

# POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria Informatica



Tesi di laurea Magistrale

## La Realtà Aumentata come strumento di formazione industriale: il caso applicativo dell'automotive

Relatore

Prof. Andrea SANNA

Candidato

Alessia MONTAGNA

Aprile, 2022



# Abstract

Con l'avvento dell'Industria 4.0 le industrie tradizionali si stanno evolvendo in fabbriche intelligenti grazie all'utilizzo delle nuove tecnologie. Tra queste, la Realtà Aumentata ha un ruolo chiave grazie alla sua capacità di integrare le informazioni che l'utente percepisce dal mondo reale con elementi virtuali coerentemente posizionati rispetto alle controparti reali. La Realtà Aumentata, applicata in un contesto industriale, è in grado di supportare gli operatori nelle loro postazioni di lavoro, in tempo reale e durante operazioni manuali, offrendo numerosi vantaggi in termini di risparmio di tempo, riduzione degli errori e aumento della qualità del prodotto finale.

La visualizzazione delle informazioni digitali sovrapposte a un modello reale permette infatti agli operatori di accedere alle informazioni più importanti in maniera diretta, veloce ed efficace. Nello specifico, l'utilizzo dell'AR come strumento tecnologico nelle industrie ha diversi casi applicativi: può essere di supporto ad attività di formazione, permette la visualizzazione e l'analisi di dati, può essere impiegata in attività di assistenza da remoto ed operazioni di manutenzione complesse.

Questo progetto di tesi è stato realizzato presso il centro di ricerca e sviluppo CIM 4.0 di Torino ed in collaborazione con Bylogix, azienda che si occupa di fornire servizi e soluzioni di ingegneria elettrica ed elettronica per l'industria automobilistica. L'attività di tesi ha portato allo sviluppo di una soluzione AR, fruibile attraverso il dispositivo Microsoft HoloLens 2, per attività di formazione in ambiente automotive, con l'obiettivo di semplificare e velocizzare questo tipo di mansione. Il caso d'uso riguarda una procedura per scollegare i sistemi ad alta tensione di un'automobile elettrica per rendere sicuro il veicolo prima di una qualsiasi procedura di riparazione o manutenzione. La procedura viene illustrata all'utente sottoforma di istruzioni testuali suddivise in diversi step, ognuno dei quali può contenere testo, immagini o video, ed eventualmente dei pannelli di avviso che contengono informazioni per la sicurezza dell'utente. Alcuni step prevedono l'inserimento mediante tastiera virtuale di valori misurati da un operatore, che verranno collezionati in un file di report accessibile dal supervisore. Grazie a un sistema di tracking del veicolo, che riesce a riconoscere l'oggetto reale e a dedurre

la sua posizione nell'ambiente, l'utente visualizzerà l'istruzione nei pressi del punto in cui deve avvenire l'azione e verrà guidato verso quella posizione grazie a una curva che lo congiunge al punto di interesse. Oltre all'applicazione per Hololens è stata sviluppata un'applicazione desktop, che permette ad un supervisore di modificare la procedura visualizzata in Realtà Aumentata in maniera semplice ed efficace. L'Hololens può ricevere i dati aggiornati da un server su cui sono conservate tutte le procedure e i relativi file, così da essere sempre disponibili agli operatori. L'applicazione desktop permette anche di visualizzare lo storico dei dati di report inseriti dagli utenti durante l'uso dell'applicazione Hololens. Per valutare l'efficacia e l'efficienza del sistema proposto sono stati svolti dei test utente con l'obiettivo di simulare la procedura di manutenzione, confrontando l'utilizzo dell'Hololens 2 con la manutenzione tradizionale basata su istruzioni cartacee. Sono stati raccolti sia dati oggettivi che dati soggettivi tramite la compilazione di questionari e dall'analisi dei risultati è stato possibile valutare l'impatto del sistema proposto.



# Ringraziamenti

Vorrei ringraziare in primo luogo chi mi ha seguito durante questi mesi di tesi: il mio tutor aziendale Orlando Tovar, il mio relatore Andrea Sanna, ma anche Federico Manuri e Sebastiano Lamacchia per la pazienza e la gentilezza con cui mi hanno aiutata e seguita.

Ringrazio il CIM 4.0 e Bylogix per avermi fornito tutta la strumentazione per poter sviluppare questo progetto di tesi.

Grazie anche alla mia famiglia, in particolare i miei genitori, per avermi sempre sostenuta e supportata in ogni mia scelta e per avermi permesso di concludere con serenità il mio percorso universitario. Senza il loro aiuto non sarei mai riuscita ad arrivare a questo punto.

Ringrazio il mio ragazzo Antonio per avermi sopportata e supportata in questi 5 anni, per avermi sempre sostenuta e spronata in ogni mia scelta e per aver sempre creduto in me anche quando io non ci credevo. Se oggi ho acquisito un buon livello di confidenza lo devo principalmente a lui.

Ringrazio Sofia, Nadia, Anna, Paola, Stella, amiche storiche con cui condivido un profondo legame da più di 10 anni per i momenti di spensieratezza che mi hanno regalato durante tutti questi anni, per avermi sempre ascoltata e per aver condiviso con me una grande parentesi della mia vita e per far parte di quelle persone preziose e rare da trovare.

Ringrazio Alice e Martina, amiche da più di 15 anni, perchè sono sempre state un mio punto di riferimento. Nonostante il fatto che abbiamo preso tutte e tre strade diverse il legame originale è rimasto intatto.

Ringrazio Federica, l'amica che conosco letteralmente da quando sono nata e con cui ho sempre avuto un rapporto leale e saldo e che ogni anno si fortifica sempre di più.

Ringrazio Greta, Giorgio e Christian, i miei ex colleghi universitari, con cui ho condiviso le gioie e i dolori dei progetti di gruppo e con cui si è creata una solida amicizia duratura. Senza il loro aiuto non avrei affrontato la triennale con la stessa serenità.

Ringrazio Edoardo e Manfredi per la loro simpatia e per i momenti di spensieratezza che mi hanno regalato.

Ringrazio infine Martina, Stefano, Emanuele, Davide e Alfonso, i miei ex colleghi al CIM con cui in pochissimi mesi si è creato un rapporto solido e sincero e che spero di mantenere a lungo.

Grazie infine anche a Nicola, Giorgio, Stefania, Mariachiara e Betheliem del CIM per aver condiviso insieme a me gioie e dolori durante lo svolgimento della tesi e per aver reso più spensierata e allegra questa esperienza, per avermi fatto ridere e avermi supportata nell'arco di questi mesi, specialmente quelli conclusivi.



# Indice

<b>Elenco delle tabelle</b>	IX
<b>Elenco delle figure</b>	X
<b>Acronimi</b>	XIV
<b>1 Introduzione</b>	1
1.1 Motivazioni e obiettivi . . . . .	3
1.2 Organizzazione dei capitoli . . . . .	4
<b>2 Concetti Generali</b>	6
2.1 Origini della Realtà Aumentata . . . . .	6
2.2 Architettura della Realtà Aumentata . . . . .	9
2.2.1 Display AR . . . . .	10
2.2.2 Registration, Tracking, Calibration . . . . .	11
<b>3 Stato dell'arte</b>	14
3.1 Industria 4.0 . . . . .	14
3.1.1 CIM 4.0 . . . . .	17
3.2 Utilizzo della Realtà Aumentata in ambito Automotive . . . . .	19
3.2.1 Applicazioni commerciali . . . . .	19
3.2.2 Applicazioni industriali . . . . .	20
<b>4 Tecnologie e strumenti</b>	30
4.1 Microsoft Hololens 2 . . . . .	30
4.2 Unity . . . . .	32
4.2.1 Mixed Reality Toolkit . . . . .	33
4.2.2 Vuforia Engine . . . . .	35
4.3 Visual Studio . . . . .	37
4.4 Blender . . . . .	38

<b>5</b>	<b>Design e sviluppo</b>	<b>42</b>
5.1	Descrizione dell'applicativo . . . . .	42
5.2	Architettura . . . . .	45
5.2.1	Comunicazione e sincronizzazione delle app . . . . .	47
5.2.2	Scambio di dati tra server e client . . . . .	49
5.3	Applicazione desktop . . . . .	49
5.3.1	Interfaccia e interazioni . . . . .	49
5.3.2	Struttura interna . . . . .	52
5.4	Applicazione per Hololens . . . . .	57
5.4.1	Interfaccia . . . . .	57
5.4.2	Modelli di interazione . . . . .	63
5.4.3	Struttura del progetto . . . . .	66
<b>6</b>	<b>Test e Risultati</b>	<b>73</b>
6.1	Test comparativi . . . . .	73
6.1.1	Test oggettivi . . . . .	74
6.1.2	Test soggettivi . . . . .	74
6.2	Test utente . . . . .	77
6.2.1	Valutazione del design . . . . .	77
6.2.2	Valutazione dell'ergonomia del dispositivo . . . . .	81
<b>7</b>	<b>Conclusioni e sviluppi futuri</b>	<b>82</b>
	<b>Bibliografia</b>	<b>84</b>

# Elenco delle tabelle

6.1	Tabella dei tempi di ogni utente . . . . .	74
-----	--	----

# Elenco delle figure

2.1	The Sword of Damocles . . . . .	7
2.2	Diagramma VR Continuum . . . . .	7
3.1	Tecnologie dell'Industria 4.0 . . . . .	15
3.2	Digital Pilot Line . . . . .	18
3.3	Comau . . . . .	18
3.4	Hyundai Virtual Guide . . . . .	20
3.5	Applicazione della Ferrari . . . . .	20
3.6	Dynamics 356 Remote Assistant . . . . .	24
3.7	Common Augmented Reality Platform di Bosch . . . . .	25
3.8	Esempio delle funzionalità di Varjo con Volvo . . . . .	26
3.9	Analisi CFD del veicolo in real-time . . . . .	26
3.10	Funzionalità e meccanismi del veicolo Toyota . . . . .	27
3.11	Individuazione di errori tramite machine learning . . . . .	27
3.12	Remote Assistant con Faurecia . . . . .	28
3.13	Esempio applicazione AR di Lockheed Martin Comporation . . . . .	29
4.1	Hardware Hololens 2 . . . . .	31
4.2	Finestra Ispezione di Unity . . . . .	33
4.3	Finestra Gerarchia di Unity . . . . .	33
4.4	Mixed Reality Feature Tool . . . . .	35
4.5	Output del Model Target Generator . . . . .	37
4.6	Finestra di Gerarchia del Model Target . . . . .	37
4.7	Interfaccia di Blender . . . . .	38
4.8	Tesla prima della semplificazione . . . . .	39
4.9	Tesla dopo la semplificazione . . . . .	39
4.10	Telaio prima della semplificazione . . . . .	39
4.11	Telaio dopo la semplificazione . . . . .	39
5.1	Posizione delle componenti elettriche . . . . .	42
5.2	Posizione delle componenti elettriche in AR . . . . .	43

5.3	Curva spline di giunzione tra utente e pannello dello step . . . . .	44
5.4	Indicatore . . . . .	44
5.5	Pannello di Warning . . . . .	44
5.6	Utilizzo della tastiera AR nell'applicazione . . . . .	45
5.7	Architettura del sistema . . . . .	46
5.8	Organizzazione del file JSON . . . . .	47
5.9	Sezione Report dell'app Desktop . . . . .	48
5.10	Interfaccia dell'applicazione desktop . . . . .	50
5.11	Interfaccia dei warning . . . . .	52
5.12	Interfaccia per selezionare i punti dell'auto . . . . .	52
5.13	Struttura in Unity della gerarchia . . . . .	53
5.14	Prefab BluPanel . . . . .	55
5.15	Gerarchia del prefab . . . . .	55
5.16	Prefab Button . . . . .	56
5.17	Interfaccia di avvio in assenza di risposta dal server . . . . .	57
5.18	Interfaccia di avvio . . . . .	58
5.19	Esempio di un pannello di tutorial . . . . .	58
5.20	Pannello di avvio applicazione . . . . .	59
5.21	Interfaccia del pannello di istruzione. . . . .	60
5.22	Interfaccia di Warning . . . . .	61
5.23	Visualizzazione dei warning dopo l'Ok . . . . .	61
5.24	Interfaccia dell'inserimento di valori di input da tastiera . . . . .	61
5.25	Interfaccia per caricare il file sul server . . . . .	62
5.26	Interfaccia finale . . . . .	62
5.27	Diagramma di flusso dell'applicazione . . . . .	63
5.28	Puntatore Hand Gaze & Dwell . . . . .	65
5.29	Interfaccia suggerimento comando vocale . . . . .	65
5.30	Interazione con sguardo + gesture . . . . .	65
5.31	Hand Menu . . . . .	65
5.32	Interazione con tastiera AR . . . . .	66
5.33	Struttura gerarchica scena 0 . . . . .	66
5.34	Struttura gerarchica scena 1 . . . . .	67
5.35	Struttura gerarchica scena 2 . . . . .	68
5.36	Visualizzazione del Model Target della Tesla . . . . .	69
5.37	Posizionamento del modello 3D dell'auto sull'auto reale . . . . .	70
5.38	Prefab del pannello con tastiera . . . . .	71
5.39	Prefab del pannello di Warning . . . . .	72
5.40	Pannello di istruzioni con il warning . . . . .	72
6.1	Report questionario NASA procedura cartacea . . . . .	76
6.2	Report questionario NASA procedura Hololens . . . . .	76



6.3	Report questionario procedura cartacea . . . . .	77
6.4	Report questionario procedura Hololens . . . . .	77
6.5	Istogramma di valutazione sul design del sistema . . . . .	78
6.6	Istogramma di valutazione sul tutorial . . . . .	79
6.7	Istogramma di valutazione sulla modalità di visualizzazione dei Warning . . . . .	79
6.8	Istogramma di valutazione sulla visualizzazione del telaio . . . . .	80
6.9	Istogramma di valutazione sull'utilità dell'utilizzo di Hololens . . . .	80
6.10	Istogramma di valutazione sull'ergonomia dell'utilizzo di Hololens .	81



# Acronimi

**HMD**

Head Mounted Display

**AR**

Augmented Reality

**VR**

Virtual Reality

**SDK**

Software Development Toolkit

**XR**

Extended Reality

**FOV**

Field Of View

**IoT**

Internet of Things

**API**

Application Programming Interface

**MRTK**

Mixed-Reality Toolkit

**HMI**

Human - Machine Interaction

**CPU**

Central Processing Unit

**IT**

Information Technology

# Capitolo 1

## Introduzione

La Realtà Aumentata è una tecnologia che permette di visualizzare oggetti digitali sovrapposti al mondo reale. In questo modo l'utente non è completamente immerso in un mondo virtuale (come accade nella Virtual Reality), ma è in grado di visualizzare anche l'ambiente reale circostante. Le applicazioni della Realtà Aumentata sono numerose, e spaziano dal campo medico, al campo dell'entertainment, all'ambito artistico/culturale, all'ambito industriale. È in quest'ultimo che si è soffermato il lavoro di ricerca per questa tesi. I casi d'uso principali in ambito industriale sono:

- **Educazione e formazione.** I tecnici vengono allenati per affrontare scenari complessi o attività pericolose che potrebbero causare lesioni gravi se gestite in modo improprio da un principiante.
- **Visualizzazione e analisi di dati.** I dati vengono presentati e visualizzati in maniera coinvolgente, possono essere interattivi. In questo modo l'utente ha una completa libertà di movimento per visualizzare o manipolare i dati 3D.
- **Assistenza e collaborazione remota.** Quando c'è bisogno di supporto tecnico, ma l'operatore si trova in un altro luogo geografico, la Realtà Aumentata può essere d'aiuto perché dà la possibilità al tecnico di fornire istruzioni all'operaio come se fosse fisicamente presente nello stesso spazio fisico. Il tecnico può infatti accedere alla camera dell'operaio e può guidarlo nel risolvere i diversi task aggiungendo elementi tridimensionali (come frecce che indicano il punto di interesse) alla scena dell'operaio. Questa feature permette di risparmiare tempo e costi.
- **Riparazioni complesse e lunghe ispezioni.** I casi che richiedono lunghe ispezioni di routine e riparazioni complesse possono avvalersi dell'uso delle tecnologie di Realtà Aumentata per aumentare la produttività ed evitare gli errori: ad esempio, gli utenti possono visualizzare le componenti nascoste di

un prodotto direttamente su di esso, oppure possono visualizzare la cronologia degli errori diagnosticati sul macchinario precedentemente in modo tale da esserne a conoscenza.

- **Coinvolgimento del cliente.** La Realtà Aumentata può essere un ottimo strumento per aumentare la soddisfazione del cliente in un'esperienza di acquisto. Ad esempio, il cliente ha la possibilità di personalizzare il design della propria automobile e visualizzare immediatamente il risultato finale prima di inviarla alla linea di produzione.

Con l'avvento dell'Industria 4.0, le industrie tradizionali si stanno evolvendo in fabbriche intelligenti grazie all'utilizzo delle nuove tecnologie: Big data, Cloud Computing, Cybersecurity, Robot autonomi, ambienti di simulazione per citarne alcune. Tra queste si trova anche la Realtà Aumentata, che offre numerosi vantaggi sia in termini di risparmio di tempo, sia in termini di riduzione degli errori, sia in termini di sicurezza. Nello specifico, alcuni studi[1] dimostrano che il supporto dell'AR per le attività di assemblaggio permette di decrementare del 33% i tempi di completamento dei diversi task e di avere una First Time Quality del prodotto di otto volte superiore rispetto a una soluzione più tradizionale basata su desktop o tablet senza supporto AR. Altri studi [2], invece, sottolineano quanto l'utilizzo di occhiali AR possa essere utile a ridurre il carico di lavoro cognitivo, migliorando l'efficienza lavorativa e la qualità del lavoro. I vantaggi offerti dall'utilizzo di questa tecnologia sono soprattutto dovuti al fatto che la visualizzazione diretta, immediata ed efficiente delle informazioni necessarie permette di velocizzare i processi produttivi. Negli ultimi anni l'industria automobilistica inizia a utilizzare la Realtà Aumentata per diversi casi applicativi, tra cui:

- Design, prototipazione visiva e validazione. Durante la produzione di un veicolo il mancato rilevamento di errori può aumentarne i costi e i ritardi. La Realtà Aumentata permette di visualizzare i componenti di un'automobile senza che questi siano fisicamente presenti, in modo tale da permettere a progettisti e ingegneri di testare virtualmente il design delle componenti, riducendo drasticamente il numero di prototipi sbagliati nella fase di produzione fisica. In questo modo si riducono costi e tempi di sviluppo, e parallelamente aumentano produttività, efficienza e qualità dei nuovi prodotti.
- Simulare proprietà e fenomeni fisici a cui sono sottoposti i veicoli. Ad esempio il processo per l'analisi della dinamica dei fluidi computazionali viene migliorato permettendo di acquisire e visualizzare questa analisi su un veicolo in tempo reale per semplificare il processo di revisione. L'utente girando intorno a un veicolo fermo dotato di un dispositivo AR può simulare e analizzare come il suo design può influire sull'aerodinamica.

- Formazione interattiva del personale per migliorarne qualità ed efficienza. L'utilizzo della Realtà Aumentata per il processo di formazione ha un notevole impatto sulla qualità di apprendimento di un processo di assemblaggio, ad esempio.

Parlando nello specifico di quest'ultimo punto, bisogna tenere a mente che è molto importante formare il personale tramite istruzioni e step specifici che garantiscano anche la sicurezza e il corretto utilizzo delle macchine, al fine di ottenere migliori performance nel processo di produzione. Questa necessità ha portato le industrie a creare ambienti che simulino eventi specifici, dato che una formazione inadeguata e un'esperienza lavorativa insufficiente sono tra le principali cause di infortuni sul lavoro che riguarda l'uso di macchinari e attrezzature moderne, e che potrebbero essere evitati se gli operatori sono adeguatamente formati nell'area di lavoro. A questo proposito, l'AR può migliorare notevolmente l'apprendimento e la comprensione del personale sia esperto sia inesperto, perchè aumenta il coinvolgimento dell'utente e riduce i costi di formazione e la curva di apprendimento. L'utilizzo dell'AR infatti permette all'utente di avere un apprendimento **pratico e interattivo**, potendo esercitare le attività lavorative in un ambiente simulato senza pericolo di avere problemi di sicurezza. Inoltre, i costi di formazione vengono abbattuti perchè l'attrezzatura utilizzata per la formazione può essere riutilizzata.

## 1.1 Motivazioni e obiettivi

Alla luce degli studi precedentemente citati trova origine questa tesi e il relativo progetto, il cui scopo è ricercare e sviluppare una soluzione di Realtà Aumentata, fruita attraverso l'Head Mounted Display di Microsoft, le Hololens 2, per il processo di formazione industriale nel settore automotive.

Il caso d'uso specifico riguarda una procedura da seguire per scollegare i sistemi ad alta tensione di un'automobile elettrica, una Tesla Model 3, al fine di rendere sicuro il veicolo prima di qualsiasi riparazione. La procedura originale si divide in 18 step, e prevede che l'operatore legga le istruzioni su un manuale e segua quanto scritto. Nella soluzione che prevede l'utilizzo di Hololens 2 il sistema è in grado di riconoscere la posizione nello spazio dell'automobile grazie al riconoscimento dell'oggetto. Una volta determinato il posizionamento dell'oggetto di interesse nel mondo reale, verrà visualizzato un pannello nel punto in cui si dovrà svolgere l'operazione da fare (l'utente sarà aiutato anche da una freccia che indica il punto esatto). All'interno del pannello è indicato:

- Numero dello step
- Istruzione testuale da eseguire in quel punto

- Immagini o video
- Eventuali avvertenze a cui prestare attenzione per la sicurezza personale

Una volta completata l'operazione scritta nel panel, l'utente potrà accedere all'istruzione successiva sia mediante comandi vocali sia indicando lo sguardo verso il pulsante con la scritta *Next*. Se si rende conto che deve rivedere l'istruzione precedente, si potrà interagire con il pulsante *Back*. Il fruitore dell'esperienza è guidato anche da una curva spline che collega la sua posizione alla posizione in cui deve effettuare l'operazione.

L'obiettivo consiste nel registrare una minore tempistica e una migliore qualità nel completamento dei diversi task nella soluzione AR rispetto alla soluzione cartacea. A tal proposito verrà confrontata la soluzione che prevede l'utilizzo di Hololens 2 con una soluzione che prevede l'utilizzo di istruzioni cartacee, per poter valutare l'impatto che ha l'AR in termini di qualità e tempo.

Per rendere l'applicazione modulare, è stata anche sviluppata un'applicazione Desktop che permetta di modificare testo, immagini/video e posizione dei pannelli con le istruzioni visualizzati nelle Hololens. L'interfaccia Desktop mira a essere molto intuitiva per garantire una User Experience ottimale, prendendo spunto dall'interfaccia di un normale Power Point. L'applicazione AR verrà aggiornata automaticamente tramite una connessione a internet quando ci saranno modifiche inserite da desktop. Questa la rende estremamente versatile anche per poter costruire nuove procedure senza necessità di sviluppare un'altra applicazione, e senza fare un'altra distribuzione.

## 1.2 Organizzazione dei capitoli

Nel capitolo 2 verrà affrontata una panoramica completa circa i concetti generali della Realtà Aumentata, a partire da una piccola introduzione sulla sua origine storica per arrivare poi alla descrizione della sua architettura. Nel capitolo 3 verrà presentato lo stato dell'arte attuale sull'uso dell'AR, a partire dall'avvento dell'Industria 4.0 e del suo ruolo nello sviluppo di applicazioni di Realtà Aumentata in campo industriale (con una parentesi sul CIM 4.0, centro di ricerca e sviluppo che mi ha fornito la strumentazione adeguata per poter lavorare allo sviluppo dell'applicativo) entrando poi nello specifico sull'utilizzo dell'AR nelle applicazioni commerciali e industriali per quello che riguarda il settore automotive. Nel capitolo 4 verranno approfondite le tecnologie utilizzate per lo sviluppo del progetto di tesi: verranno riassunte funzionalità base dei software Blender e Unity, i Framework specifici per lo sviluppo su Hololens (Vuforia per il tracking, MRTK per sviluppo applicazioni AR) e si parlerà della tecnologia Hardware del dispositivo su cui girerà



l'applicativo finale, ovvero le Hololens 2.

Il capitolo 5 entrerà nello specifico del progetto di questa tesi, descrivendone funzionamento e modalità di sviluppo delle applicazioni Desktop e Hololens. Nel capitolo 6 verranno mostrati i risultati ottenuti dall'utilizzo delle Hololens 2 per questo caso d'uso, evidenziandone pregi e limiti. Nel capitolo 6 oltre alle conclusioni verranno proposti ipotetici sviluppi futuri dell'applicativo.

## Capitolo 2

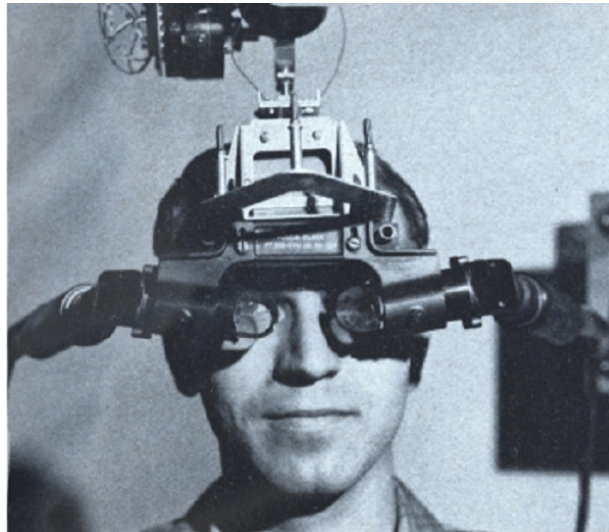
# Concetti Generali

In questo capitolo si entrerà nel merito della Realtà Aumentata, sia parlando della sua storia e della sua evoluzione, sia approfondendo il suo funzionamento in termini di architettura Hardware e Software.

### 2.1 Origini della Realtà Aumentata

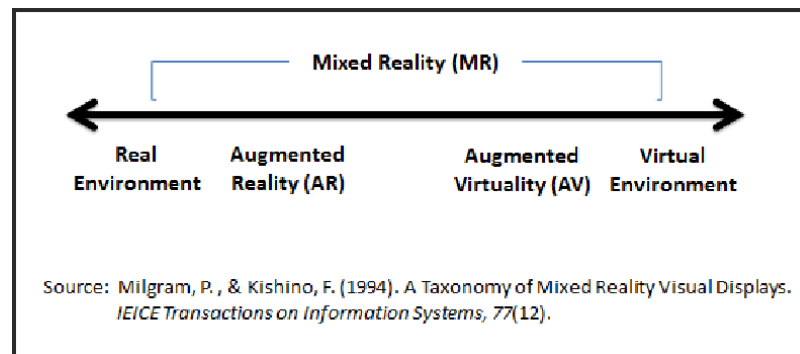
Con il termine Realtà Aumentata si indica la sovrapposizione di oggetti 3D digitali in un mondo reale, in modo tale da fornire all'utente informazioni appartenenti a entrambi i mondi. Al fine di comprendere l'uso della realtà aumentata nei giorni nostri, si andranno ad esplorare le sue origini fino ad arrivare all'impatto che ha nei giorni nostri.

Nel 1966 **Ivan Sutherland**, informatico e ricercatore statunitense pioniere di internet, ideò un prototipo che prende il nome di "The Sword of Damocles"[3], che sarà poi considerato il primo sistema HMD di realtà aumentata. Obiettivo del ricercatore era circondare l'utente di informazioni tridimensionali, e per farlo utilizzò una grafica wireframe e un Head Mounted Display molto pesante per costruire il primo sistema di AR. Questo sistema, a differenza della computer grafica (il cui avvento è precedente) implica che la visualizzazione degli elementi grafici cambi a seconda dalla posizione della testa utente. Questo richiedeva una nuova forma di tecnologia mai usata precedentemente, ossia un sensore che riuscisse a misurare posizione e angolo di rotazione della testa dell'utente. In base a queste informazioni, veniva visualizzata un'angolazione diversa dell'oggetto virtuale rappresentato.



**Figura 2.1:** The Sword of Damocles

Per vedere la prima applicazione di Realtà Aumentata in ambito industriale bisognerà aspettare il 1990 quando due ingegneri di Boeing, **Tom Caudell** e **David Mitzell**, utilizzarono unHMD per poter sovrapporre e stabilizzare un diagramma prodotto da un computer in una posizione specifica del mondo reale. [4]Questo utilizzo della Realtà Aumentata permetteva di ridurre tempi e costi per le operazioni di assemblaggio, aumentando l'efficacia nella produzione di aereomobili. Caudell è inoltre colui che ha coniato il termine Realtà Aumentata.



**Figura 2.2:** Diagramma VR Continuum

Nel 1994 **Paul Milgram** pubblicò un articolo dal nome “*Augmented Reality: A class of displays on the reality-virtuality continuum*” [5], in cui viene introdotto il Reality-Virtuality continuum, ossia un diagramma che classifica le diverse tecnologie vicine alla Realtà Aumentata. Come si può vedere in figura 2.2, agli estremi troviamo

ambienti o completamente virtuali (a sinistra) o completamente reali (a destra). Al centro c'è un insieme di ambienti di Realtà Mista, che integrano in vario modo e in percentuale diversa le varie realtà. A seconda della predominanza della parte reale rispetto alla componente virtuale, troviamo, in ordine, la Realtà Aumentata, la Realtà Mista e la Virtualità Aumentata. L'unione della Realtà Mista e della Realtà Virtuale costituisce la Realtà Estesa (XR).

Seguirono poi negli anni altre ricerche. Finora tutti i sistemi di Realtà Aumentata erano pensati per funzionare in ambienti chiusi, sia perché le condizioni di luce possono essere controllate, sia perché alcune risorse (come sensori, calcolatori e batterie) sono limitate in relazione a quanto un utente può trasportare. Nel 1994 **Ronald Azuma**, ricercatore americano, ideò una tecnologia che permetteva alle applicazioni AR di essere utilizzate in ambienti outdoor, con lo scopo di trovare nuove aree di applicazione dell'AR. Sviluppò un sistema di tracking ibrido, combinando diversi sensori (giroscopi, bussola e sensore di orientamento) per ottenere un risultato molto accurato. [6]

Sempre Ronald Azuma nel 1997 definì la Realtà aumentata come combinazione di mondo reale e oggetti virtuali, con cui si può interagire in real time, integrati nel campo visivo dell'utente.

La svolta si ha nel 1999: **Hirokazu Kato**, professore giapponese presso Nara Institute of Science and Technology, rilascia la prima libreria software open-source multiplatforma per permettere agli sviluppatori di realizzare applicazioni di realtà aumentata. Tale libreria prende il nome di *ARToolKit*. Attualmente, è ancora disponibile il download su GitHub. Questo SDK fornisce i tool necessari per permettere di individuare le informazioni di posizione e orientamento della camera grazie all'utilizzo di marker quadrati posizionati nello spazio. In questo modo era possibile posizionare una camera virtuale nello stesso punto della camera reale, per permettere di visualizzare i modelli 3D sovrapposti al marker reale. ARToolKit quindi affronta due problemi noti della Realtà Virtuale: il viewport tracking e l'interazione virtuale. [7]

**ARToolKit** ha permesso di avvicinare molti programmatori nello sviluppo di applicazioni di Realtà Aumentata. Grazie anche all'aumento della potenza di calcolo dei PC, nei primi anni 2000 nascono le prime applicazioni AR dedicate all'entertainment: il primo esempio fu il gioco AR Quake di Bruce H. Thomas, che permetteva all'utente di muoversi nel mondo fisico e al contempo interagire con mostri e oggetti generati da un computer. Il sistema hardware per la realizzazione di tale gioco prevedeva un computer da tenere in uno zaino, un giroscopio, un sensore, un GPS, un HMD.[8]

Ad ARToolKit si susseguono nel corso degli anni diversi kit di sviluppo software analoghi a lui.

Nel 2015 l'azienda PTC Inc. acquista **Vuforia**, un kit di sviluppo software sia per dispositivi mobile che per dispositivi indossabili HMD. Vuforia utilizza le

tecnologie di computer vision per riconoscere e tracciare immagini 2D o oggetti 3D in realtime. In questo modo gli sviluppatori possono posizionare e orientare modelli virtuali in relazione all'oggetto reale.

Nel 2017 Apple ha introdotto **ARKit** come parte integrante del sistema operativo iOS 11. ARKit è una piattaforma di sviluppo che consente agli sviluppatori di creare facilmente esperienze di Realtà Aumentata nelle applicazioni. Per farlo si combinano le informazioni di fotocamera e sensori di rilevamento del dispositivo, in modo tale da permettere al dispositivo iOS di percepire il suo movimento nella stanza per poterne analizzare il layout. Inoltre è in grado di rilevare piani orizzontali come tavoli e pavimenti.

Nel 2018 Google lancia **ARCore**, un kit di sviluppo software per la creazione di esperienze di realtà aumentata. Questo kit utilizza diverse API che permettono ad un cellulare di rilevare l'ambiente circostante, di comprendere il mondo e interagire con le informazioni tridimensionali. La tecnologia di ARCore utilizza la fotocamera del telefono per evidenziare dei punti di interesse di cui si tiene traccia del movimento. Combinando i sensori di movimento del telefono e il movimento nel tempo degli oggetti in questione, si riesce a determinare la posizione e l'orientamento del cellulare mentre si muove nello spazio, ed è anche in grado di rilevare superfici piane e di stimare l'illuminazione media nell'area circostante.

## 2.2 Architettura della Realtà Aumentata

Dal momento che si è parlato di Realtà Aumentata come miscela di percezioni dirette e mediate dal computer, è necessario entrare nel merito per capire come vengono calcolati gli elementi digitali. Per consentire l'integrazione tra mondo fisico e mondo virtuale è necessario che il calcolatore riceva in input informazioni appartenenti ad entrambi i mondi, e fornisca in output informazioni da far visualizzare all'utente. Per gestire queste connessioni è necessario avere:

- **Sensori** (telecamere, microfoni, accelerometri, sistemi inerziali) che acquisiscono informazioni sul contesto reale che circonda l'utente, per capire le sue caratteristiche.
- **Analizzatore di contesto.** Riceve le informazioni catturate dai sensori e le elabora per dare una descrizione semantica del contesto circostante, come etichette o dati posizionali (ad esempio dove si trova l'utente rispetto a un sistema di riferimento comune). Queste informazioni servono all'applicazione per garantire l'integrazione di elementi reali e virtuali.
- **Motore** dell'AR, il vero e proprio cervello dell'applicazione. Riceve l'output fornito dall'analizzatore di contesto e dai singoli dati dei sensori, per poter gestire la logica dell'applicazione, definita tramite un insieme di regole specificate

dal designer. Il motore cattura i comandi che l'utente manda all'applicazione con diversi tipi di interazioni a seconda di come è stato sviluppato il prodotto: una UI cliccabile, un tasto di un controller, un movimento del corpo, delle gesture, la direzione dello sguardo o la voce.

- **Servizi esterni** (Opzionali). Alcune applicazioni possono comunicare con elementi di terze parti integrati nell'applicazione, come server specializzati nell'elaborazione di dati provenienti dai sensori. Ad esempio Azure fornisce delle API che permette di fare lo "speech to text" direttamente sui loro server.

### 2.2.1 Display AR

In questa sezione verranno analizzati gli strumenti hardware che si utilizzano per permettere all'utente di interfacciarsi con un'applicazione AR. E' bene ricordare che, a differenza della VR dove vengono visualizzati elementi totalmente sintetici, qui è necessario combinare insieme stimoli reali e virtuali in maniera efficace, ed esistono tre diversi paradigmi di integrazione tra i due elementi.

**See-through** In questo paradigma l'integrazione avviene tramite un HMD. Questo tipo di integrazione è distinto in due casi diversi a seconda dell'opacità della lente. I dispositivi con lenti trasparenti permettono di vedere direttamente il mondo reale come un qualsiasi occhiale, permettendo la sovrimpressione degli elementi virtuali grazie a delle proiezioni sulle lenti, che copriranno parzialmente i contenuti reali a seconda del potere con cui le immagini proiettate sembrano solide. Attualmente sul mercato sono presenti le HoloLens e i Magic Leap. I dispositivi con lenti opache con display LCD o OLED creano invece una visualizzazione mediata da una o due telecamere. Le informazioni che acquisiscono le telecamere vengono combinate con le informazioni sintetiche grazie a processi di rendering, in modo tale da generare un'immagine sul display che mostri entrambi i contenuti, reali e virtuali.

**Display based** In questo paradigma la fruizione di elementi di Realtà Aumentata avviene attraverso un display posizionato in un punto nello spazio. Più precisamente, se il dispositivo rimane in una posizione fissa nello spazio si parla di **Monitor Based Augmentation**, altrimenti si parla di **Hand Held Devices** (utilizzo di tablet o smartphone). Gli hand held devices rappresentano il modo più comune che ha l'utente standard per accedere a contenuti di realtà aumentata. Gli smartphone e i tablet di adesso hanno sia l'hardware sia le capacità elaborative necessarie per implementare un sistema di AR e gestirne le sue complessità: una telecamera ad alta risoluzione, un display per mostrare l'ambiente aumentato, un sistema per gestire le interazioni, una capacità di calcolo sufficiente per gestire le risorse

3D visualizzate. L'utente in questo caso controlla i movimenti della camera coi movimenti della mano e non direttamente con la testa come accade invece nel paradigma See-Through. I Monitor Based Augmentation sono concettualmente simili agli Hand Held Devices, ma con due differenze sostanziali. La prima è che fotocamera, dispositivo di elaborazione e display di visualizzazione sono tre dispositivi separati non racchiusi all'interno di un unico dispositivo. La seconda è che il monitor rimane in una posizione fissa all'interno dell'ambiente. L'unico elemento che si muove è la telecamera.

**Projector based / Spatial AR** In questo paradigma si proiettano i contenuti virtuali direttamente nello spazio fisico mediante un proiettore, senza l'uso esplicito di un display. Gli elementi sintetici da visualizzare vengono proiettati direttamente sulle superfici degli oggetti reali. Agendo in maniera opportuna è possibile far assumere all'oggetto una forma diversa senza cambiarne la forma originale, grazie all'aggiunta di dettagli, textures e ombre.

## 2.2.2 Registration, Tracking, Calibration

Per fare in modo che la sovrimpressione di mondo reale e mondo digitale sia corretta e accurata è necessario risolvere il problema del tracking, della calibration e della registration.

**Registration** Per registrazione si intende l'allineamento tra le caratteristiche spaziali degli elementi coinvolti all'interno dell'applicazione AR, ossia, la trasformazione dei sistemi di riferimento locali di ogni elemento (fotocamera, la posizione spaziale dell'utente, gli oggetti reali e sintetici) in un sistema di riferimento comune di tutto il mondo AR. Per garantire la coerenza tra mondo virtuale e mondo reale è infatti necessario avere un'illuminazione globale coerente e una gestione corretta delle occlusioni (se un oggetto virtuale si posiziona nello spazio dietro a un oggetto reale, dovrà essere parzialmente o totalmente occluso). In sostanza, il processo di registrazione si deve occupare di due processi distinti: Geometric Registration e Visual Coherence.

Per **Geometric Registration** si intende il posizionamento di un oggetto virtuale nello spazio del mondo 3D rispetto a un riferimento noto, come ad esempio un marker. Di quest'ultimo, grazie ai metodi di tracking, sono infatti note le informazioni di orientamento e posizione nel mondo reale, e definisce un suo sistema di riferimento locale. Tutti gli oggetti virtuali si posizionano nello spazio in relazione alla posizione del marker che, se viene spostato, sposta insieme a lui tutti gli elementi virtuali a lui associati.

Per **Visual Coherence** si intende una coerenza tra i due mondi sia dal punto di vista dell'illuminazione sia dal punto di vista delle occlusioni. Un oggetto

virtuale deve avere un'illuminazione coerente con il mondo reale per poter essere integrato. Per gestire la luce è necessario avere un sensore che riesca a identificare le fonti di luce ambientali. Attualmente la tecnologia AR è limitata su questo campo, riesce a fare solo una stima dell'illuminazione globale grazie ad algoritmi di Machine Learning. Le occlusioni sono un altro aspetto complicato nel mondo AR, ed è ad ora realizzabile soltanto utilizzando un approccio di tipo SLAM, ossia, con una ricostruzione 3D dell'ambiente circostante, che risulta tuttavia molto approssimativa.

**Tracking** Alla base del processo di registrazione c'è il tracking, che permette di misurare in realtime posizione e orientamento della camera (o della testa dell'utente se si parla di HMD) utilizzando una tecnologia di tracking ottico. Quando viene spostata la camera nel sistema di riferimento globale è necessario aggiornare le informazioni di registrazione per poter visualizzare correttamente i contenuti dell'ambiente, ottenendo un processo di registrazione dinamica che evolve nel tempo in funzione dei movimenti all'interno dell'ambiente.

A seconda del dispositivo e della telecamera utilizzata si distinguono due impostazioni diverse: tracking **outside-in**, in cui le camere sono fisse nell'ambiente e osservano gli oggetti che si muovono, e tracking **inside out**, con la camera agganciata direttamente nella testa dell'utente. In questo caso per ottenere le informazioni di tracking per la registrazione si usano diversi approcci:

- Utilizzo di marker all'interno dell'ambiente. I marker sono immagini 2D semplici, con caratteristiche riconoscibili grazie ad algoritmi di Computer Vision. Un esempio di marker possono essere i QRCode, simboli bianchi e neri. Il tracciamento in questo caso è poco costoso in termini di elaborazione.
- Tracking di features naturali. Vengono estratte le features naturali degli oggetti (grazie al tracciamento di punti di interesse, ad esempio individuazione di pattern particolari) che vengono confrontate con delle features contenute in un database, si cerca match tra le due, per poter costruire un sistema di riferimento dell'oggetto.
- Image target. Ricalca il concetto dei marker, con la differenza che contengono immagini generiche che li rendono esteticamente migliori rispetto ai marker bianchi e neri. Danno la possibilità al designer dell'applicazione AR di personalizzare le immagini.
- SLAM. L'ambiente reale viene ricostruito in 3D.

**Calibration** È un processo che si applica ai sensori che permette di ottenere misurazioni o dati su una scala nota e su un sistema di riferimento noto, e dipende



dal dispositivo di misurazione. Può essere eseguito offline o una volta ogni tanto in live prima di iniziare le operazioni. Lo scopo è fornire una corrispondenza perfetta tra la camera reale (intesa come occhio umano in un sistema HMD, o come la camera stessa in caso di dispositivi display) e la camera virtuale. Questo perché se si deve garantire una perfetta sovrapposizione, la camera virtuale deve avere parametri identici a quelli della camera reale, come la stessa focale o la stessa dimensione dei pixel (parametri intrinseci), la stessa posizione o lo stesso orientamento (parametri estrinseci). Se non vengono calcolati in maniera corretta i parametri estrinseci, la sovrapposizione risulterà sbagliata.

## Capitolo 3

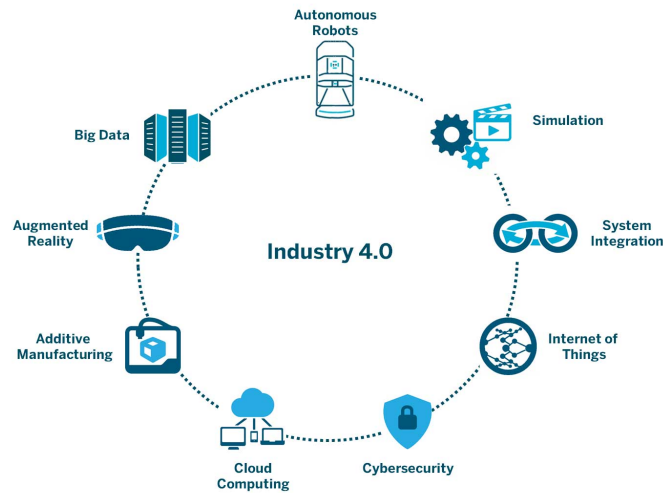
# Stato dell'arte

In questo capitolo verrà analizzato l'utilizzo delle tecnologie di Realtà Aumentata in ambito industriale, partendo da un'analisi dell'avvento dell'Industria 4.0 per poi entrare nel merito dei suoi utilizzi nel corso degli anni soprattutto per quanto riguarda il settore automotive.

### 3.1 Industria 4.0

La quarta rivoluzione industriale ha origine nel 2011 come un progetto di strategia high-tech del governo tedesco. Il termine Industria 4.0 è stato coniato da Henning Kagermann, Wolf-Dieter Lukasnel e Wolfgang Wahsler nel 2011 alla fiera di Hannover. E' considerata una rivoluzione guidata dalla tecnologia per ottenere una maggiore efficienza e produttività, in cui i sistemi di produzione possono comunicare tra loro in tempo reale. I requisiti che vuole soddisfare sono un aumento di produttività, flessibilità, velocità, qualità e competitività.

La prima rivoluzione industriale fu segnata dal passaggio da metodi di produzione manuale a metodi di produzione con macchine alimentate a vapore, la seconda rivoluzione industriale trasformò le fabbriche in linee di produzione moderne grazie all'applicazione di energia a una produzione di massa, mentre la terza rivoluzione industriale, conosciuta come rivoluzione Digitale, si riferisce all'utilizzo di elettronica e componenti IT per automatizzare la produzione. Nella quarta rivoluzione industriale i sistemi di produzione possono invece collaborare tra loro comunicando in real-time per prendere decisioni *smart* attraverso la cooperazione con altri sistemi produttivi. Per capire perchè l'avvento dell'Industria 4.0 sia considerata una Rivoluzione bisogna riconoscere le qualità distinte che i suoi concetti portano alla produzione industriale. Nell'immagine 3.1 vengono illustrate le diverse tecnologie associate al fenomeno dell'Industria 4.0.



**Figura 3.1:** Tecnologie dell'Industria 4.0

Alcuni concetti e tecnologie fondamentali che soddisfano i principi di progettazione dell'industria 4.0 sono:

- **Internet delle cose (IoT)**. Si intendono oggetti fisici dotati di sensori, software e altre tecnologie che creano una rete di interazione e scambio dati tramite il web.
- **Cloud Computing**. Si intende una distribuzione di servizi di calcolo, come server, risorse di archiviazione, database, rete, software tramite Internet ("il cloud"), che offrono innovazione, risorse flessibili ed economie di scala.
- **Cybersecurity**, un insieme di tecnologie il cui fine è proteggere le aziende o le reti dagli attacchi digitali.
- **Big Data e Intelligenza Artificiale**, ossia, l'abilità di un computer di avere capacità umane come ragionamento, apprendimento, pianificazione e creatività.
- **Robot autonomi**. Le azioni dei robot possono essere coordinate e automatizzate se collegati a un controller logico programmabile. In questo modo possono completare le attività in modo intelligente con il minimo input umano.
- **Simulazione**, concetto fortemente legato a quello di Fabbrica Digitale. Si intende un'imitazione del funzionamento di un processo o sistema del mondo reale nel tempo. Aiuta a identificare i colli di bottiglia nella produzione, e di identificare opportunità di riduzione dei costi.

- **Integrazione di sistemi**, in particolare si parla di integrazione orizzontale per i processi produttivi, ossia la connessione di sistemi cyber-fisici per introdurre nuovi livelli di automazione, e integrazione verticale per l'integrazione della produzione con altre aree aziendali a livello più alto (acquisti, controllo qualità)
- **Produzione additiva**. E' un insieme di tecnologie di produzione che permette di ottenere un prodotto finale attraverso generazione e aggiunta di strati di materiale. In questo modo si producono meno sprechi e scarti rispetto alle tecniche di fresatura o tornitura e taglia tempi e costi di prototipazione fornendo una prova quasi istantanea dei miglioramenti effettuati al prodotto.
- **Augmented / Virtual Reality**, di cui si parlerà di seguito. Queste tecnologie forniscono un nuovo asset a supporto dei processi decisionali: del design di prodotto, alla formazione del personale, alla manutenzione degli impianti fino ad arrivare al controllo della sicurezza.

Di seguito verranno analizzati i quattro temi cardine dell'Industria 4.0.

**Interconnessione** E' l'abilità di devices, sensori e macchine di connettersi e di comunicare tra loro attraverso le tecnologie IoT. La comunicazione macchina-macchina e macchina-persona è una parte cruciale di questa rivoluzione industriale. L'Internet of Things permette questa connessione, consentendo ai produttori di collezionare informazioni su macchinari e attrezzature in maniera facile.

**Trasparenza delle informazioni** Talvolta la produzione è frenata dalla mancanza di visibilità: nella linea di produzione ci sono innumerevoli punti ciechi in cui gli operatori non hanno dati necessari per apportare miglioramenti. Grazie all'interconnessione che fornisce un grande quantitativo di dati e informazioni, gli operatori possono valutare inefficienze e problemi che prima non erano visibili. Inoltre, i produttori hanno la possibilità di raccogliere dati anche dopo che il prodotto ha lasciato la fabbrica.

**Assistenza tecnica** L'apprendimento automatico consente a dispositivi e applicazioni di valutare i dati in modo molto più rapido e accurato rispetto agli esseri umani. Per quanto riguarda processi che richiedono ancora il processo decisionale umano, la tecnologia supporta e assiste l'essere umano nel processo decisionale e nel problem-solving, aiutandolo in compiti difficili o non sicuri, fornendo avvisi e notifiche che lo aiutino a identificare guasti o punti deboli. La tecnologia può anche supportare gli esseri umani per la manutenzione predittiva: l'intelligenza artificiale, i big data e l'IoT consentono ai dispositivi di riconoscere quando è necessaria un'operazione di manutenzione prima che si verifichi un guasto.

**Decisioni decentralizzate** E' l'abilità di sistemi informatici di prendere decisioni per conto proprio per proseguire nei loro task autonomamente. I compiti sono delegati a un livello superiore soltanto in caso di eccezione, interferenza o conflitti. Questo consente agli operatori di concentrarsi maggiormente sulle situazioni prioritarie.

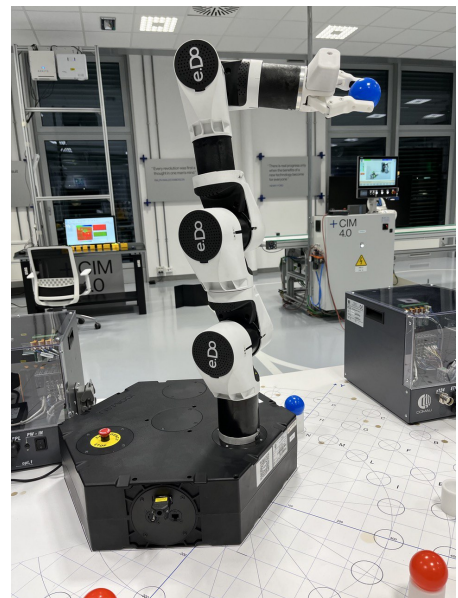
Come visto precedentemente, la Realtà Aumentata è una delle tecnologie più promettenti per migliorare il trasferimento delle informazioni dal mondo digitale al mondo fisico in maniera non intrusiva. Grazie alla sua capacità di integrare informazioni virtuali nella percezione dell'utente nel mondo reale, l'AR può supportare gli operatori nella loro stazione di lavoro, in real time e durante operazioni manuali, riducendo l'errore umano. Negli ultimi anni, diversi sistemi basati su Augmented Reality sono stati sviluppati dai ricercatori per diversi settori di industria manifatturiera a supporto delle operazioni, specialmente in quelle di assemblaggio, formazione, manutenzione e ispezione, che tipicamente coinvolgono la rilevazione di discrepanze di design tra quanto è stato pianificato e quanto è stato costruito. [9] L'AR è classificata dall'Unione Europea come una delle principali tecnologie che guideranno lo sviluppo della fabbrica intelligente abilitando le interazioni uomo-macchina attraverso la visualizzazione di informazioni interattive e contestuali. Al fine di facilitare la collaborazione e l'interazione tra esseri umani e sistemi di produzione basati su dati digitali, i ricercatori si concentrano sull'utilizzo dell'AR nelle applicazioni industriali. Inoltre, continua ad essere di crescente importanza a causa del progresso tecnologico, della digitalizzazione e, più recentemente, delle esigenze indotte dal COVID-19 in diversi settori di introdurre e/o migliorare le comunicazioni e la sicurezza in un ambiente senza carta.

### 3.1.1 CIM 4.0

Il Centro di competenza Competence Industry Manufacturing 4.0 (abbreviato in CIM 4.0) è uno degli otto Competence Center italiani sui temi dell'Industria 4.0. Il suo scopo è realizzare un supporto concreto alle imprese dei settori Automotive, Aerospace e dell'industria manifatturiera orientate alla trasformazione digitale dei processi dalla progettazione alla produzione, dall'R&D alla Supply chain, dalla sicurezza alla Blockchain. Ha messo a disposizione la strumentazione e le strutture necessarie per lo sviluppo dell'applicativo. In figura 3.2 e 3.3 si possono visualizzare alcune delle tecnologie che si possono trovare presso l'area Digital.



**Figura 3.2:** Digital Pilot Line



**Figura 3.3:** Comau

## 3.2 Utilizzo della Realtà Aumentata in ambito Automotive

In molte industrie l'implementazione di moderne tecnologie può portare numerosi benefici, come un minore costo di produzione e una massimizzazione dei profitti. Le applicazioni dell'AR spaziano infatti dalle industrie manifatturiere, all'industria della robotica, alla sanità, all'educazione, ambito militare, campo dell'entertainment e così via. I produttori di automobili in particolare sono sempre stati in prima linea nello sfruttare i moderni progressi tecnologici. L'industria automobilistica ha infatti sviluppato, negli ultimi anni, un crescente interesse nella Realtà Aumentata grazie alla sua accessibilità, alla sua interfaccia intuitiva e al suo potenziale nel saper generare soluzioni innovative. La Realtà Aumentata è una tecnologia che abilita l'interazione uomo-macchina (HMI) mediante la sovrapposizione di informazioni virtuali interattive su un ambiente reale. E' uno strumento utile in quanto permette di ottenere informazioni necessarie riguardo alcuni aspetti o procedure, in maniera diretta e istantanea, nell'ambiente lavorativo. Può contribuire in diversi altri aspetti dell'industria automotive, quali manutenzione, riparazione, diagnostica, ispezione, formazione o come strumento di marketing per un pubblico consumer.

### 3.2.1 Applicazioni commerciali

Nel mondo odierno stiamo sperimentando un rapido avanzamento di dispositivi come smartphone e tablet che hanno una capacità hardware tale da permettere la fruizione di contenuti di Realtà Aumentata (GPS, sensori, fotocamere di alta qualità) non solo alle aziende, ma anche a un pubblico consumer. Molte note aziende automobilistiche hanno sfruttato questa tecnologia come una nuova strategia di marketing per i loro prodotti, o come mezzo per rendere più semplice l'esperienza di acquisto per il cliente.

L'azienda **Volkswagen** ha lanciato una campagna di marketing per pubblicizzare la Golf Cabriolet, completamente incentrata sulla Realtà Aumentata e fruibile attraverso smartphone e tablet. Gli utenti dovevano inquadrare con la camera del telefono gli annunci pubblicitari dell'auto contenuti in diverse riviste. Attraverso un algoritmo di riconoscimento dell'immagine e tracking, veniva lanciata l'applicazione di realtà aumentata che permetteva agli utenti di partecipare a una competizione virtuale per poter vincere diversi premi, il più importante era la possibilità di fare un giro di prova sul veicolo. Secondo la casa automobilistica, gli utenti partecipanti sono stati più di 50.000.

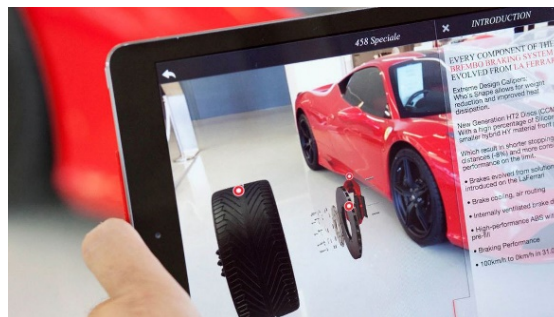
Nel 2012 all'evento Genova Auto Show, la **Volvo** ha dato la possibilità all'utente di vedere gli interni di una Volvo V40, un nuovo modello. L'applicazione utilizzava il tracking mediante un marker 2D che, se inquadrato con un tablet o uno smartphone, mostrava gli interni dell'auto in 3D.

L'azienda **Hyundai** nel 2015 ha sviluppato un'applicazione AR dal nome di *Hyundai Virtual Guide*, un manuale *smart*, con lo scopo di mostrare direttamente nei punti interessati dettagli riguardanti la procedura di manutenzione, la riparazione o altri strumenti appositi che guidano nella risoluzione di problemi minori, senza necessità di cercare le istruzioni in un manuale. In questo modo si garantisce un aumento della sicurezza del cliente proprietario dell'auto. [10]



**Figura 3.4:** Hyundai Virtual Guide

Nel 2015 la **Ferrari** utilizza una tecnologia di tracking 3D per permettere al software di riconoscere le macchine in un mondo reale, dando la possibilità di sovrapporre informazioni quali forme e colori in modo tale che il cliente possa costruire le specifiche personalizzate per le proprie macchine senza che il venditore debba ordinare preventivamente gli accessori. [11]



**Figura 3.5:** Applicazione della Ferrari

### 3.2.2 Applicazioni industriali

Questa tecnologia human-centered si è rivelata utile per operatori e lavoratori nell'area della produzione di auto. Le aziende di automobili hanno introdotto la tecnologia AR in ambienti di manifattura per assemblaggio, manutenzione o ispezione, fornendo assistenza agli operatori. I tecnici ad esempio possono utilizzare



i sistemi AR per visualizzare istruzioni posizionate di fronte ai loro occhi che illustrano la metodologia che devono adottare per la diagnostica e la riparazione del veicolo, come devono agire, riducendo così il tempo di apprendimento e l'effort mentale. [12]

Le soluzioni di AR offrono innovazione e tecnica per progettare e creare auto moderne. Attualmente le sfide comuni nelle operazioni automobilistiche sono:

- Risoluzione di problemi nel design operativo, dato che richiede continua innovazione e miglioramento.
- Calibrazione manuale e setup per il processo di assemblaggio. Molti processi di assemblaggio sono eseguiti manualmente da un operatore umano. Anche una piccola inefficienza operativa può portare a un processo incompleto o mancato.
- Carenza di formazione e coaching di competenze. Le aziende produttrici di automobili devono soddisfare gli standard di alta precisione nell'assemblaggio delle parti. Per questo hanno bisogno di un alto livello di competenza nel servizio, e di dipendenti qualificati per il montaggio, manutenzione e assistenza sul campo.

L'introduzione di dispositivi di realtà aumentata ha permesso di migliorare alcune delle fasi produttive per le aziende automotive. In particolare:

- Svolge un ruolo importante nella fase di progettazione di un'auto, fornendo un'immagine perfetta dell'aspetto del veicolo su strada. I produttori possono infatti esplorare e modificare il design, riducendo costi e tempo necessari.
- Automatizza le fasi di produzione delle automobili. Per gli assemblatori di linea, la loro formazione tramite HMD permette di visualizzare sul display informazioni che mostrano direttamente le operazioni da applicare in ogni sezione. In questo modo si migliora velocità e qualità dell'assemblaggio.
- Introduzione di un laboratorio virtuale in cui si possono risolvere problemi relativi a ergonomia, crash test o funzionalità dell'esperienza utente. I dipendenti possono vedere come il prodotto interagisce in vari ambienti e situazioni.
- Formazione virtuale: consente ai dipendenti di apprendere i principi di progettazione, costruzione e assemblaggio usando modelli di formazione interattivi e automatici che guidano passo passo, anziché utilizzare modelli fisici in argilla.

Indubbiamente quindi l'esperienza di realtà aumentata si è fatta strada nel settore automobilistico, dando numerosi vantaggi per le operazioni, dalla progettazione alla produzione e infine alla vendita. A supporto di questa teoria verranno citati

diversi studi effettuati nel corso degli anni in cui viene sottolineato l'impatto delle tecnologie AR in campo industriale. [13]

Nel processo di assemblaggio, le aziende dipendono sempre più dalla tecnologia per ridurre il numero di processi manuali, che vede coinvolto un grande numero di lavoratori umani. A questo scopo, sono state effettuate numerose ricerche accademiche a favore del fatto che l'utilizzo della Realtà Aumentata possa essere importante per il risparmio di tempo e soldi, aumentando la *first time quality* e la diminuzione degli errori.

Nel 1992 i due ingegneri **Caudell e Mizell** furono i primi ad utilizzare un sistema di Realtà Aumentata attraverso un dispositivo HMD per le attività a supporto della manutenzione industriale. In un display PC chiamato *The Private Eye* è attaccato un sistema di tracking che utilizza un campo magnetico alternato per calcolare fino a 50 volte al secondo i 6 gradi di libertà relativi all'unità di trasmissione. Questo sistema è stato utilizzato per un processo di rimozione di una parte da un aeromobile a scopo di manutenzione. Veniva visualizzato un wireframe 3D sovrapposto alla parte da rimuovere, con l'animazione che mostra all'utente l'esatto percorso da compiere per essere rimosso correttamente. Nonostante il sistema presentasse molte debolezze (la sovrapposizione grafica era visibile soltanto nell'occhio destro, il FOV era di circa 7 gradi, c'è un lag di tempo tra il movimento della testa e il cambio della grafica), ha potuto dimostrare il potenziale della tecnologia per le applicazioni di manutenzione: i lavoratori migliorano efficienza e qualità nella loro performance di attività di manutenzione. [4]

Nel 1999 fu fatta una ricerca che esaminò come la Realtà Aumentata potrebbe essere utilizzata per aiutare un operatore a svolgere un'attività di assemblaggio di produzione. Nello specifico furono confrontate due soluzioni che utilizzarono display AR (HMD see-through e HMD monoculare opaco) e due soluzioni che non prevedono utilizzo di tecnologie di realtà aumentata (manuali di carta o su computer). I risultati indicarono che entrambe le soluzioni che prevedevano AR risultavano più efficaci in termini di tempi di assemblaggio e in termini di riduzione degli errori, anche se i caschetti indossabili risultavano fastidiosi e scomodi. [14].

In uno studio del 2017 [1] sono stati paragonati 3 diversi modalità di fruizione di istruzioni per l'assemblaggio: Desktop based, Tablet MBI e Tablet AR. Si può notare che con l'utilizzo di Tablet AR si ha una riduzione del tempo di assemblaggio del 33% rispetto alle altre due soluzioni. Questo perché si tiene conto del fatto che le istruzioni AR garantiscono più tempo per concentrarsi sull'assemblaggio stesso, mentre le soluzioni desktop-based richiedevano uno spostamento dell'attenzione da display fisso a punto di applicazione.

In un altro studio, sempre del 2017 [15] è stato chiesto a un gruppo di studenti di Ingegneria Meccanica di valutare le Hololens come device adeguato per fornire informazioni di assemblaggio. L'obiettivo era valutare e migliorare l'assistenza alle attività di assemblaggio mediante AR. Per l'esperimento, sono stati interpellati

32 studenti che dovevano seguire delle istruzioni step-by-step mostrati mediante animazioni 3D per assemblare 8 parti. Dal questionario si evince che le hololens sono uno strumento potente come guida a queste procedure, pur avendo qualche limitatezza: dopo un po' di tempo infatti alcuni sentivano fastidio nell'indossare a lungo un oggetto relativamente pesante in testa, il FOV era ridotto, chi indossava gli occhiali non si trovava bene.

Altri studi più recenti (2021) [16] hanno dimostrato che l'utilizzo degli occhiali AR gioca un ruolo fondamentale nella riduzione del carico cognitivo degli operatori che devono fare assemblaggi. Lo studio è stato fatto combinando fattori di scala soggettivi e oggettivi (battito cardiaco, movimenti degli occhi, EEG). Per l'esperimento è stato ricreato un ambiente che ricalchi una linea di assemblaggio, gli operatori dovevano completare dei task che fanno normalmente nella loro giornata lavorativa. Lo scopo dell'utilizzo di occhiali AR è ridurre il carico cognitivo riducendo di conseguenza il rischio di errori. Confrontando l'esecuzione dei task con AR e senza AR, dai risultati sia soggettivi sia oggettivi (che combaciano) si evince che l'uso della tecnologia di Realtà Aumentata influenza positivamente nel carico cognitivo.

Alla luce di queste ricerche accademiche, vediamo adesso come alcune note case automobilistiche utilizzano la tecnologia AR per aumentare l'efficienza nelle diverse fasi di produzione, dalla formazione, all'ispezione, all'assistenza.

**Mercedes-Benz** sta aumentando l'efficienza dei tecnici e riducendo il proprio impatto ambientale con l'assistenza remota in Realtà Aumentata. Negli ultimi anni, parallelamente al progresso tecnologico delle automobili, troviamo veicoli sempre più complessi, con una conseguente maggiore complessità nelle attività di manutenzione. Per questo motivo Mercedes necessitava di aiutare i tecnici a padroneggiare complesse procedure di manutenzione, riducendo al minimo i viaggi e le visite in loco ai centri di assistenza dei concessionari. Sfruttando le Hololens, con l'applicazione *Dynamics 365 Remote Assistant* (figura 3.6) si è trovata una soluzione a questo problema. Quando un tecnico ha bisogno della guida di esperti, lanciando l'applicazione sul caschetto, si può collegare a uno specialista in remoto che potrà visualizzare ciò che il tecnico sta riprendendo con le camere del dispositivo HMD. Lo specialista potrà interagire in real time con le Hololens, disegnando direttamente da desktop una freccia o una linea sullo schermo del PC, che il tecnico potrà visualizzare direttamente nell'ambiente. Questa tecnologia ha permesso anche di evitare i rallentamenti produttivi dovuti alla pandemia Covid-19. Ciò che si può notare di questo sistema è la sua efficienza: è possibile avere supporto istantaneo in tutto il mondo con un sistema semplice e intuitivo. [17]



**Figura 3.6:** Dynamics 365 Remote Assistant

L'azienda **Bosch** è pioniere nell'utilizzo di applicazioni AR nel settore automotive, che hanno ottimizzato molti processi lavorativi in diverse aree nel corso degli anni. Bosch ha sviluppato il *Common Augmented Reality Platform* (figura 3.7), disponibile per HoloLens 2, ed è la prima compagnia nel mondo a usare e testare le HoloLens 2 come parte di una cooperazione strategica. Secondo Bosch la Realtà Aumentata consente un risparmio di tempo medio del 15% per ogni passaggio effettuato, in caso di attività di riparazione meno sofisticate. La *Common Augmented Reality Platform* CAP consente l'integrazione rapida e semplice di contenuti digitali e visivi, e combinata con HoloLens 2 si creano scenari fino a poco tempo fa considerati visionari. L'obiettivo della piattaforma è permettere ai supervisori della formazione di creare e aggiornare facilmente scenari di addestramento e assistenza per gli aggiornamenti del veicolo, permettendo ai tirocinanti di officina di rimanere aggiornati sugli ultimi sviluppi. In questo scenario, i tecnici utilizzano o tablet o dispositivi indossabili per visualizzare nuove informazioni aggiornate sul veicolo, come una componente nascosta, e poterla visualizzare e interagirci (ruotarla, ingrandirla). La natura interattiva di queste soluzioni di formazione ha dimostrato di ottimizzare significativamente la qualità e il tempo dei corsi di formazione. Inoltre CAP supporta la Modalità Trainer, in cui l'insegnante condivide lo stesso contenuto 3D con tutti i partecipanti alla formazione. Questo permette ai meccanici di lavorare in modo più efficiente, con una diminuzione del tasso di errori durante le attività di riparazione e manutenzione. [18]



**Figura 3.7:** Common Augmented Reality Platform di Bosch

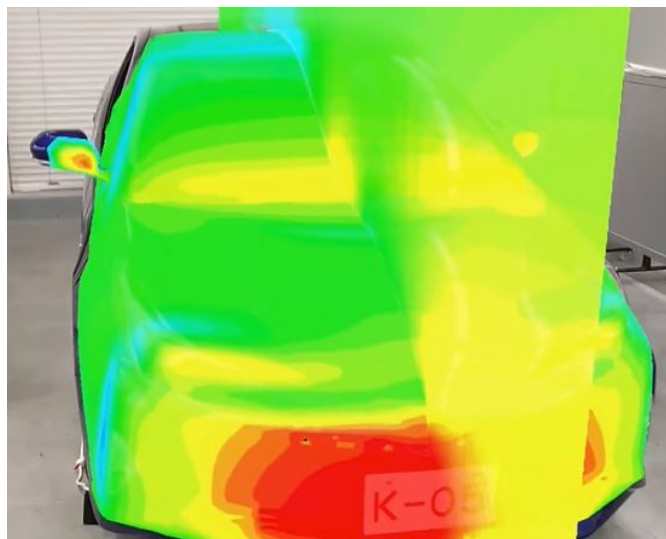
**Volvo** invece utilizza Varjo, un AR Headset di fascia alta che fornisce visioni di XR estremamente realistiche, per creare il primo approccio di Mixed Reality per la valutazione di prototipi, design e tecnologie di sicurezza. Per la prima volta, è possibile guidare una macchina indossando un caschetto AR, aggiungendo elementi virtuali che sembrano reali sia al guidatore sia ai sensori dell'auto, per scopi di sviluppo. In questo modo si può iniziare a valutare il design immediatamente e in tutte le fasi di sviluppo, offrendo potenziali risparmi sui costi perchè permette di identificare per tempo le priorità e di eliminare i colli di bottiglia molto prima nel processo di progettazione e sviluppo. Il Varjo permette anche agli ingegneri di sviluppare e valutare soluzioni di sicurezza più facilmente: gli esperti di sicurezza possono guidare macchine indossando l'headset, testando sistemi di sicurezza attiva imposti tramite la realtà aumentata. Inoltre la tecnologia di tracciamento oculare rende facile valutare come i conducenti si avvicinano a una nuova funzionalità, per vedere se sono distratti, in modo tale da permettere di svilupparne nuove senza causare distrazioni aggiuntive. [19]



**Figura 3.8:** Esempio delle funzionalità di Varjo con Volvo

**Toyota** si rivolge spesso alla tecnologia per migliorare qualità, efficienza e abbassare i costi. L'azienda vuole sviluppare applicazioni di realtà mista nel suo processo di produzione automobilistica, utilizzando le Hololens 2 per diversi campi:

- Miglioramento e semplificazione del processo di revisione del progetto per l'analisi della dinamica dei fluidi computazionale. Indossando il caschetto, l'utente può simulare e analizzare come il suo design influisce sull'aerodinamica. Lo scenario può anche essere condiviso in team se più persone indossano le Hololens, per poter dare la possibilità di comunicare e collaborare durante la revisione.



**Figura 3.9:** Analisi CFD del veicolo in real-time

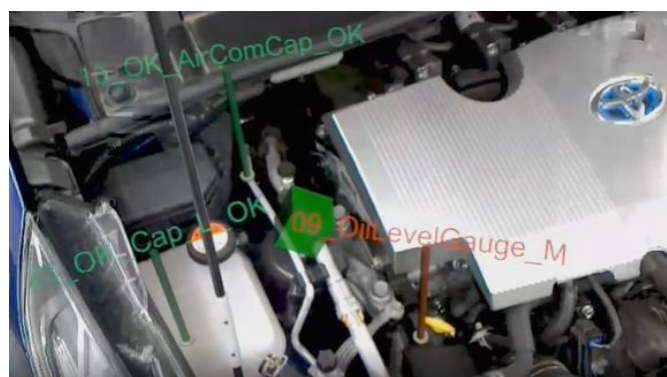


- Mostrare funzionalità e meccanismi nascosti all'interno del veicolo una volta assemblato. L'utente che indossa le Hololens può spostarsi e visualizzerà il funzionamento interno di un veicolo in movimento mediante ologrammi animati rappresentativi.



**Figura 3.10:** Funzionalità e meccanismi del veicolo Toyota

- Ridurre l'errore umano durante le ispezioni tramite l'apprendimento automatico. Anche i tecnici esperti possono commettere errori, che comportano conseguenze se non vengono rilevati immediatamente. Con l'aiuto del machine learning, gli ingegneri vengono guidati a riconoscere e correggere le incoerenze che in un'ispezione ordinaria potrebbero essere ignorate.



**Figura 3.11:** Individuazione di errori tramite machine learning

- Ridurre gli errori nella configurazione del cablaggio elettrico. Grazie alla visualizzazione dello schema elettrico in 3D, i tecnici di assistenza possono visualizzare la posizione dei sistemi di cablaggio senza perdere tempo nella rimozione delle parti fisiche.
- Abilitazione dell'assistenza remota nel servizio. Anche Toyota utilizza Microsoft Dynamics 365 Remote Assistant per la collaborazione a distanza. In questo modo il personale da remoto può ispezionare il lavoro e formare gli ingegneri sul campo.
- Velocizza la creazione di applicazioni di formazione mediante tutorial che seguono passo passo l'attività dei tecnici grazie all'applicazione Microsoft Dynamics 365 Guides.

**Faurencia**, gruppo francese di ingegneria e produzione di attrezzature per le case automobilistiche, ha implementato HoloLens 2 con *Dynamics 365 Remote Assistant* come soluzione per aiutare gli operatori a risolvere guasti in modo semplice ed efficiente. Grazie alla proiezione di ologrammi mediante HoloLens 2, i tecnici possono comprendere le cause dei guasti in tempo reale. In più, anche in questo caso, i tecnici possono farsi aiutare da un esperto in remoto, ed eventualmente se una riparazione prevede di effettuare un ordine, questo può essere avviato molto più velocemente. In questo modo, i tempi di produzione si riducono drasticamente. Si stima che si passa da 20 ore a 2 ore di fermo macchina.



**Figura 3.12:** Remote Assistant con Faurencia

**Lockheed Martin Corporation**, un'azienda americana che si occupa di aerospazio, armi, difesa e tecnologia, utilizza le HoloLens come strumento che mostri come poter assemblare i sedili dell'equipaggio per una navicella spaziale. Il dispositivo permetteva infatti di visualizzare, mediante animazioni e disegni



tecnici, l'operazione da svolgere direttamente sull'oggetto interessato. La navicella spaziale ha 57.000 elementi di fissaggio, ognuno da misurare manualmente. Grazie alla visualizzazione in Realtà Aumentata, è possibile vedere una versione digitale del dispositivo di fissaggio, di conseguenza anzichè fare misurazioni si può vedere esattamente dove posizionare l'elemento. In questo modo c'è un risparmio del 90% del lavoro, passando da una tempistica di 8 ore per contrassegnare le posizioni, a 45 minuti. [20]



**Figura 3.13:** Esempio applicazione AR di Lockheed Martin Corporation

# Capitolo 4

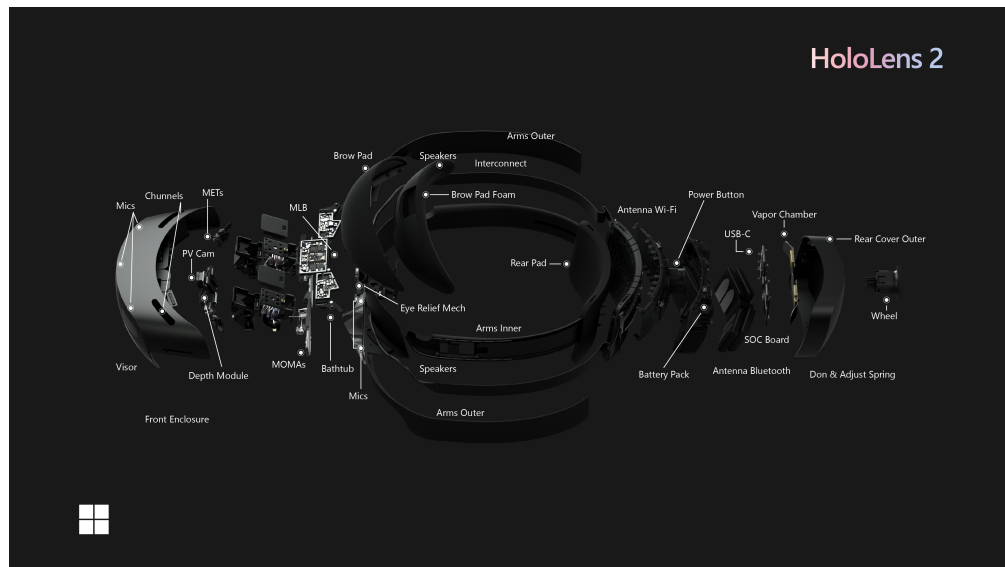
## Tecnologie e strumenti

Il primo step nella fase di design per la realizzazione dell'applicazione finale consiste nella scelta di soluzioni Hardware e Software ad oggi disponibili per arrivare alla realizzazione finale. A tal proposito, in questo capitolo verranno approfonditi i diversi strumenti hardware e software utilizzati per costruire l'applicazione.

### 4.1 Microsoft Hololens 2

Microsoft Hololens 2 è un dispositivo HMD indossabile di tipo see-through con lenti trasparenti che permette di fruire di esperienze di Realtà Aumentata mediante visualizzazione di contenuto olografico sovrapposto all'ambiente reale. E' un dispositivo untethered, ossia, ha internamente una CPU pertanto non ha bisogno di cavi di collegamento a una fonte esterna per funzionare. Si basa sul sistema operativo Windows Holographic, basato su Windows 10, ed è stato rilasciato nel Novembre 2019. I componenti hardware dell'Hololens sono:

- **Visore**, composto da Hololens e schermi.
- **Fascia** per posizionare il dispositivo mediante una rotellina di regolazione che la espande o la riduce.
- **Pulsanti per la luminosità** che si trovano sul lato sinistro del visore, per regolare la luminosità dello schermo.
- **Pulsanti per il volume** che si trovano sul lato destro del visore, per regolare l'intensità del volume.
- **Pulsante di accensione**, si trova sul lato destro del visore, e serve per accenderlo/spegnere/metterlo in stand-by.
- **Porta USB-C**, sempre sul lato destro, serve per poter caricare il dispositivo.



**Figura 4.1:** Hardware Hololens 2

Il peso del dispositivo è 566 grammi, e ha un campo visivo di 52 gradi, un notevole passo avanti rispetto a Hololens 1 che ne aveva solo 34. [21] Alcune delle caratteristiche che offre Hololens sono:

- **Mappatura spaziale.** Grazie ai sensori di profondità è possibile fare una stima dell'ambiente circostante, che verrà ricostruito in 3D sottoforma di mesh poligonali. Questa caratteristica permette di posizionare gli ologrammi sulle superfici reali e di creare occlusioni, ma anche di simulare un'interazione con l'ambiente reale con le forze fisiche (forza di gravità, rimbalzo).
- **Tracciamento della mano,** che permette all'utente di interagire con i contenuti virtuali con semplici gesti naturali (prendere un oggetto col gesto del pinch, schiacciare sui pulsanti premendoci sopra)
- **Comandi vocali,** che consentono al sistema di gestire le interazioni qualora l'utente avesse le mani occupate.
- **Tracciamento oculare.** Grazie alle due camere a infrarossi, Hololens capisce esattamente la direzione dello sguardo dell'utente al fine di adattare la visualizzazione degli ologrammi agli occhi in tempo reale. E' anche utile in fase di accensione per l'autenticazione: è infatti possibile fare riconoscimento dell'iride per evitare di inserire la password di accesso.

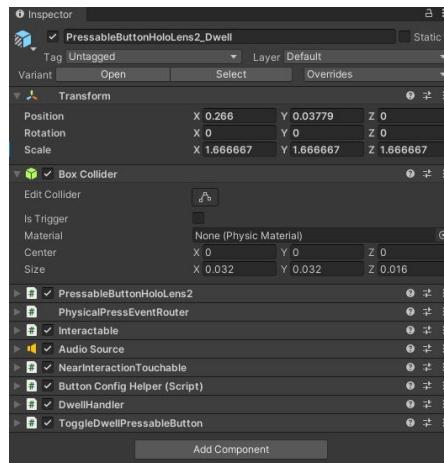
- **Tracciamento della testa** utilizzando un tracciamento a 6 gradi di libertà, ossia, fornisce al sistema l'esatto orientamento e posizione del dispositivo rispetto all'ambiente reale.

## 4.2 Unity

Unity è un game engine multiplatforma, che può essere utilizzato per creare applicazioni 2D, 3D ma anche applicazioni di Realtà Virtuale e Realtà Aumentata. E' stato scritto nel linguaggio di programmazione C++, ma le applicazioni si sviluppano utilizzando lo scripting con C#. La scelta di Unity come software principale di sviluppo per l'applicazione AR è dovuta in primo luogo al fatto che è un software gratuito (se utilizzato a livello personale), in secondo luogo perchè supporta lo sviluppo di applicazioni di Realtà Aumentata in maniera semplice ed efficace, in terzo luogo perchè i toolkit di programmazione forniti dalla Microsoft per la programmazione su Hololens sono compatibili con il software, e offrono tutta la documentazione necessaria per lo sviluppo. La versione di Unity utilizzata per questo progetto è stata la 2020.3.14f1.

Ogni elemento presente nella scena di Unity è considerato come un `GameObject` separato dagli altri. Ogni `GameObject` possiede diversi componenti, visualizzabili nella finestra di ispezione, tra cui:

- **Transform.** Elemento a sua volta composto da Posizione, Rotazione e fattore di Scala dell'oggetto.
- **Collider.** Elemento che indica una primitiva di collisione dell'oggetto.
- **Mesh Renderer.** Elemento che si occupa di renderizzare la geometria, mostrandola nel punto definito nella componente `Transform.position`. Contiene informazioni su materiali della geometria o su come la luce interagisce con l'elemento geometrico.
- **Rigid Body.** Abilita l'oggetto a comportarsi in relazione a determinate leggi fisiche: può ricevere forze esterne, può reagire alla forza di gravità, può interagire con altri oggetti dotati di Rigid Body.
- **Script** che ne definiscono il comportamento. Come detto precedentemente, gli script sono scritti utilizzando il linguaggio di programmazione C#, e ogni script è associato a una classe che, per poter essere aggiunto come componente a un `GameObject`, deve ereditare la classe `MonoBehaviour`. In questo modo, è possibile utilizzare alcune funzioni base di Unity come `Start()` (funzione chiamata al primo frame, responsabile dell'inizializzazione di variabili locali o globali) e `Update()` (funzione chiamata a ogni frame).



**Figura 4.2:** Finestra Ispezione di Unity

I diversi GameObject presenti nella scena si possono visualizzare nella finestra Gerarchia. Un concetto molto importante alla base della disposizione degli oggetti nella scena è il concetto di imparentamento. Unity utilizza il concetto di gerarchie di imparentamento per raggruppare più GameObject. Un oggetto può contenere altri GameObject che ereditano le sue proprietà (posizione, scalamento e rotazione). Questo implica che manipolando l'oggetto padre, gli oggetti figli si subiranno lo stesso comportamento, ma non il contrario. In questo scenario gli oggetti figli avranno sia una posizione globale rispetto al mondo di Unity, sia una posizione relativa al loro oggetto padre.



**Figura 4.3:** Finestra Gerarchia di Unity

Per creare il file eseguibile di un'applicazione sviluppata su Unity è necessario eseguirne la build, cliccando su File -> Build Settings -> Build.

### 4.2.1 Mixed Reality Toolkit

MRTK è un kit di sviluppo gestito da Microsoft che fornisce un set di componenti e funzionalità che permettono di sviluppare facilmente applicazioni di Realtà Mista multiplatforma su Unity e su Unreal. In particolare, fornisce il sistema di input multiplatforma e i blocchi predefiniti per le interazioni spaziali, abilita la creazione

rapida di prototipi, e da la possibilità agli sviluppatori di sostituire i comportamenti di alcuni prefab con comportamenti definiti dal programmatore. Tra i dispositivi supportati troviamo:

- Microsoft Hololens 1 e 2
- Visori VR di Windows Mixed Reality
- Oculus Quest e Oculus Rift
- Dispositivi mobili iOS e Android

Per poter sviluppare soluzioni di Realtà Aumentata con Hololens 2 è necessario configurare il progetto di Unity per lo sviluppo con Windows Mixed Reality. Innanzitutto il progetto deve essere impostato per l'esportazione come applicazione Universal Windows Platform (UWP). Successivamente è necessario importare nel progetto tutte le features desiderate, tramite un'applicazione esterna, *Mixed Reality Feature Tool for Unity*, che contiene una vasta scelta di toolkit installabili nel progetto Unity (la sua interfaccia si può vedere in figura 4.4). Per il progetto sono stati utilizzati:

- Mixed Reality Toolkit Foundation: un set di componenti e features per accelerare lo sviluppo di applicazioni di Mixed Reality.
- Mixed Reality Toolkit Standard Assets: include shaders, materiali e font.
- Mixed Reality Toolkit Examples: insieme di scene e asset da cui prendere spunto per capire le diverse funzionalità presenti.

La versione utilizzata è la 2.72. Una volta importati i pacchetti su Unity, è necessario configurare il progetto mediante il *MRTK Project Configurator*. In questo caso, per far funzionare correttamente l'applicazione con Vuforia, bisognerà cliccare su *Build-in Unity plugins (non-OpenXR)* e poi, nelle impostazioni di Unity alla voce XR Plug-in Management, bisognerà checkare l'opzione *Windows Mixed Reality*. E' necessario utilizzare il Plug-in di windows e non OpenXR perchè l'utilizzo di Vuforia con quest'ultimo causa un tracking impreciso e traballante. Una volta completati questi passaggi, è possibile iniziare lo sviluppo di applicazioni di realtà mista semplicemente inserendo nella scena due GameObject:

- *MixedRealityToolkit*. Questo oggetto contiene uno script omonimo, che da la possibilità all'utente di personalizzare i profili che costituiscono le informazioni di configurazione per l'inizializzazione dei sottoinsiemi e delle funzionalità di MRTK, come ad esempio impostazioni della fotocamera (qualità, trasparenza), o del sistema di input da utilizzare, o della consapevolezza spaziale (il dispositivo può fare un'analisi dell'ambiente circostante, trasformandolo in mesh, per poter gestire le occlusioni)

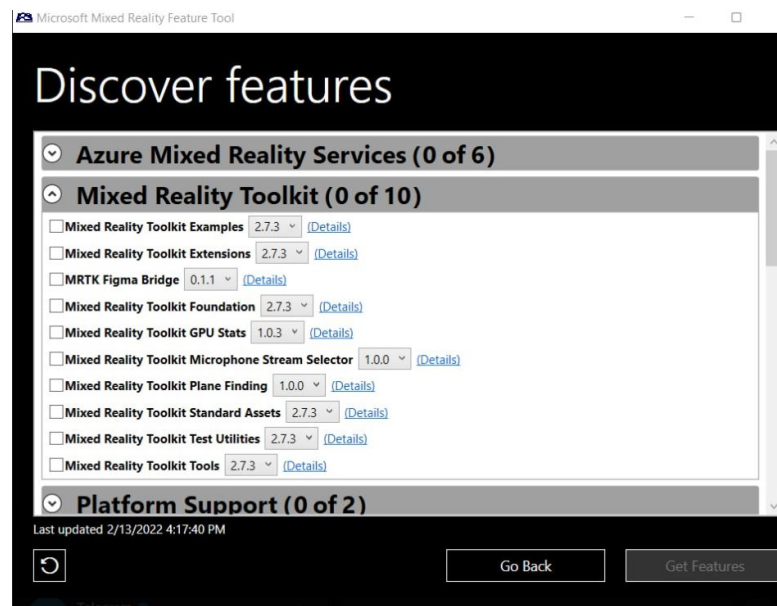


Figura 4.4: Mixed Reality Feature Tool

- *MixedRealityPlaySpace*, oggetto che ha come figlio la camera principale che è dotata del componente *MixedRealityInputModule*, necessario per far funzionare il progetto in ambiente XR.

### 4.2.2 Vuforia Engine

Vuforia è un kit di sviluppo software di realtà aumentata sia per dispositivi mobile sia per Hololens 2 e Magic Leap. E' possibile integrare su Unity le sue funzionalità importando il pacchetto `.unitypackage` (pacchetto importabile in Unity, che ha tutte le funzioni necessarie per lo sviluppo di un'app in Realtà Aumentata), scaricabile dall'Asset Store di Unity. Successivamente, per permettere all'engine di funzionare, è necessario aggiungere il componente *VuforiaBehavior* alla camera. Per abilitare il suo funzionamento bisogna configurare Vuforia inserendo la chiave di licenza che si ottiene registrandosi nel Portale degli Sviluppatori sul sito. La scelta di Vuforia è dovuta al fatto che è di facile utilizzo, integrabile facilmente con Unity e gratuita (se pur con qualche limitazione) se utilizzata a livello personale. La versione utilizzata per il progetto è stata la 10.2.5.

Lo scopo dell'engine di Vuforia ai fini del progetto è stato sfruttare la sua capacità di tracciamento degli oggetti attraverso il sistema di Model Target Tracking. Questa funzionalità abilita l'applicazione ad utilizzare l'engine di Vuforia per riconoscere e fare il tracking di oggetti presenti nel mondo reale, basandosi su un modello 3D in scala reale dell'oggetto. [22]

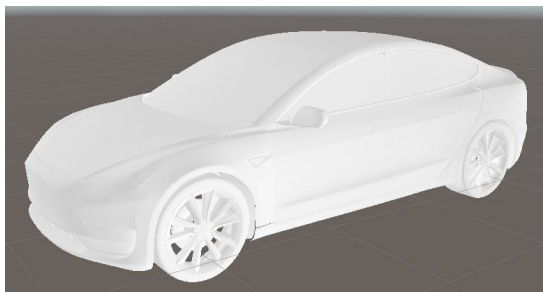
Per permettere al motore di riconoscere l'oggetto presente nell'ambiente, è necessario generare un Model Target mediante un'applicazione, *Model Target Generator*. Questa applicazione converte un modello 3D in un database di Vuforia Engine che può essere utilizzato per eseguire il tracking del modello. Il Model Target Generator può leggere una varietà di formati 3D, che verranno poi convertiti nel formato di mesh proprietario di Vuforia. L'engine funziona meglio nei modelli CAD con caratteristiche particolari, come ad esempio una corrispondenza di colori con l'oggetto fisico. A tal fine verranno elencate le migliori pratiche da adottare se si vuole utilizzare il Model Tracking: [23]

- Posizione dell'oggetto nello spazio stazionaria. Il sistema di tracciamento lavora meglio se l'oggetto è statico e non si muove dopo essere stato rilevato. L'utente può muoversi intorno all'oggetto ma idealmente l'oggetto stesso non dovrebbe muoversi.
- Gli oggetti devono presentano diverse colorazioni. Gli oggetti composti da un unico colore uniforme sono difficili da tracciare, a maggior ragione se sono posizionati in uno sfondo con la stessa colorazione.
- Oggetti molto riflettenti, neri o trasparenti rendono il sistema di tracciamento difficile perchè non forniscono dettagli di superficie.
- I dettagli geometrici dell'oggetto devono essere complessi, con caratteristiche particolari, per rendere l'oggetto facilmente distinguibile da altri oggetti presenti nella stanza. Le figure semplici, come cubi o sfere, si possono facilmente confondere con altri elementi presenti nella stanza.
- Gli oggetti simmetrici o con parti ripetute possono causare confusione per il rilevamento e il tracciamento.
- Gli oggetti devono essere rigidi e non flessibili. Se ci sono parti flessibili che potrebbero non essere presenti nell'oggetto fisico, o parti che si muovono, è necessario rimuovere dal modello CAD, altrimenti il sistema di tracciamento potrebbe fallire.
- L'oggetto 3D deve avere la stessa forma e dimensione del modello reale. E' bene assicurarsi quindi che il modello CAD abbia la dimensione in metri più accurata possibile rispetto all'oggetto originale, il massimo errore tollerato dal motore di Vuforia è una discrepanza del 10%.
- Il modello 3D da utilizzare per il training nell'applicazione deve avere meno di 400.000 poligoni e/o meno di 20 parti distinte, per poter consentire all'applicazione di funzionare in real-time.

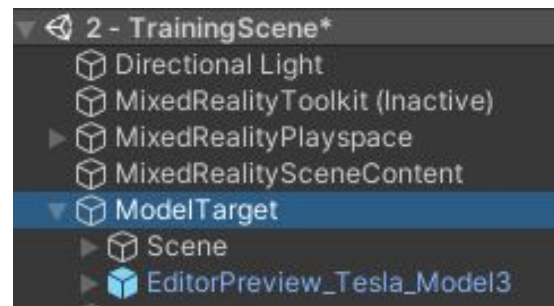


- Le parti che possono avere una posizione variabile (ad esempio la rotazione di una ruota di un'auto) devono essere rimosse perchè altrimenti interferirebbero con il sistema di tracciamento.
- Le parti interne del modello 3D che non vengono visualizzate dall'esterno possono essere rimosse perchè irrilevanti.

Dopo aver verificato che i requisiti appena elencati sono rispettati, si può procedere con l'inserimento del modello 3D all'interno del Model Target Generator. Ulteriori dettagli su questa procedura verranno approfonditi in 5.4.3. Una volta terminato il processo, il sistema darà come output un file di tipo .unitypackage. Sarà necessario importarlo nel progetto Unity per poterlo utilizzare. Infine, nel menù di Unity, bisognerà selezionare la voce GameObject -> Vuforia Engine -> Model Target. Questa opzione permette di visualizzare il modello 3D prodotto in output dal processo di Model Target Generator, e creerà nella finestra di Gerarchia un GameObject chiamato ModelTarget. Tutti gli oggetti figli di questo oggetto avranno una posizione nello spazio reale relativa alla posizione del padre.



**Figura 4.5:** Output del Model Target Generator



**Figura 4.6:** Finestra di Gerarchia del Model Target

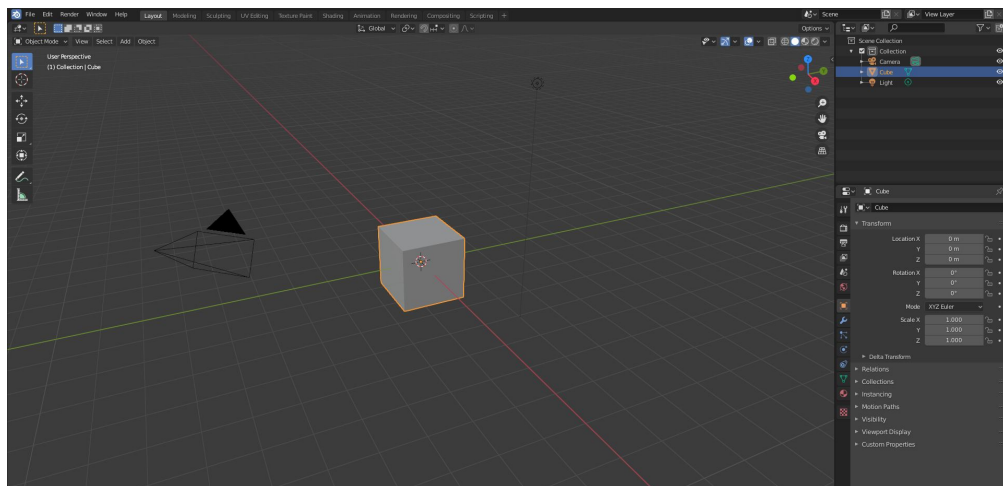
### 4.3 Visual Studio

Microsoft Visual Studio è un ambiente di sviluppo integrato sviluppato da Microsoft. Supporta diversi linguaggi di programmazione, tra cui C#, C++ e Java. Ai fini del progetto, Visual Studio è servito sia per definire i comportamenti mediante scripting dei GameObject di Unity con il linguaggio di programmazione C#, sia per poter distribuire un'applicazione Universal Windows Platform sulle Hololens. La versione utilizzata per questo progetto è stata la versione 2019. Per distribuire applicazioni sulle Hololens 2 è necessario prima di tutto abilitare l'opzione *Modalità Sviluppatore* sia sul PC Desktop Windows sia sul dispositivo Hololens 2. Successivamente, è necessario aprire il file .sln, che si trova nella cartella dell'output prodotto da Unity

dopo aver premuto "Build". Una volta aperto, per poter installare l'eseguibile sul caschetto si deve scegliere l'opzione di configurazione ARM64, e si seleziona Dispositivo o Macchina Remota a seconda se il dispositivo Hololens è collegato al PC con un cavo o se si vuol far avvenire la distribuzione attraverso la rete Wi-Fi. Infine il progetto viene compilato e distribuito sul caschetto. Il processo ha una durata dell'ordine di qualche minuto.

## 4.4 Blender

Blender è un software gratuito open source multiplatforma, per cui può girare sui sistemi operativi Linux, Windows e Macintosh. La sua interfaccia utilizza OpenGL. La potenzialità del programma è data dal fatto che fornisce tutti gli strumenti necessari per modellazione, l'animazione, il compositing e il rendering 3D, ma anche le API per lo scripting con Python sia per poter personalizzare l'applicazione sia per creare tool specializzati. Essendo inoltre un software gratuito, sul web ci sono intere community da cui poter imparare facilmente a utilizzare i suoi strumenti.



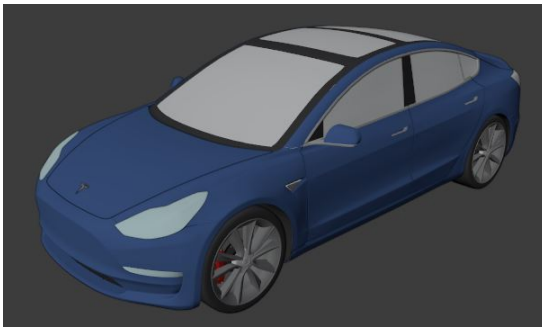
**Figura 4.7:** Interfaccia di Blender

Gli strumenti forniti da Blender sono stati fondamentali per ridurre la complessità dei modelli 3D forniti. Ai fini dell'applicazione, era infatti necessario avere:

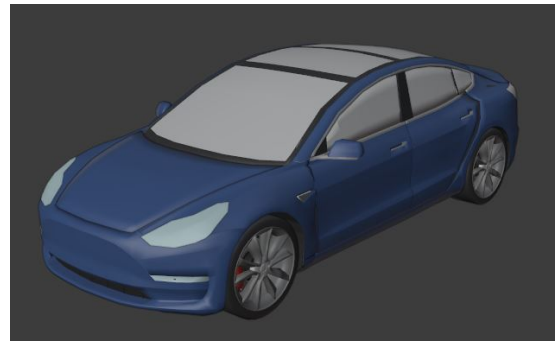
- Un modello 3D di Tesla Model 3 [24] fedele all'originale, sia per forme che per dimensioni, per poter fare Model Tracking nella maniera più accurata possibile.
- Un modello 3D del telaio dell'auto [25], sempre in scala reale, con tutte le componenti elettriche. Serve sia per creare una mappatura tra modello 3D

e modello reale per definire correttamente i punti in cui devono avvenire le diverse operazioni della procedura, sia per dare la possibilità all'utente fruitore dell'applicazione di visualizzare le parti nascoste dell'auto e avere una visione più chiara dell'insieme.

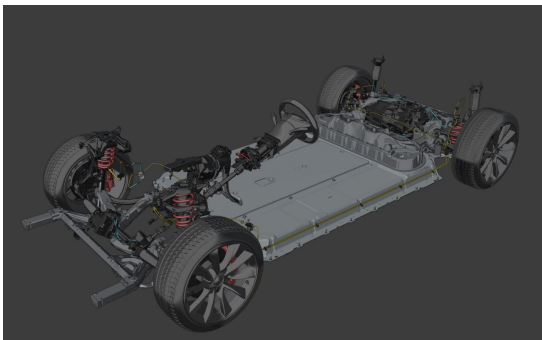
I modelli, entrambi acquistati su TurboSquid, erano estremamente complessi e l'applicazione su Hololens non sarebbe riuscita a mantenere un framerate adeguato. Per questo motivo è stato necessario semplificare le geometrie, talvolta togliendo anche alcune parti non necessarie. Nello specifico, il modello della Tesla presentava 475.866 vertici, mentre il modello del telaio ne aveva 8.921.182, chiaramente un numero troppo elevato. Dopo la semplificazione, si è arrivati relativamente a 92.319 e 436.857 vertici. La versione di Blender utilizzata per questo scopo è stata la 2.83.



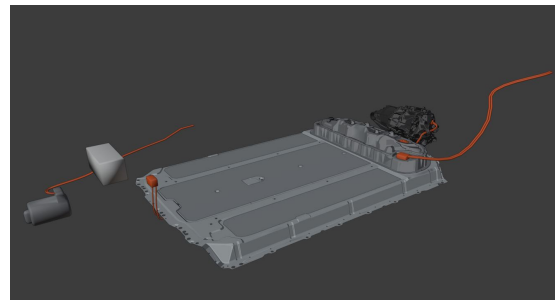
**Figura 4.8:** Tesla prima della semplificazione



**Figura 4.9:** Tesla dopo la semplificazione



**Figura 4.10:** Telaio prima della semplificazione



**Figura 4.11:** Telaio dopo la semplificazione

Fortunatamente, il modello 3D della Tesla era stato modellato in Blender, e non erano stati ancora applicati alcuni modificatori, tra cui il *Subdivision Surface* (che divide le facce di una mesh in facce più piccole, conferendole un aspetto uniforme ma al contempo aumentando il numero di vertici). Di fatto quindi eliminando questo modificatore, si arrivava già a un risultato apprezzabile. Per ridurre il numero di vertici generalmente è stato utilizzando il tool Clean Up, il Decimate Modifier e in alcuni casi è stato fatto un lavoro manuale per ridurre al minimo ogni geometria.

**Clean Up** Questo strumento si trova selezionando, in edit mode (ovvero la modalità che permette di manipolare facce, linee e vertici), il menù Mesh -> Clean Up. Nello specifico è stato usato:

- **Delete Loose**, per rimuovere i vertici e le linee sconnesse dal resto della geometria
- **Degenerate Dissolve**, per rimuovere le geometrie che non servono, come le linee senza lunghezza o le facce senza area.
- **Decimate Geometry**, per ridurre il numero di vertici o il numero di facce di una mesh cercando di minimizzare i cambiamenti della forma originale. Con questo tool era possibile definire un parametro, *Ratio*, che indica la proporzione dei triangoli da ridurre. Il valore inserito variava a seconda della mesh selezionata.
- **Merge By Distance**. Unisce vertici che sono vicini di una certa distanza definibile dall'utente nel parametro *Merge Distance*

**Decimate Modifier** Questo strumento permette di ridurre il numero di vertici o di facce di una mesh con la minima modifica. Questo modifier ha tre diverse modalità, tutte utilizzate a seconda della tipologia delle geometrie e con parametri variabili a seconda del modello:

- **Collapse**, che unisce insieme e progressivamente i vertici.
- **Unsubdivide**, che rimuove i lati risultanti di un'operazione di *Subdivide*.
- **Planar**, che riduce i dettagli su forme composte principalmente da superfici piane.

**Intervento manuale** Anche dopo le semplificazioni sopra citate, il modello risultante era comunque troppo complesso. Per questo motivo è stato necessario intervenire manualmente rimuovendo alcuni vertici che non davano particolare valore

alla geometria e, in alcuni casi, anche rimodellando alcune parti. Soprattutto per quanto riguarda il modello del Telaio, sono state eliminate sia tutte le componenti che non servivano per la procedura, sia tutti gli spessori dei vari modelli, forte del fatto che questi non verranno visualizzati.

# Capitolo 5

## Design e sviluppo

### 5.1 Descrizione dell'applicativo

Lo scopo della tesi è quello di sviluppare una soluzione di Realtà Aumentata a supporto degli operatori che devono disconnettere i sistemi ad alta tensione per rendere sicuro un veicolo elettrico (nel caso specifico una Tesla Model 3) prima di qualsiasi riparazione. La procedura da seguire è fornita direttamente da Tesla, e prende il nome di *Body Repair Tech Note: Disconnecting 12V and High Voltage Power*. Può essere eseguita soltanto da personale qualificato e certificato con una conoscenza dei motori del veicolo e delle componenti elettriche.



**Figura 5.1:** Posizione delle componenti elettriche

Nell'opuscolo, come si vede in figura 5.1, è anche mostrata la posizione delle componenti di interesse per la riparazione. Uno degli obiettivi dell'applicazione

AR è mostrare queste informazioni direttamente sul veicolo mediante ologrammi tridimensionali, così che l'operatore possa visualizzare e comprendere meglio la posizione delle diverse componenti. Per permetterlo, è necessario un sistema di Model tracking, ovvero un sistema che riesca a riconoscere la posizione dell'auto nel mondo reale per poi sovrapporci in maniera appropriata le informazioni digitali.

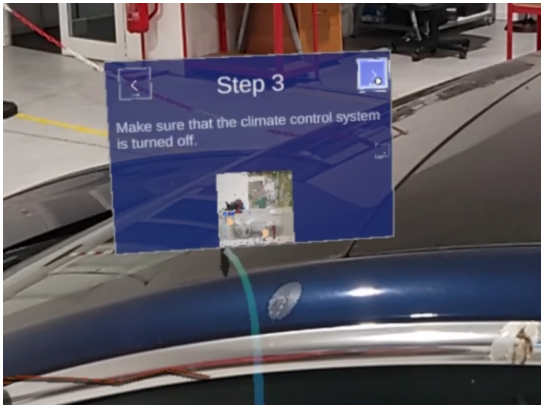


**Figura 5.2:** Posizione delle componenti elettriche in AR

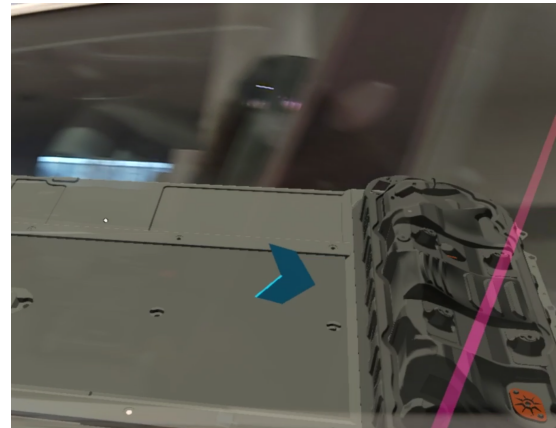
Per insegnare all'operatore come interagire con il sistema è stata una scena di tutorial iniziale, suddiviso in step, ognuno dei quali simula uno degli eventi associati agli step nella procedura. Questa necessità è dovuta al fatto che l'utilizzo del dispositivo Hololens non è ancora largamente diffuso e, di conseguenza, il suo metodo di utilizzo per chi si avvicina a questa tecnologia per la prima volta non è subito chiaro. Il tutorial può essere anche saltato.

Ogni istruzione presente nella procedura indica un'operazione da eseguire in un punto specifico dell'auto. Nell'applicazione AR l'operatore visualizzerà l'istruzione direttamente nel punto in cui dovrà avvenire l'operazione. A guidarlo sarà una curva spline che collegherà la posizione dell'utente alla posizione finale (figura 5.3), e un indicatore che suggerisce all'utente la direzione verso cui spostare la testa (figura 5.4). Le istruzioni visualizzate in Realtà Aumentata sono in formato testuale, con eventuali allegati multimediali (immagini o video) che aiutano ulteriormente l'operatore a comprendere l'azione da svolgere.



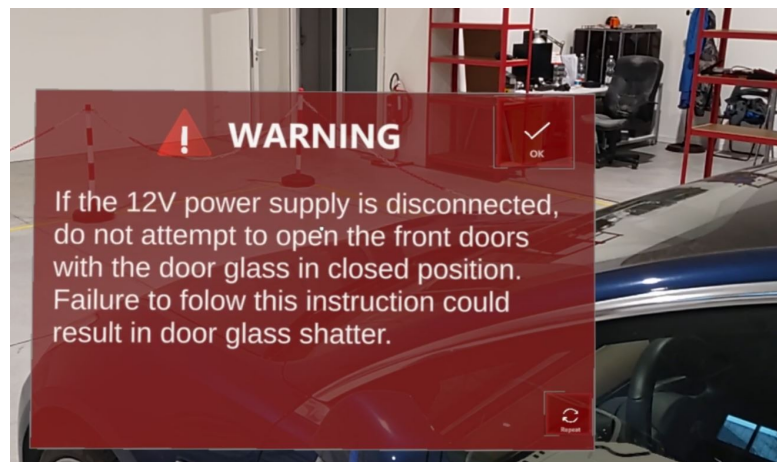


**Figura 5.3:** Curva spline di giunzione tra utente e pannello dello step



**Figura 5.4:** Indicatore

E' possibile che alcuni step contengano dei Warning: sono finestre caratterizzate da un colore rosso scuro che compariranno di fronte allo sguardo dell'utente con lo scopo di avvertirlo per evitare che esegua operazioni che potrebbero compromettere la sua sicurezza (come si vede in figura 5.5).



**Figura 5.5:** Pannello di Warning

Quando uno step prevede una misurazione (ad esempio la misura della tensione elettrica), l'utente ha la possibilità di inserire i valori indicati dal multimetro in una sezione apposita del pannello di istruzioni (come si vede in figura 5.6. Alla fine della procedura, i valori inseriti nel sistema potranno essere visualizzati su un'applicazione desktop. Questa feature è stata richiesta poichè, essendo questa un'applicazione a supporto della formazione, la visualizzazione di report da parte



del supervisore è una caratteristica importante per la valutazione dell'attività di addestramento.



**Figura 5.6:** Utilizzo della tastiera AR nell'applicazione

L'applicazione Hololens è infatti collegata a un'applicazione Desktop e permette anche al supervisore di modificare o addirittura cambiare la procedura visualizzata. L'inserimento di questa feature ha più motivazioni:

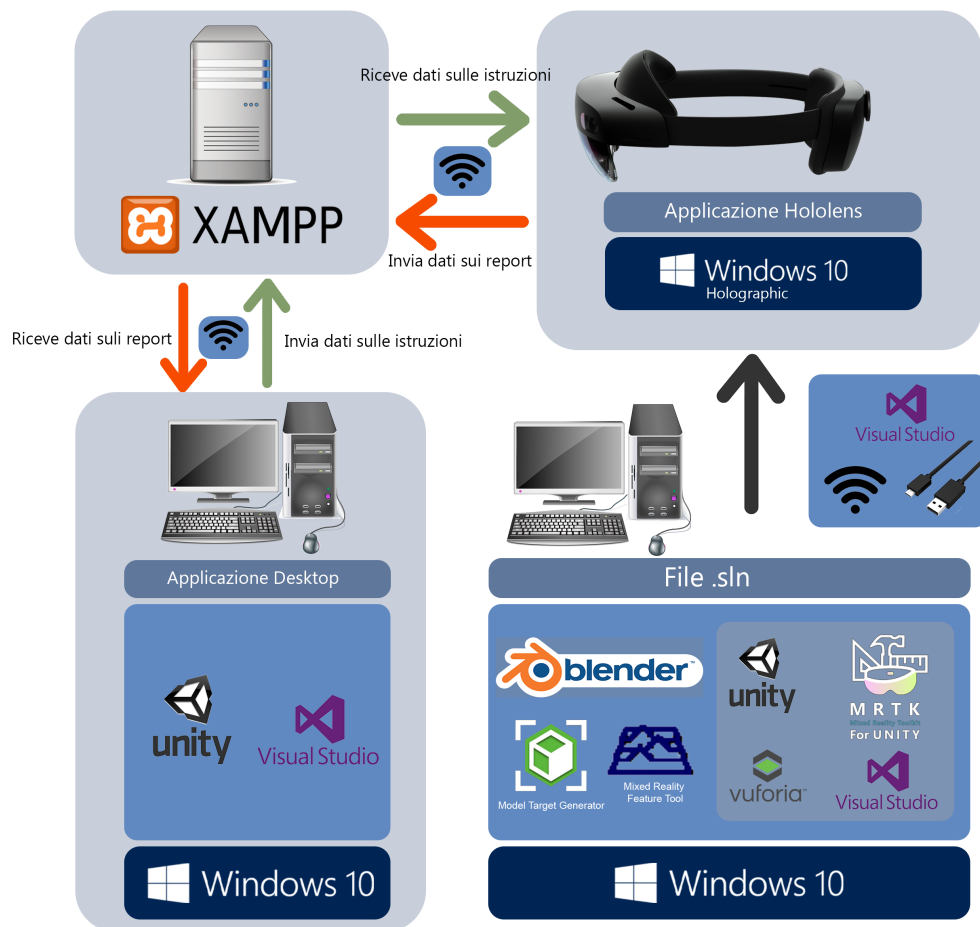
- Alcuni step contengono informazioni sul eventuali attrezzature da usare. Queste attrezzature nel corso degli anni potrebbero cambiare o potrebbero semplicemente essercene di nuove da utilizzare. Il supervisore ha la possibilità di modificare il testo dell'istruzione con la versione aggiornata dell'attrezzatura da utilizzare. In questo modo l'utente non dovrà sfogliare un secondo manuale la versione aggiornata dell'attrezzatura.
- Rende l'applicazione AR scalabile. Se esistono altre procedure da effettuare sullo stesso modello di macchina, l'applicazione desktop ha la potenzialità di permetter al supervisore di creare nuove procedure, senza necessità che lui abbia conoscenze di sviluppo di applicazioni AR.
- Come detto precedentemente, contiene una sezione per poter visualizzare i report delle misurazioni degli operatori, in modo semplice e intuitivo.

## 5.2 Architettura

Al fine di comprendere meglio la comunicazione tra le applicazioni, in figura 5.7 viene mostrato un piccolo schema illustrativo che mostra sia gli strumenti di

sviluppo (di cui è stato abbondantemente discusso nel capitolo 4) sia i paradigmi di comunicazione tra le due applicazioni. Lo scambio di dati avviene mediante un'architettura di tipo client-server. In particolare:

- Il client Desktop deposita (e riceve) i dati sulle istruzioni aggiornate sul server, il client Hololens le riceve e salva su disco
- Il client Hololens deposita le informazioni dei report sul server, il client desktop le riceve.



**Figura 5.7:** Architettura del sistema

Il web server utilizzato per lo scopo è stato XAMPP. L'obiettivo di XAMPP è quello di fornire agli sviluppatori una distribuzione di facile installazione per entrare nel mondo di Apache, con tutte le funzioni di XAMPP già attive e pronte all'uso.

### 5.2.1 Comunicazione e sincronizzazione delle app

Lo scambio di dati tra le due entità avviene mediante l'ausilio di file testuali. Le istruzioni sono organizzate in una struttura dati di tipo JSON. In informatica, nell'ambito della programmazione web, JSON, acronimo di JavaScript Object Notation, è un formato adatto all'interscambio di dati fra applicazioni client/server. In particolare, per ogni step, la struttura dati è organizzata in questo modo:

- **n\_step**: valore intero, indica il numero di step a cui fa riferimento l'istruzione
- **description**: stringa che indica la descrizione dell'istruzione
- **is\_a\_warning**: valore booleano. Se true indica che l'istruzione è associata a uno o più warning.
- **warningDescription[]**: è un array di stringhe. Vuoto se il valore **is\_a\_warning** è false, altrimenti conterrà da 1 fino a un massimo di 10 elementi, ognuno dei quali indicherà la descrizione del warning. Il limite a 10 è stato imposto per evitare di sovraccaricare le informazioni percepite dall'utente.
- **img\_video\_name[]**: è un array di stringhe che contiene il nome e l'estensione del contenuto multimediale associato all'istruzione (ad esempio: 0\_1.jpg). Il numero di elementi può variare da 0 a 3. Il limite a 3 è stato imposto perchè più di 3 contenuti multimediali in simultanea non sarebbero stati visualizzati correttamente nell'applicazione AR.
- **isInputData**: è un booleano. Se true, l'istruzione conterrà una sezione apposita per permettere all'utente di inserire, mediante tastiera AR, i valori richiesti in **description** (come ad esempio il valore misurato della tensione o dell'isolamento).
- **position**: è un intero. Indica il punto in cui deve essere mostrata l'istruzione. L'approccio adottato per indicare i diversi punti dell'auto è stato definire dei punti di interesse sull'automobile e mapparli con un numero intero.

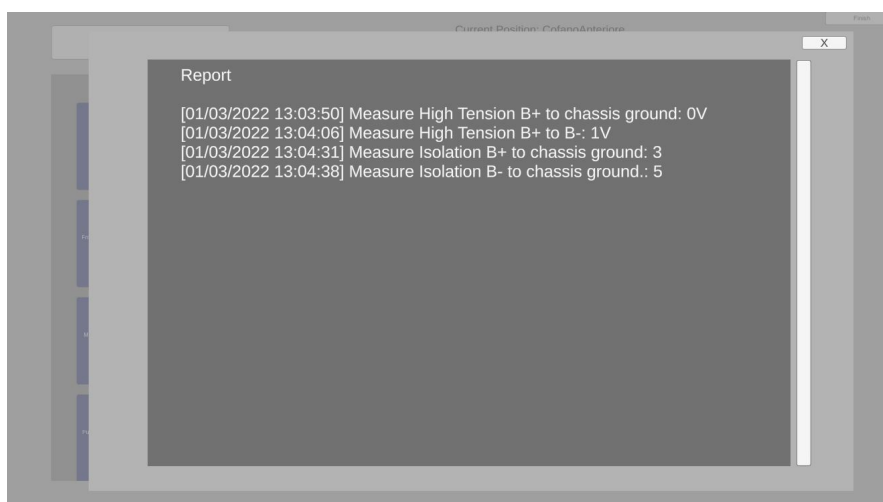
```
"n_step":18,  
"description":"Take the multimeter with  
"is_a_warning":true,  
"warningDescription":["Use a multimeter  
"Use test probes  
],  
"img_video_name":["18_1.jpg"],  
"isInputData":false,  
"position":9
```

Figura 5.8: Organizzazione del file JSON

Il file, prodotto dall'applicazione desktop, verrà trasmesso sul server con il nome di *JsonText.txt*, insieme a tutte le immagini relative a ogni istruzione, con la giusta nomenclatura definita da *img\_video\_name*. Ogni volta che da desktop verrà caricata una versione aggiornata della procedura, verrà anche aggiornato un altro file su server, chiamato *Version.txt*. Al suo interno si trova un numero che viene essere incrementato a ogni aggiornamento.

Lato Hololens, verrà confrontato il numero di versione contenuto nel file sul server con il numero di versione memorizzato su disco. Qualora i due non dovessero coincidere, all'avvio dell'applicazione verrà visualizzato un pulsante con il quale si può interagire per poter scaricare dal server le nuove immagini e il file JSON aggiornato in modo tale da poter essere salvati su disco; se al contrario coincidono il pulsante non apparirà e si utilizzeranno i file già presenti su disco. In questo modo, l'applicazione può essere utilizzata anche in mancanza di una connessione a Internet con l'unica limitazione che non è possibile visualizzare da desktop i report inseriti dall'utente.

I file di report invece sono semplici file testuali descrittivi che contengono l'output inserito dall'utente in quelle istruzioni che hanno il valore *isInputData* settato a true. Raccolgono uno storico dei dati inseriti dai diversi utenti sull'applicazione Hololens nel corso del tempo. I nuovi dati inseriti, alla fine della procedura, aggiornano un file sul server che prende il nome di *Report.txt*. E' possibile visualizzare quanto scritto cliccando sulla sezione Report nell'applicazione desktop.



**Figura 5.9:** Sezione Report dell'app Desktop

## 5.2.2 Scambio di dati tra server e client

Per poter scaricare o caricare i dati dal server, è necessario in primo luogo che client e server siano collegati alla stessa rete Wi-Fi. Questo perchè l'applicazione è stata testata su rete locale, di conseguenza, le richieste di GET / POST HTTP mandate da applicazione a server necessitano l'inserimento dell'indirizzo IPv4 privato del server, non accessibile da rete esterna. In secondo luogo il server XAMPP deve aver attivato il modulo Apache. All'interno della cartella htdocs di XAMPP, è stata creata un'altra cartella specifica per la comunicazione di queste applicazioni, ed è stata chiamata *unity*, in cui è presente il file *index.php* il cui codice gestisce il processo di upload dei file su server, che verranno caricati nella sottocartella *uploaded\_images*. Per le richieste web Unity mette a disposizione le API di UnityNetworking, in particolare UnityWebRequest.

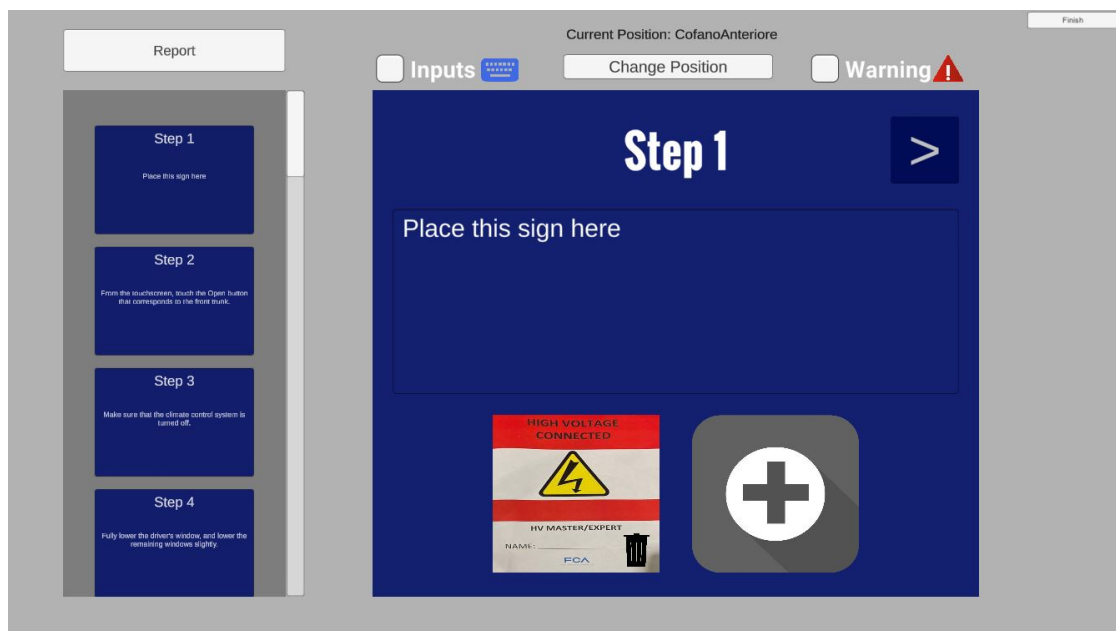
- Per scaricare i dati è necessario iniziare una Coroutine in cui si faccia una UnityWebRequest di tipo GET, inserendo come url l'indirizzo IPv4 del server, seguito da *unity/uploaded\_images/* a sua volta seguito dal nome del file (ad esempio *JsonFile.txt*). UnityWebRequest fornisce anche delle funzioni che permettono di gestire gli errori di rete, come *ConnectionError* o *ProtocolError*.
- Per caricare i dati da disco a server è necessario iniziare una Coroutine e salvare in un array di tipo byte i dati del file da caricare, che verranno successivamente aggiunti all'oggetto di tipo *WWWForm* insieme al nome che si vuole dare al file. Successivamente si fa una UnityWebRequest di tipo POST, inserendo come parametri l'indirizzo URL del server (in questo caso composto da indirizzo IPv4 del server seguito da */unity/*) e il nome della variabile dell'oggetto di tipo *WWWForm*.

## 5.3 Applicazione desktop

Questa sezione contiene una panoramica circa l'interfaccia e le funzionalità contenute nell'applicazione Desktop.

### 5.3.1 Interfaccia e interazioni

L'interfaccia dell'applicazione desktop mira a essere quanto più intuitiva possibile, con le interazioni essenziali per permettere di costruire la struttura dati in JSON senza che il supervisore che costruirà la procedura sia esperto di informatica. Come si può vedere in figura 5.10 l'organizzazione degli elementi ricorda quella di un Power Point. La procedura è suddivisa in diversi step. Ogni step è composto testo ed eventuali contenuti multimediali (immagini, video) e deve essere associato a una



**Figura 5.10:** Interfaccia dell'applicazione desktop

posizione nell'auto; può essere uno step in cui si richiede all'utente di inserire dei dati misurati, e può avere dei warning a lui associati, che contengono informazioni relative alla sicurezza personale.

**Navigazione** L'utente può navigare tra i vari step sia mediante l'ausilio delle frecce poste ai lati superiori del pannello blu che permettono una navigazione lineare, sia accedendo direttamente allo step da lui desiderato scrollando la finestra posta sulla sinistra e cliccando sull'icona blu relativa. A raccogliere tutti i componenti del pannello blu degli step è il Prefab nominato **BluPanel**, il prefab **Button** invece indica l'icona blu che si trova nella finestra di scrolling.

**Modifica** Una volta scelto lo step in cui apportare le modifiche, l'utente ha a disposizione diversi strumenti per poterlo modificare:

- Può cambiare il testo, semplicemente scrivendo nell'apposito campo di Input.
- Può modificare, eliminare o inserire nuove immagini o video (fino a un massimo di 3 per step). Se vuole aggiungerne può cliccare sull'icona con il segno "+". Per sostituire, basterà cliccare sopra all'immagine da sostituire. Per cancellare basterà premere l'icona del cestino associata al contenuto multimediale. La selezione del contenuto avviene mediante File Explorer che consentirà di accedere ai file presenti su disco.

- Può aggiungere o modificare o eliminare i warning. Se vuole inserire un warning associato allo step l'utente dovrà spuntare la checkbox affiancata alla scritta Warning. A questo punto comparirà un altro pannello rosso sovrapposto all'interfaccia con un campo di Input in cui l'utente può scrivere la descrizione (l'interfaccia si può visualizzare in figura 5.11). Contemporaneamente verrà creato a destra un pulsante nella scroll list, seguendo la stessa logica dei pannelli blu, per permettere all'utente di scegliere se navigare tra i warning in maniera incrementale con le frecce poste sul pannello rosso o se accedere ad un warning specifico semplicemente cliccandoci sopra nella lista di pulsanti. Se vuole aggiungere un nuovo warning, si deve prima cliccare sull'ultimo pannello warning inserito e poi sulla freccia a destra, che creerà una nuova istanza sia di pannello rosso sia di pulsante in lista. Se si vuole eliminare un warning basta cliccare sull'icona del cestino a lui associata nella lista di pulsanti. Se invece non si vuole più associare l'istruzione a un warning, basterà uncheckare la checkbox e automaticamente spariranno i pannelli.
- Può decidere la posizione nell'auto in cui quell'istruzione verrà visualizzata. Per farlo dovrà cliccare sul pulsante che prende il nome di Select Position. A questo punto si aprirà un'interfaccia che presenta una lista di immagini, ognuna relativa a una precisa posizione dell'auto. (figura 5.12). Basterà cliccare sull'immagine che rappresenta il punto di interesse per definire la nuova posizione dell'istruzione.
- Può decidere che in quello step l'utente dovrà inserire un valore misurato, semplicemente spuntando la checkbox affiancata da Input.

**Visualizzazione dei report** Per visualizzare il file di Report, l'utente dovrà cliccare sul pulsante Report in alto a sinistra. Si aprirà una finestra scrollabile che mostrerà in formato testuale lo storico dei dati inseriti dagli utenti nel corso dell'applicazione (figura 5.9)

**Caricare i dati su server** Una volta che l'utente è sicuro delle modifiche, dovrà premere il pulsante in alto a sinistra con la scritta Finish, che si occuperà di caricare il file JSON, e le immagini sul server.

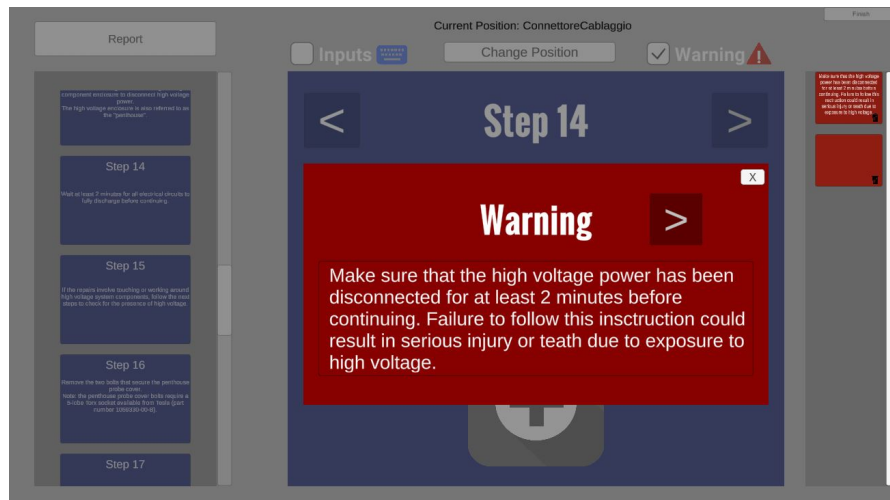


Figura 5.11: Interfaccia dei warning

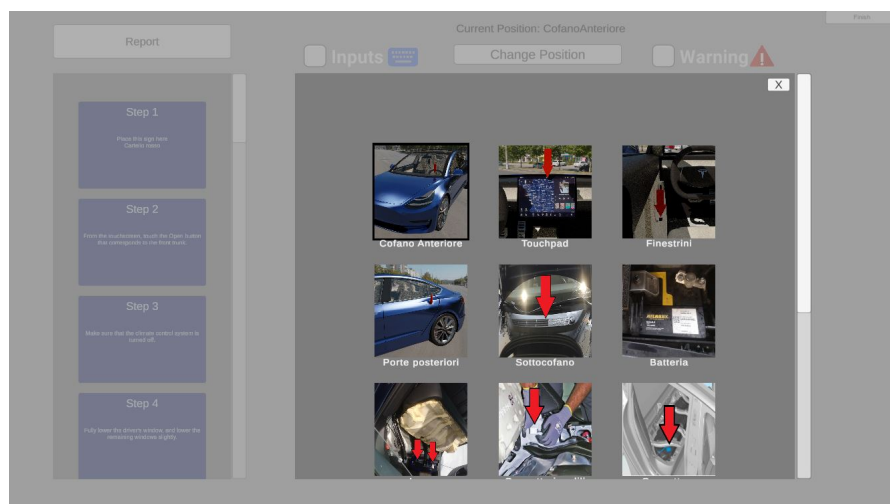
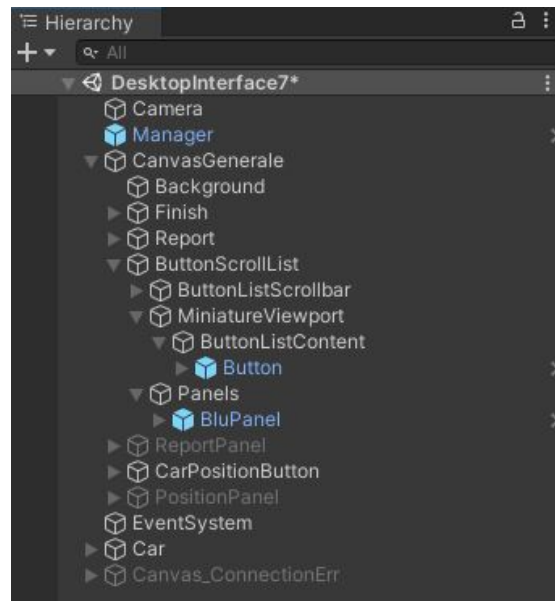


Figura 5.12: Interfaccia per selezionare i punti dell'auto

### 5.3.2 Struttura interna

Entriamo ora nel merito delle funzionalità dell'applicazione, approfondendo la pipeline di esecuzione da quando l'utente apre l'applicazione a quando la chiude. In figura 5.13 viene illustrata l'organizzazione dei diversi elementi della scena di Unity. Il GameObject CanvasGenerale raccoglie tutti gli elementi visualizzabili dell'interfaccia, di cui adesso entreremo nel dettaglio.





**Figura 5.13:** Struttura in Unity della gerarchia

**Manager** A gestire la maggior parte delle funzionalità c'è il GameObject nominato Manager, che ha come componenti le seguenti classi:

- **JSON Writer:** Contiene i metodi per scaricare i file dal server, per inizializzare l'interfaccia dell'applicazione al suo avvio, per aggiornare le variabili in cui sono contenute le informazioni da caricare sul server, e per caricare sia il file JSON sia il file di versione sul server.
- **File Explorer:** Contiene i metodi per poter accedere ai file del disco (con un'interfaccia di File Explorer di Windows). Una volta selezionato il contenuto multimediale, consente la sua visualizzazione nella sezione delle immagini dello step e viene memorizzato in memoria, per poterlo caricare sul server con l'appropriata nomenclatura associata.
- **Pressed Button:** Gestisce l'interfaccia per quanto riguarda la navigazione tra i vari step, contiene i metodi che richiamano i metodi di JSON Writer per aggiornare i dati inseriti dall'utente.
- **Show Output Files:** si occupa di scaricare dal server il file di Report e di gestire l'interfaccia di visualizzazione quando viene premuto il pulsante Report.
- **Positions Manager:** gestisce l'interfaccia per la visualizzazione del panel dei diversi punti di interesse dell'automobile. Si occupa anche di salvare la posizione dell'auto scelta dall'utente per ogni step.

Le variabili più importanti che verranno utilizzate per poter essere caricate sul server sono principalmente due:

- La variabile di tipo Dizionario **instructions**, definita in JSON Writer, che contiene al proprio interno una lista di coppie chiave-valore, dove la chiave è il numero dello step e il valore è un oggetto di tipo Instructions, una classe che possiede le stesse variabili che devono essere contenute nel file JSON (come spiegato in sezione 5.2.1).
- La variabile di tipo Dizionario **allTextures**, definita in File Explorer, che contiene come chiave il numero dello step e come valore una lista di tipo Multimedia (che può contenere o immagini o video) di massimo 3 elementi, che rappresentano il contenuto multimediale associato allo step.

Come prima azione appena si lancia l'applicazione un metodo della classe JSON Writer scaricherà dal server il file *JsonText.txt*, e popolerà la variabile **instructions** con le informazioni già presenti sul file. In base al numero totale di step scritti sul file, verranno generati tanti GameObject di tipo BluPanel figli del GameObject Panels, e tanti GameObject di tipo Button figli del GameObject ButtonListContent, che rappresentano le miniature dei BluPanel visualizzabili nella finestra di scrolling a sinistra. BluPanel e Button sono due Prefab, ovvero elementi dei GameObject che vengono salvati nel progetto Unity già dotati delle componenti necessarie. Di seguito analizzeremo la funzionalità e le caratteristiche di ogni elemento.

**BluPanel** Questo Prefab, come illustrato in figura 5.14 indica il pannello in cui si possono modificare le diverse istruzioni. In figura 5.15 troviamo l'organizzazione gerarchica del GameObject nella scena di Unity. Al suo interno gli script più importanti sono:

- **Blu Panel Shell**: una raccolta delle referenze di tutti i figli dell'oggetto. Contiene anche i metodi per referenziare lo script Pressed Button e FileExplorer del GameObject Manager, richiamati rispettivamente alla pressione dei tasti Next e Back e alla pressione dei riquadri multimediali. Inoltre, contiene le informazioni riguardanti l'attributo della posizione dell'auto associata all'istruzione.
- **Warning Panel**: Si occupa della gestione dei pannelli di Warning: memorizza la lista delle descrizioni dei warning, ne gestisce l'interfaccia, la creazione, la modifica, l'eliminazione. Mantiene al proprio interno anche una variabile booleana che memorizza se l'istruzione è di tipo Input o meno. Quando si avvia l'applicazione, recupera le informazioni della variabile **instruction** contenuta nello script JSON Writer per poter eventualmente istanziare i pannelli di Warning affianco all'istruzione (con relative descrizioni), e per identificare lo stato della checkbox Input.

- **Read JSON Data:** Si occupa di popolare il panel, quando si avvia l'applicazione, con le informazioni pescate dal file JSON sfruttando la variabile `instructions` contenuta nello script JSON Writer: popola il campo Input inserendo il testo relativo allo step a cui il pannello è associato, scarica i relativi contenuti multimediali da mostrare nel pannello delle immagini.

La pressione dei pulsanti *Next* e *Back* richiama i metodi dello script Pressed Button, che oltre ad aggiornare l'interfaccia mostrando lo step successivo (o precedente) chiamerà anche il metodo `ReadStringInput()` di JSON Writer per poter aggiornare l'i-esimo elemento (dove i indica il numero dello step) della variabile `instruction` in cui vengono aggiornate le vecchie informazioni con quelle nuove raccolte nel panel.

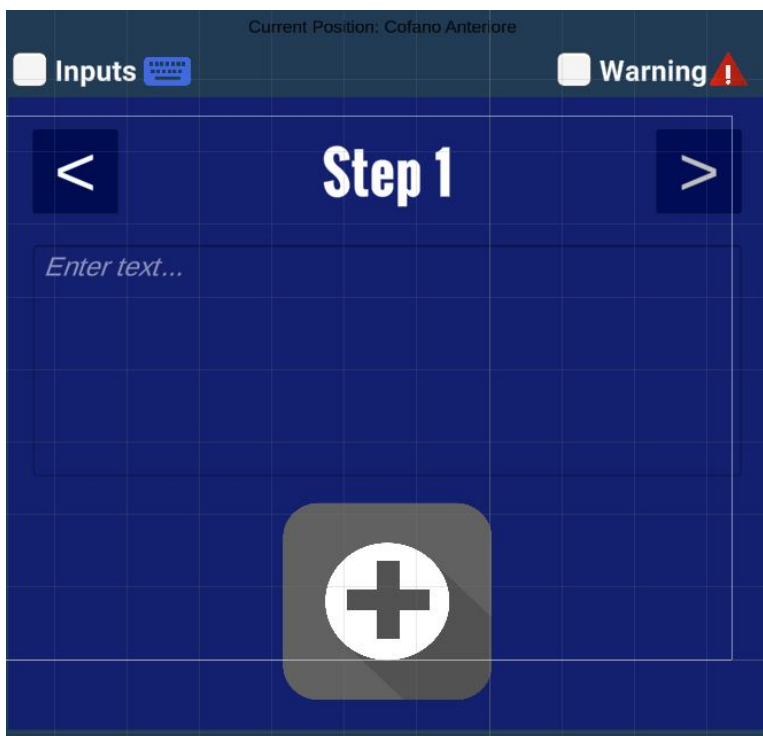


Figura 5.14: Prefab BluPanel

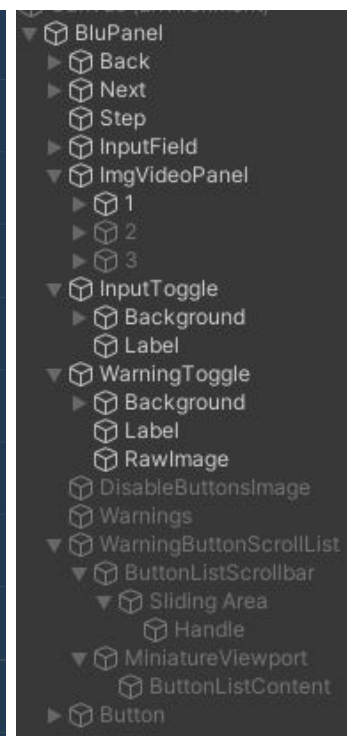


Figura 5.15: Gerarchia del prefab

**Button** Questo prefab vuole essere una rappresentazione in miniatura del Blu-Panel, e contiene soltanto descrizione e numero di step. Permette all'utente una navigazione non lineare tra i diversi step. Premendo su questo GameObject, il comportamento è analogo a quanto succede premendo i pulsanti Back e Next, ossia viene richiamato il metodo di Pressed Button per aggiornare l'interfaccia e

le informazioni di instruction dell'i-esimo elemento dello step. L'unica differenza risiede nel fatto che anzichè mostrare lo step precedente o successivo, viene mostrato lo step del BluPanel associato al Button.



**Figura 5.16:** Prefab Button

**Report** Cliccando su questo pulsante viene richiamato il metodo della classe Show Output Files del GameObject Manager, che si occupa di gestire l'interfaccia per la visualizzazione del contenuto del file di report (come mostrato in 5.9).

**Position** Cliccando sul questo pulsante l'utente visualizzerà l'interfaccia della figura 5.12. In base allo step che sta visualizzando l'utente, la posizione associata allo step verrà evidenziata con un Outline nero. Se si clicca un altro elemento della lista, verrà automaticamente aggiornata la posizione associata allo step nel relativo script BluPanelShell.

**Finish** Cliccando su questo pulsante, verranno richiamati i seguenti metodi:

- `outputJSON()` di JSON Writer: si occupa di trasformare gli elementi della variabile **instructions** in formato JSON.
- `UpdateVersion()` di JSON Writer: si occupa di aggiornare il file di versione sul server per indicare all'applicazione Hololens che c'è stato un aggiornamento
- `UploadImages()` di File Explorer: si occupa di caricare le immagini e i video memorizzati nella variabile **allTextures** sul server.
- `UploadJsonFile()` che si occupa di caricare il file JSON sul server.

Una volta terminato il processo di caricamento, l'utente può chiudere l'app.

## 5.4 Applicazione per Hololens

In questa sezione verrà approfondita la parte di sviluppo dell'applicazione per Hololens.

### 5.4.1 Interfaccia

**Avvio** All'avvio dell'applicazione, se il client non ottiene una risposta dal server visualizzerà il pannello come mostrato in figura 5.17. Questo perchè la mancata risposta del server può avvenire per due motivazioni: o manca la connessione a internet, oppure la rete è cambiata e bisogna reinserire l'indirizzo IP del server da tastiera. In ogni caso, l'utente può scegliere di proseguire senza connessione a internet. Ovviamente, in questo caso, non sarà possibile caricare i file di Report sul server.



**Figura 5.17:** Interfaccia di avvio in assenza di risposta dal server

Successivamente, verrà visualizzata l'interfaccia visualizzata in figura 5.18, che darà la possibilità all'utente di scegliere se imparare a utilizzare le diverse funzionalità dell'applicazione mediante un tutorial o se saltarlo e procedere direttamente con l'avvio. E' inoltre presente un piccolo pulsante in alto a sinistra, che comparirà solamente se c'è stato un aggiornamento lato desktop dei dati inseriti sul server. In questo modo, verranno aggiornati i dati presenti su disco coi nuovi dati scaricati da server.

**Tutorial** (opzionale). Il tutorial è suddiviso in 4 step, ognuno dei quali spiega all'utente le diverse funzionalità che potrà trovare nell'app. Riassumendole:

- Navigazione tra i vari step mediante l'interazione coi pulsanti Back e Next.



**Figura 5.18:** Interfaccia di avvio

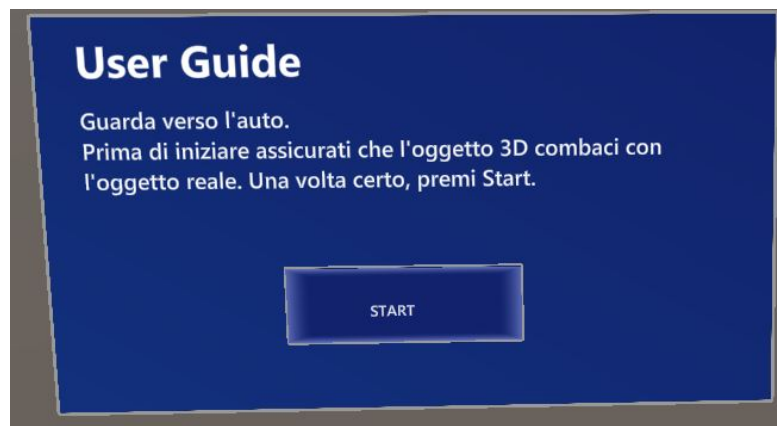
- Illustrazione dell'interfaccia dei Warning: viene spiegato a cosa servono e quali interazioni sono bloccate finchè non c'è stata da parte dell'utente una conferma di lettura.
- Possibilità di avere un'istruzione associata a più punti dell'auto. In questo caso, dall'utente partiranno più curve. La curva blu lo porterà al punto in cui verrà visualizzata l'istruzione, la curva rosa indicherà soltanto gli altri punti di azione (figura 5.19)
- Possibilità di inserire dei valori da tastiera.

Le istruzioni su ciò che l'utente deve fare nel tutorial sono fornite sia in formato testuale sia in formato vocale grazie al TextToSpeech. L'interfaccia inoltre ricalca quella che l'utente visualizzerà nell'applicazione vera e propria.



**Figura 5.19:** Esempio di un pannello di tutorial

**Start** Dopo che l'utente termina il tutorial, o se decide di saltarlo, si troverà nella scena di inizio dell'applicazione vera e propria. Come prima cosa visualizzerà il pannello che viene visualizzato in figura 5.20 senza il pulsante Start, che suggerisce di muovere la testa verso la direzione in cui si trova l'automobile, in modo tale da permettere al sistema di tracking di localizzare la posizione della Tesla nell'ambiente reale. Una volta che il sistema è riuscito a localizzare la posizione dell'auto, verrà mostrata una versione tridimensionale dell'auto sovrapposta all'auto reale, e comparirà nel pannello il pulsante Start con cui l'utente può interagire per iniziare.



**Figura 5.20:** Pannello di avvio applicazione

Quando la versione reale e la versione digitale dell'auto saranno perfettamente sovrapposte, l'utente dovrà interagire con il pulsante di Start. A questo punto, il modello 3D dell'automobile scomparirà e comparirà soltanto la versione 3D del telaio dell'auto, che suggerisce all'utente quali sono gli elementi coinvolti nella procedura e la loro posizione. Normalmente gli elementi del telaio sono nascosti nell'auto, la loro visualizzazione in scala reale direttamente sull'oggetto reale ha l'obiettivo di rendere più efficace l'apprendimento e la memorizzazione delle diverse componenti e della loro locazione.

Parallelamente a questo, verrà anche mostrato nella scena il primo pannello con raffigurato lo step dell'istruzione che deve eseguire nel punto indicato dalla freccia. Le informazioni riguardanti ogni step vengono recuperate a partire dal file JSON che l'applicazione aveva precedentemente scaricato dal server (o aveva già memorizzato su disco se non ci sono stati aggiornamenti). A questo punto, una volta che verrà effettuata l'operazione indicata dal pannello, l'utente dovrà semplicemente interagire coi pulsanti Next o Back per navigare tra i diversi step e visualizzare le prossime (o le precedenti) operazioni da eseguire. L'interfaccia dei pannelli è quella visualizzata nella figura 5.21. Oltre a testo e immagini, quando il

pannello si trova nel campo di vista dell'utente le istruzioni testuali vengono lette grazie al meccanismo di Text To Speech.



**Figura 5.21:** Interfaccia del pannello di istruzione.

Come già spiegato precedentemente, alcune istruzioni possono essere associate a dei Warning. Quando l'utente si trova in questa situazione visualizzerà di fronte a lui un pannello rosso che seguirà la sua testa e la sua posizione, che contiene un testo che verrà letto dal Text To Speech preceduto dalla dicitura "Warning". L'interfaccia di questo pannello è visualizzata in figura 5.22. Quando si interagirà col pulsante Ok, ci sarà un'animazione che muoverà il pannello rosso dalla posizione iniziale (di fronte alla testa dell'utente) alla posizione finale (ossia affianco al pannello blu principale, come si può vedere in figura 5.23). Finché non verranno visualizzati tutti i warning associati all'istruzione, l'interazione coi pulsanti Next e Back non darà alcun feedback.



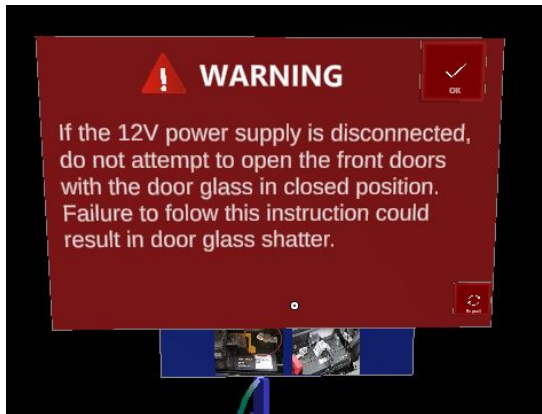


Figura 5.22: Interfaccia di Warning

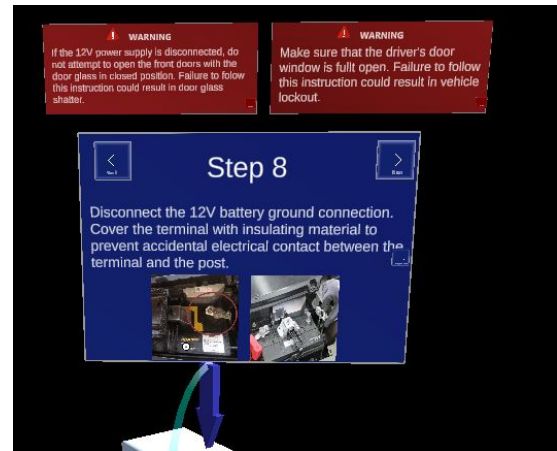


Figura 5.23: Visualizzazione dei warning dopo l'Ok

Inoltre, alcune istruzioni daranno la possibilità all'utente di inserire dei valori misurati. In questo caso l'interfaccia visualizzata sarà quella della figura 5.24. L'output dei valori inseriti da tastiera AR saranno visualizzabili direttamente nel panel, sostituiti dalla sezione in cui compare la scritta *Open Keyboard to type text*. Se l'utente inserisce un valore da tastiera, questo verrà salvato insieme ad un timestamp e alla descrizione dell'istruzione a cui è associato in modo tale da essere visualizzabile nel file di report.

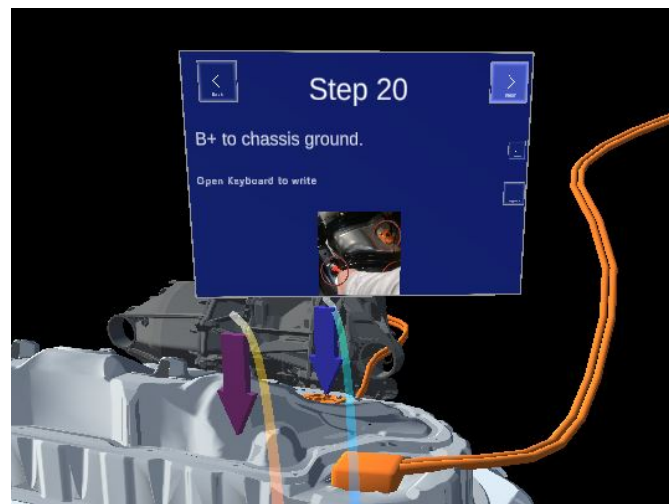
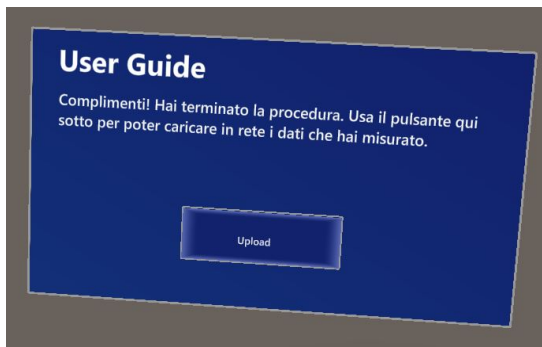
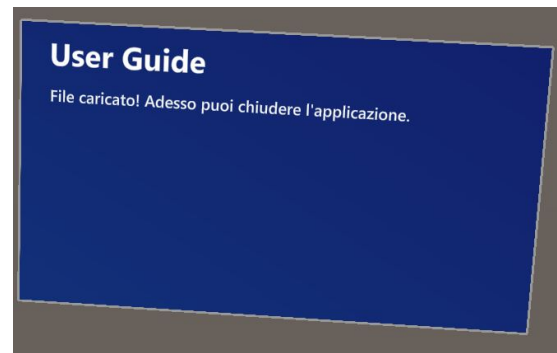


Figura 5.24: Interfaccia dell'inserimento di valori di input da tastiera

**Fine** Una volta terminati tutti gli step, ci sarà un pannello (che seguirà la testa dell'utente) con cui dovrà interagire per poter caricare sul server i dati di report inseriti da input (figura 5.25), per poi visualizzare il pannello finale che dice all'utente che può chiudere l'applicazione (figura 5.26).

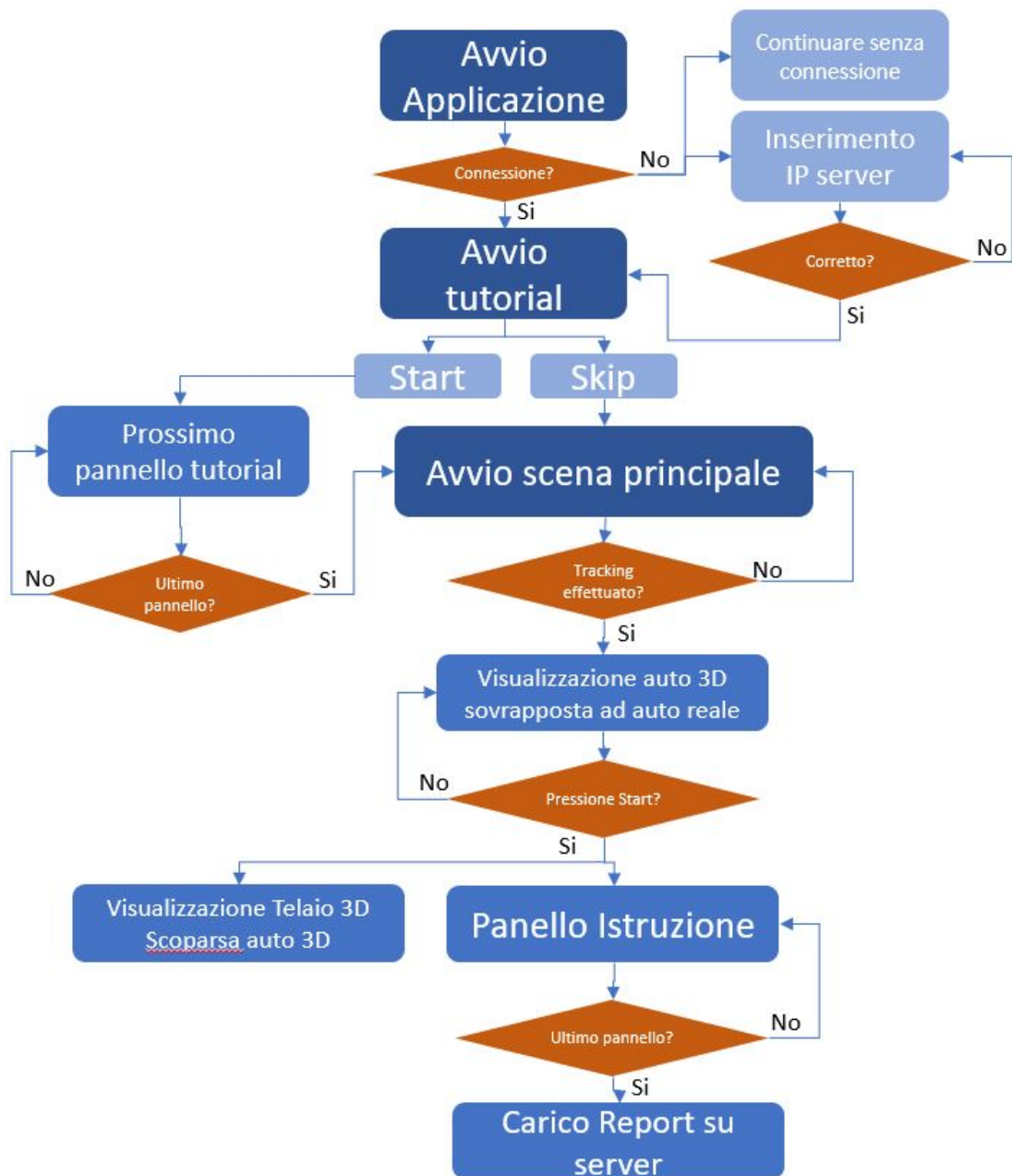


**Figura 5.25:** Interfaccia per caricare il file sul server



**Figura 5.26:** Interfaccia finale

**Diagramma di flusso** In figura 5.27 viene mostrato un riassunto schematico di quanto accade all'interno dell'applicazione.



**Figura 5.27:** Diagramma di flusso dell'applicazione

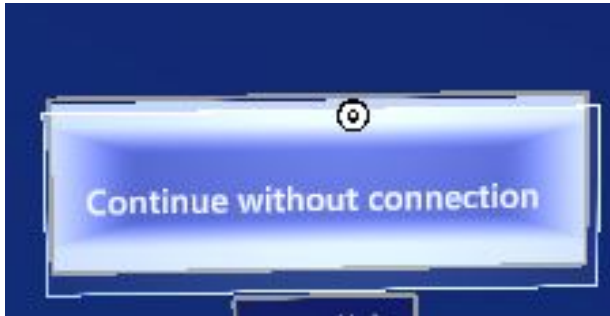
### 5.4.2 Modelli di interazione

Gli unici elementi interattivi presenti nella scena sono i pulsanti presenti nei diversi pannelli con cui l'utente può decidere l'azione da fare. Poichè lo scenario prevede che in alcune situazioni le mani degli operatori siano occupate da attrezzature,

ma anche per evitare di interagire per sbaglio con i pulsanti mentre si fanno determinate operazioni, è stato scelto un modello di interazione che non prevede l'uso di interazioni manuali con i pulsanti AR. In particolare, per interagire con i diversi elementi presenti nella scena l'utente ha tre possibilità diverse:

- Utilizzare il modello di interazione Puntamento con la testa e attesa (Hand Gaze & Dwell). Il sistema crea un raggio a partire dalla testa dell'utente, e al primo elemento di collisione che incontra il raggio viene visualizzato un puntatore. Quando c'è una collisione con un elemento interattivo, se l'utente vuole compiere l'azione associata all'interazione con l'elemento, dovrà tenere la testa ferma in quella direzione per 1,5 secondi per far avvenire l'evento associato all'interazione. Inoltre, mentre si mantiene lo sguardo verso il pulsante, questo si colorerà progressivamente di bianco per indicare all'utente un feedback visivo dell'azione (figura 5.28).
- Puntare la direzione della testa verso il pulsante con cui si vuole interagire, e confermare l'azione prima del passare dei secondi semplicemente posizionando la mano di fronte al campo visivo e unendo pollice e indice (come si vede in figura 5.30).
- In alternativa, l'utente può interagire mediante comandi vocali, che vengono suggeriti dall'applicazione quando un pulsante interattivo viene puntato con la testa (come si vede in figura 5.29). I comandi vocali presenti nell'applicazione sono:
  - **Next**: navigazione in avanti tra i diversi pannelli di istruzione. Svolge l'azione presente nel pulsante Next.
  - **Back**: navigazione indietro tra i diversi pannelli di istruzione. Svolge l'azione presente nel pulsante Back.
  - **Repeat**: possibilità di riascoltare il Text To Speech dell'istruzione visualizzata.
  - **Okay**: conferma di lettura del pannello di Warning.
  - **Keyboard**: apertura della tastiera AR nelle istruzioni che prevedono l'utilizzo di input vocali.
  - **Start**: inizializzazione dell'applicazione.
  - **Skip**: saltare il tutorial.
  - **Upload**: Carica il file di report sul server.

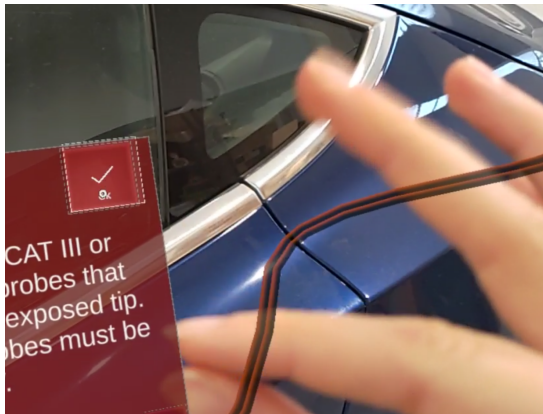
Inoltre, rivolgendo la mano verso gli occhi con il palmo rivolto verso l'alto è possibile visualizzare un Hand Menu che contiene dei pulsanti per permettere all'utente sia di nascondere la vista del telaio dell'auto, qualora questa potesse



**Figura 5.28:** Puntatore Hand Gaze & Dwell



**Figura 5.29:** Interfaccia suggerimento comando vocale



**Figura 5.30:** Interazione con sguardo + gesture



**Figura 5.31:** Hand Menu

risultare fastidiosa per il proseguimento della procedura, sia di disabilitare il Text To Speech.

L'unico momento in cui all'utente è richiesto l'utilizzo delle mani è quando deve inserire dei valori di input dalla tastiera AR: l'inserimento dei valori numerici avviene mediante pressione dei tasti utilizzando le dita della mano, come si può vedere in figura 5.32.



**Figura 5.32:** Interazione con tastiera AR

### 5.4.3 Struttura del progetto

Il progetto Unity è stato innanzitutto predisposto con le impostazioni per lo sviluppo e il deploy di applicazioni di Mixed Reality, come discusso nel capitolo 4.2.1. La build finale contiene al proprio interno tre diverse scene, che verranno di seguito analizzate.

**Scena 0** Manager delle scene, è presente in contemporanea a tutte le altre scene per tutta la durata dell'applicazione. La struttura gerarchica della scena in Unity è mostrata nella figura 5.33. La scena 0 non viene mai inizializzata da sola, all'avvio



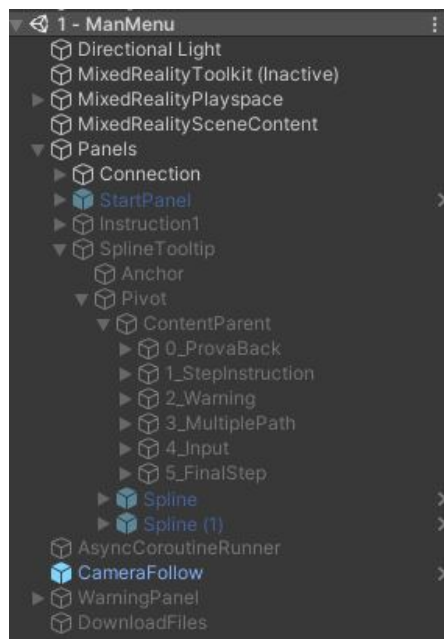
**Figura 5.33:** Struttura gerarchica scena 0

lo script `LoadContentScene` appartenente al `GameObject SceneLoader` si occuperà di caricare immediatamente la scena successiva (Scena 1). Il suo scopo principale è fornire alle altre scene la camera e i profili di configurazione del `MixedRealityToolkit` (che saranno infatti disattivati nelle altre scene). Il profilo di configurazione principale di `MRTKToolkit` è stato `DefaultHololens1ConfigurationProfile` (che consente di gestire le interazioni mediante puntatore dalla testa). Questo è stato

poi clonato per poter modificare alcuni attributi. In particolare sono stati modificati i seguenti parametri:

- Sezione Input System Profile - sezione Mixed Reality Speech Command: sono stati aggiunti i nuovi comandi vocali elencati in 5.4.2
- Sezione Scene System - è stato abilitato il sistema delle scene (che consente il passaggio da una scena all'altra nel mondo di Unity)

**Scena 1** Scena di avvio dell'applicazione. La struttura gerarchica della scena in Unity è mostrata in figura 5.34. Contiene gli elementi dell'interfaccia di Avvio



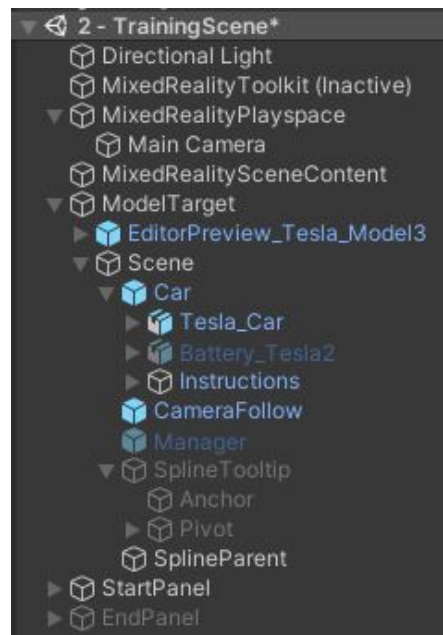
**Figura 5.34:** Struttura gerarchica scena 1

(5.4.1) e Tutorial (5.4.1). Allo start della scena viene innanzitutto verificato se c'è connessione col server mediante la classe ConnectionStatus attaccato al GameObject Connection. L'indirizzo IP del server è scritto su un file salvato su disco, che si chiama IPAddress.txt. Se non c'è risposta dal server, verrà chiesto all'utente di inserire mediante la tastiera AR il nuovo indirizzo IP del server. Se l'indirizzo inserito dall'utente è valido, viene aggiornato sul file IPAddress.txt in modo tale che a un successivo lancio dell'applicazione nella stessa rete non verrà richiesto nuovamente l'IP. Una volta stabilita la connessione col server verrà attivato il GameObject DownloadFiles, che verificherà se sul server è presente un aggiornamento dei dati lato desktop, controllando se il numero contenuto nel file su server Version.txt coincide con il numero contenuto nel file su disco Version.txt.



In caso positivo, viene visualizzato un pulsante nella schermata di Start che dà la possibilità all'utente di scaricare gli aggiornamenti da server. Interagendo con il pulsante, viene aggiornato il file su disco Version.txt e viene scaricato il file JSON aggiornato, con tutte le immagini relative, e vengono salvate su disco. Successivamente viene mostrato il pannello che dà la possibilità all'utente di saltare o procedere con il tutorial. Se l'utente procede con il tutorial, verranno visualizzati i pannelli di istruzione, ognuno dei quali contiene un'istruzione specifica di ogni funzionalità offerta dal sistema. L'interfaccia di ogni step viene visualizzata attivando e disattivando progressivamente i GameObject figli di ContentParent alla pressione dei pulsanti Next o Back. Alla fine dei 5 step di tutorial, l'utente può procedere con la scena successiva.

**Scena 2** Scena principale dell'applicazione. La struttura gerarchica della scena in Unity è mostrata in figura 5.35. Contiene gli elementi dell'interfaccia di Start (5.4.1) e Fine (5.4.1).

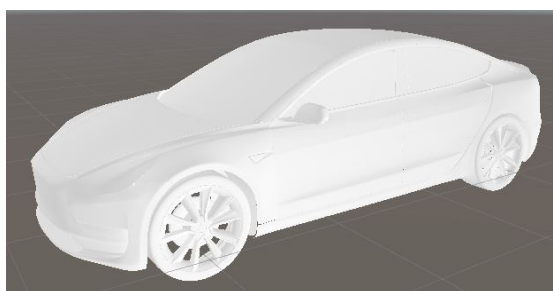


**Figura 5.35:** Struttura gerarchica scena 2

Per integrare il comportamento di Model Tracking di Vuforia è necessario importare il GameObject ModelTarget dal menù di Unity (GameObject -> Vuforia Engine -> Model Target). Nello script ModelTargetBehaviour attaccato al GameObject ModelTarget si deve selezionare il database a cui fare riferimento per definire l'oggetto da tracciare. Per farlo bisogna importare nel progetto Unity il file di output (in formato .unitypackage) che ha fornito l'applicazione di Vuforia



Model Target Generator (come è stato spiegato nel capitolo 4.2.2). In questo modo nella scena di Unity si visualizzerà un oggetto che avrà la forma dell'automobile (come si vede in figura 5.36) e servirà per aiutare lo sviluppatore a posizionare nella scena gli elementi 3D relativamente alla posizione dell'auto nella scena reale. Infatti, tutti gli elementi che nell'applicazione finale devono avere una posizione nello spazio relativa alla posizione tracciata dell'automobile devono essere figli del GameObject ModelTarget, poichè questo rappresenta il loro principale sistema di riferimento. Per questo motivo tutti gli oggetti principali della scena sono contenuti sotto il GameObject Scene, a sua volta figlio di Model Target. Inoltre, tutti gli oggetti figli di Model Target verranno attivati nella scena soltanto quando avverrà il tracking dell'oggetto reale.



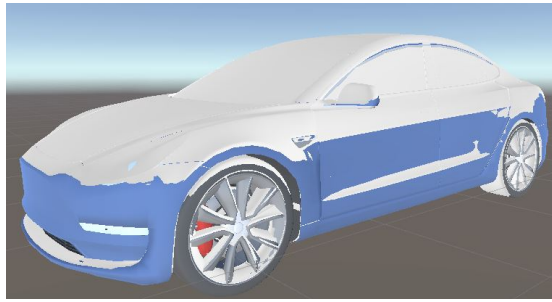
**Figura 5.36:** Visualizzazione del Model Target della Tesla

La scena inizia con la visualizzazione del GameObject StartPanel (rappresentato nella figura 5.20), che seguirà la posizione e la rotazione della testa dell'utente (questo comportamento avviene grazie agli script Follow e SolverHandler). Quando la camera rileva la presenza dell'automobile e la riesce a tracciare, avvengono due eventi:

- Viene attivato l'evento On Target Found dello script Default Observer Event attaccato al GameObject ModelTarget, a cui è associata la comparsa, nel pannello di avvio, del pulsante di Start con cui l'utente potrà interagire per far comparire i pannelli delle istruzioni.
- Viene visualizzata una versione 3D dell'automobile sovrapposta all'automobile reale. Questo comportamento in fase di sviluppo viene attivato posizionando nella scena di Unity il modello 3D dell'auto nella stessa posizione in cui si trova il modello del Model Target (come si vede in figura 5.37).

L'utente è invitato a interagire con il pulsante di Start quando troverà una perfetta sovrapposizione del modello 3D all'auto reale. La pressione del tasto Start comporta i seguenti comportamenti:

- Disattiva il GameObject associato al modello 3D dell'auto

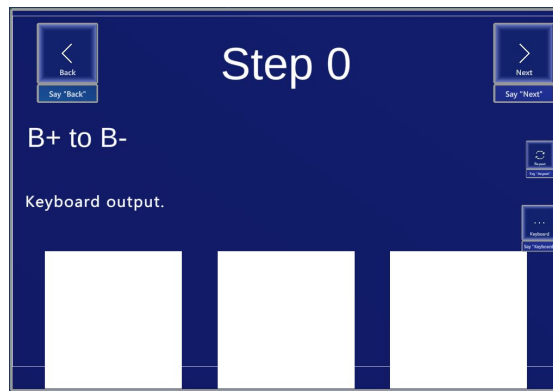


**Figura 5.37:** Posizionamento del modello 3D dell'auto sull'auto reale

- Attiva il GameObject associato al modello 3D del telaio dell'auto
- Salva le coordinate del GameObject Scene (che contiene tutti gli elementi di Realtà Aumentata che si dovranno visualizzare) e lo toglie come figlio del Model Target (che viene disattivato), mantenendolo però nella stessa posizione. Dato che si prevede che l'automobile rimanga ferma, è inutile continuare a tenere attivo il meccanismo di tracking. Inoltre, il tracking viene perso quando si entra nell'automobile, e questo avrebbe fatto sparire le istruzioni posizionate all'interno dell'auto.
- Attiva il GameObject Manager che contiene lo script InstructionsDownload, che si occupa di accedere alle informazioni contenute nel file JSON presente su disco (precedentemente scaricato da server) per poter istanziare nella scena tanti pannelli quanti sono gli step presenti sul file, con i relativi contenuti (quindi popolandolo con testo e immagini appropriate). I pannelli vengono creati come figli del GameObject ContentParent, che contiene uno script nominato Billboard, che fa in modo che gli oggetti figli siano sempre rivolti verso la camera dell'utente. I pannelli di istruzioni dopo la creazione verranno disattivati (non saranno quindi visibili nella scena), ad eccezione dello step numero 1. I pannelli istanziati sono presi da un Prefab presente nel progetto Unity. Se lo step prevede l'inserimento di un input (ossia ha il valore inputData settato a True), verrà aggiunto al pannello un pulsante che consente di aprire la tastiera, con una sezione di testo dedicata alla visualizzazione dell'output (come si vede in figura 5.38).

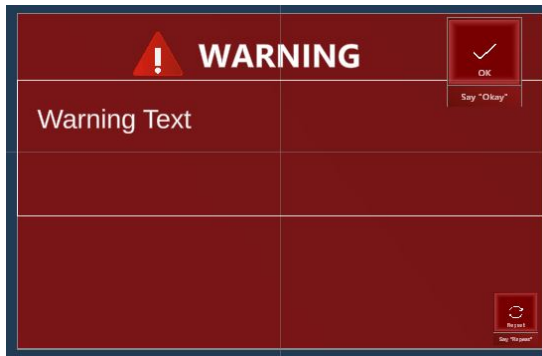
A questo punto, l'utente può interagire coi pulsanti Back e Next per poter navigare progressivamente tra i diversi step. L'interazione con questi pulsanti è associata a un metodo che scaturirà i seguenti comportamenti:

- Controlla se l'utente ha completato la conferma di lettura a tutti i warning a cui è associata l'istruzione, altrimenti non può navigare tra gli step.

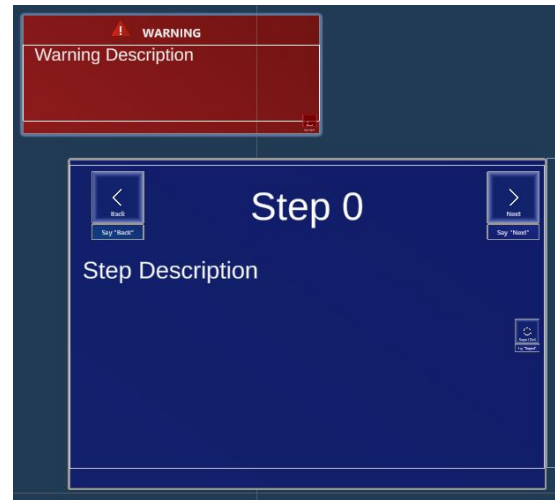


**Figura 5.38:** Prefab del pannello con tastiera

- (Solo per il pulsante Next) Se il pannello prevede l'inserimento di un valore di input da tastiera, viene controllato che l'utente abbia effettivamente inserito dei dati. In caso negativo, non potrà navigare tra i vari step. In caso positivo, salverà il valore inserito sul file di report.
- (Solo per il pulsante Next) Viene controllato se la prossima istruzione che si visualizzerà è associata a uno o più Warning. In caso positivo, oltre ad attivare il prossimo panel, verrà istanziato nella scena un pannello di tipo Warning (figura 5.39) che seguirà la testa dell'utente, e contenente la descrizione del primo elemento dell'array del parametro warningDescription del file JSON relativo a quello step. L'interazione con il pulsante OK inoltre causerà due eventi:
  - Farà partire un'animazione che trasporterà il pannello rosso dalla posizione iniziale davanti alla testa dell'utente, a una posizione finale accanto al pannello di istruzioni principale (figura 5.40)
  - Controllerà se a quello step sono associati altri warning. In caso positivo, verrà nuovamente istanziato un altro pannello di tipo Warning contenente la descrizione dell'elemento successivo dell'array del parametro warningDescription.
- Se l'utente può navigare tra gli step, viene aggiornata la posizione del GameObject Pivot con la posizione a cui è associata l'istruzione successiva o precedente.
- (Solo per il pulsante Next) Se lo step attuale è l'ultimo, viene mostrato il pannello EndPanel, che provvedrà a caricare sul Server il file di Report e ad avvertire l'utente che la procedura è terminata.



**Figura 5.39:** Prefab del pannello di Warning



**Figura 5.40:** Pannello di istruzioni con il warning

Inoltre, a ogni cambio di step, verrà aggiornata la posizione del GameObject Pivot, figlio del GameObject Tooltip. Il GameObject ToolTip contiene lo script Tooltip che abilita la visualizzazione di una curva spline che collega due punti: un punto di Ancora, ovvero il punto di inizio della curva, e un punto di Pivot, ossia un punto di fine della curva. Come punto di ancora è stato messo il GameObject CameraFollow, che segue la direzione della camera con un piccolo offset in modo tale da far partire la curva poco più giù della testa, e come punto di pivot il GameObject Pivot, che al suo interno contiene uno script che prende il nome di FollowInstructionPlace. Questo script contiene un metodo che consente che la posizione del Pivot viene aggiornata di volta in volta a seconda del parametro position a cui è associato lo step. L'aggiornamento della posizione del Pivot causa automaticamente lo spostamento dei pannelli di istruzione, in quanto suoi figli secondari. Per definire i diversi punti di interesse sull'automobile sono stati creati diversi GameObject empty posizionati adeguatamente facendo riferimento al modello 3D dell'automobile. Ogni empty viene mappato su un numero intero, ed è questo a cui fa riferimento il parametro position. Può capitare che i punti di interesse di un'istruzione sia più di uno: in questo caso l'empty avrà tanti figli (posizionati adeguatamente) quanti sono i punti di interesse, e lo script FollowInstructionPlace si occuperà di istanziare tanti oggetti di tipo Tooltip nella scena quanti sono i figli, facendo partire dalla posizione dell'utente altre curve contraddistinte dal colore rosa, ognuna verso la direzione di un figlio.

# Capitolo 6

## Test e Risultati

In questo capitolo verrà discussa la fase di valutazione del sistema proposto, a partire dai test a cui sono stati sottoposti gli utenti al fine di valutare le funzionalità e i vantaggi che offre l'utilizzo di questa applicazione. A tale scopo sono state eseguite due tipologie di test, entrambe richiedevano la compilazione di un questionario finale.

La prima tipologia è stata eseguita nei laboratori dell'azienda Bylogix, da ingegneri Elettronici e Meccanici. E' stato chiesto agli utenti di simulare lo svolgimento della procedura sull'automobile reale, e sono state confrontate due metodologie, una che prevede la lettura delle istruzioni mediante Hololens e una che prevede l'utilizzo delle istruzioni cartacee.

La seconda seconda tipologia, invece, è stata eseguita presso la Digital Pilot Line del CIM 4.0. E' stata utilizzata una versione demo dell'applicazione finale che non richiedeva la presenza dell'automobile reale, che è stata sostituita dalla sua versione digitale. Il campione testato comprende 10 utenti, a cui è stato chiesto di interagire con l'applicazione visualizzando tutti gli step della procedura e simularla sull'oggetto virtuale.

### 6.1 Test comparativi

Al fine di valutare l'utilità e le performance che l'applicazione sviluppata porta in termini di tempo e consapevolezza dei pericoli che si possono verificare nell'esecuzione della procedura, sono stati condotti presso l'azienda Bylogix dei test utente a cui hanno partecipato 6 ingegneri meccanici o elettronici. Alla metà dei partecipanti è stato chiesto di svolgere la procedura simulando le azioni da compiere, sull'automobile reale utilizzando le istruzioni cartacee, mentre all'altra metà è stato chiesto di utilizzare l'applicazione su Hololens. In entrambi i gruppi, 1 persona su 3 ha svolto questa procedura per la prima volta, e nessuno dei partecipanti ha mai

utilizzato la tecnologia Hololens. Al termine del compito, a ogni utente è stato chiesto di compilare un questionario.

A entrambi i gruppi è stato misurato il tempo di completamento della procedura. Al termine della procedura a ogni utente è stato sottoposto lo stesso questionario, al fine di poter effettuare un confronto sia sui tempi sia sulle risposte ai questionari.

### 6.1.1 Test oggettivi

Per valutare l'impatto che l'applicazione AR ha in termini di tempo della procedura, sono state confrontate le tempistiche delle due soluzioni in fase di test. Nella tabella 6.1 si può visualizzare il tempo che ogni utente ha impiegato per completare la simulazione della procedura.

Tipologia procedura	Utente 1	Utente 2	Utente 3	Media Temporale
Versione Hololens	12'57"	12'10"	11'30"	11'99"
Versione Cartacea	12'32"	10'42"	12'16"	11'63"

**Tabella 6.1:** Tabella dei tempi di ogni utente

Ad una prima occhiata i due risultati sembrano quasi coincidere, tuttavia in queste tempistiche vanno tenute in considerazione diversi fattori:

- La procedura su Hololens conteneva degli step aggiuntivi che non erano presenti nella procedura cartacea
- Tutti gli utenti che hanno svolto la procedura su Hololens hanno fatto uso della funzione Text To Speech, quindi hanno prestato più attenzione a quanto c'era scritto sui pannelli blu e soprattutto hanno fatto più attenzione ai pannelli di warning.

Di conseguenza, alla luce di queste informazioni aggiuntive, la strategia di utilizzo delle Hololens si può considerare leggermente vantaggiosa in termini di tempo rispetto alla strategia di utilizzo delle istruzioni cartacee.

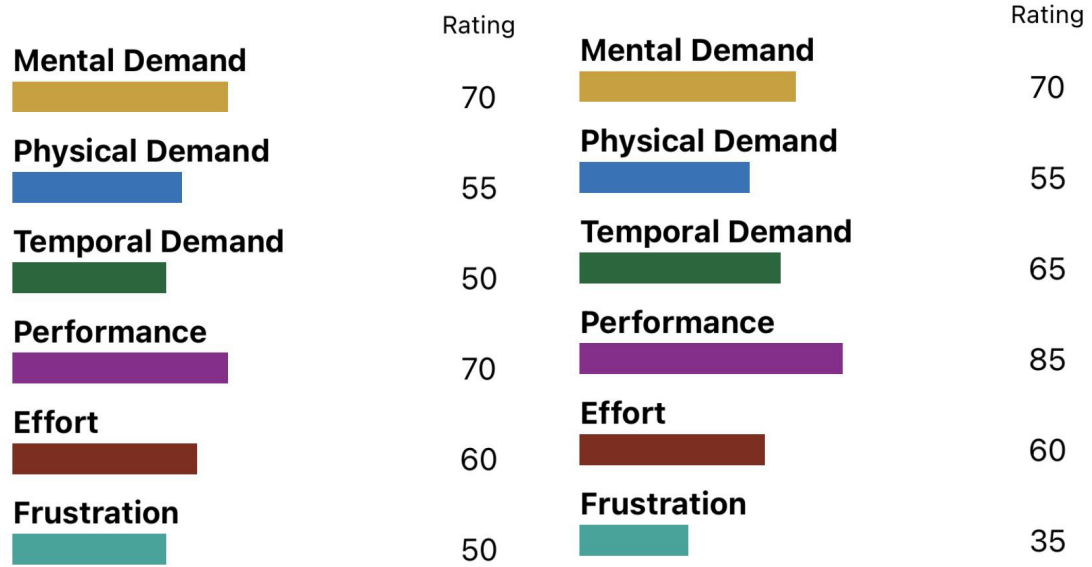
### 6.1.2 Test soggettivi

Alla fine della procedura, gli utenti di entrambi i gruppi sono stati sottoposti a un questionario per valutare l'impatto che ha l'utilizzo di questa applicazione rispetto all'utilizzo dei fogli cartacei per l'esecuzione di questa procedura.

**Nasa-TLX** Il Task Load Index TLX ufficiale della NASA è uno strumento di valutazione del carico di lavoro soggettivo, che permette di valutare il carico di lavoro sugli operatori che lavorano con sistemi di interfaccia uomo-macchina. Questo tipo di questionario è diventato standard per la misurazione del carico di lavoro soggettivo in una vasta gamma di applicazioni. Il NASA-TLX attribuisce un punteggio del carico di lavoro complessivo basato su una media ponderata di valutazioni su sei sottoscale:

- Domanda mentale: quanta attività mentale e percettiva è stata richiesta (pensare, decidere, calcolare, ricordare, guardare, cercare, ecc.)? Il compito era facile o impegnativo, semplice o complesso, impegnativo o indulgente?
- Domanda fisica: quanta attività fisica è stata richiesta (spingere, tirare, girare, controllare, attivare)? Il compito era facile o impegnativo, lento o veloce, lento o faticoso, riposante o laborioso?
- Domanda temporale: Quanta pressione hai sentito a causa della frequenza o del ritmo con cui si svolgevano le attività? il ritmo era lento e lento o rapido e frenetico?
- Prestazioni: quanto successo pensi di aver avuto nel raggiungimento degli obiettivi del compito fissato dallo sperimentatore (o da te stesso)? quanto eri soddisfatto della tua performance nel raggiungimento di questi obiettivi?
- Sforzo: Quanto hai dovuto lavorare (mentalmente e fisicamente) per raggiungere il tuo livello di performance?
- Frustrazione: quanto ti sei sentito insicuro, scoraggiato, irritato, stressato e infastidito rispetto a sicuro, gratificato, contento, rilassato e compiacente durante il compito?

Nella figura 6.1 e 6.2 si possono visualizzare i report inseriti dagli utenti in relazione alle diverse domande. Si può notare anche quì che i due risultati sono molto simili, anche se in termini di valutazione della performance e di livello di frustrazione c'è un grande stacco a favore della procedura lato Hololens.

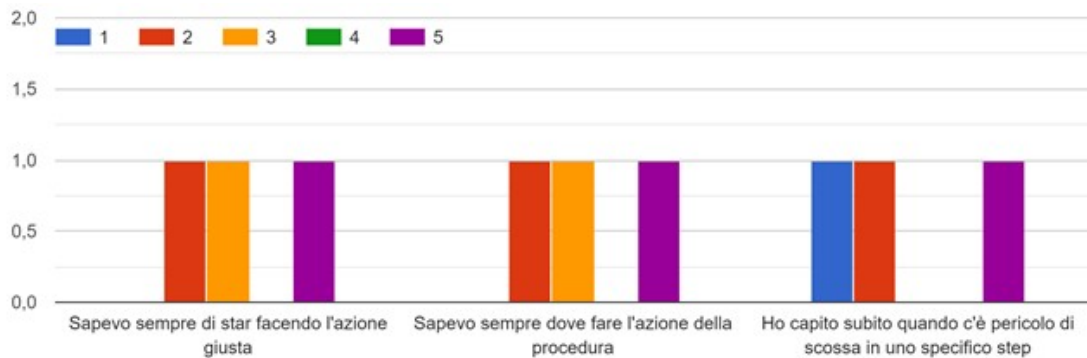


**Figura 6.1:** Report questionario NASA procedura cartacea

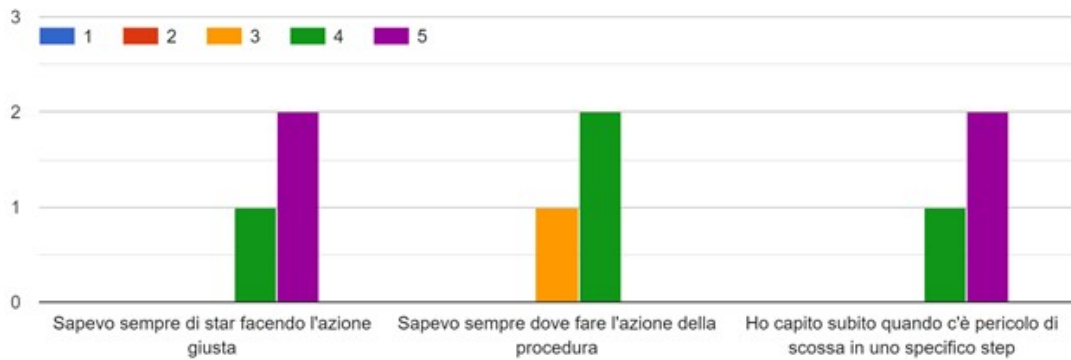
**Figura 6.2:** Report questionario NASA procedura Hololens

**Domande specifiche** Gli utenti sono anche stati sottoposti a domande specifiche riguardanti la procedura stessa. In figura 6.3 e 6.4 si può visualizzare un confronto delle risposte tra le due procedure. Come si può vedere dalle risposte, l'impatto delle Hololens è fortemente rilevante: gli utilizzatori di Hololens sono più consapevoli della posizione in cui le azioni vengono eseguite, e sono sicuri che siano corrette, rispetto agli utilizzatori di istruzioni cartacee. Inoltre, uno dei parametri più importanti da prendere in considerazione, è la consapevolezza che alcuni step possano causare scosse elettriche che potrebbero essere anche mortali. Nella procedura cartacea si acquisisce nettamente meno consapevolezza delle azioni che possono compromettere la salute, a differenza della procedura Hololens.





**Figura 6.3:** Report questionario procedura cartacea



**Figura 6.4:** Report questionario procedura Hololens

## 6.2 Test utente

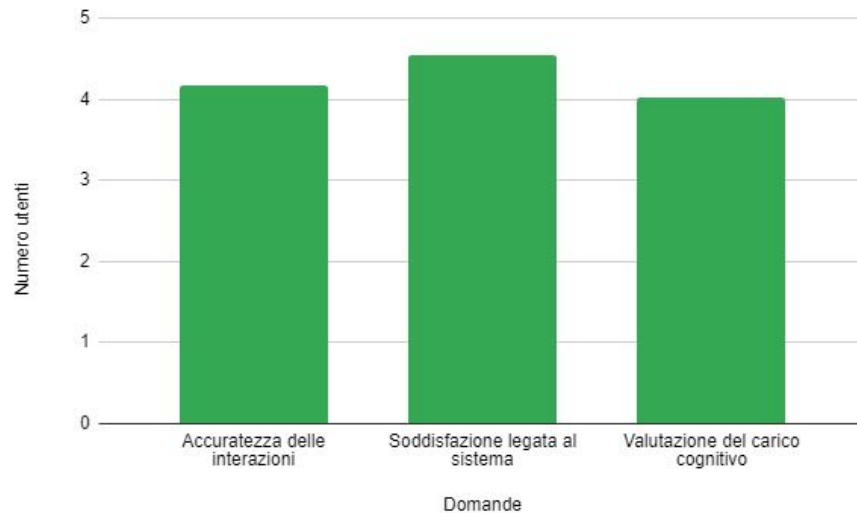
In questa sezione verranno analizzati l'usabilità e il design dell'applicazione in base alle risposte fornite dal questionario. A tale scopo, sono state raccolte e unite sia le risposte fornite dai 3 utenti di Bylogix che hanno completato la procedura con il dispositivo Hololens, sia le risposte fornite dai 10 utenti testati presso i laboratori del CIM 4.0, per un totale di 13 test.

### 6.2.1 Valutazione del design

Per poter valutare il design dell'applicazione, a ogni utente è stato chiesto di rispondere a una serie di domande presenti sul questionario.

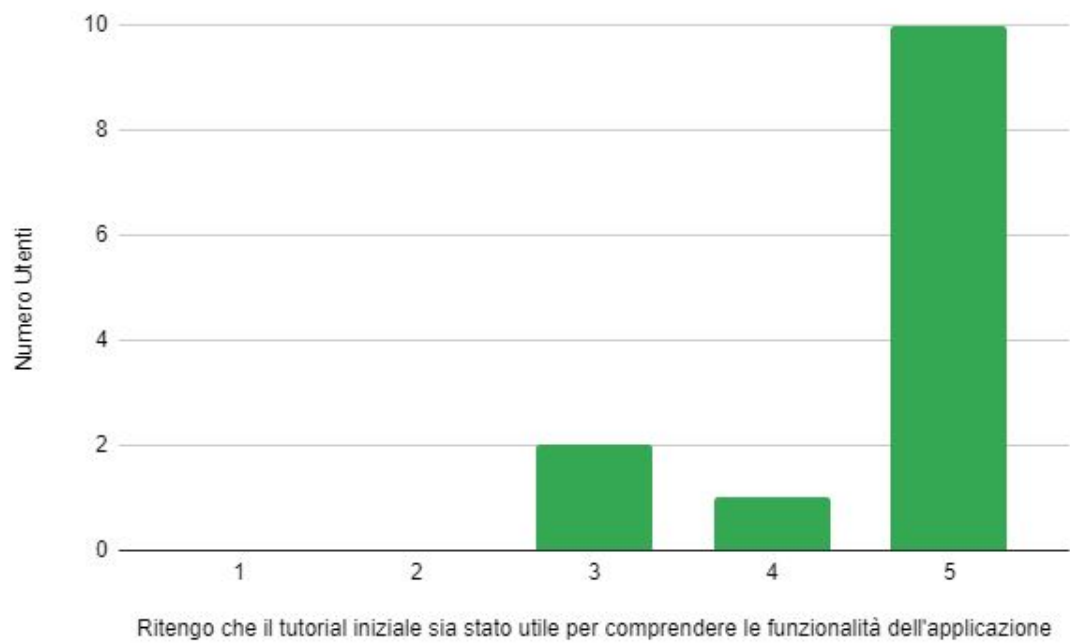
Una prima serie di domande sono state catalogate per accuratezza delle informazioni, soddisfazione legata al sistema e valutazione del carico cognitivo. Ognuna di queste domande richiedeva l'inserimento di un valore in una scala da 1 a 5, in base

a quanto fosse d'accordo con le affermazioni mostrate dalla domanda (1: fortemente in disaccordo, 5: fortemente d'accordo). In figura 6.5 si può visualizzare una media matematica dei punteggi assegnati per ogni categoria. Alla luce dei risultati, si può affermare che l'utilizzo dell'applicazione ha avuto globalmente un impatto positivo per l'utente, sia a livello di accuratezza delle interazioni, sia a livello di carico cognitivo dell'utente, sia a livello di soddisfazione generale nell'utilizzo dell'applicativo come strumento per eseguire la procedura.

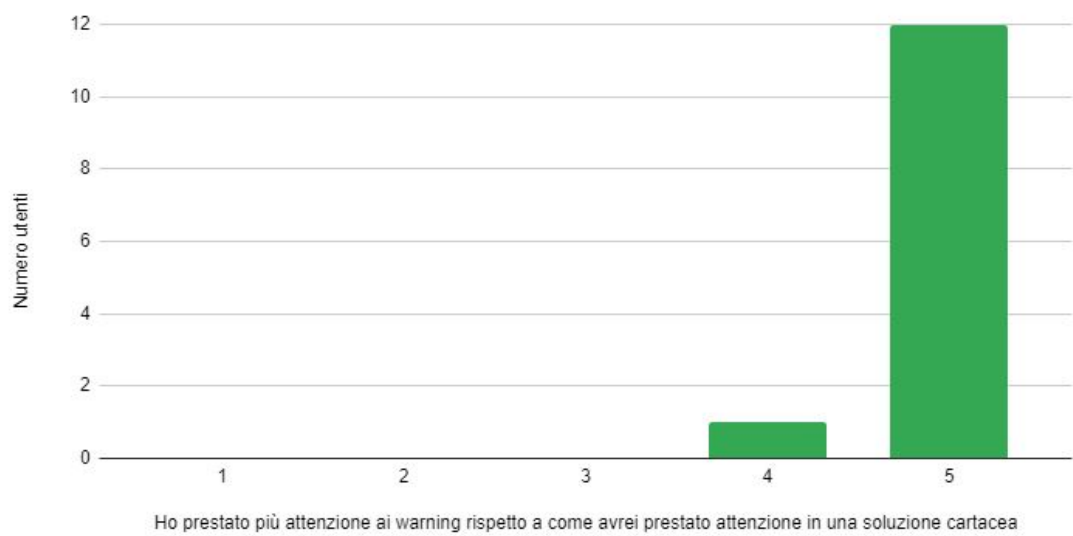


**Figura 6.5:** Istogramma di valutazione sul design del sistema

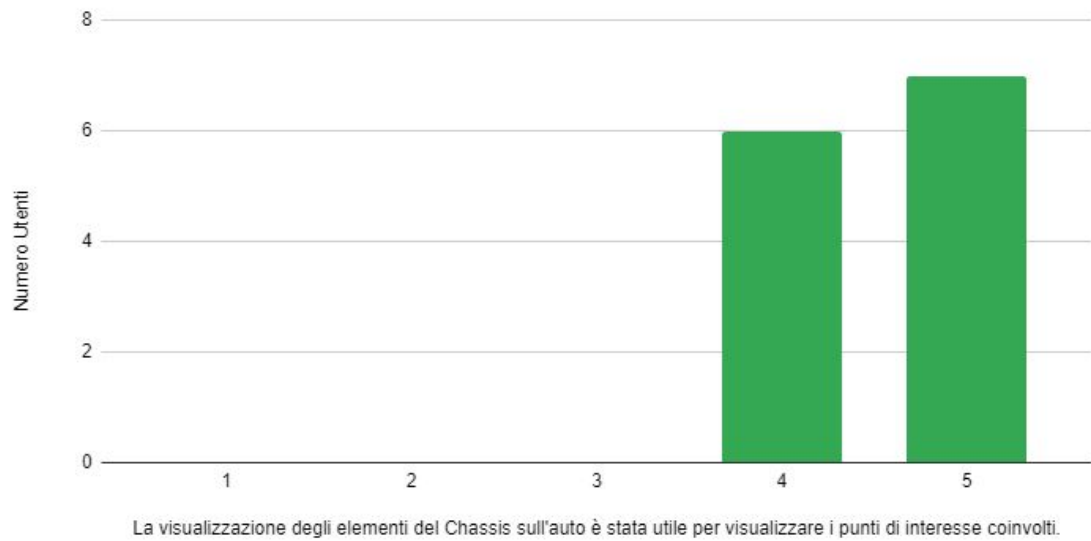
Inoltre è stato posto un ulteriore set di domande (sempre con scala di valutazione da 1 a 5) relative a utilità del tutorial (figura 6.6), utilità della metodologia di visualizzazione dei pannelli di warning (figura 6.7), l'utilità della visualizzazione degli elementi del telaio dell'automobile (figura 6.8) e infine l'utilità dell'utilizzo dell'applicazione Hololens per comprendere meglio la procedura rispetto al cartaceo (figura 6.9). Di seguito verranno visualizzati i grafici che mostrano i risultati di queste domande



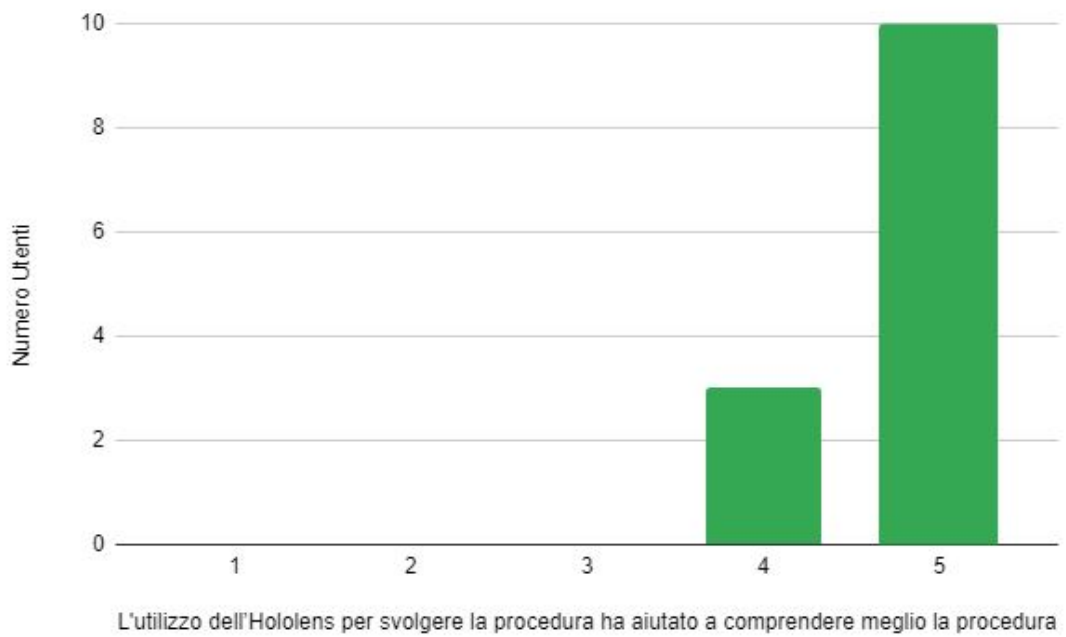
**Figura 6.6:** Istogramma di valutazione sul tutorial



**Figura 6.7:** Istogramma di valutazione sulla modalità di visualizzazione dei Warning



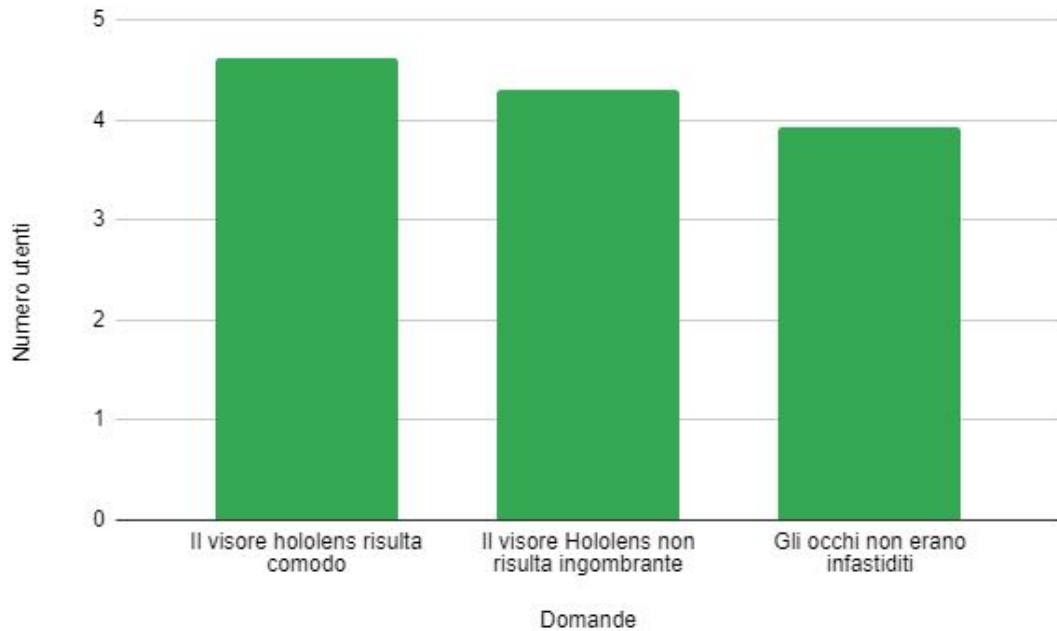
**Figura 6.8:** Istogramma di valutazione sulla visualizzazione del telaio



**Figura 6.9:** Istogramma di valutazione sull'utilità dell'utilizzo di HoloLens

### 6.2.2 Valutazione dell'ergonomia del dispositivo

Al fine di valutare l'ergonomia del dispositivo Hololens, per poter approfondire un suo utilizzo prolungato in un contesto industriale, sono state poste anche domande relative al fastidio che ha provato l'utente durante il suo utilizzo. Anche in questo caso, all'utente è stato richiesto di dare un valore in una scala da 1 a 5 in base a quanto fosse d'accordo con tale affermazione. Di seguito, in figura 6.10 troviamo un valore medio per tutte le risposte fornite dagli utenti per ogni affermazione.



**Figura 6.10:** Istogramma di valutazione sull'ergonomia dell'utilizzo di Hololens

## Capitolo 7

# Conclusioni e sviluppi futuri

La ricerca svolta ha cercato di fornire un contributo sia bibliografico sia applicativo per quanto riguarda l'utilizzo di tecnologie di Realtà Aumentata in ambiente industriale, in particolare per quello che riguarda il settore automotive.

La premessa su cui si è basato lo sviluppo di questa applicazione era registrare una minore tempistica e un maggior livello di soddisfazione dell'utente che svolge la procedura, diminuendo gli errori e rendendolo consapevole dei rischi corsi in determinati step della procedura. I test utente svolti si sono rilevati parzialmente coerenti con questa affermazione. Per quanto riguarda le tempistiche di esecuzione, i risultati di entrambe le metodologie, sia cartacea sia hololens, sono quasi paragonabili, con un leggero vantaggio delle Hololens se si considerano tutti i fattori esterni. Confrontando invece i risultati soggettivi delle due metodologie, si può affermare che l'utilizzo del dispositivo Hololens per compiere la procedura sia estremamente utile per rendere l'operatore consapevole sia dei punti in cui si devono svolgere le operazioni, ma soprattutto dei rischi che si possono correre in determinati step. La modalità di visualizzazione dei pannelli di Warning si è infatti dimostrata efficace, portando l'utente a prestare molta più attenzione alle avvertenze e di conseguenza a essere più prudente negli step più pericolosi dal punto di vista infortunistico.

Inoltre la visualizzazione delle componenti del telaio dell'automobile direttamente sull'auto reale è stato uno degli elementi che ha aiutato maggiormente l'operatore a comprendere i punti di interesse della procedura. In particolare, si è rivelato estremamente utile per questo caso specifico la visualizzazione dei cavi elettrici, soprattutto per chi non si è mai interfacciato coi veicoli elettrici.

Per quanto riguarda l'applicazione desktop, è risultata di facile comprensione e utilità al supervisore. In particolare, una delle difficoltà dell'operatore risultava individuare i tasti da premere nel touchscreen, ad esempio per spegnere il climatizzatore. Nei pannelli visualizzabili in Realtà Aumentata vengono inserite delle immagini che indicano all'utente i punti da premere, tuttavia quando il sistema si aggiorna le posizioni dei punti da premere potrebbero cambiare. Di conseguenza

il fatto di poter caricare facilmente le informazioni aggiornate nelle immagini dei pannelli risulta estremamente utile.

Alla luce di questi risultati è bene ricordare che la maggior parte degli utenti non si erano mai interfacciati con la tecnologia Hololens, per cui con un utente già esperto di tale tecnologia si possono porre le basi per uno sfruttamento più massiccio di applicativi simili in un contesto industriale.

Una raccomandazione per ulteriori ricerche future potrebbe essere quella di rendere l'applicazione ancora più scalabile sfruttando l'applicazione desktop, ad esempio dando la possibilità al supervisore di inserire facilmente nuovi oggetti su cui effettuare il tracking, con la possibilità di inserire nuovi punti di interesse del nuovo oggetto selezionato. L'applicazione è stata sviluppata apposta per rendere semplice la modifica di questa feature. Si potrebbe inoltre integrare la funzionalità del Remote Assistant in modo tale che un supervisore possa visionare come viene operata la procedura senza che sia fisicamente presente nel luogo.

# Bibliografia

- [1] Gabriel Evans, Jack Miller, Mariangely Iglesias Pena, Anastacia MacAllister e Eliot Winer. «Evaluating the Microsoft HoloLens through an augmented reality assembly application». In: *Degraded Environments: Sensing, Processing, and Display 2017*. A cura di John (Jack) N. Sanders-Reed e Jarvis (Trey) J. Arthur III. Vol. 10197. International Society for Optics e Photonics. SPIE, 2017, pp. 282–297. DOI: 10.1117/12.2262626. URL: <https://doi.org/10.1117/12.2262626> (cit. alle pp. 2, 22).
- [2] Hilal Atici-Ulusu, Yagmur Dila Ikiz, Ozlem Taskapilioglu e Tulin Gunduz. «Effects of augmented reality glasses on the cognitive load of assembly operators in the automotive industry». In: *International Journal of Computer Integrated Manufacturing* 34.5 (2021), pp. 487–499. DOI: 10.1080/0951192X.2021.1901314. eprint: <https://doi.org/10.1080/0951192X.2021.1901314>. URL: <https://doi.org/10.1080/0951192X.2021.1901314> (cit. a p. 2).
- [3] 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. 6).
- [4] Thomas Caudell e David Mizell. «Augmented reality: An application of heads-up display technology to manual manufacturing processes». In: vol. 2. Feb. 1992, 659–669 vol.2. ISBN: 0-8186-2420-5. DOI: 10.1109/HICSS.1992.183317 (cit. alle pp. 7, 22).
- [5] Paul Milgram, Haruo Takemura, Akira Utsumi e Fumio Kishino. «Augmented reality: a class of displays on the reality-virtuality continuum». In: *Telemanipulator and Telepresence Technologies*. A cura di Hari Das. Vol. 2351. International Society for Optics e Photonics. SPIE, 1995, pp. 282–292. DOI: 10.1117/12.197321. URL: <https://doi.org/10.1117/12.197321> (cit. a p. 7).



- 
- [6] R. Azuma, B. Hoff, H. Neely e R. Sarfaty. «A motion-stabilized outdoor augmented reality system». In: *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*. 1999, pp. 252–259. DOI: 10.1109/VR.1999.756959 (cit. a p. 8).
- [7] H. Kato e M. Billinghurst. «Marker tracking and HMD calibration for a video-based augmented reality conferencing system». In: *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*. 1999, pp. 85–94. DOI: 10.1109/IWAR.1999.803809 (cit. a p. 8).
- [8] Wayne Piekarski e Bruce Thomas. «ARQuake: The Outdoor Augmented Reality Gaming System». In: *Commun. ACM* 45.1 (gen. 2002), pp. 36–38. ISSN: 0001-0782. DOI: 10.1145/502269.502291. URL: <https://doi.org/10.1145/502269.502291> (cit. a p. 8).
- [9] Emanuele Marino, Loris Barbieri, Biagio Colacino, Anna Kum Fleri e Fabio Bruno. «An Augmented Reality inspection tool to support workers in Industry 4.0 environments». In: *Computers in Industry* 127 (2021), p. 103412. ISSN: 0166-3615. DOI: <https://doi.org/10.1016/j.compind.2021.103412>. URL: <https://www.sciencedirect.com/science/article/pii/S0166361521000191> (cit. a p. 17).
- [10] Alex Samuely. «A Hyundai modernizes car owner's manual with augmented reality spin». In: (nov. 2015). URL: <https://www.marketingdive.com/ex/mobilemarketer/cms/news/software-technology/21686.html> (cit. a p. 20).
- [11] Azreen Zuriathy Halim. «Applications of Augmented Reality For Inspection and Maintenance Process in Automotive Industry». In: feb. 2018. DOI: 10.4314/jfas.v10i3s.35 (cit. a p. 20).
- [12] Răzvan Gabriel Boboc, Florin Gîrbacia e Eugen Valentin Butilă. «The Application of Augmented Reality in the Automotive Industry: A Systematic Literature Review». In: *Applied Sciences* 10.12 (2020). ISSN: 2076-3417. DOI: 10.3390/app10124259. URL: <https://www.mdpi.com/2076-3417/10/12/4259> (cit. a p. 21).
- [13] Sulakshana Iyer. «Why mixed reality solution gives an upper hand to automotive manufacturers». In: lug. 2021. URL: <https://www.softwebsolutions.com/resources/mixed-reality-in-automotive-industry.html> (cit. a p. 22).
- [14] Barfield W. Baird K.M. «Evaluating the effectiveness of augmented reality displays for a manual assembly task». In: (dic. 1999). DOI: <https://doi.org/10.1007/BF01421808> (cit. a p. 22).
- [15] Rafael Radkowski e Jarid Ingebrand. «HoloLens for Assembly Assistance - A Focus Group Report». In: mag. 2017, pp. 274–282. ISBN: 978-3-319-57986-3. DOI: 10.1007/978-3-319-57987-0\_22 (cit. a p. 22).

- [16] Hilal Atici-Ulusu, Yagmur Dila Ikiz, Ozlem Taskapilioglu e Tulin Gunduz. «Effects of augmented reality glasses on the cognitive load of assembly operators in the automotive industry». In: *International Journal of Computer Integrated Manufacturing* 34.5 (2021), pp. 487–499. DOI: 10.1080/0951192X.2021.1901314. eprint: <https://doi.org/10.1080/0951192X.2021.1901314>. URL: <https://doi.org/10.1080/0951192X.2021.1901314> (cit. a p. 23).
- [17] Alisa Taylor. «Under the hood of the Mercedes-Benz Virtual Remote Support system». In: (ott. 2020). URL: <https://cloudblogs.microsoft.com/dynamics365/bdm/2020/10/08/under-the-hood-of-the-mercedes-benz-virtual-remote-support-system/> (cit. a p. 23).
- [18] Nico Krespach. «Bosch Augmented Reality applications now also work with the new Microsoft HoloLens 2». In: (feb. 2019). URL: <https://www.bosch-presse.de/pressportal/de/en/bosch-augmented-reality-applications-now-also-work-with-the-new-microsoft-hololens-2-183680.html> (cit. a p. 24).
- [19] «Under the hood of the Mercedes-Benz Virtual Remote Support system». In: (mag. 2019). URL: <https://www.media.volvocars.com/global/en-gb/media/pressreleases/253105/volvo-cars-and-varjo-launch-world-first-mixed-reality-application-for-car-development> (cit. a p. 25).
- [20] «Lockheed Martin builds for the next stage in human space exploration with Microsoft HoloLens 2». In: (gen. 2021). URL: <https://customers.microsoft.com/en-us/story/857748-lockheed-martin-discrete-manufacturing-hololens-en-united-states> (cit. a p. 29).
- [21] <https://docs.microsoft.com/en-us/hololens/hololens2-hardware>. Feb. 2021 (cit. a p. 31).
- [22] *Vuforia Model Target*. <https://library.vuforia.com/objects/model-targets> (cit. a p. 35).
- [23] *Vuforia Model Target Generator*. <https://library.vuforia.com/objects/model-target-generator-user-guide> (cit. a p. 36).
- [24] *Modello 3D Tesla Model 3*. <https://www.turbosquid.com/3d-models/3d-model-tesla-3-interior-chassis/1140406> (cit. a p. 38).
- [25] *Modello 3D del telaio Tesla Model 3*. <https://www.turbosquid.com/3d-models/3d-model-tesla-3-interior-chassis/1140406> (cit. a p. 38).