

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA

Corso di Laurea in Ingegneria Informatica Magistrale

ARtour: un sistema di navigazione
in Augmented Reality con
Large Language Models per tour culturali

Relatore:
Chiar.mo Prof.
PASQUALE CASCARANO

Presentata da:
ALESSIO TROFFEI

Correlatore:
PROF. GUSTAVO MARFIA

Sessione III
Anno Accademico 2024-2025

*Per aspera,
ad astra...*

Indice

Abstract	7
Introduzione	9
1 Realtà Aumentata	11
1.1 Definizione	11
1.2 Storia	12
1.2.1 Dalle Origini ai Primi Esperimenti	12
1.2.2 La Nascita dei Sistemi di Visualizzazione	12
1.2.3 Consolidamento e Definizione	13
1.2.4 L’Era Digitale: Diffusione e Standardizzazione	13
1.3 Tipologie di AR	14
1.3.1 Realtà Aumentata a marcatori (Marker-Based AR)	14
1.3.2 Realtà aumentata senza marcatori (Markerless AR)	15
1.4 Framework per Sistemi AR	16
1.4.1 SDK disponibili	16
2 Strumenti e Tecnologie di Sviluppo	20
2.1 Unity	20
2.2 Architettura di Unity	21
2.2.1 GameObjects e Components	21
2.2.2 Scene e Hierarchy	22
2.2.3 Asset Workflow	23
2.3 Sviluppo Cross-Platform	25
2.3.1 Mono Framework	25
2.4 Large Language Model (LLM)	27
2.4.1 Architettura Transformer	27
2.5 Matterport	29
2.6 Digital Twin	30
2.6.1 Definizione	30
2.6.2 Nell’industria	31

2.6.3	Nell'ambito AR	32
3	Stato dell'arte	37
3.1	Soluzioni AR per la Navigazione e il Turismo	37
3.2	Soluzioni AR con Assistenti Virtuali e LLM	39
3.3	Innovazioni e Vantaggi della Soluzione Proposta	40
4	Implementazione e Sviluppo	41
4.1	Architettura dell'Applicazione	41
4.1.1	Introduzione	41
4.1.2	Workflow Applicativo	43
4.1.3	Script Principali	46
4.2	Dettagli Tecnici	52
4.2.1	Organizzazione dei Dati	52
4.2.2	Interpretazione Dati	54
4.2.3	Gestione Trasparenza	56
4.3	Assistente LLM e Accessibilità	58
4.3.1	Sistema di Chat con LLM	58
4.3.2	Text-to-Speech	60
4.4	Pathfinding e Navigazione	61
4.4.1	Unity AI Navigaiton	61
4.4.2	NPC	62
4.4.3	Rilevamento Punti di Interesse Irraggiungibili	63
5	Conclusioni e Sviluppi Futuri	66
	Appendice	70
	Ringraziamenti	70

Abstract

La realtà aumentata (AR) è una tecnologia che consente di integrare contenuti digitali nel mondo reale, offrendo un'esperienza arricchita e interattiva.

Questo lavoro presenta un'applicazione AR sviluppata con Unity per dispositivi mobili usando il framework AR Foundation, progettata per guidare gli utenti verso punti di interesse che si trovano in luoghi turistici. La soluzione proposta supera i limiti dei tradizionali sistemi basati su mappe 2D e geolocalizzazione, offrendo un'esperienza interattiva e soprattutto accurata in ambienti interni.

L'applicazione tenta di replicare il comportamento di una guida turistica attraverso indicatori come frecce direzionali, scie e marcatori, che si adattano ai movimenti dell'utente. Viene inoltre integrato un sistema markerless che consente di visualizzare informazioni contestuali sulla destinazione raggiunta inquadrandola con il proprio smartphone e, grazie all'integrazione di Google Gemini Flash 2.0, è possibile fare domande all'assistente in-app, ottenendo risposte in linguaggio naturale che vengono poi lette ad alta voce tramite un sistema di Text-to-Speech (TTS).

Introduzione

Prima di approfondire il tema sulla realtà aumentata, è importante sottolineare le differenze tra i vari tipi di tecnologie immersive spesso confuse tra loro. Secondo Meta for Work [25] possiamo distinguere:

- **Realtà virtuale (VR):** La realtà virtuale utilizza modellazione e simulazione per creare un ambiente digitale completamente immersivo isolando l'utente dal mondo reale. Un esempio di dispositivo pensato la VR è *Meta Quest 3* ormai accessibile e ampiamente diffuso sul mercato consumer, oppure *HTC Vive Pro 2* ottimo per attività come l'addestramento degli operai in ambienti pericolosi. I software tipici di questa categoria spaziano dal settore videoludico (*Half-Life: Alyx*, *Resident Evil 4 VR*, *Beat Saber*) al settore medico, con strumenti di chirurgia virtuale come *Osso VR*.
- **Realtà aumentata (AR):** La realtà aumentata sovrappone elementi digitali al mondo reale, arricchendo la percezione dell'ambiente attraverso una visione composita.
Ad oggi l'AR è diventata diffusa grazie all'integrazione in numerose applicazioni mobile che spaziano dal settore del gaming, come *Pokémon GO* di Niantic al settore arredamento tramite l'app *IKEA*, allo shopping grazie ad Amazon, fino ad arrivare alle mappe tramite *Google Maps Live View*. Esistono anche dispositivi creati appositamente per AR come *Microsoft HoloLens* (ormai obsoleto).
- **Realtà mista (MR):** La realtà mista combina elementi del mondo reale con quelli virtuali, permettendo un'interazione tra i due. Un esempio di applicazione è la progettazione architettonica, in cui gli utenti possono posizionare ologrammi di edifici che reagiscono agli ostacoli fisici e permettono di manipolarli come se fossero reali. Tra i dispositivi pensati per la MR troviamo *Magic Leap 2*, *Microsoft HoloLens 2* e *Varjo XR-4*.
- **Realtà estesa (XR):** Il termine XR è un'espressione generica che racchiude tutte le tecnologie immersive, includendo VR, AR e MR.

La crescente diffusione di queste tecnologie, in particolare della realtà aumentata, testimonia il suo potenziale applicativo in contesti pratici e professionali.

In questa condizione di espansione, si può confermare che si tratta di una delle tecnologie più promettenti del XXI secolo. Le previsioni di mercato confermano la tendenza: si stima infatti che il settore del software AR raggiungerà un valore di 17,98 miliardi di dollari entro il 2029, con una crescita del 38,58% nel periodo 2024-2029 [39]. Questo sviluppo non solo riflette un crescente interesse da parte del mercato, ma testimonia anche il progressivo perfezionamento della tecnologia e la sua integrazione in contesti sempre più diversificati.

Le applicazioni della realtà aumentata si estendono ormai a numerosi settori, tra cui musei, siti culturali, centri commerciali, turismo urbano, campus universitari, fiere, esposizioni, strutture sanitarie, ambienti industriali e il settore immobiliare.

In questo scenario di rapida evoluzione si inserisce il presente lavoro di tesi, che propone di esplorare le potenzialità della realtà aumentata nell'ambito della navigazione e del turismo.

Il progetto sviluppato è un'applicazione Android realizzata con Unity che guida gli utenti verso punti di interesse in luoghi specifici, dando informazioni contestuali attraverso l'uso di tecnologie all'avanguardia come gli LLM, creando così una vera e propria guida turistica digitale. La soluzione farà uso di modelli 3D in scala 1:1 che rappresentano le varie location e utilizzerà un sistema di navigazione con indicazioni visive per guidare l'utente.

L'azienda presso la quale ho avuto l'opportunità di svolgere il mio tirocinio formativo e contribuire attivamente allo sviluppo del progetto è **Touchlabs** di Bologna. Touchlabs si è affermata nel panorama nazionale e internazionale grazie alla sua specializzazione nella creazione di soluzioni digitali quali siti web, applicazioni mobile e smartwatch, strategie di web marketing e, in particolare, nelle tecnologie di realtà virtuale e realtà aumentata. Durante il periodo di tirocinio ho potuto sperimentare direttamente l'ambiente aziendale, partecipando alla progettazione e allo sviluppo dell'applicazione **ARTour** che verrà presentata nei prossimi capitoli.

Questa esperienza ha rappresentato un'importante occasione per mettere in pratica le competenze teoriche acquisite durante il percorso accademico, confrontandomi con un contesto professionale, dinamico e altamente innovativo nel settore delle tecnologie immersive.



Figura 1: Logo di Touchlabs

Capitolo 1

Realtà Aumentata

1.1 Definizione

Nella sua fondamentale pubblicazione, Azuma [5] definisce la realtà aumentata non come mera tecnologia legata a dispositivi specifici (es. Head-Mounted Display), ma come un sistema che deve soddisfare tre requisiti fondamentali:

1. Combinazione di elementi reali e virtuali in un singolo ambiente;
2. Interattività real-time tra utente e contenuti digitali;
3. Registrazione di oggetti virtuali allineandoli col mondo reale.

Questa definizione, ancora oggi punto di riferimento nella letteratura scientifica, cerca di distinguere l'AR da tecnologie simili quali:

- **Effetti speciali cinematografici:** Uniscono reale e virtuale ma mancano di interattività.
- **Overlay bidimensionali su video (es. filtri social):** Sono privi di registrazione 3D contestuale.
- **Realtà Virtuale (VR):** Diversa, in quanto sostituisce l'intero ambiente reale anziché arricchirlo.

Di grande importanza è anche il framework teorico di Milgram et al. [26], che va a introdurre il concetto fondamentale di **Reality-Virtuality Continuum** il quale presenta una scala lineare per descrivere le varie possibilità di unione tra ambienti reali e virtuali. L'AR si posiziona nella parte a sinistra, più vicina all'ambiente reale che a quello virtuale e fa parte della Mixed Reality (MR).

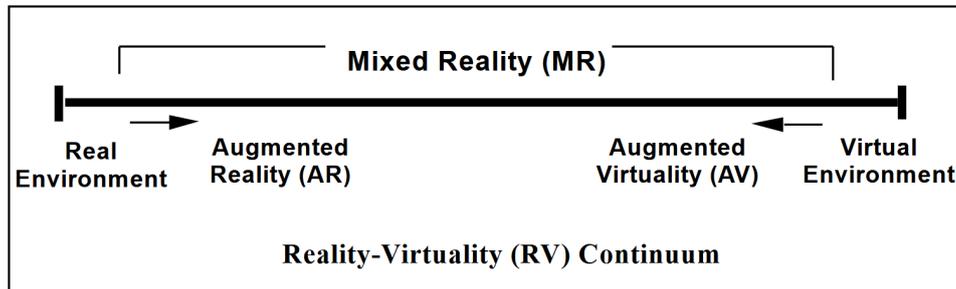


Figura 1.1: Reality-Virtuality (RV) Continuum

1.2 Storia

1.2.1 Dalle Origini ai Primi Esperimenti

Secondo quanto riportato in Nee e Ong [29], questa tecnologia ha origini sorprendentemente antiche.

L'autore Frank L. Baum (famoso per aver scritto *Il Meraviglioso Mago di Oz*) anticipò questo concetto nel suo romanzo *The Master Key*; egli descriveva degli occhiali capaci di mostrare il carattere delle persone attraverso delle lettere posizionate sopra le loro teste (es. "G" per i buoni, "E" per i cattivi).

I primi veri esperimenti avvennero nel 1957, quando il cinematografista americano Morton Heilig descrisse un'idea riguardante un teatro multisensoriale la quale venne brevettata nel 1962 sotto il nome di *Sensorama*. Si trattava di un dispositivo meccanico capace di proiettare video 3D su un display stereoscopico a colori, il tutto accompagnato da suoni, profumi, vento e vibrazioni tramite una sedia mobile. L'obiettivo era quello di simulare diverse esperienze come quella di un giro in motocicletta.

1.2.2 La Nascita dei Sistemi di Visualizzazione

Nel 1968, Ivan Sutherland rivoluzionò il campo con *The Sword of Damocles*, il primo head-mounted display (HMD) dotato di tracking meccanico. Questo dispositivo permetteva di sovrapporre degli oggetti 3D al mondo reale attraverso delle lenti semi-trasparenti. Negli stessi anni, Thomas Furness sviluppò per l'aeronautica militare statunitense il *Virtual Cockpit*, un simulatore di volo basato su HMD che forniva in tempo reale dati di navigazione. Questo progetto dimostrò il potenziale della realtà aumentata nel campo delle simulazioni, anticipando molte delle applicazioni future della tecnologia.

1.2.3 Consolidamento e Definizione

Negli anni '70 e '80 possiamo vedere la nascita dei primi progetti collaborativi: nel 1975, Myron Krueger creò *Videoplace*, un sistema che combinava tecniche di proiezione e varie videocamere per mostrare delle ombre sullo schermo. La particolarità stava nel fatto che utenti in luoghi diversi potevano interagire in uno spazio condiviso, anticipando il concetto di realtà condivisa. Nel 1984 lo studioso Stephen W. G. Mann presentò *Eye-Tap*, degli occhiali digitali che permettevano la sovrapposizione di immagini generate dal computer alla visione dell'utente.

Negli anni '90, il termine *augmented reality* venne formalizzato da due ingegneri di Boeing: Thomas Caudell e David Mizell, che utilizzarono questa tecnologia per guidare il cablaggio negli aeromobili. Parallelamente, Louis Rosenberg sviluppò *Virtual Fixtures*, un sistema AR nato per addestrare i piloti militari che andava a combinare feedback aptici e overlay visivi. Nel 1993, il progetto *KARMA* della Columbia University presentò il primo prototipo di un sistema AR per supportare attività di manutenzione.

Uno dei maggiori contributi teorici avvenne nel 1994, con Paul Milgram che introdusse il concetto di *Reality-Virtuality (RV) Continuum* il quale rimane tutt'ora un riferimento accademico fondamentale. Nello stesso anno, l'AR fece il suo debutto in ambito teatrale con lo spettacolo *Dancing in Cyberspace* e in campo medico con applicazioni di visualizzazione di feti in 3D. Nel 1996, Sony Computer Science Laboratories presentò il primo sistema AR basato su marker 2D, denominato *CyberCode*. Questo divenne il modello di riferimento per i futuri progetti AR marker-based.

1.2.4 L'Era Digitale: Diffusione e Standardizzazione

Con l'avvento dell'era digitale, l'AR diventò una tecnologia sempre più diffusa; nel 1999, il professore Hirokazu Kato rilasciò *ARToolKit*, la prima libreria open-source per il tracking di marker 2D, ancora oggi ampiamente utilizzata e adottata poi in numerosi progetti come *ARQuake (2002)* e *Wikitude (2008)*, ovvero il primo browser AR per Android.

Nei giorni nostri, l'AR è un pilastro dell'Industria 4.0. Progetti come *Google Glass (2013)* e *Microsoft HoloLens (2016)* hanno permesso di introdurla anche in contesti di formazione industriale e manutenzione predittiva. Dal lato consumer, giochi come *Pokémon GO (2016)* e app di navigazione come *Google Maps Live View* hanno dimostrato un enorme successo combinando geolocalizzazione ibrida (GPS + SLAM) e AR. Infine, nel 2017, nascono gli SDK *ARKit* di Apple e *ARCore* di Google, che hanno reso ancora più accessibile questa tecnologia agli sviluppatori di tutto il mondo, garantendo esperienze tecnologicamente complesse anche su smartphone.

1.3 Tipologie di AR

La realtà aumentata può essere classificata in due tipologie principali: *marker-based AR* e *markerless AR*.

1.3.1 Realtà Aumentata a marcatori (Marker-Based AR)

Si basa sul riconoscimento di un'immagine statica, detta marker o trigger photo, che l'utente deve scansionare con il proprio dispositivo per attivare i contenuti AR. Una volta identificato il marker, il sistema ne utilizza la posizione, l'orientamento e la dimensione come riferimento per sovrapporre elementi virtuali nel mondo reale.

Questo approccio rappresenta una delle prime implementazioni della tecnologia ed è relativamente semplice da sviluppare grazie alla presenza di riferimenti visivi ben definiti. Il riconoscimento dei marcatori può avvenire in due modalità:

- **Locale:** Il database dei marker è memorizzato direttamente sul dispositivo, permettendo un'elaborazione rapida e offline.
- **Cloud:** Il dispositivo invia le nuvole di punti a un server remoto che esegue il riconoscimento, garantendo maggiore scalabilità e aggiornabilità.

È importante notare che la trigger photo deve essere univoca: se si utilizza un sistema di riconoscimento basato su cloud, non è possibile impiegare immagini generiche o foto stock, poiché potrebbero essere già in uso in altri progetti, causando interferenze e problemi di identificazione.

Vediamo ora qualche esempio che va ad applicare questa tipologia di AR:

- **Invizimals:** Sviluppato da Novarama e rilasciato per PlayStation Portable (PSP) nel 2009, rappresenta uno dei primi esempi di videogiochi basati su marker AR che hanno avuto successo. Il gioco sfruttava la PSP Camera insieme a una speciale trap card che veniva data nella confezione e che fungeva da marker. I giocatori dovevano inquadrare questa carta per far apparire le creature virtuali, chiamate Invizimals. L'interazione avveniva sempre attraverso il riconoscimento del marker, che fungeva da punto di ancoraggio per il contenuto virtuale.
- **HP Reveal (precedentemente nota come Aurasma):** Questa app è stata ormai abbandonata, ma risulta essere un esempio comunque importante: sfruttava marker stampati come immagini, loghi o illustrazioni, per attivare contenuti digitali. Ad esempio, un dipinto o un poster poteva essere "animato" con informazioni interattive una volta riconosciuto dal sistema.



Figura 1.2: Invizimals, la trap card funge da marker *Source:* [15]

1.3.2 Realtà aumentata senza marcatori (Markerless AR)

Questa tecnologia è nata successivamente, quando l'evoluzione della computer vision, l'incremento della potenza di calcolo dei dispositivi mobili e lo sviluppo di algoritmi più sofisticati hanno reso possibile il riconoscimento e il tracking dell'ambiente in real-time senza la necessità di marcatori fisici.

Il sistema identifica elementi naturali come superfici piane (pavimenti, tavoli), pareti, oggetti, o altre caratteristiche per capire come posizionare i contenuti digitali. Gli SDK più recenti come ARCore di Google e ARKit di Apple usano proprio questo approccio: attraverso la fotocamera e la combinazione di dati ottenuti da vari sensori (accelerometro, giroscopio) vanno a creare una mappa dell'ambiente e a posizionare gli oggetti digitali in modo preciso e naturale.

Alcuni esempi particolarmente importanti sono:

- **Pokémon GO:** Videogioco per dispositivi mobili rilasciata da Niantic nel 2016, in cui è possibile catturare e vedere i Pokémon tramite la propria fotocamera nell'ambiente circostante. L'app riconosce automaticamente il terreno e le superfici permettendo ai Pokémon di apparire senza l'utilizzo di marker.
- **Applicazione Amazon:** L'applicazione di Amazon per dispositivi mobili ha una funzionalità chiamata *view in your room* che permette di mostrare il prodotto nell'ambiente circostante prima di procedere all'acquisto.
- **Google Maps Live View:** Introdotta nel 2019, si tratta di una funzionalità integrata all'interno dell'app di Google Maps che permette di sovrapporre al campo

della fotocamera elementi digitali come frecce, indicazioni stradali e informazioni sui punti di interesse che guidano l'utente alla destinazione voluta. Prima di poterla usare è necessario scansionare l'ambiente circostante per far capire la posizione dell'utente al sistema.



Figura 1.3: Pokémon GO *Source:* [47]

1.4 Framework per Sistemi AR

1.4.1 SDK disponibili

ARCore e ARKit

Sono i principali SDK per lo sviluppo di applicazioni di realtà aumentata sviluppati da Google e da Apple per Android e iOS rispettivamente [16].

- **Definizione e integrazione** ARCore supporta Android 7.0 o successivi e utilizza la fotocamera del dispositivo insieme a dati raccolti da sensori come accelerometri, giroscopi e magnetometri. Questo gli permette di tracciare il movimento e l'orientamento del dispositivo nello spazio in modo accurato.

ARKit, invece, è stato sviluppato per iOS 11 o successivi e funziona con chip A9 e superiori. Sfrutta hardware dedicato come *LiDAR (Light Detection and Ranging)* che permette di misurare le distanze attraverso impulsi luminosi generando nuvole di punti che mappano l'ambiente in 3D in modo da ottenere depth map molto precise.

- **Tracciamento del dispositivo:** Avviene attraverso la tecnica *6DoF (Six Degrees of Freedom)* che va a combinare la lettura dei movimenti di traslazione (avanti/indietro, sinistra/destra, su/giù) e di rotazione (pitch, yaw, roll) nello spazio tridimensionale.

Entrambi gli SDK utilizzano il metodo *Simultaneous Localization And Mapping (SLAM)*, che sfrutta i feature point, ovvero punti visivamente distinti all'interno del campo visivo della fotocamera, per fornire dei riferimenti spaziali utili a determinare la posizione del dispositivo.

Viene inoltre impiegata la tecnica *Visual Inertial Odometry (VIO)*, che combina i dati visivi della fotocamera con quelli dei sensori inerziali (accelerometro e giroscopio) per stimare con precisione la posizione del dispositivo rispetto agli elementi fisici circostanti, come pareti, soffitto, pavimento, mobili e altro.

- **Comprensione ambientale:** ARCore si basa principalmente sul rilevamento di feature points per identificare superfici piane (*plane detection*), ma non offre nativamente una ricostruzione della mesh 3D.

ARKit, dall'altra parte, riesce mediante LiDAR a generare una mesh 3D dettagliata dell'ambiente circostante.

- **Analisi della profondità:** ARCore utilizza *Depth API* che scatta più immagini e poi le confronta per stimare la distanza da ogni pixel e creare una struttura tridimensionale dell'ambiente circostante.

ARKit, invece, utilizza sensori appositi (LiDAR) per creare delle depth map molto precise.

- **Face Detection:** ARCore usa la tecnologia *Augmented Face* che consente di tracciare la geometria del viso, la posizione della punta del naso, i lati della fronte e il centro della testa, in modo da ricreare un modello 3D del volto. ARKit, grazie alla tecnologia *Face Tracking, Multiple Face Tracking* e il sensore *TrueDepth*, permette il tracciamento di più volti contemporaneamente seguendone il movimento e le espressioni facciali.

AR Foundation

AR Foundation è un SDK gratuito e open-source sviluppato da Unity Technologies che consente di creare applicazioni di realtà aumentata (AR) multiplatforma. È stato scelto come base per lo sviluppo di questo progetto.

Il pacchetto permette essenzialmente di abilitare le funzionalità dell'SDK nativo per la piattaforma dove viene eseguita l'applicazione [43].

- Per dispositivi Android, AR Foundation utilizza *ARCore XR Plug-in* per interfacciarsi con ARCore.

- Per dispositivi iOS, viene utilizzato *ARKit XR Plug-in* per interfacciarsi con ARKit.

AR Foundation funge quindi da wrapper tra Unity e gli SDK nativi, creando un livello di astrazione che permette agli sviluppatori di scrivere un unico codice compatibile con più piattaforme, semplificando così lo sviluppo e la manutenzione.

AR Foundation introduce alcuni oggetti chiave per il ciclo di vita dell'applicativo: [44]

- **ARSession:** Gestisce il ciclo di vita della sessione AR.
- **ARSessionOrigin (*deprecato*):** Consente di convertire lo spazio di tracciamento generato dalla sessione AR (il cosiddetto session space) nello spazio del mondo Unity (Unity world space). Va quindi ad applicare le necessarie trasformazioni per fare in modo che i dati di tracking forniti dal dispositivo, come la posizione e l'orientamento della fotocamera, corrispondano correttamente alle coordinate della Scena Unity.
- **XR Origin (*nuovo*):** Sostituisce *ARSessionOrigin* da AR Foundation 5.0 e svolge il suo stesso compito ma unificato sia per l'AR che la VR. Viene quindi utilizzato in tutte le Scene XR, invece che essere specifico per soltanto quelle AR.
- **ARCameraManager:** Controlla le impostazioni della fotocamera e gestisce funzionalità come l'autofocus e la stima della luce.
- **ARPlaneManager:** Permette di rilevare superfici piane nel mondo reale.
- **ARRaycastManager:** Permette di effettuare raycast sul mondo reale per identificare dove posizionare oggetti, interagire con superfici e rilevare punti di interesse.
- **AR Anchor Manager:** Gestisce la creazione e il mantenimento degli ancoraggi (anchors) per fissare gli oggetti virtuali a punti specifici nel mondo reale.

Vuforia

Vuforia è un altro SDK che consente di creare applicazioni di realtà aumentata. È stato sviluppato dall'azienda statunitense Qualcomm e nel 2015 è stato acquisito dalla multinazionale PTC [54].

La differenza principale rispetto a AR Foundation sta nel fatto che Vuforia funziona utilizzando un layer completamente separato da Unity e sfrutta un motore di computer vision proprietario, quindi non si appoggia a ARCore/ARKit. Inoltre, non è limitato solo a Unity, è possibile infatti integrarlo con numerose altre soluzioni quali Android Studio, Xcode e addirittura Unreal Engine o framework basati su OpenCV.

Di seguito è possibile trovare alcune delle sue caratteristiche principali: [33]

- **Image Targets:** Permettono di riconoscere immagini bidimensionali quali loghi, poster o opere d'arte, e utilizzare questi elementi come ancore per l'integrazione di contenuti AR.
- **Multi Targets:** Offrono la possibilità di riconoscere e tracciare simultaneamente più immagini o superfici. È quindi possibile creare un oggetto con più facce, come un cubo o un pacchetto, dove ognuna di esse può mostrare contenuti specifici.
- **Cylinder Targets:** Vengono utilizzati per il riconoscimento di oggetti cilindrici, come bottiglie, lattine o altri imballaggi. Consentono di posizionare contenuti digitali attorno a articoli di forme irregolari. Sono particolarmente importanti in ambito marketing e packaging, dove il design gioca un ruolo fondamentale.
- **Model Targets:** Permettono il riconoscimento e il tracking di oggetti tridimensionali. Utilizzando modelli CAD o 3D il sistema è in grado di confrontare l'oggetto virtuale con quello reale e verificare se corrispondono. Questo è utile in contesti industriali o di manutenzione, dove è fondamentale sovrapporre informazioni tecniche e istruzioni direttamente sull'oggetto stesso.
- **Area Targets:** Permettono di riconoscere interi ambienti fisici (come stanze o corridoi) tramite scansione 3D. Questa funzionalità consente di ancorare contenuti digitali in determinati punti e in contesti quali musei o centri espositivi.
- **Ground Plane:** Consente di posizionare contenuti digitali su superfici piane, come pavimenti o tavoli, senza l'utilizzo di marker. Questa tecnologia è ideale per anteprime di prodotti, arredamento virtuale e dimostrazioni interattive.
- **VuMarks:** Permettono di personalizzare graficamente i marker e di fargli trasmettere dati univoci associati all'oggetto stesso. Gli sviluppatori possono creare esperienze AR in cui il marker non solo avvia il riconoscimento, ma porta con sé informazioni specifiche che guidano la logica dell'applicazione.

In questo lavoro, tuttavia, si è scelto di non utilizzare il pacchetto Vuforia: sebbene sia gratuito per uso personale, richiede un abbonamento per usi commerciali e per sfruttarne a pieno le capacità; AR Foundation è gratuito, open-source e necessita soltanto di una licenza Unity per poterlo integrare nei progetti. Inoltre, sebbene Vuforia offra un maggior numero di funzionalità, AR Foundation supporta le ultime novità introdotte proprio perché funge da layer per ARCore/ARKit.

Capitolo 2

Strumenti e Tecnologie di Sviluppo

2.1 Unity

Unity è un motore di grafico multiplatforma sviluppato da Unity Technologies. È stato annunciato e rilasciato nel 2005 esclusivamente per Mac OS X, ma da allora si è evoluto per supportare varie piattaforme desktop, mobile e console.

Grazie alla sua versatilità, Unity è ampiamente utilizzato per lo sviluppo di videogiochi su smartphone e rappresenta un'ottima scelta per utenti neofiti. La sua accessibilità e facilità d'uso lo hanno reso particolarmente popolare nel settore indie, pur essendo adottato anche da studi professionali (AAA).

Oltre alla creazione di videogiochi 2D e 3D, Unity viene impiegato in numerosi altri ambiti, tra cui simulazioni interattive, esperienze in realtà aumentata (AR) e realtà virtuale (VR), nonché nella produzione di cortometraggi animati. La sua adozione si estende anche a settori non videoludici come il cinema, l'industria automobilistica, l'architettura, l'ingegneria, le costruzioni e persino l'ambito militare.

Si basa su scripting tramite linguaggio C# e garantisce un'integrazione completa con *Visual Studio* e altri IDE come *JetBrains Rider*.

Offre inoltre una versione gratuita, chiamata *Unity Personal* che permette di pubblicare i propri progetti a patto che il profitto sia inferiore a 200.000 USD negli ultimi 12 mesi. Per esigenze più avanzate, sono disponibili piani in abbonamento con funzionalità aggiuntive:

Personal, Pro, Enterprise, Industry. [40]

Gli utenti di tier superiori hanno a disposizione caratteristiche aggiuntive: dal livello Enterprise sarà infatti possibile richiedere l'accesso al codice sorgente, a un supporto specifico per sviluppatori e molto altro.

Grazie alla sua lunga presenza sul mercato, Unity ha costruito una vasta comunità di sviluppatori e una ricca libreria di risorse. La piattaforma dispone di una documenta-

zione dettagliata e di un ampio catalogo di video tutorial, facilitando l'apprendimento e l'adozione da parte di nuovi utenti.

2.2 Architettura di Unity

2.2.1 GameObjects e Components

- **GameObject:** È l'unità di base di Unity. Ogni oggetto presente nel progetto è in realtà un GameObject, inoltre, è possibile visualizzare la lista di quelli presenti nell'attuale Scena attraverso la *Hierarchy View*. Luci, telecamere, sprite, effetti particellari, sono tutti GameObject e, ad ognuno di essi, bisogna dare delle proprietà e dei *Components* in modo da fargli eseguire dei compiti. Possiamo quindi vedere un GameObject così:

$$\text{GameObject} = \text{Proprietà} + \text{Components}$$

Unity offre una vasta gamma di Components built-in, ma consente anche di crearne di personalizzati tramite script. Inoltre, i GameObjects possono contenere al loro interno altri GameObjects, permettendo così una gestione gerarchica e un'organizzazione strutturata attraverso il meccanismo del *parenting*.

Quando si crea un nuovo GameObject, questo include automaticamente il Component *Transform*, che è essenziale per definirne la posizione, la rotazione e la scala all'interno della Scena. Senza di esso, l'oggetto non avrebbe alcuna collocazione nello spazio virtuale.

- **Component:** Sono dei blocchi modulari di funzionalità che vengono collegati ai GameObject e ne descrivono i compiti che devono eseguire: quando un oggetto richiede un comportamento specifico, basta aggiungere il Component corrispondente. Ad esempio, se un corpo deve essere affetto dalla gravità, è necessario assegnargli un *Rigidbody* che gli permette di interagire con il motore fisico di Unity; potrà quindi essere spostato da forze esterne e reagire alle collisioni. Inoltre, è possibile creare Component personalizzati tramite Script: *PlayerMovement.cs* può essere utilizzato per gestire il movimento di un personaggio in risposta agli input della tastiera. Tuttavia, affinché uno Script possa essere utilizzato come Component, deve necessariamente ereditare dalla classe base *MonoBehaviour*. Una caratteristica cruciale dei Component è la flessibilità: sarà infatti possibile regolare diverse impostazioni nell'editor in modo da ottenere il comportamento desiderato.

2.2.2 Scene e Hierarchy

Ogni progetto sviluppato con Unity può essere visto come un insieme di **Scene**, ovvero risorse che contengono un `GameObject` e possono essere usate per creare un menu principale, delle schermate individuali, dei livelli o qualsiasi altra cosa. Permettono quindi di progettare l'applicazione utilizzando un approccio modulare.

In ogni Scena, verranno collocati i `GameObject` che dovranno essere presenti per farla funzionare in modo appropriato, come ad esempio il giocatore, eventuali nemici, una telecamera, ecc...

La finestra *Hierarchy* fornisce una panoramica di tutti i `GameObject` presenti nella Scena stessa: al suo interno vengono visualizzati elementi come modelli, telecamere e interfacce grafiche, consentendo allo sviluppatore di aggiungere, eliminare, organizzare e raggruppare gli oggetti con facilità.

I `GameObject` sono rappresentati in una struttura ad albero che permetterà di configurarli come *parent* (genitori) o *child* (figli) di altri oggetti, creando quindi delle relazioni gerarchiche in modo semplice e veloce. Quando un oggetto diventa parent, tutti i suoi child ereditano le trasformazioni applicate al genitore, mantenendo però la loro posizione relativa. Questo meccanismo è particolarmente utile per gestire insiemi di elementi correlati.

Ad esempio, un menu di gioco potrebbe essere strutturato con un `GameObject` principale come parent, contenente al suo interno pulsanti, testi e immagini come child. In questo modo, qualsiasi modifica alla posizione o alla scala del menu influenzerà automaticamente tutti i suoi elementi, semplificando la gestione della UI.

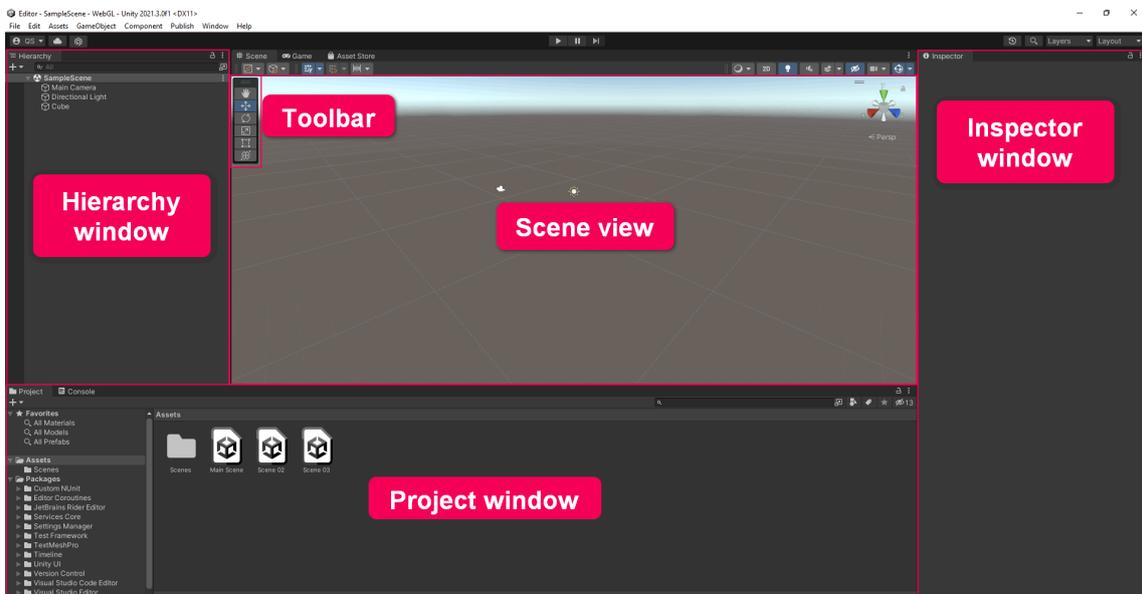


Figura 2.1: Unity Editor *Source:* [45]

2.2.3 Asset Workflow

Prefab e Resource System

Un **Prefab** è un `GameObject` memorizzato in modo persistente all'interno del progetto, completo di tutti i suoi `Components` e proprietà. Può essere visto come un template da cui è possibile creare nuove istanze di `GameObject` all'interno della Scena.

Ad esempio, se si vuole riutilizzare un `GameObject` configurato in un certo modo (come quello di un menu) più volte nella stessa scena o in scene diverse, è buona pratica convertirlo in un Prefab. Questo approccio è preferibile rispetto alla semplice duplicazione, poiché mantiene sincronizzate tutte le istanze: qualsiasi modifica apportata al Prefab originale si rifletterà automaticamente su tutte le sue copie.

Le **Resource** sono invece asset salvati all'interno del progetto e differiscono dai Prefab per il modo in cui vengono caricati. Mentre i Prefab vengono istanziati dinamicamente nella Scena, le Resource vengono caricate così come sono e possono essere recuperate a runtime tramite il metodo `Resources.Load()`.

Da notare che i cambiamenti apportati nell'editor a una Resource non vanno a modificare il file nel progetto, e quindi si applicheranno solo al particolare oggetto su cui sono stati fatti.

```
public class ResourcesExample : MonoBehaviour {

    private TextAsset textAssetExample;

    private void Start() {
        // Carica un file di testo (JSON, CSV, TXT, ecc.)
        textAssetExample = Resources.Load<TextAsset>("Data/MyFile");
        if (textAssetExample != null) {
            Debug.Log("Contenuto del file: " + textAssetExample.text);
        }
    }
}
```

Listing 2.1: Esempio di caricamento di Resources

Project Window

In questa finestra vengono mostrati tutti i file relativi al progetto ed è il modo principale per navigare all'interno della sua struttura.

Sarà possibile accedere a modelli, texture, Script, Prefab e Scene.

Asset Store

Lo Unity Asset Store è una libreria di risorse a pagamento o gratuite rese disponibili agli utenti che utilizzano il software.

Gli Asset qui presenti vengono creati e pubblicati da Unity Technologies o dalla community e comprendono texture, animazioni, modelli 3D, pacchetti di suoni, esempi di progetti, tutorial e persino estensioni per l'editor.

È inoltre possibile visitarlo in due modi: tramite il sito web o attraverso Unity stesso.

Tra gli Asset disponibili nello Store possiamo trovare:

- **2D/3D Assets:** La sezione 2D include sprites, texture, personaggi, ambienti, fonts, materiali ed elementi della UI. Quella 3D, veicoli, personaggi, oggetti di scena, vegetazione e animazioni.
- **Add-ons:** Si tratta di funzioni più avanzate che possono essere importate nel progetto, come ad esempio Unity Ads, acquisti in-app e strumenti di analisi.
- **Audio:** In questa sezione sono presenti librerie di file audio come ad esempio musica, effetti sonori, sottofondo ambientale, ecc.
- **Templates:** Qui è possibile scaricare vari tutorial e starter pack, utili soprattutto per i principianti.
- **Tools:** Permette di scaricare strumenti utili allo sviluppo come AI e Visual Scripting.
- **VFX (Visual Effects):** Ovvero gli effetti visivi. In questa sezione è possibile scaricarne ed importarne di già pronti. Tra gli esempi più comuni possiamo trovare effetti particellari e Shader.

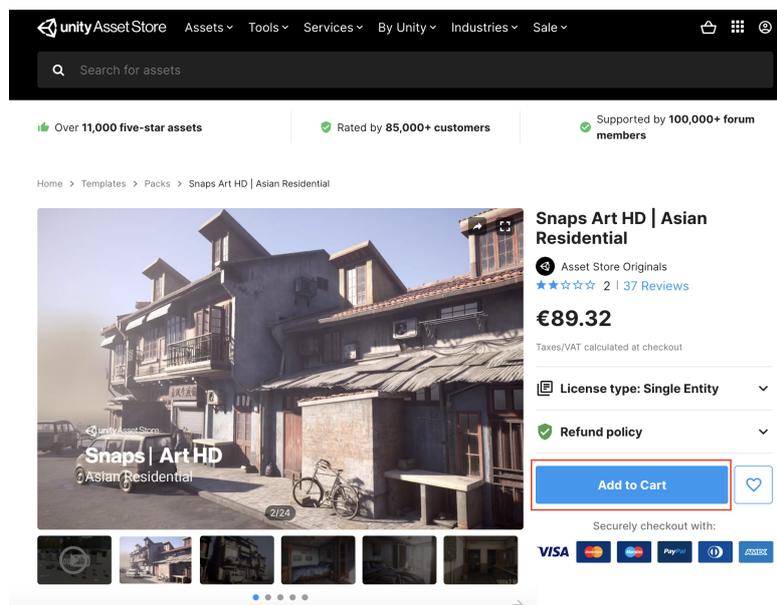


Figura 2.2: Pagina iniziale dell'Asset Store *Source:* [46]

2.3 Sviluppo Cross-Platform

Uno dei punti di forza di Unity è la sua filosofia *Build once, deploy anywhere*, che consente agli sviluppatori di scrivere il codice dell'applicazione una sola volta, rendendolo eseguibile su tutte le piattaforme supportate.

Questo approccio migliora significativamente l'efficienza dello sviluppo, offrendo un livello di astrazione che semplifica la gestione degli elementi più complessi. Unity infatti nasconde molte delle difficoltà legate alle operazioni di basso livello attraverso componenti e API ad alto livello, permettendo agli utenti di concentrarsi sulla logica applicativa piuttosto che sui dettagli tecnici. Ad esempio, non sarà necessario interfacciarsi direttamente con OpenGL o con le API di ARCore/ARKit, in quanto è tutto già gestito da Unity, riducendo così notevolmente il carico di lavoro.

Sul piano pratico, lo sviluppo multipiattaforma richiede comunque alcuni adattamenti. In questo caso Unity semplifica un processo fondamentale che prende il nome di *porting*, ovvero l'adattamento del software per l'esecuzione in ambienti diversi da quello originale (ad esempio, su CPU o sistemi operativi differenti) [51].

A differenza dello sviluppo tradizionale, dove il porting può richiedere la riscrittura di ampie porzioni di codice, con Unity gli interventi si limitano spesso a ottimizzazioni specifiche e configurazioni di build, riducendone drasticamente la complessità.

È inoltre possibile fare il deployment della propria applicazione su un ampio numero di piattaforme, tra cui:

- Dispositivi mobili (Android, iOS);
- Piattaforme di realtà virtuale (Meta Quest, HTC Vive, ...);
- Sistemi desktop (Windows, macOS, Linux);
- Console di gioco (PlayStation 5, Xbox Series X, Nintendo Switch, ...);
- Smart TV;
- Web.

2.3.1 Mono Framework

Alla base dello sviluppo cross-platform in Unity troviamo il *Mono Framework*, un progetto open-source nato nel 2001 come implementazione alternativa del framework .NET. Grazie a Mono è possibile eseguire applicazioni sviluppate con tecnologie Microsoft anche su sistemi operativi diversi da Windows. Nel corso degli anni, Mono ha ampliato il proprio supporto fino a includere quasi tutte le funzionalità di .NET, continuando a evolversi con una suite di strumenti sempre più completa.

Per ogni applicazione sviluppata con Unity, il *Mono Runtime Environment* viene incluso

nella build e funge da macchina virtuale per l'esecuzione del codice su piattaforme differenti. Il bytecode compilato viene quindi convertito in codice nativo direttamente dal runtime, garantendo compatibilità con vari sistemi operativi.

Un aspetto fondamentale di Mono è che non incorpora l'intero framework all'interno della build: le classi inutilizzate vengono rimosse automaticamente, mantenendo solo le parti effettivamente necessarie all'applicazione. Nonostante questa ottimizzazione, le build di Unity tendono a essere di dimensioni considerevoli, proprio perché devono includere il runtime e le librerie indispensabili per garantire l'esecuzione su più piattaforme [48].

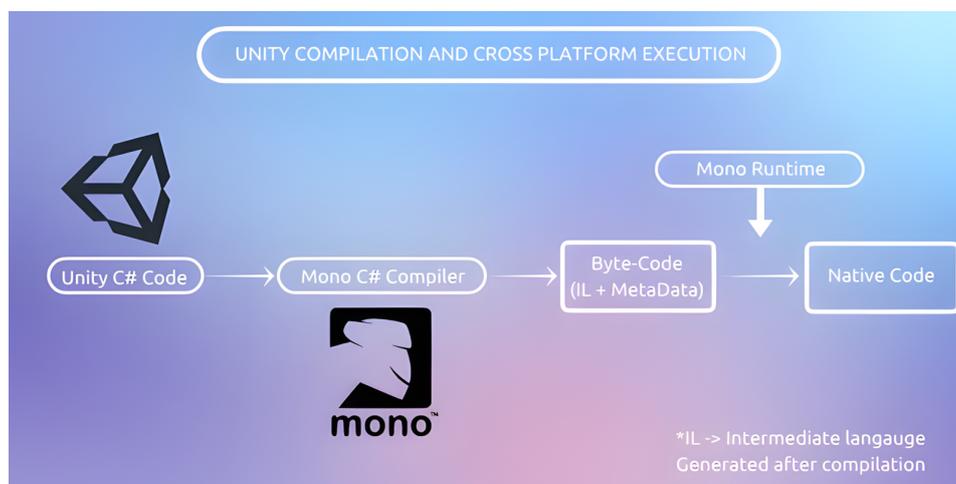


Figura 2.3: Unity Cross-Platform Execution. *Source:* [48]

Architettura di Mono Framework

Per comprendere appieno come Mono permetta a Unity di compilare per piattaforme diverse, è essenziale esaminare la sua architettura.

Il framework è composto da diversi componenti, ciascuno con un ruolo specifico nell'ecosistema di sviluppo: [28]

- **Compilatore C#:** Il compilatore C# di Mono è completo di funzionalità per C# 1.0, 2.0, 3.0, 4.0, 5.0 e 6.0 (ECMA).
- **Runtime di Mono:** Il runtime implementa il *Common Language Infrastructure (CLI)*. Il runtime fornisce un compilatore *Just-in-Time (JIT)*, un compilatore *Ahead-of-Time (AOT)*, un loader delle librerie, un garbage collector, un sistema di threading e funzionalità di interoperabilità.
- **Libreria di classi Framework .NET:** La piattaforma Mono fornisce un insieme completo di classi che definiscono una solida base per la creazione di applicazioni. Queste classi sono compatibili con le classi del framework .NET di Microsoft.

- **Libreria di classi Mono:** Vengono fornite anche molte classi che vanno oltre la *Base Class Library* di Microsoft le quali permettono l'aggiunta di funzionalità utili, soprattutto nella creazione di applicazioni Linux.

2.4 Large Language Model (LLM)

Un Large Language Model è un modello di intelligenza artificiale progettato per comprendere, generare ed elaborare il linguaggio naturale.

Gli LLM contengono un elevatissimo numero di parametri e vengono addestrati con un apprendimento di tipo *self-supervised* su una quantità di testo molto vasta.

Questi modelli acquisiscono capacità predittive riguardo alla sintassi, alla semantica e alle ontologie intrinseche nei corpora del linguaggio umano, ma ereditano anche bias e imprecisioni presenti nei dati di training.

I più grandi e potenti LLM sono attualmente i *Generative Pretrained Transformers (GPT)* una famiglia di cui fa parte anche *ChatGPT* che è sicuramente l'esponente più importante [52]. Rilasciato nel novembre del 2022 da OpenAI, con il suo modello denominato *GPT-3*, non solo ha introdotto l'intero mondo a questa nuova tecnologia, ma addirittura ne ha stabilito lo stato dell'arte e ha fatto capire le potenzialità che può raggiungere.

A differenza dei modelli linguistici tradizionali, che utilizzano tecniche statistiche per la predizione sequenziale delle parole, ChatGPT e simili implementano un'architettura chiamata *Transformer* che consente l'elaborazione parallela di enormi quantità di dati. Questa innovazione ha garantito una capacità di gran lunga superiore nel comprendere e generare testo: è possibile generare codice sorgente, produrre opere letterarie, composizioni poetiche e persino simulare contenuti scientifici come abstract accademici [2].

2.4.1 Architettura Transformer

È importante capire la tecnologia base di tali modelli, che prende il nome di *Transformer*. Si tratta di un'architettura di deep learning sviluppata dai ricercatori Google e fondata sul meccanismo di **multi-head attention**, proposto nel paper del 2017 *Attention Is All You Need*.

Inizialmente concepiti per migliorare la traduzione automatica, i Transformer hanno rapidamente trovato applicazione in numerosi campi: elaborazione del linguaggio naturale su larga scala, computer vision (*Vision Transformer*), reinforcement learning, elaborazione audio, apprendimento multimodale, robotica e persino in giochi come quello degli scacchi.

L'architettura originale si compone di due elementi principali: un encoder e un decoder. È interessante notare che non tutti i modelli utilizzano entrambi i componenti: *BERT*

(*Bidirectional Encoder Representations from Transformers*) si basa esclusivamente sugli encoder, mentre i modelli GPT impiegano solo i decoder.

Le varianti successive dei Transformer sono state ampiamente adottate per l'addestramento di Large Language Model (LLM) su vasti dataset linguistici, portando allo sviluppo di sistemi pre-addestrati che hanno rivoluzionato il campo dell'intelligenza artificiale. [53]

L'architettura è stata fundamentalmente progettata per trasformare o modificare una sequenza di input in una sequenza di output e il suo principio di funzionamento si articola in cinque fasi fondamentali [3].

1. **Tokenization:** Si utilizza un dataset di token contenente tutte le parole, i segni di punteggiatura e altri elementi testuali. Si analizza il testo in ingresso, scomponendo ogni parola, prefisso, suffisso e segno di punteggiatura per poi associarlo a dei token specifici presenti nel dataset di riferimento.
2. **Embedding:** Si procede alla conversione delle parole in rappresentazioni numeriche. Si mappa ogni elemento testuale in un vettore numerico; la caratteristica fondamentale di questa rappresentazione è che elementi testuali semanticamente simili vengono mappati in vettori con componenti numeriche simili, mentre elementi testuali semanticamente diversi vengono rappresentati da vettori con componenti significativamente differenti.
3. **Positional encoding:** Si vanno a combinare i vettori corrispondenti a ciascun token della frase precedente in un unico vettore che verrà poi elaborato. Il metodo più comune per unire un insieme di vettori in uno solo è sommarli componente per componente. Questo approccio può funzionare, ma presenta una limitazione importante: la somma è commutativa, il che significa che sommando gli stessi numeri in un ordine diverso si ottiene lo stesso risultato. Di conseguenza, la frase "*Non sono triste, sono felice*" e la frase "*Non sono felice, sono triste*" produrrebbero lo stesso vettore, dato che contengono le stesse parole ma in ordine diverso. Per ovviare a questo problema, è stata introdotta una tecnica chiamata *positional encoding*, che assegna a ogni token della frase un'informazione sulla sua posizione all'interno della sequenza. Questo viene fatto aggiungendo ai vettori di embedding delle parole una sequenza di vettori predefiniti, garantendo così che frasi con le stesse parole ma in ordine diverso vengano rappresentate da vettori distinti.
4. **Molteplici Transformer blocks:** Si va a predire la parola successiva nella frase utilizzando una rete neurale estremamente grande che viene addestrata precisamente con questo obiettivo. Il componente *attention* è la chiave di questo passaggio: lo si può vedere come un modo per aggiungere contesto a ogni parola presente nel testo e viene inserito in

ogni blocco della rete neurale che va a ottenere la prossima parola. Questa rete è composta da un insieme di blocchi formati da reti neurali più piccole.

5. **Layer Softmax:** Si trasformano i punteggi ottenuti in probabilità (che quindi avranno come somma 1). I punteggi più alti corrispondono alle probabilità maggiori.

Infine, effettuando un campionamento di queste probabilità, si può predire la parola successiva.

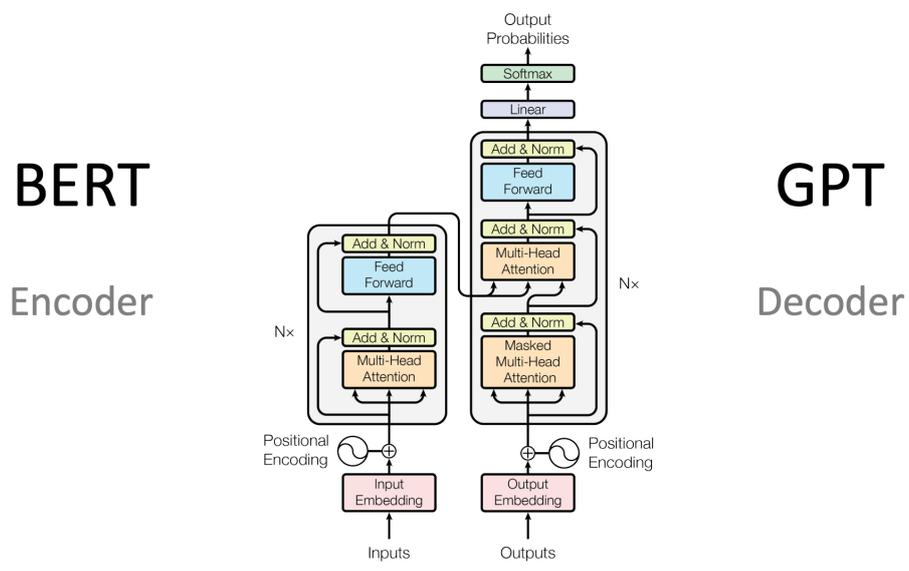


Figura 2.4: Architettura Transformer BERT e GPT *Source:* [19]

2.5 Matterport

Matterport è un'azienda specializzata nella digitalizzazione di spazi fisici attraverso la creazione di digital twin. La sua piattaforma consente agli utenti di realizzare tour virtuali interattivi, offrendo visualizzazioni dettagliate particolarmente utili nei settori dell'immobiliare, dell'architettura e del design.

Le soluzioni Matterport si basano su un'ampia gamma di fotocamere 3D e software avanzati, che permettono di eseguire scansioni tridimensionali di ambienti reali e di trasformarle in esperienze virtuali immersive. Il processo standard prevede la scansione dello spazio tramite queste fotocamere, seguita da una fase di post-produzione in cui i dati acquisiti vengono elaborati e ottimizzati mediante strumenti software dedicati. Inoltre, il sistema supporta l'integrazione con applicazioni per la sovrapposizione di tour virtuali personalizzati, adattabili alle esigenze specifiche degli utenti.

Il risultato finale è il *3D Virtual Tour*, un'esperienza interattiva che simula una visita reale e che consente agli utenti di esplorare gli ambienti senza vincoli di tempo. Tra le funzionalità più avanzate spiccano i *Mattertag*, punti di interesse interattivi che arricchiscono il tour con contenuti multimediali, come testo, immagini, audio o video. Questi strumenti possono essere impiegati in diversi contesti: ad esempio, un agente immobiliare può utilizzarli per evidenziare le caratteristiche chiave di una proprietà ai potenziali acquirenti, mentre un istituto scolastico può sfruttarli per agevolare l'orientamento degli studenti all'interno del campus [20] [35].

Metterport mette quindi a disposizione dell'utente diverse caratteristiche che lo hanno fatto emergere rispetto a soluzioni simili:

- Tour virtuali 3D per esperienze online immersive;
- Foto 4K ad alta risoluzione adatte per la stampa e il marketing digitale;
- Planimetrie schematiche accurate per la rappresentazione dettagliata della proprietà;
- Tour guidati e video personalizzati per professionisti del settore immobiliare impegnati.

2.6 Digital Twin

2.6.1 Definizione

Il concetto di Digital Twin è stato originariamente sviluppato in ambito aerospaziale dalla NASA e dall'Air Force. Nella sua definizione iniziale, un digital twin è descritto come una simulazione probabilistica integrata, multifisica e multiscala, di un veicolo o sistema "*così com'è costruito*" (as-built) che utilizza i migliori modelli fisici disponibili, aggiornamenti dei sensori, storico della flotta, ecc. per rispecchiare la sua versione fisica. [...]

Oltre a questo, il Digital Twin integra i dati dei sensori provenienti dal sistema di gestione della salute del veicolo (*IVHM*) di bordo, la cronologia della manutenzione e tutti i dati storici e della flotta disponibili che son stati ottenuti tramite data mining e text mining [18].

Sebbene questa definizione sia stata originariamente concepita per veicoli aerospaziali, il concetto si è dimostrato estremamente versatile, e, attualmente, un digital twin viene descritto come una replica virtuale di un oggetto fisico, di una persona o di un processo che può essere utilizzata per simularne il comportamento e capire meglio come funziona nella vita reale. Anche qui vengono collegati a fonti di dati provenienti dall'ambiente, per far sì che si aggiornino in tempo reale riflettendo la versione originale.

I digital twin, se interconnessi all'interno di un sistema, possono creare un ambiente digitale che replica ogni aspetto di un'organizzazione, permettendo di: [24]

- Eseguire simulazioni più accurate;
- Pianificare diversi scenari possibili;
- Prendere decisioni più informate.

2.6.2 Nell'industria

Ad oggi i digital twin sono uno strumento che sta prendendo piede nell'ambito dell'innovazione industriale. Queste rappresentazioni virtuali consentono di descrivere digitalmente dei sistemi, processi e asset fisici, permettendo alle aziende di monitorare, analizzare e ottimizzare le proprie operazioni in tempo reale.

Tra i principali tipi di digital twin in questo settore possiamo trovare: [32]

- **Digital Twin di Componenti:** Rappresenta il livello base e corrisponde agli elementi più piccoli del sistema, ovvero ad una determinata parte dell'apparecchiatura o del prodotto (es. un sensore, un interruttore, una valvola, ecc). Grazie alla rappresentazione virtuale, è possibile monitorarne le prestazioni e simulare le condizioni operative per testarne resistenza, stabilità e efficienza. Così facendo si facilita la manutenzione tempestiva e si assicura la qualità del prodotto finale.
- **Digital Twin di Prodotti o Asset:** Questo livello combina diversi twin per creare una rappresentazione virtuale di asset più complessi come un motore, una pompa o un edificio. Questi digital twin analizzano come le singole parti interagiscono tra loro e come funzionano nell'ottica di una soluzione complessiva. Ciò permette agli ingegneri di monitorare le prestazioni, identificare possibili miglioramenti, aumentare la produttività, ridurre il tempo medio tra i guasti, il tempo di riparazione e diminuire il consumo di risorse.
- **Digital Twin di Sistema:** Permettono di duplicare gli asset a livello di sistema per formare unità funzionali. Questi twin offrono una panoramica sull'intera fabbrica o impianto, rendendo possibile testare le diverse configurazioni del sistema per raggiungere l'efficacia ottimale o rivelare nuove opportunità di business per l'introduzione di fonti di profitto aggiuntive. Un digital twin di sistema include tutti gli asset coinvolti in una specifica operazione (come la fornitura di energia) e fornisce dati in tempo reale utili per decisioni strategiche e in modo da avere una visione d'insieme del processo produttivo.

- **Digital Twin di Processi:** Si tratta del livello più alto nella gerarchia e collega i twin di sistema in un'unica entità. Offre una visione integrata di tutti i processi e i flussi di lavoro di un impianto o di una fabbrica. Permette inoltre di monitorare gli input (come la temperatura di produzione) e ottenere dati sugli output senza interrompere il processo di produzione e mettere a rischio la qualità della produzione stessa. In questo modo, i dirigenti possono testare diverse ipotesi, monitorare le metriche chiave e prendere decisioni informate migliorando l'efficienza e la qualità della produzione.

Secondo lo studio di Fortune Business Insights [12], i leader nello sviluppo di prodotti ritengono che i digital twin possano accelerare i processi di sviluppo, migliorare i risultati e ridurre i costi. Di conseguenza, le aziende stanno investendo rapidamente in questa tecnologia, come dimostrano le proiezioni di crescita del mercato globale.

Nel 2023, il settore dei digital twin è stato valutato a 12,91 miliardi di dollari, con una crescita stimata da 17,73 miliardi di dollari nel 2024 fino a raggiungere i 259,32 miliardi di dollari entro il 2032, registrando un tasso di crescita annuale composto (CAGR) del 39,8% nel periodo di previsione. Nel 2023, il Nord America ha detenuto la quota di mercato più elevata, pari al 34,55%.

L'espansione è guidata principalmente dalle applicazioni avanzate in ambito sanitario e dai progressi nelle tecnologie di simulazione e stampa 3D, che ne stanno ampliando le possibilità di utilizzo e l'adozione in diversi settori.

2.6.3 Nell'ambito AR

I digital twin si integrano in modo naturale con la realtà aumentata proprio perché sono una copia digitale di oggetti o sistemi e, tramite questa tecnologia, è possibile visualizzare e interagire con essi nel mondo reale. Tutto ciò apre numerose possibilità in molteplici settori applicativi, dalla navigazione urbana alla medicina, dal turismo all'intrattenimento.

Vediamo di seguito qualche esempio più nel dettaglio.

Navigazione Contestuale

Un esempio particolarmente interessante in questo ambito è l'evoluzione di Google Maps, che sta assumendo sempre più le caratteristiche di un digital twin dell'intero pianeta.

Tra le sue numerose funzionalità, possiamo citare: [8]

- Una mappa vettoriale bidimensionale con vari layer informativi;
- Una scansione tridimensionale fotogrammetrica di molte aree urbanizzate;
- Un rilievo fotografico a 360° delle strade, integrato con la mappa stradale (Street View).

Turismo

Il settore turistico rappresenta uno degli ambiti più promettenti per l'applicazione dei digital twin, grazie alla sua forte componente esperienziale e informativa.

Permettono di offrire soluzioni innovative per rispondere a diverse esigenze:

- Consultazione in anticipo di destinazioni e strutture ricettive;
- Conservazione e valorizzazione del patrimonio culturale;
- Gestione dei flussi turistici e mitigazione dell'impatto ambientale;
- Accessibilità a luoghi di interesse per persone con limitazioni fisiche o geografiche.

La loro versatilità è inoltre evidenziata dal fatto che operano su più dimensioni: consentono non solo di ricostruire il passato, monitorare il presente e simulare scenari futuri, ma anche di riprodurre fedelmente ambienti fisici e di integrare dati storici, ambientali e culturali. Questo consente la creazione di esperienze immersive che non solo migliorano la fruizione turistica tradizionale, ma aprono nuove modalità di interazione con il patrimonio culturale e naturale.

Di seguito sono illustrati alcuni esempi significativi:

- **Herrenberg (*Germania*):** Questo piccolo centro urbano ha sviluppato un digital twin per migliorare i processi decisionali urbani. Si tratta di uno strumento che permette di prevedere l'impatto delle future iniziative urbane sull'umore dei cittadini in modo da migliorare il processo decisionale. Inoltre, tramite visualizzazioni interattive, i residenti possono osservare in anteprima progetti in fase di definizione, come la nascita di un centro commerciale locale [56] [7].
- **Natural History Museum (*Londra*):** Si tratta di uno dei primi musei ad adottare tecnologia basata su digital twin. L'edificio ospita 15.000 sensori che misurano temperatura, umidità, vibrazioni e luce, generando dei dati che poi verranno usati per proteggere sia la struttura che le collezioni. Questo permetterà di unire informazioni, precedentemente isolate in dipartimenti, in un'unica interfaccia, consentendo al personale di identificare e risolvere rapidamente i problemi [34].
- **Grande Muraglia (*Cina*):** Questo è un esempio significativo di digital twin applicato al patrimonio culturale. Il sistema proposto si compone di tre elementi principali: una ricostruzione storica digitale, un monitoraggio in tempo reale tramite sensori IoT e simulazioni predittive per scenari futuri. Grazie alle tecnologie 3D e alle interfacce AR/VR, i visitatori possono esplorare il monumento attraverso diverse epoche storiche, ottenere informazioni contestuali durante la visita fisica e accedervi anche da remoto [22].

- **Gwanghwamun Square (Corea del Sud):** Ssin et al. [38] propongono un sistema per il turismo scientifico con diverse funzioni come raccomandazioni di contenuti, percorsi ottimali e gamification. L'architettura raccoglie dati da dispositivi mobili e sensori IoT e li integra in un digital twin per simulazioni e monitoraggio, offrendo esperienze turistiche personalizzate che valorizzano anche le attrazioni meno conosciute, contribuendo così all'economia locale.
- **Mirai Twin:** È una tecnologia che crea un digital twin degli hotel utilizzando riprese di droni. Offre ai potenziali ospiti un'esperienza immersiva prima del loro arrivo, permettendogli di esplorare ogni camera e angolo dell'edificio in modo da garantire una maggiore consapevolezza durante la fase di acquisto [27].
- **Amelia:** Si tratta della prima piattaforma italiana di intelligenza artificiale *as a service* dedicata al settore turistico. È stata presentata a Firenze nel 2023 e funziona come un digital twin delle strutture ricettive che può essere gestito in autonomia senza necessità di mantenere infrastrutture software in loco. Questo assistente virtuale agisce come un maggiordomo digitale che può essere allenato per supportare i clienti di hotel, i ristoranti o gli enti di promozione turistica [4].

Medicina

Nel settore sanitario, l'adozione dei digital twin sta aprendo diverse nuove prospettive per effettuare diagnosi, pianificazione terapeutica e monitoraggio dei pazienti.

Queste repliche virtuali, vanno a riprodurre organi e sistemi biologici, consentendo di analizzare il loro comportamento in tempo reale e di simulare scenari clinici complessi [11].

Tra i progetti particolarmente interessanti abbiamo:

- **Blue Brain Project:** Guidato dal professor Henry Markram, iniziato nel 2005 e durato fino alla fine del 2024, il suo scopo principale era quello di ottenere delle ricostruzioni e simulazioni digitali biologicamente accurate del cervello di un topo [6].
- **Sim & Cure:** Un progetto che permette di utilizzare modelli di simulazione basati sul paziente per prevedere il posizionamento ottimale di dispositivi medici per il trattamento degli aneurismi [37].
- **Philips HeartModel:** Questa tecnologia crea un digital twin del cuore a partire da immagini diagnostiche (come la tomografia computerizzata) ottenute prima dell'intervento. Il modello generico viene adattato ai dettagli anatomici di ogni paziente, permettendo ai medici di analizzare attentamente la struttura cardiaca [31].

- **Siemens Healthineers:** Questa tecnologia consente di effettuare diagnosi più rapide e accurate e di sperimentare diverse strategie terapeutiche in un ambiente virtuale prima dell'intervento reale. Inoltre, è stata presentata una demo di digital twin per il cuore che si sta piano piano espandendo nello sviluppo di modelli digitali di altri organi, con l'obiettivo di offrire strumenti avanzati per la medicina personalizzata e la pianificazione pre-operatoria [36].

Cinema

Nei film moderni, la creazione dei personaggi è spesso il risultato di una fusione tra dati ottenuti da attori reali e digital twin. Le informazioni, come i movimenti catturati tramite motion capture, fungono da base per l'animazione, mentre gli artisti degli effetti visivi costruiscono un modello digitale che controlla l'immagine renderizzata.

Questo approccio offre numerosi vantaggi: permette innanzitutto agli attori di eseguire azioni altrimenti impossibili e, allo stesso tempo, consente loro di concentrarsi sulla resa emotiva, offrendo a registi e montatori un maggiore controllo in fase di post-produzione. Possiamo considerarlo come un ponte tra la performance umana e l'arte digitale, in cui l'espressività dell'attore rimane intatta, mentre l'aspetto visivo viene trasformato attraverso la computer grafica.

Dando uno sguardo ai dati illustrati in Figura 2.5 si nota che l'impiego dei digital twin nel 2017 nell'ambito della produzione cinematografica e televisiva è passato da circa il 4% al 35%. Nel 2018, questa percentuale è ulteriormente aumentata dal 17% al 58% [23].

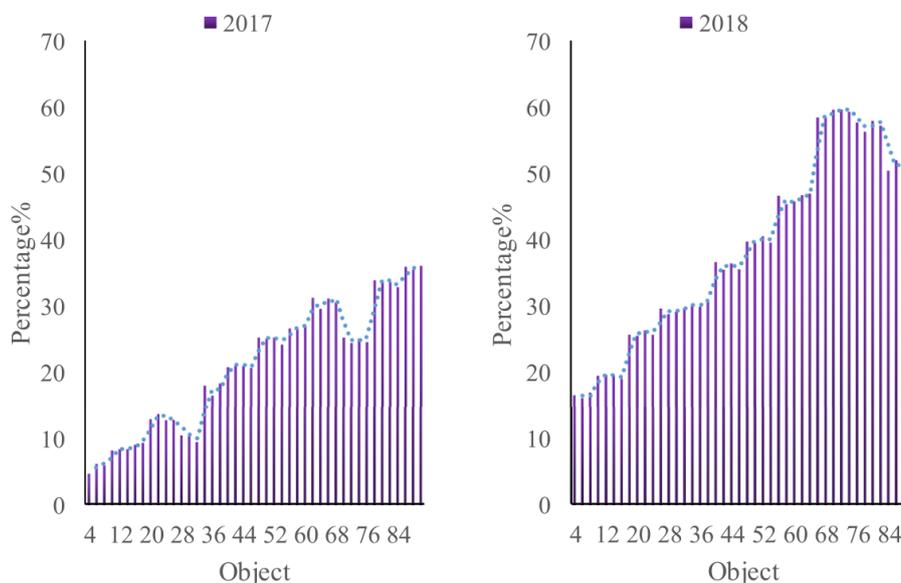


Figura 2.5: Applicazione dei digital twin nel cinema 2017 vs 2018 *Source:* [23]

Tra gli esempi più importanti di questo settore possiamo citare:

- **Walt Disney Studios' StudioLAB:** Realizzata in collaborazione con Accenture, è una tecnologia che rende possibile generare modelli 3D di set e ambientazioni, in modo da valutare in anticipo la configurazione delle riprese e ottimizzare il location scouting. Questo approccio non solo riduce tempi e costi, ma migliora anche la precisione nella pianificazione delle scene, consentendo decisioni più rapide e informate durante la fase pre-produzione [1].
- **Joker (2019):** Tramite LiDAR sono stati scannerizzati l'appartamento di Arthur e il set del talk show di Murray Franklin. Questo ha permesso al team VFX di aggiungere sangue digitale in post-produzione con una perfetta corrispondenza alla location fisica [17].
- **John Wick 3 - Parabellum (2019):** Anche qui è stato usato LiDAR per digitalizzare l'edificio One Hanson Place a Brooklyn, creando un digital twin per la sequenza della sparatoria al Continental [17].
- **Imperium: Augustus (2003):** L'azienda italiana *Carraro Lab* ha applicato il digital twin di Roma antica negli effetti speciali di questa serie TV [8].

Capitolo 3

Stato dell'arte

Questo capitolo fornisce una panoramica delle tecnologie e delle applicazioni più rilevanti che sono attualmente mercato o in fase di sviluppo, sia in ambito accademico che commerciale.

Analizzare le soluzioni esistenti è fondamentale per due motivi: da un lato, permette di comprendere le best practices e le tecnologie predominanti nel settore; dall'altro, aiuta a individuare le principali limitazioni di quelle già consolidate. Questo approccio consente di contestualizzare il contributo del sistema proposto, mettendone in evidenza le caratteristiche distintive e i vantaggi rispetto alle alternative già presenti.

L'analisi è strutturata in tre sezioni principali: la prima esamina le soluzioni AR per la navigazione e il turismo, la seconda si concentra sull'integrazione di assistenti virtuali basati su modelli di linguaggio avanzati, mentre l'ultima evidenzia le innovazioni e i benefici specifici dell'approccio adottato in questa ricerca.

3.1 Soluzioni AR per la Navigazione e il Turismo

- **Navigine AR Indoor Navigation:** Si tratta di una soluzione commerciale che sfrutta la realtà aumentata per consentire la navigazione all'interno di ambienti chiusi. Questa tecnologia permette agli utenti di accedere tramite un codice QR a mappe interattive in AR specifiche per ciascuna location e di seguire indicazioni per raggiungere i punti di interesse. Il sistema integra diverse tecnologie di posizionamento come UWB (Ultra-Wideband), Wi-Fi RTT (Round-trip time), BLE (Bluetooth Low Energy) e algoritmi avanzati di navigazione, garantendo una buona accuratezza nel determinare la posizione dell'utente anche in spazi architettonicamente complessi o affollati.

I gestori delle strutture possono sviluppare mappe digitali per i propri spazi che vengono integrate con le funzionalità di Navigine, permettendo non solo di crea-

re percorsi personalizzati, ma anche di aggiornare le informazioni per adattarsi a modifiche temporanee o permanenti.

- **Timelooper Xplore:** Un'altra applicazione commerciale che sfrutta la realtà virtuale e aumentata per offrire esperienze immersive a tema storico e culturale. La piattaforma consente agli utenti di "viaggiare nel tempo", ricreando eventi storici e ambienti del passato direttamente nel luogo in cui si trovano. Utilizzando ricostruzioni 3D, narrazioni audiovisive e effetti interattivi, la tipica visita museale viene trasformata in un'esperienza immersiva e educativa. Questo sistema viene impiegato in contesti molto diversi, dal turismo culturale alla didattica nei musei e nei siti archeologici [42].
- **ARBIN: Augmented Reality Based Indoor Navigation System:** In questa ricerca viene presentato un sistema di navigazione *indoor* (per ambienti chiusi) che sfrutta la realtà aumentata per guidare gli utenti che si trovano in grandi edifici come ospedali, aeroporti e stazioni ferroviarie. Le indicazioni sono mostrate in AR e utilizza beacon Bluetooth a basso consumo energetico (chiamati *Lbeacon*) posizionati in punti strategici come incroci e punti di interesse. Quando l'utente si avvicina a un beacon, il sistema mostra frecce direzionali che lo guidano verso la destinazione successiva. ARBIN è stato testato nell'ospedale *National Taiwan University Hospital YunLin Branch*, coprendo un'area di circa 1800 m² con 35 destinazioni diverse; è stata raggiunta una precisione media di circa 3-5 metri. Si basa su Google ARCore per la creazione e il posizionamento dei modelli 3D, sensori IMU per determinare l'orientamento dello smartphone e implementa l'algoritmo di Dijkstra per calcolare il percorso più breve verso la destinazione [21].
- **Indoor Navigation System using Augmented Reality:** Questo lavoro di ricerca presenta la progettazione e l'implementazione di un sistema per la navigazione indoor per il campus universitario *Xavier Institute of Engineering*. Utilizza varie tecnologie tra cui:
 - **Sensori LiDAR:** Per la creazione di mappe 3D ad alta risoluzione degli ambienti interni.
 - **Codici QR:** Per indicare punti di riferimento posizionati strategicamente in tutto il campus. Gli smartphone degli utenti scansionano questi codici e raccolgono dati LiDAR in tempo reale, consentendo al sistema di individuare con precisione la loro posizione.

L'interfaccia fornisce indicazioni passo-passo, ottimizzando i percorsi in base alle informazioni raccolte [30].

3.2 Soluzioni AR con Assistenti Virtuali e LLM

L'integrazione di assistenti virtuali e modelli di linguaggio avanzati come gli LLM nelle applicazioni di realtà aumentata rappresenta una scelta innovativa per personalizzare ulteriormente l'esperienza turistica. Grazie a questi assistenti è possibile dare risposte accurate a domande complesse, fornire raccomandazioni e migliorare l'interazione.

Vediamo alcuni esempi:

- **AR Travel Concierges:** Questa applicazione, sviluppata da Designium, è un esempio con obiettivi molto simili a quelli che il sistema descritto in questa tesi vuole raggiungere.

Il progetto va a creare un assistente di viaggio capace di fungere da guida turistica, consigliando punti di interesse vicini e dando informazioni contestuali.

Dal punto di vista tecnologico si basa su diverse tecnologie: [9]

- **Modello linguistico GPT-4:** Utilizzato per consentire all'assistente AI di rispondere alle richieste degli utenti e fornire descrizioni dettagliate dei luoghi visitati.
 - **Sistema VPS (Visual Positioning System):** In modo da riconoscere con precisione la posizione dell'utente e generare itinerari turistici personalizzati.
 - **Sistema di navigazione in AR:** Permette di fornire indicazioni visive sovrapposte all'ambiente reale.
 - **API Google Geospatial e Google Maps:** Così da gestire le funzionalità di navigazione AR.
- **VirtuWander:** È un sistema il cui obiettivo è quello di migliorare l'esperienza dei tour virtuali attraverso l'uso di LLM; anche se è stato progettato principalmente per musei virtuali, rappresenta un'applicazione simile a quella presente qui trattata.
Si basa su un approccio a due fasi: prima il sistema trasforma le richieste degli utenti in contesti di guida personalizzati, poi offre feedback attraverso alcune modalità, tra cui: [50]
 - **Voce:** Narrazione audio per un'esperienza più coinvolgente;
 - **Avatar:** Una guida virtuale per l'interazione visiva;
 - **Testo:** Finestre informative contestuali;
 - **Elementi visivi aggiuntivi:** Come evidenziazioni, schermi virtuali o mini-mappe per migliorare la consapevolezza spaziale.
 - **Live-Guided Tour:** Questo studio sperimentale si concentra sulla creazione di un digital twin tramite fotogrammetria digitale di un frantoio sotterraneo a Gallipoli

(Italia). L'obiettivo è integrare questo modello in un sistema che permette la creazione di tour virtuali guidati da remoto. Questa soluzione risulta particolarmente utile sia in situazioni di emergenza, come è stato il periodo COVID-19, sia per favorire l'accessibilità a persone con disabilità.

La piattaforma offre inoltre numerose funzioni quali:[14]

- Possibilità di visita multi-utente sincronizzata;
- Visualizzazione stereoscopica per aumentare l'immersione;
- Compatibilità multi-piattaforma (desktop e mobile);
- Funzionalità di e-learning integrate per valutare la comprensione dei contenuti.

3.3 Innovazioni e Vantaggi della Soluzione Proposta

L'analisi dello stato dell'arte evidenzia come soluzioni simili a quella proposta in questa tesi esistano già, sebbene siano poco diffuse e prevalentemente limitate all'ambito accademico. Inoltre, queste applicazioni risultano spesso vincolate a singole istituzioni culturali, come musei e complessi storico-artistici, o a settori commerciali specifici, quali l'ospitalità alberghiera e il retail specializzato.

Sebbene efficaci nei rispettivi contesti, tali implementazioni presentano limitazioni in termini di scalabilità e riutilizzabilità. Al contrario, l'applicazione proposta mira a diventare una piattaforma espandibile, progettata per integrare progressivamente un numero sempre maggiore di luoghi da esplorare tramite aggiornamenti futuri. Tutto ciò è reso possibile dalla natura modulare dell'architettura del sistema, che facilita l'adattabilità e l'evoluzione della piattaforma nel tempo.

Un ulteriore elemento distintivo è la stretta integrazione con la piattaforma Matterport per l'acquisizione e la gestione dei digital twin. Matterport rappresenta uno standard consolidato nella digitalizzazione di ambienti reali attraverso la scansione 3D e questo garantisce significativi vantaggi in termini di scalabilità e sostenibilità, consentendo l'espansione continua del catalogo di location disponibili attraverso processi standardizzati e ottimizzati e, così facendo, i tempi di produzione e le risorse necessarie per integrare nuovi ambienti nell'ecosistema applicativo vengono drasticamente ridotti.

Un aspetto chiave del progetto riguarda l'accuratezza e l'affidabilità dei punti di interesse. A differenza di soluzioni come *AR Travel Concierges*, che si basano su contenuti generati automaticamente dall'intelligenza artificiale, il sistema proposto prevede che tutti i punti di interesse vengano inseriti manualmente da un team dedicato tramite l'interfaccia di Matterport. Questo approccio fa sì che le informazioni siano precise, contestualmente rilevanti e curate nei dettagli: l'IA viene utilizzata esclusivamente per fornire approfondimenti sui punti di interesse già definiti, evitando così errori o incongruenze tipici dei sistemi completamente automatizzati.

Capitolo 4

Implementazione e Sviluppo

4.1 Architettura dell'Applicazione

4.1.1 Introduzione

Di seguito si analizzeranno nel dettaglio gli elementi e le scelte che hanno guidato l'implementazione dell'applicazione. Si inizierà con una panoramica generale del funzionamento dei componenti principali, per poi approfondire progressivamente gli aspetti più tecnici nelle sezioni successive.

Architettura Manager-based

È fondamentale capire cosa sono i **Manager** e quale ruolo fondamentale ricoprono. Questi `GameObject` sono la base di un'architettura molto comune tra i progetti Unity che prende il nome di *architettura Manager-based*.

Essi utilizzano un pattern specifico detto *Pattern Singleton* che assicura l'esistenza di un'unica loro istanza (pubblica e statica) durante tutta l'esecuzione dell'applicazione evitando duplicazioni. Ogni volta che avviene un cambio di Scena (ad esempio si passa dalla scena del menu principale alla scena del tour guidato) tutti i `GameObject` presenti in quella precedente vengono distrutti, e quindi persi. I Manager devono persistere tra le Scene poiché ad essi sono solitamente affidati dei compiti fondamentali.

Tra gli esempi tipici di `GameObject` che vengono strutturati come dei Manager possiamo trovare:

- **AudioManager**: Si occupa di riprodurre suoni o tracce musicali.
- **UIManager**: Contiene e gestisce la UI dell'applicazione (punteggi, menu, dialoghi e altro).

- **GameManager:** Per i videogiochi si tratta di un componente cruciale che è sempre presente e gestisce il gameplay e gli stati principali come la vittoria, il game over ecc. Può inoltre tenere traccia dei punteggi, dei progressi, e altro.
- **InputManager:** Gestisce gli input dell'utente (tastiera, controller, mobile).

Solitamente una classe che utilizza il pattern Singleton è così definita:

```
public class GameManager : MonoBehaviour {

    public static GameManager instance;
    public int score = 0;

    private void Awake() {

        if(instance != null) {
            Destroy(this.gameObject);
            return;
        }
        instance = this;
    }
    // ...
}
```

Listing 4.1: Esempio di classe che usa il pattern Singleton

Master Manager

Nel progetto è stato adottato un approccio centralizzato per la gestione dei vari Manager, implementando un unico componente denominato per convenienza *MasterManager*. Questo funge da punto di accesso unificato, mantenendo al suo interno le istanze di tutti i Manager e semplificando la comunicazione tra gli oggetti della Scena.

Quando un GameObject necessita di un Manager specifico, può interfacciarsi direttamente con il MasterManager, lui gli fornirà l'istanza appropriata, favorendo un'architettura modulare e scalabile.

Per ottimizzare la fase di prototipazione questo oggetto attualmente contiene soltanto il database (locale) dei luoghi disponibili nell'app. In questo modo, qualsiasi componente che necessiti di queste informazioni può accedervi in modo centralizzato ed efficiente.

Vediamone il funzionamento nel dettaglio:

```
public class MasterManager : MonoBehaviour {

    public static MasterManager instance { get; private set; }
    public LocationDatabase locationDatabase { get; private set; }

    private void Awake() {
```

```

        if(instance != null) {
            Destroy(this.gameObject);
            return;
        }
        instance = this;

        locationDatabase = GetComponentInChildren<LocationDatabase>();
    }
}

```

Listing 4.2: Classe MasterManager

4.1.2 Workflow Applicativo

Come già detto, lo scopo di questa applicazione è quello di guidare l'utente verso dei punti di interesse (POI) come farebbe una guida turistica fornendo informazioni contestuali. Per ottenere tutto ciò, il progetto è stato organizzato in più Scene e, principalmente, possiamo distinguerne due tipologie:

- **Menu principale:** È il punto d'ingresso dell'applicazione, qui l'utente può scegliere il luogo da cui far partire tour e modificare le opzioni.
- **Scene delle Location:** Ogni location è rappresentata da una Scena dedicata. Al suo interno sono contenuti diversi GameObject come il modello 3D in scala 1:1, i Point of Interest (POI), gli Starting Point (SP), il sistema di navigazione AR con pathfinding e la UI.

Per comprendere al meglio come sarà l'esperienza dell'utente finale, è importante analizzare il flusso completo di utilizzo dell'applicazione dall'avvio al tour guidato. Andremo ora a trattare nel dettaglio le varie fasi principali.

Avvio e Selezione Location

Quando l'utente avvia l'applicazione, sarà lanciata la Scena del menu principale. Qui sono presenti diverse voci tra cui selezionare:

- **Select location:** Consente di selezionare il luogo in cui si trova attualmente l'utente per poi far partire il tour virtuale.
- **Options:** Permette di configurare eventuali preferenze, come se abilitare i joystick virtuali oppure il Text-to-Speech.
- **Info:** Mostra una schermata con la versione dell'app e i crediti.
- **Quit:** Termina l'applicativo.

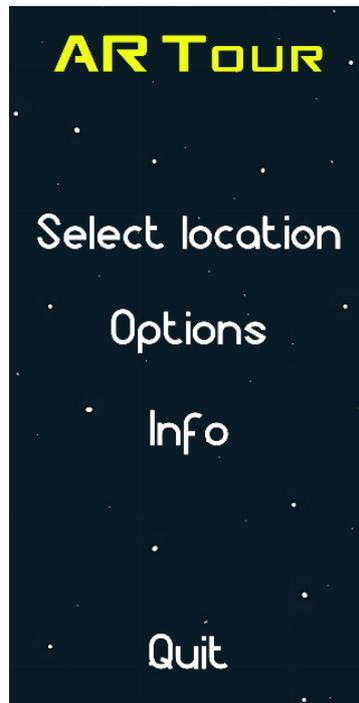


Figura 4.1: Menu principale dell'applicazione

Inizializzazione nella Location

Il processo di inizializzazione comincia con il caricamento degli Starting Point (SP) dall'apposito file e la loro istanziazione all'interno della Scena.

Successivamente, l'utente può avviare il tour scansionando il QR code. Questo passaggio è cruciale poiché consente al sistema di allineare la posizione reale dell'utente con quella del mondo virtuale, trasportando la sua controparte digitale alla posizione dello Starting Point presente nella Scena di Unity. Così facendo, viene garantita la corrispondenza biunivoca tra spazio fisico e virtuale.

Dopo aver fatto ciò, il sistema carica dai file i vari punti di interesse (POI), sbloccando la possibilità di selezionare e raggiungere le destinazioni disponibili.

Navigazione e Interazione

Appena scansionato il QR code, l'utente può selezionare il punto di interesse che vuole raggiungere attraverso un apposito menu. Il sistema procede istanziano un *NPC* (*Non-Player Character*), ovvero un GameObject sferico che si muove nello spazio, lasciando dietro di sé una scia rossa che, se seguita, consentirà di percorrere la strada più veloce verso il POI. Per aiutare ulteriormente la navigazione, l'interfaccia si aggiorna mostrando

una freccia rossa che punta costantemente verso la destinazione, insieme a informazioni utili come la distanza in metri.



Figura 4.2: Schermata di navigazione con NPC

Esplorazione dei POI

Quando l'utente raggiunge un punto di interesse, potrà inquadrarlo con la fotocamera del dispositivo per accedere a un menu avente diverse opzioni:

1. **Descrizione base:** Viene subito mostrata una descrizione statica che è caricata direttamente dal file dove sono memorizzati i vari POI.
2. **Ricerca online:** Per chi desidera saperne di più, è possibile effettuare una ricerca online che aprirà automaticamente una pagina Google sul punto scelto.
3. **Domanda all'assistente LLM:** L'utente può chiedere informazioni direttamente all'assistente virtuale. Per una maggiore accessibilità, ogni risposta verrà letta ad alta voce grazie al sistema Text-to Speech.



Figura 4.3: Schermata visualizzata una volta aver inquadrato il POI

4.1.3 Script Principali

L'applicazione fa uso di numerosi script per funzionare; di seguito verranno descritti soltanto quelli principali:

- **NavigationManager;**
- **LoaderManager;**
- **QRScannerManager;**
- **LLMChatManager;**
- **UIController;**
- **ARPlayerController;**
- **TextToSpeech.**

NavigationManager

Il NavigationManager è il componente che si occupa di gestire la navigazione dell'utente. La sua funzione principale è guidarlo verso i punti di interesse utilizzando tre oggetti:

1. **NPC:** Già spiegato, conduce l'utente al punto di interesse seguendo il percorso più breve.
2. **Indicatore di direzione:** Freccia rossa che appare sulla parte sinistra in fondo allo schermo. Viene continuamente aggiornata per puntare verso la destinazione scelta.
3. **Informazioni sulla distanza:** È un testo continuamente aggiornato che mostra la distanza rimanente verso il POI in metri.

Il sistema di pathfinding degli NPC è basato sul nuovo package di Unity chiamato *AI Navigation* che permette di creare delle NavMesh personalizzate con pochi click.

```
// Gestione dello stato principale nel metodo Update
if (currentPOI != null && isTourStarted) {
    if (!uiController.isDialogueOpen) {
        // Attiva e aggiorna le frecce e la distanza
        ActivatePOIArrowInfoContainer();
        UpdatePOIArrow();
        ActivatePOIDistanceText();
        UpdateDistanceText();
    } else {
        // Disattiva elementi di navigazione durante i dialoghi
        DeactivatePOIArrowInfoContainer();
        DeactivatePOIDistanceText();
    }
}
```

Listing 4.3: Classe NavigationManager

LoaderManager

Il LoaderManager si occupa principalmente di inizializzare e gestire il caricamento delle risorse necessarie all'inizializzazione della Scena.

Il suo funzionamento si articola in:

1. **Caricamento dati:** Esegue il parsing dei file che contengono informazioni su POI e SP.
2. **Istanziamento oggetti:** Vengono istanziati i prefab relativi ai POI e ai vari SP nella Scena.

3. Gestione preferenze: Permette l'applicazione delle opzioni.

```
private void InstantiatePointsOfInterest() {
    for (int i = 0; i < pointsOfInterest.Count; i++) {

        // Istanziio il prefab
        GameObject newPOI = Instantiate(pointOfInterestPrefab,
            pointsOfInterest[i].position, Quaternion.identity);
        newPOI.name = "POI - " + pointsOfInterest[i].label;

        // Ci cambio il nome mostrato dalla bubble
        ChangeWorldBubbleDisplayName(newPOI, i, true);

        // Per leggere le info sul POI
        PointOfInterestComponent poiComponent = newPOI.GetComponent<
            PointOfInterestComponent>();
        if (poiComponent != null) {
            poiComponent.SetPointOfInterest(pointsOfInterest[i]);
        }
    }
}
```

Listing 4.4: Metodo per l'istanziamento dei POI

Particolarmente importante è la struttura dei file relativi ai Point of Interest (POI) e agli Starting Point (SP). Ognuno di essi è formattato in modo coerente e funzionale:

- Il file dei POI è in formato *.csv* ed è creato direttamente da Matterport. Per adattarlo è stato necessario codificare un layer di interpretazione in modo da estrarne i dati. Tra le informazioni principali abbiamo l'identificativo univoco del punto di interesse, il nome, la descrizione dettagliata, la posizione digitale nella quale verrà mostrato e altro.
- Il file degli Starting Point, invece, è di tipo *.txt* ed è stato progettato da zero per essere facile da modificare e gestire. All'interno sono memorizzate informazioni quali l'ID, il nome e le coordinate che ne descrivono la posizione dentro alla Scena.

QRScannerManager

Il QRScannerManager gestisce la scansione dei codici QR presenti negli Starting Point. Per fare il parsing dei dati letti è stato scelto *ZXing.Net*, un porting di *ZXing*, ovvero una libreria open-source per l'elaborazione di immagini di codici a barre 1D/2D multiformato, originariamente implementata in Java.

Il comportamento di questo script è diviso in quattro fasi:

1. **Acquisizione e decodifica:** A intervalli regolari, la fotocamera del dispositivo cattura un frame, che viene analizzato per rilevare la presenza di codici QR. Se ne viene rilevato uno, il sistema procede alla sua decodifica.
2. **Parsing dei dati:** Sono estratte le informazioni sugli Starting Point dal contenuto del QR.
3. **Validazione:** Si verifica se lo Starting Point è presente nel database all'interno dell'applicazione.
4. **Posizionamento utente:** Viene spostato l'utente virtuale nella posizione corrispondente allo Starting Point, creando così la biunivocità tra realtà e digital twin.

```
private void Update() {
    // Condizioni iniziali
    if (!uiController.isDialogueOpen &&
        (Time.frameCount % scanQRCodeEveryXFrame) == 0 &&
        !isProcessing) {

        // Acquisizione e processamento immagine
        if (playerCamera.TryAcquireLatestCpuImage(out currentCPUImage))
        {
            isProcessing = true;
            StartCoroutine(ProcessImage(currentCPUImage));
            currentCPUImage.Dispose();
        }
    }
}
```

Listing 4.5: Metodo per l'acquisizione dei QR code

LLMChatManager

Questo script implementa l'interazione con il modello linguistico Gemini di Google e permette di ottenere informazioni dettagliate sui POI attraverso un'interfaccia simile a quella di una chat.

Le sue funzionalità principali sono:

1. **Comunicazione con Gemini:** Gestisce le richieste API e la trasmissione dati col modello. La API key viene cifrata per garantire un ulteriore layer di sicurezza.
2. **Interfaccia di chat:** Visualizza la cronologia dei messaggi scritti dall'utente e le risposte dell'IA.

3. Integrazione TTS: Converti le risposte testuali in audio.

È stato anche incluso un sistema di prompt engineering che aiuta il modello a fornire risposte pertinenti e contestualizzate.

```
private IEnumerator SendToGemini(string userMessage) {
    // Costruisce il messaggio con contesto
    string contextMessage = basePrompt;
    if (currentPOI != null) {
        contextMessage += $"
You are currently providing information
about the Point of Interest named \"{currentPOI.label}\"
located in {location}.";
    }

    // Messaggio completo
    string fullMessage = $"
{contextMessage}
User's message: {
    userMessage}";

    // Conversione in JSON per l'API...

    // Esecuzione della richiesta HTTP...
}
```

Listing 4.6: Metodo per l'invio di messaggi a Gemini

ARPlayerController

Questo script gestisce il movimento e l'interazione dell'utente nell'ambiente aumentato.

1. **Rilevamento POI:** È necessario identificare quando l'utente sta effettivamente guardando un punto di interesse.
2. **Visualizzazione informazioni:** In caso affermativo, attiva i dialoghi informativi.
3. **Movimento nell'ambiente tramite joystick a schermo:** Permette di muovere l'utente nella Scena tramite un joystick apposito. Questa funzionalità è stata implementata per debug e testing, in quanto su PC non è possibile muoversi senza l'utilizzo di un dispositivo AR.

La sua funzionalità principale risiede proprio nel rilevamento dei punti di interesse. Quando ne viene inquadrato uno, compare a schermo un apposito menu che mostra informazioni contestuali. Per ottenere questo comportamento viene utilizzato un *raycast* ovvero un "raggio invisibile" che parte dalla fotocamera del giocatore e, se interseca un POI, viene interpretato come se l'utente stesse effettivamente osservando quel punto specifico. Così facendo, questi dialoghi sono attivati automaticamente senza il bisogno di marker o tecnologie simili.

```

private void DetectPOI() {

    // Crea un raggio dalla camera dell'utente
    Vector3 rayOrigin = playerCamera.transform.position;
    Vector3 rayDirection = playerCamera.transform.forward;
    Ray ray = new Ray(rayOrigin, rayDirection);
    RaycastHit hit;

    // Verifica se il raggio colpisce un POI
    if (Physics.Raycast(ray, out hit, detectionDistance, poiLayer)) {

        // Ottengo le informazioni del POI...

        // Mostro il menu del POI
        ShowPOIInfoDialogue();
    }

    // ...
}

```

Listing 4.7: Metodo per il rilevamento di un POI

UIController

Questo componente garantisce il funzionamento dell'intera interfaccia utente, coordinando l'apertura e la chiusura dei dialoghi e menu.

1. **Gestione dei dialoghi:** Controlla l'apertura e la chiusura di tutti i pannelli e le finestre dell'applicazione. È anche stato implementato un sistema di gestione degli stati che impedisce la sovrapposizione di più dialoghi.
2. **Preferenze utente:** Controlla le varie impostazioni come l'abilitazione dei joystick e il sistema Text-to-Speech (TTS).

```

public void ShowPOISelector() {

    // Se non ci sono altri dialoghi aperti
    if(!isDialogueOpen) {
        poiSelectorDialogue.SetActive(true);
        isDialogueOpen = true;
        // Nascondo il joystick virtuale
        if(loaderManager.isJoystickEnabled) SetJoysticksActive(false);
    }
}

```

Listing 4.8: Metodo l'apertura di un menu

4.2 Dettagli Tecnici

In questa sezione si definiranno approfonditamente i sistemi più particolari dell'applicazione, in modo da capirne al meglio il funzionamento.

4.2.1 Organizzazione dei Dati

L'architettura dei dati comprende diversi tipi di entità che interagiscono tra loro. In particolare, si possono identificare due elementi principali: i Points of Interest (POI) e gli Starting Points (SP).

Queste informazioni vengono poi organizzate in database locali che ne facilitano l'accesso e la gestione.

Point of Interest (POI)

Questo elemento rappresenta sia un punto di interesse nell'ambiente circostante che un punto specifico collocato al di sopra del digital twin, ovvero al di sopra del modello 3D in scala 1:1.

È stato implementato come un'entità che racchiude un insieme di metadati e informazioni quali:

- Un identificativo univoco che permette di distinguere il POI all'interno del sistema;
- Un nome;
- Una descrizione dettagliata che offre informazioni aggiuntive. Queste verranno inserite dalle persone addette alle scansioni dei luoghi tramite l'interfaccia di Matterport;
- Le coordinate tridimensionali (x, y, z) che ne definiscono la posizione esatta.

Starting Point (SP)

Lo Starting Point rappresenta invece il punto di partenza verso cui l'utente dovrà recarsi per iniziare il tour e, come già accennato, permette di stabilire la corrispondenza tra il mondo reale e virtuale.

Idealmente, lo Starting Point sarà fisicamente presente nel luogo in cui si trova l'utente e avrà la forma di un piedistallo con un codice QR da scansionare. Una volta averlo inquadrato, l'applicazione permetterà di selezionare il punto di interesse al quale si vuole essere guidati.

Uno Starting Point è caratterizzato da:

- Un identificativo univoco che ne permette il riconoscimento nel sistema;

- Un nome;
- Una posizione (x, y, z) che corrisponde esattamente alla sua collocazione sul digital twin.



Figura 4.4: Interpretazione artistica di un futuro Starting Point fisico

Database Locali

I dati utilizzati dall'applicazione sono stati organizzati in tre database locali rappresentati da tre GameObject distinti. Ognuno contiene informazioni importanti che possono essere consultate in ogni istante dagli altri sistemi:

- **LocationDatabase:** Mantiene le informazioni sulle location disponibili, inclusi identificatori, nomi, descrizioni e tipo (indoor/outdoor). Queste non sono lette da un file, ma inserite direttamente nel GameObject.
- **POIDatabase:** Gestisce la collezione di Point of Interest. Questo oggetto verrà popolato andando a leggere il file .csv dopo che l'utente scannerizza il codice QR.
- **SPDatabase:** Contiene tutti gli Starting Point. A differenza del precedente, è popolato andando a leggere il corrispettivo file .txt appena viene avviata la Scena.

4.2.2 Interpretazione Dati

Il sistema deve essere in grado di processare correttamente i dati ottenuti da piattaforme esterne. Il modello 3D e i file dei punti di interesse sono presi direttamente da Matterport e per questo di seguito sono spiegate le metodologie di conversione che hanno permesso l'interpretazione di questi dati.

Dati di Matterport

Il processo di conversione si articola in due fasi principali:

1. **Gestione e conversione del modello 3D;**
2. **Elaborazione ed interpretazione dei dati dei punti di interesse.**

Partiamo analizzando la prima fase: Unity supporta specifici formati di file per l'importazione di modelli 3D, tra cui: [41]

- .fbx
- .dae (Collada)
- .dxf
- .obj.

Il file che rappresenta la location è originariamente dato in formato **.obj (Wavefront Object)** che è un formato relativamente semplice, progettato per la rappresentazione della geometria 3D. Al suo interno vengono memorizzate poche informazioni tra cui: [55]

- Dati sulle mesh (vertici, spigoli, facce);
- Normali e coordinate UV;
- Informazioni base sulla posizione, rotazione e scala degli oggetti;
- Informazioni basilari sui materiali (attraverso un file .mtl separato).

La scelta dell'utilizzo di questo tipo di file è stata probabilmente dettata dal fatto che è universalmente supportato e ampiamente compatibile con la maggior parte delle applicazioni e software di modellazione 3D.

Tuttavia, per l'integrazione con Unity è preferito il formato **.fbx (Filmbox)**, poiché viene utilizzato come parte della catena di importazione ed è quindi considerata una best practice usarlo ogni volta che è possibile [41].

Inoltre, rispetto al formato .obj, il .fbx offre diversi vantaggi:

- Supporto completo per scene 3D complesse, includendo camera, mesh, luci e altre informazioni;
- Mantenimento delle gerarchie degli oggetti e delle loro relazioni spaziali;
- Gestione più accurata dei materiali e delle texture;

Per convertire da un formato all'altro è stato utilizzato il software gratuito *Blender*.

Prendiamo ora in considerazione la seconda fase, ovvero l'elaborazione ed interpretazione dei dati dei punti di interesse. Matterport fornisce le informazioni relative ai punti di interesse in un file .csv, che contiene dettagli su ciascun POI all'interno della location. Per processare questi dati è stato implementato un layer software dedicato, che si occupa di:

- Leggere e parsare il file CSV;
- Estrarre le informazioni rilevanti come identificativi, etichette, descrizioni e coordinate spaziali;
- Convertire i dati in un formato utilizzabile dall'applicazione.

Per la fase di parsing del file, si è scelto un package esterno ed open-source chiamato *CSV-Parser* di *Yuto Fushimi* [13]. La motivazione è derivata dalla complessità dei contenuti e dalla quantità di lavoro necessaria per implementare un parser personalizzato da zero. È comunque stato adattato per essere compatibile con la struttura degli oggetti presenti nel progetto.

Una volta estratti i dati necessari, vengono generati dinamicamente dei *GameObject* di tipo *PointOfInterest*, istanziando poi i corrispondenti *Prefab* all'interno della scena.

```
public static List<CsvData> ReadPOIFile(string data) {
    var parsedData = CSVParser.LoadFromString(data, Delimiter.Semicolon);
    var poiList = new List<CsvData>();
    foreach (var row in parsedData.Skip(1)) {
        var poi = new CsvData {
            Id = row.ElementAtOrDefault(0),
            LabelIt = CleanHtmlContent(row.ElementAtOrDefault(1)), //
                Pulisco i campi HTML
            // ...
        };
        poiList.Add(poi);
    }
    return poiList;
}
```

Listing 4.9: Metodo per lettura/conversione file .csv

QR Code

Ogni codice QR dovrà essere generato in un modo specifico per poter essere riconosciuto. Al suo interno è memorizzata una stringa di testo codificata così:

STARTING_POINT/ID:<id>/LABEL:<nome>/POSITION:(x,y,z)

Nel caso in cui non venga rilevato questo formato, il QR sarà scartato mostrando un messaggio d'errore all'utente.

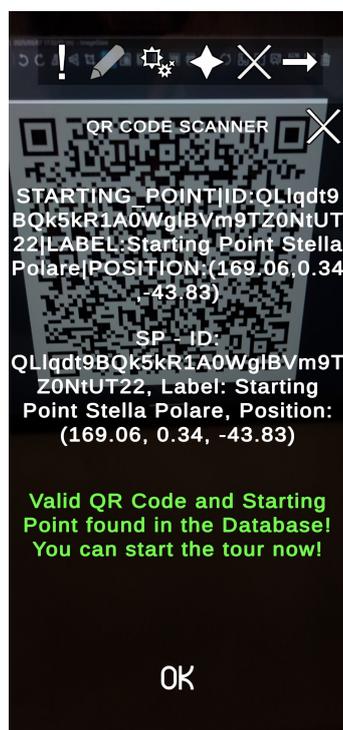


Figura 4.5: QR Code riconosciuto con successo



Figura 4.6: QR Code non valido

4.2.3 Gestione Trasparenza

Durante lo sviluppo dell'applicazione, è emerso un particolare problema legato alla gestione della visualizzazione del digital twin e degli NPC.

Poiché il digital twin è una replica esatta in scala 1:1 della location reale, è necessario che questo modello 3D rimanga invisibile all'utente durante l'utilizzo dell'applicazione. Verrà utilizzato esclusivamente a livello di sistema per creare il pathfinding, permettendo agli NPC di muoversi verso il POI scelto dall'utente.

La problematica riguarda proprio l'interazione tra NPC e l'ambiente virtuale: l'NPC

esiste soltanto nel mondo virtuale ma deve comportarsi in modo coerente rispetto agli ostacoli presenti nel mondo reale. Quando ad esempio si troverà dietro a una parete del modello 3D è importante che venga nascosto in quanto anche nel mondo reale sarà presente.

Per risolvere, però, non è possibile semplicemente "disattivare" il GameObject del digital twin, poiché questo approccio renderebbe l'NPC sempre visibile, anche quando non dovrebbe esserlo. È stato quindi necessario implementare uno Shader che viene applicato a tutti i materiali del modello 3D.

Questo Shader consente di rendere il modello completamente invisibile, mantenendolo comunque attivo nella scena.

- *Blend Zero One*: Moltiplica il contributo visivo dell'oggetto per zero, eliminandolo completamente dalla visualizzazione.
- *Queue=Geometry-1*: Lo posiziona nella coda di rendering leggermente prima degli oggetti geometrici standard.

Grazie a questa soluzione, quando il modello è reso invisibile, l'NPC può muoversi liberamente risultando opportunamente nascosto anche quando si trova dietro a ostacoli o pareti, poiché questi elementi, seppur invisibili, continuano a influenzare il rendering degli altri oggetti nella Scena.

```
Shader "Custom/Invisible"  
{  
    Properties  
    {  
    }  
    SubShader  
    {  
        Tags { "RenderType" = "Opaque" "Queue"="Geometry-1" }  
        LOD 100  
        Blend Zero One  
  
        Pass  
        {  
        }  
    }  
}
```

Listing 4.10: Codice dello Shader



Figura 4.7: Con Shader l'occlusione è corretta

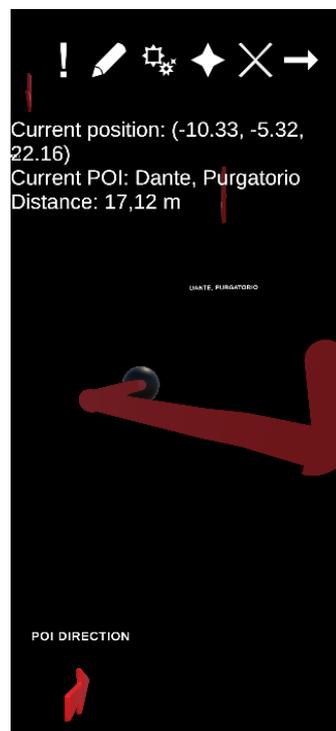


Figura 4.8: Senza Shader è possibile vedere attraverso i muri

4.3 Assistente LLM e Accessibilità

Quello che distingue questa applicazione da molte altre è l'integrazione di tecnologie avanzate per migliorare l'accessibilità e l'esperienza utente.

In questo capitolo viene approfondita l'implementazione di un assistente virtuale basato su un Large Language Model (LLM) e tecnologia Text-to-Speech (TTS). La combinazione di queste due funzionalità consente di simulare una guida turistica interattiva, offrendo un'esperienza più coinvolgente e personalizzata per gli utenti.

4.3.1 Sistema di Chat con LLM

Si è scelto di implementare una chat in-app utilizzando l'API di *Gemini 2.0 Flash*, il nuovissimo modello di Google; si tratta di un notevole miglioramento rispetto alla precedente versione, la 1.5, e permette di generare risposte molto più consistenti e pertinenti. L'integrazione è stata realizzata attraverso la classe *LLMChatManager*, che gestisce l'intera comunicazione tra utente e modello.

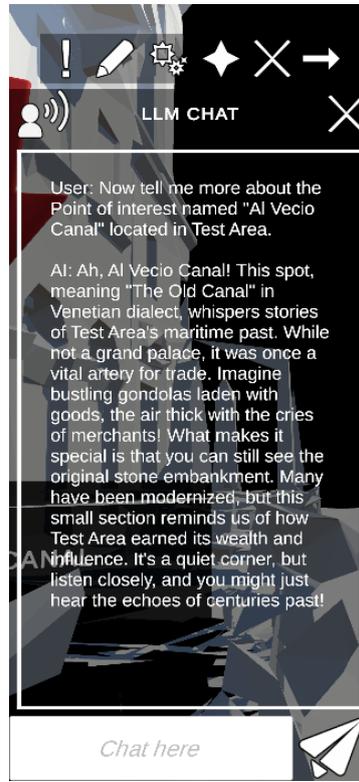


Figura 4.9: Interfaccia della chat con l'assistente

Comportamento e Contesto

È stato utilizzato un prompt di base per definire il comportamento che l'assistente dovrà mantenere, ovvero quello di una guida turistica in grado di fornire informazioni storiche, aneddoti e dettagli culturalmente rilevanti.

```
private string basePrompt = @"You are a knowledgeable tour guide.
When asked about Points of Interest, provide: key historical facts,
unique stories, and cultural significance. Keep responses brief,
engaging, and in English. Write everything in plain text only,
without any special formatting and DO NOT search online! Focus on
the most interesting aspects that a tourist would want to know.";
```

Listing 4.11: Prompt di base per l'LLM

Gestione della API Key

Per proteggere la API key di Gemini, è stata sviluppata una classe dedicata, *APIKeyEncryption*, che gestisce la cifratura e la decifrazione della chiave utilizzando l'algoritmo AES.

Questo approccio evita di memorizzarla direttamente nel codice sorgente, riducendo il rischio di esposizione tramite reverse engineering o accessi non autorizzati ai repository, che potrebbero consentire a malintenzionati di utilizzarla per generare risposte sfruttandola.

Per semplificare la gestione delle chiavi, è stato implementato uno Script che va a creare un Custom Editor (Editor Extension) ovvero un menu all'interno dell'editor di Unity che consente di gestire l'API key graficamente, salvandola in un file interno al progetto. Con questo metodo sarà inclusa direttamente nella build dell'applicazione senza esporla nel codice.

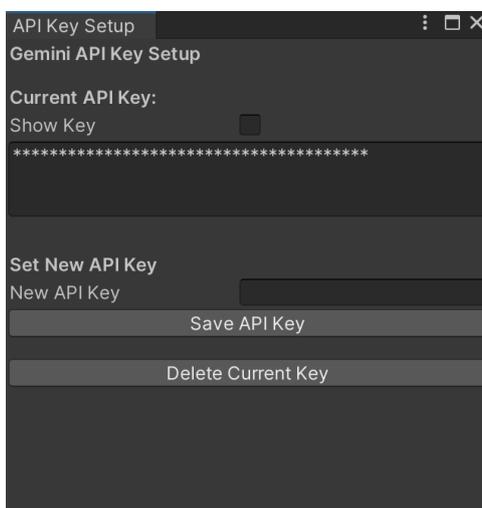


Figura 4.10: Interfaccia del menu per la gestione dell'API key

4.3.2 Text-to-Speech

Questa tecnologia è un importante elemento aggiunto per garantire una maggiore accessibilità, risultando particolarmente utile in contesti di mobilità o per utenti con disabilità visive.

Ogni risposta generata dall'IA verrà letta ad alta voce, così da simulare una vera e propria guida turistica. L'implementazione si basa su una libreria nativa chiamata *ttsrust*, scritta in Rust, che è stata integrata nell'applicazione tramite la classe statica *TextToSpeech* [10].

Questa classe funge da wrapper: espone dei metodi appositi per interfacciarsi con la libreria nativa attraverso l'uso di chiamate *P/Invoke* (*Platform Invocation Services*), che permettono a C# di utilizzare funzioni esposte da librerie native.

```

public static class TextToSpeech {

    [DllImport("ttsrust.dll", CallingConvention = CallingConvention.
        Cdecl)]
    private static extern void ttsrust_say(string text);

    // ...

    public static void Speak(string text) {
        try {
            ttsrust_say(text);
        } catch (System.Exception e) {
            Debug.LogError($"Text-to-speech - impossibile parlare: {e.
                Message}");
        }
    }

    // ...

}

```

Listing 4.12: Classe TextToSpeech

4.4 Pathfinding e Navigazione

L'applicazione sviluppata implementa un sistema di pathfinding avanzato basato sul nuovo pacchetto *AI Navigation* di Unity per guidare l'utente verso il punto di interesse da lui scelto.

Di seguito verranno trattati i dettagli implementativi.

4.4.1 Unity AI Navigaiton

Il pacchetto AI Navigation semplifica l'implementazione del pathfinding, permettendo la generazione automatica di una *NavMesh* attraverso un processo che prende il nome di *baking*. Così facendo vengono definite le aree percorribili e gli ostacoli, consentendo agli agenti di muoversi in modo da evitare collisioni.

A partire dalla versione 2022 di Unity, questa funzionalità non è più integrata direttamente nel core dell'editor, ma è stata separata in un pacchetto dedicato.

Le principali caratteristiche offerte sono: [49]

- **Generazione dinamica del NavMesh tramite *NavMesh Surface*:** È possibile aggiornare dinamicamente e a runtime la mesh di navigazione in risposta ai cambiamenti dell'ambiente (nuovi ostacoli, ecc.).

- **Agenti personalizzabili:** Si possono creare agenti personalizzati regolandone parametri come la velocità, accelerazione e dimensioni.
- **Algoritmi di pathfinding ottimizzati:** Sono stati migliorati gli algoritmi per il calcolo dei percorsi, garantendo movimenti più naturali senza però appesantire le prestazioni;
- **NavMesh Links:** Sono degli elementi speciali che permettono di gestire ostacoli come salti, scale, o percorsi non convenzionali (es. un agente che salta da un tetto a un altro).
- **Debug avanzato:** Sono disponibili strumenti che permettono agli sviluppatori di configurare, monitorare e modificare in tempo reale i parametri di navigazione facilitando il lavoro.

4.4.2 NPC

Gli agenti che guidano l'utente verso i punti di interesse sono stati chiamati internamente *NPC (Non-Player Character)*. Hanno l'aspetto di sfere semitrasparenti che lasciano dietro di sé una scia rossa.

Grazie a Unity AI Navigation, è stato possibile personalizzarli e introdurne due tipologie:

- **Indoor:** Creati per operare in spazi interni come edifici, palazzi e, in generale, strutture chiuse. Sono caratterizzati da un parametro *radius* impostato a 0.25, che garantisce la navigazione di passaggi più stretti tipici degli ambienti confinati.
- **Outdoor:** Ottimizzati per scenari esterni come piazze, strade e spazi urbani. Questo tipo ha un radius più ampio impostato a 0.5, il che lo rende più appropriato per luoghi spaziosi dove non è necessaria un'alta precisione di movimento.

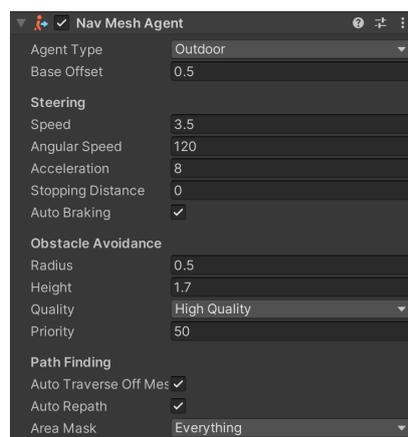


Figura 4.11: Impostazioni agent per la navigazione

Questa differenziazione è stata creata dopo una serie di test. Si è notato che per i luoghi interni un radius pari a 0.5 era troppo grande e non permetteva di generare correttamente la NavMesh nel caso di percorsi stretti come porte o corridoi. Questo portava ad alcune aree inaccessibili agli agenti e, di conseguenza, non era possibile guidare correttamente l'utente lungo l'intero percorso.

La soluzione permette una selezione automatica a runtime di una di queste due tipologie da parte della classe NavigationManager in base al parametro *LocationType* presente nel LocationDatabase.

```
private void SpawnNPCBasedOnLocationType(LocationType locationType) {  
  
    // Scelgo il tipo di NPC da istanziare in base alla location  
    GameObject npcPrefab = locationType == LocationType.Indoor ?  
        npcIndoorPrefab : npcOutdoorPrefab;  
    SpawnNPC(npcPrefab, npcSpawnDistance);  
}
```

Listing 4.13: Metodo per la selezione dell'NPC in base al tipo di luogo

4.4.3 Rilevamento Punti di Interesse Irraggiungibili

Il sistema implementa un algoritmo per determinare se un certo punto di interesse è raggiungibile o meno dalla posizione attuale dell'utente. Nel caso negativo, viene mostrato un menu appropriato informa dell'errore.

Questo è fondamentale perché potrebbero esserci alcune incongruenze durante la navigazione, come ad esempio:

- **Errori nella codifica delle coordinate del POI:** Il file dei POI potrebbe non essere configurato correttamente, e quindi le loro coordinate possono trovarsi al di fuori dei confini navigabili.
- **Problemi di baking della NavMesh:** Aree complesse o con geometrie particolari potrebbero non essere elaborate correttamente, creando discontinuità.

Per risolvere questi inconvenienti, è stato scritto un algoritmo che sfrutta il NavMesh di Unity per determinare se esiste un percorso valido tra due punti.

```
private bool IsDestinationReachable(Vector3 destination, Vector3  
startPosition) {  
  
    NavMeshHit hit;  
  
    if (NavMesh.SamplePosition(destination, out hit, maxDistance,  
        NavMesh.AllAreas)) {
```

```
bool hasPath = NavMesh.CalculatePath(startPosition, hit.  
    position, NavMesh.AllAreas, pathToCheck);  
  
return hasPath && pathToCheck.status == NavMeshPathStatus.  
    PathComplete;  
}  
return false;  
}
```

Listing 4.14: Metodo per verificare se esiste un percorso valido

1. **Si ricerca il punto navigabile più vicino:** Grazie a *NavMesh.SamplePosition*, il sistema identifica il punto sulla NavMesh più vicino alla destinazione desiderata.
2. **Calcolo e validazione del percorso:** Una volta identificato un punto valido sulla NavMesh, si usa il metodo *NavMesh.CalculatePath* che tenta di calcolare un percorso completo dalla posizione dell'utente fino al punto navigabile più vicino alla destinazione.
3. La verifica finale richiede il soddisfacimento di due condizioni:
 - (a) L'esistenza di un percorso (indicata dal valore booleano restituito da *CalculatePath*).
 - (b) La completezza del percorso (verificata attraverso lo stato *NavMeshPathStatus.PathComplete*).

Questa doppia verifica garantisce che esista non solo un percorso teorico, ma che questo sia effettivamente completo e percorribile dall'inizio alla fine senza interruzioni.

4. Un ultimo aspetto rilevante è il parametro *maxDistance*, che definisce il raggio massimo entro cui cercare un punto navigabile. Dopo diversi test, si è stabilito che un valore di 5.0 unità rappresenta il miglior compromesso tra precisione e flessibilità.

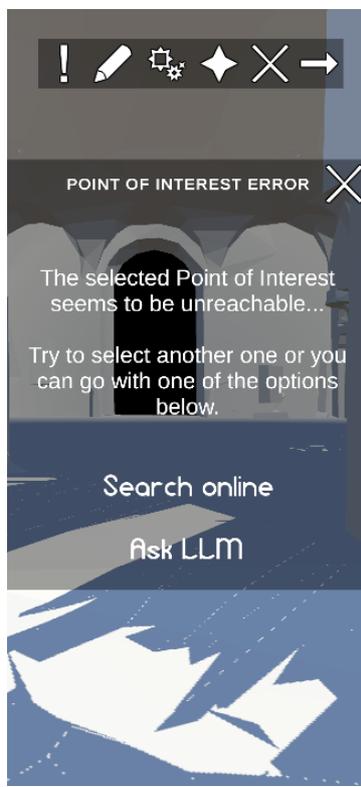


Figura 4.12: Menu che avvisa la scelta di un POI non raggiungibile

Capitolo 5

Conclusioni e Sviluppi Futuri

Lo sviluppo di questa applicazione ha presentato alcune problematiche principalmente legate all'architettura del sistema e alla precisione dell'allineamento tra il mondo virtuale e quello reale:

- **Duplicazione di codice e componenti:** Manager e UI sono presenti in molteplici Scene. Queste ripetizioni vanno ad aumentare la complessità nella gestione delle modifiche.
- **Difficoltà di manutenzione:** Qualsiasi cambiamento a questi componenti richiede modifiche a catena in ogni Scene.

L'altra criticità riguarda proprio il sistema di posizionamento e la scansione del QR code:

- **Precisione SP virtuale e fisico:** È il principale problema, infatti, la posizione dello Starting Point virtuale deve essere perfettamente allineata con la posizione di quello fisico. Questo è fondamentale per permettere di avviare dei tour coerenti con il luogo in cui si trova l'utente.
- **Dipendenza da QR code:** Il sistema attuale dipende dalla scansione di un codice QR come punto di riferimento iniziale. Questo approccio, sebbene funzionale, introduce un problema qualora venga danneggiato o spostato.

Nonostante queste limitazioni, la scelta di un'architettura più semplice è stata consapevole, poiché ha permesso di accelerare lo sviluppo e il testing delle funzionalità. In un contesto prototipale, privilegiare la velocità di implementazione rispetto all'eleganza architettonica ha consentito di esplorare rapidamente diverse soluzioni tecniche e validarne l'efficacia.

Per le future evoluzioni dell'app, sarà fondamentale adottare un'architettura più modulare, con componenti riutilizzabili che riducano la duplicazione del codice e migliorino la

manutenibilità complessiva. Inoltre, per mitigare la dipendenza dalla precisione del posizionamento iniziale, si potranno utilizzare tecnologie più avanzate per la localizzazione. Soluzioni come il *Bluetooth Low Energy (BLE)* con beacon strategicamente posizionati potrebbero offrire una triangolazione più accurata della posizione dell'utente, riducendo l'affidamento su un singolo riferimento QR. Altre alternative includono sistemi di localizzazione indoor basati su *Wi-Fi RTT (Round Trip Time)* e *Ultra-wideband (UWB)* per misurazioni ad alta precisione.

L'analisi delle problematiche evidenziate ha inoltre permesso di individuare alcune aree di miglioramento che potrebbero essere affrontate in futuri aggiornamenti dell'applicazione. Di seguito, vengono illustrate alcune possibili migliorie:

- **Implementazione completa del sistema *MasterManager*:** Come già accennato, attualmente il *MasterManager* contiene solo un oggetto: il *LocationDatabase*. In futuro, un'ottima soluzione sarebbe quella di inserirvi all'interno anche il resto dei *Manager* e tutto ciò che riguarda la UI. In questo modo non c'è la necessità di modificare ogni *Scena* se si tocca uno di questi componenti.
- **Miglioramento dell'assistente in-app:** Si potrebbero aggiungere funzionalità avanzate come i *quick prompt*, ovvero bottoni cliccabili sull'interfaccia della chat che mostrano altre domande pertinenti da fare all'assistente. Ovviamente queste cambieranno a seconda del contesto e delle scorse conversazioni fatte con l'IA e, se l'utente ne preme uno, verrà inoltrato immediatamente il messaggio alleviando la fatica di scrivere ogni cosa a mano.
- **Migliorare il *Text-to-Speech*:** Utilizzare soluzioni più avanzate oppure asset a pagamento dal Marketplace di Unity che permettono di includere modelli vocali molto più avanzati e naturali. L'attuale TTS, sebbene funzionale, ha alcuni problemi: innanzitutto utilizza la voce del *Text-to-Speech* di sistema e, di conseguenza, la lingua del sistema operativo. Purtroppo, questo porta ad una lettura con un accento errato.
- **Aggiungere più lingue:** Attualmente l'app supporta soltanto l'inglese. In futuro bisognerebbe implementare anche un sistema di cambio della lingua in modo da garantire una maggiore fruibilità.
- **Implementazione di un avatar per la guida virtuale:** Integrare un avatar 3D animato che rappresenti l'assistente virtuale, aumentando significativamente la naturalezza dell'interazione. Questo avatar andrebbe ad accompagnare l'utente durante la navigazione, potrebbe eseguire gesti per indicare i punti di interesse e addirittura mostrare espressioni facciali appropriate al contesto della conversazione. L'implementazione richiederebbe modelli 3D leggeri e un sistema di animazioni procedurali per non appesantire l'applicazione.

- **Aggiunta di ulteriori modelli LLM:** Permettere all'app di supportare diversi modelli linguistici oltre a Gemini, come Claude di Anthropic, Llama di Meta o GPT di OpenAI. Questo permetterebbe di scegliere il modello più appropriato in base a prestazioni, costi e caratteristiche.

Inoltre, si potrebbe implementare un sistema di fallback che utilizzi modelli alternativi quando uno non è disponibile, garantendo così maggiore robustezza.

Un'altra possibilità sarebbe quella di specializzare diversi modelli per compiti specifici (uno per le informazioni storiche, uno per la navigazione, ecc.).
- **Aggiunta di un abbonamento premium:** Sviluppare un modello di business sostenibile attraverso un sistema di abbonamento che offra funzionalità avanzate come accesso a modelli LLM più potenti, contenuti esclusivi, download offline di tour specifici, integrazione con audio-guide professionali, rimozione di pubblicità e limiti di utilizzo.

Il sistema potrebbe prevedere diversi livelli (mensile, annuale, familiare) per massimizzare l'accessibilità a diverse tipologie di utenti.
- **Creazione di un sistema di feedback:** Implementare un meccanismo che permetta agli utenti di segnalare informazioni errate o imprecise fornite dal sistema e molto altro. Questo feedback verrebbe poi utilizzato per migliorare costantemente la qualità dell'app attraverso la creazione di un database di correzioni. Il programma potrebbe anche includere una funzione di *fact-checking* che verifichi automaticamente le informazioni usando fonti affidabili prima di presentarle all'utente.
- **Sistema di aggiunta di punti di interesse da parte della community:** Sviluppare una piattaforma collaborativa che consenta agli utenti di suggerire nuovi POI da aggiungere al database dell'applicazione.

I suggerimenti dovrebbero prima passare attraverso un processo di moderazione e verifica da parte del team di sviluppo in modo da essere integrati definitivamente nell'app. Il sistema potrebbe anche implementare meccanismi di gamification, come badge o riconoscimenti per gli utenti più attivi e collaborativi, incentivando così la partecipazione della community alla crescita della piattaforma.

Appendice

Ringraziamenti

Eccomi alla fine di un viaggio, uno lungo aggiungerei. Anche questo capitolo della mia vita volge al termine e, insieme ad esso, tanti ricordi. Sono arrivato fino a qui grazie a molte persone, non lo nego, e per questo sono davvero grato.

In primis ringrazio il mio relatore, il Prof. Cascarano e al mio tutor aziendale Luca Vitale. Siete entrambi delle persone davvero pazienti e alla mano, è stato un vero piacere lavorare con voi.

Un grazie speciale va ai miei genitori, che mi hanno sostenuto sia emotivamente che finanziariamente in questo percorso senza chiedermi troppe volte "*ma quando ti laurei?*". Un altro grazie va a mia sorella Valeria e a suo marito Amir.

E ora inizia la parte del lungo ringraziamento agli amici che mi hanno sostenuto, sì, proprio tutti.

Partiamo dalla mia compagnia di Mantova, ringrazio Dano, Miguel, Lav, Jashn, Andrea, Eros, Manu, Tonno, Alyssa, Giulia, Duegor, Francesca, Pio, Cavi, Marco e Bara.

Importante includere anche i nuovi acquisti, ovvero Bonzo, Mork, Pinci e Kelle.

Si passa ora ai miei coinquilini, Elia, l'altro Elia (anche se non è più con noi, nel senso che si è trasferito) e Lorenzo.

E per ultimi, ma non per importanza, i cari amici che ho conosciuto qui a Bologna, senza di voi questa magistrale sarebbe stata davvero noiosa, mi avete rallegrato le giornate.

Partiamo da Leo, Giulia, Marco, Claudia (le sue coinquiline Francesca, Cecilia e Francesca), Cristina, Chiara, Sté, Matteo, Steve, Rickyzz, Luca, FraBug, Fabio Antonello, Gloria, Fiore, Zak, Angy, Alessandra, Lisa, Virginia, e Lorenzo.

Che dire, ho passato del bellissimo tempo con ognuno di voi, ne sono davvero contento. Potrei infine esordire con "*magari fosse durato di più*", ma purtroppo niente dura per sempre. Bisogna andare avanti e continuare a migliorarsi un passo alla volta, trasformando le esperienze negative in lezioni preziose, anziché lasciare che distruggano il morale. Dopo tutto, sono proprio gli ostacoli che fanno crescere, esattamente come dice il detto ad inizio tesi.

Lascio queste parole anche per il me del futuro, affinché un giorno, rilegendole, possa riconoscere che il me del passato aveva buone ragioni per essere così speranzoso.

Bibliografia

- [1] Accenture. *Empowering film creatives with digital twins*. Accenture. 2025. URL: <https://www.accenture.com/it-it/case-studies/communications-media/empowering-film-creatives-digital-twins> (visitato il giorno 19/02/2025).
- [2] Ian L. Alberts et al. «Large language models (LLM) and ChatGPT: what will the impact on nuclear medicine be?» In: *European Journal of Nuclear Medicine and Molecular Imaging* 50.6 (2023), pp. 1549–1552. ISSN: 1619-7089. DOI: 10.1007/s00259-023-06172-w. URL: <https://link.springer.com/article/10.1007/s00259-023-06172-w>.
- [3] Amanatullah. *Transformer Architecture Explained*. 2023. URL: <https://medium.com/@amanatulla1606/transformer-architecture-explained-2c49e2257b4c> (visitato il giorno 21/02/2025).
- [4] Askanews. *Al Bto 2023 arriva Amelia, la 'Digital twin' italiana del turismo*. 2023. URL: <https://www.idealista.it/news/finanza/economia/2023/11/24/176815-al-bto-2023-arriva-amelia-la-digital-twin-italiana-del-turismo> (visitato il giorno 03/03/2025).
- [5] Ronald T. Azuma. «A Survey of Augmented Reality». In: *Presence: Teleoperators and Virtual Environments* 6.4 (1997), pp. 355–385. DOI: 10.1162/pres.1997.6.4.355. URL: <https://www.cs.unc.edu/~azuma/ARpresence.pdf>.
- [6] Blue Brain Project. *Blue Brain - Research Domains*. École Polytechnique Fédérale de Lausanne. 2024. URL: <https://bbp.epfl.ch/bbp/research/domains/bluebrain/index.html> (visitato il giorno 19/02/2025).
- [7] Antonino Caffo. *La città tedesca che usa il digital twin per lanciare il turismo*. 2020. URL: <https://www.datamanager.it/2020/06/la-citta-tedesca-che-usa-il-digital-twin-per-lanciare-il-turismo/>.
- [8] Carraro Lab. *Digital Twin e oggetti virtuali*. Carraro Lab. 2024. URL: <https://www.carraro-lab.com/digital-twin-e-oggetti-virtuali/> (visitato il giorno 19/02/2025).

- [9] Designium. *Designium develops a travel app "AR Travel Concierge", a new sightseeing experience realized with AI assistant and AR navigation using ChatGPT*. 2023. URL: <https://medium.com/designium-storyhub/designium-develops-a-travel-app-ar-travel-concierge-provisional-name-a-new-sightseeing-b570ea3b6cb1> (visitato il giorno 27/02/2025).
- [10] Docs.rs. *Text-to-Speech Rust library*. 2025. URL: <https://docs.rs/tts/latest/tts/> (visitato il giorno 24/02/2025).
- [11] Tolga Erol, Arif Furkan Mendi e Dilara Doğan. «The Digital Twin Revolution in Healthcare». In: *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. 2020. DOI: 10.1109/ISMSIT50672.2020.9255249. URL: <https://ieeexplore.ieee.org/abstract/document/9255249>.
- [12] Fortune Business Insights. *Digital Twin Market Size*. Fortune Business Insights. 2024. URL: <https://www.fortunebusinessinsights.com/digital-twin-market-106246> (visitato il giorno 19/02/2024).
- [13] Yuto Fushimi. *CSV Parser for C#*. 2023. URL: <https://github.com/yutokun/CSV-Parser> (visitato il giorno 23/02/2025).
- [14] Francesco Gabellone. «Digital Twin: a new perspective for cultural heritage management and fruition». In: *Acta IMEKO* 11.1 (mar. 2022). identifier: IMEKO-ACTA-11 (2022)-01-05, pp. 1–7. DOI: 10.21014/acta_imeko.v11i1.1214. URL: [https://acta.imeko.org/index.php/acta-imeko/article/view/IMEKO-ACTA-11%20\(2022\)-01-05](https://acta.imeko.org/index.php/acta-imeko/article/view/IMEKO-ACTA-11%20(2022)-01-05).
- [15] Gamespot. *Inizimals Review*. URL: <https://www.gamespot.com/reviews/invizimals-review/1900-6283426/> (visitato il giorno 02/03/2025).
- [16] Francesco Garofalo. *ARKit vs ARCore*. 2022. URL: <https://www.francescogarofalo.it/post/arkit-vs-arcore/> (visitato il giorno 15/02/2025).
- [17] Leica Geosystems. *Blockbuster Scanning: Using LiDAR for Joker and John Wick*. 2020. URL: <https://leica-geosystems.com/case-studies/reality-capture/blockbuster-scanning-using-lidar-for-joker-and-john-wick> (visitato il giorno 26/02/2025).
- [18] Edward H Glaessgen e David S Stargel. «The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles». In: *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. NASA. 2012. DOI: 10.2514/6.2012-1818. URL: <https://ntrs.nasa.gov/citations/20120008178>.
- [19] Niklas Heidloff. *Foundation Models, Transformers, BERT and GPT*. 2023. URL: <https://heidloff.net/article/foundation-models-transformers-bert-and-gpt/> (visitato il giorno 02/03/2025).

- [20] HomeTrack. *What is Matterport?* 2024. URL: <https://www.hometrack.net/blog/what-is-matterport> (visitato il giorno 20/02/2025).
- [21] Bo-Chen Huang et al. «ARBIN: Augmented Reality Based Indoor Navigation System». In: *Sensors* 20.20 (2020). ISSN: 1424-8220. DOI: 10.3390/s20205890. URL: <https://www.mdpi.com/1424-8220/20/20/5890>.
- [22] Haomei Jia e Jing Yan. «Construction of Heritage Digital Resource Platform Based on Digital Twin Technology». In: *Mathematical Problems in Engineering* 2022 (ago. 2022), pp. 1–8. DOI: 10.1155/2022/4361135. URL: https://www.researchgate.net/publication/362767296_Construction_of_Heritage_Digital_Resource_Platform_Based_on_Digital_Twin_Technology.
- [23] Liang Li. «The influence of digital twins on the methods of film and television creation». In: *Computers and Electrical Engineering* 103 (2022), p. 108314. ISSN: 0045-7906. DOI: <https://doi.org/10.1016/j.compeleceng.2022.108314>. URL: <https://www.sciencedirect.com/science/article/pii/S0045790622005365>.
- [24] McKinsey & Company. *What is digital twin technology?* McKinsey & Company. 2024. URL: <https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-digital-twin-technology> (visitato il giorno 19/02/2025).
- [25] Meta for Work. *Differenza tra i vari tipi di Realtà*. 2025. URL: <https://forwork.meta.com/it/blog/difference-between-vr-ar-and-mr/> (visitato il giorno 14/02/2025).
- [26] Paul Milgram et al. «Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum». In: *Proceedings of SPIE. Telemanipulator and Telepresence Technologies* 2351 (1994), pp. 282–292. DOI: 10.1117/12.197321. URL: https://www.researchgate.net/publication/231514051_A_Taxonomy_of_Mixed_Reality_Visual_Displays.
- [27] Mirai. *Mirai Twin: an immersive hotel experience through a 3D digital twin*. 2024. URL: <https://www.mirai.com/blog/mirai-twin-an-immersive-hotel-experience-through-a-3d-digital-twin/> (visitato il giorno 03/03/2025).
- [28] Mono Project. *About Mono*. Mono Project. 2025. URL: <https://www.mono-project.com/docs/about-mono/> (visitato il giorno 20/02/2025).
- [29] Andrew Yeh Ching Nee e Soh Khim Ong, cur. *Springer Handbook of Augmented Reality*. Singapore: Springer, 2023. ISBN: 978-3-030-67821-0.
- [30] Devang Kishor Parab et al. «Indoor Navigation System using Augmented Reality». In: *2024 International Conference on Inventive Computation Technologies (ICICT)*. 2024, pp. 720–725. DOI: 10.1109/ICICT60155.2024.10545015. URL: <https://ieeexplore.ieee.org/abstract/document/10545015>.

- [31] Philips Healthcare. *HeartModel - Ultrasound Solutions*. Philips Healthcare. 2025. URL: <https://www.philips.it/healthcare/resources/feature-detail/ultrasound-heartmodel> (visitato il giorno 19/02/2025).
- [32] Program Ace. *Types of Digital Twins*. 2023. URL: <https://program-ace.com/blog/types-of-digital-twins/> (visitato il giorno 18/02/2025).
- [33] PTC Inc. *Vuforia Features*. Vuforia. 2024. URL: <https://developer.vuforia.com/library/getting-started/vuforia-features> (visitato il giorno 18/02/2025).
- [34] Jim Richardson. *What Digital Twin Technology Means for Museums*. 2020. URL: <https://www.museumnext.com/article/what-digital-twin-technology-means-for-museums/> (visitato il giorno 03/03/2025).
- [35] Scene3D. *What is Matterport?* 2024. URL: <https://scene3d.co.uk/what-is-matterport/> (visitato il giorno 20/02/2025).
- [36] Siemens Healthineers. *Siemens Healthineers - Medical Technology Company*. Siemens Healthineers. 2025. URL: <https://www.siemens-healthineers.com> (visitato il giorno 19/02/2025).
- [37] Sim&Cure. *Patient-Based Simulation for Intracranial Aneurysm Treatment*. Sim&Cure. 2024. URL: <https://sim-and-cure.com/> (visitato il giorno 19/02/2025).
- [38] Seungyoub Ssin et al. «Science Tour and Business Model Using Digital Twin-Based Augmented Reality». In: *Augmented Reality and Virtual Reality*. A cura di M. Claudia tom Dieck, Timothy H. Jung e Sandra M. C. Loureiro. Cham: Springer International Publishing, 2021, pp. 267–276. ISBN: 978-3-030-68086-2. URL: https://link.springer.com/chapter/10.1007/978-3-030-68086-2_20#citeas.
- [39] Statista. *Revenue in the AR Software segment worldwide from 2019 to 2029*. Statista. 2024. URL: <https://www.statista.com/forecasts/1337258/ar-software-b2c-market-revenue-worldwide> (visitato il giorno 18/02/2025).
- [40] Unity Technologies. *Compare Unity Plans*. 2024. URL: <https://unity.com/products/compare-plans> (visitato il giorno 18/02/2025).
- [41] Unity Technologies. *Supported 3D file formats*. Unity Technologies. 2025. URL: <https://docs.unity3d.com/Manual/3D-formats.html> (visitato il giorno 23/02/2025).
- [42] TimeLooper. *TimeLooper – Immersive Historical Experiences*. URL: <https://www.timelooper.com/> (visitato il giorno 02/03/2025).
- [43] Unity Technologies. *AR Foundation*. Version 6.0. Unity Technologies, 2024. URL: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@6.0/manual/index.html> (visitato il giorno 15/02/2025).

- [44] Unity Technologies. *AR Foundation Upgrade Guide*. 2025. URL: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.1/manual/version-history/upgrade-guide.html> (visitato il giorno 28/02/2025).
- [45] Unity Technologies. *Explore the Unity Editor*. URL: <https://learn.unity.com/tutorial/explore-the-unity-editor-1> (visitato il giorno 05/03/2025).
- [46] Unity Technologies. *What is the Unity Asset Store and how do I purchase Assets?* URL: <https://support.unity.com/hc/en-us/articles/210142503-What-is-the-Unity-Asset-Store-and-how-do-I-purchase-Assets> (visitato il giorno 05/03/2025).
- [47] Studio - UXPin. *Beyond the Hype: A UX Reality Check on Pokémon Go*. URL: <https://www.uxpin.com/studio/blog/beyond-hype-ux-reality-check-pokemon-go/> (visitato il giorno 02/03/2025).
- [48] Niraj Vishwakarma. *How Unity Supports Cross Platform Feature*. 2020. URL: <https://niraj-vishwakarma.medium.com/how-unity-supports-cross-platform-feature-ae722321cfa> (visitato il giorno 19/02/2025).
- [49] VRChat Creators. *AI Navigation - VRChat Creator Documentation*. VRChat. 2024. URL: <https://creators.vrchat.com/worlds/udon/ai-navigation/> (visitato il giorno 25/02/2025).
- [50] Zhan Wang et al. «VirtuWander: Enhancing Multi-modal Interaction for Virtual Tour Guidance through Large Language Models». In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. CHI '24. ACM, mag. 2024, pp. 1–20. DOI: 10.1145/3613904.3642235. URL: <http://dx.doi.org/10.1145/3613904.3642235>.
- [51] Wikipedia contributors. *Porting* — *Wikipedia, The Free Encyclopedia*. 2025. URL: <https://en.wikipedia.org/wiki/Porting> (visitato il giorno 20/02/2025).
- [52] Wikipedia contributors. *Porting* — *Wikipedia, The Free Encyclopedia*. 2025. URL: https://en.wikipedia.org/wiki/Large_language_model (visitato il giorno 21/02/2025).
- [53] Wikipedia contributors. *Transformer (deep learning architecture)* — *Wikipedia, The Free Encyclopedia*. 2025. URL: [https://en.wikipedia.org/wiki/Transformer_\(deep_learning_architecture\)](https://en.wikipedia.org/wiki/Transformer_(deep_learning_architecture)) (visitato il giorno 21/02/2025).
- [54] Wikipedia contributors. *Vuforia* — *Wikipedia, The Free Encyclopedia*. Wikipedia. 2025. URL: https://en.wikipedia.org/wiki/Vuforia_Augmented_Reality_SDK (visitato il giorno 19/02/2025).
- [55] Wikipedia contributors. *Wavefront .obj file* — *Wikipedia, The Free Encyclopedia*. 2025. URL: https://en.wikipedia.org/wiki/Wavefront_.obj_file (visitato il giorno 23/02/2025).

- [56] Sarah Wray. *How a small German town is using an advanced digital twin*. 2020.
URL: <https://cities-today.com/how-a-small-german-town-is-using-an-advanced-digital-twin/>.