost Automation*

**LCD** *Liquid Crystal Display*, display a cristalli liquidi

**LED** *light-emitting diode*, diodo ad emissione di luce

**LOD** *Level Of Detail*, Livello di dettaglio

**MR** *Mixed Reality*, Realtà mista

**PMI** Piccola-Media Impresa

**RAM** *Random Access Memory*, Memoria ad accesso casuale

**UI** *User Interface*, Interfaccia utente

**UX** *User Experience*, Esperienza dell'utente

**VR** *Virtual Reality*, Realtà virtuale

# Elenco delle figure

1.1	Parametri di un sistema VR . . . . .	2
1.2	Desktop VR . . . . .	2
1.3	Simulatore di volo . . . . .	2
1.4	VR immersiva . . . . .	3
1.5	Le quattro rivoluzioni industriali . . . . .	4
1.6	Le tecnologie abilitanti di I4.0 . . . . .	5
1.7	Low cost automation . . . . .	7
2.1	Sensorama, 1962 . . . . .	10
2.2	The Sword of Damocles, 1968 . . . . .	11
2.3	CAVE . . . . .	13
2.4	Oculus Rift, HTC Vive, PlayStation VR . . . . .	13
2.5	VR Continuum, 1994 . . . . .	16
2.6	Interazione tra utente e sistema VR . . . . .	17
2.7	Outside looking in e Inside looking out . . . . .	19
2.8	Pipeline di rendering grafico . . . . .	22
2.9	Pipeline di rendering tattile . . . . .	23
2.10	Real walking e barriere di sicurezza . . . . .	25
2.11	Virtual hand . . . . .	26
2.12	Raycasting . . . . .	26
2.13	Interfaccia non diegetica, meta-diegetica, diegetica . . . . .	27
3.1	Rappresentazione dell'assemblaggio virtuale della pompa, 1999 . . . . .	30
3.2	Modellazione basata sui vincoli geometrici . . . . .	31
3.3	Modellazione basata sulla fisica . . . . .	32
3.4	Gazal-1, 2010 . . . . .	33
3.5	Schermate dell'ambiente di assemblaggio virtuale di Gazal-1, 2010 . . . . .	33
3.6	Architettura del sistema IVR, Gazal-1, 2010 . . . . .	34
3.7	Sistema di allenamento VR e AR, 2012 . . . . .	35
3.8	IKEA VR Experience, 2016 . . . . .	36
3.9	Lusetti M., Scena 2D, 2021 . . . . .	37

3.10	Lusetti M., Scena 3D, 2021 . . . . .	37
3.11	Soluzione definitiva di progetto e realizzazione del sistema 4 DOF, 2022 . . . . .	38
3.12	Plancia modellata per i test e riproduzione virtuale del laboratorio, 2022 . . . . .	38
3.13	BMW Group Production, Vista 1 . . . . .	39
3.14	BMW Group Production, Vista 2 . . . . .	40
3.15	R3DT . . . . .	41
3.16	Ergo-Check, R3DT . . . . .	41
3.17	Interfaccia Tecnomatix Process Simulate . . . . .	42
4.1	Visori e relative caratteristiche . . . . .	44
4.2	Architettura del sistema . . . . .	44
4.3	HTC Vive Pro . . . . .	45
4.4	Input di HTC Vive Controller . . . . .	47
4.5	Viste di Unity . . . . .	49
4.6	Box collider . . . . .	50
4.7	Estensione di MonoBehaviour . . . . .	52
4.8	Plug-in SteamVR su Unity Asset Store . . . . .	54
4.9	Sistema input di Steam VR . . . . .	54
4.10	Skeleton Input di SteamVR . . . . .	56
5.1	Impostazioni generali dell'ambiente virtuale . . . . .	59
5.2	Assi Gizmo al centro dell'ambiente virtuale . . . . .	60
5.3	Assets . . . . .	61
5.4	Menu principale . . . . .	62
5.5	Pointer per utilizzare una funzione . . . . .	63
5.6	Menu dell'oggetto (oggetti composti) . . . . .	64
5.7	Menu dell'oggetto (oggetti semplici) . . . . .	64
5.8	Lista di prefab . . . . .	65
5.9	Collider Points . . . . .	67
5.10	Oggetti semplici per la realizzazione di un oggetto composto . . . . .	68
5.11	Oggetti composti . . . . .	68
5.12	Realizzazione di LCA Braccio . . . . .	70
5.13	Realizzazione di LCA Ruota . . . . .	70
5.14	Freno e LCA Braccio . . . . .	71
5.15	LCA Rulliera e LCA Rulliera Completa . . . . .	71
5.16	Struttura complessa . . . . .	71
5.17	Traslazione di un componente della struttura complessa . . . . .	73
5.18	Assemblaggio di due strutture complesse . . . . .	74
5.19	Assemblaggio di due strutture complesse . . . . .	74

5.20	Assemblaggio di due strutture complesse . . . . .	74
5.21	Struttura complessa risultante . . . . .	75
5.22	File XML di salvataggio . . . . .	76
6.1	Flusso delle operazioni fondamentali per l'utilizzo dell'applicativo .	77
6.2	Tutorial . . . . .	78
6.3	Tutorial: controller con etichette . . . . .	79
6.4	Struttura del tutorial . . . . .	80
6.5	Esempio di struttura LCA . . . . .	81
6.9	Ulteriori valutazioni soggettive . . . . .	87

# Elenco delle tabelle

5.1	Oggetti singoli . . . . .	66
5.2	Oggetti composti . . . . .	69
6.1	Tempi di apprendimento del sistema e di prototipazione dell'LCA . . . . .	84
6.2	Risultati SUS . . . . .	84



# Capitolo 1

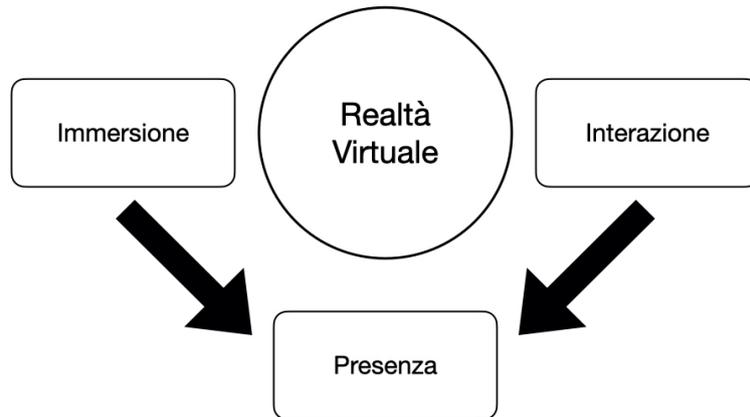
## Introduzione

### 1.1 Realtà Virtuale e Industria 4.0

Le tecnologie digitali hanno rapidamente cambiato il modo di comunicare e lavorare delle persone. In particolare, l'interesse verso il settore della realtà virtuale è cresciuto, in breve tempo, in maniera esponenziale sia a livello commerciale che professionale. La realtà virtuale (Virtual Reality, VR) consiste nella creazione di ambienti o mondi digitali in cui l'utente può interagire direttamente con questa dimensione. L'istituto di ricerca Juniper [1] definisce la VR come “la tecnologia che sfrutta degli audio e dati visuali predeterminati da una fonte digitale, creando un ambiente interattivo ed artificiale per gli utenti mediante l'utilizzo di software e hardware specifici.”

Tale tecnologia si serve di un linguaggio grafico o tridimensionale e permette di vivere dinamicamente esperienze controllate, ovvero programmate da qualcuno, i cui stimoli sono percepiti come reali da parte degli utenti. Il funzionamento della VR deriva dall'unione di tre componenti strettamente collegate tra loro: immersione, interazione, presenza.

L'interazione comprende tutte le modalità con cui l'utente entra in contatto con il mondo virtuale ed implica la simulazione in tempo reale e il coinvolgimento dei canali sensoriali umani (vista, udito, tatto). L'immersione avviene quando l'utente perde il contatto con la realtà fisica, percependo unicamente stimoli provenienti dal mondo virtuale. È necessario che tali feedback sensoriali, relativi al corpo virtuale, siano coerenti con le azioni e i cambiamenti che impongono all'ambiente (virtual embodiment) e che gli utenti tendano a rispondere nell'ambiente virtuale come nella vita reale. Il grado di immersione indica uno dei metri di successo dell'ambiente virtuale. Le modalità di immersione offerte dalla VR sono principalmente tre.



**Figura 1.1:** Parametri di un sistema VR

La VR non immersiva (Desktop VR o Window on World) consiste in tutti quei sistemi da “scrivania”, possiede una ridotta disponibilità di interfacce (monitor, mouse, joystick e tastiera), con limitata complessità dell’interazione.

I sistemi semi-immersivi forniscono un’esperienza più coinvolgente, in cui l’utente è parzialmente inserito in un ambiente virtuale. Essi sono alimentati da sistemi di calcolo grafico ad alte prestazioni spesso accoppiati a sistemi di proiezione o schermi di grandi dimensioni (es. simulatori di volo).



**Figura 1.2:** Desktop VR, Fonte: <https://bit.ly/35X1FSB>



**Figura 1.3:** Simulatore di volo, Fonte <https://bit.ly/3zD2I1k>

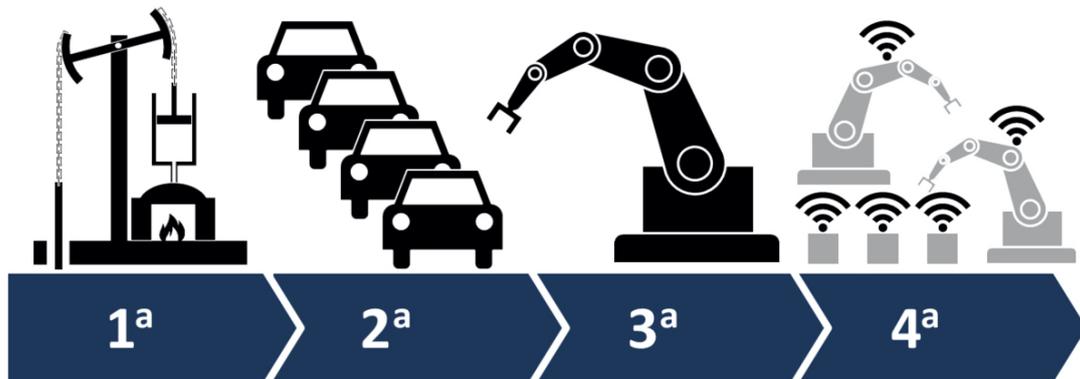
Infine, i sistemi immersivi (IVR) includono contemporaneamente tutte e tre le componenti (immersione, interazione, presenza), se pur in quantità variabile. Essi offrono visione stereoscopica in scala realistica e legano la prospettiva ai movimenti della testa dell'utente. L'utente è in grado di interagire realisticamente con l'ambiente immersivo attraverso interfacce specifiche per la manipolazione e per il controllo, restituendo feedback multimodali (audio, tattile, motorio etc.).

La presenza consiste nella sensazione di appartenenza all'ambiente virtuale. È necessario che il mondo virtuale sia realizzato in modo da massimizzare questa sensazione e che l'utente sia "predisposto" ad accettare la sensazione di presenza in tale ambiente (sospensione dell'incredulità). Per misurare il livello di presenza nel sistema virtuale è possibile effettuare valutazioni in termini quantitativi attraverso la registrazione di parametri fisiologici in relazione ai tempi di risposta dell'utente oppure valutazioni qualitative tramite questionari. Il grado di presenza è funzione del livello di immersione e interazione. Alcuni elementi che ingannano la percezione fino a credere di essere in un altro luogo sono la qualità delle informazioni sensoriali, la modellazione ed il rendering, la mobilità ed il controllo dei sensori, il controllo sull'ambiente.



**Figura 1.4:** VR immersiva, Fonte: VIVE, <https://bit.ly/32TTrK3>

I contesti applicativi di questa tecnologia sono molteplici: dal campo videoludico a quello culturale, dal settore medico a quello dell'Industria 4.0. Il termine Industria 4.0 (I4.0) è stato introdotto per la prima volta dal governo tedesco alla Fiera di Hannover nel novembre 2011 per indicare la trasformazione digitale della produzione industriale (quarta rivoluzione industriale).



**Figura 1.5:** Le quattro rivoluzioni industriali, Fonte: <https://bit.ly/3zEYf1w>

L'obiettivo di ogni rivoluzione industriale è aumentare l'efficienza degli impianti, migliorare la qualità dei prodotti e creare nuovi modelli di business. Le prime tre rivoluzioni industriali hanno avuto luogo quando sono state sfruttate, rispettivamente, la potenza del vapore (XVIII° secolo), l'elettricità (XIX° secolo), l'elettronica e l'informatica (XX° secolo) per aumentare la produttività (Tunzelmann [2]). La quarta rivoluzione sta creando un nuovo campo industriale che dipenderà dall'acquisizione e dalla condivisione dei dati lungo l'intera catena di fornitura (Brettell e Friederichsen [3]), connettendo il mondo digitale con quello fisico attraverso i sistemi cyber-fisici potenziati dall'Internet of Things (IoT). Nel 2016 Oesterreich e Teuteberg [4] hanno affermato che l'I4.0 può essere descritta come "l'aumento della digitalizzazione e dell'automazione dell'ambiente produttivo oltre a una maggiore comunicazione resa possibile dalla creazione di una catena del valore digitale". Questo nuovo paradigma industriale influenza direttamente l'intero ciclo di vita del prodotto, modificando l'ambiente di lavoro e ristrutturando il mercato. Le tecnologie che caratterizzano l'I4.0 sono:

- **Robotica:** sistemi avanzati di produzione che consistono in macchinari interconnessi, rapidamente programmabili e dotati di intelligenza artificiale (robot collaborativi o cobot);
- **Simulazioni:** possibilità di simulare nuovi processi tra macchine interconnesse prima di metterli in pratica nella realtà;
- **Integrazione orizzontale e verticale:** tutti gli step della catena del valore, dal produttore al consumatore, comunicano tra di loro e si scambiano informazioni;

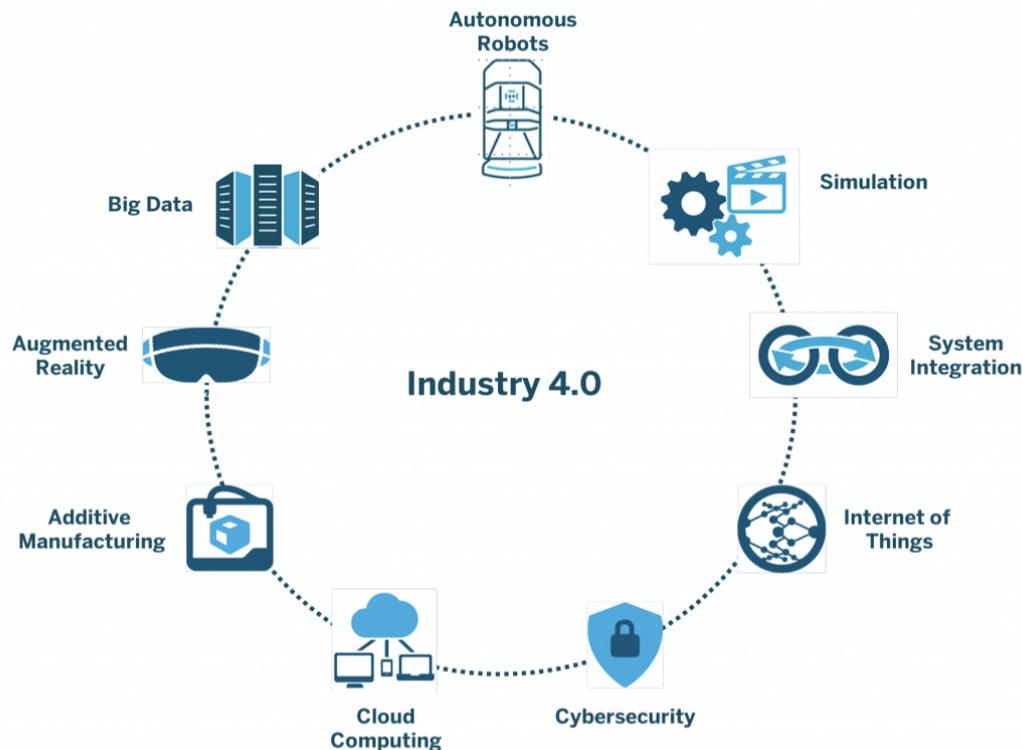


Figura 1.6: Le tecnologie abilitanti di I4.0, Fonte: <https://bit.ly/3HONwEU>

- **Internet of Things:** Internet delle “cose” (IoT) applicato all’industria, ovvero oggetti riconoscibili e intelligenti in grado di poter comunicare dati grazie all’utilizzo della rete Internet;
- **Sicurezza informatica:** sistemi che garantiscono la sicurezza delle informazioni durante le operazioni in rete e su sistemi in cloud;
- **Cloud:** gestione di elevate quantità di dati direttamente in rete (archiviazione online delle informazioni, cloud computing), attraverso interconnessioni interne ed esterne;
- **Manifattura additiva:** sistemi di produzione che aumentano l’efficienza dell’uso dei materiali (stampa 3D, fabbricazione digitale);
- **Realtà aumentata:** sistemi AR che aiutano l’operatore durante lo svolgimento del processo produttivo (es. HoloLens);

- **Big Data:** sistemi di gestione di un'ampia base di dati attraverso sistemi aperti che permettono previsioni o predizioni per l'ottimizzazione dei prodotti e dei processi produttivi.

I sistemi IVR hanno notevolmente contribuito alla trasformazione digitale della produzione industriale. L'adozione delle tecnologie VR rappresenta un segno evidente della maturità digitale di un'azienda, il cui obiettivo finale è quello di aumentare la produttività e la redditività. In particolare la realtà virtuale consente di abbreviare i tempi necessari per progettare, testare e controllare la qualità dei prodotti dall'inizio del processo di sviluppo fino alla loro commercializzazione (time-to-market). Essa permette di ridurre al minimo i rischi di incidenti e migliora la qualità dei prodotti in quanto può intervenire in ogni fase del processo di design market, rendendo l'intero processo più coinvolgente e consentendo una collaborazione tra i lavoratori più semplice, anche in remoto. Inoltre, partendo da un set di modelli 3D, è possibile costruire intere catene di lavorazione in digitale, adeguarle alle esigenze dell'operatore ed interagire con esse: tale concetto, in ottica I4.0, si chiama Smart Product Design [5].

## 1.2 Obiettivo della tesi

Questo studio ha come obiettivo la ricerca e lo sviluppo di un framework in Immersive Virtual Reality (IVR) che semplifichi e velocizzi la progettazione di postazioni di lavoro al fine di valutarne l'efficienza, la produttività e l'ergonomia, utilizzando il game engine e motore grafico open source Unity3D. Il caso d'uso scelto per validare lo spazio di lavoro, una volta arredato, è la Low Cost Automation (LCA).

## 1.3 Low Cost Automation

In ambito I4.0, l'automazione occupa un posto centrale ed i dispositivi intelligenti (macchine, robot etc.) in grado di realizzare tale compito ne sono il mezzo. Per definizione [6], l'automazione utilizza un insieme di tutte le misure volte a sostituire il lavoro umano attraverso le macchine e utilizza la tecnologia per questo scopo. Essa include, ad esempio, sistemi di movimentazione e di stoccaggio, linee di assemblaggio e di produzione. L'automazione richiede ingenti spese in termini di tempo e capitale e le industrie di piccola e media scala non possono permettersi tali investimenti, così, generalmente, evitano la strada dell'automazione.

Il concetto di LCA [7] deriva dal principio giapponese ottocentesco Karakuri Kaizen

(trad. “bambola meccanica”). Tale principio punta ad evitare gli sprechi e si basa sui seguenti punti chiave:

- Automatizzazione delle catene e dei meccanismi pneumatici, idraulici o elettrici per lo spostamento degli oggetti, senza l'intervento dell'uomo;
- Sfruttamento creativo dell'energia a disposizione (es. gravità), tramite la combinazione di diversi sistemi;
- Limitazione dei costi.



**Figura 1.7:** Low cost automation, Fonte: <https://bit.ly/3F305sF>

I meccanismi LCA sfruttano dunque l'energia cinetica rilasciata per azionare l'intera struttura: consideriamo, ad esempio, una spinta impressa da un carrello oppure la forza di gravità che spinge un contenitore industriale a percorrere una corsia. I sistemi descritti comportano sicuramente numerosi vantaggi, miglioramento della qualità del lavoro e sicurezza in primis, con un risparmio di energie e risorse e possono essere utilizzati per ciò che riguarda l'ergonomia delle postazioni di lavoro. Il costo di un LCA varia a seconda della dimensione, del materiale utilizzato, della progettazione e della realizzazione: le soluzioni più semplici possono costare da poche centinaia di euro a 4-5k€, sistemi elettrici con i motori che permettono di risolvere problemi più complessi possono costare più di 30k€. Di norma, le strutture LCA sono progettate e costruite interamente sulla base di disegni tecnici: l'ambiente IVR realizzato simula un'attività in cui l'utente, attingendo ad una libreria predefinita di oggetti e meccanismi, possa parametrizzare ed assemblare una struttura LCA ed inoltre, sfruttando il motore fisico di Unity3D, riprodurre il comportamento dei carichi che si vogliono movimentare.

## 1.4 CIM 4.0 e Stellantis

Il Competence Industry Manufacturing 4.0 (CIM 4.0) [8] è uno degli otto centri di competenza italiani finanziati dal Ministero dello Sviluppo Economico, con sede a Torino, in corso Settembrini 178. Si tratta di un polo di eccellenza che ha l'obiettivo di realizzare attività di orientamento, accompagnare e sostenere le imprese verso quella che è definita la quarta rivoluzione industriale. CIM 4.0 intende essere un supporto strategico per le piccole e medie imprese (PMI) in diverse fasi: dalla progettazione alla produzione, dalla ricerca e sviluppo alla gestione della catena di distribuzione, dalla sicurezza alla digitalizzazione delle transazioni. Tra i partner fondatori del centro di competenza ritroviamo i due maggiori atenei del territorio piemontese (Politecnico di Torino e Università di Torino) e altre 23 grandi aziende che mettono a disposizione le strumentazioni e le strutture necessarie per le sperimentazioni e lo sviluppo di nuove tecnologie.

Tra queste troviamo Stellantis [9], società di controllo multinazionale produttrice di autoveicoli, nata dalla fusione tra i gruppi Peugeot S.A. e Fiat Chrysler Automobiles e che controlla quattordici marchi automobilistici tra cui Alfa Romeo, Citroën, FIAT, Maserati e Peugeot.

Per questo studio, CIM 4.0, Stellantis e Politecnico di Torino hanno messo a disposizione risorse umane interne alle aziende, CIM 4.0 ha messo a disposizione gli spazi di ricerca ed ha fornito la strumentazione necessaria.

## 1.5 Organizzazione dei capitoli

La tesi è articolata in sette capitoli. In questo primo capitolo sono stati descritti gli obiettivi dello studio effettuato, introducendo i concetti di realtà virtuale e Industria 4.0. Nel secondo capitolo è analizzata la tecnologia della realtà virtuale, la sua storia, i suoi sviluppi e le sue architetture. Il terzo capitolo introduce il legame tra l'I4.0 e la realtà virtuale, concentrandosi sull'analisi dello stato dell'arte. Il quarto ed il quinto capitolo si concentrano rispettivamente sulle scelte tecnologiche (hardware e software) e di design effettuate per la realizzazione del progetto. Nel sesto capitolo sono esposti ed analizzati dettagliatamente i risultati dei test effettuati sugli utenti al fine di analizzare la User Experience (UX) e l'ergonomia dell'applicativo. Infine il settimo capitolo consiste in un approfondimento sugli sviluppi futuri legati al framework realizzato.

## Capitolo 2

# VR: Concetti generali

*“Che vuol dire reale? Dammi una definizione di reale. Se ti riferisci a quello che percepiamo, a quello che possiamo odorare, toccare e vedere, quel reale sono semplici segnali elettrici interpretati dal cervello. Questo è il mondo che tu conosci. Il mondo com’era alla fine del ventesimo secolo. E che ora esiste solo in quanto parte di una neuro-simulazione interattiva che noi chiamiamo Matrix.”*

La metafora di Matrix, capolavoro cinematografico del 1999, scritto e diretto dai fratelli Wachowski e vincitore di 4 premi Oscar, aiuta a sciogliere ogni nodo cruciale legato ai concetti di realtà e virtualità. In questo capitolo verrà posto l’accento sulla realtà virtuale, sulla sua storia e sui suoi sviluppi.

### 2.1 Le origini

La storia della VR ebbe inizio circa un secolo fa, sebbene questa tecnologia stia entrando nelle nostre case solo negli ultimi anni. La sua evoluzione è correlata all’informatica, all’incremento delle prestazioni tecnologiche dei calcolatori, allo sviluppo delle interfacce di interazione uomo-macchina e alla computer grafica.

L’idea primordiale di VR compare in un’opera del 1935 intitolata “The Pygmalion’s Spectacles” [10] di Stanley G. Weinbaum, in cui l’autore menziona una particolare invenzione, degli occhiali speciali che permettono di guardare “un film che offre vista e suono [...] gusto, olfatto e tatto. [...] Sei nella storia, parli alle ombre e loro rispondono e invece di essere su uno schermo, la storia è tutta su di te, e tu sei nella storia”.

L’industria cinematografica è stata la prima in assoluto a sperimentare questa nuova tecnologia. Nel 1955 il regista e cameraman americano Morton Heilig, nel



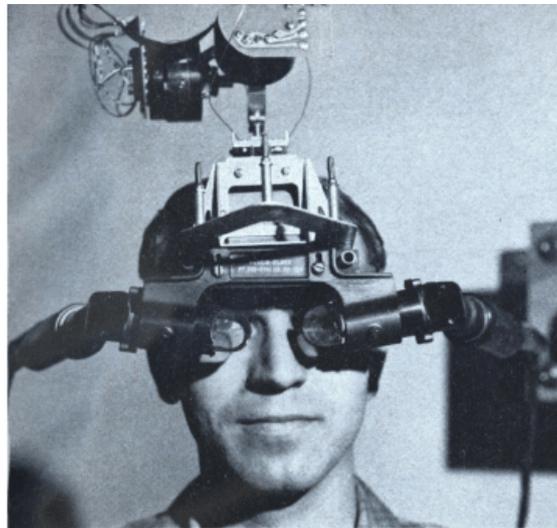
Figura 2.1: Sensorama, 1962, Fonte: <https://bit.ly/3qWR3u0>

saggio intitolato “The Cinema of the Future” [11] , teorizzò il cinema dell’esperienza, in grado di coinvolgere tutti i sensi in modo realistico. L’anno successivo egli realizzò Sensorama, un apparecchio di grandi dimensioni brevettato nel 1962, che consisteva in una cabina con schermi stereoscopici, altoparlanti stereofonici ed una sedia semi movibile, dotato di odori, vento e vibrazioni. Essendo però uno strumento rivolto al cinema, esso non permetteva consentiva un particolare grado di interattività. Il progetto fallì per la mancanza di finanziamenti e non ottenne successo presso l’industria cinematografica, tanto che Heilig realizzò solo cinque brevi film dimostrativi destinati al Sensorama (Helicopter, Motorcycle, Belly Dancer, Dune Buggy, I’m a Coca-Cola Bottle).

Nel 1960 Heilig brevettò un apparato stereoscopico ad uso individuale che includeva meccanismi in grado di stimolare l’utente ed ottenere sensazioni olfattive e uditive, oltre che visive. Tale dispositivo presentava molte similitudini con i successivi Head-Mounted Display (HMD). L’anno successivo Charles Comeau e James Bryan, ingegneri della PhilCo, realizzarono il primo HMD che pilotava una videocamera

in remoto, seguendo i movimenti della testa.

Nel 1963 lo studente Ivan Sutherland realizzò Sketchpad [12] , il primo programma in grado di realizzare della computer grafica interattiva, utilizzando una tastiera ed una penna ottica in grado di manipolare oggetti e disegnare in un ambiente virtuale. Durante la presentazione all’International Federation for Information Processing (IFIP) Congress del 1985, Sutherland introdusse il concetto di “Ultimate Display” [13] , “uno specchio per il paese delle meraviglie matematico”. Si tratta di un dispositivo tramite il quale l’utente poteva interagire con oggetti appartenenti ad un mondo ideale in cui non erano rispettate le leggi di quello fisico, includendo feedback di tipo visivo e tattile.



**Figura 2.2:** The Sword of Damocles, 1968, Fonte: <https://bit.ly/3HM2hIt>

Nel 1968 Sutherland, insieme allo studente Bob Sproull creò il primo HMD per la MR chiamato “The Sword Of Damocles” [14] . Il nome fu ispirato dal fatto che il visore era appeso al soffitto del laboratorio tramite un braccio meccanico per agevolare il movimento dell’utente e non far gravare il peso sulla testa. Questo visore si collegava a un computer che si occupava di eseguire tutti i calcoli necessari a generare le immagini. Esso è considerato anche un precursore della realtà aumentata in quanto dotato di lenti trasparenti non occlusive che lasciavano intravedere l’ambiente circostante.

Nel 1977 al MIT (Massachusetts Institute of Technology) [15] venne creato l’Aspen Movie Map, uno dei primi sistemi VR che consisteva in un tour nella città di Aspen,

in Colorado, in tre versioni differenti: estiva, invernale, digitale. Le prime due si basavano su una serie di filmati montati in modo tale da coprire ogni possibile percorso all'interno della città. La terza versione era una ricostruzione poligonale in cui l'utente, interagendo con delle icone, era in grado di muoversi lungo le vie della città assistito da una cartina virtuale. Cliccando inoltre sugli edifici, egli poteva ottenere delle informazioni ad essi legate: tale sistema è, a tutti gli effetti, il precursore del moderno Google Street View.

Il termine realtà virtuale divenne popolare solo a partire dagli anni 80 quando Jaron Lanier, fondatore della compagnia Visual Programming Lab Research Inc. [16], avviò un vero e proprio ramo di ricerca ad essa dedicato ed un numero sempre maggiore di accessori iniziarono ad essere prodotti e venduti: nel 1984 fu la prima società in assoluto a vendere occhiali e guanti per la realtà virtuale.

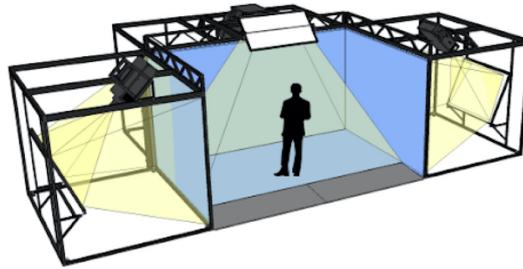
Nel 1989 la Mattel produsse Power Glove, un guanto che fungeva da controller per la console Nintendo Entertainment System.

Nel 1987 Scott Foster fu contattato dalla NASA per creare un dispositivo che simulasse l'emanazione del suono generato in uno specifico spazio 3D, utilizzando un algoritmo sviluppato presso l'Università del Wisconsin.

Nel 1990 la W-Industries produsse Virtuality, una serie di sistemi che comprendevano HMD con schermi LCD, speaker, microfoni, joystick e volanti, destinati alle sale giochi. Questi dispositivi permettevano di effettuare un riconoscimento dei movimenti della testa e della posizione dei controller attraverso sensori e campi magnetici, riproducendoli nell'ambiente virtuale.

Nel 1992 l'Electronic Visualization Laboratory [17] realizzò il primo CAVE (Cave Automatic Virtual Environment), un teatro di realtà virtuale delle dimensioni di una stanza: il sistema consiste in quattro video proiettati dall'esterno sulle tre pareti e sul pavimento, mediante meccanismi composti da proiettori ad alte prestazioni e specchi. La visione stereoscopica era realizzata tramite degli speciali occhiali dotati di sensori di movimento. L'utente era in grado di muoversi liberamente all'interno della stanza e la prospettiva corretta di visione veniva applicata in real time garantendogli una totale immersione.

Nel 1993 la casa di produzione di videogiochi Sega Corporation annunciò l'uscita di Sega VR, un headset VR per la propria console, esteticamente molto simile ai moderni visori. Questo dispositivo non superò mai la fase di prototipazione a causa dei forti problemi legati al motion sickness e delle potenziali conseguenze negative



**Figura 2.3:** CAVE, Fonte: <https://bit.ly/32SNb1P>

sulla vista degli utilizzatori.

Due anni dopo Nintendo rilasciò prima console stereoscopica destinata al grande pubblico, il Virtual Box, che non raggiunse le vendite auspicate e si rivelò un insuccesso a causa della scelta di display monocromatici in rosso e nero e della scarsa ergonomia del dispositivo.

Bisognerà attendere circa due decenni affinché questa tecnologia si affermasse e attirasse l'attenzione del mercato: il 2014 segna un anno di rinascita per la realtà virtuale, a seguito di un periodo di stagnazione, durante il quale grandi aziende dell'intrattenimento hanno fatto grandi annunci in questo settore. La prima in assoluto fu l'azienda di Mark Zuckerberg, Facebook, che acquistò Oculus Rift [18] per la cifra di 2 miliardi di dollari: il visore verrà immesso sul mercato nel marzo del 2016. Contemporaneamente Google annuncia il Google Cardboard [19], un semplice accessorio dal costo ridottissimo in grado di trasformare uno smartphone in un visore di fascia bassa. Infine, sempre nel 2014, Sony [20] annuncia di essere al lavoro su un visore per la realtà virtuale per la PlayStation 4 [21] chiamato Project Morpheus, poi rinominato PlayStation VR [22] e disponibile dal 2016. Tutti questi dispositivi hanno enormi potenzialità sia per l'industria del gaming che per l'entertainment.



**Figura 2.4:** Oculus Rift, HTC Vive, PlayStation VR, Fonte: <https://bit.ly/3nakiIq>

Esistono due tipologie di visori :

- visori che necessitano di un'unità di elaborazione separata, collegata al dispositivo;
- visori svincolati da un'unità di elaborazione esterne (es. dispositivi mobile con supporto).

Oculus Rift (Oculus VR), HTC Vive [23] (Valve [24] e HTC [25]) e Playstation VR (Sony) hanno delle unità di elaborazione esterne a loro connesse per ovvi motivi di potenza computazionale necessaria per l'elaborazione delle immagini in real time ad alto dettaglio. Questi visori implementano anche un sistema di tracciamento dell'ambiente mediante delle telecamere posizionate attorno all'utente per acquisire al meglio i movimenti del corpo e della testa.

Nel 2018 Oculus ha annunciato il suo visore di seconda generazione che promette di superare molte delle attuali limitazioni. HTC invece ha già lanciato sul mercato il nuovo HTC Vive Pro, un dispositivo non cablato con una maggiore risoluzione dei display che promette una room-scale experience sino a 100 metri quadri.

### 2.1.1 Hype Cycle di Gartner

Le nuove tecnologie tendono sempre a seguire uno ciclo abbastanza ricorrente in termini di aspettative e di maturità, l'Hype Cycle di Gartner [26]. Questo schema permette di capire a che punto si trova tale tecnologia. Lo sviluppo della tecnologia emergente raramente procede in maniera lineare. Molto spesso essa acquisisce un andamento simile ad un'ondata e la VR rappresenta la quarta ondata tecnologica destinata a cambiare le vite degli utilizzatori. Le prime tre sono universalmente riconosciute come il personal computer, Internet ed i dispositivi mobili e ciascuna ondata è stata più veloce della precedente. Il ciclo inizia con un evento che attira fortemente l'attenzione di media e investitori (Innovation Trigger) che, nel caso della realtà virtuale, corrisponde alla presentazione di un prototipo dell'Oculus Rift all'E3 del 2012. Segue così il picco di massime aspettative (Peak Of Inflated Expectations) durante il quale molte compagnie decidono di investire e entrare nel mercato: tale momento corrisponde all'acquisizione di Oculus Rift da parte di Zuckemberg. Quello che segue è il momento più complicato per una tecnologia (Trough Of Disillusionment): lo sviluppo non riesce a stare al passo con le aspettative generate ed il numero di investitori diminuisce: tra il 2014 e il 2016 la notorietà e l'attenzione sulla VR sono nettamente calate. Le tecnologie che riescono a superare la fase precedente entrano nello Slope of Enlightenment, che solitamente corrisponde con il rilascio di una seconda o terza generazione della tecnologia, più adatta al mercato e con costi di produzione in calo. L'ultima fase descritta nel

diagramma di Gartner (Plateau of Productivity) coincide con il momento in cui la tecnologia viene adottata in massa e gli investitori vedono arrivare i profitti attesi nel picco iniziale di aspettative. Nel 2021 gli ambienti immersivi sono possono definirsi definitivamente alla portata di chiunque.

### 2.1.2 Il futuro della VR

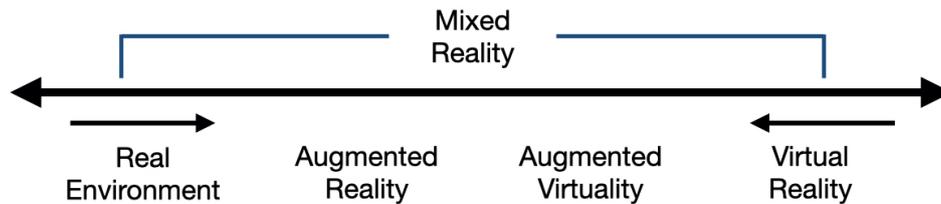
Un tema strettamente correlato alla VR che sta assumendo un'importanza sempre maggiore nel mondo contemporaneo della tecnologia è quello del Metaverso. Esso deriva da “Snow Crash”, romanzo del 1992 scritto da Neal Stephenson ed indica un mondo virtuale tridimensionale abitato da avatar di persone reali. Il 28 ottobre 2021 il CEO di Facebook Mark Zuckerberg ha annunciato la sua decisione di rinominare la società con un nuovo nome, Meta, riproponendo l'idea di Metaverso descritta quasi trent'anni prima da Stephenson. Alcune interessanti definizioni di Metaverso sono attribuite a Herrman e Browning [27] (“un mondo digitale pienamente realizzato che esiste oltre quello analogico in cui viviamo”) e a Caulfield [28] (“un mondo 3D virtuale condiviso, o mondi, che sono interattivi, immersivi e collaborativi”) nel 2021. Il Metaverso si pone come un successore più elaborato di Internet, un ambiente digitale in continua evoluzione che offre più possibilità con un senso di presenza individuale (Ball, 2021), un mondo in cui le esperienze virtuali possono avere conseguenze dirette nel mondo reale. Le precedenti definizioni evidenziano gli attributi comuni del Metaverso:

- persistenza dell'identità e degli oggetti all'interno di un ambiente condiviso,
- utilizzo di avatar,
- sincronizzazione,
- tridimensionalità,
- interoperabilità tra gli utenti,
- interattività, immersività, presenza (già tre componenti fondamentali della VR),
- social network.

Si tratta in di un tema estremamente ampio e in continua evoluzione, che può variare con la costruzione e con l'utilizzo di questa nuova realtà.

## 2.2 Reality-Virtuality Continuum

Nel 1994 Paul Milgram e Fumio Kishino introducono per la prima volta i concetti di realtà mista e Reality-Virtuality Continuum in uno studio intitolato “A Taxonomy of Mixed Reality Visual Display” [29]: esso costituisce un’ottima base teorica per comprendere il passaggio tra mondo reale ed ambiente virtuali.



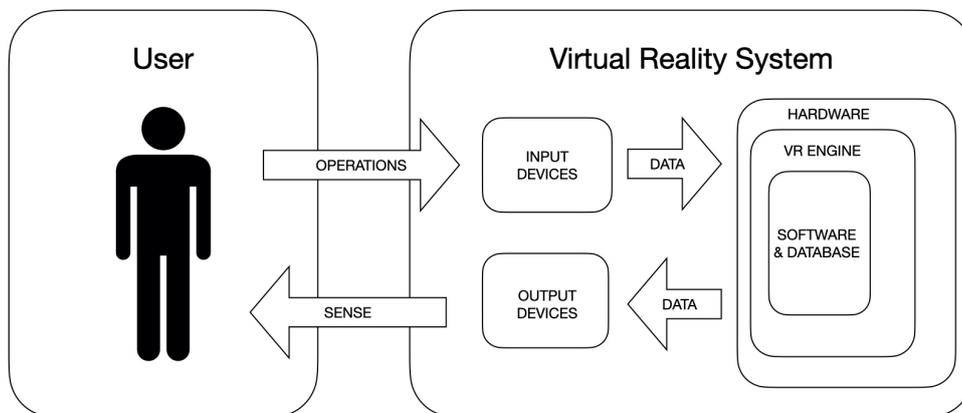
**Figura 2.5:** VR Continuum, 1994

Il Continuum include tutte le possibili varianti e combinazioni di oggetti reali e virtuali ed è composto da quattro diversi stadi. Ai due estremi ritroviamo la realtà fisica e la VR; la Mixed Reality MR comprende la realtà aumentata (Augmented Reality, AR) e la virtualità aumentata (Augmented Virtuality, AV). Secondo l’istituto di ricerca Juniper [1], per realtà aumentata (Augmented Reality, AR) s’intende “una tecnologia che identifica e traccia i dati appartenenti al mondo fisico e vi combina dati originati da una fonte digitale, in grado di presentare all’utente una visione del mondo reale sovrapposto da informazioni generate dal computer”. Gli utenti possono interagire con il sistema AR attraverso strumenti quali smartphone, tablet, dispositivi indossabili come i Google Glasses, schermi fissi o proiettori. Utilizzando diversi tipi di sensori, il sistema raccoglie in input i dati relativi all’ambiente circostante che vengono immediatamente processati al fine di creare una proiezione 2D o 3D. Alcune applicazioni AR, ad esempio, permettono di aggiungere informazioni sintetiche attraverso un’immagine tag scannerizzata con un dispositivo che può rilasciare l’immagine aumentata oppure di aumentare lo spazio che ci circonda tramite elementi virtuali. Questo tipo di applicazioni sono utilizzate per aggiungere contenuti ed informazioni rispetto al luogo dove ci si trova, ad esempio nei musei o nei giochi. La differenza tra AR e AV consiste nel fatto che mentre la prima arricchisce il mondo reale con oggetti virtuali, la seconda aggiunge elementi del mondo reale a quello virtuale. Alcuni sistemi AV, ad esempio, permettono la visualizzazione delle mani dell’utente mentre interagiscono con oggetti virtuali. Per realtà mista (Mixed Reality, MR), invece, si intende quel tipo di realtà ibrida in cui i contenuti digitali coesistono con l’ambiente fisico ed

interagiscono tra di essi, combinando VR e AR per creare un nuovo ambiente. A differenza della AR, la MR non si limita ad anteporre delle immagini digitali su uno schermo o su una lente, ma è in grado di riconoscere e comunicare con gli spazi circostanti: si tratta di un'ulteriore evoluzione dell'interazione uomo-computer-ambiente. Il dispositivo che permette al meglio di comprendere questa tecnologia è Microsoft HoloLens [30]. Questo visore possiede lenti trasparenti che permettono di osservare l'ambiente circostante durante l'utilizzo e dispone di sensori avanzati e audio spaziale che permette all'utente di capire la direzione di provenienza della luce e del suono. Inoltre non presenta nessun cavo, quindi chi lo utilizza ha piena libertà di movimento. Tra i vari stadi, la VR fornisce l'esperienza più immersiva per l'utente.

## 2.3 Architetture input

Le modalità con cui gli utenti interagiscono con un sistema VR influenzano fortemente la loro esperienza nel mondo virtuale ed incidono sulle componenti di immersione, illusione e presenza. In particolare, esse condizionano la semplicità d'uso del sistema e le possibili azioni che gli utenti possono compiere durante la simulazione.



**Figura 2.6:** Interazione tra utente e sistema VR

Il componente chiave di questa tecnologia è il tracker (VR positional tracking system), sistema che permette di monitorare la posizione dell'utente e degli oggetti fisici (supporti) a loro accessibili. Esiste una vasta gamma di dispositivi che permettono di effettuare il tracking. Ognuna di questi possiede vantaggi e svantaggi ma

è possibile combinare varie opzioni per migliorare le prestazioni del tracciamento. Questi sistemi si differenziano in base al modo in cui l'utente immette i propri comandi per il mondo virtuale (sistemi di input).

È possibile distinguere due tipologie di input:

- **Input attivo:** gli eventi sono azionati specificatamente dall'utente;
- **Input passivo:** gli eventi sono azionati dal sistema che monitora il partecipante.

Gli aspetti che permettono di classificare i sistemi percettivi alla posizione (position-sensing) e valutarne l'efficienza sono:

- la velocità di aggiornamento della posizione da parte del sensore e la sua precisione. Il rumore, la scarsa precisione e la latenza correlati al dispositivo diminuiscono il grado di immersività dell'esperienza;
- il materiale di costruzione (es. metallo);
- l'ingombro (es. dispositivo cablato, collegamenti meccanici).

Per quanto costoso, nessun dispositivo fornisce delle condizioni ottimali per tutti e tre i parametri analizzati. Di seguito sono descritte le tecnologie di tracciamento più diffuse, implementate nei sistemi VR odierni.

### **Tracker meccanici**

I tracker meccanici sono strutture cinematiche i cui giunti contengono dei sensori per la misurazione degli angoli solidi tra i segmenti. Data la posizione del riferimento e le caratteristiche degli appartenenti alla catena cinematica, è possibile calcolare la posizione e dell'oggetto da tracciare (end effector). Questi sistemi sono caratterizzati da latenza estremamente bassa, accuratezza costante e sono immuni da interferenze esterne. Presentano comunque svantaggi in quanto essi limitano la libertà di movimento dell'utente e possono essere pesanti, se indossati.

### **Tracker magnetici**

I tracker magnetici sono i primi sistemi di tracciamento non a contatto realizzati. Essi sfruttano un campo magnetico prodotto da tre bobine poste in maniera ortogonale tra loro all'interno di una sorgente fissa (trasmitter) per determinare la posizione e l'orientamento relativi ad una sorgente in movimento (receiver). A differenza dei tracker meccanici, i tracker magnetici consentono all'utente di muoversi liberamente anche in ambienti che presentano ostacoli visivi tra lui e il trasmettitore: in questo caso, il sistema wireless che riduce notevolmente

l'ingombro. Questi sistemi di tracciamento possono essere intralciati da possibili interferenze causate dai metalli presenti nell'ambiente circostante o dalla ridotta portata del campo magnetico generato. Infatti qualora l'utente dovesse allontanarsi dal trasmettitore, la precisione potrebbe calare drasticamente.

### Tracker acustici

I tracker acustici utilizzano ultrasuoni a bassa frequenza per determinare la posizione nello spazio. Il suono è prodotto da una sorgente triangolare in posizione fissa composta da tre altoparlanti che sono attivati in sequenza. Il ricevitore è composto da tre microfoni. Ogni suono emesso dalla sorgente determina tre misurazioni al ricevitore e, conoscendo la posizione dell'emettitore, si determina la posizione spaziale del microfono. Questi sistemi risultano relativamente economici. Nonostante ciò, i tracker acustici possiedono diversi svantaggi. È possibile che altre sorgenti di rumore ultrasonico (es. hard disk) interferiscano con esso, richiedono una visuale diretta tra trasmettitore e ricevitore e dipendono dalla velocità del suono, calcolata in funzione di fattori esterni quali la temperatura, pressione e umidità dell'aria.

### Tracker ottici

I tracker ottici sono sistemi contactless che utilizzano la luce (fotoni) per manipolare, memorizzare e trasmettere dati. È possibile distinguerli in due categorie, i tracker ottici "Outside looking in" e "Inside looking out". I primi consistono in sensori collocati in postazioni fisse ed emettitori di segnali luminosi in movimento nell'ambiente. Per semplificare la gestione del tracciamento, si utilizzano i marker, ovvero dispositivi in grado di riflettere luce (marker passivi) o di emettere luce (marker attivi). I tracker "Inside looking out" consistono invece in sensori (es. telecamere) posizionati direttamente sull'oggetto in movimento.

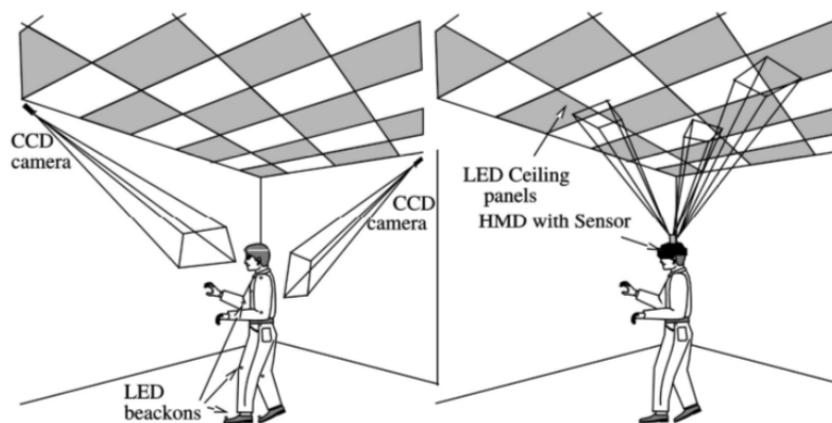


Figura 2.7: Outside looking in e Inside looking out

### **Tracker inerziali**

I tracker inerziali sono sistemi microelettromeccanici (Micro Electro-Mechanical Systems) che calcolano velocità angolari attraverso tre giroscopi ortogonali e accelerazioni spaziali grazie a tre accelerometri, solidali con i giroscopi. A differenza dei tracker magnetici e acustici, questi sistemi non subiscono nessuna interferenza da segnali provenienti dall'esterno ed hanno valori di refresh rate elevatissimi. La velocità angolare e l'accelerazione sono ottenute per integrazione (o doppia integrazione), quindi, per le proprietà di additività e linearità degli integrali, il rumore sulle misure non si elimina ma si somma. Ciò comporta uno svantaggio poiché l'errore cresce geometricamente nel tempo.

### **Motion capture**

I dispositivi di tracking possono essere utilizzati anche per catturare il movimento dettagliato del corpo umano. In particolare i dati catturati possono essere utilizzati per l'animazione di un modello umanoide digitale. Esistono sistemi che permettono di tracciare il corpo intero (es. XSense), il volto o parti singole del corpo (mani, gambe etc.) e risultano essere interfacce estremamente naturali. Molte tecnologie motion capture utilizzano strutture meccaniche (es. esoscheletri) per il tracking. Queste tecnologie possono essere adoperate anche in ambienti aperti e la area virtuale può raggiungere dimensioni molto grandi. Risultano essere strutture altamente intrusive che limitano i movimenti dell'utente che li indossa.

### **Interfacce non convenzionali**

Esistono molte tecniche sperimentali di tracking che possono essere applicate potenzialmente in ambienti tradizionali (es. medicina). È possibile, ad esempio, utilizzare comandi vocali come input (riconoscimento del parlato). I sistemi attuali funzionano bene in ambienti controllati ma possono presentare problemi in ambienti virtuali a causa delle interferenze da parte di altri utenti partecipanti alla simulazione oppure dal rumore di fondo. Un'interessante tecnologia è Brain Computer Interface (BCI) che si basa sull'analisi dell'attività cerebrale effettuata grazie a tecniche non intrusive quali risonanza magnetica ed elettroencefalogramma.

### **Tracker ibridi**

I tracker ibridi consistono in sistemi combinano due o più tecnologie di misura della posizione per effettuare un tracciamento dell'oggetto migliore di quello garantito da una singola tecnologia. Un esempio di tracker ibrido è WiiMote, interfaccia di Nintendo che utilizza due tipi di sensori, ottico e inerziale. Un sistema di infrarossi rileva la posizione del controller mentre una serie di accelerometri rileva l'orientamento del dispositivo.

## 2.4 Architetture output

Il secondo aspetto chiave delle esperienze VR riguarda come la percezione fisica dell'ambiente virtuale da parte dell'utente, in modo la simulazione virtuale fornisce feedback specifici al sistema percettivo umano. In particolare esistono architetture output specifiche per la vista (display grafici), tatto (force e touch feedback), udito (3-D audio hardware) ed inerziali.

### Display grafici, Rendering grafico

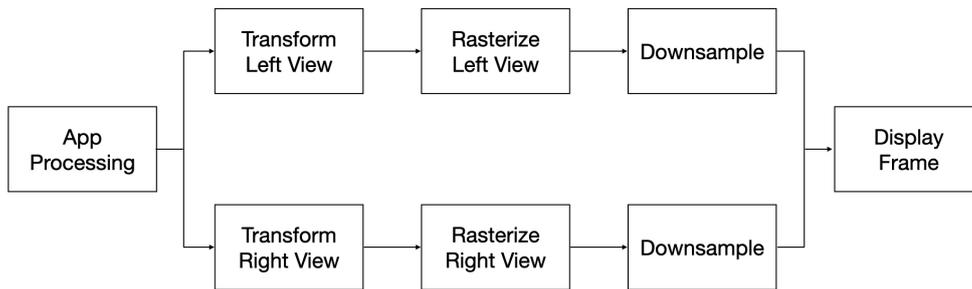
I display grafici sono interfacce che visualizzano direttamente il mondo virtuale. È possibile distinguere questi dispositivi a seconda del tipo di visualizzazione (monoscopica o stereoscopica), della risoluzione dell'immagine, del FOV, della tecnologia del display (LCD/LED/OLED), dei fattori ergonomici e dei costi. Tutti i sistemi VR possiedono necessariamente almeno un dispositivo di visualizzazione.

Esistono tre grandi famiglie di dispositivi di visualizzazione:

- **Personal Graphic Display:** utilizzabili da un singolo utente (es. HMD),
- **Desk Supported Display:** utilizzabili da più utenti (es. monitor),
- **Large Volume Display:** schermi di grandi dimensioni, utilizzabili da un gran numero di utenti.

Un Head-Mounted Display (HMD) è un caschetto indomabile munito di due display che permettono la migliore separazione possibile delle immagini, ai fini della visione stereoscopica, provvisti di cuffie e tracker. Gli HMD proiettano l'immagine come se fosse posta a 1-5 metri davanti all'occhio dell'utente, riducono la fatica di focalizzazione ed aumentano il FOV e la granularità dell'immagine (gradi per pixel). Gli HMD si distinguono in untethered e tethered: i primi dispositivi non richiedono collegamenti con computer o sensori esterni per il tracking, forniscono all'utente maggior libertà di movimento ma prestazioni minori in termini di elaborazione, grafica ed accuratezza.

Il modello concettuale che descrive il numero di passaggi che un sistema grafico deve eseguire per generare un'immagine digitale di una scena 3D su uno schermo 2D prende il nome di pipeline di rendering (grafico). I passaggi per compiere questa operazione dipendono dall'hardware, dal software e dalle caratteristiche di visualizzazione desiderate. Non esiste una pipeline grafica standard per tutti i casi. Esistono tuttavia interfacce di programmazione delle applicazioni grafiche (API, es. OpenGL) che permettono di unificare passaggi simili per controllare la pipeline di rendering di un acceleratore hardware, impedendo al programmatore di manipolare



**Figura 2.8:** Pipeline di rendering grafico

tali acceleratori (es. Intel, NVIDIA). Le simulazioni VR sfruttano il modello della pipeline di rendering grafico in tempo reale, suddiviso in tre fasi principali:

- **Fase dell'applicazione:** fase eseguita dal software sulla CPU. In questa fase vengono apportate le modifiche alla scena a seconda delle esigenze dell'utente. La nuova scena, composta solitamente da punti, linee e triangoli, viene passata alla fase successiva della pipeline. In un moderno motore di gioco come Unity, il programmatore si occupa quasi esclusivamente della fase applicativa ed utilizza un linguaggio di alto livello come C#;
- **Fase della geometria:** fase responsabile della maggior parte delle operazioni con i poligoni e i loro vertici. È possibile suddividere ulteriormente questa fase, in base alle caratteristiche di scena manipolate. In base all'implementazione di questa fase dipende l'organizzazione delle fasi reali della pipeline parallela;
- **Rasterizzazione:** fase conclusiva che applica il colore e trasforma gli elementi grafici in pixel.

### Display aptici, Rendering tattile

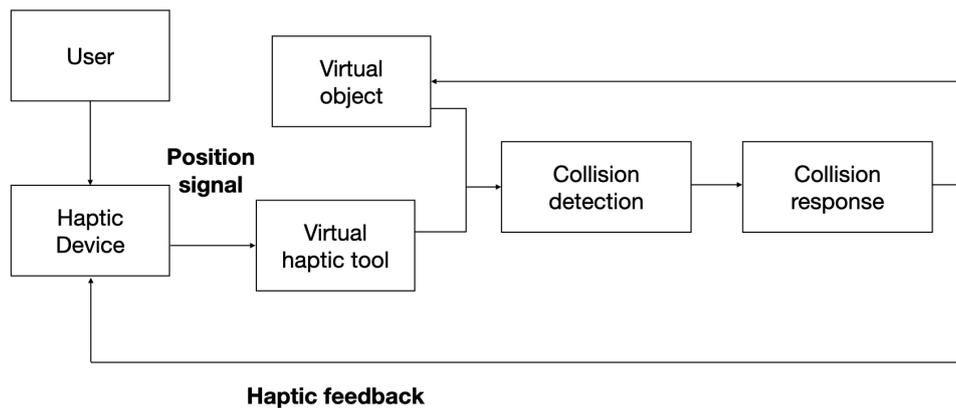
Le interfacce aptiche trasmettono all'utente informazioni relative al tatto. In particolare esse vanno ad identificare le informazioni tattili degli oggetti virtuali. I display aptici forniscono un feedback praticamente solo sulla mano ma contengono la maggior parte dei sensori tattili del corpo umano.

È possibile distinguere due tipi di display:

- **Touch feedback:** forniscono informazioni in tempo reale su temperatura, scivolosità, rugosità e geometria dell'oggetto 3D e non oppongono una resistenza attiva al movimento dell'utente che ha determinato il contatto (es. Handheld VR Controller);

- **Force feedback:** forniscono informazioni in tempo reale su resistenza, inerzia e peso della superficie di contatto. Utilizzano dei sensori sui tendini dei muscoli e sulle giunzioni per opporre una resistenza attiva al movimento dell'utente che ha determinato il contatto (es. CyberTouch Glove).

Non è possibile definire una struttura standard di pipeline di rendering tattile in quanto essa è legata ai singoli dispositivi. Essa è gestita quasi totalmente dal software e richiede sforzi computazionali complessi.



**Figura 2.9:** Pipeline di rendering tattile

L'identificazione delle collisioni (collision detection) può essere approssimata oppure esatta. Nel primo caso l'oggetto 3D è rappresentato da un bounding box che si avvicina quanto più alla sua geometria. I bounding box possono essere:

- **fissi:** maggiore velocità di calcolo computazionale;
- **variabili:** orientati o allineati lungo gli assi del sistema di riferimento.

L'identificazione delle collisioni avviene ad ogni frame. Nel caso di oggetti in movimento, una collisione può non essere identificata se:

$$d < \frac{v}{fps}$$

in cui  $d$  equivale alla distanza massima tra due bounding box sulla traiettoria e  $v$  è la velocità relativa degli oggetti. Infine la risposta alle collisioni (collision response) consiste nel calcolo del force feedback e del touch feedback.

### Display acustici e inerziali

I display acustici forniscono un feedback sonoro. Il suono può essere di tipo

monoaurale (entrambe le orecchie sentono lo stesso suono) oppure binaurale (ogni orecchio sente un suono diverso). I due concetti chiave legati a queste interfacce sono la localizzazione e la spazializzazione. Il primo riguarda la capacità del cervello di identificare la posizione spaziale di emissione di un suono ed è correlato alla variazione di tempo e di volume della percezione, riferita alle due orecchie. La spazializzazione invece riguarda la generazione di un suono che sembri provenire da una determinata posizione spaziale da parte del display. La localizzazione e la spazializzazione non sono gli unici fattori di rilevanza per la generazione dell'audio 3D ma è necessario anche considerare l'interazione con l'ambiente quali, ad esempio, la riflessione del suono, echi, riverberi ed effetto Doppler.

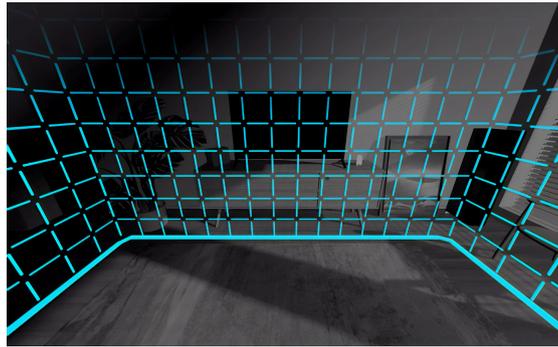
I display inerziali permettono all'utente di percepire le forze di inerzia, in accordo con le azioni e movimenti effettuati nel mondo virtuale (es. simulatori di movimento). Utilizzano dispositivi elettromagnetici e non devono essere confusi con le interfacce aptiche in quanto non coinvolgono il senso del tatto.

## 2.5 Metafore d'interazione

L'interazione è una caratteristica chiave della VR e il senso di presenza dell'utente all'interno del mondo virtuale aumenta quanto più il mondo stesso risponde adeguatamente alle azioni svolte. La vera sfida all'interno delle esperienze VR consiste nel raggiungere la maggior naturalezza nelle interazioni tra utente e interfacce e, nel tempo, gli utenti hanno acquisito familiarità con nuove tecnologie che permettono di relazionarsi con l'ambiente virtuale tramite alle metafore di interazione. Di conseguenza il sistema deve fornire dei feedback adeguati nel momento in cui avviene l'interazione in modo tale da comunicare all'utente il suo effettivo avvenimento. Di seguito sono descritte le quattro forme di interazione in un sistema VR con alcune delle rispettive metafore.

### Navigazione

La navigazione è l'operazione più comune in mondi virtuali ed è composta da due componenti, spostamento (travel) e orientamento (wayfinding). La prima è la componente che si riferisce al movimento fisico da un posto del mondo virtuale ad un altro, la seconda è la componente cognitiva o decisionale della navigazione. È necessario considerare quali sono i vincoli legati al sistema VR utilizzato. Ad esempio la IVR con HMD tracciato (6 DOF) permette di utilizzare la posizione reale dell'utente per la navigazione nel mondo virtuale (real walking), a differenza di un sistema HMD 3 DOF che richiede modalità alternative per indicare lo spostamento.



**Figura 2.10:** Real walking e barriere di sicurezza, Fonte: <https://bit.ly/32SLjtf>

Altri fattori che il progettista VR non dovrebbe trascurare riguardano le condizioni di avvio e di arresto del moto (quando e come inizia il movimento), il controllo della velocità e dell'accelerazione e la modifica della vista. In sistemi parzialmente immersivi, la vista è funzione della direzione di movimento; in sistemi immersivi (HMD e CAVE) ciò non è necessario. Il real walking è una metafora naturale che riduce il motion sickness ma presenta potenziali problemi di sicurezza nel movimento a causa del limitato spazio fisico dell'utente. Per prevenire, i sistemi IVR presentano barriere virtuali di sicurezza per evitare di oltrepassare l'area di simulazione.

Un'altra interessante metafora di navigazione è il teletrasporto (teleportation), ovvero la possibilità di trasportarsi velocemente da punto a punto all'interno della simulazione tramite, ad esempio, un raggio virtuale.

### **Selezione e manipolazione**

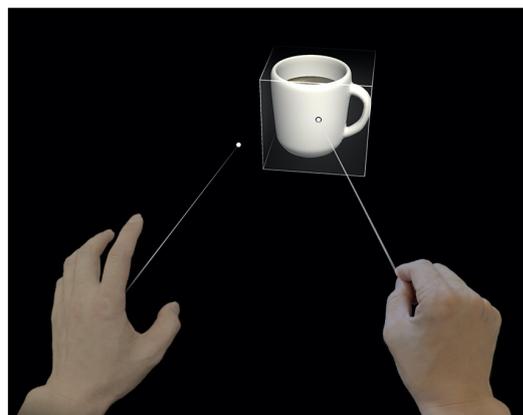
La selezione e la manipolazione sono due azioni che l'utente può compiere nel mondo virtuale strettamente collegate tra loro e condividono le stesse metafore d'interazione. La selezione è l'identificazione di un oggetto o di un insieme di oggetti all'interno della simulazione. La selezione può essere utilizzata da sola, ad esempio per individuare un oggetto da eliminare. La manipolazione consiste nella definizione delle proprietà di quell'oggetto, come ad esempio posizione, orientamento o scala ed è una conseguenza della selezione. È necessario considerare le modalità di integrazione tra queste due operazioni: spesso le tecniche di selezione svolgono anche la funzione di manipolazione. Infine è importante attribuire all'oggetto rilasciato un comportamento coerente con le dinamiche del sistema: ad esempio

l'oggetto potrebbe rimanere fermo nella sua ultima posizione oppure cadere per gravità. In 2D VR gli oggetti sono accessibili per selezione e manipolazione diretta, i sistemi 3D VR aggiungono aspetti più complessi che non possono essere trascurati quali, ad esempio, a valutazione della profondità o i problemi di occlusione.

La prima interessante metafora d'interazione è il virtual hand, la selezione diretta di un agente attraverso una mano virtuale con conseguente aggancio dopo la collisione. Risulta essere la metafora di selezione e manipolazione più naturale tra tutte, nonostante potrebbe presentare problemi legati alla raggiungibilità di oggetti distanti.



**Figura 2.11:** Virtual hand, Fonte: <https://bit.ly/3zB89kZ>



**Figura 2.12:** Raycasting, Fonte: <https://bit.ly/3JQgVA9>

La metafora del raycasting (space wand) risolve tale problema. Essa consiste in un raggio virtuale 3D che parte dall'utente e seleziona l'oggetto colpito. Si tratta dell'operazione più semplice da implementare ed utilizzare. Risulta invece complicato selezionare e manipolare oggetti occlusi. Un'ultima interessante metafora d'interazione permette di selezionare un oggetto pronunciandone il nome o una sua proprietà (speech). Si tratta di un'operazione sicuramente immediata ma potrebbe presentare difficoltà legate al riconoscimento del parlato o alla non univocità dei nomi degli oggetti di scena.

### Controllo del sistema

L'ultima operazione fondamentale consiste nella modifica interattiva dello stato del sistema o delle modalità di interazione, come ad esempio il caricamento di un

livello o la scelta della vista (solidale alla prospettiva dell'utente, terza persona etc.) tramite menu 2D o 3D (Graphics User Interface, GUI). Il controllo del sistema deve essere immediato e non deve distogliere l'attenzione dell'utente dai task della simulazione virtuale. Esso deve restituire feedback efficaci ed avere una rappresentazione visuale adeguata.

È possibile classificare diverse tipologie di UI (User Interface):

- **Interfacce diegetiche:** interfacce spaziali perfettamente integrate nel mondo 3D ma slegate dalla narrativa;
- **Interfacce meta-diegetiche:** interfacce generalmente 2D, che si integrano con la narrazione della simulazione ma non con lo spazio virtuale;
- **Interfacce non diegetiche:** interfacce 2D, sovra impresse al display utilizzato, slegate dalla narrativa.



**Figura 2.13:** In alto a sinistra: interfaccia non diegetica, a destra: interfaccia meta-diegetica, in basso: interfaccia diegetica, Fonte: <https://bit.ly/3toSDaX>



## Capitolo 3

# VR e I4.0: Stato dell'arte

In questo capitolo sarà descritto il solido legame tra le tecnologie di realtà virtuale e l'Industria 4.0.

### 3.1 VR per la formazione e l'assemblaggio

La VR permette di generare un ambiente virtuale per visualizzare fenomeni non propriamente visibili all'occhio umano, ad esempio simulando i processi di assemblaggio.

Nel 1999 l'Università di Birmingham [31] ha condotto uno studio al fine di comprendere in qual misura la VR faciliti le attività di formazione per consentire la manipolazione diretta dei componenti di assemblaggio. I protagonisti di tale esperimento sono studenti di ingegneria ai quali è stato chiesto di montare una pompa dell'acqua, costituita da otto componenti separati. Diversi gruppi di studenti hanno ricevuto le informazioni relative all'assemblaggio attraverso modalità differenti: disegni tecnici (bidimensionali), un primo sistema VR "desktop + mouse 2D", un secondo sistema VR analogo al precedente integrato con occhiali stereoscopici (per fornire tridimensionalità alle immagini) ed infine un terzo sistema VR con HMD e mouse 3D, altamente immersiva. In particolare, le condizioni VR sono state condotte su un workstation Silicon Graphics Indigo2, occhiali stereoscopici Crystal Eyes, HMD VR4 e un sistema di tracciamento Polhemus Fastrak mentre i modelli 3D sono stati generati 3D Studio Max (Autodesk). Ai partecipanti è stata innanzitutto fornita una breve introduzione su come utilizzare il software VR e l'opportunità di sperimentare le varie tecniche di interazione. I tempi di completamento dell'attività sono più lunghi quando i partecipanti si allenano per assemblare la struttura mediante il disegno tecnico, prima di procedere all'operazione nel mondo reale, rispetto alle condizioni di VR le quali invece non presentano sostanziali differenze tra le tempistiche di apprendimento, sebbene i partecipanti hanno inoltre

sottolineato come la IVR fosse più intuitiva rispetto alle altre. Nonostante ciò sono emersi diversi problemi legati all'aspetto tecnologico ed ergonomico del sistema VR: in particolare 8 utenti su 20 hanno definito "lento" il controllo del sistema, 7 utenti su 20 hanno indicato che il sistema VR limitava il partecipante a condurre l'attività utilizzando una sola mano ed infine 4 su 20 hanno indicato difficoltà nella selezione degli oggetti e nel raggiungere i componenti generati dal computer, poiché non è stato fornito alcun feedback tattile. In termini di formazione, la VR consente all'utente di manipolare oggetti senza l'uso di artefatti reali, quindi offre vantaggi notevoli durante le fasi di prototipazione e valutazione, passaggi indispensabili dell'attuale processo di creazione del prodotto.



**Figura 3.1:** Rappresentazione dell'assemblaggio virtuale della pompa, 1999, Fonte: <https://bit.ly/3n5uaDm>

Secondo uno studio del 2010 dell'Iowa State University [32], le sfide tecniche da superate per realizzare simulazioni di assemblaggi virtuali vincenti riguardano i seguenti aspetti: rilevamento delle collisioni e gestione dei vincoli tra componenti di un assieme, simulazione realistica della fisica, trasferimento dei dati tra sistemi CAD e VR, manipolazione degli oggetti (tra cui inclusione di feedback visivi, uditivi e aptici). In particolare, per applicazioni IVR, sono stati sviluppati diversi algoritmi che utilizzano dati poligonali per il rilevamento delle collisioni che impedisce la compenetrazione delle parti. Il rilevamento delle collisioni non fornisce informazioni all'utente su come modificare posizione e orientamento delle parti per consentire le operazioni di assemblaggio.

Le due classificazioni principali delle tecniche per l'implementazione del posizionamento delle parti includono metodi di modellazione basata sui vincoli geometrici (concentricità, parallelismo, superfici complanari etc.) applicati tra le parti, riducendo i gradi di libertà e facilitando i compiti di assemblaggio, e metodi di modellazione basata sulla fisica, che simula il comportamento realistico dei componenti in una

scena virtuale, attraverso interazioni fisiche simulate, calcolate in tempo reale. Gli approcci di modellazione basati sui vincoli geometrici per determinare le relazioni tra i componenti di un assieme traducono i vincoli in un sistema di equazioni algebriche, in cui sono necessari buoni valori iniziali per gestire il numero potenzialmente esponenziale di possibili soluzioni: se combinati con modelli accurati, tali approcci consentono agli utenti di manipolare e posizionare le parti in un assieme con elevata fedeltà. Tuttavia, sperimentalmente [33], si è giunti alla conclusione che i requisiti di calcolo di questi metodi li rendono inappropriati per le applicazioni del mondo reale. Alcune di queste applicazioni richiedevano inoltre toolkit CAD speciali per estrarre i metadati CAD rilevanti, necessari per preparare uno scenario di assemblaggio. Questi requisiti di calcolo speciali e la loro dipendenza da specifici sistemi CAD, hanno reso poco favorevole la diffusione di questi metodi.

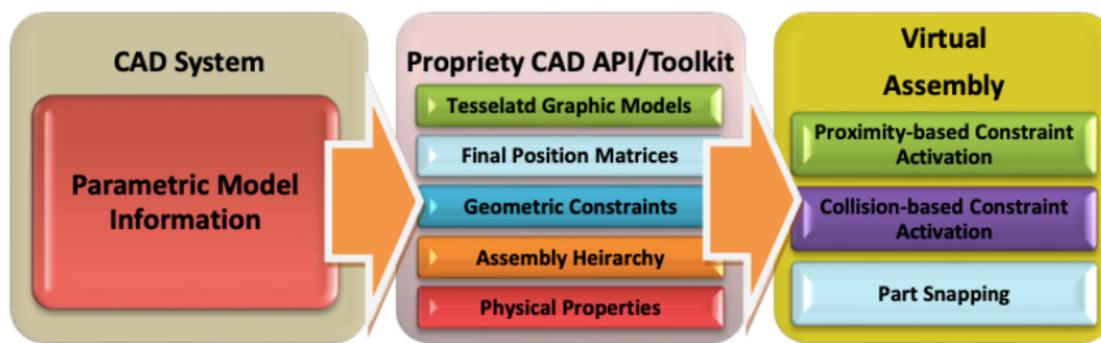


Figura 3.2: Modellazione basata sui vincoli geometrici

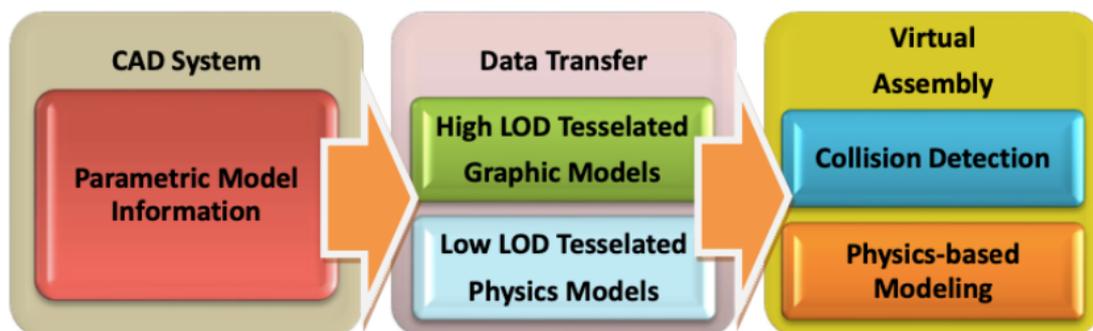
Le tecniche di modellazione basate sulla fisica simulano le forze che agiscono sui corpi per modellare un comportamento realistico, migliorando notevolmente il senso di immersione e interattività dell'utente. Questi algoritmi risolvono le equazioni del moto degli oggetti ad ogni passo temporale, in base alle loro proprietà fisiche, alle forze e ai momenti che agiscono su di essi e possono essere classificati in tre categorie, a seconda del metodo utilizzato. Essi sono:

- **Metodo delle forze di penalizzazione:** metodo che utilizza un sistema di ammortizzatori a molla per impedire la compenetrazione tra i modelli, facile da implementare e poco costoso dal punto di vista computazionale. Tuttavia questi approcci sono caratterizzati da problemi causati da una rigidità elastica molto elevata che porta a equazioni numericamente intrattabili;
- **Metodo dell'impulso:** metodo che simula le interazioni tra oggetti utilizzando impulsi di collisione, in particolare i contatti statici sono modellati come una serie di impulsi di collisione ad alta frequenza che si verificano tra gli oggetti. Questi approcci sono più stabili e robusti dei metodi analizzati

precedentemente, tuttavia, presentano problemi nel gestire contatti simultanei e, in alcuni casi, nella modellazione dell'attrito statico;

- **Metodo analitico:** metodo che controlla le compenetrazioni. In base ai punti di contatto, viene risolto un sistema di equazioni di vincolo per generare forze e impulsi di contatto per ogni punto di contatto. I risultati di questo metodo sono molto accurati, tuttavia richiede molto tempo di calcolo quando si verificano più contatti contemporaneamente.

Dunque, sebbene vari algoritmi per la modellazione basata sulla fisica si siano evoluti nel corso degli anni, la simulazione di un comportamento realistico tra componenti complessi in modo interattivo e accurato è ancora un compito oneroso. Tali metodi, generalmente, sacrificano l'accuratezza del calcolo per mantenere realistica la velocità di aggiornamento della simulazione visiva. Le risposte alle collisioni e alla fisica, infatti, devono essere calcolate abbastanza velocemente da tenere il passo con la frequenza di aggiornamento della grafica della simulazione (circa 30 Hz). Inoltre la maggior parte degli algoritmi utilizzava rappresentazioni di mesh poligonali semplificate e approssimate di modelli CAD per calcoli fisici e di collisione più rapidi o addirittura rappresentazioni ancora più grossolane, utilizzando elementi voxel cubici (controparte tridimensionale del frame buffer bidimensionale dei pixel).



**Figura 3.3:** Modellazione basata sulla fisica

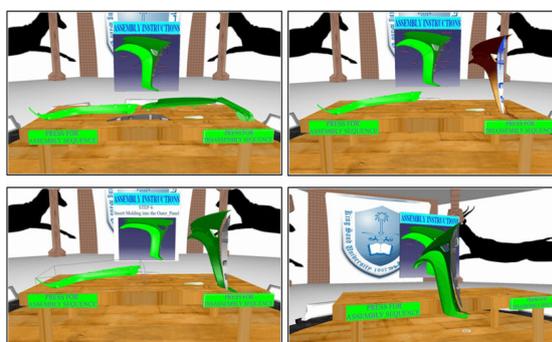
Entrambi questi approcci servono a scopi diversi che sono cruciali per il successo di una simulazione di assemblaggio virtuale. All'interno dell'ambiente virtuale, l'utente deve sperimentare quell'effetto di integrazione col mondo reale tipico di noi esseri umani: maggiore è il numero di canali sensoriali coinvolti, maggiore è il senso di immersione. In sintesi, un'applicazione IVR deve considerare ed integrare coerentemente tra loro i tre concetti di percezione, interazione, presenza.

Un interessante studio che effettua un confronto tra le prestazioni dell'utente,

sottoposto a tre diverse condizioni di formazione (ingegneria tradizionale mediante disegno tecnico, CAD, IVR), è descritto in un articolo del 2013 in “International Journal of Rapid Manufacturing” [34]. Il caso studio riguarda l'assemblaggio del parafrangente della prima auto progettata dall'Arabia Saudita, nota come Gazal-1. I partecipanti sono stati assegnati in modo casuale alle tre diverse condizioni. Tutti i partecipanti hanno completato lo studio fino alla fine. Tutti i partecipanti hanno riportato un'acuità visiva normale o corretta a normale e un udito normale. Tutti i partecipanti, tranne due, erano destrorsi. In particolare, per la formazione IVR, viene utilizzato un sistema di visualizzazione che consiste in un grande schermo, un proiettore ad alta risoluzione, retro-proiettato per produrre un ambiente tridimensionale immersivo, occhiali con otturatore attivo per la visione stereoscopica, un workstation Dell Precision T5400 con processore Intel Xeon da 2,33 GHz e scheda grafica Nvidia FX5800 da 1,5 GB. Inoltre un sistema di tracciamento inerziale (Intersense IS900) fornisce una misurazione dinamica e in tempo reale della posizione e dell'orientamento della testa e delle mani dell'utente. È stato sviluppato un ambiente di assemblaggio virtuale del parafrangente, in cui tutti i vari componenti giacciono su un tavolo, ed è stata fornita una schermata di istruzioni sulla quale venivano visualizzate le istruzioni di assemblaggio ogni qualvolta che il componente veniva selezionato dall'utente. La validità dell'interfaccia dell'ambiente virtuale è stata valutata da esperti in materia del team di progettazione del gruppo automobilistico Gazal-1.



**Figura 3.4:** Gazal-1, 2010, Fonte: <https://bit.ly/3qZDnOS>



**Figura 3.5:** Schermate dell'ambiente di assemblaggio virtuale di Gazal-1, 2010, Fonte: <https://bit.ly/3t6QUXz>

L'utente inoltre è dotato di due interruttori digitali in ambiente virtuale sulla parte anteriore destra e sinistra del tavolo: premendo il pulsante sinistro si avvia una sequenza di montaggio e premendo il pulsante destro si avvia la sequenza di smontaggio. Ciò offre una migliore comprensione delle fasi di assemblaggio coinvolte. Infine, in caso di collisioni di tipo geometrico tra i diversi componenti, il

sistema prontamente fornisce un feedback di tipo visivo. Al termine dell'esperienza, ai partecipanti è stato proposto un questionario di soddisfazione soggettiva: i risultati hanno rivelato che i partecipanti alla condizione di formazione IVR sono più soddisfatti delle prestazioni complessive del sistema, rispetto a quelli delle altre due condizioni e che il miglioramento delle prestazioni di formazione IVR è dovuto alla maggiore interattività e alla visualizzazione stereoscopica, utilizzata nell'ambiente di assemblaggio virtuale.

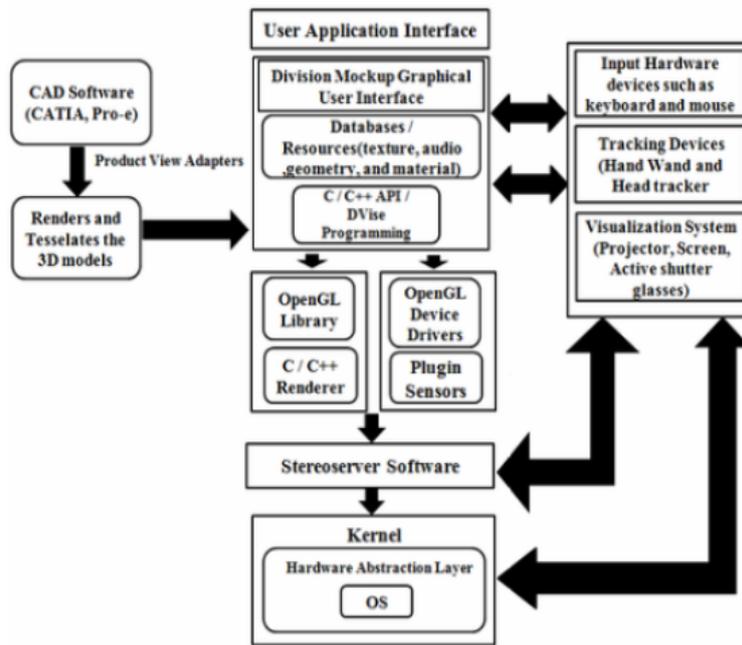


Figura 3.6: Architettura del sistema IVR, Gazal-1, 2010

Esistono diversi metodi formativi in cui le tecnologie VR risultano le più promettenti, soprattutto per quei settori in cui è in gioco la vita umana, come la medicina, l'aviazione e l'aerospaziale. Tuttavia essa ha il limite che, a prescindere da quanto sia realistico l'ambiente virtuale, ci sarà sempre una differenza con l'ambiente reale: per questo motivo, non in tutti i casi, la VR può sostituire completamente i metodi convenzionali di formazione, ma aiuta a sostituirne una parte importante. Al contrario, la realtà aumentata è una tecnologia in gli elementi generati dal computer vengono aggiunti a un ambiente reale, tutta la conoscenza acquisita nella formazione può essere utilizzata direttamente nell'esecuzione del compito.

Uno studio [35] del 2012 presentato dal World Scientific and Engineering Academy and Society (WSEAS) descrive un processo di formazione composto da due fasi: il

primo, utilizzando la VR, in cui l'utente acquisisce alcune delle abilità per sviluppare il compito, il passo successivo sfrutta la AR, in cui l'utente entra in contatto diretto con la macchina, tutti i suoi componenti e parti e integra tutto ciò che ha appreso con il sistema VR. Il sistema di formazione in VR ha un'architettura modulare, con un dispositivo di localizzazione e di visione stereoscopica, integrati nel sistema. Il modulo principale del sistema di formazione in assemblaggio (Assembly Trainer) è responsabile dell'intera logica del sistema ed è gestito dal Centro Tecnologico per l'Intrattenimento della Carnegie Mellon University, e Python 2.7 come linguaggio di programmazione. Questo modulo si chiama Assembly Trainer, ed è responsabile dell'intera logica del sistema.



**Figura 3.7:** A sinistra: sistema di allenamento VR, a destra: sistema di allenamento AR, 2012, Fonte: <https://bit.ly/3zEXTrS>

Il modulo che controlla l'interazione del sistema con il dispositivo di tracciamento ottico si chiama Input Manager, e comunica direttamente con il controllore di tracciamento per ottenere i dati e trasmetterli all'Assembly Trainer. Il sistema di tracciamento ottico utilizzato è un OptiTrack FLEX: V100R2. Lo Stereoscopic Manager è invece il modulo responsabile della visualizzazione dell'immagine 3D: esso crea la doppia finestra e controlla le due telecamere che catturano le immagini da due diverse angolazioni. In sintesi, combinare realtà virtuale e realtà aumentata in un unico processo di formazione risulta estremamente efficace, consente un uso più efficiente delle risorse e una maggiore flessibilità nel tipo di aziende che possono beneficiare di queste tecnologie. La VR e la AR hanno applicazioni diverse in determinate condizioni, rendendone una migliore per alcuni casi e l'altra migliore per altri casi, in modo che possano completarsi a vicenda.

## 3.2 VR per la prototipazione

L'impiego della VR per la prototipazione e per l'arredamento di postazioni ha permesso di semplificare notevolmente processi complessi e costosi: la tecnologia

aiuta, ad esempio, nella pianificazione della produzione e del montaggio. I processi di pianificazione che sfruttano strumenti VR potrebbero, potenzialmente, evitare situazioni di lavoro pericolose o danni fisici causati dalla mancanza di considerazione dell'ergonomia. Il termine ergonomia deriva dal greco (*ergo*: lavoro, *nomos*: norma) ed indica la disciplina scientifica che studia le performance lavorative e il loro benessere, in relazione alle attività svolte da un operatore, alle attrezzature e all'ambiente di lavoro. Di seguito saranno analizzate alcune soluzioni di applicazioni della VR in ambito di prototipazione: nello specifico, saranno descritte un'applicazione commerciale, due trattazioni accademiche, tre soluzioni industriali.

### 3.2.1 Ikea VR Experience

Il primo caso analizzato è IKEA VR Experience, app del 2016 di proprietà del colosso svedese IKEA e disponibile gratuitamente su Steam [36]. Attraverso l'ausilio dei sensori e del sistema di controllo del visore Vive, l'applicazione permette di camminare all'interno di un ambiente virtuale (cucina), cambiare la composizione degli elementi e i materiali in cui sono realizzati, interagire con l'arredamento ad esempio aprendo ante, cassetti, spostando pentole e gettando i rifiuti nei contenitori per la raccolta differenziata.



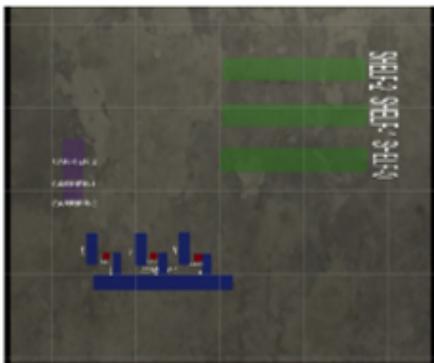
Figura 3.8: IKEA VR Experience, 2016, Fonte: <https://bit.ly/3n8x8XP>

#### Requisiti minimi di sistema

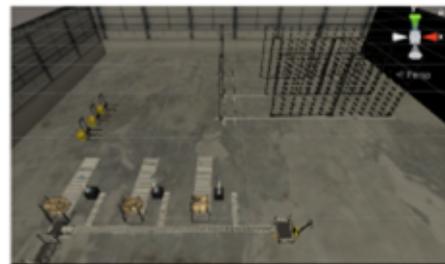
- **Sistema operativo:** Windows 7 o successivi;
- **Processore:** Intel(R) Core(TM) i7 -4790K CPU @ 4.00GHz;
- **Memoria:** 16 GB di RAM;
- **Scheda video:** GeForce GTX 980;
- **Memoria:** 2 GB di spazio disponibile;
- **Scheda audio:** Si;
- **Note aggiuntive:** HTC Vive VR Headset e HTC Controller.

### 3.2.2 Trattazioni accademiche

Nel 2021, una tesi accademica [37] condotta presso l'Università degli Studi di Modena e Reggio Emilia, ha proposto un software, realizzato mediante Unity3D, atto alla progettazione di ambienti in VR non immersiva, per accorciare il più possibile le tempistiche necessarie al completamento delle postazioni. Tale software permette ad un progettista di comunicare, attraverso un file di testo, con un utente esterno in modo tale da tener traccia dei parametri di interesse, passando da una scena 2D ad una 3D, in cui vengono visualizzate le modifiche progettuali. La valutazione degli utenti legata al software risulta positiva. L'autore conclude il suo elaborato proponendo, come sviluppo futuro, l'approdo in ambiente immersivo, in grado di rispondere con relativa immediatezza a qualsiasi esigenza progettuale.



**Figura 3.9:** Lusetti M., Scena 2D, 2021



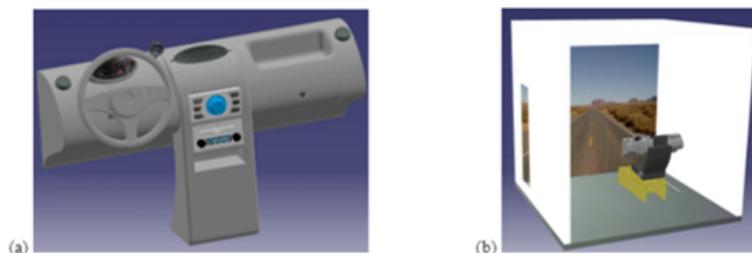
**Figura 3.10:** Lusetti M., Scena 3D, 2021

Nel 2022, il Dipartimento di Progettazione e Gestione Industriale dell'Università di Napoli Federico II ha condotto uno studio [38] che riguarda la progettazione e lo sviluppo di un ambiente IVR per effettuare valutazioni ergonomiche e di usabilità rispetto ad organi di comando di apparecchiature. L'obiettivo finale di quest'attività era quello di realizzare, in modo economico, in termini di tempo ed investimenti, un ambiente di simulazione mediante l'utilizzo di un simulatore meccanico 4 DOF, di un dispositivo aptico dedicato alla simulazione del comportamento fisico di una manopola, integrato in un sistema VR costituito da un HMD, un guanto e un sistema di tracking.



**Figura 3.11:** Soluzione definitiva di progetto e realizzazione del sistema 4DOF, 2022

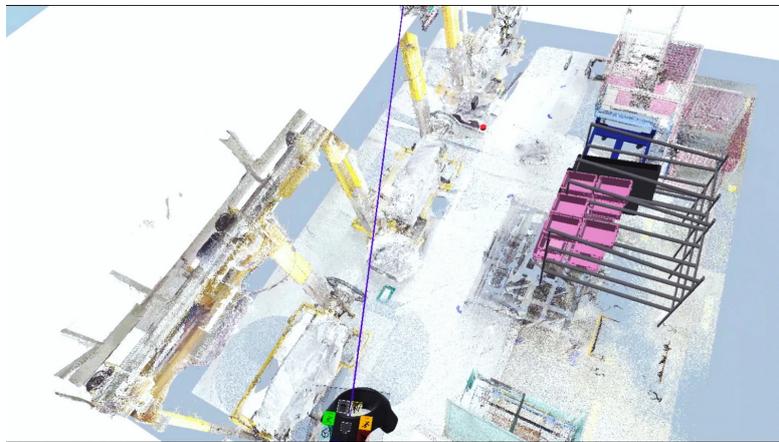
Il caso studio selezionato consiste nella valutazione di usabilità di un comando a manopola dell'aria condizionata sulla plancia di una minicar: i test effettuati hanno dimostrato che l'ambiente IVR realizzato in questo lavoro risulta idoneo a valutazioni di usabilità su organi di comando. Grazie al simulatore costruito, è possibile percepire la sensazione tattile della manopola virtuale, assegnando giudizi positivi sull'interazione con questo comando.



**Figura 3.12:** Plancia modellata per i test e riproduzione virtuale del laboratorio, 2022

### 3.2.3 Applicazioni industriali

Applicando gli concetti analoghi in ambito industriale, la casa automobilistica BMW ha avuto modo di sperimentare la VR attraverso l'installazione virtuale delle singole postazioni di lavoro impiegate per la produzione della BMW Serie 3 Berlina. In particolare, mediante l'ausilio di un visore è stato possibile pre-assemblare virtualmente la cellula di pilotaggio prima che essa venisse materialmente installata nel veicolo. L'impiego di tecnologie VR ha permesso a tutte le risorse umane, impiegate nella linea di montaggio della vettura, di valutare attentamente la nuova area di produzione e di testare nuove procedure in un ambiente 3D. Matthias Schindler [39], responsabile della pianificazione virtuale e dell'implementazione nella produzione di BMW Group ha dichiarato: "La tecnologia della realtà virtuale ci ha permesso di allestire in modo rapido ed efficiente le postazioni di pre-montaggio della cellula di pilotaggio. Non sono state più necessarie lunghe installazioni di prova che replicano le postazioni di lavoro nelle loro dimensioni reali. E il fatto che tutti gli specialisti coinvolti - dagli esperti di logistica, ai progettisti, ai dipendenti di produzione - sono stati in grado di scambiarsi le idee in una fase preliminare è stato un importante vantaggio per il team. Nel complesso siamo stati più trasparenti, più flessibili e più veloci".



**Figura 3.13:** BMW Group Production, Vista 1

Questo metodo abbandona un approccio strettamente sequenziale e si basa invece su un processo agile. Più in generale, BMW Group Production utilizza tecnologie VR di tendenza, facili da utilizzare ed efficaci: le simulazioni VR consistono in potenti strumenti in materia di formazione e qualificazione, pianificazione di postazioni di

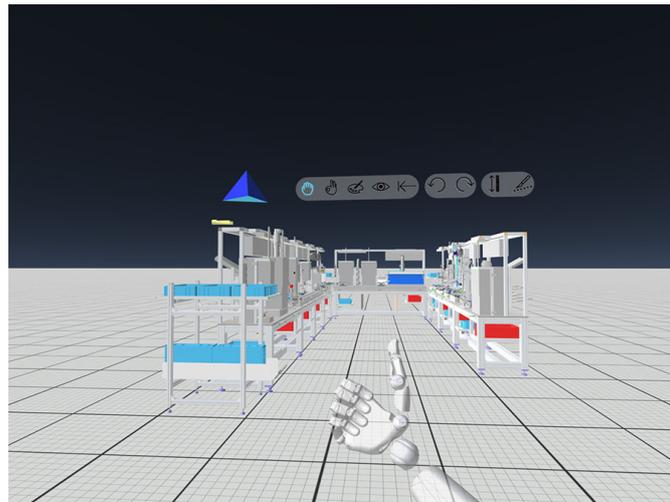
lavoro in catena di montaggio o controllo di qualità. Grazie alla VR, i pianificatori nella costruzione, l'impiantistica, la logistica e l'assemblaggio possono valutare nuove aree di produzione e testare nuovi processi 3D: quando si pianificano le future postazioni, le business unit di BMW Group Production combinano dati esistenti con nuove librerie virtuali, composte da modelli 3D.



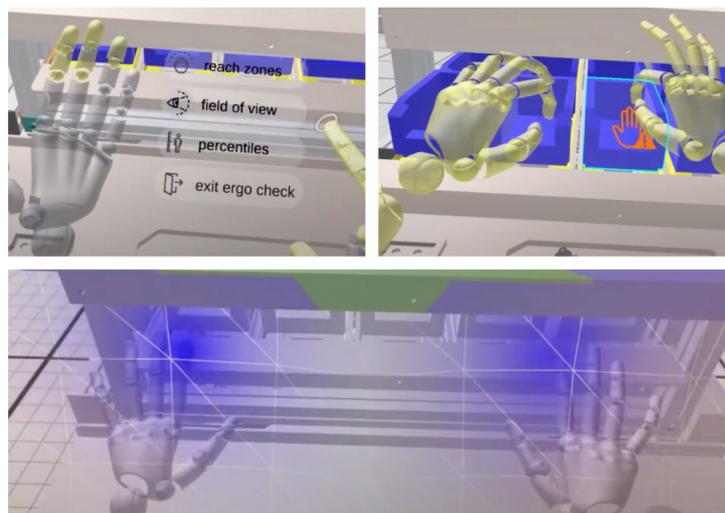
**Figura 3.14:** BMW Group Production, Vista 2

Il software tedesco R3DT [40] offre diversi strumenti intuitivi per lavorare insieme sui progetti di pianificazione: esso permette la progettazione collaborativa dei sistemi di lavoro attraverso occhiali VR. Con l'aiuto del sistema di tracciamento delle mani di Leap Motion [41] è possibile lavorare su un modello virtuale a mani nude, come nel mondo reale, sulla base dei dati 3D esistenti, che vengono proiettati nel mondo VR, premendo semplicemente un pulsante. I prototipi virtuali possono essere modificati o adattati in qualsiasi momento. Ciò consente di eseguire un gran numero di verifiche: di conseguenza, un numero maggiore di iterazioni assicura una riduzione notevole del time-to-market.

Con l'aiuto dell'Ergo-Check in VR, sviluppato da R3DT, le questioni ergonomiche essenziali possono essere chiarite oggettivamente nelle prime fasi: il progettista si posiziona sul prototipo virtuale, si posiziona a diverse altezze del corpo (percentili) e verifica se le zone di portata e i campi visivi sono sufficienti per il montaggio. I display VR "Hand Collider" e "Body Collider" consentono alla persona che agisce di controllare ergonomicamente da quale posizione la mano o il corpo virtuale entrano in collisione con la macchina o un componente.



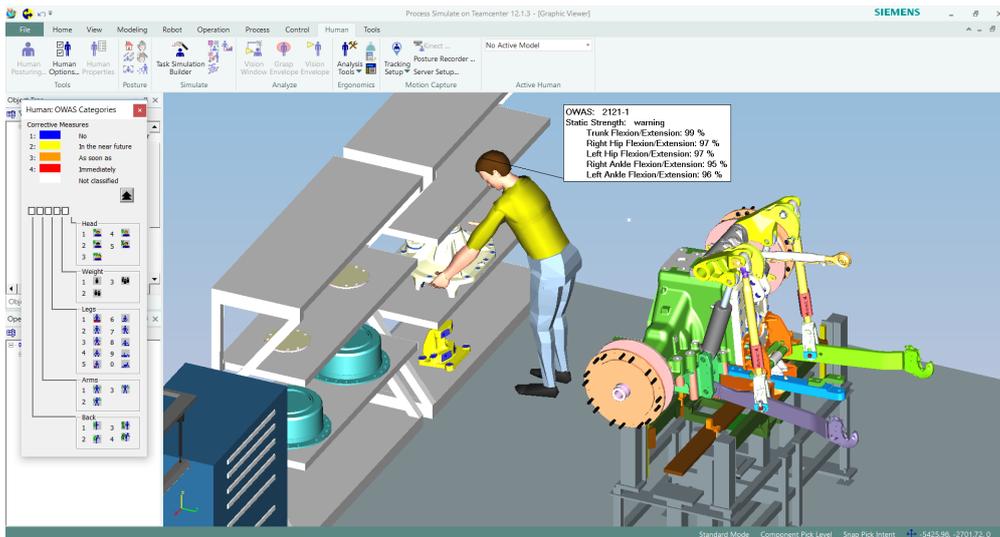
**Figura 3.15:** R3DT



**Figura 3.16:** Ergo-Check, R3DT

La VR ha reso possibile valutare quanto spazio fosse necessario per l'installazione dei macchinari relativi alla linea di montaggio risultato quindi preziosa in una fase preliminare. L'impiego della realtà virtuale parte da un'opera di digitalizzazione che ha interessato le strutture reali ed i macchinari che sono stati replicati con una precisione millimetrica grazie all'utilizzo di speciali scanner 3D e di telecamere ad alta risoluzione. Un sistema che consente la prototipazione virtuale di postazioni di lavoro è Teamcenter VR / Process Simulate (Tecnomatix 16.1) [42] di proprietà

di Siemens AG, multinazionale tedesca fondata nel 1847 attiva nei settori delle tecnologie, della mobilità e dei servizi. Teamcenter VR include l'interazione con il gemello digitale (digital twin) della postazione, ovvero una rappresentazione virtuale connessa con la controparte fisica. Il modello digitale può scambiare dati e informazioni con l'ambiente reale sia in modalità sincrona che asincrona. Teamcenter VR permette all'utente di fare training: egli veste il ruolo dell'operatore



**Figura 3.17:** Interfaccia Tecnomatix Process Simulate, Fonte: <https://bit.ly/3f78dj2>

in VR, utilizza i controller dei dispositivi compatibili (Oculus Rift e HTC Vive) ma vede mani personalizzabili ed può ottenere informazioni relative all'operazione che deve svolgere in real time. Ad esempio il sistema indica in quale punto della stazione l'operatore deve collocarsi per svolgere una determinata simulazione oppure l'ordine dei componenti da manipolare. Teamcenter VR utilizza modelli 3D proprietari in formato .JT, sviluppato direttamente da Siemens PLM Software. Tale software non permette ampliamento del database di primitive, oggetti o postazioni di lavoro da parte di programmatori esterni.

## Capitolo 4

# Tecnologie

L'applicativo realizzato ha l'obiettivo di semplificare e velocizzare la prototipazione di strutture LCA, attingendo ad una libreria predefinita di oggetti e offrendo un sistema di aggancio magnetico tra componenti. Il sistema deve fornire gli strumenti per:

- navigare un ambiente virtuale in modalità immersiva;
- interagire con gli asset;
- addestrare un utente ad utilizzare ogni componente del sistema, tra cui le diverse tecnologie e le loro interfacce di interazione.

Il primo step nel design dell'applicazione consiste nella scelta delle soluzioni hardware e software per la realizzazione. La figura 4.1 permette di confrontare le caratteristiche dei principali visori presenti nel mercato.

A parità di FOV, i dispositivi di casa Valve (HTC Vive Pro, HTC Vive Cosmos e Valve Index) risultano più performanti rispetto ai visori di META (Oculus Go, Oculus Rift S, Oculus Quest) in termini di risoluzione del display, refresh rate e IDP. Il dispositivo scelto è il visore HTC Vive Pro, il più leggero ed economico della famiglia di visori Valve, uno strumento professionale per progetti complessi che offre molte possibilità, tra cui l'integrazione con Unity3D tramite il plug-in SteamVR che consente di gestire le azioni dell'utente per ottimizzare l'esperienza virtuale.

CIRCUIT STREAM Decision Guide: HTC vs. Oculus

	 OCULUS GO	 OCULUS RIFT S	 OCULUS QUEST	 HTC VIVE PRO	 HTC VIVE COSMOS	 VALVE INDEX
<b>TYPE</b>	Self-contained	PCVR	Self-contained	PCVR	Wireless PCVR	PCVR
<b>DISPLAY TYPE</b>	LCD 1280 x 1440 2x	LCD 1280 x 1440	OCLED 1600 x 1400 2x	OCLED 1600 x 1400 2x	LCD 1440 x 1600 2x	LCD 1440 x 1600 2x
<b>REFRESH RATE</b>	72Hz	80Hz	72Hz	90Hz	90Hz	80 - 144Hz
<b>FIELD OF VIEW</b>	101°	110°	110°	110°	110°	130°
<b>IPD</b>	61.5 - 65.5 fixed at 63.5mm	0.5 - 65.5 fixed at 63.5mm	56 - 74 adjustable	60 - 73 adjustable	61 - 72 adjustable	58 - 70 adjustable
<b>WEIGHT</b>	468g	500g	571g	470g	645g	500 - 600g
<b>BATTERY LIFE</b>	1.5 - 2hrs	PC-powered	2 - 3 hrs	PC-powered	PC-powered	PC-powered
<b>DEGREE OF FREEDOM</b>	3DOF	6DOF	6DOF	6DOF	6DOF	6DOF
<b>PRICE</b>	\$149 (32GB) \$199 (64GB)	\$399	\$399 (64GB) \$499 (128GB)	\$599 (headset only)	\$699	\$999 (full VR kit)
<b>RELEASE DATE</b>	May 2018	May 2019	May 2019	January 2018	October 2018	June 2018

Figura 4.1: Visori e relative caratteristiche

Inoltre, dovendo gestire un ambiente 3D ricco di elementi e con interazione in tempo reale, è necessario un elaboratore ad alte prestazioni: il framework è stato sviluppato e distribuito su una workstation dotata di CPU Intel Core i7, scheda grafica NVIDIA GeForce Quadro 4000 e 128 GB di RAM con Windows 10.

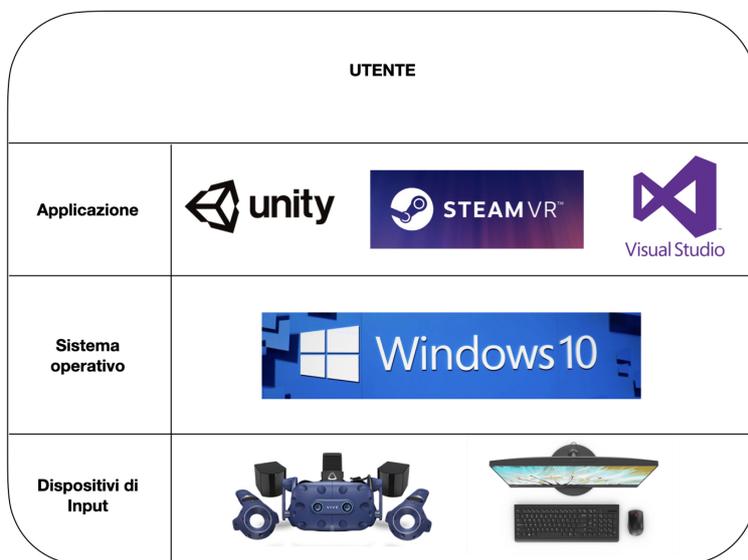


Figura 4.2: Architettura del sistema

## 4.1 Hardware

### 4.1.1 HTC Vive Pro

HTC Vive è un visore di realtà virtuale disponibile sul mercato a partire dal 2016. Si tratta di uno dei dispositivi HMD più immersivi della propria generazione in quanto permette interazioni di tipo visivo, sonoro, aptico, inerziale all'interno di uno spazio fisico predefinito. Il dispositivo è abbinato a due telecomandi (controller) che tracciano i movimenti dell'utente nello spazio e li riproduce nella simulazione virtuale. Di proprietà di Valve, già famosa per la propria piattaforma di gioco, e HTC, casa produttrice di smartphone, il dispositivo possiede potenzialità infinite e risulta uno dei sistemi più promettenti per la VR grazie anche a prezzi accessibili, soprattutto per una PMI. Il costo di questo HMD infatti corrisponde circa a quello di un telefono di fascia alta (800/1000€).



**Figura 4.3:** HTC Vive Pro

Esso risulta ingombrante dal punto di vista ergonomico. La struttura, simile ad occhiali da sci, è composta interamente in plastica rigida ed è caratterizzata da una maschera frontale con numerose rientranze che hanno lo scopo di ospitare i sensori di tracciamento.

Il dispositivo è largo 20 cm, alto 13 cm e pesa circa 470 grammi. Il peso viene discretamente bilanciato sulla testa grazie ad un sistema di cinghie regolabili in tessuto che permettono di alleggerire la pressione del visore sul cranio.

#### **Lenti e regolazioni**

All'interno del dispositivo ci sono due grandi lenti di tipo Fresnel, specifiche per le piccole distanze focali, che presentano anelli concentrici, invisibili una volta indossato l'HMD, che permettono una diversa messa a fuoco. È possibile regolare l'HTC Vive per adattarsi a qualsiasi utente: è presente infatti un pulsante di controllo disposto sul visore per gestire la distanza fra le lenti, per permettere a

chi ha un IDP minore di vedere correttamente all'interno delle lenti. Per garantire l'adattamento del dispositivo a tutti i visi, il dispositivo presenta una serie di guarnizioni in spugna ed è possibile allungare lo spazio al suo interno, tramite un pulsante.

### **Display**

HTC Vive possiede due schermi AMOLED, a bassa persistenza, con risoluzione 1080 x 1200 ciascuno. Il display raggiunge la risoluzione massima di 2160x1200, con una proporzione di 9:5: questa caratteristica rende le immagini leggermente più alte, ma anche più realistiche e naturali una volta indossato il dispositivo.

### **Tracking e sensori**

Il sistema di tracciamento dell'HTC Vive è gestito interamente da una tecnologia ottica chiamata Lighthouse. Essa utilizza diverse camere ad infrarossi wireless (almeno due), dette base station, posizionate agli angoli opposti della stanza. Le base station creano un reticolo a infrarossi di 4.5x4.5 m (room-scale) che interagisce con i sensori posti sul visore e sui controller grazie al sistema di tracking inerziale, composto da giroscopio ed accelerometro. Tale tecnologia permette all'utente di spostarsi in quasi totale libertà all'interno del room-scale e di riprodurre accuratamente, in VR, lo stesso movimento, rendendo molto più immersiva l'esperienza esplorativa.

### **Fotocamera**

HTC Vive presenta frontalmente una fotocamera che permette all'utente di osservare ciò che lo circonda nella realtà, grazie al sistema Chaperon. Gli sviluppatori si sono resi conto che, mentre l'utente indossa il visore, il rischio di scontrarsi con oggetti appartenenti all'ambiente fisico è piuttosto alto. Chaperon è un software di prevenzione per gli incidenti domestici dovuti all'HMD: attivato un opportuno trigger, esso permette di vedere le sagome degli oggetti che potrebbero ostacolare l'utente. Tale sistema, in futuro, potrebbe essere sfruttato anche per la creazione di sistemi Mixed Reality. HTC Vive è collegato al computer mediante un grosso cavo che si biforca in tre più piccoli: un HDMI, un USB 3.0 e la presa dell'alimentazione.

### **Requisiti minimi di sistema**

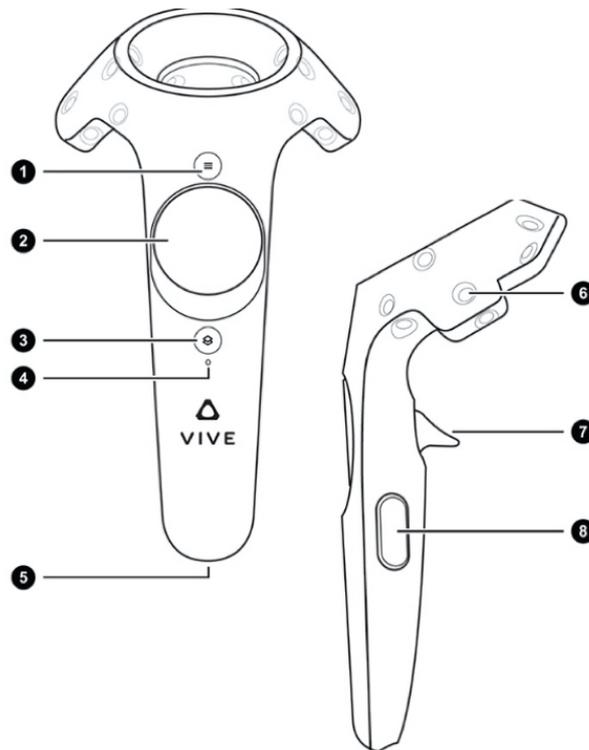
- CPU: processore Intel Core i5 4590 o AMD FX 8350;
- GPU: GeForce GTX 970 o AMD Radeon R9 290;
- RAM: 8GB;
- 2 USB 3.0 e 1 HDMI;

- Window 7 64 bit con Service Pack 1

Trattandosi di un prodotto Valve, il software per la gestione della libreria per la realtà virtuale è Steam VR.

### 4.1.2 HTC Vive Controller

I controller di HTC Vive sono telecomandi pesanti circa 200 g l'uno, lunghi circa 20 cm e larghi 5 cm sul manico mentre si allargano nella sezione tonda, dove sono presenti i sensori. Per rendere l'esperienza più realistica anche sensorialmente è stato inserito un motore di forze feedback che costringe ad esercitare più forza durante l'utilizzo.



**Figura 4.4:** Input di HTC Vive Controller

I controller presentano i seguenti pulsanti:

1. **Menu Button**, bottone per interagire col menu;
2. **Trackpad**, che svolge da solo la maggioranza delle funzioni;
3. **System Button**, bottone per accendere il controller;
4. **Status Light**, LED luminoso ad indicare lo stato di connessione e di carica;
5. **Porta Micro-USB**;
6. **Sensori di Tracking**;
7. **Trigger**, grilletto nella sezione posteriore;
8. **Grip Button**.

L'intero sistema funziona senza fili: non vi sono cavi che collegano i controlli al visore o al PC ma il segnale viene inviato direttamente all'HMD e, successivamente, trasferito al computer tramite cavo o wireless. I controller dispongono di una batteria ricaricabile, con un'autonomia di 3-5 ore.

## 4.2 Software

Il framework è stato sviluppato in Unity 3D (versione 2020.3.21f1) su Windows utilizzando il plug-in SteamVR (versione Steam 2.0) che consente all'applicazione di funzionare sul visore di realtà virtuale immersiva HTC Vive Pro.

### 4.2.1 Unity3D

Unity3D [43] è un motore di sviluppo per la creazione di contenuti interattivi 3D e videogiochi in VR e AR per web, console e mobile, disponibile sia in versione Free che in versione Pro. In Unity è possibile costruire un ambiente VR integrando effetti speciali video e audio, sistemi fisici, animazioni e modelli 3D realizzati con un sistema di modellazione esterno (3Ds Max, Maya, Blender etc.) che consente di esportare file in formati .fbx e .obj. Unity è un motore cross-platform: le applicazioni realizzate possono essere pubblicate su Mac OSX, Windows, Linux, Web Browsers, device iOS e Android, Nintendo Wii, PlayStation e Xbox 360. Il motore grafico Unity3D è stato realizzato in C/C++ ed è in grado di supportare codice scritto in C# e Javascript. Esso è composto da due sottosistemi: Unity Editor, il motore di sviluppo che riguarda l'aspetto interattivo, e lo Unity Engine, il motore di sviluppo che comprende la gestione della fisica, del rendering, della UI e del meccanismo di scripting.

#### Unity Editor

Unity Editor è il motore di sviluppo di Unity che comprende:

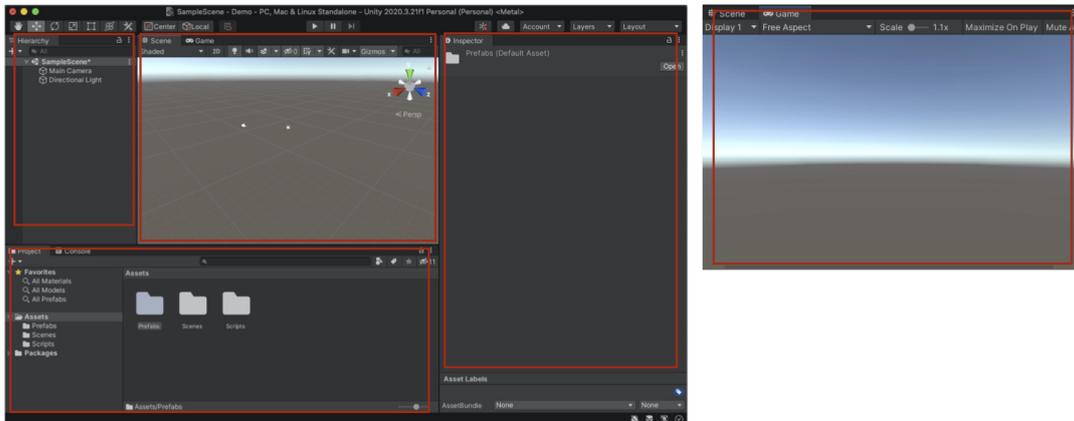


Figura 4.5: Viste di Unity

- **Inspector** : sezione attraverso la quale è possibile aggiungere, manipolare o rimuovere proprietà pubbliche (Component) di un oggetto di scena, detto GameObject. Di base Unity offre come GameObject un insieme di primitive (sfera, cubo, capsula, cilindro, piano), sistemi di illuminazione, telecamera virtuale, elementi per la GUI, effetti speciali etc. GameObjects più elaborati (modelli 3D) possono essere realizzati tramite un programma di modellazione esterno a Unity e successivamente importati in un formato compatibile. Ogni GameObject possiede, di default, il Component Transform che rappresenta la posizione, l'orientamento e la scala dell'oggetto. Altri Components sono gli script, i sistemi fisici e di collisione, gli effetti speciali visivi e sonori, le proprietà di visualizzazione dell'oggetto.  
I modelli 3D possono essere eventualmente essere resi dei Prefabs. Si tratta di GameObjects manipolati che possiedono determinate proprietà universali: non è necessario ripetere l'operazione di cambiare le opzioni dell'oggetto ogni qualvolta importato nella scena. La finestra Inspector permette inoltre di assegnare un tag ad un GameObject;
- **Hierarchy**: sezione popolata da tutti i GameObject aggiunti nella scena, sia attivi che inattivi (non visibili). È possibile creare delle gerarchie di parentela tra gli oggetti, trascinando un GameObject (figlio) all'interno di quello che si intende utilizzare come genitore (padre). Ogni componente associato al padre sarà ereditato dal figlio (ereditarietà): cambiando, ad esempio, le trasformazioni geometriche del padre, tutti i figli avranno una variazione delle proprie trasformazioni geometriche;

- **Project:** gli elementi che compongono l'ambiente virtuale, compresi script, modelli 3D e le scene stesse, vengono salvate all'interno di cartelle e sottocartelle nella sezione Project, per una gestione più organizzata dell'intero lavoro. Le risorse possibili da utilizzare all'interno della scena prendono il nome di Asset. Unity offre degli Standard Assets, ovvero un preset di Asset, da usare all'interno dei progetti.
- **Game Scene:** la sezione Game viene utilizzata ad ogni esecuzione del programma e mostra le modifiche dello stato del sistema in tempo reale;
- **Scene View :** la sezione Scene rappresenta il mondo virtuale che il programmatore sta costruendo. In questa finestra è possibile manipolare tutti i GameObjects presenti, trasformarli (posizione, rotazione, scala), rimuoverli o aggiungerne nuovi.

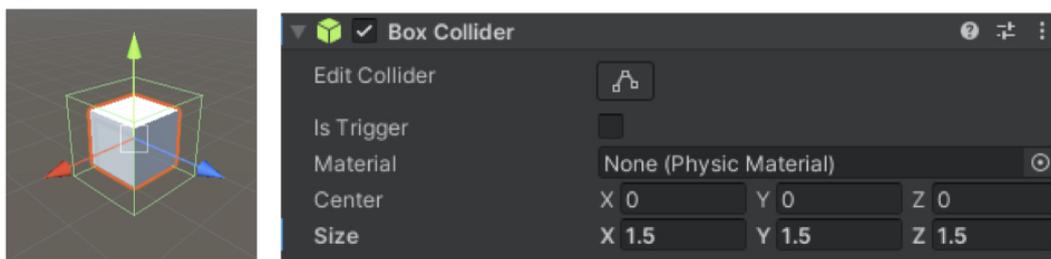
## Unity Engine

Unity Engine è il motore di sviluppo di Unity che comprende:

- **Fisica:** riguarda tutti gli aspetti legati alla simulazione della fisica del mondo virtuale e comprende componenti (Collider, Rigidbody) e funzionalità (Raycasting). In Unity è gestita da NVIDIA Physx, un motore fisico molto potente realizzato da NVIDIA.

### Collider

Un collider è un componente che permette la collisione e l'interazione tra gli oggetti. Dal punto di vista grafico, un collider può essere inteso come la superficie di contorno dell'oggetto, dove avviene la collisione. Dal punto



**Figura 4.6:** Box collider

di vista implementativo, un collider è una classe estesa da numerose classi tra cui BoxCollider (collider a forma di prisma), SphereCollider (collider

a forma di sfera), CapsuleCollider (collider a forma di capsula).

### **Rigidbody**

Un rigidbody è un componente in Unity e permette agli oggetti di agire in risposta all'azione della fisica, essere soggetto alle forze o imprimerle. È possibile stabilire i parametri legati a tali forze selezionando l'oggetto equipaggiato con un rigidbody all'interno dell'Inspector.

### **Raycasting**

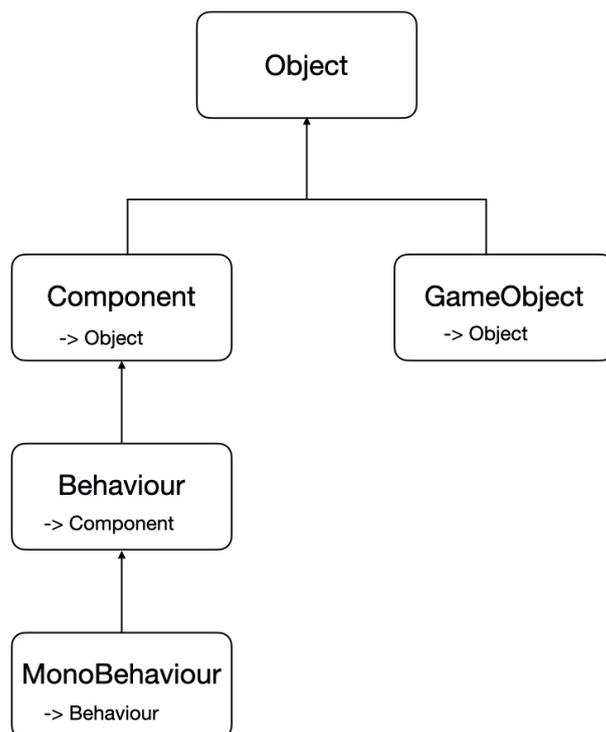
Il raycasting è una funzionalità in Unity e consente di generare un raggio, a partire da una sorgente (es. controller), verso una determinata direzione e con una certa lunghezza, permettendo di sapere se ha colpito o meno qualcosa.

- **Rendering:** il rendering in Unity si basa sui Rendering Paths. Essi determinano la resa grafica della simulazione, tra cui luci ed ombre, e dipendono dalle caratteristiche dell'hardware del dispositivo su cui l'applicazione viene eseguita. Il rendering opera su tre elementi che definiscono le proprietà di visualizzazione di un GameObject: Shaders, Materials e Textures.
- **Scripting:** uno script è un frammento di codice scritto, in questo caso, con il linguaggio di programmazione C#. Esso consiste in una serie di istruzioni, funzioni o comportamenti che gli agenti appartenenti al sistema virtuale devono eseguire, in base alle specifiche del programmatore. In Unity gli script estendono dalla classe MonoBehaviour, la quale a sua volta estende da Behaviour che estende da Component, dunque sono considerati Component.

Uno script può essere assegnato ad un GameObject, trascinando lo script sull'oggetto target oppure aggiungendolo manualmente all'interno dell'Inspector. Alcuni script possono intervenire direttamente sul comportamento dei GameObjects, ne definiscono le azioni e li controllano: questa tipologia di script estende da MonoBehaviour. Altri script invece sono utilizzati unicamente come contenitori di dati (es. Game Manager), non estendono da MonoBehaviour, dunque non sono assegnati direttamente agli oggetti.

Gli script che estendono da MonoBehaviour seguono una determinata struttura composta da una serie di funzioni alle quali è associato un preciso istante di tempo in cui deve essere eseguita da Unity Engine:

1. *Awake()*: funzione invocata una sola volta, nel momento in cui l'oggetto associato allo script è creato;



**Figura 4.7:** Estensione di MonoBehaviour

2. *Start()*: funzione invocata una sola volta, richiamata subito dopo *Awake()*, nel preciso frame in cui lo script viene attivato;
3. *Update()*: funzione invocata ad ogni frame, se lo script è attivato all'interno dell'Inspector. La sua frequenza dipende dalla velocità alla quale il computer su cui è eseguito l'applicativo renderizza le immagini;
4. *FixedUpdate()*: funzione invocata ad intervalli prefissati, non dipende dalle prestazioni del computer;
5. *LateUpdate()*: funzione invocata una volta completate tutte le chiamate alle funzioni *Update()*;

Per questo studio inoltre sono state utilizzate altre funzioni di libreria, offerte da Unity, per la gestione delle collisioni:

1. *OnCollisionEnter(Collision col)*: funzione invocata quando un *GameObject* dotato di un componente di tipo *Rigidbody* entra in collisione con un altro oggetto. L'argomento *col* appartiene alla classe *Collision* e contiene informazioni riguardanti la collisione come, ad esempio, i punti di contatto o la velocità d'impatto;

2. *OnCollisionExit (Collision col)*: funzione invocata quando un corpo esce dalla collisione con un altro oggetto;
3. *OnCollisionStay (Collision col)*: funzione invocata per tutto il periodo di tempo in cui un corpo resta in collisione con un altro oggetto.

### 4.2.2 Visual Studio

Visual Studio [44] è un ambiente utilizzato per sviluppare programmi, siti web, web application e app mobile di proprietà di Microsoft Windows. Microsoft fornisce una versione gratuita di Visual Studio (edizione comunitaria) che supporta i plug-in ed è disponibile gratuitamente. L'ambiente include un editor di codice che supporta IntelliSense, ovvero il componente di completamento del codice, il refactoring del codice e il debugger incorporato, sia a livello di origine che a livello macchina. Supporta diversi linguaggi di programmazione tra cui Java e JavaScript: nel caso di questo progetto, il linguaggio adottato per gli script è C#.

C# (C sharp) è un linguaggio multi-paradigma che ingloba elementi di programmazione imperativa, dichiarativa, funzionale, ed orientata agli oggetti. Sviluppato nei primi anni 2000 da Anders Hejlsberg della Microsoft, come alternativa al linguaggio Java, C# è concepito per essere un linguaggio semplice e moderno e può essere utilizzato con qualsiasi implementazione .NET (.NET Core, .NET Framework etc.).

### 4.2.3 SteamVR Unity plug-in

Steam VR è il software per la gestione della libreria per la realtà virtuale in Unity3D: è necessario anzitutto scaricare il plug-in SteamVR Unity dall'Asset Store [45].

Una volta importato, occorre selezionare il pulsante "Save and generate" per salvare le azioni: il sistema inizializza così alcune classi, in modo tale da facilitare l'accesso ad esse nell'editor e tramite il codice.

Durante le simulazioni VR, l'intero campo visivo umano è coperto dal caschetto, dunque è utile avere rappresentazioni virtuali dei controller che l'utente ha in mano.

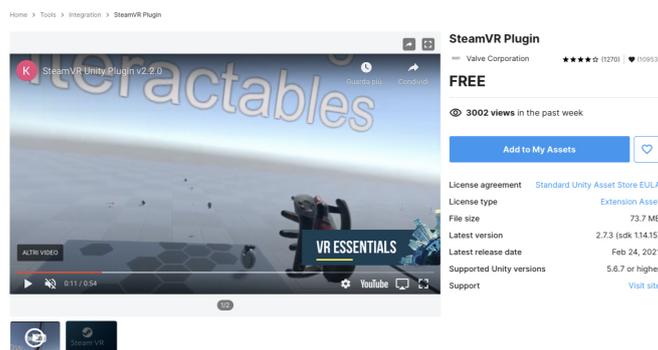


Figura 4.8: Plug-in SteamVR su Unity Asset Store

SteamVR permette di ottenere un modello di controller in VR con componenti ad azionamento individuale, per incoraggiare la sensazione di presenza. Ad esempio premendo il grilletto sul controller nel mondo reale, tale azione sarà replicata anche nel mondo virtuale.

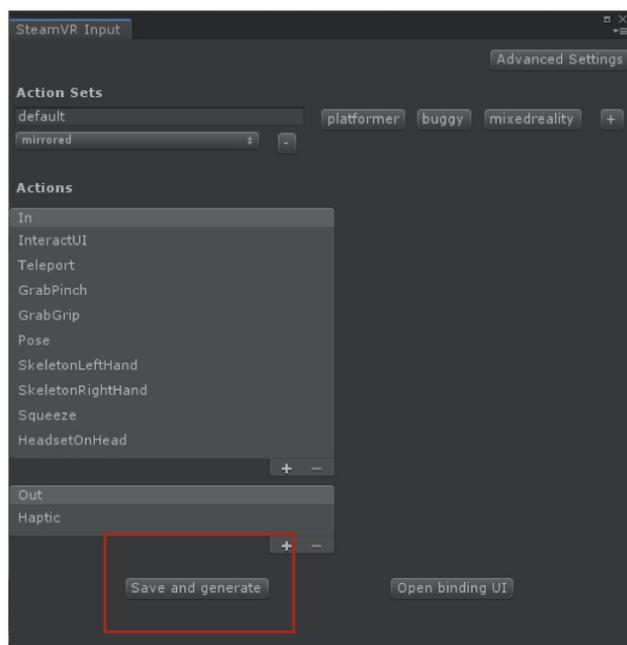


Figura 4.9: Sistema input di Steam VR

### Sistema di input e output

Il maggior contributo che il plugin SteamVR offre per la realtà virtuale riguarda le azioni che l'utente può compiere, attraverso l'hardware. L'utente può configurare

le impostazioni di input attraverso un'interfaccia standard, a seconda delle proprie esigenze e senza intervenire direttamente sul codice.

SteamVR ha separato le azioni in sei diversi tipi di input:

- Boolean: true o false;
- Single: un valore  $x$ ;
- Vector2: due valori  $(x, y)$ ;
- Vector3: tre valori analogici  $(x, y, z)$ ;
- Pose: posizione, rotazione, velocità e velocità angolare;
- Skeleton : orientamenti per ogni osso in una mano;

e un tipo di output:

- Vibration: feedback tattili.

#### **Input dello scheletro** (Skeleton Input)

SteamVR permette all'utente di poter interagire con il mondo virtuale osservando le proprie mani, attraverso i controller. HTC Vive Controller permette di ottenere non solo la pressione dei pulsanti ma anche una buona stima delle dita a mezz'aria, monitorando posizione e rotazione dei giunti.

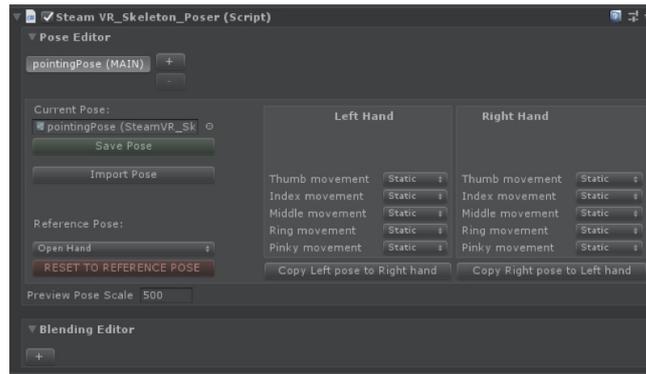
Ci sono due intervalli di movimento per i quali possibile acquisire questi dati ossei:

- **WithController**: gamma di movimento per cui si cerca di stimare dove si trovano le dita dell'utente nel mondo reale, nel miglior modo consentito dal controller;
- **WithoutController**: gamma di movimento completa, da una mano piatta con le dita allargate a un pugno chiuso.

#### **Sistema d'interazione**

Il sistema di interazione di SteamVR consiste in una serie di script, prefab e funzionalità, tra cui Steam VR Input e Skeleton Input.

Il prefab Player imposta il componente principale dell'utente e le mani, agganciandosi a tutte le azioni di input di SteamVR precedentemente configurate: a questo punto è possibile vedere la scena virtuale sul display dell'HMD e le mani che toccano e premono i pulsanti del controller. Player include anche alcune proprietà utili:



**Figura 4.10:** Skeleton Input di SteamVR

1. *HMDTransform( )*: metodo che restituisce la Transform della telecamera corrente;
2. *FeetPositionGuess( )*: metodo che restituisce approssimativamente la posizione dei piedi dell'utente in base alla posizione dell'headset.

La classe Hand compie la maggior parte del lavoro per il sistema di interazione. La mano controlla un oggetto alla volta e invia messaggi al GameObject con cui sta interagendo:

1. *OnHandHoverBegin( )*: inviato quando la mano inizia a librarsi sopra l'oggetto;
2. *HandHoverUpdate( )*: invia ogni fotogramma in cui la mano passa sopra l'oggetto;
3. *OnHandHoverEnd( )*: inviato quando la mano smette di librarsi sull'oggetto;
4. *OnAttachedToHand( )*: inviato quando l'oggetto viene attaccato alla mano;
5. *HandAttachedUpdate( )*: inviato ogni fotogramma in cui l'oggetto è attaccato alla mano;
6. *OnDetachedFromHand( )*: inviato quando l'oggetto si stacca dalla mano;
7. *OnHandFocusLost( )*: inviato quando un oggetto collegato perde il fuoco perché un altro oggetto è stato attaccato.

Per interagire con qualsiasi GameObject presente nella scena occorre aggiungere ad essi il componente Interactable: tutti gli altri componenti su questo oggetto inizieranno a ricevere messaggi rilevanti dalle mani del Player.

Tra le classi Interactable ci sono i componenti Throwable, che permette di lanciare e raccogliere l'oggetto e Skeleton Poser, che conferisce alla mano virtuale una posa approssimativamente corretta quando interagisce con l'oggetto.

#### **Posa dello scheletro** (Skeleton Poser)

Il componente SteamVR\_Skeleton\_Poser funziona indipendentemente dal sistema di interazione SteamVR ed implica una particolare posa della mano e alcune informazioni di animazione per dito. In particolare, è possibile aggiungere e modificare un numero qualsiasi di pose, attraverso l'interfaccia utente Inspector. Esistono diversi tipi di movimento delle dita:

- **Static:** nessun movimento delle dita;
- **Free:** movimento libero delle dita;
- **Extend:** le dita possono sollevarsi fino a una posizione completamente estesa, ma non piegarsi più in basso rispetto a dove le ha la posa standard;
- **Contract:** le dita possono piegarsi in una posizione completamente contratta, ma non possono aprirsi oltre il punto in cui le ha la posa standard.

L'ultimo elemento che permette di creare una scena VR immersiva di base è il Teleport. Questo prefab permette di impostare tutta la logica del teletrasporto, insieme ai componenti TeleportPoint e TeleportArea, per aggiungere posizioni in cui l'utente può teletrasportarsi.

Infine, per la realizzazione di interfacce grafiche (menu), è possibile sfruttare gli elementi messi a disposizione da Unity (Canvas, Panel) oppure creare sistemi d'interazione diegetici, come i pulsanti. I pulsanti fisici risultano più soddisfacenti da usare rispetto alle interfacce non diegetiche, ma i sistemi di interazione fisica possono diventare complessi rapidamente. Il prefab Player crea un InputModule che consente alle mani di imitare gli eventi del mouse per interagire facilmente con le GUI.

Il sistema di interazione include una modalità Fallback 2D che consente i tipici controlli della Player utilizzando la tastiera e il mouse che agisce come una delle mani dell'utente: questa modalità è utile per testare interazioni molto semplici durante lo sviluppo dell'applicativo.



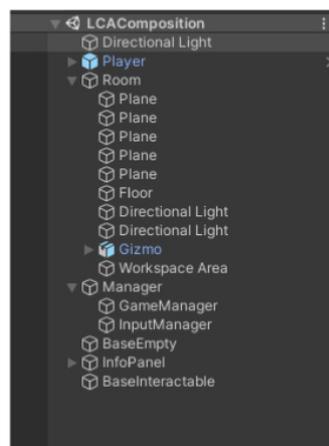
# Capitolo 5

## Sviluppo e design

In questo capitolo saranno evidenziate le caratteristiche e le funzionalità del sistema realizzato. Verrà posto l'accento sulle strategie adottate per garantire immersione, interazione, presenza all'utente e sarà esaminata nel dettaglio la costruzione delle architetture viste nel capitolo precedente.

### 5.1 Impostazioni generali

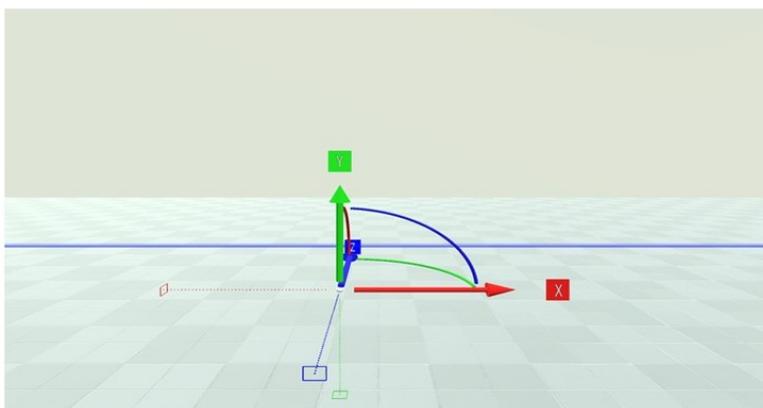
La prototipazione della postazione di lavoro avviene in una stanza vuota all'interno della quale l'utente potrà svolgere le operazioni.



**Figura 5.1:** Impostazioni generali dell'ambiente virtuale

I GameObjects che consentono la costruzione di una stanza sono:

- **Directional Light:** insieme delle luci di scena;
- **Player:** componente principale dell'utente e delle mani;
- **Room:** insieme di soffitto, pavimento e pareti della stanza entro i quali l'utente è libero di muoversi (real walking). Il sistema permette di stabilire i confini virtuali per evitare incidenti durante la sessione: l'area di sicurezza è calibrata in base alla posizione dei tracker ottici di HTC Vive nello spazio reale ed è evidenziata, all'interno della stanza virtuale, da un rettangolo blu. Infine, al centro della scena, è presente un sistema di assi cartesiani ortogonali Gizmo ( $X, Y, Z$ ), riferimento per localizzare gli oggetti e l'utente;
- **Manager:** sistema di gestione globale della scena; ad esso sono associati i componenti per il rilevamento delle azioni dell'utente, degli input e delle collisioni;
- **Info Panel:** insieme dei componenti utili per il controllo del sistema;
- **Base Interactable** e **Base Empty:** risorse aggiuntive per la costruzione di nuovi prefab.



**Figura 5.2:** Assi Gizmo al centro dell'ambiente virtuale: asse  $x$  in rosso,  $y$  in verde,  $z$  in blu e area di sicurezza

Gli assets principali sono:

- **Icons:** insieme delle icone e dei loghi presenti nell'applicativo;
- **Materials:** insieme dei materiali attribuiti agli oggetti;
- **Resources:** contiene i prefab e tutti gli elementi richiamati via script durante la simulazione, tramite il metodo *Instantiate()*;

- **Scenes**: contiene la scena principale;
- **Script**: insieme degli script;
- **SteamVR, SteamVR Input, SteamVR Resources, Streaming Assets, XR**: prefab, script e altri componenti offerti da Unity e dal plug-in Steam VR;
- **TextMesh Pro**: componenti per la gestione dei testi.



**Figura 5.3:** Assets

## 5.2 Controllo del sistema

L'operatore può interagire con tutti gli oggetti di scena presenti tramite i controller HTC Vive i quali, nell'ambiente virtuale, assumono le sembianze di una mano. I pulsanti che permettono le interazioni sono:

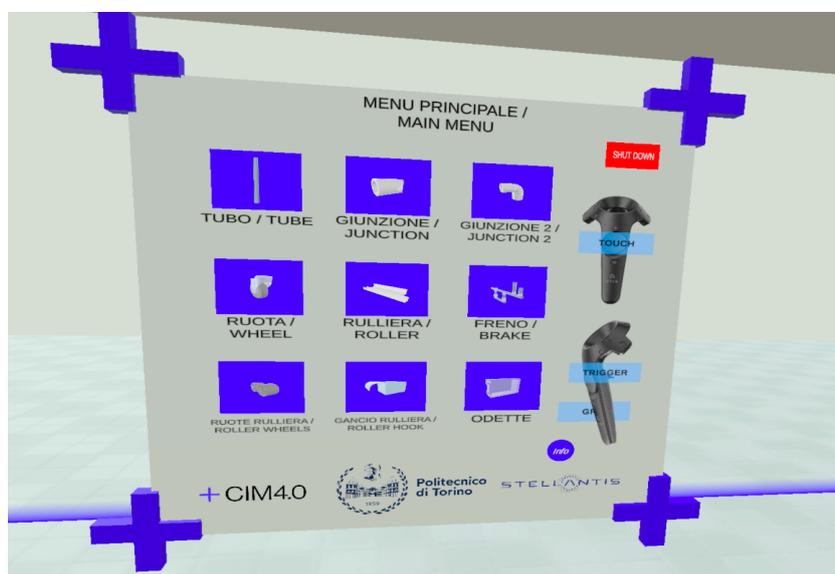
- *Menu*: consente l'apertura e la chiusura dei menu di scena (Menu principale e Menu dell'oggetto), ovvero pannelli che si integrano con la narrazione virtuale, il cui design ricalca lo stile grafico del CIM 4.0, caratterizzato dai colori blu e grigio metallico;
- *Trigger*: permette la chiusura completa della mano e consente all'operatore di afferrare oggetti;
- *Grip*: premendo questo pulsante, le dita della mano virtuale corrispondente risulteranno serrate ed emergerà solo l'indice puntato;
- *Touch*: permette il movimento del pollice.

L'utente può selezionare un'opzione corrispondente ad un bottone colpendo lo stesso con la mano virtuale, di conseguenza il sistema riconosce una collisione tra la mano ed il bottone:

1. *OnCollisionEnter()*: il bottone acquisisce un colore giallo;

2. *OnCollisionExit()*: il bottone torna blu ed avviene il cambiamento di stato del sistema.

Il **Menu principale/Main menu** permette all'utente di aggiungere nuovi oggetti all'ambiente virtuale tramite la funzione *Instantiate()*, andando a creare un'istanza (ovvero un clone) dell'oggetto prefab che si va a richiamare tra parentesi. Il pannello contiene diversi bottoni raffiguranti gli oggetti, un bottone **Info** per visualizzare un ulteriore pannello contenente la lista dei comandi, delle istruzioni e delle funzioni dell'applicazione, un bottone per chiudere l'applicazione. Nel Menu principale, infine, sono presenti le proiezioni ortogonali del controller con le etichette indicanti i pulsanti: quando l'utente preme uno di essi, l'etichetta corrispondente sarà evidenziata. Per chiudere questo menu è sufficiente premere nuovamente il pulsante *Menu* del controller.

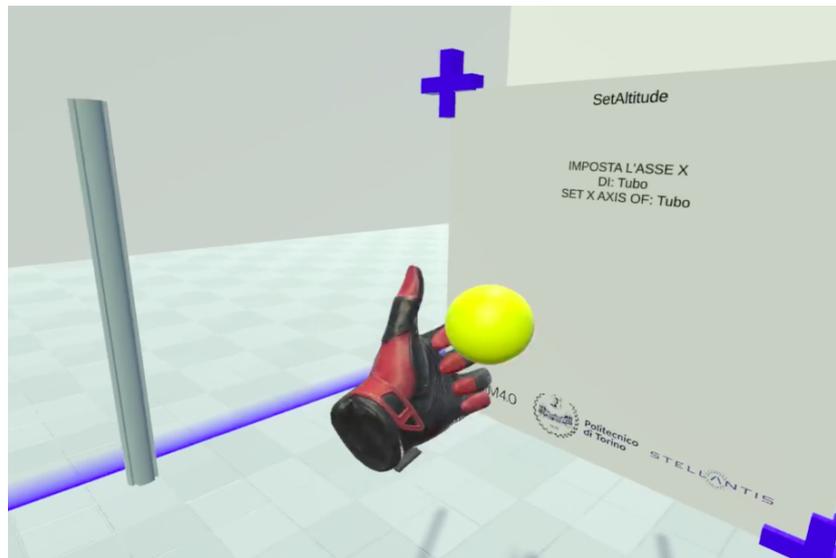


**Figura 5.4:** Menu principale

Le funzioni offerte dall'applicazione sono:

- **Copy Dimensions From/ Copy Dimensions To:** copia le informazioni di scala da un oggetto A ad un oggetto B (e viceversa), non è possibile copiare le dimensioni di due oggetti di diverso tipo;
- **Delete:** distrugge l'oggetto;
- **Gravity:** attiva la forza di gravità;
- **Split:** disassembla Game Objects composti;

- **Set Axis:** copia le informazioni di posizione di un oggetto A in un oggetto B, rispetto ad uno dei tre assi ortogonali;
- **Set Dimensions:** afferrando un oggetto con entrambe le mani e allargando o chiudendo le braccia, è possibile variare la scala dell'oggetto.



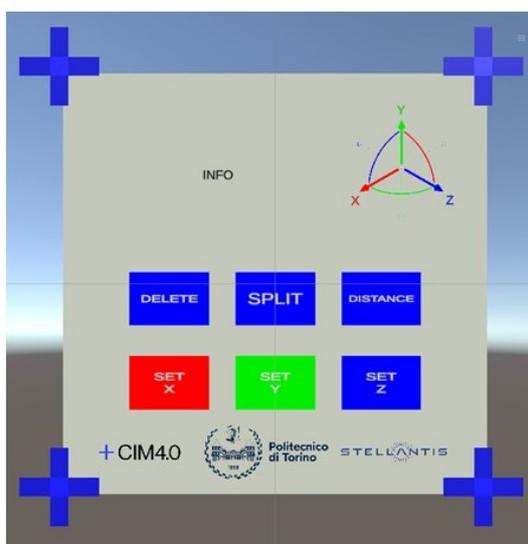
**Figura 5.5:** Pointer per utilizzare una funzione

È possibile utilizzare le funzioni avvicinando la mano virtuale ad un oggetto (di conseguenza il suo bordo risulterà evidenziato di giallo), detto **oggetto attivo** o **reference**, premendo il pulsante *Menu* del controller. L'operatore avrà a disposizione un pannello (**Menu dell'oggetto/Object menu**) che contiene informazioni specifiche del Game Object stesso e diversi bottoni corrispondenti alle funzioni disponibili. Selezionando le funzioni Copy Dimensions e Set Dimensions compare una sfera gialla (**Pointer**), accompagnata da un nuovo pannello sul quale è indicata la funzione di riferimento. Per eseguire l'operazione, è sufficiente afferrare il Pointer e colpire l'oggetto da cui si desidera copiare o impostare la dimensione. Per disattivare la funzione, quindi distruggere il Pointer, è sufficiente premere il pulsante *Menu*. L'operatore può afferrare liberamente un oggetto per spostarlo o ruotarlo, premendo il pulsante *Trigger*. Al rilascio, l'oggetto si posizionerà in modo ortogonale agli assi Gizmo, approssimando i valori di rotazione lungo  $(x, y, z)$  a  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$ ,  $360^\circ$ .

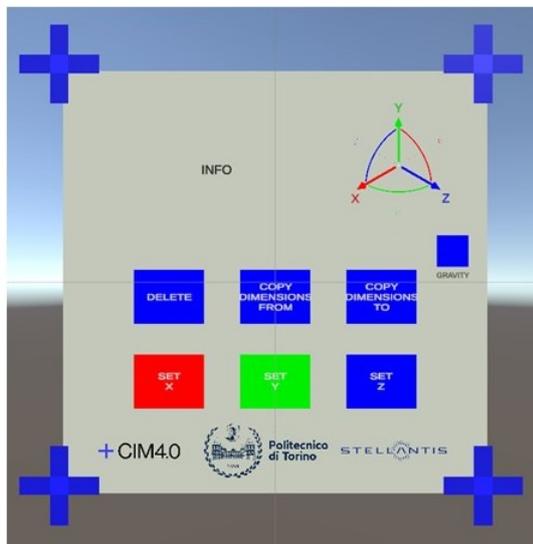
## 5.3 Classi di oggetti

Esistono tre tipologie di oggetti:

- **Oggetti singoli:** prefab i quali non possiedono né un GameObject padre né figli. Alcuni oggetti singoli possiedono **Collider Points**, ovvero punti in cui è possibile effettuare operazioni di unione con altri oggetti;
- **Oggetti composti:** Game Objects formati da più oggetti singoli, uniti da vincoli di parentela, è possibile separarli mediante la funzione Split. Alcuni oggetti composti possiedono Collider Points;
- **Strutture complesse:** strutture formate da più oggetti composti, non sono uniti da vincoli di parentela, è possibile separarli.



**Figura 5.6:** Menu dell'oggetto (oggetti composti)



**Figura 5.7:** Menu dell'oggetto (oggetti semplici)

### 5.3.1 Oggetti singoli

Gli oggetti singoli sono prefab invocabili attraverso il Menu principale. Alcuni oggetti possiedono uno o due Collider Points (punti di aggancio) che corrispondono, all'incirca, alle proprie estremità.



**Figura 5.8:** Lista di prefab

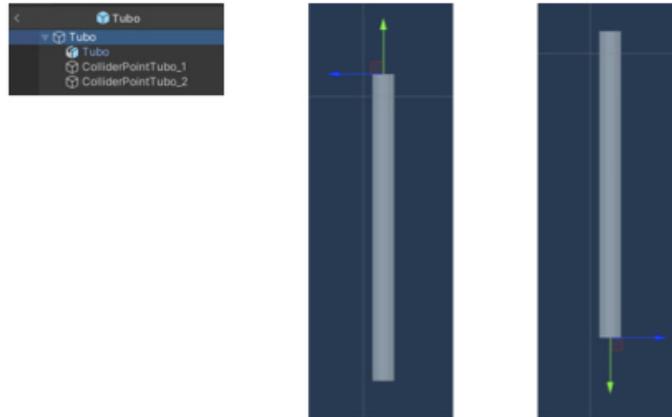
Di seguito saranno descritti gli oggetti singoli offerti dall'applicativo, le proprietà visualizzate sul Menu dell'oggetto, i rispettivi Collider Points, gli oggetti con i quali è possibile comporne nuovi, le funzioni associate:

Oggetto	Caratteristiche	Può unirsi a	Collider Points	Funzioni associate
<b>Tubo</b>	Posizione Rotazione Lunghezza del tubo	Giunzione Ruota Freno	Un C.P. per ciascuna estremità del tubo	Copy Dimensions, Delete, Set Axis, Set Dimensions
<b>Giunzione</b>	Posizione Rotazione	Tubo	Un C.P. nel punto di aggancio interno	Delete, Set Axis
<b>Giunzione 2</b>	Posizione Rotazione	Tubo	Un C.P. nei due punti di aggancio interno	Delete, Set Axis
<b>Ruota</b>	Posizione Rotazione	Tubo	Un C.P. nel punto di superiore	Delete, Set Axis
<b>Freno</b>	Posizione Rotazione	LCA Braccio (oggetto composto)	Un C.P. in un punto prossimo al tubo di freno	Delete, Set Axis
<b>Rulliera</b>	Posizione Rotazione Lunghezza della rulliera	Gancio per la rulliera	Un C.P. per ciascuna estremità del trafilato	Copy Dimensions, Delete, Set Axis, Set Dimensions
<b>Gancio per la rulliera</b>	Posizione Rotazione	Rulliera	Un C.P. nel punto di aggancio interno	Delete, Set Axis
<b>Ruote per la rulliera</b>	Posizione Rotazione	LCA Rulliera (oggetto composto)	Un C.P. nel centro della geometria dell'oggetto	Delete, Set Axis
<b>Odette</b>	Posizione Rotazione	-	-	Gravity, Delete, Set Axis, Set Dimensions

Tabella 5.1: Oggetti singoli

### 5.3.2 Oggetti composti

L'utente può unire diversi oggetti per crearne uno composto: è sufficiente afferrare un oggetto ed avvicinarlo ad un altro per far sì che questo sia attratto magneticamente. È necessario impostare a priori la lista di oggetti con i quali un oggetto attivo può essere unito e la posizione esatta dei Collider Points all'interno del modello 3D.



**Figura 5.9:** Collider Points

L'operazione di assemblaggio corrisponde ad un algoritmo che verifica:

- la collisione tra i bounding boxes dei due oggetti;
- la possibilità di unione, scorrendo la lista associata ad ogni oggetto;
- la presenza di Collider Points liberi;
- il Collider Point più vicino al punto in cui è avvenuta la collisione.

In fase di collisione, l'oggetto che si trova in mano all'utente è chiamato *Colliding Object*, l'oggetto urtato è detto *Collided Object*. Quest'ultimo si posiziona in modo tale che il Collider Point individuato dall'algoritmo e quello del Colliding Object coincidano. Per garantire la corretta posizione ed il giusto orientamento del Collided Object, l'asse *Y* di ciascun Collider Point è rivolto sempre con verso uscente. Una volta rilevata la collisione, il Collided Object si posiziona in modo tale che l'asse *Y* dei Collider Point interessato e quello del Collider Point del Colliding Object condividano la stessa direzione ma verso opposto. Di conseguenza, il Collided Object diventa gerarchicamente figlio del Colliding Object.

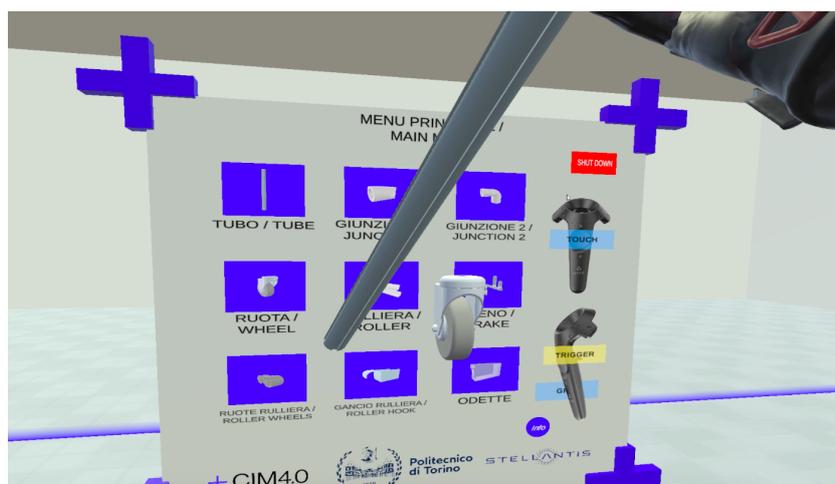


Figura 5.10: Oggetti semplici per la realizzazione di un oggetto composto

L'operatore può afferrare liberamente un oggetto composto avvicinando la mano ad esso e premendo il pulsante *Trigger*, al momento del rilascio, l'oggetto si posizionerà in modo ortogonale agli assi Gizmo, analogamente agli oggetti singoli.

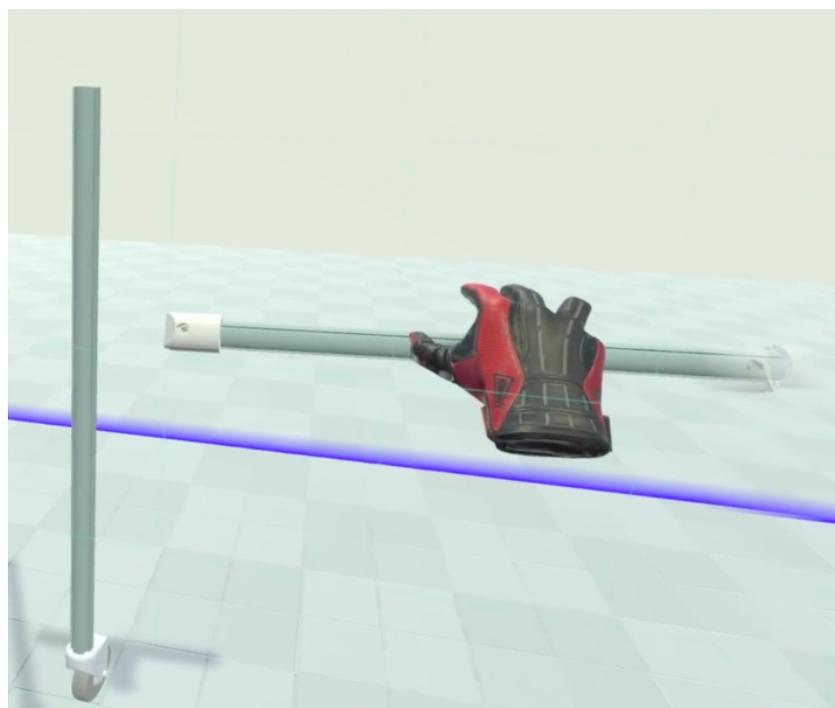


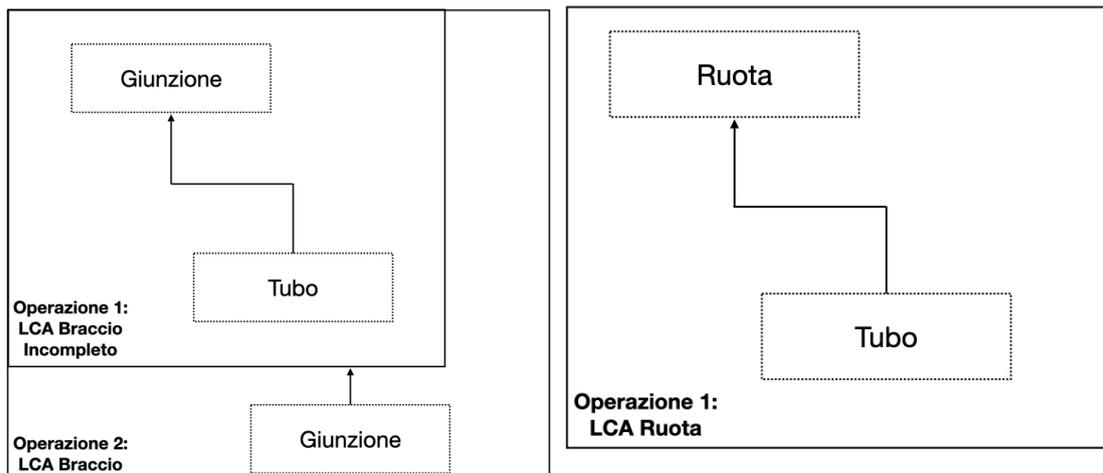
Figura 5.11: Oggetti composti, a sinistra: LCA Ruota, a destra: LCA Braccio

Di seguito saranno descritti gli oggetti composti offerti dall'applicativo, le proprietà visualizzate sul Menu dell'oggetto, gli oggetti da cui è formato, gli oggetti con i quali è possibile comporne nuovi, i rispettivi Collider Points:

Oggetto	Caratteristiche	Composto da	Può unirsi a	Collider Points
<b>LCA Braccio</b>	Posizione, Rotazione, Lunghezza del tubo da cui è composto	1 Tubo 2 Giunzioni	LCA Ruota per formare una struttura complessa	Un C.P. alle estremità dell'oggetto
<b>LCA Ruota</b>	Posizione, Rotazione, Lunghezza del tubo da cui è composto	1 Tube 1 Ruota	LCA Braccio per formare una struttura complessa	-
<b>LCA Rulliera</b>	Posizione, Rotazione, Lunghezza del tubo da cui è composto	1 Rulliera 2 Ganci per rulliera	-	-
<b>LCA Rulliera completa</b>	Posizione, Rotazione, Lunghezza del trafilato da cui è composto	1 LCA Rulliera 1 Ruote per la rulliera	LCA Braccio per formare una struttura complessa	Un C.P. alle estremità dell'oggetto

**Tabella 5.2:** Oggetti composti

Tutti gli oggetti composti possiedono le funzioni Set Axis, Split e Distance. Quest'ultima, esattamente come Set Axis e Set Dimensions, offre all'utente un Pointer. Colpendo un secondo oggetto, il pannello associato al Pointer mostrerà la distanza, espressa in metri, tra l'oggetto attivo e quello colpito. Inoltre, nel caso in cui l'oggetto attivo e quello colpito siano di tipo LCA Braccio, il pannello indicherà la dimensione che occorre ad un oggetto LCA Rulliera, per coprire tale distanza. In ordine gerarchico, gli oggetti di tipo Tubo o Rulliera sono sempre padri rispetto ai Collided Objects con i quali si uniscono. Alcuni oggetti composti possiedono dei Collider Points i quali permettono l'assemblaggio dello stesso con altri oggetti, formando strutture complesse.



**Figura 5.12:** Realizzazione di LCA Braccio

**Figura 5.13:** Realizzazione di LCA Ruota

Inoltre, alcuni oggetti possiedono alcune proprietà legate alla geometria del modello 3D e alle funzionalità specifiche:

- **Freno:** la collisione tra un oggetto semplice Freno e un oggetto composto LCA Braccio permette all'operatore di fissare un ostacolo verticale per impedire, ad esempio, lo scorrimento di un carico. In particolare, il Freno sarà posizionato in modo tale che il tubo che compone LCA Braccio sia incastrato tra i due trafilati verticali che fungono da ostacolo, in un punto specifico detto **Lock Point**;
- **LCA Rulliera:** la collisione tra un oggetto composto LCA Rulliera e un oggetto semplice Ruote permette all'operatore di creare un LCA Rulliera completa con un numero di ruote proporzionale alla lunghezza del trafilato.

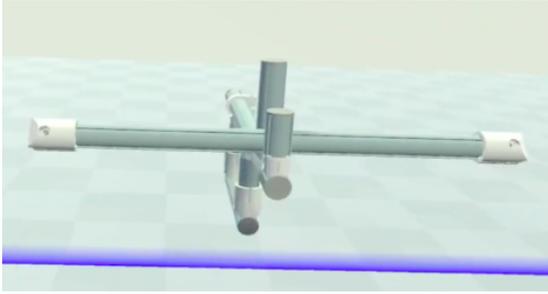


Figura 5.14: Freno e LCA Braccio

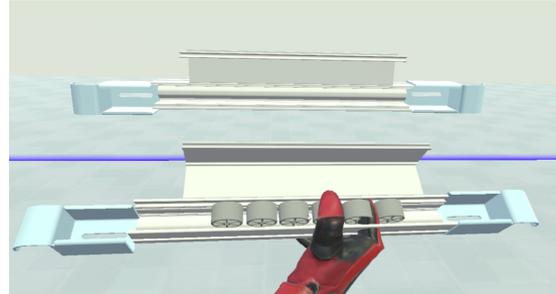


Figura 5.15: LCA Rulliera e LCA Rulliera Completa

### 5.3.3 Strutture complesse

L'operatore può assemblare oggetti composti per realizzare strutture complesse, formata da almeno un oggetto di tipo LCA Braccio ed uno LCA Ruota. È possibile creare strutture complesse utilizzando un numero variabile di oggetti, in base alle esigenze di progettazione dell'operatore. Nello specifico, l'oggetto LCA Braccio possiede un Collider Point per ciascuna delle due estremità, indicati come **Lock Points**. Dopo aver verificato la collisione tra i bounding boxes ed effettuato il controllo per l'unione dei due oggetti, l'algoritmo di assemblaggio ricerca i Lock Points liberi in LCA Braccio.

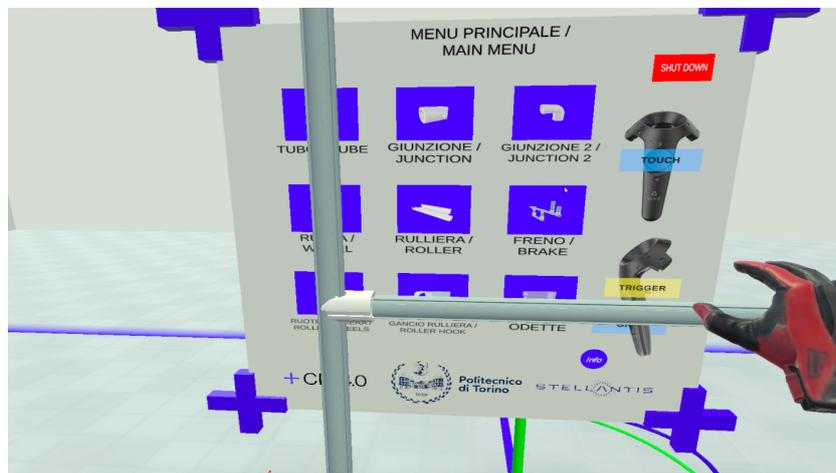


Figura 5.16: Struttura complessa

La corretta posizione e il corretto orientamento di LCA Braccio, rispetto a LCA

Ruota, sono identificati da un algoritmo che verifica la distanza tra le componenti  $x$  e  $z$  dei due oggetti. Le operazioni eseguite sono:

1. calcolo della differenza tra la coordinate di posizione  $x$  di LCA Braccio e LCA Ruota (distanza lungo  $x$ );
2. calcolo della differenza tra la coordinate di posizione  $z$  di LCA Braccio e LCA Ruota (distanza lungo  $z$ );
3. scelta dell'asse di LCA Ruota lungo il quale disporre LCA Braccio: se la distanza lungo  $x$ , calcolata al punto 1) risulta minore della distanza lungo  $z$ , ottenuta al passo 2), LCA Braccio sarà disposto lungo l'asse  $z$  di LCA Ruota, viceversa sarà collocato lungo il suo asse  $x$ ;
4. scelta della posizione di LCA Braccio rispetto a LCA Ruota: se la coordinata  $x$  ( $z$ ) di LCA Braccio è maggiore rispetto alla  $x$  ( $z$ ) di LCA Ruota, l'oggetto sarà posizionato lungo l'asse  $x$  ( $z$ ) positivo di LCA Ruota, viceversa sarà posizionato lungo il suo asse negativo. L'oggetto LCA Braccio sarà posto nel punto  $x$  ( $z$ ) individuato dalla somma tra  $x$  ( $z$ ) di LCA Ruota, metà della lunghezza del tubo di cui LCA Braccio è composto, lunghezza della giunzione.

Ciascun oggetto (singolo o composto) possiede un componente che permette di identificare e gestire una struttura complessa:

- *ID* (di tipo intero): prima di costituire una struttura complessa, ciascun oggetto possiede un ID pari a -1; in seguito ad una collisione, tutti gli oggetti che formano la struttura assumono un ID univoco. La collisione tra due oggetti con ID uguale a -1 corrisponde alla creazione di una nuova struttura complessa. Ogni ID è assegnato da un componente globale (nel Game Manager), incrementato ogni qualvolta l'utente crea una nuova struttura complessa, partendo da 0. La collisione tra un oggetto, con ID uguale a -1, ed una struttura complessa, con ID diverso da -1, l'ID dell'oggetto pari a quello della struttura complessa;
- Lista che contiene tutti i GameObjects con cui l'oggetto attivo ha colliso per la realizzazione della struttura;
- *LockVector* (di tipo Vector3): vettore che contiene le coordinate di posizione dell'oggetto una volta che entra a far parte della struttura complessa e che sia aggiorna ogni qualvolta l'utente muove l'intera struttura oppure il singolo componente.

L'operatore può interagire con una struttura complessa utilizzando i tasti Trigger, Touch e Grip del controller HTC per eseguire una specifica operazione di manipolazione:

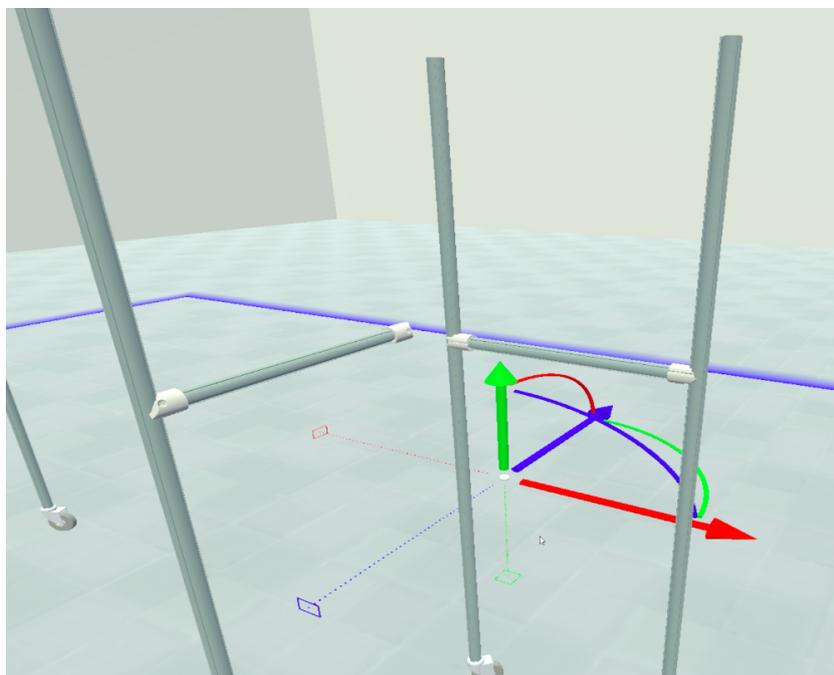
- *Trigger*: traslazione rigida dell'intera struttura. È sufficiente afferrare un singolo componente (oggetto attivo) per spostare tutti i GameObjects che ne condividono lo stesso ID. La posizione aggiornata degli oggetti non attivi è calcolata sommando un offset al proprio Lock Vector. L'offset corrisponde alla differenza tra il valore corrente di posizione dell'oggetto attivo e il suo Lock Vector. Al rilascio, i Lock Vectors dei componenti della struttura complessa sono aggiornati;
- *Trigger + Touch*: traslazione rigida di un singolo componente della struttura composta verso l'alto o il basso. In particolare, le coordinate x e z dell'oggetto attivo sono sempre uguali a  $LockVector.x$  e  $LockVector.z$ . Al rilascio,  $LockVector.y$  del componente traslato è aggiornato col valore della posizione corrente;
- *Grip*: separazione di un componente dalla struttura complessa. Separando gli ultimi due oggetti della struttura complessa, essa cessa di esistere. L'ID dell'oggetto separato assume il valore dell'ID precedente all'assemblaggio, analogamente avviene per la lista. Le liste dei singoli componenti della struttura complessa non conterranno più il riferimento al GameObject separato.



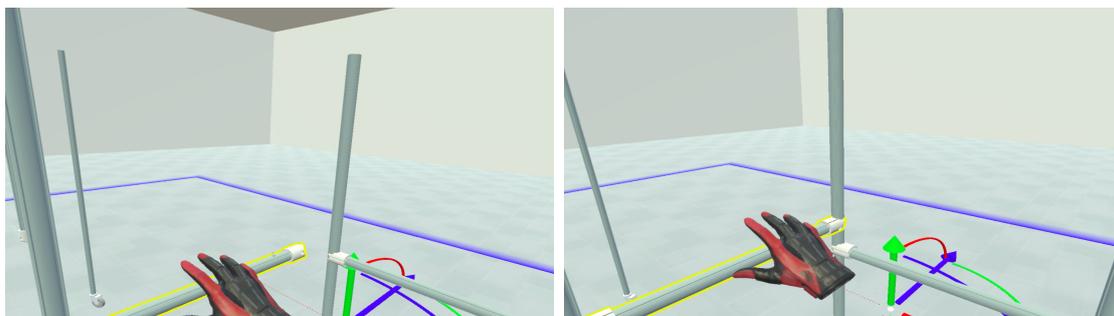
**Figura 5.17:** Traslazione di un componente della struttura complessa

L'operatore può unire due strutture complesse per crearne una nuova: è sufficiente che l'oggetto composto attivo (componente della prima struttura complessa) collida con un componente della seconda struttura complessa con cui possa unirsi. Ad esempio l'utente può afferrare un LCA Braccio di una generica struttura complessa A per unirlo ad un LCA Ruota facente parte della struttura complessa B. Di conseguenza la struttura complessa B acquisirà l'ID della struttura complessa A,

la lista dei Collided Object di LCA Braccio ingloberà LCA Ruota, LCA Ruota aggiungerà LCA Ruota alla sua lista di Collided Object.



**Figura 5.18:** Assemblaggio di due strutture complesse



**Figura 5.19:** Assemblaggio di due strutture complesse: l'utente afferra LCA Braccio della struttura complessa a sinistra e la muove verso la LCA Ruota della struttura a destra  
**Figura 5.20:** Assemblaggio di due strutture complesse: collisione tra LCA Braccio della struttura complessa a sinistra e la LCA Ruota della struttura a destra

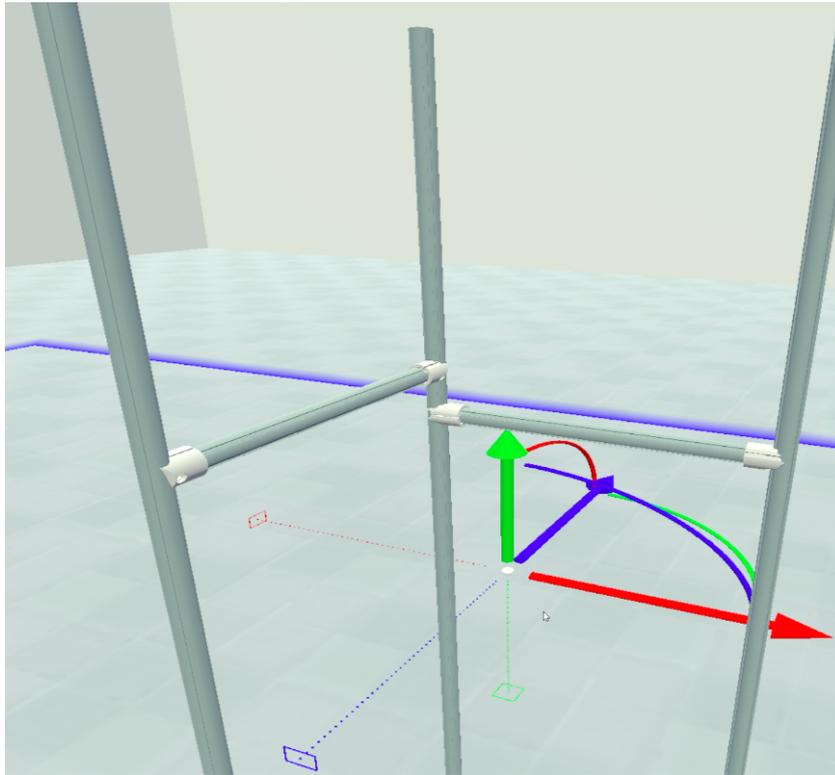


Figura 5.21: Struttura complessa risultante

## 5.4 Salvataggio e arresto del sistema

Il bottone *Shut Down* del Menu principale permette all'utente di abbandonare la simulazione virtuale e di salvare le informazioni della postazione di lavoro realizzata: quest'operazione consiste nella scrittura di un file. Il linguaggio di markup scelto per il file è XML in quanto permette di creare documenti strutturati utilizzando tag personalizzati. Ciascun tag fornisce un'informazione distinta secondo lo standard:

1	<code>&lt;tag&gt; Information &lt;/tag&gt;</code>
---	---

È possibile salvare le informazioni di tutti gli oggetti che possiedono il componente *Scriptable Object*. Le informazioni salvate sono:

- ID;
- nome dell'oggetto;
- posizione (coordinate globali  $(x,y,z)$ );

- rotazione (coordinate globali  $(x,y,z)$ );
- lunghezza del trafilato: parametro valido per gli oggetti semplici Tubo e Base Rulliera o oggetti da essi composti, per gli altri Game Objects vale -1;
- codice della struttura complessa oppure -1;



```
<Save FileName="File_01">
  <Object>
    <objectName>LCABraccio</objectName>
    <objID>0</objID>
    <objectPositionX>0,4832152</objectPositionX>
    <objectPositionY>1,134711</objectPositionY>
    <objectPositionZ>-3,616591</objectPositionZ>
    <objectRotationX>-4,834389E-08</objectRotationX>
    <objectRotationY>270</objectRotationY>
    <objectRotationZ>-4,598566E-08</objectRotationZ>
    <objectLength>0,4100001</objectLength>
    <structureCode>-1</structureCode>
  </Object>
  <Object>
    <objectName>DISCARD Giunzione</objectName>
    <objID>1</objID>
    <objectPositionX>0,4833</objectPositionX>
    <objectPositionY>0,8971107</objectPositionY>
    <objectPositionZ>-3,616098</objectPositionZ>
    <objectRotationX>5,122642E-06</objectRotationX>
    <objectRotationY>9,768346</objectRotationY>
    <objectRotationZ>90</objectRotationZ>
    <objectLength>-1</objectLength>
    <structureCode>-1</structureCode>
  </Object>
</Save>
```

**Figura 5.22:** File XML di salvataggio

L'output sarà contenuto nella cartella Streaming Assets del progetto e assumerà un nome in formato *Anno-Mese-Giorno-T Ore-Minuti-Secondi*DataXML.text per salvare la data e l'ora di realizzazione della postazione.

L'applicativo è disponibile in lingua italiana e inglese.

# Capitolo 6

## Risultati e analisi

A fine di valutare l'applicativo in termini di usabilità ed efficienza sono stati svolti dei test utente, simulando la creazione di un LCA a partire da un modello virtuale: in questo capitolo sono analizzati i risultati soggettivi e oggettivi scaturiti dai test.

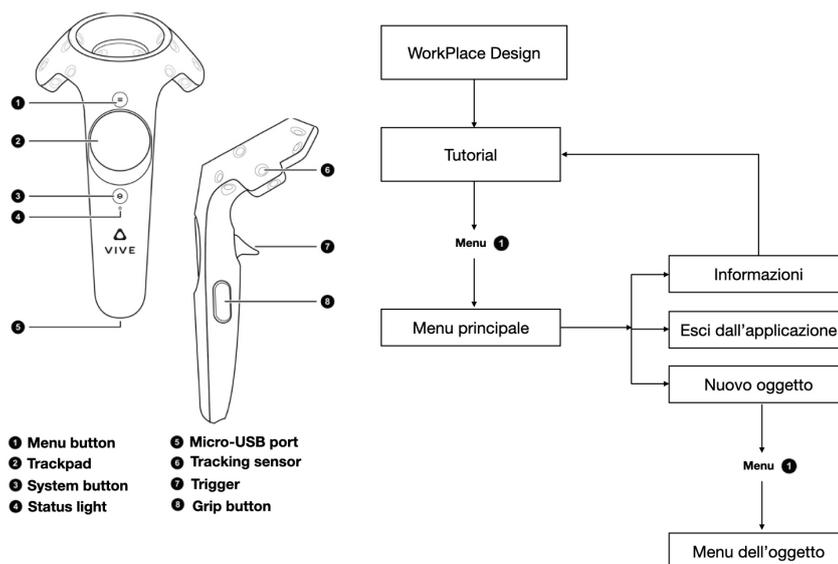


Figura 6.1: Flusso delle operazioni fondamentali per l'utilizzo dell'applicativo

## 6.1 Test

Il test utente si suddivide in tre parti: apprendimento delle funzioni e dei comandi (tutorial), assemblaggio dell'LCA, compilazione di un questionario di valutazione. Il primo step che l'utente dovrà compiere consiste nell'indossare il visore e nell'adattamento dello stesso alla testa, di conseguenza potrà esplorare l'ambiente virtuale e familiarizzare con i tasti del controller. All'utente sarà proposto un tutorial per la comprensione delle meccaniche principali: al centro della stanza virtuale saranno presenti due pannelli, il primo, a sinistra, contiene una spiegazione testuale delle azioni da svolgere, il secondo consiste in uno schermo che proietta un video dimostrativo. I pannelli descrittivi contengono quattro bottoni coi quali l'utente potrà interagire (*Back*, *Next*, *Skip Tutorial*, *Function List*) per andare avanti col tutorial, tornare allo step precedente, saltarlo o visualizzare l'elenco delle funzioni.

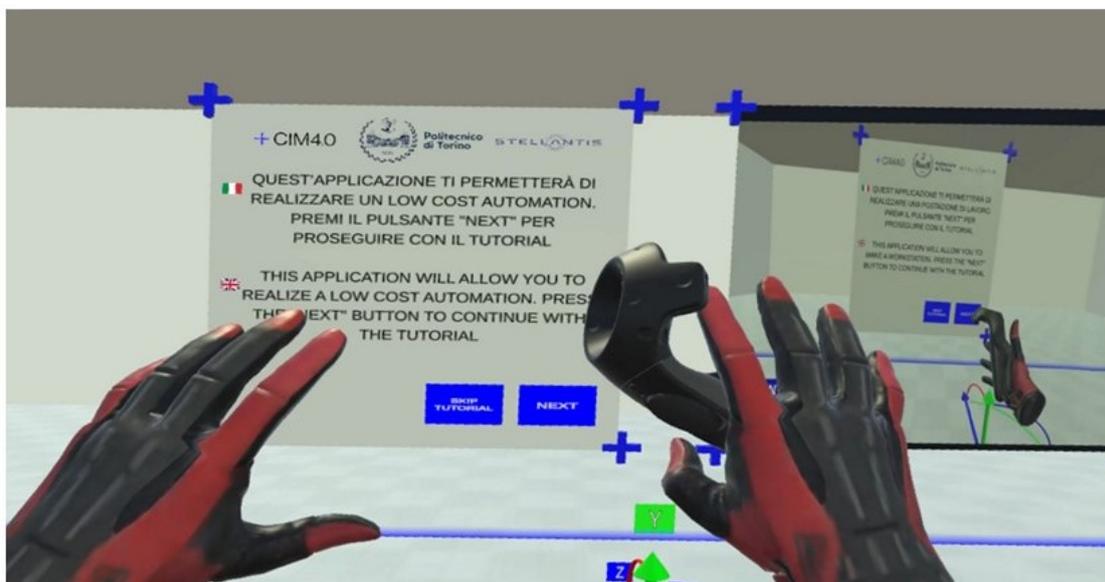


Figura 6.2: Tutorial

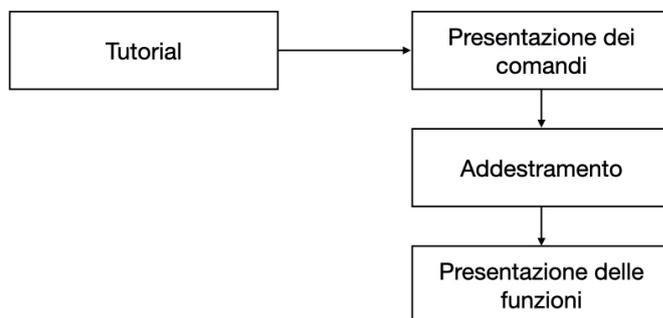
Le operazioni illustrate sono:

1. interazione con i pannelli e con i bottoni;
2. interazione con i pulsanti di HTC Vive Controller (sul pannello sono presenti le proiezioni ortogonali del controller con le etichette indicanti i pulsanti *Menu*, *Grip*, *Trigger*, *Touch*; quando l'utente preme uno di essi, l'etichetta corrispondente sarà evidenziata);



**Figura 6.3:** Tutorial: controller con etichette

3. apertura e chiusura del *Menu principale*;
4. funzione di scalamento degli oggetti;
5. funzioni generali dell'applicativo (*Distance*, *Set Axis*, *Set Dimensions*, *Split*, *Delete*, *Gravity*);
6. composizione di oggetti con attrazione magnetica;
7. realizzazione di strutture complesse;
8. manipolazione di strutture complesse;
9. movimento di componenti appartenenti alla struttura complessa;
10. separazione di componenti della struttura complessa;
11. riepilogo.



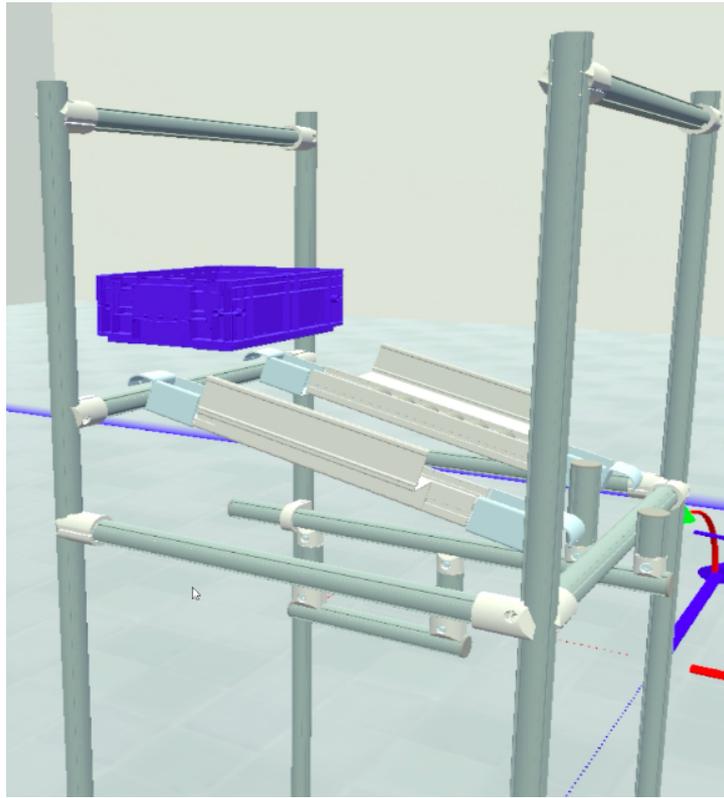
**Figura 6.4:** Struttura del tutorial

Il secondo step consiste nella realizzazione di una struttura LCA in completa autonomia da parte dell'utente che potrà sempre visualizzare la lista dei comandi e delle funzioni dell'applicativo tramite il pulsante *Info*, nel Menu principale. L'utente potrà simulare la creazione dell'LCA a partire da un modello virtuale presente nella stanza. Il Menu principale offre un pulsante *LCA* che permette di attivare e disattivare in tempo reale il modello.

Sono stati raccolti alcuni parametri al fine di effettuare una valutazione oggettiva dell'applicativo:

- tempo di vestizione e adattamento al visore HTC Vive;
- tempo di apprendimento del sistema;
- tempo effettivo per il completamento della procedura di realizzazione dell'LCA;
- numero di errori risolti in autonomia da parte del tester;
- numero di errori risolti mediante i pannelli informativi;
- tempo totale della simulazione virtuale.

La terza parte consiste nella compilazione del questionario di valutazione dell'esperienza VR e sull'usabilità del sistema, presente in Appendice A. Il questionario proposto si suddivide in quattro sezioni:



**Figura 6.5:** Esempio di struttura LCA

1. la prima parte raccoglie le caratteristiche demografiche del soggetto intervistato (età, sesso, precedenti esperienze VR con l'uso di un HMD);
2. la seconda parte consiste nel rispondere alle domande tratte dal questionario SUS [46];
3. la terza parte consiste nel rispondere alle domande tratte dal questionario TLX della NASA [47];
4. domande aggiuntive al fine di dare un'ultima valutazione soggettiva e fornire suggerimenti per migliorare l'esperienza complessiva.

Il questionario è disponibile in lingua italiana e inglese.

### 6.1.1 System Usability Survey

Il questionario **SUS** (System Usability Scale) è utilizzato durante i test per fornire una misurazione della valutazione dell'usabilità complessiva e si compone delle seguenti caratteristiche:

- **efficacia**: raggiungimento degli obiettivi da parte dell'utente;
- **efficienza**: spesa in termini sforzi e risorse per il raggiungimento degli obiettivi;
- **soddisfazione**: adeguazione piena alla misura di una richiesta.

Creato originariamente da John Brooke nel 1986, esso consiste in un questionario composto da dieci affermazioni ed è uno dei più robusti strumenti psicometrici per l'usabilità percepita. A ciascuna risposta viene assegnato un valore per il calcolo del punteggio SUS. La ripartizione dei punti per le risposte è:

- **Assolutamente in disaccordo**: 1 punto
- **Disaccordo**: 2 punti
- **Neutro**: 3 punti
- **D'accordo**: 4 punti
- **Assolutamente d'accordo**: 5 punti

È possibile tabulare il punteggio SUS complessivo utilizzando il seguente schema:

- sommare il punteggio totale per tutte le domande dispari, quindi sottrarre 5 dal totale per ottenere (X);
- sommare il punteggio totale per tutte le domande con numero pari, quindi sottrai il totale da 25 per ottenere (Y);
- sommare il punteggio totale dei nuovi valori ( $X + Y$ ) e moltiplicare per 2.5;

Il valore medio di un questionario SUS, calcolato su oltre 500 applicazioni, è di 68; i punteggi inferiori a 68 indicano problemi con il design che devono essere ricercati e risolti, mentre i punteggi superiori a 68 indicano la necessità di piccoli miglioramenti al design. Il questionario SUS risulta facile da somministrare ai partecipanti del test e affidabile anche su campioni di piccola dimensione.

### 6.1.2 NASA-Task Load Index

Il questionario **NASA-TLX** (Task Load Index) è uno strumento che permette di valutare il carico di lavoro percepito dalle azioni proposte e verificare se l'utente ha sofferto di nausea o malessere durante l'esperienza VR, è stato sviluppato dallo Human Performance Group presso l'Ames Research Center della NASA nel corso di un ciclo di sviluppo di tre anni che ha incluso più di 40 simulazioni di laboratorio. Le valutazioni effettuate, su una scala da 0 a 100 con passo di 5 unità, sono relative a:

- **richiesta mentale:** quantificazione dell'attività mentale e percettiva;
- **domanda fisica:** quantificazione dell'attività fisica;
- **esigenza temporale:** pressione temporale a causa del ritmo con cui i compiti o gli elementi del compito sono stati svolti;
- **prestazione complessiva:** svolgimento effettivo del compito e grado di soddisfazione della prestazione;
- **sforzo:** carico mentale e fisico di lavoro per raggiungere il livello di prestazione;
- **livello di frustrazione:** grado di irritazione, stress e fastidio rispetto al livello di soddisfazione.

Al termine del questionario, sono state aggiunte ulteriori domande per dare una valutazione soggettiva rispetto a:

1. ergonomia del sistema (HTC Vive e controller);
2. utilità del tutorial e dei pannelli.

Infine è presente uno spazio in cui l'utente può inserire un breve paragrafo per eventuali suggerimenti.

## 6.2 Risultati

L'applicazione è stata testata presso la digital pilot line di CIM 4.0. I test sono stati effettuati su un campione di 11 utenti di cui 4 di sesso femminile e 7 di sesso maschile, con età compresa tra i 20 e i 50 anni, selezionati per avere un'ampia varietà di competenze in ambito tecnologico e un background di tipo informatico e ingegneristico.

### 6.2.1 Risultati oggettivi

I risultati oggettivi (tabella 6.1) evidenziano che, in media, il tempo necessario per l'apprendimento dei comandi e delle funzionalità di sistema è di 8:36 minuti mentre il tempo effettivo di realizzazione dell'LCA è 14:29 minuti.

Gli utenti che hanno impiegato più tempo per l'apprendimento dei comandi hanno avuto modo di testare in maniera più dettagliata le meccaniche di sistema (manipolazione con gli oggetti e con i pannelli, funzioni). Di conseguenza, in fase di prototipazione, tali utenti si sono dimostrati più sicuri nell'utilizzo dell'applicativo e hanno commesso meno errori.

Utente	Apprendimento del sistema	Prototipazione
1	10:59	16:36
2	10:47	14:45
3	05:26	18:45
4	06:40	10:08
5	05:34	11:53
6	11:21	16:00
7	12:38	15:57
8	11:40	21:10
9	04:15	19:02
10	09:54	07:25
11	08:23	12:46
<b>MEDIA</b>	<b>08:36</b>	<b>14:29</b>

**Tabella 6.1:** Tempi di apprendimento del sistema e di prototipazione dell'LCA

### 6.2.2 Risultati soggettivi

Il risultato ottenuto dal questionario SUS (tabella 6.2) risulta maggiore di 68: è plausibile pensare che, avendo ottenuto un punteggio abbastanza alto, l'applicazione possa richiedere poche migliorie ma andrebbe verificato su un numero più ampio di tester.

Utente	Risultati pari	Risultati dispari	Totale
1	19	19	95
2	18	12	75
3	18	19	92.5
4	15	17	80
5	17	20	92.5
6	15	14	72.5
7	16	11	67.5
8	17	18	87.5
9	20	19	97.5
10	14	14	70
11	15	17	80
<b>MEDIA</b>	<b>16.73</b>	<b>16.36</b>	<b>82.73</b>

**Tabella 6.2:** Risultati SUS

Le risposte degli utenti al questionario NASA-TLX evidenziano che l'attività che hanno richiesto maggiore sforzo da parte dei tester è quella mentale.

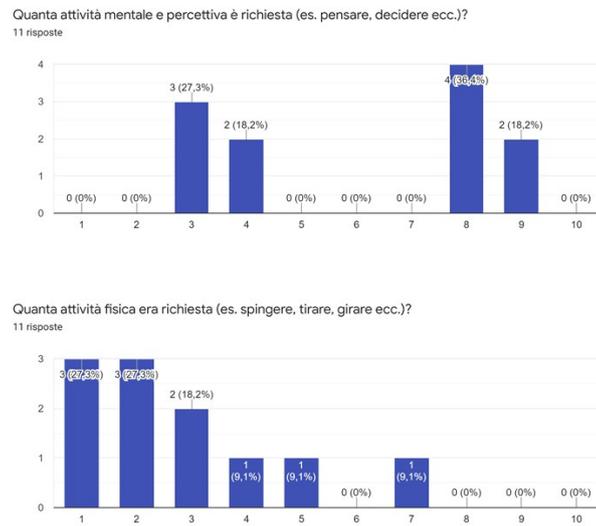
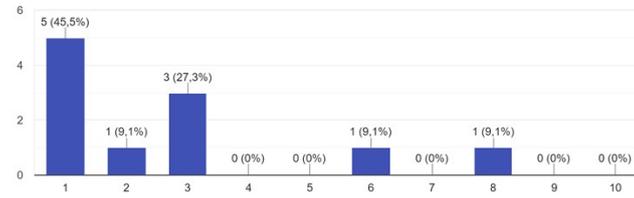


Figura 6.6: Report NASA-TLX 1

Quanta pressione temporale hai avvertito a causa della frequenza o del ritmo con cui i compiti si susseguivano?

11 risposte



Quanto hai dovuto impegnarti (mentalmente e fisicamente) per raggiungere il tuo livello di prestazione?

11 risposte

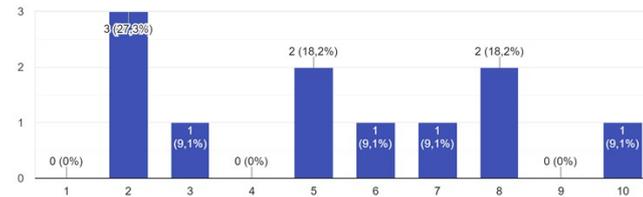
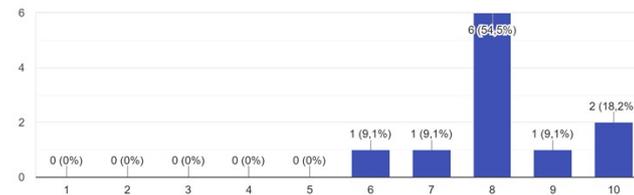


Figura 6.7: Report NASA-TLX 2

Quanto pensi di aver raggiunto gli obiettivi del compito stabiliti dallo sperimentatore?

11 risposte



Durante il compito, quanto ti sei sentito incerto, scoraggiato, irritato, stressato e infastidito rispetto a sicuro, gratificato, appagato, rilassato e soddisfatto?

11 risposte

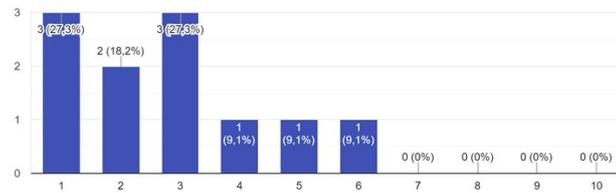
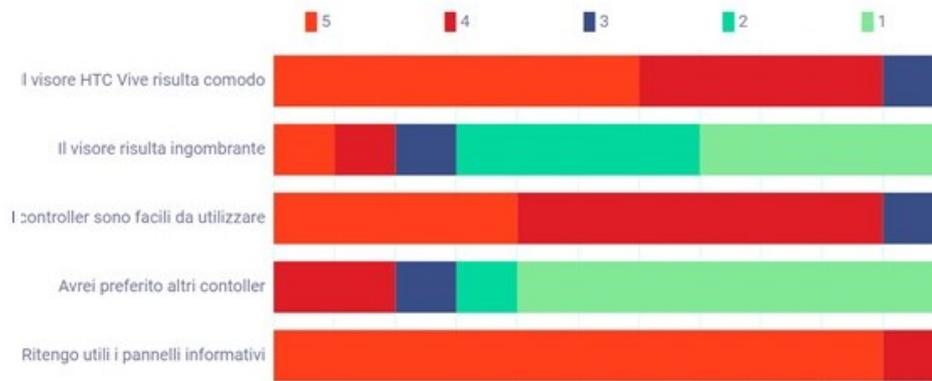


Figura 6.8: Report NASA-TLX 3

Di seguito sono riportati i risultati dettagliati delle ulteriori domande soggettive proposte al termine del test:



**Figura 6.9:** Ulteriori valutazioni soggettive: 5: Assolutamente in accordo - 1: Assolutamente in disaccordo



## Capitolo 7

# Conclusione e sviluppi futuri

Il presente studio ha cercato di fornire un contributo al tema della prototipazione industriale, integrando la letteratura già esistente relativa alla realtà virtuale ed al suo impiego in processi di formazione, assemblaggio e design: è stata svolta un'indagine bibliografica mediante l'utilizzo delle principali banche dati (IEEE Xplore [48], Google Scholar [49], ResearchGate [50]) e sono state descritte alcune soluzioni commerciali (Ikea VR Experience, R3DT, Siemens Tecnomatix).

L'elaborato vuole evidenziare le grandi potenzialità della realtà virtuale e quanto essa occupi una posizione di primaria importanza nella rivoluzione industriale 4.0. Le aziende stanno migrando verso l'utilizzo di tecnologie immersive (HMD, CAVE) per la Smart Product Design e la digitalizzazione delle linee. Si è tentato di raggiungere l'obiettivo prefissato tramite l'ideazione di un'applicazione in realtà virtuale il cui intento è quello di rendere più efficiente il processo di prototipazione grazie all'ottimizzazione delle procedure di assemblaggio e di manipolazione delle strutture, migliorandone la velocità e minimizzando il supporto da parte di un operatore più esperto, che risultasse di facile impiego anche per le piccole e medie imprese.

Il caso di studio della Low Cost Automation è risultato efficace per testare l'utilizzo e l'utilità dell'applicazione realizzata. La valutazione dei tester si è rivelata complessivamente positiva: l'applicazione appare semplice e intuitiva. Inoltre i risultati oggettivi sono soddisfacenti: nonostante gli utenti abbiano commesso qualche errore nello svolgere la procedura, questa è stata comunque portata a termine da tutti. È possibile affermare che lo strumento sviluppato risulta conforme alle specifiche descritte all'inizio dell'elaborato.

L'applicativo è strutturato in maniera modulare sia a livello di architettura che a livello di codice, per favorire la comprensione e l'aggiunta di nuove funzionalità. Al fine di ottenere di un software in grado di rispondere a esigenze progettuali differenti, il sistema potrebbe includere nuovi set di modelli 3D ed offrire nuove funzionalità. Alcune proposte per futuri avanzamenti riguardano la possibilità di

caricare a run-time le scene precedentemente salvate, di limitare i vincoli legati alle trasformazioni geometriche, garantendo migliore precisione nelle operazioni di assemblaggio ed infine l'ipotesi di realizzare postazioni di lavoro in ambienti virtuali collaborativi.

# Appendice A

## Questionario

### LCA Workplace Design

---

\*Campo obbligatorio

Select the language / Seleziona la lingua \*

- English
- Italiano

#### LCA Workplace Design ITA

Età \*

La tua risposta

---

Genere \*

- Uomo
- Donna
- Preferisco non specificarlo

Hai mai vissuto un'esperienza in Realtà Virtuale utilizzando un visore? \*

- Sì
- No

LCA Workplace Design ITA

Parte 1: Il questionario proposto permette di misurare l'esperienza dell'utente durante l'intera simulazione (SASSI)

Valutazione dell'accuratezza della risposta del sistema [1: Fortemente in disaccordo - 7: Fortemente d'accordo] \*

	1	2	3	4	5	6	7
Il sistema è accurato	<input type="radio"/>						
Il sistema è inaffidabile	<input type="radio"/>						
L'interazione col sistema è imprevedibile	<input type="radio"/>						
Il sistema non fa sempre ciò che voglio	<input type="radio"/>						
Il sistema non fa sempre ciò che mi aspetto che faccia	<input type="radio"/>						
Il sistema è affidabile	<input type="radio"/>						
Il sistema fa pochi errori	<input type="radio"/>						
L'interazione col sistema è coerente	<input type="radio"/>						
L'interazione col sistema è efficiente	<input type="radio"/>						

## Questionario

Valutazione della soddisfazione legata al sistema [1: Fortemente in disaccordo - 7: Fortemente d'accordo] \*

	1	2	3	4	5	6	7
Il sistema è utile	<input type="radio"/>						
Il sistema è piacevole	<input type="radio"/>						
Il sistema è user-friendly	<input type="radio"/>						
Sono stato in grado di recuperare facilmente se facevo errori	<input type="radio"/>						
Mi è piaciuto usare il sistema	<input type="radio"/>						
L'interazione col sistema è chiara	<input type="radio"/>						
È facile imparare a utilizzare il sistema	<input type="radio"/>						
Userei questo sistema	<input type="radio"/>						
Penso di avere il controllo dell'utilizzo del sistema	<input type="radio"/>						

Valutazione del livello di carico cognitivo legato al sistema [1: Fortemente in disaccordo - 7: Fortemente d'accordo] \*

	1	2	3	4	5	6	7
Mi sento in confidenza a utilizzare il sistema	<input type="radio"/>						
Mi sento teso a utilizzare il sistema	<input type="radio"/>						
Mi sento calmo a utilizzare il sistema	<input type="radio"/>						
Per usare il sistema è richiesto un alto livello di concentrazione	<input type="radio"/>						
Il sistema è semplice da utilizzare	<input type="radio"/>						

Valutazione del livello di abitabilità del sistema [1: Fortemente in disaccordo - 7: Fortemente d'accordo] \*

	1	2	3	4	5	6	7
A volte mi sono chiesto se stavo compiendo l'azione giusta	<input type="radio"/>						
Spesso mi sono chiesto se stavo compiendo l'azione giusta	<input type="radio"/>						
Ho sempre saputo cosa fare	<input type="radio"/>						
Non sempre ero sicuro di sapere cosa fare	<input type="radio"/>						

Valutazione di velocità di risposta del sistema [1: Fortemente in disaccordo - 7: Fortemente d'accordo] \*

	1	2	3	4	5	6	7
L'interazione col sistema è veloce	<input type="radio"/>						
La risposta del sistema è troppo lenta	<input type="radio"/>						

## Questionario

### LCA Workplace Design ITA

Parte 2: Il questionario proposto permette di fornire una misurazione della valutazione dell'usabilità complessiva (SUS)

Valutazione dell'accuratezza della risposta del sistema [1: Fortemente in disaccordo - 5: Fortemente d'accordo] \*

	1	2	3	4	5
Penso che mi piacerebbe usare questo sistema frequentemente	<input type="radio"/>				
Ho trovato il sistema inutilmente complesso	<input type="radio"/>				
Penso che il sistema sia facile da usare	<input type="radio"/>				
Penso che avrei bisogno del supporto di una persona tecnica per poter utilizzare questo sistema	<input type="radio"/>				
Ho trovato le varie funzioni di questo sistema ben integrate	<input type="radio"/>				

Ho pensato che ci fosse troppa incoerenza in questo sistema	<input type="radio"/>				
Immagino che la maggior parte delle persone imparerebbe ad usare questo sistema molto rapidamente	<input type="radio"/>				
Ho trovato il sistema molto complicato da usare	<input type="radio"/>				
Mi sono sentito molto sicuro nell'usare il sistema	<input type="radio"/>				
Ho dovuto imparare molte cose prima di poter utilizzare questo sistema	<input type="radio"/>				

### LCA Workplace Design ITA

Parte 3: Il questionario proposto permette di misurare il carico di lavoro percepito dalle azioni proposte (NASA)

Quanta attività mentale e percettiva è richiesta (es. pensare, decidere ecc.)? \*

1	2	3	4	5	6	7	8	9	10
<input type="radio"/>									

Quanta attività fisica era richiesta (es. spingere, tirare, girare ecc.)? \*

1	2	3	4	5	6	7	8	9	10
<input type="radio"/>									

Quanta pressione temporale hai avvertito a causa della frequenza o del ritmo con cui i compiti si susseguivano? \*

1	2	3	4	5	6	7	8	9	10
<input type="radio"/>									

Quanto hai dovuto impegnarti (mentalmente e fisicamente) per raggiungere il tuo livello di prestazione? \*

1	2	3	4	5	6	7	8	9	10
<input type="radio"/>									

Quanto pensi di aver raggiunto gli obiettivi del compito stabiliti dallo sperimentatore? \*

1	2	3	4	5	6	7	8	9	10
<input type="radio"/>									

Durante il compito, quanto ti sei sentito incerto, scoraggiato, irritato, stressato e infastidito rispetto a sicuro, gratificato, appagato, rilassato e soddisfatto? \*

1	2	3	4	5	6	7	8	9	10
<input type="radio"/>									

LCA Workplace Design ITA

Parte 4: Ulteriori parametri di valutazione soggettiva

Ulteriori valutazioni [1: Fortemente in disaccordo - 5: Fortemente d'accordo] \*

	1	2	3	4	5
Il visore HTC Vive risulta comodo	<input type="radio"/>				
Il visore HTC Vive risulta ingombrante	<input type="radio"/>				
I controller risultano facili da utilizzare	<input type="radio"/>				
Avrei preferito altri controller	<input type="radio"/>				
Ritengo utili i pannelli che mostrano i comandi	<input type="radio"/>				

Suggerimenti

La tua risposta

---



# Ringraziamenti

Desidero ringraziare il prof. Andrea Sanna, il dott. Federico Manuri e il dott. Sebastiano Lamacchia per avermi dato la grande opportunità di svolgere questo lavoro di tesi in azienda.

Ringrazio i miei genitori e la mia famiglia per avermi permesso di arrivare fino a questo punto, per l'affetto e il sostegno ricevuto soprattutto durante questi anni lontano da casa.

Grazie a tutte le persone incontrate durante il percorso universitario a Torino. In particolare, un ringraziamento speciale va a Beatrice, Christian, Flavio, Giuseppe e Marika per essere stati i miei principali punti di riferimento.

Ringrazio i miei amici e colleghi del CIM per avermi accompagnato durante questa ultima fase di vita accademica: Alessia, Bet, Davide, Emanuele, Giorgio, Mariachiara, Martina, Stefania, Stefano.

Infine ringrazio i miei amici "di giù" perché la distanza non ha mai rappresentato un ostacolo per il nostro solido legame.



# Bibliografia

- [1] Stuart Barnes. «Understanding Virtual Reality in Marketing: Nature, Implications and Potential». In: *SSRN Electronic Journal* (gen. 2016). DOI: 10.2139/ssrn.2909100 (cit. alle pp. 1, 16).
- [2] Nick von Tunzelmann. «Historical coevolution of governance and technology in the industrial revolutions». In: *Structural Change and Economic Dynamics* 14.4 (2003), pp. 365–384. URL: <https://EconPapers.repec.org/RePEc:eee:streco:v:14:y:2003:i:4:p:365-384> (cit. a p. 4).
- [3] M. Brettel, Niklas Friederichsen, M. Keller e N. Rosenberg. «How virtualization, decentralization and network building change the manufacturing landscape: An Industry 4.0 Perspective». In: *International Journal of Science, Engineering and Technology* 8 (ago. 2014), pp. 37–44 (cit. a p. 4).
- [4] Thuy Duong Oesterreich e Frank Teuteberg. «Understanding the implications of digitisation and automation in the context of Industry 4.0: A triangulation approach and elements of a research agenda for the construction industry». In: *Computers in Industry* 83 (2016), pp. 121–139. ISSN: 0166-3615. DOI: <https://doi.org/10.1016/j.compind.2016.09.006>. URL: <https://www.sciencedirect.com/science/article/pii/S0166361516301944> (cit. a p. 4).
- [5] W. Karwowski, M.M. Soares e N.A. Stanton. *Human Factors and Ergonomics in Consumer Product Design: Methods and Techniques*. Handbook of Human Factors in Consumer Product Design. Taylor & Francis, 2011. ISBN: 9781420046281. URL: <https://books.google.it/books?id=1CcVxu6V6M4C> (cit. a p. 6).
- [6] Sameer Gajmal e Sudhir Bhatwadekar. «Low Cost Automation(LCA): A Case Study». In: *International Journal of Advanced Engineering and Global Technology* Vol-2 (dic. 2014), pp. 172–181 (cit. a p. 6).
- [7] *Karakuri Kaizen*. <http://his.diva-portal.org/smash/get/diva2:1331843/FULLTEXT01.pdf> (cit. a p. 6).
- [8] *CIM4.0*. <https://cim40.com/> (cit. a p. 8).

- 
- [9] *Stellantis*. <https://www.stellantis.com/en> (cit. a p. 8).
- [10] S.G. Weinbaum. *Pygmalion's Spectacles*. Floating Press, 2012. ISBN: 9781775562986| URL: <https://books.google.it/books?id=NzNZaZ19ENQC> (cit. a p. 9).
- [11] Morton Leonard Heilig. «EL Cine del Futuro: The Cinema of the Future». In: *Presence: Teleoperators & Virtual Environments 1* (1992), pp. 279–294 (cit. a p. 10).
- [12] I.E. Sutherland. *Sketchpad: A Man-machine Graphical Communication System*. Outstanding dissertations in the computer sciences. Garland Pub., 1980. ISBN: 9780824044114. URL: <https://books.google.it/books?id=pb87LQAACAAJ> (cit. a p. 11).
- [13] Ivan E. Sutherland. «The Ultimate Display». In: 1965 (cit. a p. 11).
- [14] Ivan E. Sutherland. «A Head-Mounted Three Dimensional Display». In: *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*. AFIPS '68 (Fall, part I). San Francisco, California: Association for Computing Machinery, 1968, pp. 757–764. ISBN: 9781450378994. DOI: 10.1145/1476589.1476686. URL: <https://doi.org/10.1145/1476589.1476686> (cit. a p. 11).
- [15] *Massachusetts Institute of Technology*. <https://www.mit.edu> (cit. a p. 11).
- [16] *Visual Programming Lab Research Inc*. <https://www.vrs.org.uk/virtual-reality-profiles/vpl-research.html> (cit. a p. 12).
- [17] Thomas DeFanti et al. «The future of the CAVE». In: *Open Engineering 1* (mar. 2010), pp. 16–37. DOI: 10.2478/s13531-010-0002-5 (cit. a p. 12).
- [18] *Oculus Rift*. <https://www.oculus.com/rift/> (cit. a p. 13).
- [19] *Google Cardboard*. <https://vr.google.com/cardboard/> (cit. a p. 13).
- [20] *Sony*. <https://www.sony.it> (cit. a p. 13).
- [21] *PlayStation 4*. <https://www.playstation.com/it-it/explore/ps4/> (cit. a p. 13).
- [22] *PlayStation VR*. <https://www.playstation.com/it-it/explore/playstation-vr/> (cit. a p. 13).
- [23] *HTC Vive*. <https://www.vive.com/eu/product/> (cit. a p. 14).
- [24] *Valve*. <http://www.valvesoftware.com> (cit. a p. 14).
- [25] *HTC*. <http://www.htc.com/it/> (cit. a p. 14).
- [26] J. Fenn e M. Raskino. *Mastering the Hype Cycle: How to Choose the Right Innovation at the Right Time*. Gartner, Inc./Harvard Business School Press. Harvard Business Review Press, 2008. ISBN: 9781422135129. URL: <https://books.google.it/books?id=ApeSoRHCCGwC> (cit. a p. 14).

- [27] J. Herrman e K. Browning. *Are We in the Metaverse Yet?*, *The New York Times*. <https://www.nytimes.com/2021/07/10/style/metaverse-virtual-worlds.html>. 2021 (cit. a p. 15).
- [28] Brian Caulfield. *What Is the Metaverse?*, *The Official NVIDIA Blog*. <https://blogs.nvidia.com/blog/2021/08/10/what-is-the-metaverse/>. 2021 (cit. a p. 15).
- [29] Paul Milgram e Fumio Kishino. «A Taxonomy of Mixed Reality Visual Displays». In: *IEICE Trans. Information Systems* vol. E77-D, no. 12 (dic. 1994), pp. 1321–1329 (cit. a p. 16).
- [30] *Microsoft HoloLens*. <https://www.microsoft.com/it-it/hololens> (cit. a p. 17).
- [31] A.C. Boud, D.J. Haniff, C. Baber e S.J. Steiner. «Virtual reality and augmented reality as a training tool for assembly tasks». In: *1999 IEEE International Conference on Information Visualization (Cat. No. PR00210)*. 1999, pp. 32–36. DOI: 10.1109/IV.1999.781532 (cit. a p. 29).
- [32] Abhishek Seth, Judy Vance e James Oliver. «Virtual reality for assembly methods prototyping: A review». In: *Virtual Reality* 15 (mar. 2011), pp. 5–20. DOI: 10.1007/s10055-009-0153-y (cit. a p. 30).
- [33] Ioannis Fudos e Christoph Hoffmann. «A Graph-Constructive Approach to Solving Systems of Geometric Constraints». In: *ACM Transactions on Graphics* 16 (mar. 2004). DOI: 10.1145/248210.248223 (cit. a p. 31).
- [34] Mustufa Abidi, El-Tamimi A.M., Abdulrahman Al-Ahmari e Emad Abouel Nasr. «Assessment and comparison of immersive virtual assembly training system». In: *International Journal of Rapid Manufacturing* 3 (gen. 2013), pp. 266–283. DOI: 10.1504/IJRAPIDM.2013.055973 (cit. a p. 33).
- [35] Amaury Peniche, Christian Diaz, Gabriel Paramo e Helmuth Trefftz. «Combining Virtual and Augmented Reality to Improve the Mechanical Assembly Training Process in Manufacturing». In: gen. 2012 (cit. a p. 34).
- [36] *IKEA VR Experience*. [https://store.steampowered.com/app/447270/IKEA\\_VR\\_Experience/](https://store.steampowered.com/app/447270/IKEA_VR_Experience/) (cit. a p. 36).
- [37] Lusetti M. *Progettazione automatica di ambienti di lavoro per la prototipazione rapida*. [https://morethesis.unimore.it/theses/available/etd-03102021-233049/unrestricted/Tesi\\_Mattia\\_Lusetti\\_.pdf](https://morethesis.unimore.it/theses/available/etd-03102021-233049/unrestricted/Tesi_Mattia_Lusetti_.pdf) (cit. a p. 37).
- [38] Francesco Caputo, Di Gironimo e Stefano Papa. «REALIZZAZIONE DI UN AMBIENTE VIRTUALE PER LA VALUTAZIONE DELL'ERGONOMIA E DELL'USABILITÀ DI ORGANI DI COMANDO DI ...» In: (feb. 2022) (cit. a p. 38).

- 
- [39] Matthias Schindler e Alexander W. Verl. «A Novel Method for Agile Planning Production Systems based on 3D Digitalization». In: *2018 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)* (2018), pp. 1–6 (cit. a p. 39).
- [40] *R3DT*. <https://r3dt.com/> (cit. a p. 40).
- [41] *Leap Motion*. <https://www.ultraleap.com/product/leap-motion-controller/> (cit. a p. 40).
- [42] *Siemens VR*. <https://www.plm.automation.siemens.com/global/it/products/collaboration/virtual-reality.html> (cit. a p. 41).
- [43] *Unity*. <https://unity.com> (cit. a p. 48).
- [44] *Microsoft Visual Studio*. <https://visualstudio.microsoft.com/it/> (cit. a p. 53).
- [45] *Asset Store*. <https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647> (cit. a p. 53).
- [46] James R. Lewis. «Measuring Perceived Usability: The CSUQ, SUS, and UMUX». In: *International Journal of Human–Computer Interaction* 34.12 (2018), pp. 1148–1156. DOI: 10.1080/10447318.2017.1418805. eprint: <https://doi.org/10.1080/10447318.2017.1418805>. URL: <https://doi.org/10.1080/10447318.2017.1418805> (cit. a p. 81).
- [47] Sandra G. Hart e Lowell E. Staveland. «Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research». In: *Human Mental Workload*. A cura di Peter A. Hancock e Najmedin Meshkati. Vol. 52. Advances in Psychology. North-Holland, 1988, pp. 139–183. DOI: [https://doi.org/10.1016/S0166-4115\(08\)62386-9](https://doi.org/10.1016/S0166-4115(08)62386-9). URL: <https://www.sciencedirect.com/science/article/pii/S0166411508623869> (cit. a p. 81).
- [48] *IEEE Xplore*. <https://ieeexplore.ieee.org/Xplore/home.jsp> (cit. a p. 89).
- [49] *Google Scholar*. <https://scholar.google.com/> (cit. a p. 89).
- [50] *ResearchGate*. <https://www.researchgate.net/> (cit. a p. 89).