

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

DIPARTIMENTO DI INFORMATICA – SCIENZA E INGEGNERIA
Corso di Laurea in Ingegneria e Scienze Informatiche

STUDIO E ANALISI
DELL'ARCHITETTURA DI APPLICAZIONI
DI REALTÀ MISTA IN HOLOLENS 2:
UN CASO DI STUDIO

Elaborato in
SISTEMI EMBEDDED E INTERNET OF THINGS

Relatore
Prof. ALESSANDRO RICCI

Presentata da
FILIPPO VISSANI

Corelatore
Dott. Ing. ANGELO CROATTI

Anno Accademico 2020 – 2021

“Reality is what you want it to be.”

M. Pell

Indice

Introduzione	vii
1 Introduzione alla Mixed Reality	1
1.1 Cenni Storici	1
1.2 Definizioni di VR, MR e AR	3
1.3 Caratteristiche di un'Applicazione MR	5
1.3.1 Determinare la Posizione dell'Utente	5
1.3.2 Sistemi di Coordinate	7
1.3.3 Interazioni con gli Ologrammi	8
1.3.4 Esperienze Multi-Utente	9
1.4 Tassonomia dei Dispositivi MR	9
1.4.1 Head Mounted Display	10
1.4.2 Head-up Display	12
1.4.3 Smart-Glasses	13
1.4.4 Dispositivi Mobili	13
1.5 Framework di Sviluppo MR	14
1.5.1 Microsoft Mixed Reality Toolkit	14
1.5.2 Apple ARKit	15
1.5.3 Google ARCore	15
1.5.4 OpenXR	16
1.5.5 Vuforia	17
2 Microsoft HoloLens 2	19
2.1 Specifiche Tecniche del Device	19
2.2 Architettura di un'Applicazione HoloLens 2	21
2.2.1 Sistemi di Coordinate	22
2.2.2 Ologrammi	22
2.2.3 Mappatura Spaziale	23
2.2.4 Ancore	24
2.2.5 Hand Tracking e Gaze	25
2.3 Strumenti di Sviluppo	26
2.3.1 Sistema di Input	27

2.3.2	Percezione Ambientale	28
2.3.3	Definizione dei Confini	29
2.4	Esempio di Applicazione HoloLens 2	29
2.4.1	Configurazione del Progetto Unity	29
2.4.2	Funzionamento dell'Applicazione	30
3	Esempi di Applicazioni per Hololens 2	35
3.1	Ambito Industriale	35
3.2	Ambito Sanitario	37
3.3	Ambito Educativo	38
3.4	Ambito Museale	41
4	Caso di Studio	43
4.1	Descrizione del Caso di Studio	43
4.2	Analisi dei Requisiti	45
4.2.1	Casi d'Uso	45
4.2.2	Identificazione delle Informazioni e dei Dispositivi IoT nell'Ambiente	48
4.2.3	Allineamento dei Sistemi di Coordinate	48
4.2.4	Interazioni fra Entità Virtuali e Reali	49
4.3	Progettazione del Sistema	49
4.3.1	Back-End Engine	51
4.3.2	Holographic App	53
4.4	Sviluppo del Prototipo	56
4.4.1	Tecnologie Utilizzate	56
4.4.2	Struttura del Progetto Unity	57
4.4.3	Identificazione delle Informazioni nell'Ambiente	58
4.4.4	Gestione degli Ologrammi Relativi alle Informazioni	60
4.4.5	Allineamento dei Sistemi di Coordinate	61
4.4.6	Persistenza degli Ologrammi nel Tempo e nello Spazio	63
4.4.7	Prototipo in Esecuzione	64
4.5	Considerazioni Finali	66
4.5.1	Sviluppi Futuri	66
	Conclusioni	69
	Ringraziamenti	71

Introduzione

Negli ultimi anni realtà virtuale e realtà aumentata hanno iniziato a diffondersi sempre più rapidamente. Ogni anno vengono introdotte tecnologie che garantiscono esperienze sempre migliori. Una di queste tecnologie è HoloLens 2, un dispositivo concepito da Microsoft per permettere esperienze di realtà mista.

In questa tesi si vogliono presentare i risultati degli studi effettuati sullo sviluppo di applicazioni Mixed Reality per HoloLens 2.

La realtà mista viene prima di tutto descritta da un punto di vista teorico, ma vengono anche definite architettura e struttura delle applicazioni relative. L'obiettivo è quello di fornire una visione completa della Mixed Reality, non solo da un punto di vista teorico, ma anche tecnico e pratico.

Nel primo capitolo, dopo l'introduzione di alcuni cenni storici, vengono presentati i principali aspetti della realtà mista, che viene anche comparata alla realtà virtuale e alla realtà aumentata. Successivamente vengono descritte le caratteristiche che un'applicazione Mixed Reality ha. Infine, vengono introdotti i dispositivi attualmente in commercio che supportano questa tecnologia e i framework di sviluppo relativi.

Il secondo capitolo introduce HoloLens 2 da un punto di vista tecnico, insieme all'architettura e allo sviluppo di applicazioni Mixed Reality per questo dispositivo. In questo capitolo vengono evidenziate le potenzialità del dispositivo e viene anche fornito un primo esempio di applicazione.

Nel terzo capitolo vengono descritti alcuni ambiti in cui l'applicazione di HoloLens 2 può migliorare determinati aspetti, come la produttività, l'efficienza e la sicurezza.

Per mostrare in pratica alcune delle potenzialità della Mixed Reality descritte nei capitoli precedenti, nel quarto capitolo viene presentato il caso di studio, relativo all'ambito sanitario. Qui viene fornita una descrizione dettagliata del caso, corredata dall'analisi dei requisiti e da una proposta di progetto. Infine, viene definito lo sviluppo del prototipo relativo al progetto, in modo da rendere il lettore cosciente di cosa può fare HoloLens 2 concretamente.

Capitolo 1

Introduzione alla Mixed Reality

1.1 Cenni Storici

I concetti di realtà virtuale (VR) e realtà aumentata (AR) comparirono per la prima volta nel 1965, quando Ivan Sutherland (ricercatore dell'Università di Harvard) pubblicò un saggio [5] contenente la seguente citazione:

“The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal. With appropriate programming such a display could literally be the Wonderland into which Alice walked.”

Anni dopo, nel 1968, Sutherland riuscì a sviluppare con successo il primo head-mounted display (HMD) della storia (Figura 1.1). A causa del suo peso, doveva essere appeso al soffitto e per questo motivo venne soprannominato “Spada di Damocle”.

Lo sviluppo tecnologico avvenuto fra gli anni ‘80 e ‘90 fu fondamentale per permettere alla realtà aumentata di diventare un campo di ricerca indipendente.

Il termine “realtà aumentata” venne coniato nel 1992 da Thomas Caudell e David Mizell [1], impiegati alla Boeing Computer Services Research. I due ricercatori svilupparono un HMD che permetteva di sovrapporre schemi specifici di un aereo (generati al computer) su schede riutilizzabili (Figura 1.2). In questo modo, invece di riconfigurare manualmente ogni scheda per ogni fase del processo di produzione, le istruzioni di cablaggio personalizzate sarebbero state visualizzate dal lavoratore e modificate in modo rapido ed efficiente grazie al visore.



Figura 1.1: "La Spada di Damocle" è il soprannome dato al primo head mounted display al mondo, costruito da Ivan Sutherland nel 1968.

Nel 1994 Paul Milgram pubblicò un articolo intitolato: "Augmented reality: A class of displays on the reality-virtuality continuum" [3], introducendo i concetti di "reality-virtuality (RV) continuum" e di "mixed reality (MR)", che definiscono la correlazione fra mondo reale e virtuale.

Negli anni successivi il campo della realtà aumentata ha continuato a evolversi e si è sempre più affermato nella cultura popolare. le tecnologie AR e VR sono diventate rapidamente di qualità superiore, più economiche e più ampiamente disponibili agli utenti.



Figura 1.2: I ricercatori di Boeing hanno utilizzato un see-through HMD, creato da David Mizell, per guidare l'assemblaggio dei fasci di cavi per gli aerei.

1.2 Definizioni di VR, MR e AR

Spesso i termini "realtà aumentata" e "realtà virtuale" vengono erroneamente interscambiati, quando in verità si tratta di due tecnologie con caratteristiche ben distinte.

Nella realtà aumentata le informazioni generate al computer vengono sovrapposte al mondo fisico, abbiamo quindi un mondo composto da una parte reale e una parte virtuale che sono in stretta correlazione fra loro.

Entrando più nel dettaglio, gli aspetti chiave della realtà aumentata sono i seguenti:

- Il mondo reale viene aumentato da informazioni digitali sovrapposte a esso.
- Le informazioni vengono visualizzate in registrazione spaziale e temporale con il mondo fisico.

- Le informazioni visualizzate dipendono dalla posizione e dalla prospettiva dell'utente nel mondo reale.
- L'esperienza di realtà aumentata è interattiva, ovvero l'utente può interagire con le informazioni e apportare modifiche a esse.
- Il livello d'interattività può variare dal semplice cambiamento della prospettiva alla manipolazione e persino alla creazione di nuove informazioni.

Nella realtà virtuale invece l'utente viene proiettato in un ambiente generato interamente al computer, non necessariamente correlato con il mondo reale. I sensi vengono occlusi in tutto o in parte, in modo da far sembrare reale l'ambiente creato artificialmente. Una volta definite le caratteristiche di queste due tecnologie la domanda che sorge spontanea è:

”Che rapporto c'è fra VR e AR?”

La risposta la diede Paul Milgram nel 1994, pubblicando “Augmented Reality: A class of displays on the reality-virtuality continuum” [3] che, come detto nella sezione 1.1, definisce i concetti di Reality-Virtuality Continuum e di Mixed Reality.

Il RV continuum presenta una gamma di realtà che varia dal completamente reale al completamente virtuale (Figura 1.3). Secondo Milgram tutto ciò che risiede fra questi due estremi fa parte della Mixed Reality, che è composta dalla Augmented Reality, dove la realtà viene aumentata da entità virtuali e dalla Augmented Virtuality, dove il mondo virtuale viene aumentato da entità appartenenti al mondo reale.

Ciò che distingue la realtà mista da quella virtuale è l'integrazione delle entità virtuali nell'ambiente, chiamata registrazione spaziale; queste infatti, hanno una posizione e occupano uno spazio nel mondo fisico esattamente come se fossero reali, mentre nelle applicazioni AR il contenuto digitale si sovrappone al mondo reale senza integrarsi. La Figura 1.4 descrive meglio questo concetto: abbiamo un'applicazione MR che introduce entità virtuali all'interno di un ambiente reale; la posizione di queste entità non dipende dalla posizione dell'utente nell'ambiente, quindi se ad esempio l'utente si spostasse verso un altro lato del tavolo vedrebbe la casa virtuale da un'altra prospettiva. Naturalmente è possibile anche interagire con le entità virtuali, che possono essere spostate, ridimensionate o anche eliminate.

In un certo senso, le entità digitali devono essere gestite come oggetti fisici almeno per quanto riguarda la questione della registrazione. Per questo motivo la registrazione con il mondo reale deve essere sia spaziale che temporale.

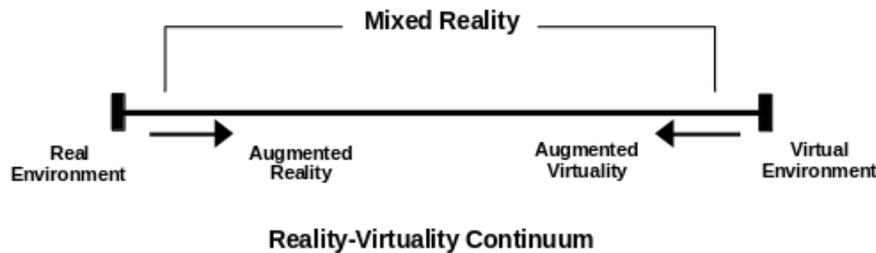


Figura 1.3: Rappresentazione semplificata del Reality-Virtuality Continuum teorizzato da Paul Milgram.

1.3 Caratteristiche di un'Applicazione MR

Nelle applicazioni MR le entità digitali vengono rappresentate dagli ologrammi (Figura 1.4), che possono assumere qualsiasi forma ed emettere suoni. Inoltre, gli ologrammi possono modificare lo spazio fisico arricchendolo.

Ogni volta che un sistema software basato su AR deve aggiornare la vista dell'utente sull'ambiente Mixed Reality per mostrargli gli ologrammi, vengono richiesti due passaggi. Prima di tutto, il sistema deve identificare lo stato attuale del mondo fisico e calcolare di conseguenza lo stato del mondo digitale. In secondo luogo, emerge l'esigenza di visualizzare ologrammi in registrazione con la realtà in modo efficace considerando le percezioni umane. Generalmente lo stato in tempo reale del mondo fisico è ottenuto sfruttando i sensori forniti dal dispositivo, in particolare a questo scopo vengono utilizzate telecamere e tecniche di computer vision. Un problema significativo nella realtà aumentata è sapere dove si trova l'utente rispetto al sistema di riferimento del mondo reale, ovvero occuparsi del rilevamento del movimento e della posizione.

1.3.1 Determinare la Posizione dell'Utente

Nella realtà aumentata, determinare la prospettiva dell'utente significa calcolare la posa della telecamera, che è rappresentata da una posizione nello spazio 3D con sei gradi di libertà (6DOF). La posa si riferisce alla posizione e all'orientamento della fotocamera. Per identificare la posa della fotocamera è possibile utilizzare sensori inerziali (come accelerometro o giroscopio) e di posizione (come bussola o GPS). Una volta rilevata la posa della fotoca-



Figura 1.4: Applicazione AR in cui le entità virtuali occupano una posizione nel mondo reale indipendentemente dalla collocazione e dai movimenti dell'utente.

mera, ovvero quando il sistema conosce la posizione e l'orientamento attuali dell'utente, gli ologrammi e i contenuti digitali possono essere presentati in allineamento con il mondo reale e proposti all'utente in un dato momento secondo la sua prospettiva. Secondo la letteratura, possono essere considerate diverse tecniche, eventualmente combinate, per determinare dove si trova l'utente e quale è il suo punto di vista. La tecnica più elementare, nota come AR marker based, dove il marker viene utilizzato per consentire il tracciamento al fine di determinare la posa del dispositivo. Oltre ai marker, possono essere applicate anche tecniche di tracciamento visivo in tempo reale. In particolare, il riferimento è a quelle tecniche che consentono una mappatura continua del mondo reale sfruttando un ampio insieme di punti rilevati, note anche come markerless AR.

Comunemente, un marker è un'immagine 2D asimmetrica che rappresenta un pattern molto semplice con l'obiettivo di essere rapidamente riconoscibile dagli algoritmi di visione artificiale (tipicamente modelli quadrati, in bianco e nero). In alcuni casi (ad esempio sfruttando Vuforia) è possibile sfruttare anche oggetti 3D come marker. Grazie alla loro forma e struttura predefiniti, i marker possono essere localizzati facilmente dal visore. Quindi, vengono utilizzati per un rapido calcolo della posa. Inoltre, per quanto riguarda quelli 2D, l'elevato contrasto dei quadrati in bianco e nero aiuta a facilitare il rilevamento. I marker vengono posizionati nell'ambiente reale con lo scopo di individuare un sistema di riferimento da utilizzare per accoppiare il layer digitale con quello reale. Utilizzando un marker posto in una posizione ben nota del mondo reale, la funzione di tracciamento utilizza la fotocamera per stimare la posa

del dispositivo in tempo reale in base a ciò che sta "vedendo".

In condizioni di scarsa illuminazione o quando la telecamera è lontana dal marker tracciato, è possibile utilizzare tecniche basate sul rilevamento di *natural feature point*. Natural Feature Tracking (NFT) è una tecnica AR per riconoscere e tracciare una scena per determinare la posa della fotocamera, evitando deliberatamente di usare i marker. NFT può sfruttare elementi incorporati nel mondo reale per migliorare i punti e le regioni di tracciamento. Il risultato può essere definito "markerless tracking" perché i "marker" non sono visibili all'utente. Sebbene gli approcci senza marker possano sembrare la soluzione migliore, bisogna tenere in considerazione che tracciare la posa della fotocamera in ambienti sconosciuti può essere complicato. Negli ultimi anni è stata sviluppata una tecnologia nota come SLAM (Simultaneous Localization and Mapping), che ridefinisce l'idea originale di AR markerless basata su feature point. Tuttavia, l'NFT può essere meno costoso dal punto di vista computazionale di un sistema SLAM completo e generalmente è più adatto per i dispositivi mobili. SLAM inizia con un ambiente sconosciuto e vuoto in cui il dispositivo di realtà aumentata cerca di generare una mappa e localizzarsi all'interno di essa. Attraverso una serie di algoritmi computazionalmente costosi, SLAM utilizza i dati del sensore IMU (unità di misura inerziale) per costruire una mappa dell'ambiente sconosciuto e allo stesso tempo per identificare la sua posizione. Infine, l'AR markerless utilizza in genere anche la funzione GPS per tracciare la posizione dell'utente in scenari all'aperto combinata con il tracciamento visivo per ottenere l'orientamento correlato.

1.3.2 Sistemi di Coordinate

Per familiarità, accuratezza e semplicità, la maggior parte degli ambienti virtuali è definita utilizzando un sistema di coordinate cartesiane standard, come mostrato in Figura 1.5. All'interno di questo sistema, ogni punto è specificato in modo univoco utilizzando tre coordinate numeriche (x , y , z) che rappresentano distanze specifiche (misurate nella stessa unità di lunghezza) da tre piani reciprocamente perpendicolari.

Oltre a essere l'approccio più comunemente utilizzato per la mappatura di spazi reali e virtuali, il sistema cartesiano viene impiegato anche per definire la posizione e il punto di vista di un utente nello spazio.

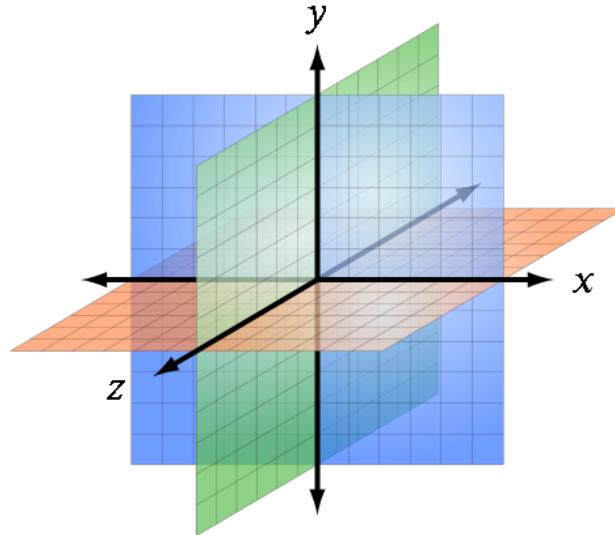


Figura 1.5: Ogni punto nel sistema di coordinate cartesiane 3D è identificato in modo univoco da valori x , y e z .

Un ologramma posto in un ambiente tridimensionale deve avere l'abilità di muoversi liberamente e di ruotare sui tre assi perpendicolari. Poiché il movimento lungo ognuno degli assi è indipendente quanto per gli assi di traslazione che per gli assi di rotazione, il movimento ha quindi sei gradi di libertà (6DoF: Six Degrees Of Freedom). Per garantire questa caratteristica è possibile sfruttare gli angoli di Eulero o i quaternioni. Solitamente vengono utilizzati i quaternioni perché con gli angoli di Eulero può presentarsi il fenomeno del blocco cardanico (in inglese gimbal lock), che avviene quando vengono allineati i due assi rotanti verso la stessa direzione, causando la perdita di un grado di libertà sull'asse bloccato.

1.3.3 Interazioni con gli Ologrammi

La realtà aumentata è, per sua natura, un mezzo interattivo. Se il contenuto digitale deve essere percepito come parte del mondo reale, poiché le persone possono interagire con oggetti fisici anche nei sistemi AR gli utenti dovrebbero essere in grado di interagire con gli ologrammi in modo simile. In effetti, l'interazione gioca un ruolo chiave nell'esperienza complessiva dell'utente. In un ambiente AR, l'interazione è generalmente limitata all'azione da eseguire sugli elementi digitali, ad esempio osservarli o afferrarli e spostarli da una posizione a un'altra. L'interazione in AR è un aspetto che dipende principalmente dal

dispositivo utilizzato per tracciare e mostrare ologrammi nell'ambiente Mixed Reality. La capacità di interagire con gli ologrammi aumenta proporzionalmente e diventa più potente in relazione all'efficienza del dispositivo AR utilizzato in termini di propri sensori e capacità computazionali.

1.3.4 Esperienze Multi-Utente

Il mondo reale è un ambiente cooperativo multiutente, gli ambienti di realtà mista diventano più interessanti quando anche il livello digitale e gli ologrammi possono essere condivisi tra più utenti che interagiscono contemporaneamente. Proporre un'esperienza multiutente cooperativa significa avere un layer digitale condiviso in tempo reale. Prima di tutto, gli utenti dovrebbero essere in grado di osservare gli stessi ologrammi ma dalla propria prospettiva. In questo caso, utilizzando dei marker, questa caratteristica potrebbe essere semplicemente ottenuta, ad esempio, con due utenti che osservano contemporaneamente lo stesso marker che propone per entrambi gli stessi sistemi di riferimento e orientamento. Sfortunatamente, questa è una soluzione che nasconde il vero problema: in questo caso, lo stato dell'ologramma non è realmente condiviso tra gli utenti. Ad esempio, se un utente sposta l'ologramma, l'altro utente continuerà a vedere l'ologramma nella posizione originale. Per risolvere i problemi proposti c'è l'esigenza che entrambi gli utenti facciano riferimento alla stessa istanza dell'ologramma osservato da entrambi, una sorta di centralizzazione dell'istanza del livello digitale, a cui accedono più utenti. Non sorprende che per sperimentare un ambiente MR multiutente cooperativo completo emergano esigenze che sono ben lungi dall'essere affrontate dalle soluzioni infrastrutturali software attualmente disponibili. Soprattutto quando la cooperazione non è intesa solo tra umani ma coinvolge anche entità software autonome e altre cose fisiche appartenenti al mondo reale. Un altro problema è dato dai diversi dispositivi che si possono usare per accedere all'esperienza condivisa. Con dispositivi (ad esempio smartphone e visori) basati su piattaforme differenti (come Windows, Lumin OS, Android e iOS) sorge l'esigenza di sviluppare API comuni per garantire l'interoperabilità delle applicazioni MR indipendentemente dalla piattaforma che si sta utilizzando. Una delle soluzioni proposte per venire a capo di questo problema è OpenXR, ovvero un'API che si occupa di fornire un accesso nativo alle piattaforme e ai dispositivi di Augmented, Mixed e Virtual Reality.

1.4 Tassonomia dei Dispositivi MR

Al giorno d'oggi è disponibile una moltitudine di dispositivi MR, che si contraddistinguono in base ai sistemi di input, output ed elaborazione che

utilizzano.

In generale, è possibile identificare due macrocategorie di sistemi AR: wearable (indossabili) e non-wearable (non indossabili).

Un'altra suddivisione può essere fatta distinguendo i dispositivi che sono stati concepiti appositamente per esperienze MR da quelli più generici (come PC e smartphone).

Alcuni dispositivi, come le lenti a contatto per la realtà aumentata, sono ancora in fase di sviluppo e non sono ancora disponibili prodotti commerciali.

1.4.1 Head Mounted Display

Un head mounted display è un visore incorporato in un casco, dotato di dispositivi di registrazione dell'orientamento per il movimento sincronizzato della testa e degli occhi dell'utente. Esistono due tipi di HMD attualmente in commercio, quelli che supportano esclusivamente esperienze VR (HTC Vive e Oculus Rift, Figura 1.6) e quelli progettati per esperienze AR (come HoloLens 2 e Magic Leap 1, Figura 1.7).

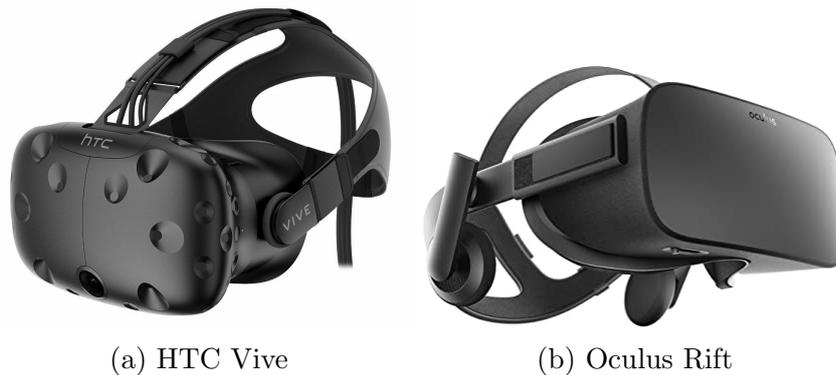


Figura 1.6: HTC Vive e Oculus Rift messi a confronto

Rift utilizza un singolo sensore esterno, un cilindro nero che si trova su un supporto da tavolo in metallo alto 23 centimetri. Il sensore può inclinarsi verso l'alto e verso il basso e deve essere posizionato in un punto in cui possa mantenere una visione chiara del visore quando è in uso. Un secondo sensore identico traccia i controller Oculus Touch, questi due sensori inoltre lavorano in tandem per migliorare il tracciamento di tutti i dispositivi e coprire un'area più ampia rispetto alla posizione fissa consentita da un solo sensore.

HTC Vive è composto non solo da un visore per la realtà virtuale e da due sofisticati controller, ma anche da un sistema di due telecamere laser (base station) che hanno il compito di tracciare il movimento dell'utente all'interno della stanza. HTC Vive supporta un'area che può avere una dimensione

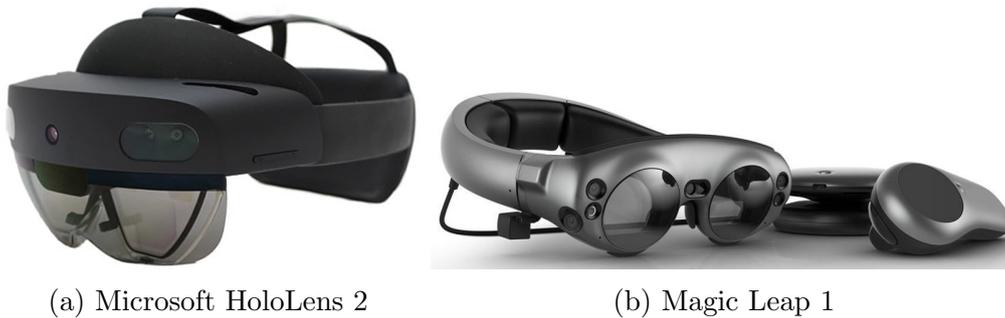


Figura 1.7: HoloLens 2 e Magic Leap 1 messi a confronto

massima di cinque metri di diagonale, pari alla massima distanza consigliata per le due base station, sufficiente per coprire interamente una stanza di circa 12 metri quadri, e dare così l'impressione di potersi muovere liberamente nel mondo virtuale.

Gli HMD che supportano la realtà aumentata sfruttano una tecnologia definita *optical see-through*, che permette di vedere attraverso il display.

Magic Leap 1 è dotato di uno schermo LCOS prodotto da Omnivision, che offre una definizione di 1280 x 960. Mentre il primo HoloLens aveva uno schermo con definizione HD (720p) e un campo visivo di 30 gradi, il suo successore è stato migliorato notevolmente e infatti ha un campo visivo di 52 gradi. HoloLens 2 supera Magic Leap in termini di definizione con il suo schermo che offre una definizione di circa 2K per occhio. In totale, questo nuovo visore visualizza 47 pixel per grado. Lo schermo di HoloLens 2 è quindi superiore in termini di definizione, mentre i due visori AR sono uguali in termini di campo visivo.

Magic Leap 1 viene fornito con il suo controller "Control". Questo controller offre sei gradi di libertà di movimento senza la necessità di sensori esterni aggiuntivi. Si basa sul tracciamento del campo elettromagnetico. È questo accessorio, simile ai controller per visori VR Oculus Rift o HTC Vive, che permette di interagire nella realtà aumentata di Magic Leap. Magic Leap 1 offre anche funzioni di tracciamento dell'iride e della mano. Quest'ultimo punto permette di interagire usando le mani in modo naturale. In realtà, questo sistema non rileva la posizione delle dita dell'utente nello spazio, ma solo gesture predefinite. HoloLens 2 si distingue per una tecnologia di tracciamento della mano decisamente accurata. Infatti, questo nuovo visore AR è in grado di seguire la posizione delle dieci dita dell'utente. Pertanto, è persino possibile suonare il pianoforte in realtà aumentata. Come con il primo HoloLens, purtroppo, è ancora necessario che le mani rimangano nel campo visivo di HoloLens 2 per il tracciamento. Troviamo anche la stessa tecnologia di eye tracking degli smart-

phone Lumia. L'Eye Tracking di HoloLens 2 è molto più avanzato, incluso lo sblocco del dispositivo sfruttando l'iride con Windows Hello. Anche il riconoscimento vocale è stato migliorato e ora è possibile controllare il dispositivo sfruttando i comandi vocali.

1.4.2 Head-up Display

Gli head-up display (Figura 1.8) sono stati sviluppati inizialmente per essere adottati in ambito militare, in particolare per proiettare informazioni utili nel campo visivo dei piloti di aerei. Successivamente gli HUD sono stati impiegati anche nel settore commerciale, ad esempio nelle auto vengono utilizzati per rendere immediatamente disponibili alcuni dati utili alla guida, senza che il guidatore debba distogliere lo sguardo dalla strada per concentrarsi sul quadro strumenti.



Figura 1.8: Head-up Display.

1.4.3 Smart-Glasses

Gli smart-glasses (Figura 1.9) sono occhiali che permettono all'utente di ricevere informazioni direttamente sulle lenti. I primi modelli potevano eseguire solo attività base, delegando il carico computazionale a dispositivi mobili come smartphone e tablet, connessi tramite Bluetooth, oggi invece sono in grado di eseguire vere e proprie applicazioni autonomamente.



Figura 1.9: Google Glass.

1.4.4 Dispositivi Mobili

La categoria dei dispositivi mobili riguarda device come smartphone e tablet che eseguono Android o iOS. Questi apparati non sono stati progettati appositamente per esperienze AR, tuttavia negli ultimi anni sono state sviluppate piattaforme come Google ARCore e Apple ARKit, che rendono possibile lo sviluppo di applicazioni AR sfruttando la sensoristica dei device.

I dispositivi mobile sfruttano una tecnologia di tipo *video see-through*. La videocamera cattura il mondo reale dal punto di vista dell'utente, in fase di elaborazione vengono aggiunte le informazioni digitali e infine il risultato viene riportato sul display del device (Figura 1.10).

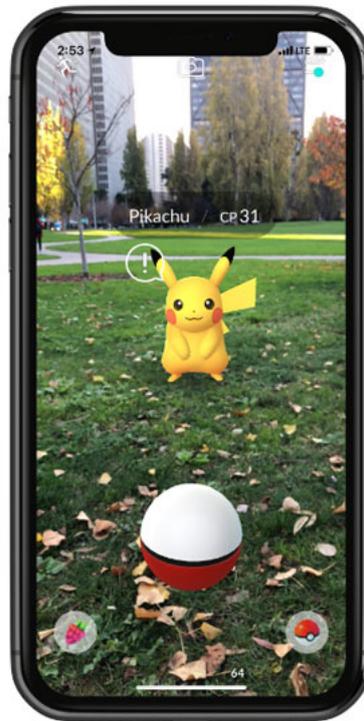


Figura 1.10: Applicazione AR in esecuzione su un dispositivo mobile.

1.5 Framework di Sviluppo MR

Per realizzare applicazioni MR è necessario considerare le piattaforme e i dispositivi di destinazione, questo perché oggi sono disponibili vari framework di sviluppo, realizzati per piattaforme specifiche o per garantire l'interoperabilità fra i diversi sistemi attualmente in commercio.

1.5.1 Microsoft Mixed Reality Toolkit

Mixed Reality Toolkit (MRTK) ¹ è un progetto gestito da Microsoft che fornisce un set di componenti e funzionalità che consentono lo sviluppo di app di realtà mista multiplatforma. Per sfruttare le funzionalità offerte da questo framework è necessario integrarlo in ambienti di sviluppo come Unity o Unreal Engine. MRTK è modulare, questo significa che è possibile installare nell'ambiente di sviluppo solo i componenti necessari per lo sviluppo dell'applicazione; in questo modo la dimensione del progetto rimane contenuta. Inoltre, poiché è costruito con oggetti scriptabili, è anche possibile sostituire i componenti

¹<https://github.com/Microsoft/MixedRealityToolkit-Unity>

inclusi con i propri, per supportare altri servizi, sistemi e piattaforme. MRTK è in continuo sviluppo, vengono introdotte nuove funzionalità e risolti problemi costantemente, allo stesso tempo alcune funzionalità vengono deprecate. MRTK viene impiegato nello sviluppo di applicazioni AR per HoloLens 1 e 2.

1.5.2 Apple ARKit

ARKit ² è la piattaforma di realtà aumentata (AR) di Apple per dispositivi iOS. Questa piattaforma di sviluppo che consente agli sviluppatori di app di creare rapidamente e facilmente esperienze AR nelle loro app e giochi. Utilizza la fotocamera, i processori e i sensori di movimento del dispositivo per creare interazioni coinvolgenti.

Per creare una corrispondenza tra spazio reale e virtuale, ARKit utilizza una tecnica chiamata Visual Inertial Odometry. Questo processo combina le informazioni provenienti dall'hardware di rilevamento del movimento del dispositivo iOS con l'analisi della visione artificiale della scena visibile dalla fotocamera del dispositivo. ARKit riconosce i feature point nell'immagine della scena, tiene traccia delle differenze nelle posizioni di tali punti tra i fotogrammi video e confronta tali informazioni con i dati di rilevamento del movimento. Il risultato è un modello ad alta precisione della posizione e del movimento del dispositivo.

Il monitoraggio del mondo analizza e comprende anche i contenuti di una scena. Vengono utilizzati metodi di ray-casting per trovare superfici del mondo reale corrispondenti a un punto nell'immagine della telecamera. Abilitando l'impostazione planeDetection nella configurazione della sessione, ARKit rileva le superfici piane nell'immagine della telecamera e ne riporta la posizione e le dimensioni. È anche possibile utilizzare i risultati ray-cast o i piani rilevati per posizionare o interagire con il contenuto virtuale nella scena.

1.5.3 Google ARCore

ARCore ³ è un framework di sviluppo software sviluppato da Google che consente di creare applicazioni di realtà aumentata. La tecnologia di tracciamento del movimento di ARCore utilizza la fotocamera del telefono per identificare i feature point e tiene traccia di come questi punti si muovono nel tempo. Con una combinazione del movimento di questi punti e delle letture dei sensori inerziali del telefono, ARCore determina sia la posizione che l'orientamento del telefono mentre si muove nello spazio.

²<https://developer.apple.com/augmented-reality/arkit/>

³<https://developers.google.com/ar/>

Oltre a identificare i punti chiave, ARCore può rilevare superfici piane, come un tavolo o il pavimento, e può anche stimare l'illuminazione media nell'area circostante. Queste capacità si combinano per consentire ad ARCore di costruire la propria comprensione del mondo che lo circonda.

La comprensione di ARCore del mondo reale consente di posizionare oggetti, annotazioni o altre informazioni in un modo che si integra perfettamente con il mondo reale. Grazie al rilevamento del movimento l'utente può spostarsi e visualizzare gli oggetti virtuali da qualsiasi angolazione.

ARCore fornisce SDK per molti degli ambienti di sviluppo più diffusi. Questi SDK forniscono API native per tutte le funzionalità AR essenziali come il rilevamento del movimento, la comprensione dell'ambiente e la stima della luminosità. Con queste funzionalità è possibile creare esperienze AR completamente nuove o migliorare le app esistenti con funzionalità AR.

1.5.4 OpenXR

OpenXR ⁴ è un'Application Program Interface (API), ad alte prestazioni, royalty-free (licenza che permette all'utente di utilizzare la risorsa senza limiti di tempo e spazio senza dover sostenere ulteriori costi) sviluppata da Khronos, la quale si occupa di fornire un accesso nativo alle piattaforme e ai dispositivi di Augmented, Mixed e Virtual Reality. Il termine XR, infatti, sta per eXtended Reality e comprende descritte nella sezione 1.2. OpenXR dal punto di vista del programmatore, consiste in un insieme di funzioni che si interfacciano con una runtime, per eseguire operazioni comunemente richieste come ad esempio: accedere al controller o allo stato della periferica utilizzata, ottenere le posizioni di tracciamento dei dispositivi di input o della testa dell'utente e inviare frame renderizzati.

OpenXR mira a semplificare lo sviluppo del software AR/VR, consentendo alle applicazioni di raggiungere una gamma più ampia di piattaforme hardware senza dover trasferire o riscrivere il codice e, successivamente, consentendo ai fornitori di piattaforme che supportano OpenXR di accedere a più applicazioni.

Senza uno standard multiplatforma, le applicazioni e gli engine VR/AR devono utilizzare le API proprietarie di ciascuna piattaforma. I nuovi dispositivi di input richiedono l'integrazione di driver personalizzati (Figura 1.11a).

OpenXR fornisce accesso multiplatforma e ad alte prestazioni direttamente alle diverse runtime dei dispositivi XR su più piattaforme. OpenXR consente l'esecuzione di applicazioni ed engine, incluso WebXR, su qualsiasi sistema che esponga le API OpenXR (Figura 1.11b).

⁴<https://www.khronos.org/OpenXR/>

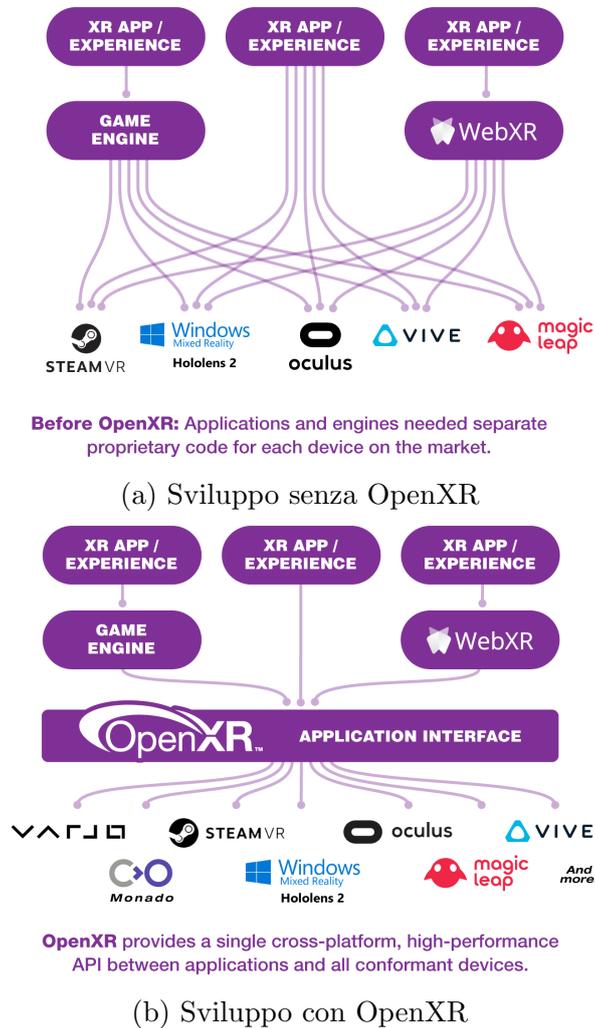


Figura 1.11: Problema della frammentazione risolto da OpenXR

1.5.5 Vuforia

Vuforia⁵ è un framework di sviluppo per applicazioni AR ed è stato introdotto da Qualcomm, un'azienda inizialmente specializzata nella ricerca e nello sviluppo dei mezzi di comunicazione wireless e dei microchip elettronici SoC.

Nel cuore della piattaforma si trova una libreria QCAR scritta in C++, che supporta vari tipi di target (image cube, cuboid, cylinder, word target e frame marker) e funzionalità di rendering dell'immagine. I target in Vuforia sono, fondamentalmente, oggetti reali, che possono essere sfruttati dalla applicazione

⁵<https://developer.vuforia.com/>

per posizionare oggetti virtuali o per condividere sistemi di coordinate con altri dispositivi.

Vuforia dispone anche di Extended Tracking: dopo il primo rilevamento il dispositivo mantiene il tracciamento anche quando il target non è più visibile, quindi una volta trovato il target l'utente è libero di muoversi nell'ambiente.

Quella che segue è una descrizione di alto livello del processo di rilevazione del target:

1. Il Tracker di Vuforia Engine riconosce il target.
2. Il tracciamento del target viene quindi inizializzato.
3. La posizione e la rotazione del target vengono analizzate per fornire una stima della posa al dispositivo.
4. Vuforia Engine trasforma la posa del target nel sistema di coordinate del dispositivo.
5. Il dispositivo assume il controllo nel momento in cui il target non è più visibile. Quando il target compare nuovamente nel campo visivo Vuforia esegue nuovamente il tracciamento.

Capitolo 2

Microsoft HoloLens 2

Il primo sistema operativo olografico al mondo è stato introdotto da Microsoft nel 2015 appositamente per dispositivi HoloLens. Originariamente noto come *Windows Holographic*, questa variante del sistema operativo Windows ha fornito una piattaforma di realtà mista che potesse essere sperimentata dagli sviluppatori e ha permesso di integrare le app universali di Windows esistenti in questo nuovo ambiente. Successivamente, Microsoft ha consentito ai visori compatibili di utilizzare un ambiente *Windows Mixed Reality* all'interno di Windows stesso, integrando parte dell'esperienza utente introdotta in precedenza da Microsoft HoloLens.

2.1 Specifiche Tecniche del Device

Come detto in precedenza, HoloLens 2 è un visore per la realtà mista annunciato da Microsoft nel Mobile World Congress del 2019. Questa nuova versione va a rimpiazzare la prima generazione di HoloLens, introdotta nel 2016. Di seguito vengono presentate le caratteristiche hardware del dispositivo (Figura 2.1), comparate anche a quelle del suo predecessore.

HoloLens 2 vanta un campo visivo ampliato e una risoluzione ottica notevolmente aumentata, passando da circa 720p a 2K di risoluzione per occhio. Ciò consente agli utenti di visualizzare ologrammi molto più dettagliati e di vederli da angoli più ampi, piuttosto che dover rimanere concentrati su punti molto specifici.

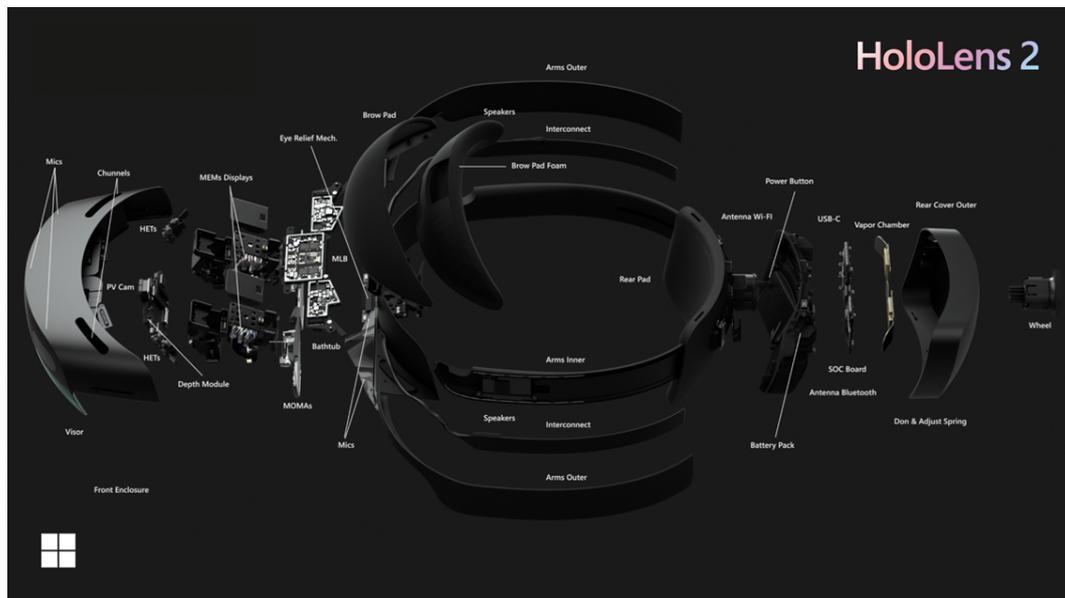


Figura 2.1: Componenti di HoloLens 2.

Display	
Optics	See-through holographic lenses (wave-guides)
Holographic resolution	2k 3:2 light engines
Holographic density	2.5k radiants (light points per radian)
Eye-based rendering	Display optimization for 3D eye position

HoloLens 2 dispone di un comparto sensoristico all'avanguardia, utilizzato per tracciare mani e occhi dell'utente e per stabilire la sua posizione nell'ambiente.

Sensors	
Head tracking	4 visible light cameras
Eye tracking	2 Infrared (IR) cameras
Depth	1-MP Time-of-Flight depth sensor
Inertial measurement unit	Accelerometer, gyroscope, magnetometer
Camera	8-MP stills, 1080p30 video

Esattamente come il display, gli speaker di HoloLens 2 sono additivi, questo significa che l'utente è in grado di percepire sia i suoni prodotti dal device che quelli provenienti dall'ambiente.

Audio	
Microphone array	5 channels
Speakers	Built-in spatial sound

Grazie al processore Snapdragon 850 e ai moduli per le comunicazioni wireless (Wi-Fi e Bluetooth), HoloLens 2 è in grado di collegarsi alla rete per navigare nel web o per partecipare a esperienze condivise con altri dispositivi.

Compute and connectivity	
System on chip	Qualcomm Snapdragon 850 Compute Platform
Holographic processing unit	Second-generation custom-built holographic processing unit
Memory	4-GB LPDDR4x system DRAM
Storage	64-GB UFS 2.1
Wi-Fi	802.11ac 2x2
Bluetooth	5.0
USB	USB Type-C DRP

Un'altra area in cui HoloLens 2 migliora rispetto al suo predecessore è l'interazione, con l'intelligenza artificiale integrata che misura il modo in cui gli utenti manipolano gli ologrammi proiettati dal visore.

Device capabilities	
Hand tracking	Two-handed fully articulated model, direct manipulation
Eye tracking	Real-time tracking
Voice	Command and control on-device; Cortana natural language with internet connectivity
Six Degrees of Freedom tracking	World-scale positional tracking
Spatial mapping	Real-time environment mesh
Mixed reality capture	Mixed hologram and physical environment photos and videos

2.2 Architettura di un'Applicazione HoloLens 2

2.2.1 Sistemi di Coordinate

Tutte le applicazioni di grafica 3D utilizzano sistemi di coordinate cartesiane per stabilire posizione e orientamento degli oggetti virtuali. Nella realtà mista, le applicazioni ragionano su sistemi di coordinate virtuali e fisici. In Windows un sistema di coordinate che ha un significato reale nel mondo fisico viene definito sistema di coordinate spaziali. I sistemi di coordinate spaziali esprimono i loro valori di coordinate in metri. Ciò significa che gli oggetti posizionati a due unità di distanza sull'asse X, Y o Z appariranno a due metri di distanza l'uno dall'altro quando renderizzati in realtà mista. Sapendo questo, è possibile eseguire il rendering di oggetti e ambienti su scala reale. In generale, i sistemi di coordinate cartesiane possono essere *right-handed* (destrorsi) o *left-handed* (sinistrorsi). I sistemi di coordinate spaziali su Windows sono sempre *right-handed*, il che significa che l'asse X positivo punta a destra, l'asse Y positivo punta verso l'alto e l'asse Z positivo punta verso il visore.

2.2.2 Ologrammi

Gli ologrammi sono entità digitali poste all'interno di un ambiente fisico e possono rappresentare elementi reali (come una sedia o un tavolo) o immaginari. HoloLens, grazie ai suoi sistemi di I/O, fornisce all'utente la possibilità di interagire con gli ologrammi come se fossero concreti. Un ologramma può anche interagire con l'ambiente circostante. Ad esempio, è possibile posizionare una palla olografica che rimbalza ed emette un suono nel momento in cui colpisce il suolo.

Gli ologrammi che vengono renderizzati da HoloLens appaiono nel frame olografico, direttamente davanti agli occhi dell'utente. Per effettuare il rendering HoloLens usa un display additivo, che aggiunge luce a una determinata area del frame, il che significa che è possibile vedere sia la luce emessa dal display che la luce del mondo circostante.

Gli ologrammi possono anche produrre suoni che sembrano provenire da un luogo specifico nell'ambiente. Su HoloLens, il suono proviene da due altoparlanti posizionati direttamente sopra le orecchie. Come il display olografico, gli altoparlanti sono additivi, quindi i nuovi suoni vengono introdotti senza occludere quelli provenienti dall'ambiente.

Gli ologrammi possono avere una posizione fissa all'interno dell'ambiente o seguire l'utente. Nel primo caso è anche possibile utilizzare le ancore per rendere la posizione degli ologrammi persistente anche attraverso diverse sessioni. Alcuni scenari invece richiedono che gli ologrammi rimangano facilmente individuabili o visibili durante la sessione. Ci sono due approcci di alto livello per questo tipo di posizionamento, chiamati *display-locked* e *body-locked*.



Figura 2.2: Grazie ad HoloLens l'utente può interagire con gli ologrammi

Nel caso del *display-locked* i contenuti vengono fissati sul display, questo approccio crea delle complicazioni, infatti il campo visivo dell'utente viene limitato.

Con l'approccio *body-locked* l'ologramma si lega al corpo dell'utente, in questo modo è possibile muoversi e spostarsi senza perderlo di vista e allo stesso tempo il campo visivo rimane inalterato.

2.2.3 Mappatura Spaziale

La mappatura spaziale genera un modello 3D dell'ambiente chiamato *mesh* (Figura 2.3), consentendo di posizionare oggetti virtuali su superfici reali. Uno degli usi principali della *mesh* creata dalla mappatura spaziale è quello di occludere gli ologrammi, che vengono percepiti come se fossero reali.

L'occlusione fornisce anche informazioni all'utente; quando un ologramma viene occluso da una superficie del mondo reale, questo fornisce un feedback visivo aggiuntivo sulla sua posizione in quell'ambiente. Inoltre, l'occlusione può anche nascondere in modo utile le informazioni all'utente; occludere ologrammi dietro i muri può ridurre il disordine visivo.

La mappatura spaziale è utile anche nel momento in cui si vuole applicare qualsiasi tipo di forza fisica agli ologrammi, sfruttando la *mesh* infatti è pos-

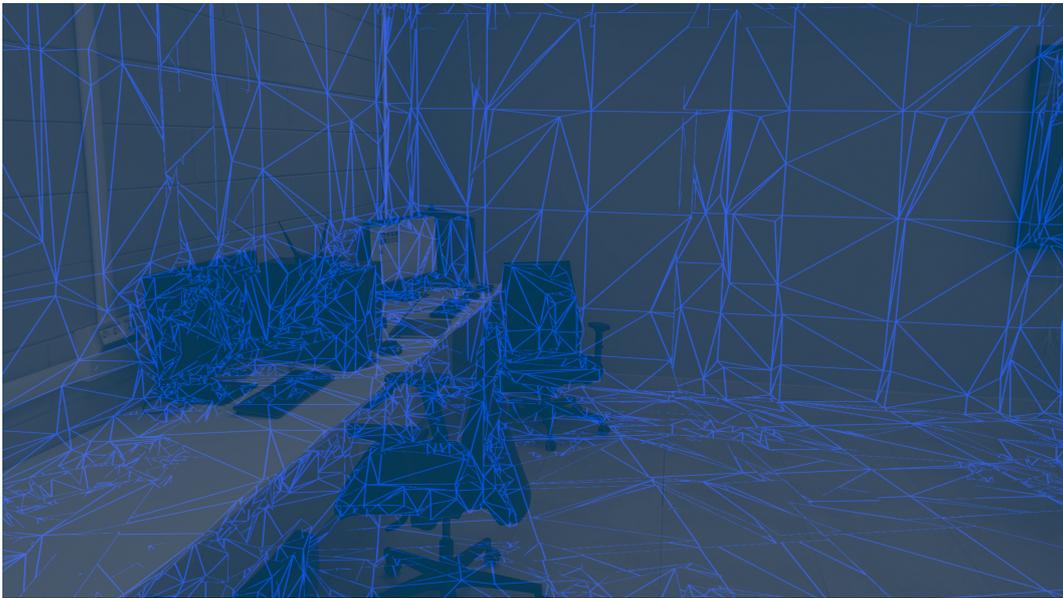


Figura 2.3: Mesh di una stanza, generata da HoloLens 2.

sibile farli collidere con gli ostacoli presenti nell'ambiente. In alcuni casi, ad esempio se sono presenti superfici in vetro o riflettenti, la mesh potrebbe essere imprecisa e non corrispondere con l'ambiente, di conseguenza le collisioni degli ologrammi potrebbero essere inaccurate.

L'archiviazione, il rendering e l'elaborazione delle mesh possono consumare notevoli risorse di calcolo e di archiviazione. Pertanto, ogni applicazione dovrebbe adottare uno schema di caching appropriato alle proprie esigenze, determinando quali mesh mantenere, quali scartare e quando aggiornare quelle salvate.

2.2.4 Ancore

Un'ancora rappresenta un punto nel mondo del quale il sistema tiene traccia nel tempo. Ogni ancora ha un sistema di coordinate regolabile, basato su altre ancore o sistemi di riferimento, per garantire che gli ologrammi ancorati rimangano esattamente al loro posto. È anche possibile mantenere e condividere ancore tra le sessioni dell'applicazione e tra diversi dispositivi:

Salvando le ancore su disco e caricandole di nuovo in un secondo momento, l'applicazione può calcolare la stessa posizione nel mondo reale in più sessioni dell'applicazione su un singolo HoloLens.

Usando i servizi Azure è possibile condividere un'ancora in cloud con più dispositivi HoloLens, iOS e Android. Facendo in modo che ogni dispositivo

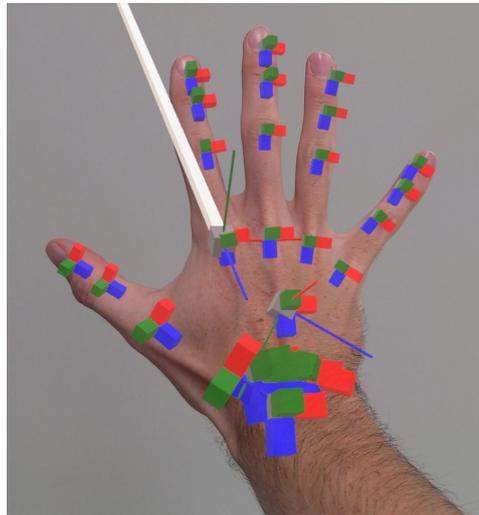
esegua un ologramma utilizzando la stessa ancora spaziale, gli utenti vedranno l'ologramma apparire nello stesso posto nel mondo reale. Ciò consente esperienze condivise in tempo reale.

Le ancore condivise in cloud possono persistere anche in modo asincrono, quindi i vari dispositivi presenti nell'ambiente possono osservare e interagire con lo stesso ologramma in momenti diversi.

Gli ologrammi che vengono ancorati purtroppo non possono essere spostati o ruotati, quindi ogni volta che si desidera interagire con un ologramma ancorato bisogna prima eliminare l'ancora, per poi crearla nuovamente nel momento in cui termina l'interazione. Questo pone un forte limite alle esperienze condivise che sfruttano questa funzionalità; infatti se in un ambiente sono presenti più utenti e uno di questi interagisce con un ologramma ancorato gli altri non sono in grado di vedere la fase d'interazione.

2.2.5 Hand Tracking e Gaze

La manipolazione diretta è un modello di input primario su HoloLens 2, che usa il nuovo sistema di rilevamento della mano. HoloLens 2 è in grado di riconoscere mani e dita dell'utente (Figura 2.4a) e di rilevare le collisioni con gli ologrammi, quindi è possibile interagire con questi come se fossero reali. È anche possibile sfruttare l'indice della mano come puntatore 3D (gaze, Figura 2.4b), in questo modo è possibile interagire con gli ologrammi più distanti senza doversi spostare; solitamente in un'applicazione HoloLens 2 i gaze sono due, uno per ogni mano. Il gaze può anche essere utilizzato sfruttando gli occhi, lasciando così le mani libere. Grazie a questa funzionalità l'utente può "fissare" l'ologramma alla testa, di conseguenza l'oggetto si muoverà e ruoterà nell'ambiente seguendo la testa dell'utente. Le possibili interazioni con gli ologrammi sono varie, ad esempio sfruttando una mano sola si può puntare il gaze su un ologramma, unire indice e pollice (air tap) e spostare la mano nella direzione in cui si vuole posizionare l'ologramma. Se invece si vuole ridimensionare un ologramma è necessario usare tutte e due le mani, in questo caso bisogna puntare entrambi i gaze sull'ologramma ed eseguire la gesture air tap, allargando o avvicinando le mani l'ologramma si ridimensionerà. Un altro tipo di interazione particolarmente interessante è la rotazione dell'oggetto, per ruotare un ologramma basta eseguire la stessa procedura descritta precedentemente per il movimento, solo che invece di spostare la mano bisogna ruotarla esattamente nel modo in cui si desidera ruotare l'oggetto.



(a) Hand tracking.



(b) Gaze.

2.3 Strumenti di Sviluppo

Per lo sviluppo di applicazioni HoloLens 2 esistono varie opzioni che si distinguono in base agli obiettivi e alle necessità degli sviluppatori:

- Unity è la piattaforma leader a livello mondiale per la creazione e l'utilizzo di contenuti 3D interattivi in tempo reale; il codice di runtime sottostante è C++, mentre lo scripting di sviluppo viene eseguito in C#.
- Unreal Engine 4 è un engine open source avanzato con supporto completo per la realtà mista sia in C++ che in Blueprints.
- Gli sviluppatori nativi che hanno esperienza nella scrittura di renderer 3D personalizzati possono creare un engine custom grazie ad OpenXR.
- Gli sviluppatori Web che vogliono creare esperienze Web AR/VR per browser web possono usare WebXR.

Per il caso di studio si è scelto di utilizzare Unity in combinazione con il Mixed Reality Toolkit (MRTK) fornito da Microsoft, questo perché al momento è l'opzione migliore, sia dal punto di vista delle funzionalità offerte, che dalla stabilità.

MRTK per Unity è un progetto gestito da Microsoft che fornisce un set di componenti e funzionalità che consentono lo sviluppo di app di realtà mista multiplatforma.

MRTK è modulare, questo significa che è possibile installare in Unity solo i componenti necessari per lo sviluppo dell'applicazione; in questo modo la dimensione del progetto rimane contenuta. Inoltre, poiché è costruito con oggetti scriptabili, è anche possibile sostituire i componenti inclusi con i propri, per supportare altri servizi, sistemi e piattaforme.

MRTK è in continuo sviluppo, vengono introdotte nuove funzionalità e risolti problemi costantemente, allo stesso tempo alcune funzionalità vengono deprecate, in base anche alle nuove versioni di Unity che vengono rilasciate. Ad oggi è possibile scegliere fra quattro opzioni, che si distinguono prevalentemente per la versione di Unity che si vuole utilizzare:

- L'attuale configurazione consigliata da Microsoft per HoloLens 2 e per lo sviluppo di Windows Mixed Reality è Unity 2020.3 LTS con il plug-in OpenXR più recente.
- Con Unity 2019 è possibile utilizzare plug-in XR integrato nell'ambiente di sviluppo.
- Con Unity 2021.2 il plug-in Windows XR è stato deprecato in favore di OpenXR, che quindi rimane l'unica opzione disponibile.
- Per Unity 2018.4 LTS è terminato il supporto di due anni e non vengono più rilasciati aggiornamenti.

I componenti del MRTK sono molti e offrono funzionalità di vario tipo: percezione ambiente, gestione dei sistemi di input e gestione delle ancore sono alcuni esempi. La gestione di questi componenti all'interno di Unity viene affidata a dei profili specifici per ogni funzionalità.

2.3.1 Sistema di Input

Il sistema di input in MRTK consente di:

- Rilevare input da una varietà di sorgenti, come controller 6DOF, mani o voce.
- Definire azioni astratte, come Select o Menu e associarle a diversi input.
- Impostare puntatori collegati ai controller per guidare i componenti dell'interfaccia utente tramite eventi del puntatore (focus e tap).

Gli input sono prodotti dagli Input Data Providers (Device Manager). Ogni provider corrisponde a una particolare fonte di input: Open VR, Windows Mixed Reality (WMR), Unity Joystick o Windows Speech, I provider vengono aggiunti al progetto tramite il profilo Registered Service Providers nel componente Mixed Reality Toolkit e producono eventi di input automaticamente quando sono disponibili le sorgenti di ingresso corrispondenti (ad esempio quando viene rilevato un controller WMR o collegato un gamepad).

Le azioni di input sono astrazioni intese a isolare la logica dell'applicazione dalle specifiche sorgenti di input. Può essere utile, ad esempio, definire un'azione Select e mapparla sul pulsante sinistro del mouse, un pulsante in un gamepad e un trigger in un controller 6DOF. È quindi possibile fare in modo che la logica dell'applicazione ascolti gli eventi dell'azione Seleziona in input invece di dover essere consapevole di tutti i diversi input che possono produrlo. Le azioni di input sono definite nel profilo delle azioni di input, che si trova all'interno del profilo Input System nel componente Mixed Reality Toolkit.

I controller vengono creati dai provider di input quando i dispositivi di input vengono rilevati e distrutti quando vengono persi o disconnessi. Il provider di input WMR, ad esempio, crea un controller WMR per dispositivi 6DOF e controller manuali WMR per le mani. Gli eventi di input generati dai controller includono l'eventuale azione di input associata.

I controller possono avere dei Pointer collegati che interrogano la scena per determinare l'oggetto di gioco con il focus e generare Pointer Event su di esso. Ad esempio, il Pointer che gestisce il gaze esegue un raycast sulla scena utilizzando la posizione del controller per calcolare l'origine e la direzione del raggio. I Pointer creati per ogni controller sono impostati nel profilo Pointer, sotto il profilo Input System.

2.3.2 Percezione Ambientale

Il sistema di percezione ambientale (Spatial Awareness) fornisce la consapevolezza ambientale del mondo reale nelle applicazioni di realtà mista. Introdotta in un'applicazione HoloLens, Spatial Awareness fornisce una mesh, che rappresenta la geometria dell'ambiente, così da consentire interazioni tra gli ologrammi e il mondo reale. Il sistema di percezione spaziale è simile al sistema di input descritto precedentemente, in quanto i Data Provider forniscono al sistema dati su una mesh relativa al mondo reale. Il profilo Spatial Awareness deve avere almeno uno Spatial Observer registrato. Gli Spatial Observer sono generalmente componenti specifici della piattaforma che fungono da provider per fornire vari tipi di dati relativi alla mesh generata da un endpoint specifico (ad esempio HoloLens). Il profilo Spatial Awareness permette di gestire vari aspetti per quanto riguarda la generazione della mesh, ad esempio è pos-

sibile definire la frequenza di aggiornamento (che tipicamente varia fra 0.1 e 0.5 secondi). L'Observer può essere impostato come stazionario, ovvero viene generata solo la porzione di mesh relativa al punto dove si è avviata l'applicazione. L'Observer ha una forma che definisce il tipo di volume della mesh che verrà utilizzato. Le opzioni supportate sono:

- Cubo allineato agli assi: forma rettangolare che rimane allineata con gli assi del sistema di coordinate globali, come determinato all'avvio dell'applicazione.
- Cubo allineato dall'utente: forma rettangolare che ruota per allinearsi al sistema di coordinate locale dell'utente.
- Sfera: Un volume sferico con centro all'origine del mondo.

2.3.3 Definizione dei Confini

Il Boundary System fornisce il supporto per far visualizzare all'utente i limiti imposti dal mondo reale. I confini definiscono l'area in cui gli utenti possono muoversi in sicurezza e sono importanti per aiutarli a evitare ostacoli mentre indossano un visore AR.

Molte piattaforme di realtà virtuale forniscono una visualizzazione automatica, ad esempio un contorno bianco sovrapposto al mondo virtuale quando l'utente o il suo controller si avvicina al confine. Il Boundary System estende questa funzionalità per consentire di visualizzare un contorno dell'area tracciata da HoloLens e fornisce altre funzionalità che possono essere utilizzate per informare gli utenti sugli ostacoli presenti nell'ambiente.

2.4 Esempio di Applicazione HoloLens 2

Di seguito viene presentata un'applicazione sviluppata in PSLab durante la fase di studio della documentazione relativa ad HoloLens 2. L'applicazione in questione è stata sviluppata con l'obiettivo di testare il sistema di gestione delle ancore, i sistemi di coordinate e le collisioni degli ologrammi con la mesh.

2.4.1 Configurazione del Progetto Unity

Esattamente come per lo sviluppo di un videogioco, è necessario inizializzare un progetto Unity. Una volta creato il progetto è necessario importare i componenti del MRTK necessari, in modo da poter implementare le funzionalità desiderate successivamente. Per lo sviluppo di questa applicazione sono

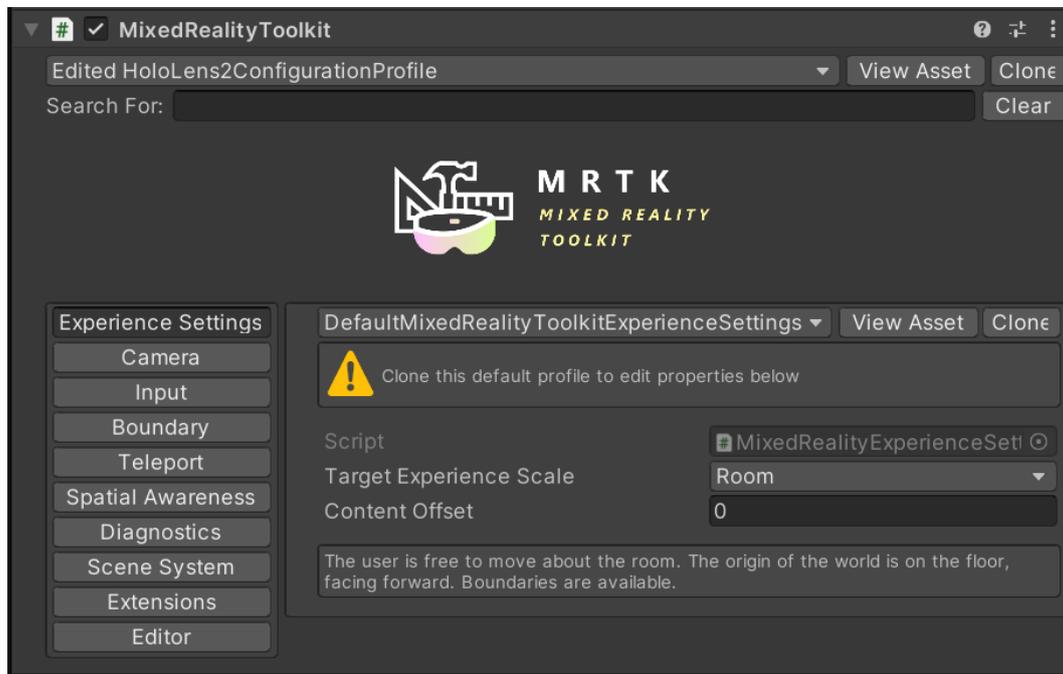


Figura 2.5: Profili MRTK.

stati utilizzati Unity 2019 e MRTK 2.5.3 ed è stato importato il componente Mixed Reality Toolkit Foundation. A questo punto, in Unity è possibile accedere alla sezione dei profili per HoloLens 2 (Figura 2.5).

In questo caso sono state abilitate le funzionalità dei profili Spatial Awareness e Boundary System ed è stata occlusa la vista della mesh. In questo modo durante la sessione di utilizzo HoloLens 2 si occupa di mappare l'ambiente senza che l'utente se ne renda conto.

Per il posizionamento degli oggetti della scena sono stati importati in Unity il pacchetto TextMeshPro e il pacchetto Tutorial Assets di Microsoft, che offre diversi asset, utili per lo sviluppo.

2.4.2 Funzionamento dell'Applicazione

Una volta importati i package correttamente è possibile aggiungere gli oggetti nella scena (Figura 2.6). In questa applicazione sono stati aggiunti:

- Un cubo (**A1**) ancorato alla mesh e un modello del Rover Curiosity (**A2**) della NASA come figlio.
- Un cubo (**B1**) disposto su un piano (**B2**), al quale viene applicata la forza di gravità.

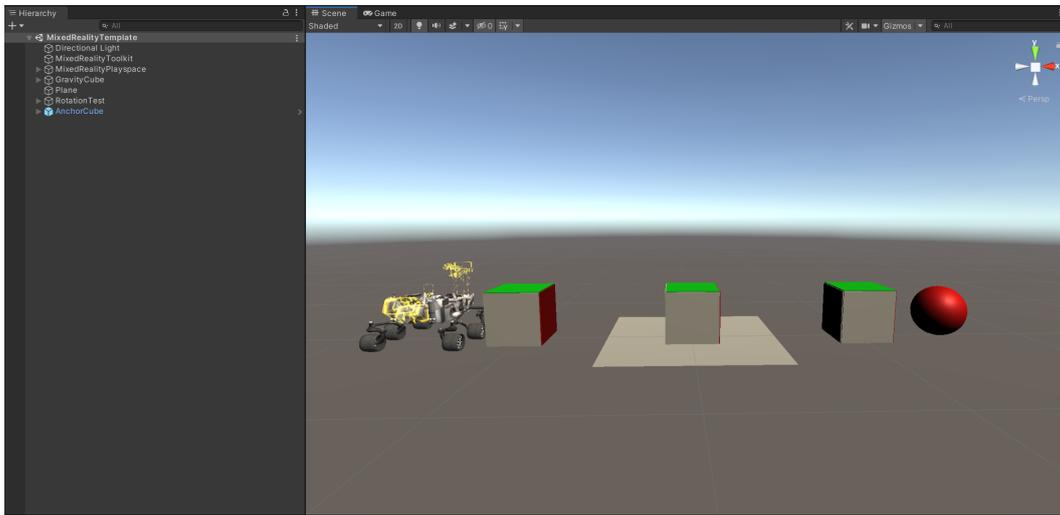


Figura 2.6: Scena di Unity.

- Un cubo (**C1**) non ancorato alla mesh con una sfera (**C2**) come figlio che gli ruota attorno.

Per abilitare la manipolazione degli ologrammi vengono aggiunti gli script `Object Manipulator` e `NearInteractionGrabbable` a tutti gli oggetti della scena, tranne che al piano **B2**. Questi due script fanno parte dei componenti del MRTK. `Object Manipulator` permette di manipolare un ologramma sfruttando il gaze, così è possibile interagire con gli ologrammi anche se sono distanti. `NearInteractionGrabbable` abilita le interazioni da vicino, permettendo di manipolare gli ologrammi prendendoli come se fossero oggetti reali.

Sfruttando il `Boundary System` e il componente `Spatial Mapping Collider` applicato al cubo **B1** è possibile abilitare le collisioni con la mesh. Per applicare la forza di gravità è stato anche aggiunto il componente `Rigid Body`. Il risultato è che nel momento in cui l'utente prende **B1**, spostandolo dal piano **B2** e lasciandolo cadere a terra, **B1** colliderà con la mesh generata da HoloLens 2, ovvero con il pavimento. Senza la presenza della mesh **B1** oltrepasserebbe il pavimento e scomparirebbe dalla scena.

In Unity, impostando l'oggetto **C2** come figlio dell'oggetto **C1**, le coordinate di **C2** saranno stabilite sulla base di quelle di **C1**. Durante l'utilizzo dell'applicazione, spostando **C1**, **C2** si muoverà di conseguenza mantenendo la rotazione attorno a **C1**, non vale il contrario.

Per quanto riguarda la gestione delle ancore è stato sfruttato il `World Anchor Store`, una funzionalità del MRTK che permette di salvare la posizione di un ologramma all'interno di un ambiente (su un solo dispositivo). È stata associata un'ancora al cubo **A1**, che viene eliminata nel momento in cui l'utente

interagisce con l'ologramma e ricrea al termine dell'interazione. L'ancora viene salvata nel World Anchor Store, che si occupa di renderla persistente e disponibile attraverso diverse sessioni. La posizione di **A2** è comunque sempre relativa a quella di **A1**; quindi, attraverso sessioni differenti, **A1** manterrà posizione e rotazione, mentre **A2** assumerà posizione e rotazione stabilite in Unity relativamente ad **A1**.

Per la gestione delle ancore è stato sviluppato uno script che una volta associato a un oggetto permette di salvare (listato 4.3) ed eliminare (listato 4.2) l'ancora associata.

```
1  public void DeleteExistingAnchor()  
2  {  
3      WorldAnchor anchor =  
4          rootGameObject.GetComponent<WorldAnchor>();  
5      if (anchor != null)  
6      {  
7          Destroy(anchor);  
8          worldAnchorStore.Delete(rootGameObject.name);  
9          this.savedRoot = false;  
10     }  
11 }
```

Listato 2.1: Metodo richiamato per eliminare l'ancora dell'ologramma associato.

```
1  public void SaveAnchor()  
2  {  
3      WorldAnchor anchor =  
4          rootGameObject.AddComponent<WorldAnchor>();  
5      if (!this.savedRoot && anchor != null)  
6      {  
7          this.savedRoot =  
8              this.worldAnchorStore.Save(rootGameObject.name,  
9              anchor);  
10     }  
11 }
```

Listato 2.2: Metodo richiamato per salvare l'ancora dell'ologramma associato.

Per ogni sessione, prima di utilizzare le ancore del World Anchor Store è necessario richiedere il caricamento in modo asincrono di quest'ultimo (listato 4.1). Una volta caricato il World Anchor Store è possibile caricare le ancore salvate nelle sessioni precedenti (listato 4.4).

```
1  void Start()
2  {
3      WorldAnchorStore.GetAsync(StoreLoaded);
4  }
5
6  private void StoreLoaded(WorldAnchorStore store)
7  {
8      worldAnchorStore = store;
9      LoadAnchor();
10 }
```

Listato 2.3: Metodo richiamato una volta caricato il World Anchor Store.

```
1  private void LoadAnchor()
2  {
3      this.savedRoot =
4          this.worldAnchorStore.Load(rootGameObject.name,
5          rootGameObject);
6  }
```

Listato 2.4: Metodo richiamato per reperire l'ancora dal World Anchor Store.

Capitolo 3

Esempi di Applicazioni per HoloLens 2

I possibili settori di applicazione di HoloLens 2 sono svariati. In generale, in qualsiasi ambiente la realtà mista può essere introdotta per aumentare produttività, efficienza e sicurezza.

Di seguito vengono riportati alcuni ambiti in cui l'introduzione di HoloLens 2 ha giocato un ruolo fondamentale per quanto riguarda gli aspetti appena descritti.

3.1 Ambito Industriale

Per quanto riguarda l'ambito industriale si può fare riferimento ai partner Microsoft, come Airbus. Airbus è un pioniere nella tecnologia aerospaziale e leader nella progettazione e produzione di aerei commerciali e militari, satelliti e veicoli di lancio, si è impegnata a trasformare i processi industriali tradizionali attraverso l'uso della realtà mista. L'azienda ha stretto una partnership con Microsoft per utilizzare la realtà mista di Azure e HoloLens 2 come mezzo per accelerare la progettazione e la produzione di aeromobili, aumentando al contempo la sicurezza e la funzionalità e garantendo opportunità di sviluppo professionale per i dipendenti. Servizi intelligenti come Azure Remote Rendering stanno cambiando il modo in cui vengono comunicate le idee complesse. La capacità di creare applicazioni multiutente con riconoscimento dello spazio con Azure Spatial Anchors promuove una maggiore collaborazione aumentando la qualità, la sicurezza e la protezione. Airbus sta usando la realtà mista di Azure per sbloccare tutto il potenziale di HoloLens 2 e, di conseguenza, prevede di ridurre i tempi di convalida del progetto dell'80% e di accelerare le attività complesse durante l'assemblaggio del 30%.

Airbus ha impiegato 40 anni per costruire i suoi primi 10.000 velivoli. Nei prossimi 20 anni, il gigante aerospaziale mira a costruirne altri 20.000, una sfida formidabile che richiederà innovazioni all'avanguardia.

Jean-Brice Dumont, Executive Vice President of Engineering di Airbus, afferma:

“La nostra sfida nei prossimi anni è quella di produrre più velivoli più velocemente, e per questo dobbiamo consentire ai nostri lavoratori di essere molto meglio equipaggiati e di essere molto più efficaci in quello che fanno. Dobbiamo alzare il tiro. Per affrontare questa sfida, intendiamo fare un uso intenso della realtà mista ed è per questo che abbiamo stretto una partnership con Microsoft. La realtà mista può aiutarci ad aumentare la qualità, la sicurezza e la protezione. Il livello di errore umano è significativamente ridotto e, nel settore aerospaziale, una maggiore qualità è una maggiore sicurezza”.

La tecnologia Microsoft Mixed Reality può essere utilizzata per aiutare gli addetti alla produzione di Airbus ad accedere a informazioni e istruzioni mentre hanno le mani occupate, ad esempio. Può anche facilitare la formazione senza impegnare attrezzature costose o persino richiedere al tirocinante di recarsi nel luogo in cui si trova l'attrezzatura. Questo è solo l'inizio: Airbus ha identificato più di 300 casi d'uso per la realtà mista.

Airbus ha ottenuto risultati impressionanti nelle sue prove e implementazioni della tecnologia di realtà mista Microsoft nella formazione, nella progettazione e nella produzione.

La realtà mista consente ai tirocinanti aerospaziali di apprendere in un ambiente virtuale immersivo senza la necessità di un vero aereo fisico o di parti. Questo ambiente 3D può offrire funzionalità che l'allenamento nella vita reale non può offrire, come la possibilità di visualizzare gli elementi in tre dimensioni da qualsiasi angolazione.

HoloLens aiuta i progettisti di Airbus a testare virtualmente i loro progetti per vedere se sono pronti per la produzione. La realtà mista accelera notevolmente il processo, riducendo il tempo impiegato dell'80%.

La tecnologia della realtà mista può anche aiutare i lavoratori della linea di produzione ad accedere a informazioni cruciali mantenendo le mani libere. Le informazioni digitali, come istruzioni o diagrammi, possono essere sovrapposte a un vero macchinario per aiutare in attività complesse o difficili. Questi tipi di soluzioni di realtà mista hanno consentito ad Airbus di ridurre di un terzo i tempi di produzione migliorando al contempo la qualità.

3.2 Ambito Sanitario

In ambito sanitario sono stati prodotti vari articoli scientifici che, grazie ad HoloLens, propongono soluzioni per migliorare la precisione e l'efficienza delle operazioni e per ridurre l'esposizione degli operatori alle radiazioni.

La tecnologia olografica di HoloLens 2 può essere utile per interventi guidati da CT ¹ rendendoli più sicuri ed efficienti. In questo articolo [4] È stato eseguito uno studio prospettico per valutare il direzionamento dell'ago guidato da CT su un fantoccio addominale (Figura 3.1a) con e senza guida AR (rispettivamente Figura 3.1d e Figura 3.1c). L'allineamento è stato eseguito automaticamente su una griglia TC utilizzata di routine nella pratica clinica, invece di utilizzare marker esterni basati su immagini, richiesti invece in altri sistemi.

Hanno partecipato in totale otto operatori con diversa esperienza clinica che hanno eseguito un totale di 86 passaggi con ago. L'efficienza della procedura, la dose di radiazioni e i tassi di complicanze sono stati confrontati con e senza guida AR.

In Figura 3.1d l'allineamento con il mondo reale appare accurato, le linee della griglia CT reali sono allineate con le linee della griglia virtuale. L'ago è allineato con la guida virtuale (linea viola) che mostra la traiettoria ideale verso la lesione selezionata (pallina verde).

Il numero medio totale di passaggi per raggiungere l'obiettivo è stato ridotto da 7,4 passaggi senza AR a 3,4 passaggi con AR (riduzione del 54,2%). Il tasso di errore, introdotto nel colpire una lesione non mirata è diminuito dall'11,9% senza AR (7/59 passaggi con ago) allo 0% con AR (0/27 passaggi con ago). I passaggi con AR erano più allineati con la traiettoria ideale del bersaglio rispetto a quelli senza AR (offset rispettivamente di 4,6° e 8,0°).

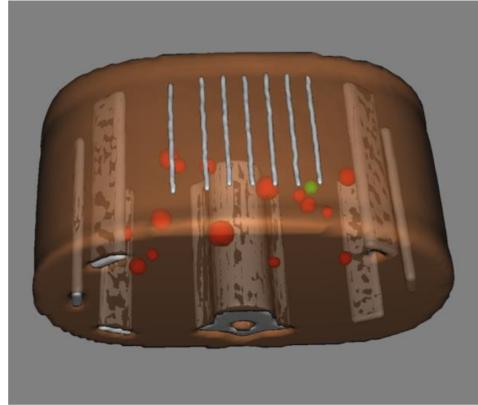
La guida 3D AR può fornire miglioramenti significativi nell'efficienza procedurale e nel risparmio della dose di radiazioni per individuare lesioni difficili e fuori dal piano. La guida AR ha elevato le prestazioni di tutti gli operatori allo stesso livello indipendentemente dalla precedente esperienza clinica. I risultati chiave dei test effettuati sono i seguenti:

- I passaggi totali con ago e la dose di radiazioni sono significativamente ridotti utilizzando la realtà aumentata.
- Il tasso di errore, introdotto nel colpire una lesione non mirata viene eliminato utilizzando la realtà aumentata.
- La realtà aumentata ha elevato le prestazioni di tutti gli operatori allo stesso livello indipendentemente dalla precedente esperienza clinica.

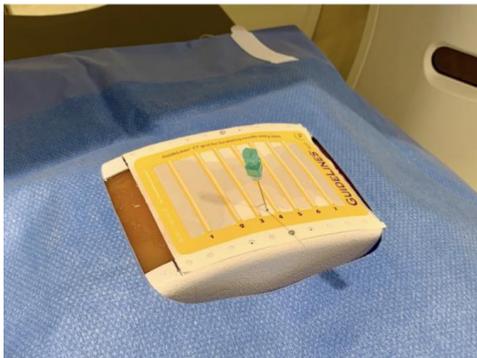
¹La tomografia computerizzata (in inglese computed tomography) è un processo che visualizza le informazioni anatomiche da un piano di sezione trasversale del corpo.



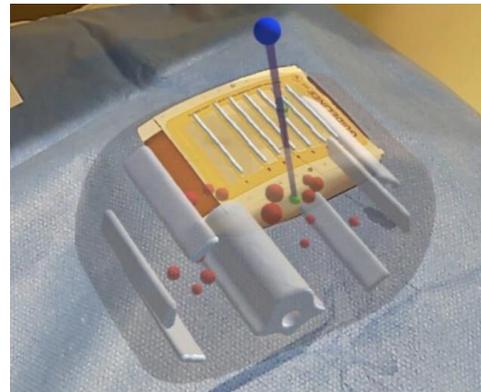
(a) Griglia CT applicata alla superficie del modello.



(b) Modello 3D del fantoccio.



(c) Inserimento dell'ago senza AR.



(d) Inserimento dell'ago con modello tridimensionale e guida dell'ago virtuale proiettati sul fantoccio tramite HoloLens 2.

3.3 Ambito Educativo

Negli ultimi anni, il settore dell'istruzione ha incorporato nelle classi l'uso di nuove tecnologie e dispositivi informatici, che hanno consentito di implementare nuovi modi per migliorare l'insegnamento e l'apprendimento. La realtà aumentata può essere impiegata per aiutare gli insegnanti a promuovere l'interesse degli studenti nell'apprendimento di determinate materie e concetti astratti in nuove modalità visive. Questo articolo [6] propone di sfruttare il potenziale di HoloLens al fine di integrare esperienze AR nelle lezioni e nei laboratori pratici. Nello specifico, viene proposta un'architettura per fornire servizi didattici AR a bassa latenza in un'aula o in un laboratorio. Una la-

tenza così bassa è ottenuta grazie all'uso di dispositivi di edge computing, che esonerano il cloud dalle attività tradizionali richieste dalle applicazioni AR dinamiche (ad esempio, elaborazione dei dati in tempo reale, comunicazioni tra dispositivi AR). A seconda dell'applicazione AR specifica e del numero di utenti, il collegamento wireless (di solito WiFi) potrebbe essere sovraccaricato se la rete non è stata progettata correttamente e le prestazioni complessive dell'applicazione possono essere compromesse, portando a un'elevata latenza e persino a problemi di comunicazione. Per affrontare questo problema, le misurazioni del canale radio e i risultati della simulazione sono stati eseguiti per mezzo di uno strumento di ray-launching 3D sviluppato internamente, che è in grado di modellare e simulare il comportamento di un'aula/laboratorio abilitato per AR in termini di propagazione radio e qualità del servizio.

La Figura 3.2 illustra l'architettura di comunicazione del sistema proposto. In fondo è presente il livello AR-IoT, che è diviso in due sottolivelli: il sottolivello dei dispositivi AR, che è composto dai dispositivi AR utilizzati dagli studenti e dall'insegnante, e il sottolivello dei dispositivi IoT (ovvero, oggetti smart come sensori e attuatori) che sono presenti nell'ambiente. Entrambi i dispositivi AR e IoT sono collegati tramite un punto di accesso WiFi (AP) o un gateway al livello di edge computing, dove vengono eseguiti i servizi educativi AR. Nello specifico, lo strato edge computing è conformato da due sottolivelli:

- **AR Service Sublayer:** utilizza dispositivi di edge computing come cloudlet ² o fog di gateway ³ basati su single board computer (SBC) ⁴ per fornire risposte veloci e distribuite. Gli insegnanti possono agire come amministratori locali dei dispositivi su questo livello per controllare quali contenuti vengono mandati ai dispositivi AR degli studenti. Inoltre, i dispositivi di edge computing di questo livello consentono la condivisione di esperienze AR, potendo così sincronizzare il contenuto condiviso contemporaneamente tra più utenti.
- **AR-IoT Service Sublayer:** consente di interconnettere dispositivi AR e IoT senza soluzione di continuità. Il sistema presentato utilizza Message Queue Telemetry Transport (MQTT) ⁵ e un'API (Application Programming Interface) REpresentational State Transfer (REST) per facilitare

²È un data center cloud su piccola scala ottimizzato per la mobilità che si trova ai "marginii" di Internet.

³È un'architettura che utilizza dispositivi periferici per eseguire una notevole quantità di calcolo, archiviazione e comunicazione localmente per poi instradarla su Internet.

⁴È un computer completo costruito su un unico circuito, con microprocessore, memoria, input/output e altre caratteristiche richieste da un computer.

⁵È un protocollo di rete leggero di tipo publish-subscribe, che trasporta i messaggi tra i dispositivi.

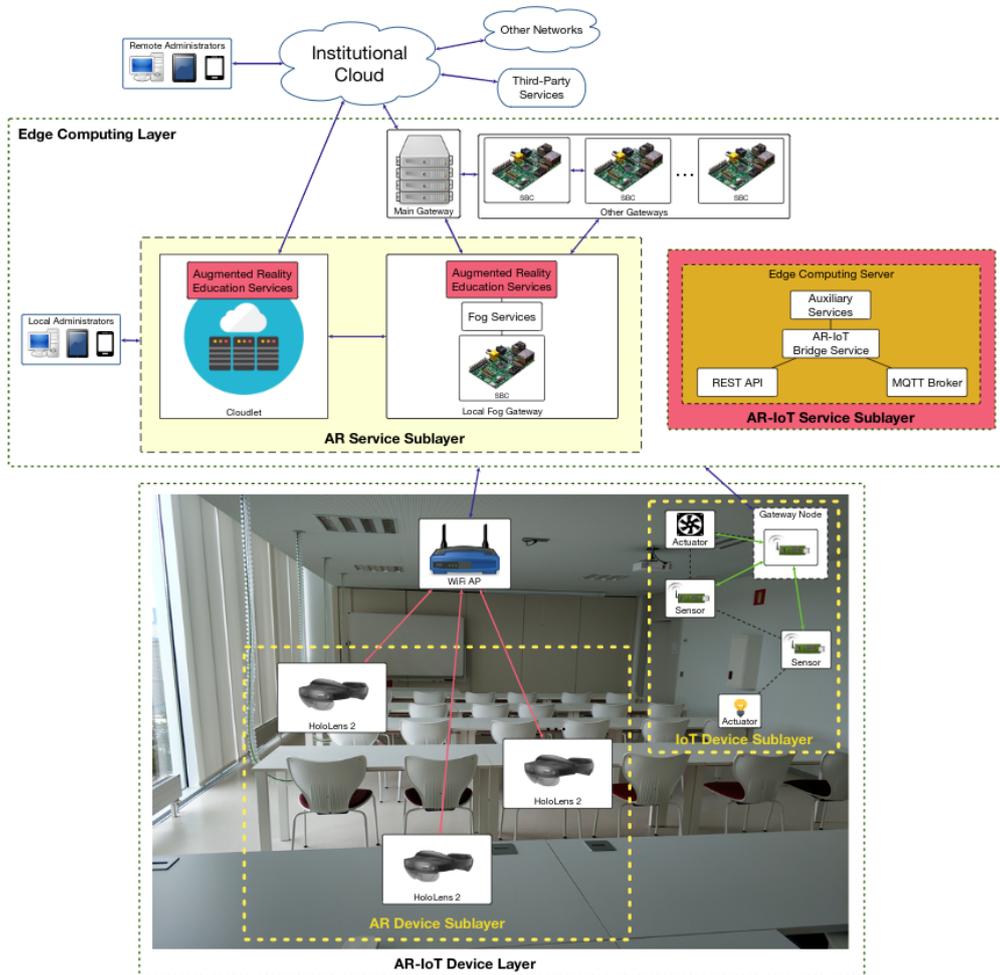


Figura 3.2: Architettura di comunicazione del sistema proposto.

gli scambi di dati tra più dispositivi eterogenei. Per quanto riguarda il servizio AR-IoT Bridge, si tratta di un componente software che collega le richieste MQTT e API REST, avvalendosi di determinati servizi ausiliari quando necessario.

Entrambi i livelli consentono di implementare le funzionalità più avanzate del sistema:

- L'insegnante è in grado di controllare da remoto quando e come il contenuto AR viene visualizzato sui visori HoloLens 2 utilizzati dagli studenti.
- Gli studenti possono condividere la stessa esperienza AR e quindi collaborare tra loro per raggiungere un obiettivo comune.

- Il sistema sviluppato può interagire con gli oggetti IoT circostanti, consentendo agli utenti AR di ottenere informazioni o di agire con essi.
- Poiché l'applicazione HoloLens 2 rileva le azioni eseguite da ogni studente, è possibile registrarle e quindi determinare le prestazioni dello studente o i suoi contributi durante le attività di collaborazione.

3.4 Ambito Museale

Un altro ambito di applicazione interessante è quello del museo. Infatti è possibile sviluppare applicazioni AR utili per guidare l'utente all'interno del museo. Entrando più nello specifico, si può proporre una guida virtuale che, oltre a guidare l'utente, permetta di interagire con i modelli 3D relativi agli artefatti presenti all'interno del museo. Una soluzione del genere viene proposta in questo articolo [2], dove è stata utilizzata la prima versione di HoloLens. Questa procedura mira a creare un quadro di orientamento semplice, interattivo e informativo da utilizzare per gli utenti dei musei. L'applicazione MR richiede all'utente di indossare Microsoft HoloLens ed esplorare una serie di contenuti virtuali attraverso l'interfaccia utente. Un ambiente popolato da reperti museali è necessario per sovrapporre virtualizzazioni digitali e informazioni affinché il sistema di guida virtuale del tour possa funzionare. L'applicazione proposta (Figura 3.3) fornisce le seguenti funzionalità:

- Controllo degli ologrammi relativi ai manufatti museali, con possibilità di ruotarli di 360 gradi.
- Pulsanti che avviano testo e immagini per permettere all'utente di ricevere informazioni.
- Interazione con il sistema di guida virtuale, che presenta informazioni audio e video in tempo reale.
- Utilizzo di piccoli cerchi che funzionano come oggetti trigger per rivelare informazioni su particolari aree di interesse.

Nell'articolo vengono presentati i risultati dei test eseguiti su un campione di utenti. Il campione di popolazione valutato includeva nove esperti in diverse discipline accademiche che vanno dall'interazione uomo-macchina (in inglese *human-computer interaction*, HCI) alla comunicazione visiva. Considerando gli esperti di HCI e di comunicazione visiva in questa valutazione è stata presa in considerazione la loro capacità di valutare l'usabilità, l'interattività e il livello di esperienza utente acquisita. Quindi, gli esperti di studi museali sono

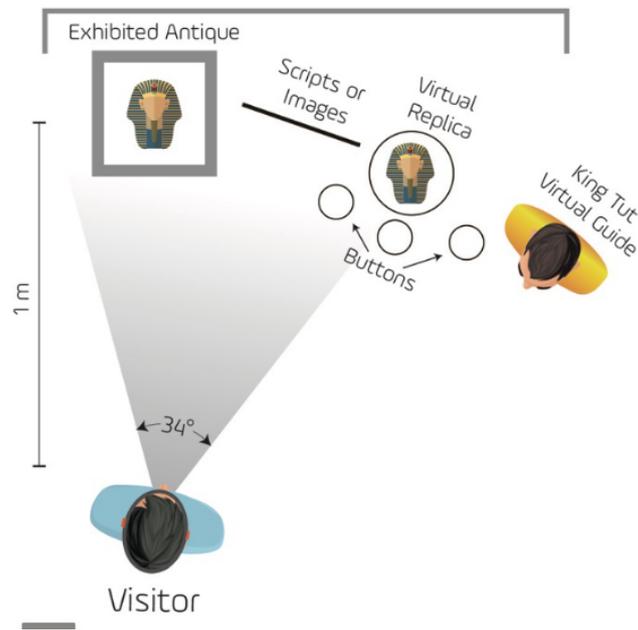


Figura 3.3: Scena di esempio dell'applicazione.

stati considerati per valutare se questo sistema può raggiungere ciò che i visitatori richiedono nei musei e per strutturarlo in base alla natura delle visite. Considerare gli esperti per questa valutazione ha un doppio vantaggio: come menzionato hanno esperienza e possono anche essere generalmente visitatori di musei, quindi le loro risposte possono essere più critiche e vantaggiose per lo studio rispetto a quelle dei normali visitatori del museo.

Il prototipo dell'applicazione ha ottenuto risultati complessivamente positivi sia in termini di usabilità che di accessibilità.

Capitolo 4

Caso di Studio

4.1 Descrizione del Caso di Studio

Come detto nel capitolo 3, l'applicazione della realtà mista in determinati settori può incrementare notevolmente efficienza, produttività e sicurezza. Grazie alla realtà mista, è possibile creare esperienze in cui gli utenti possono interagire fra loro e con l'ambiente in nuovi modi. Queste nuove interazioni possono essere introdotte digitalizzando le entità reali e le informazioni presenti nell'ambiente. Avere una rappresentazione digitale di un dispositivo può permettere di osservarne lo stato e di controllarlo senza dover interagire direttamente con quel dispositivo. Per quanto riguarda le informazioni, è possibile avere rappresentazioni che evidenziano determinati aspetti, utili al tipo di utente che sta eccedendo a quell'informazione. In un ambiente MR più utenti possono anche interagire contemporaneamente con la stessa informazione sfruttando rappresentazioni differenti. È anche possibile condividere la stessa rappresentazione di un dispositivo o di un'informazione con più utenti contemporaneamente, in modo da permettere esperienze condivise in tempo reale. Le rappresentazioni digitali possono essere generate - grazie a determinate tecniche di riconoscimento - semplicemente osservando le informazioni o gli oggetti presenti nell'ambiente.

Dal punto di vista della condivisione, le informazioni e le loro rappresentazioni possono essere condivise fra più utenti secondo le seguenti modalità:

- Vengono condivise rappresentazioni differenti della stessa informazione. In questo modo diversi utenti possono accedere alle stesse informazioni contemporaneamente, ma in luoghi e con modalità differenti.
- Viene condivisa la stessa rappresentazione dell'informazione fra più utenti, quindi in allineamento sia spaziale che temporale, così da permettere esperienze condivise e collaborazioni in uno stesso ambiente.

Considerando entrambi i punti, un singolo utente può visualizzare contemporaneamente sia rappresentazioni condivise con altri utenti che locali.

Le rappresentazioni presenti nell'ambiente devono comportarsi come se fossero oggetti reali, quindi devono essere considerate come entità attive. Per quanto riguarda le rappresentazioni relative ai dispositivi presenti nell'ambiente, un'azione su una rappresentazione produce un cambiamento di stato sul dispositivo associato. Allo stesso modo, se avviene un cambiamento di stato su un dispositivo, le sue rappresentazioni devono essere aggiornate.

Un'applicazione MR con le caratteristiche appena descritte può essere introdotta in vari ambiti, fra cui quelli presentati nel capitolo 3. Tuttavia, in questo caso di studio si vuole considerare in particolar modo il settore sanitario, nel quale la realtà mista può essere utilizzata per migliorare le collaborazioni fra medici e infermieri o per controllare in modo più efficiente l'ambiente in determinate situazioni, per esempio nel momento in cui si effettuano operazioni.

Uno dei possibili casi di utilizzo può essere l'analisi di determinati documenti o referti, contenenti informazioni che possono essere visualizzate secondo punti di vista differenti, anche in base al tipo di utente che le sta osservando.

Un esempio può essere l'analisi dei risultati di una TAC presente su un computer: il medico deve poter "prendere" le informazioni e portarle nell'ambiente, rendendole indipendenti dal dispositivo in cui risiedono. Nell'esempio della TAC può essere creato un modello 3D che permetta un'analisi più dettagliata rispetto a una serie di immagini. Inoltre, medici e infermieri devono avere la possibilità di visualizzare solo i dati della TAC interessati.

L'obiettivo del caso di studio è proprio quello di realizzare un'applicazione MR per HoloLens 2 che possa essere impiegata in ambito sanitario.

Un operatore deve essere in grado di aggiungere facilmente nuove informazioni e di poterle condividere con altri operatori, quindi sorge la necessità di rendere l'informazione indipendente dal dispositivo che si utilizza. Inoltre ogni informazione deve essere identificata in modo univoco, in modo da evitare fraintendimenti. Nell'ambiente di realtà mista possono quindi essere presenti contemporaneamente informazioni private (relative a un solo dispositivo) e condivise, in tutto o in parte.

Oltre alle informazioni devono essere identificati correttamente gli oggetti e i dispositivi dei quali si vuole gestire una rappresentazione.

Inoltre, bisogna stabilire delle politiche di accesso in lettura e scrittura sia per i dispositivi che per le informazioni. Ogni operatore ha quindi determinati diritti di accesso sulle rappresentazioni che può utilizzare.

4.2 Analisi dei Requisiti

Uno degli aspetti chiave nella realizzazione del sistema è la necessità di dover "informare" HoloLens su ciò che si sta guardando, in modo così che possa proiettare e condividere le rappresentazioni adeguate. HoloLens deve poter rilevare informazioni (come risonanze, TAC e radiografie) e gli oggetti di cui si vuole gestire una rappresentazione. Una delle soluzioni più adatte per realizzare questa funzionalità è l'utilizzo di marker. A ogni informazione e dispositivo deve essere associato un marker che possa essere rilevato correttamente da HoloLens. Nello specifico è possibile realizzare un marker ad hoc per ogni informazione o utilizzare l'informazione vera e propria come marker.

Per virtualizzare determinati tipi di informazioni non è possibile sfruttare immagini 2D, quindi sorge la necessità di rilevare oggetti 3D, per esempio un monitor multi-parametrico relativo a un paziente. Nel caso specifico, una volta riconosciuto il monitor, HoloLens deve proiettare nell'ambiente una copia delle informazioni che questo presenta.

Un altro aspetto chiave riguarda la condivisione delle informazioni, che può avvenire in modalità differenti. Per quanto riguarda l'accesso in allineamento esclusivamente temporale è necessario astrarre l'informazione dalle sue copie virtuali presenti nell'ambiente. Invece per le esperienze condivise in cui più operatori devono collaborare e interagire con gli stessi ologrammi bisogna condividere non solo le informazioni, ma anche posizione e rotazione degli ologrammi, in modo da garantire un allineamento sia temporale che spaziale. Per realizzare questa funzionalità si può sfruttare un servizio che permette di autenticare gli operatori e di condividere le informazioni nelle modalità descritte.

Il servizio in questione deve anche permettere l'interazione degli ologrammi con il mondo reale, in particolare con i dispositivi IoT presenti nell'ambiente.

L'utilizzo di un servizio permette anche l'accesso da dispositivi e piattaforme differenti, una caratteristica utile per sviluppi futuri.

4.2.1 Casi d'Uso

In questa sottosezione vengono presentati dei casi d'uso che descrivono alcune delle azioni che gli operatori possono effettuare sfruttando l'applicazione MR.

Nel diagramma presente in Figura 4.1 vengono messe in evidenza le modalità di accesso alle informazioni e di richiesta delle rappresentazioni. Un operatore, per poter accedere a qualsiasi tipo di rappresentazione, deve prima autenticarsi. Questa procedura è necessaria per poter comprendere a quali rappresentazioni l'operatore ha il diritto di accedere e con che modalità.

Una volta effettuato l'accesso, l'operatore può accedere alle rappresentazioni sfruttando i marker. In particolare, nel momento in cui viene rilevato un marker, l'operatore deve essere in grado di decidere - tramite l'utilizzo di apposite interfacce grafiche - se richiedere la rappresentazione associata. In altre parole, la rappresentazione non deve essere presentata direttamente nel momento in cui viene rilevato il marker.

Un'altra caratteristica fondamentale presentata in Figura 4.1 è la gestione dei permessi. Per ogni azione che viene eseguita dall'operatore su una rappresentazione devono essere controllati i diritti di accesso.

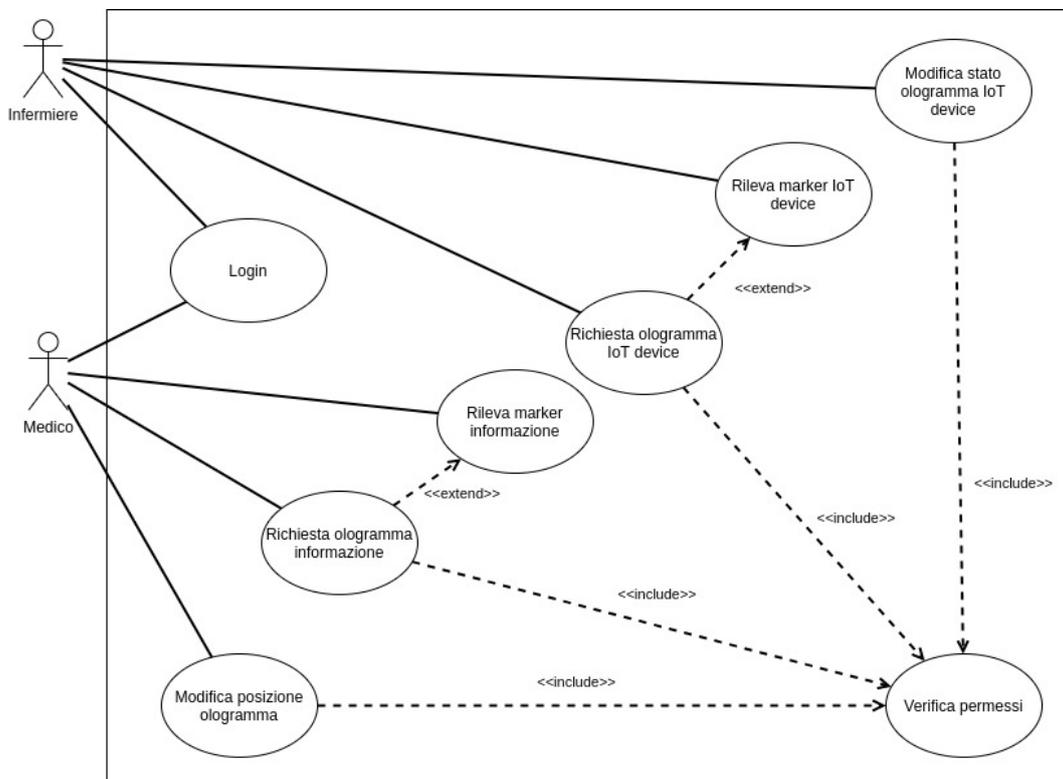


Figura 4.1: Diagramma dei casi d'uso che descrive il modo in cui informazioni e dispositivi vengono digitalizzati grazie ai marker.

Nel diagramma presente in Figura 4.2 viene messo in evidenza il fatto che determinate azioni eseguite sulle rappresentazioni apportano modifiche sia allo stato dell'ambiente che alle altre rappresentazioni.

In questo caso abbiamo due medici che partecipano a un'esperienza condivisa, quindi condividono le stesse rappresentazioni. Se uno dei due medici sposta, ruota o ridimensiona un ologramma, lo stesso effetto deve essere applicato all'ologramma che l'altro medico sta visualizzando.

La stessa cosa vale per i dispositivi IoT presenti nell'ambiente. Un'azione eseguita su una rappresentazione relativa a un dispositivo deve produrre un effetto sia sul dispositivo stesso che sulle sue rappresentazioni.

In Figura 4.2, per alleggerire il carico grafico, vengono omessi sia il login che la verifica dei permessi per ogni azione eseguita; operazioni che devono comunque essere sempre eseguite.

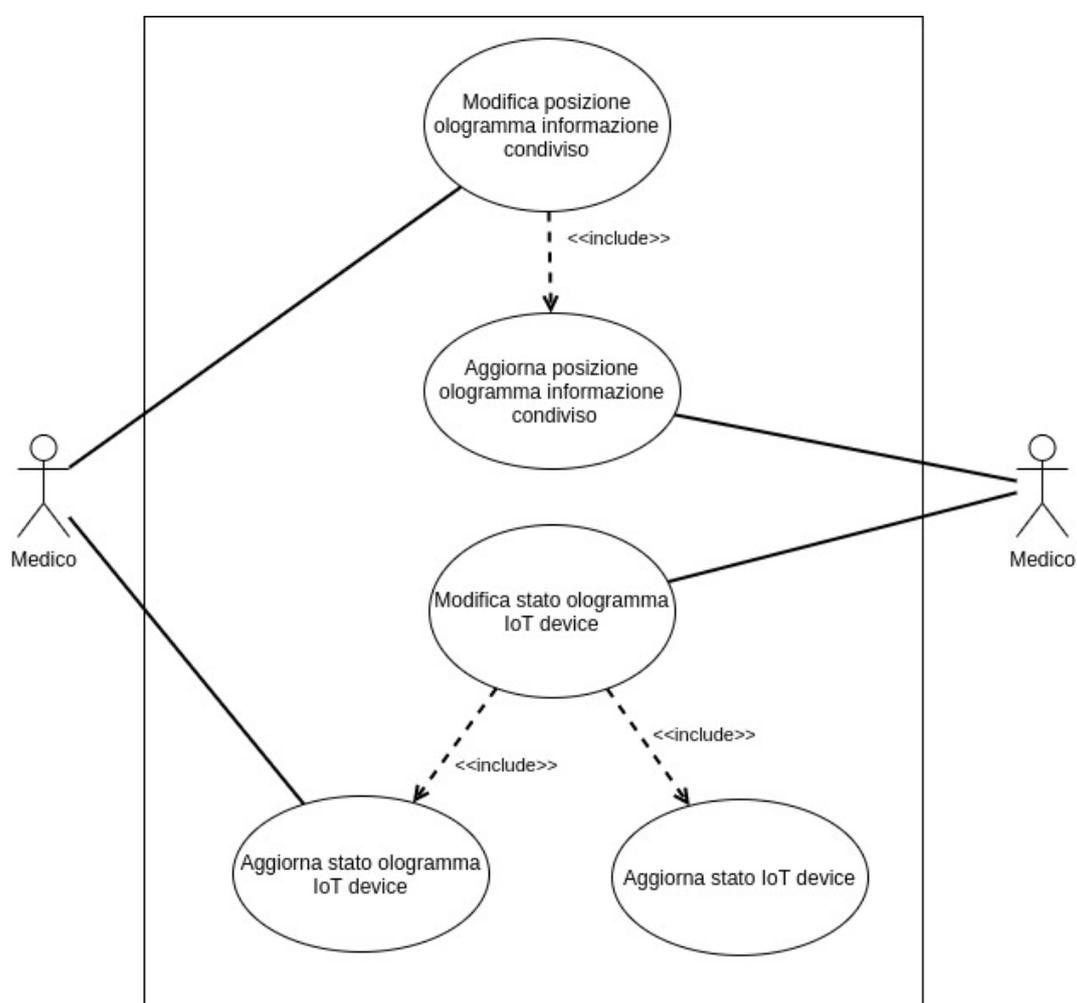


Figura 4.2: Diagramma dei casi d'uso che descrive gli effetti delle interazioni con le rappresentazioni.

4.2.2 Identificazione delle Informazioni e dei Dispositivi IoT nell'Ambiente

Come detto all'inizio di questa sezione, una delle soluzioni per identificare correttamente le informazioni presenti nell'ambiente è utilizzare i marker.

Oltre ai marker deve essere realizzato un punto di condivisione - come un servizio - che si occupi di associare correttamente le rappresentazioni ai marker. Il servizio in questione deve autenticare gli operatori e deve rendere disponibili le rappresentazioni delle informazioni e dei dispositivi IoT.

Grazie all'autenticazione dell'operatore, il servizio è in grado di gestire correttamente sia le rappresentazioni a cui è possibile accedere che i permessi relativi alle azioni che si possono eseguire sulle informazioni e sui dispositivi IoT.

È importante sottolineare che le informazioni possono avere diverse rappresentazioni, non tutte accessibili da tutti gli operatori, questo perché ogni operatore ha determinate necessità e preferisce avere rappresentazioni che evidenzino determinati aspetti dell'informazione.

In questo modo, in un ambiente MR, più utenti possono interagire con le stesse informazioni tramite rappresentazioni differenti. Ogni cambiamento di stato di un'informazione o di un dispositivo viene replicato su tutte le sue rappresentazioni grazie al servizio.

4.2.3 Allineamento dei Sistemi di Coordinate

Nelle esperienze condivise, in cui gli ologrammi devono essere presentati in allineamento spaziale e temporale, bisogna permettere la condivisione dei sistemi di coordinate fra più dispositivi.

Nello stesso ambiente, dispositivi MR possono avere sistemi di coordinate differenti. Per questo motivo, per realizzare esperienze condivise, non è possibile condividere direttamente le coordinate di un ologramma.

Bisogna stabilire dei punti di riferimento comuni all'interno dell'ambiente e condividere le coordinate degli ologrammi rispetto a quei punti.

Esprimendo le coordinate dell'ologramma rispetto a un determinato punto nell'ambiente permette di renderle indipendenti dal dispositivo che si utilizza.

Per la condivisione delle coordinate è possibile sfruttare lo stesso servizio descritto nella sottosezione precedente.

I punti di riferimento all'interno dell'ambiente possono essere stabiliti sempre sfruttando i marker, gli stessi che vengono utilizzati per riconoscere le informazioni e i dispositivi nell'ambiente.

Quindi, in un'esperienza condivisa con dispositivi differenti, nel momento in cui un operatore interagisce con un ologramma:

1. Viene applicata una trasformazione alle coordinate di quell'ologramma, che vengono espresse secondo la posizione del marker associato.
2. La posizione viene mandata al servizio.
3. Il servizio replica le coordinate verso tutti i dispositivi che partecipano all'esperienza condivisa.
4. In fase di ricezione ogni dispositivo converte le coordinate espresse rispetto al marker nel suo sistema di coordinate.

4.2.4 Interazioni fra Entità Virtuali e Reali

Per quanto riguarda la realizzazione delle interazioni fra entità reali e virtuali bisogna rendere "intelligenti" le prime, introducendo quindi dispositivi IoT nell'ambiente. Questa caratteristica può risultare utile per aumentare la capacità di controllo dell'ambiente da parte degli operatori. Per esempio, facendo riferimento al sistema di illuminazione dell'ambiente, si potrebbe permettere di spegnere o accendere le luci in tutto o in parte da qualsiasi posizione, indipendentemente dalla posizione degli interruttori fisici. Per realizzare questa funzionalità naturalmente bisogna registrare caratteristiche e stato di ogni dispositivo IoT e renderli disponibili per i dispositivi MR. Bisogna anche associare correttamente il dispositivo IoT alla sua rappresentazione digitale, che deve riportarne lo stato e in determinati casi permettere di modificarlo. Tornando all'esempio del sistema di illuminazione, è possibile gestire il suo stato tramite un ologramma che rappresenta un pannello di controllo.

Per rendere i dispositivi intelligenti si possono utilizzare sistemi integrati basati su microcontrollori o su SoC, come Arduino Uno, ESP 8266 o Raspberry Pi 4.

Per la comunicazione e l'aggiornamento dello stato dei dispositivi bisogna connettere questi ultimi alla rete e sfruttare un servizio che permetta ad HoloLens di accedere alle rappresentazioni digitali dei dispositivi intelligenti.

4.3 Progettazione del Sistema

Data l'analisi dei requisiti, per la realizzazione del sistema viene proposta la seguente soluzione, composta da due macro-componenti, che comunicano attraverso la rete sfruttando il protocollo HTTPS.

- **Back-End Engine:** offre un insieme di servizi, grazie ai quali si occupa di autenticare gli utenti e di gestire le politiche di accesso alle informazioni e ai dispositivi intelligenti.

- **Holographic App:** è un'applicazione client progettata per HoloLens, che si occupa di comunicare con il Back-End Engine per accedere alle rappresentazioni delle informazioni e dei dispositivi IoT.

In Figura 4.3 vengono presentati i due macro-componenti e il modo in cui comunicano, abbiamo il Back-End Engine al centro, dei dispositivi IoT e la Holographic App che sta rappresentando contemporaneamente sia ologrammi a informazioni che a dispositivi IoT. Il Back-End Engine viene utilizzato come punto di accesso per tutti i dispositivi che fanno parte della realtà mista. Per ogni device IoT viene espresso uno stato, che viene rappresentato dagli ologrammi associati a quel dispositivo. I due macrosistemi interagiscono attraverso la rete nei seguenti modi:

- In un'esperienza condivisa, se un operatore modifica le coordinate di un ologramma, la Holographic App invia le nuove coordinate al Back-End Engine, che le inoltra a tutti gli altri operatori.
- Nel momento in cui un operatore modifica un'informazione tramite una sua rappresentazione il Back-End Engine si occupa di rendere effettive e persistenti le modifiche.
- Se un operatore modifica lo stato di un dispositivo IoT tramite una sua rappresentazione il Back-End Engine applica effettivamente lo stato al dispositivo in questione e aggiorna tutte le sue rappresentazioni.
- Nel momento in cui un dispositivo IoT cambia stato (non a causa di una interazione con una sua rappresentazione) il cambiamento viene comunicato al Back-End Engine, che si occupa anche in questo caso di aggiornare tutte le sue rappresentazioni.

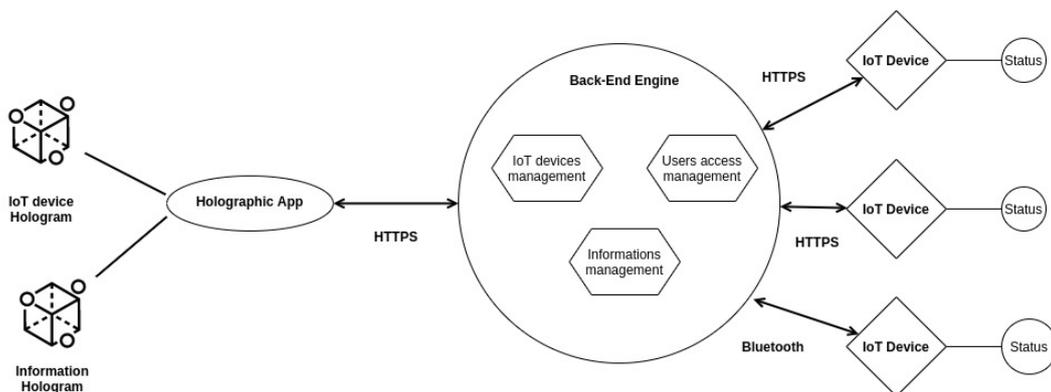


Figura 4.3: Diagramma che presenta il modo in cui i macro-componenti comunicano.

4.3.1 Back-End Engine

Il Back-End Engine (Figura 4.4) si occupa di gestire gli utenti, le informazioni e i dispositivi IoT presenti nell'ambiente.

Questo macrosistema inoltre memorizza in modo persistente le informazioni con i relativi marker e lo stato dei dispositivi IoT presenti nell'ambiente sfruttando i componenti `IoTDeviceManager` e `InformationManager`.

Un'informazione può avere rappresentazioni distinte che coesistono allo stesso tempo. Fornendo questo livello di astrazione, più utenti possono interagire con le stesse informazioni in modi differenti. Più utenti possono comunque interagire contemporaneamente con una stessa rappresentazione, condivisa in allineamento spaziale e temporale. Uno o più utenti possono accedere a uno stesso ologramma e modificarne posizione e rotazione, che verranno replicate su tutti i dispositivi MR che stanno accedendo a quell'ologramma in quel momento; in questo modo è possibile realizzare esperienze condivise.

Le politiche di accesso, gestite dal componente `UserPolicyManager`, vengono utilizzate per permettere solo a determinati utenti di accedere alle rappresentazioni delle informazioni. In particolare un utente, per una singola informazione, può avere accesso in lettura, in scrittura, entrambe o nessuna delle due.

I dispositivi IoT vengono suddivisi in attuatori (con possibilità di leggere e modificare lo stato) e sensori (che permettono unicamente di leggere lo stato). Per ogni dispositivo IoT viene fornita una singola rappresentazione e un indirizzo IP. L'accesso dai client MR verso i dispositivi IoT non è diretto, viene comunque utilizzato il servizio, perché devono essere sempre controllati i diritti di accesso. Quindi per richiedere o modificare lo stato di un determinato dispositivo è necessario contattare il servizio. Esattamente come per le informazioni, gli utenti hanno determinati diritti di accesso sui dispositivi IoT.

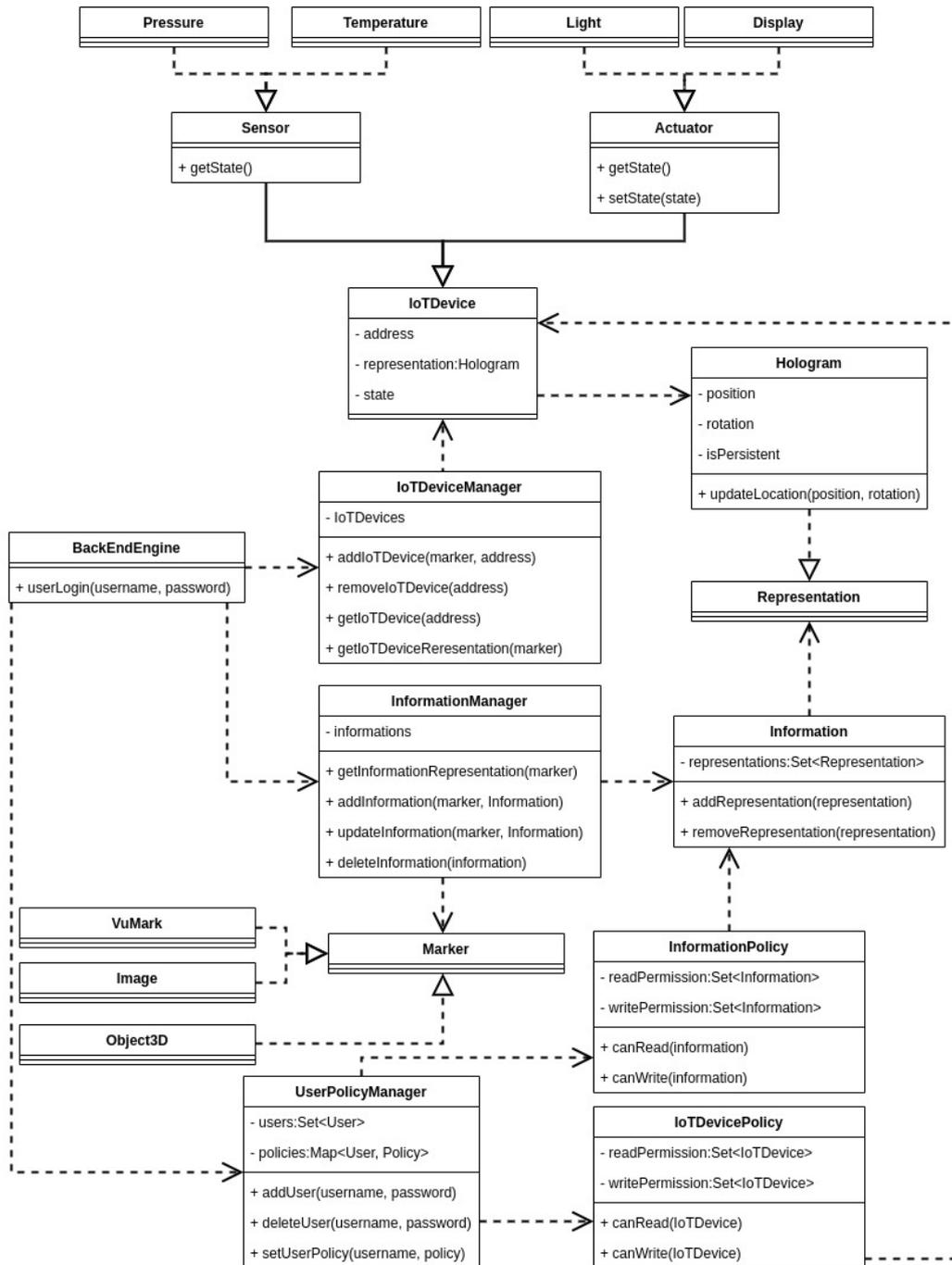


Figura 4.4: Back-End Engine - Diagramma delle classi.

4.3.2 Holographic App

La Holographic App (diagramma delle classi in 4.5) per prima cosa si occupa sempre di richiedere le credenziali di accesso all'utente e di autenticarsi comunicando con il Back-End Engine. Una volta eseguito l'accesso l'utente può richiedere e interagire con le rappresentazioni delle informazioni e dei dispositivi IoT presenti nell'ambiente.

Il componente `HolographicAppController` si occupa di gestire l'applicazione, comunicando con il Back-End Engine tramite `ServiceConnectionManager` e controllando gli ologrammi tramite `HologramManager`.

`ServiceConnectionManager` si occupa di comunicare con il Back-End Engine. In particolare, viene sfruttato per:

- Eseguire il login.
- Richiedere la rappresentazione di un'informazione nel momento in cui viene rilevato un marker.
- Richiedere la rappresentazione di un dispositivo IoT.
- Spedire e ricevere le nuove coordinate di un ologramma, in modo da garantire esperienze condivise in tempo reale.
- Modificare lo stato di un dispositivo IoT tramite la sua rappresentazione.

In un'esperienza condivisa, se un utente modifica la posizione di un ologramma, la Holographic App manda le nuove coordinate al Back-End Engine, che si occupa di memorizzarle e di replicarle verso tutti gli altri utenti che stano partecipando a quell'esperienza.

La gestione degli ologrammi viene effettuata dall'`HologramManager`, che si occupa di istanziare e gestire le rappresentazioni delle informazioni e dei dispositivi IoT fornite dal `ServiceConnectionManager`.

`HologramManager` si occupa anche di applicare affettivamente le nuove coordinate degli ologrammi fornite dal Back-End Engine durante un'esperienza condivisa.

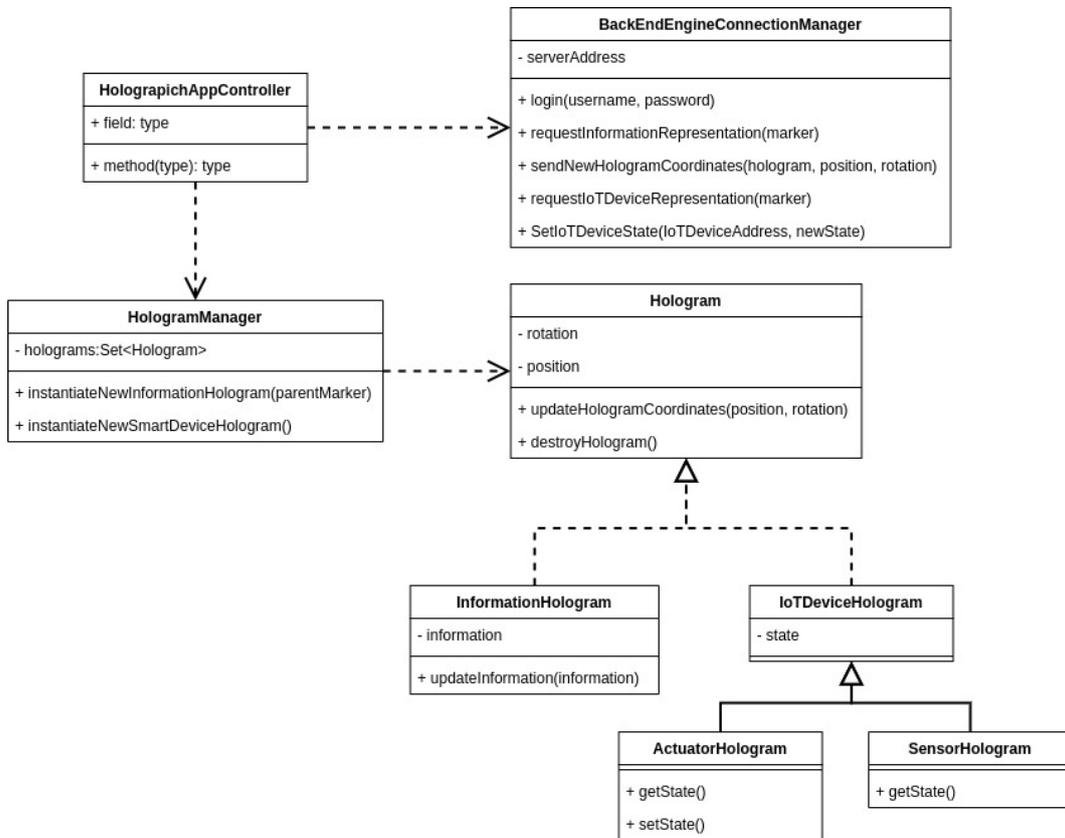


Figura 4.5: Holographic App - Diagramma delle classi.

In Figura 4.6 viene fornita una rappresentazione della sequenza di operazioni che avviene fra client e back-end per realizzare un'esperienza condivisa. Come detto in precedenza, per prima cosa viene sempre eseguito il login, successivamente viene rilevato un marker e di conseguenza viene richiesta una rappresentazione dell'informazione al Back-End Engine. Una volta ricevuta la rappresentazione l'ologramma viene istanziato effettivamente. Nel caso presentato in Figura 4.6 la stessa rappresentazione è condivisa con altri utenti, infatti il servizio manda le nuove coordinate dell'ologramma al client, dato che un altro client le ha modificate. Infine, nel momento in cui il client esce dall'esperienza condivisa elimina l'ologramma, ma solo localmente, così gli altri utenti possono continuare a interagire con quell'ologramma.

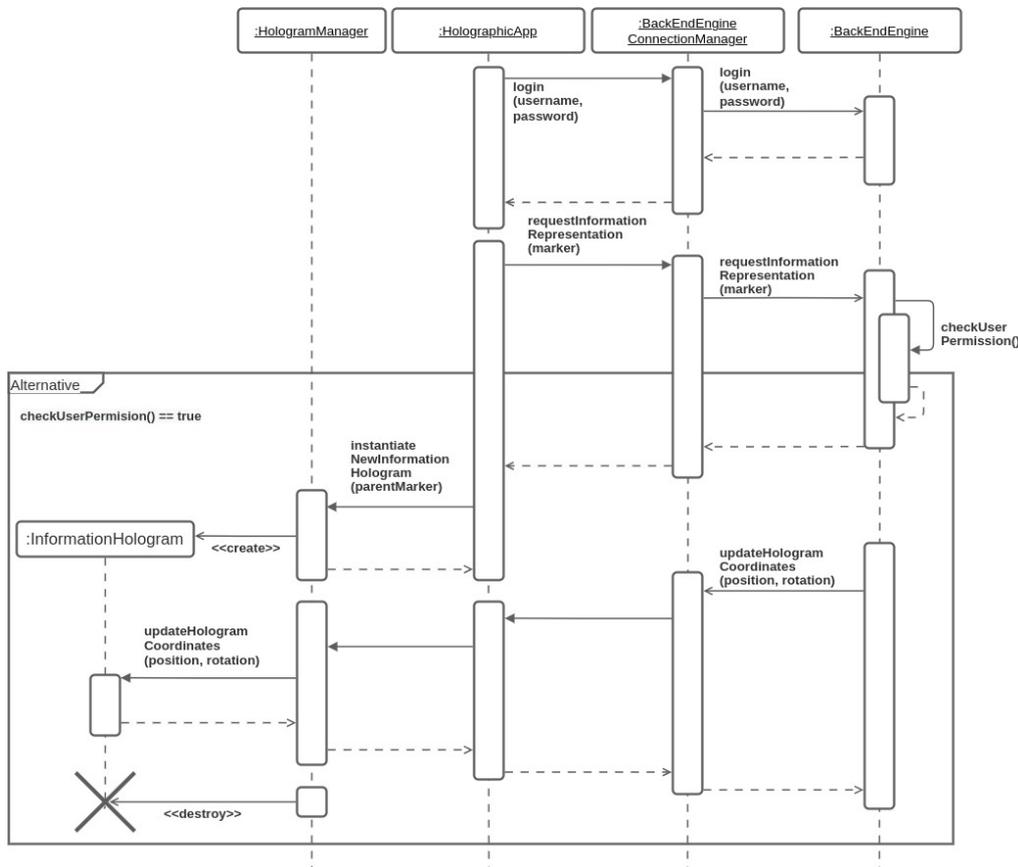


Figura 4.6: Holographic App - Diagramma di sequenza di un'esperienza condivisa.

Un altro caso interessante viene presentato in Figura 4.7. Qui abbiamo un utente che richiede la rappresentazione di un dispositivo IoT (luce) al Back-End Engine. In questo caso la parte in cui viene effettuato il login è stata omessa per alleggerire il carico grafico, ma viene comunque eseguita. Una volta controllati i permessi di accesso, il Back-End Engine restituisce la rappresentazione del dispositivo e l'ologramma relativo viene generato. A questo punto l'utente ha la possibilità non solo di accedere allo stato della luce, ma anche di modificarlo sfruttando la sua rappresentazione. L'utente interagisce con l'ologramma e accende la luce (che inizialmente era spenta); il **ServiceConnectionManager** invia la richiesta al Back-End Engine, che controlla ancora una volta i permessi e infine accende effettivamente la luce contattando il dispositivo IoT associato.

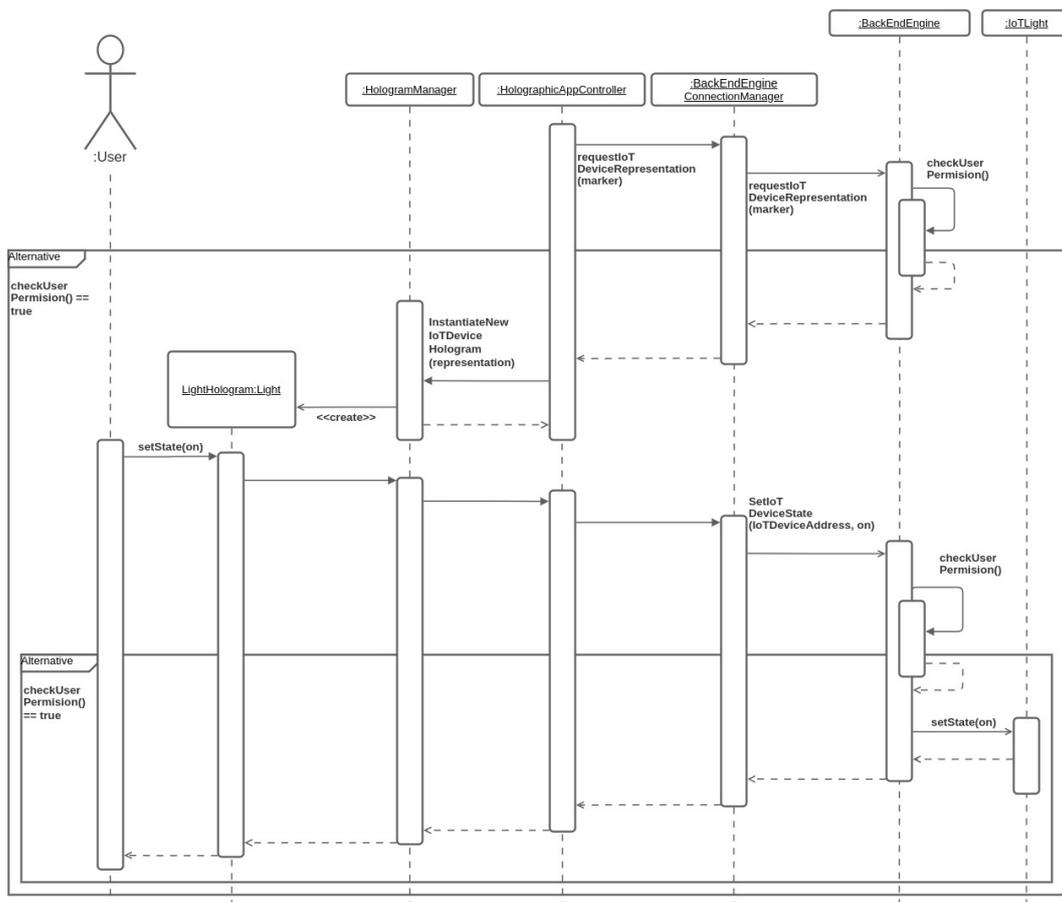


Figura 4.7: Holographic App - Diagramma di sequenza con un dispositivo IoT.

4.4 Sviluppo del Prototipo

4.4.1 Tecnologie Utilizzate

Per lo sviluppo del prototipo relativo al caso di studio¹ sono state utilizzate le seguenti tecnologie:

Unity 2020 LTS è stato utilizzato come ambiente di sviluppo principale. Si è scelto di utilizzare questa specifica versione perché al momento viene considerata come miglior soluzione secondo Microsoft.

Per lo sviluppo degli script C# presenti nell'applicazione è stato utilizzato Visual Studio 2019.

Il plugin per la gestione della realtà mista che si è scelto di integrare in Unity è OpenXR, perché nonostante sia ancora in fase sperimentale, in futuro

¹Repository GitHub: <https://github.com/FilippoVissani/three-year-degree-thesis>

rimpiazzerà tutti gli altri plugin esistenti. Inoltre, OpenXR fornisce delle API comuni per tutte le principali piattaforme, una caratteristica utile se in futuro si volessero integrare dispositivi diversi da HoloLens.

Insieme a OpenXR sono stati importati i componenti del Mixed Reality Toolkit Foundation 2.7, che permettono di accedere direttamente alle funzionalità offerte da HoloLens grazie a dei profili. Una volta aggiunto il componente a Unity, compare un menu dove è possibile aggiungere gli oggetti `MixedRealityToolkit` e `MixedRealityPlayspace` alla scena. Tramite l'oggetto `MixedRealityToolkit` è possibile gestire i profili relativi alle funzionalità di HoloLens.

Per permettere ad HoloLens di rilevare i marker è stato utilizzato Vuforia 10.0.1 integrato in Unity. Vuforia permette di riconoscere vari tipi di marker, basati su immagini o su modelli 3D. È possibile associare uno o più oggetti 3D o 2D al marker, nel momento in cui il marker viene rilevato gli oggetti associati vengono aggiunti alla scena. Un aspetto importante da considerare è che i marker Vuforia non possono trasmettere informazioni come ad esempio fanno i codici QR.

Per associare un oggetto a un marker è sufficiente impostarlo come figlio nel progetto Unity. Uno dei problemi sorti durante i test è che gli ologrammi relativi a un marker su HoloLens hanno problemi di stabilità, come sfarfallio o posizione non precisa. Nel momento in cui l'utente si muove nell'ambiente, gli ologrammi associati a un marker possono spostarsi, assumendo una posizione con un errore considerevole. Per questo motivo bisogna trovare una soluzione che permetta di creare gli ologrammi, ma senza associare direttamente gli oggetti 3D al marker.

4.4.2 Struttura del Progetto Unity

Nella scena principale del progetto (Figura 4.8) sono presenti tre componenti principali:

- **MixedRealityToolkit**: utilizzato per gestire i profili relativi alle funzionalità offerte da HoloLens.
- **HolographicAppController**: oggetto che si occupa di inizializzare la scena a runtime, generando i marker associati alle immagini.
- **HologramManager**: oggetto che si occupa della gestione degli ologrammi associati ai marker.

Gli oggetti relativi ai marker e agli ologrammi non sono presenti all'interno della scena di Unity, perché questi vengono istanziati a runtime.

Per quanto riguarda i profili del MRTK sono state abilitate le seguenti funzionalità: Camera, Input, Boundary, Spatial Awareness con occlusione della mesh.

I prefab che vengono istanziati a runtime sono composti da oggetti 3D, finestre e menu. I menu vengono utilizzati per attivare o disattivare determinate funzioni dell'ologramma. Una di queste funzioni è la capacità di seguire l'utente nell'ambiente. Gli oggetti 3D hanno integrati i seguenti componenti: Box Collider (utilizzato per rilevare correttamente qualsiasi tipo di collisione), Rigid Body (utilizzato per non far oltrepassare la mesh all'ologramma), Object Manipulator (utilizzato per abilitare le interazioni sfruttando il gaze), Near Interaction Grabbable (utilizzato per abilitare le interazioni da vicino), Follow Me Toggle (utilizzato per permettere all'utente di essere seguito dall'ologramma nel momento in cui si muove).

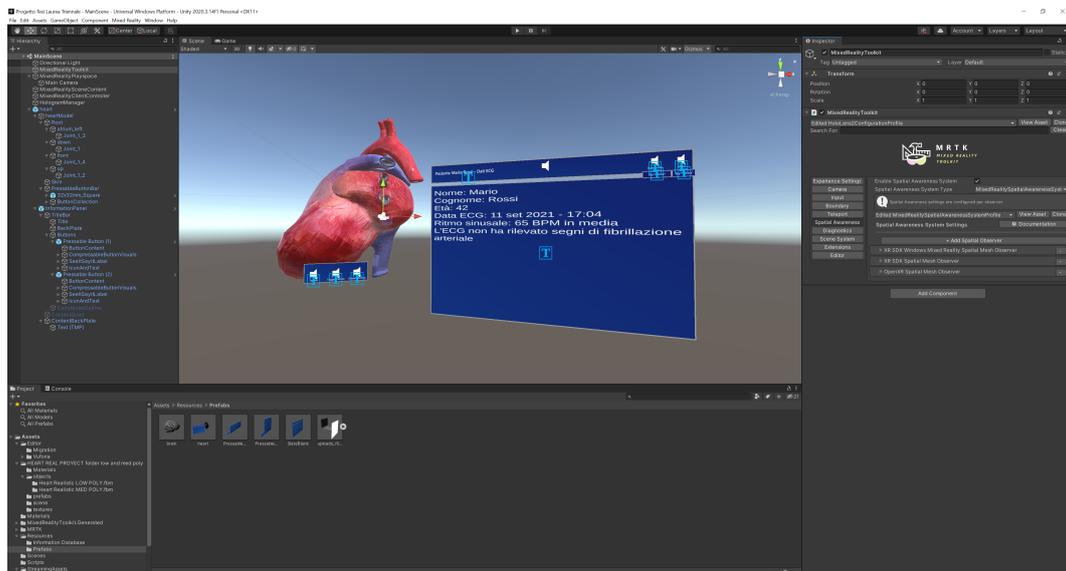


Figura 4.8: Scena di Unity relativa al prototipo.

4.4.3 Identificazione delle Informazioni nell'Ambiente

Per identificare correttamente le informazioni sono stati utilizzati i marker Vuforia. Ad ogni informazione è stato associato un marker, in questo modo ogni informazione può essere identificata in modo univoco.

I marker non sono stati caricati staticamente all'interno del progetto, questi infatti vengono scaricati a runtime da un servizio web presente all'interno della rete locale.

Il servizio contiene le immagini dei marker e un file Json dove le immagini vengono associate all'ologramma da istanziare.

All'avvio, l'applicazione si collega al servizio e scarica il file Json e le immagini dei marker (Listato 4.1).

Per il download dei dati vengono utilizzate due coroutine², `GetInformationDatabase` viene richiamata per scaricare il file Json, che viene deserializzato e convertito in un oggetto di tipo `Dictionary<string, string>`. il nome del marker viene utilizzato come chiave e il nome del prefab come valore associato.

Una volta scaricato correttamente il file Json viene avviata la coroutine `DownloadMarkerImages` per scaricare le immagini dei marker. In questa coroutine vengono considerati solo i marker presenti nel file Json. Quindi se un'immagine non viene considerata nel file Json, questa non viene neanche scaricata dall'applicazione. Per associare correttamente l'immagine scaricata al marker viene utilizzato un oggetto di tipo `Dictionary<string, Texture2D>`.

```
1
2     IEnumerator GetInformationDatabase()
3     {
4         using (UnityWebRequest uwr =
5             UnityWebRequest.Get(serverAddress +
6                 "information-database.json"))
7         {
8             yield return uwr.SendWebRequest();
9             if (uwr.result == UnityWebRequest.Result.Success)
10            {
11                informationDatabase =
12                    JsonConvert.DeserializeObject<Dictionary<string,
13                        string>>(uwr.downloadHandler.text);
14                StartCoroutine(DownloadMarkerImages());
15            }
16        }
17    }
18
19    private IEnumerator DownloadMarkerImages()
20    {
21        markerImages = new Dictionary<string, Texture2D>();
22        foreach (var marker in informationDatabase.Keys)
23        {
24            using (UnityWebRequest uwr =
25                UnityWebRequestTexture.GetTexture(serverAddress +
26                    marker + ".jpg"))
27            {
```

²Una coroutine è come una funzione che ha la capacità di mettere in pausa l'esecuzione e restituire il controllo a Unity, ma poi continuare da dove era stata interrotta nel frame successivo.

```
22         yield return uwr.SendWebRequest();
23         if (uwr.result == UnityWebRequest.Result.Success)
24         {
25             var texture =
26                 DownloadHandlerTexture.GetContent(uwr);
27             markerImages.Add(marker, texture);
28         }
29     }
30     currentState = State.InformationReady;
31 }
```

Listato 4.1: coroutine utilizzate per scaricare il file Json e le immagini dei marker dal servizio.

A questo punto le immagini vengono sfruttate per creare effettivamente i GameObject relativi ai marker presenti nella scena di Unity (Listato 4.2). Questa funzionalità viene implementata sfruttando le API di Vuforia, che permettono di istanziare gli oggetti relativi ai marker a runtime.

```
1     private void CreateImageTargetFromTexture(string markerName,
2         Texture2D image)
3     {
4         var mTarget = VuforiaBehaviour.Instance.ObserverFactory
5             .CreateImageTarget(image, 0.1f, markerName);
6         mTarget.gameObject
7             .AddComponent<DefaultObserverEventHandler>();
8     }
```

Listato 4.2: Metodo richiamato per creare i target in base alle immagini scaricate precedentemente.

4.4.4 Gestione degli Ologrammi Relativi alle Informazioni

La gestione degli ologrammi dell'applicazione viene delegata allo script `HologramManager.cs`. Questo script si occupa di associare a ogni marker un pulsante, che genera l'ologramma associato una volta premuto (Listato 4.3). Ci sono vari motivi per cui, nel momento in cui si rileva un marker, si genera un pulsante e non direttamente l'ologramma associato. L'operatore potrebbe non avere bisogno dell'ologramma anche se sta guardando il marker. Prendiamo come esempio il caso in cui si sta leggendo un documento che contiene anche il marker. Se l'ologramma venisse generato all'istante creerebbe confusione, si

vuole fare in modo di crearlo solamente nel momento in cui risulta necessario all'operatore. Un altro caso potenzialmente critico si genera nel momento in cui si osservano contemporaneamente più marker. In questo caso, evitando di usare un pulsante per ogni marker, verrebbero generati contemporaneamente tutti gli ologrammi associati, i quali occuperebbero inutilmente uno spazio nell'ambiente, creando confusione all'operatore.

A ogni pulsante viene anche associato un `Listener`, che viene richiamato alla pressione.

```
1 public void addHologram(string marker, string hologram)
2 {
3     GameObject buttonInstance = Instantiate(buttonPrefab);
4     buttonInstance.transform.eulerAngles = new Vector3(90f, 0f,
5         0f);
6     buttonInstance.transform.parent =
7         GameObject.Find(marker).transform;
8     buttonInstance.GetComponent<Interactable>()
9         .OnClick.AddListener(() => SpawnHologram(marker, hologram));
10 }
```

Listato 4.3: Metodo richiamato per associare un bottone ad ogni marker.

Il metodo `SpawnHologram` (Listato 4.4) viene associato a ogni pulsante, alla pressione del quale viene istanziato il prefab associato.

```
1 private void SpawnHologram(string marker, string hologram)
2 {
3     GameObject hologramInstance =
4         Instantiate(Resources.Load<GameObject>("Prefabs/" +
5         hologram));
6     hologramInstance.transform.position = new
7         Vector3(GameObject.Find(marker).transform.position.x,
8         GameObject.Find(marker).transform.position.y + 0.2f,
9         GameObject.Find(marker).transform.position.z);
10 }
```

Listato 4.4: Listener richiamato per generare l'ologramma associato al marker.

4.4.5 Allineamento dei Sistemi di Coordinate

L'allineamento dei sistemi di coordinate fra più dispositivi è necessario nel momento in cui si partecipa a un'esperienza condivisa. Per realizzare questa funzionalità si può procedere sfruttando:

- Azure Spatial Anchors
- Sistema di condivisione delle posizioni degli ologrammi realizzato su misura.

Azure Spatial Anchors permette di condividere in allineamento spaziale e temporale le ancore presenti in un ambiente. Tuttavia, non è possibile interagire direttamente con un ologramma ancorato, per permettere l'interazione bisogna prima eliminare l'ancora. Questo significa che più utenti non possono interagire contemporaneamente con lo stesso ologramma, perché non c'è modi di condividere la fase di interazione. Per questo motivo è opportuno realizzare un sistema più adeguato al caso di studio, che permetta di condividere anche le interazioni degli utenti con gli ologrammi.

Nelle applicazioni HoloLens realizzate con Unity l'origine del mondo viene impostata in corrispondenza del punto in cui si è avviata l'applicazione. Questo è un problema per la realizzazione delle esperienze condivise, perché diversi dispositivi hanno sistemi di coordinate con origine in punti differenti, quindi non è possibile condividere direttamente la posizione degli ologrammi. Un modo per realizzare questa funzionalità è sfruttare il sistema di coordinate esposto dai marker Vuforia e il Back-End Engine, che permette la condivisione delle coordinate degli ologrammi fra più client.

Nel momento in cui un client interagisce con un ologramma la nuova posizione viene inviata al servizio (Listato 4.5). Il servizio, nel momento in cui riceve una nuova posizione, la replica su tutti gli altri client sfruttando le WebSocket (Listato 4.6).

```

1  private async void SendPosition()
2  {
3      Vector3 position = childGameObject.transform.position -
4          marker.transform.position;
5      string message = "{\"X\": " + position.x
6          + ", \"Y\": " + position.y
7          + ", \"Z\": " + position.z + "}";
8      await ws.SendText(message);
9      lastPosition = childGameObject.transform.position;
10 }

```

Listato 4.5: Il client manda la nuova posizione dell'ologramma al servizio.

```

1  wss.on('connection', function connection(ws, req) {
2      ws.on('message', function incoming(data) {
3          wss.clients.forEach(function each(client) {
4              if (client !== ws && client.readyState === WebSocket.OPEN) {

```

```
5         client.send(data);  
6     }  
7     });  
8     });  
9     });
```

Listato 4.6: Il servizio replica la nuova posizione ricevuta su tutti gli altri client.

4.4.6 Persistenza degli Ologrammi nel Tempo e nello Spazio

In determinati casi può essere necessario ancorare gli ologrammi all'ambiente in modo persistente.

Prendiamo ad esempio il caso in cui un ambiente sia stato arredato con entità virtuali, magari anche in condivisione con più utenti. Può essere utile memorizzare posizione e rotazione degli ologrammi in modo persistente, perché altrimenti questi ultimi andrebbero riposizionati manualmente a ogni avvio dell'applicazione.

Per far persistere un ologramma localmente in un dispositivo è possibile utilizzare i sistemi di ancore offerti dal MRTK. Per le diverse versioni di Unity presenti oggi sono disponibili diversi sistemi di gestione delle ancore. Le API messe a disposizione permettono di salvare in modo persistente la posizione di un ologramma in un determinato ambiente. Attraverso diverse sessioni HoloLens riconosce l'ambiente e mette a disposizione le ancore salvate nelle sessioni precedenti. Tuttavia, questa funzionalità ha delle forti limitazioni, perché è possibile memorizzare le ancore solo localmente. Una possibile soluzione sarebbe utilizzare i servizi Azure, che permettono di condividere le ancore in cloud. L'unico problema è che al momento i servizi di condivisione delle ancore offerti da Azure sono compatibili solo con HoloLens e altri prodotti Microsoft. Quindi, per permettere di integrare dispositivi differenti, si è scelto ancora una volta di utilizzare il Back-End Engine.

Sfruttando il Back-End Engine è possibile far persistere un ologramma nello spazio e nel tempo, in altre parole è come realizzare un sistema di ancore simile al servizio offerto da Azure. L'utilizzo del servizio consente all'ologramma di "esistere" indipendentemente dal fatto che nell'ambiente ci siano utenti che interagiscono con esso. All'avvio dell'applicazione viene contattato il servizio e vengono scaricati gli ologrammi con le relative posizioni. Ogni volta che posizione o rotazione di un ologramma vengono modificate il servizio viene informato.

4.4.7 Prototipo in Esecuzione

All'interno del prototipo sono stati inseriti prefab relativi a varie informazioni, una delle quali è il risultato di un ECG di un paziente (Figura 4.9). Questo documento presenta i dati del paziente in questione, il risultato dell'ECG e il marker associato al documento.

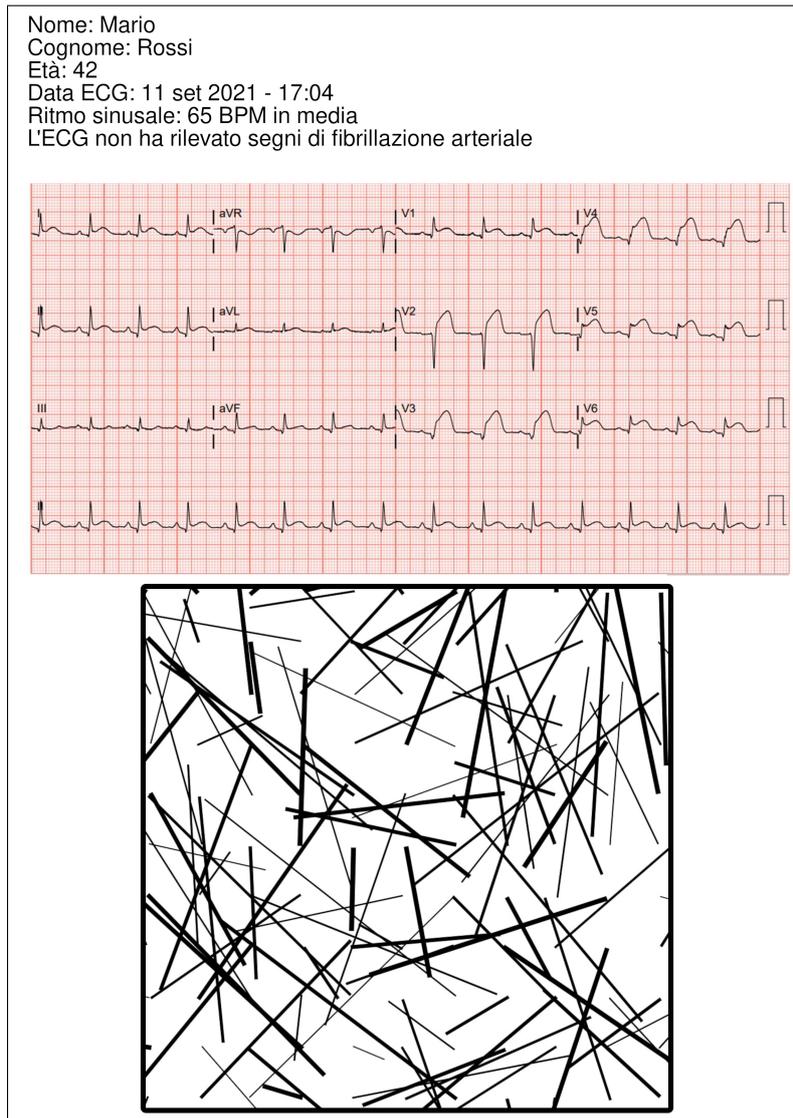


Figura 4.9: Esempio di documento contenente informazioni e marker.

In fase di esecuzione, il prototipo si collega al server presente in LAN e scarica il file Json che si occupa di associare il nome del prefab al marker. Successivamente vengono scaricate anche le immagini relative marker che vengono

riportati nel file Json. In questo caso il file Json contiene una entry del genere: "marker1": "heart".

A questo punto l'oggetto relativo al marker viene istanziato, così quando lo si osserva può essere rilevato correttamente. Nel momento in cui il marker viene riconosciuto dal prototipo, viene creato un pulsante (Figura 4.10) che permette di istanziare il prefab associato.

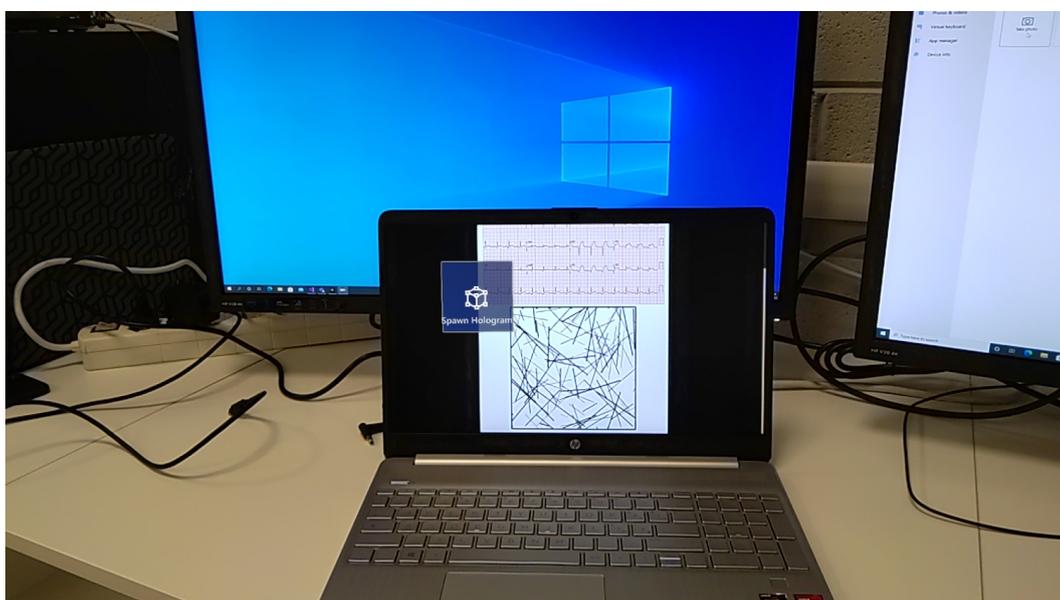


Figura 4.10: Nel momento in cui viene inquadrato il marker viene generato un pulsante che permette di creare l'ologramma.

Una volta premuto il pulsante relativo al marker presente nel documento in Figura 4.9 vengono istanziati due ologrammi: il modello 3D del cuore del paziente e una finestra che riporta le stesse informazioni presenti nel documento (Figura 4.11). A questo punto l'operatore è libero di posizionare gli ologrammi all'interno dell'ambiente come preferisce.

Sia la finestra che il cuore dispongono di un menu che permette di eliminarli e di attivare la funzionalità Follow Me, la quale permette all'utente di essere seguito dall'ologramma nel momento in cui si muove.



Figura 4.11: Ologrammi relativi al documento in Figura 4.9.

4.5 Considerazioni Finali

Con questo caso di studio si è dimostrata l'utilità della realtà mista in ambito sanitario, ma il sistema proposto può essere riutilizzato in tutto o in parte anche in altri ambiti. Il sistema può inoltre essere esteso, ampliando i casi in cui può essere utilizzato. Alcune delle possibili integrazioni che possono essere fatte vengono presentate nella sottosezione successiva.

Per quanto riguarda lo sviluppo del prototipo con MRTK e Unity è bene fare una precisazione: alcune delle funzionalità del MRTK sono ancora in fase sperimentale e Unity è stato progettato per sviluppare videogiochi, non applicazioni Mixed Reality. A causa di questi aspetti, creare e configurare correttamente un progetto per HoloLens 2 è un'operazione estremamente complessa e che richiede molto tempo. Avere un ambiente di sviluppo progettato appositamente per creare applicazioni Mixed Reality risolverebbe questo problema.

4.5.1 Sviluppi Futuri

Il macrosistema presentato nella sezione 4.3 è stato concepito per essere esteso, in modo da poter aggiungere ulteriori servizi e integrare nuovi dispositivi. Applicando determinate modifiche può anche essere riutilizzato in altri ambiti, oltre a quello sanitario. Sviluppando applicazioni specifiche per ogni

piattaforma, è possibile coinvolgere nelle esperienze di realtà mista dispositivi differenti, come smartphone o smart-glasses.

Invece per quanto riguarda le rappresentazioni, queste potrebbero essere generate dinamicamente. Prendiamo come esempio un display che riporta grafici dinamici, come l'andamento del battito cardiaco di un paziente. Può essere utile avere una rappresentazione sotto forma di ologramma di ciò che il display sta mostrando. Sfruttando al massimo la potenza della realtà mista, determinati strumenti potrebbero anche essere sostituiti completamente dalle loro rappresentazioni.

Si potrebbe anche gestire una rappresentazione degli operatori, in modo da permettere esperienze condivise sfruttando la telepresenza. In questo tipo di esperienza gli operatori possono collaborare pur non trovandosi fisicamente nello stesso ambiente. La telepresenza consentirebbe all'operatore di sentirsi come se fosse presente in un luogo diverso dalla sua vera posizione. Gli operatori presenti nell'ambiente avrebbero la possibilità di interagire con quelli in telepresenza grazie alle rappresentazioni digitali di questi ultimi.

Conclusioni

Traendo le conclusioni, in questa tesi sono stati presentati i risultati degli studi effettuati a proposito dello sviluppo di applicazioni Mixed Reality per HoloLens 2.

Il campo della realtà mista è molto ampio, può essere descritto secondo svariati punti di vista e avere infinite applicazioni. La definizione stessa di realtà mista si perfeziona con l'introduzione di nuove tecnologie che realizzano quest'ultima.

Nei quattro capitoli di questa tesi viene più volte evidenziato il fatto che HoloLens 2 è un dispositivo unico nel suo genere, che introduce importanti novità tecniche. Nessun altro dispositivo in commercio è in grado di rilevare l'ambiente e di proiettare ologrammi permettendo all'utente di interagirci come se fossero reali, nel modo in cui lo fa HoloLens 2. La realtà mista, della quale fino a qualche anno fa si è solo parlato da un punto di vista teorico, ora può essere concretizzata grazie ad HoloLens 2.

Tuttavia, questa tecnologia ha ancora dei limiti, come gli strumenti di sviluppo, che sono in parte ancora in fase sperimentale. Come detto nella Sezione 4.5, un ambiente di sviluppo creato su misura risolverebbe i problemi che si sono presentati durante le fasi di studio e di sviluppo. Probabilmente nei prossimi anni verrà fornito un supporto migliore per lo sviluppo di queste applicazioni.

Bisogna anche considerare che al momento gli utenti che hanno la possibilità di utilizzare HoloLens non sono molti perché inizialmente è stato commercializzato solo negli Stati Uniti, per poi essere reso disponibile in altri paesi solo nel corso del 2020. Probabilmente in futuro dispositivi come HoloLens verranno commercializzati su larga scala, raggiungendo un gran numero di utenti.

Alcune delle esperienze che la realtà mista permette in teoria non possono ancora essere realizzate concretamente, la telepresenza è un esempio. Con il progresso tecnologico, le potenzialità dei dispositivi Mixed Reality verranno incrementate notevolmente, questo permetterà di realizzare tutte le esperienze che finora sono state presentate solo a livello teorico. In futuro sarà possibile realizzare sistemi che ridurranno sempre di più il divario fra mondo reale e virtuale, fino a farlo scomparire del tutto.

Ringraziamenti

Prima di tutto desidero ringraziare il professor Angelo Croatti e il professor Alessandro Ricci, che mi hanno seguito costantemente durante il tirocinio e nella fase di stesura della tesi.

Un'altra persona che voglio ringraziare è Anna Vitali, con la quale ho sviluppato applicazioni Mixed Reality per comprendere il funzionamento di HoloLens 2 e Unity.

Ringrazio i miei genitori che mi hanno sempre sostenuto, appoggiando ogni mia decisione, fin dalla scelta del mio percorso di studi.

Infine un ringraziamento speciale va sia agli amici di una vita, che a quelli che ho conosciuto in questi tre anni di università.

Bibliografia

- [1] Thomas Caudell and D.W. Mitzell. Augmented reality: an application of heads-up display technology to manual manufacturing processes. *System Sciences*, 2:659–669, 01 1992.
- [2] Ramy Hammady, Minhua Ma, and Carl Strathearn. User experience design for mixed reality user experience design for mixed reality: a case study of hololens in museum. *International Journal of Technology Marketing*, 13, 07 2019.
- [3] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. *Telem manipulator and Telepresence Technologies*, 2351, 01 1994.
- [4] Brian Park, Stephen Hunt, Gregory Nadolski, and Terence Gade. 3d augmented reality-assisted ct-guided interventions: System design and preclinical trial on an abdominal phantom using hololens 2, 05 2020.
- [5] Ivan E. Sutherland. The ultimate display. In *Proceedings of the Congress of the International Federation of Information Processing (IFIP)*, volume volume 2, pages 506–508, 1965.
- [6] Aida Vidal-Balea, Óscar Blanco-Novoa, Imanol Picallo, Mikel Celaya-Echarri, Paula Fraga-Lamas, Peio Lopez Iturri, Leyre Azpilicueta, Francisco Falcone, and Tiago Fernández-Caramés. Analysis, design and practical validation of an augmented reality teaching system based on microsoft hololens 2 and edge computing. volume 2, 11 2020.