

ALMA MATER STUDIORUM - UNIVERSITA' DI BOLOGNA
CAMPUS DI CESENA

DIPARTIMENTO DI INFORMATICA – SCIENZA E INGEGNERIA
CORSO DI LAUREA IN INGEGNERIA E SCIENZE INFORMATICHE

TITOLO DELL'ELABORATO

**STUDIO DELLA REALTÀ AUMENTATA E DEL SUO IMPIEGO NEL
CAMPO DELL'INDUSTRIAL DESIGN**

Elaborato in

Programmazione di Sistemi Mobile

Relatore
Dott.ssa Catia Prandi

Presentata da
Tommaso Mandoloni

Correlatore
Dott. Marco Rossetti

Sessione II
Anno Accademico 2019/2020

Indice

INTRODUZIONE	1
1 CONCETTI FONDAMENTALI	5
1.1 REALTA AUMENTATA	6
1.1.1 Che cos'è la realtà aumentata	6
1.1.2 Terminologie Appropriate	7
1.1.3 Sviluppo nel Tempo	8
1.1.4 Location Based, Marker, Markerless AR	11
1.1.5 Contesti di Utilizzo	13
1.2 INDUSTRIAL DESIGN	16
1.2.1 IKEA Place	18
1.3 IDEA PROGETTUALE	19
1.3.1 L'idea nasce dall'esigenza	20
1.3.2 Funzionalità Basilari	20
1.3.3 Considerazioni Progettuali	21

2	PROGETTAZIONE DEL SISTEMA	23
2.1	DESIGN	23
2.1.1	Mockup	24
2.1.2	User Group	27
2.2	TECNOLOGIE UTILIZZATE	28
2.2.1	Unity	29
2.2.2	AR Foundation	32
2.2.3	ARKit	36
2.2.4	ARCore	38
3	APP SVILUPPATA	41
3.1	SCELTE IMPLEMENTATIVE	41
3.1.1	Lista Materiali	42
3.1.2	Occlusion Walls	43
3.1.3	Trasformazione Coordinate	45
3.1.4	Triangolazione Superficie	45
3.2	SCHERMATE E FUNZIONALITÀ	46
3.2.1	Tracciamento	46
3.2.2	Visualizzazione	52
3.2.3	Modifica	53
3.3	SVILUPPI FUTURI	56
3.3.1	Database	56
3.3.2	Tracciamento Verticale	57
3.3.3	Condivisione Screenshot	58
	CONCLUSIONE	61
	BIBLIOGRAFIA	63

Introduzione

Il progetto di tesi si colloca in un contesto di *industrial design*, che prevede la progettazione e lo sviluppo di applicazioni destinate ad un utilizzo per lo più economico-industriale. Attraverso tale termine vengono descritti i contesti con cui il progetto è stato pensato e realizzato: si è adottato un approccio di design del prodotto vero e proprio, poiché l'utente deve essere in grado di usare l'applicazione in modo rapido e veloce, e capirne le funzionalità in maniera chiara e intuitiva attraverso la realizzazione di un'interfaccia grafica che glielo permetta; mentre è stato adottato un approccio orientato ad un contesto industriale per calare il prodotto in uno scopo ben preciso, quello della vendita su mercato. L'applicazione dovrà essere venduta e, di conseguenza, dovranno esserci possibili acquirenti intenti ad acquistarla. Per questo motivo ci si è concentrati sull'aspetto funzionale del software, quindi su quali servizi debba offrire l'applicazione per far in modo che abbia successo nel mondo del mercato.

In questo contesto, l'obiettivo di questa tesi è il design e lo sviluppo di un'applicazione per dispositivi mobili che implementi funzionalità appartenenti ad uno dei concetti più emergenti e discussi degli ultimi tempi: l'*augmented reality*, ovvero la realtà aumentata (AR).

Con il termine realtà aumentata si intendono quelle tecnologie mirate ad arricchire la percezione sensoriale dell'uomo, in modo da fornire un'esperienza più immersiva e migliorata del mondo reale attraverso l'inserimento di elementi virtuali.

Sono state quindi impiegate tecniche di realtà aumentata in un'applicazione mobile che ha l'obiettivo di facilitare un eventuale venditore di superfici per pavimenti a trovare un possibile acquirente, mostrando in tempo reale il risultato di un pavimento virtuale direttamente a casa del cliente, solamente attraverso l'utilizzo del proprio dispositivo. L'applicazione quindi implementa tutte quelle funzionalità finalizzate alla vendita, di cui chi vende il prodotto in questione ha bisogno per avere maggiore probabilità di successo. Al giorno d'oggi, le persone fanno fatica ad immaginarsi una realtà diversa da quella a cui sono abituate quotidianamente, e per questo motivo nasce l'esigenza di avere uno strumento specializzato che permetta di creare una realtà fittizia ma che, attraverso un'esperienza più coinvolgente, metta a proprio agio l'utente.

L'applicazione, sviluppata tramite motore grafico Unity che permette di sviluppare applicazioni multiplatforma, ha subito una fase di progettazione finalizzata al suo impiego nel campo dell'industria e alla soddisfazione dell'utente (*human-centered design*) che la utilizza. Infatti, la fase di progettazione è stata di cruciale importanza poiché è in quel momento che il progettista cerca di capire quali funzionalità implementare e quali invece sarebbero superflue per un eventuale cliente. Una funzionalità assume importanza se le viene trovato un contesto utile nel momento in cui l'applicazione viene utilizzata.

Il documento di tesi è strutturato come segue:

- Primo capitolo: nel primo capitolo che compone questo documento di tesi verranno introdotti i concetti alla base del sistema e verrà presa in analisi l'idea che ha fatto nascere il progetto; si parlerà quindi di realtà aumentata e del concetto di industrial design, cercando di focalizzare l'attenzione sugli argomenti principali che hanno permesso di ideare e sviluppare l'applicazione presentata in questo volume di tesi.
- Secondo capitolo: nella seconda sezione del documento preso in considerazione verrà illustrato ed analizzato il processo di progettazione dell'applicazione, presentando i mockup realizzati e le scelte implementative adottate per realizzare l'interfaccia utente. Infine, verranno prese in analisi le tecnologie utilizzate per realizzare l'intero sistema, spiegando i motivi per i quali sono state adottate, i loro vantaggi e svantaggi, e quale ruolo svolgono all'interno del progetto.

- Terzo capitolo: nell'ultimo capitolo verrà illustrato l'intero sistema sviluppato, analizzando le sue componenti principali, le varie funzionalità che espone e le interfacce utente che offre. Inoltre, verranno presi in considerazione alcuni degli algoritmi più importanti che si sono rilevati di grande importanza all'interno del progetto, indicandone i punti di forza e piccoli frammenti di codice. Le interfacce invece verranno analizzate e discusse con l'aiuto di screenshot presi direttamente dall'applicazione in esecuzione. In fondo a questa sezione, verrà presentata una parte relativa alle funzionalità aggiuntive che l'applicazione potrebbe accogliere se soggetta a sviluppi futuri

1 Concetti Fondamentali

In questo capitolo verranno introdotti e commentati i principali concetti sui quali si basa il progetto sviluppato, del quale verranno poi anche presentate le finalità e le funzionalità più importanti. In particolar modo verranno descritti i concetti di:

- Realtà aumentata, cercando di capire cosa tratta, come si è sviluppata nel tempo e quali sono i suoi impieghi di maggior interesse.
- Industrial design, in termini di tutto ciò che riguarda il miglioramento estetico e funzionale dei prodotti che un'azienda propone a fini di mercato.

In fondo al capitolo verrà poi introdotta l'idea principale di progetto, spiegandone la nascita, le principali funzionalità che deve offrire e una valutazione degli aspetti positivi o negativi dell'applicazione.

1.1 Realtà Aumentata

1.1.1 Che cos'è la realtà aumentata

Quando si parla di realtà aumentata (Augmented Reality - AR), si parla di una tecnologia di alto livello che permette, tramite l'utilizzo di particolari strumenti digitali, di interagire con l'ambiente esterno. La realtà aumentata perciò sfrutta elementi già presenti nel mondo reale e, attraverso l'interazione con l'ambiente circostante, è in grado di ottenere una conoscenza più approfondita e specifica di determinati elementi. Grazie a un servizio di tale tipo, questa tecnologia permette di incrementare e migliorare, tramite l'interazione uomo-tecnologia, una serie di azioni che vengono regolarmente svolte nella quotidianità. Va sottolineato il fatto che la realtà aumentata garantisce un arricchimento delle percezioni grazie a elaborazioni digitali dei dati tramite software, che consentono di ottenere informazioni che non possono altrimenti essere ottenute se si fa uso solo dei cinque sensi che l'uomo possiede [DIB05].

Quindi, per riassumere, si parla di realtà aumentata quando degli oggetti virtuali vengono contestualizzati in un ambiente reale, e l'utente può interagire con essi mediante l'utilizzo di dispositivi adeguati.

Bisogna specificare inoltre, che quando si parla di realtà aumentata si possono intendere due differenti tipologie di tecnologia:

- Mobile: riguarda l'esperienza vissuta tramite dispositivi mobili come smartphone o tablet, che poiché dotati di fotocamera, GPS, bussola, accelerometro e altri sensori, possono essere utilizzati per rilevare dati e informazioni riguardo l'ambiente circostante attraverso la l'obiettivo della fotocamera. Facendo ciò, si crea una sorta di sovrapposizione tra elementi fisici reali ed elementi virtuali digitali calcolati dal dispositivo [REA20].

- Computer: questa tipologia invece è basata su un software specifico in grado di riconoscere disegni in bianco e nero che vengono poi processati ed elaborati tramite marcatori specifici (ARtags) per la realizzazione di immagini, filmati o altri contenuti multimediali interattivi, attraverso l'elaborazione dei dati acquisiti tramite webcam [REA20].

1.1.2 Terminologie Appropriate

Poiché il concetto di realtà digitalizzata sta entrando sempre più nella quotidianità delle persone, è utile fare qualche chiarimento sulle terminologie utilizzate per descrivere approcci diversi in cui essa viene utilizzata. Le principali tecnologie di realtà digitalizzata trovano il loro fondamento teorico nel *virtuality continuum* (Figura 1.1), una definizione della realtà mista (mixed reality) attraverso la scala lineare che lega lo spazio che intercorre tra la dimensione puramente reale e quella interamente virtuale [LAT18].



Figura 1.1: Rappresentazione del range del virtuality continuum [LAT18].

Ogni tipologia presente all'interno di questo range, rappresenta un particolare utilizzo dello stesso paradigma di realtà digitale:

- **Virtual Reality (VR)**: coincide con una dimensione interamente virtuale, che sostituisce del tutto il mondo reale. Si tratta quindi di un ambiente virtuale puro (virtual environment) dove l'utente vive un'esperienza in un mondo creato dal computer, abbandonando qualsiasi concetto legato alla realtà vera. Indossando un visore, l'utente è catapultato in un mondo totalmente digitale, dove è possibile interagire con un ambiente 3D.

- **Augmented Reality (AR):** consiste nell’aggiungere informazioni digitali all’ambiente reale (Real Environment). I dispositivi AR più diffusi sono smartphone e tablet; mentre le applicazioni AR più diffuse sono quelle che hanno avuto più successo in ambito consumer, come ad esempio Pokemon GO.
- **Augmented Virtuality (AV):** analoga alla realtà aumentata, con la caratteristica di rendere migliore l’interazione con contenuti virtuali piuttosto che col mondo reale.
- **Mixed Reality (MR):** termine utilizzato quando sono presenti sia realtà virtuale che aumentata nella stessa applicazione. È la fusione tra realtà fisica e quella virtuale, sovrapponendole. È possibile quindi osservare il mondo reale che ci circonda traendone informazioni utili, ma anche interagire con oggetti virtuali come fossero reali.
- **Extended Reality (XR):** definizione di carattere generico in cui vengono fatte convergere tutte le tecnologie della realtà, in una prospettiva di interazione tra uomo e contenuti digitalmente aggiunti a una realtà che diventa “estesa”.

1.1.3 Sviluppo nel Tempo

Si potrebbe pensare che, data la sua continua evoluzione e innovazione, la realtà aumentata abbia origini recenti ma in realtà la sua storia inizia nella seconda metà del 1900.

Il primo esempio di utilizzo di realtà aumentata, anche se non come la intendiamo ai giorni nostri, risale al 1962, quando Morton Heilig brevettò una macchina chiamata Sensorama, illustrata in Figura 1.2. Si tratta di un simulatore con capienza da una a quattro persone che fornisce l’illusione di una realtà cinematografica diversa attraverso l’odore, suono stereo, vibrazioni del sedile e vento nei capelli per creare l’illusione di essere fisicamente presenti nel luogo della scena, dando vita al concetto di coinvolgimento multisensoriale tecnologico. Morton Heilig perciò utilizzò la tecnologia per far comprendere come lo spettatore utilizzi tutti i sensi in maniera efficace e interattiva, anche se in maniera del tutto passiva poiché si trova davanti a un semplice schermo 3D, ma percettivamente è come se eseguisse l’attività realmente. Principio che qualche anno dopo, descrive esattamente il concetto di realtà aumentata [BAT19].



Figura 1.2: Sensorama [BA110].

Nel 1966, grazie all'idea del professor Ivan Sutherland, nacque il primo modello di uno degli apparati tecnologici cardine della realtà aumentata: l'head-mounted display (HMD), un dispositivo di visualizzazione da indossare sulla testa, dotato di un display ottico monoculare o binoculare.

Il 1975 è un anno di innovazione molto importante per lo sviluppo di questa nuova tecnologia, poiché nasce Videoplace, che per la prima volta in assoluto, per merito di Myron Krueger, permette agli utenti di interagire con gli oggetti virtuali, senza indossare occhiali o visori. Tale prodotto mirava essenzialmente a mettere in collegamento due o più individui tra loro utilizzando il computer come mezzo di contatto. Attraverso l'impiego di uno schermo posteriore retroilluminato, la sagoma dell'utente veniva scontornata, digitalizzata e poi analizzata da processori specializzati, che potevano esaminare la postura, un singolo movimento e le interazioni con gli oggetti virtuali del sistema [BAT19].

Trattandosi solo di brevetti, tali progetti non comportavano solidi finanziamenti per permettere ai creatori di portare avanti le proprie idee supportati economicamente, fino al 1989 quando Jaron Lanier monetizza per la realtà virtuale e fonda la prima azienda commerciale attorno ai mondi virtuali. L'anno successivo viene coniato il termine Augmented Reality da Tom Caudelle mentre forniva aiuto ai lavoratori della Boeing che assemblavano i cavi a bordo degli aeromobili [BAT19].

Dopo circa un decennio di studi riguardanti il campo della realtà aumentata, nel 2002 Bruce Thomas inventa il primo gioco con funzionalità di realtà aumentata, ARQuake, che oltre al successo ottenuto, segna l'inizio di una evoluzione ancora oggi in continuo sviluppo.

Dopo anni di innovazioni tecnologiche e un susseguirsi di tecnologie sempre più all'avanguardia, nel 2011, anno dell'AR inserita in prodotti dedicati, emerge Webcam Social Shopper, in cui, attraverso la videocamera del pc, lo schermo del computer diventa uno specchio digitale, dando un'idea di come un certo abito o vestito possa stare addosso all'utente che lo utilizza [BAT19]. In Figura 1.3 è possibile osservare un'interfaccia grafica del programma in questione.

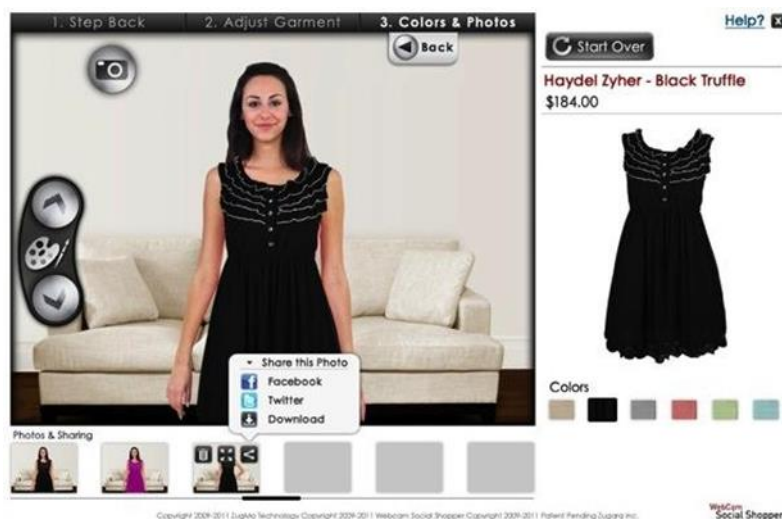


Figura 1.3: Webcam Social Shopper [ZUG20].

Ormai, nelle applicazioni odierne, siamo abituati a vedere situazioni già pronte e funzionanti, ma quanti anni di studio e ricerca ci sono dietro a tutto questo! La realtà aumentata è un mondo in continua espansione ed evoluzione, quindi non ci si può mai fermare alla novità ma bisogna sempre calarsi in un contesto di continuo cambiamento e miglioramento delle tecnologie già esistenti.

1.1.4 Location Based, Marker, Markerless AR

Si parla di realtà aumentata geo-localizzata (LBAR - Location Based Augmented Reality), quando, grazie ai sensori presenti nel dispositivo, è possibile ricavare informazioni utili per l'applicazione. Lo smartphone di ultima generazione infatti deve essere dotato di un sistema di posizionamento globale (GPS) e di un magnetometro (bussola e accelerometro). Inoltre, deve consentire la visualizzazione di un flusso video in tempo reale e deve essere dotato di collegamento Internet per ricevere dati. Inquadrando l'ambiente circostante con un dispositivo fornito di fotocamera, viene visualizzato in sovrapposizione l'elemento 3D e fissato nel mondo reale tramite un software che sfrutta i punti di interesse ricavati. In sostanza, una tecnica di visualizzazione interattiva per dati digitali che sovrappone informazioni digitalizzate su luoghi o oggetti fisici come l'AR, combinata con le capacità di consapevolezza della posizione geografica di un dispositivo cellulare è chiamata realtà aumentata basata sulla posizione.



Figura 1.4: Wikitude, applicazione che sfrutta AR geolocalizzata [KOH17].

La realtà aumentata a riconoscimento marker (*marker based AR*), invece, è basata sull'utilizzo di marcatori come tags, immagini, foto. Tali marcatori vengono mostrati alla camera del dispositivo, sono riconosciuti dal dispositivo e vengono sovrapposti in tempo reale ai contenuti multimediali generati (audio, video, oggetti 3D, ecc.). Per far in modo che questa tecnologia sia efficace e funzioni nel migliore dei modi, il marker deve essere facilmente riconoscibile dalla fotocamera, cioè deve possedere un buon numero di punti visivi unici.

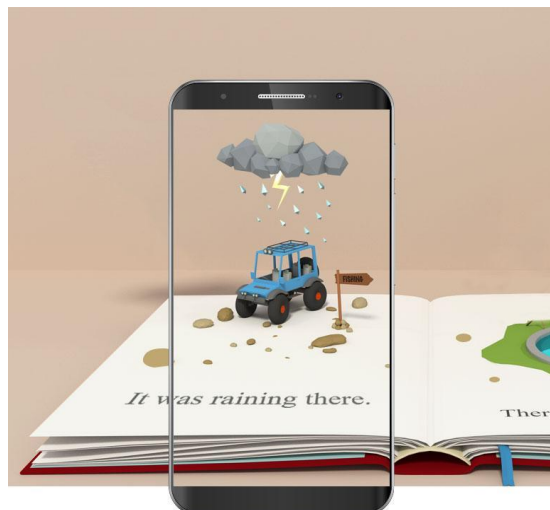


Figura 1.5: Applicazione AR che riconosce marker specifici [THE20].

Infine, quando parliamo di *markerless AR*, intendiamo una tecnologia di sviluppo e creazione di oggetti virtuali in ambiente reale basata su algoritmi di computer vision, che associano il riconoscimento dell'ambiente che viene ripreso dalla camera (Image Recognition) attraverso una localizzazione geografica e una mappatura digitale simultanea. Inoltre, come suggerisce il nome, permette di inserire contenuti digitali 3D senza l'utilizzo di marcatori specifici, quindi non necessita di una conoscenza preliminare dell'ambiente per sovrapporre il contenuto 3D a una scena reale e mantenerlo in un punto fisso nello spazio.



Figura 1.6: Pokemon GO, applicazione che sfrutta AR markerless [LAH18].

1.1.5 Contesti di Utilizzo

La realtà aumentata ha portato con sé, oltre che a una gran innovazione tecnologica, anche un cambiamento radicale nella quotidianità delle persone. Ormai viviamo in un mondo che è in continuo cambiamento tecnologico, e si ha sempre più l'esigenza di affidarsi ad un qualcosa di sicuro e preciso come un sistema informatico. Per questo motivo la realtà aumentata non è impiegata soltanto per scopi intrattenitivi, ma viene anche, e soprattutto, utilizzata in contesti ben più culturali e specifici, se non economici e produttivi. Di seguito vengono riportati i principali contesti di utilizzo in cui l'AR svolge un ruolo fondamentale:

- Medicina: gli studenti di medicina utilizzano modelli in realtà aumentata 3D per l'apprendimento, e svolgono vere e proprie simulazioni d'intervento per allenarsi con le procedure chirurgiche. In alcuni ambulatori, medici connessi in remoto possono assistere in tempo reale alle operazioni dalla visuale del chirurgo operante. Per esempio, gli scanner in realtà aumentata, aiutano a proiettare sul corpo del paziente in maniera dettagliata tutte le ramificazioni delle vene, in modo da facilitare i prelievi del sangue [NIB20].



Figura 1.7: Un medico utilizza un visore con realtà aumentata [CLE19].

- Marketing: grazie all'impiego della realtà aumentata nei loro prodotti, il tipo di comunicazione delle aziende passa da statico a dinamico grazie a volantini, opuscoli, cataloghi, annunci, poster e cartelloni pubblicitari che “prendono vita” e interagiscono direttamente con il cliente [NIB20].

- Educazione e Apprendimento: anche i bambini potranno usufruire di questa nuova tecnologia, utilizzando libri interattivi che includono modelli 3D, abbinati ad azioni sonore e coinvolgenti [NIB20].



Figura 1.8: Applicazione AR per l'apprendimento infantile [MOR18].

- Giornalismo e Mass Media: i giornali più importanti del mondo stanno iniziando ad usare la realtà aumentata per migliorare la cronaca e le notizie principali. Gli argomenti su carta stampata vengono arricchiti con contenuti esclusivi e le esperienze AR stanno portando i lettori ad avere un maggior feeling sensoriale con ciò che leggono [NIB20].
- Decorazione: per chi possiede smartphone di ultima generazione sarà molto divertente provare nella propria stanza, sala, giardino o qualsiasi altro ambiente, tutti gli oggetti per l'arredamento o la personalizzazione. Gli utenti possono modificarne inoltre la dimensione, il colore, il materiale e diversi altri parametri in tempo reale, in base alle possibilità che l'applicazione offre con il suo servizio [NIB20].
- Navigazione: i sistemi di navigazione tradizionali non sono funzionali in materia di visibilità durante la guida, poiché lo schermo sul quale vengono visualizzate le indicazioni distraggono notevolmente il guidatore e possono diventare un pericolo. Per questo motivo la realtà aumentata in questo contesto permette anche di visualizzare le istruzioni direttamente nel campo visivo di chi guida, aggiungendo informazioni in tempo reale sul traffico e senza dover togliere la vista dalla strada [NIB20].



Figura 1.9: Applicazione AR per la navigazione [SUW18].

- Manutenzione: con l'utilizzo della realtà aumentata è possibile controllare che non ci siano impostazioni errate nelle macchine per la produzione e quando c'è la necessità di correggere i loro parametri, una grafica in 3D viene proiettata nell'esatta posizione del problema da risolvere. Essa può essere usata in alternativa ai tradizionali manuali di utilizzo e guide [NIB20].



Figura 1.10: Applicazione AR che permette di riconoscere le parti di un macchinario [GRE19].

- Turismo: le guide di viaggio e le mappe, vengono implementate ed estese con contenuti di realtà aumentata in modo da fornire informazioni supplementari per il turismo. Con i dispositivi dotati di supporto AR, i turisti possono facilmente trovare informazioni preziose e più dettagliate scansionando l'area circostante, visionare punti di interesse, suggerimenti per il pernottamento e altro ancora [NIB20].

- Edilizia: la realtà aumentata viene utilizzata per progettare edifici o navigare nei condotti sotterranei per agevolare gli scavi ed evitare così di incappare in qualche errore di progettazione. I costruttori stanno anche misurando lunghezza e distanza tra oggetti utilizzando le capacità di monitoraggio della realtà aumentata. Grazie al suo impiego in questo settore infatti, gli ingegneri possono anche visualizzare il progetto finito prima che inizi il processo di costruzione effettivo [NIB20].



Figura 1.11: Un architetto osserva il risultato del suo progetto tramite un'app AR [KRO18].

- Arte: oltre ad ampliare in maniera notevole l'esperienza di visita nei musei tramite visite guidate in AR che forniscono ulteriori contenuti sugli oggetti esposti, in ambito artistico si sta evolvendo sempre di più l'uso della realtà aumentata per creare opere digitali situate in luoghi pubblici che possono essere osservate solo attraverso l'utilizzo di dispositivi appositi [NIB20].

1.2 Industrial Design

Con il termine Industrial Design si intende quella specifica categoria di progettazione e sviluppo per l'industria dove al dato tecnico, alle esigenze pratiche e funzionali si unisce un elemento estetico.

Viene introdotto in questo contesto, il concetto di 'entropia della comunicazione' coniato da Oswald Wiener: si afferma cioè che il massimo dell'informazione è fornito da quel messaggio che per la sua novità e imprevedibilità procura il massimo di sorpresa.

Bisogna precisare però che quando tale messaggio viene ripetuto troppo spesso, l'efficacia dello stimolo può decrescere, annullando l'effetto di novità desiderato. Si può affermare quindi che nel caso del design industriale l'elemento della novità formale è essenziale e rappresenta le fondamenta del successo che un oggetto o un prodotto può ottenere [VET20].

L'oggetto inteso come prodotto di un processo industriale, perciò, è creato appositamente per uno specifico interesse (pratico ed estetico) che sia immediato e strettamente legato all'uso a cui l'oggetto è destinato.

Uno dei compiti più delicati e complessi del progettista, è quello dello styling dell'oggetto, ovvero la restaurazione di un determinato prodotto per renderlo nuovamente appetibile al pubblico di consumatori.

Un'altra delle caratteristiche dell'industrial design, che lo distaccano da qualsiasi altra forma artistica, è quella della dipendenza necessaria da una richiesta specifica da parte dell'industria: non può esistere un prodotto creato o inventato liberamente dal designer per sua precisa e spontanea volontà, a meno che non si tratti di un'eventuale bozza di progetto; ma anche in tal caso è evidente che il designer avrà agito adottando una visione improntata al futuro in vista di una possibile realizzazione del prodotto nel mondo dell'industria. Ecco dunque come la novità del prodotto dovrà necessariamente adattarsi ad alcune esigenze del mercato e della produzione [VET20].

Il designer, quindi, è innanzitutto un progettista, ma è anche un pianificatore. Uno dei suoi primi obiettivi è quello di raggruppare, sintetizzare e analizzare i dati che riceve da una compagine di ricercatori, traendone dei risultati che gli permettano di individuare una prima visione generale sul tipo di prodotto che deve creare. Il designer dunque, grazie alle sue conoscenze dei prodotti esistenti e delle esigenze del pubblico, può giungere alla creazione di un oggetto che corrisponda a determinate specifiche tecniche, che non siano già state immaginate o supposte degli stessi tecnici esperti nel settore in questione. Spesso è stato proprio il designer a suggerire la tipologia dei materiali o la dimensione ottimale di un certo apparecchio. È grazie al designer dunque se si arriva ad una differente soluzione tecnica tramite la fusione di concetti distanti tra loro che, invece, possono portare alla nascita di una nuova funzione a cui il prodotto è dedicato [VET20].

È di estrema importanza che il prodotto finale sia immediatamente comprensibile al pubblico che ancora non lo conosce, che non sia mascherato con vesti non proprie, che non possa essere scambiato con un vecchio prodotto simile ma non identico, che abbia quindi sin dall'inizio una sua personalità ben distinta. È questa uno dei ruoli i più delicati e rischiosi del designer, perché il massimo requisito che senza dubbio deve soddisfare è proprio quello di saper imporre all'oggetto sin da subito una forma personale, che lo renda immediatamente individuabile sul mercato e riconoscibile tra i mille prodotti analoghi che lo circondano [VET20].

1.2.1 IKEA Place

Un esempio di applicazione che senza dubbio rispetta i canoni di vendita e sviluppo sopra citati, è Ikea Place. Con uno strumento simile, ogni persona non solo può vedere immediatamente un prodotto davanti a sé, ma capisce all'istante come starebbe all'interno della propria abitazione; tutto questo senza recarsi in nessun punto vendita ma utilizzando soltanto un dispositivo mobile. Ikea Place quindi, sfrutta a pieno i concetti dell'industria del design, aiutando chi desidera acquistare un divano piuttosto che una poltrona, a rendersi conto di come il design del prodotto che vuole portare in casa, riesca a trasformare il proprio ambiente.

L'innovazione portata da questo tipo di tecnologia, è stata quella di mettersi a completa disposizione del cliente, poiché provando loro stessi a casa loro il prodotto che vogliono acquistare, arrivano in negozio con più consapevolezza e sicurezza sull'acquisto da effettuare, riducendo in maniera notevole quei resi dovuti ad un eventuale errore di valutazione di un prodotto all'interno del proprio arredo. Capire l'aspetto di un prodotto prima di averlo acquistato è un'occasione davvero innovativa [ZUC17].

Prendendo spunto da questo prodotto di successo, si può capire come il mercato sia aperto allo sviluppo di nuove app che risolvano problemi pratici, obiettivo principale di chi si muove all'interno del mondo del design del prodotto industriale, e che le app di vendita comportano un vantaggio competitivo notevole a chi le produce, sia in termini utili per aver più controllo dei prodotti che vende l'azienda, sia in termini di reputazione e fiducia da parte del cliente che acquista. Insomma, Ikea ha fatto centro: ha avuto la lungimiranza di risolvere un problema comune dei propri clienti, prima ancora che

diventasse effettivamente un ostacolo alla vendita dei suoi prodotti, e soprattutto dopo aver valutato e analizzato con attenzione i propri dati, le richieste, i suggerimenti e le eventuali lamentele [CAZ20].

Per ottenere successo nel business odierno, le aziende devono mettere il cliente al centro delle proprie strategie di mercato aziendali, cercando di anticipare il più possibile le sue esigenze e mettendo a sua disposizione strumenti sempre più all'avanguardia. Creare esperienze di acquisto personalizzate, in cui il cliente può muoversi in completa libertà e in base alle proprie preferenze, che combinino le sensazioni di un normale acquisto in negozio con la praticità dello shopping, è un'idea vincente [CAZ20].



Figura 1.12: Ikea Place, app di vendita di prodotti per arredamenti [LIM18].

1.3 Idea Progettuale

Il progetto è volto a creare un'applicazione per smartphone con una componente in realtà aumentata che permetta ad un'eventuale azienda che vende e installa superfici per pavimenti, di mostrare in tempo reale al cliente il risultato finale del lavoro direttamente sul pavimento di casa sua.

In questo paragrafo verrà illustrata l'idea che ha dato vita al progetto e verrà spiegato quali sono le funzioni principali che dovrà svolgere l'applicazione.

1.3.1 L'idea nasce dall'esigenza

L'idea di un'applicazione simile nasce dall'esigenza di un'eventuale azienda di soddisfare le richieste di clienti specifici che vogliono comprare superfici per pavimenti. Molte aziende che vendono prodotti, si trovano in difficoltà con clienti privati poiché spesso chi compra il prodotto vuole avere un'idea di come sarebbe il risultato a fine lavoro. Le persone a volte fanno fatica ad immaginarsi una realtà diversa da quella a cui sono abituate quotidianamente, e per un venditore di superfici per pavimenti può essere difficile convincere un cliente a comprare il suo prodotto se quest'ultimo non può sapere come effettivamente sarà il risultato. A questo proposito emerge l'idea di coinvolgere la persona che compra il pavimento in una realtà diversa da quella a cui è abituato ma che allo stesso tempo lo aiuti a capire se acquistare o meno il prodotto. Chi vende la superficie per il pavimento infatti, può recarsi direttamente a casa del cliente e mostrare il nuovo pavimento in tempo reale attraverso l'utilizzo del proprio smartphone.

La necessità di chi compra, di vedere in anteprima il risultato di un lavoro ma non può farlo per motivi tecnici (come può essere il cambio piastrelle di un pavimento), stimola in chi vende tale prodotto, un senso di ricerca della soluzione che lo porta a trovare strumenti sempre più innovativi ed efficaci per soddisfare qualsiasi richiesta del cliente. Il concetto alla base del design industriale è proprio questo: mettersi nei panni di chi potrebbe aver bisogno di uno specifico strumento e sviluppare un mezzo specializzato capace di soddisfare tali richieste, anticipando i tempi di richiesta e coinvolgendo in modo totalitario il cliente che deve acquistare il prodotto.

1.3.2 Funzionalità Basilari

Trattandosi di un software dedito alla vendita di un prodotto, l'applicazione non deve essere complessa ma deve disporre di funzionalità semplici ed efficaci, in modo da essere facile da usare per chi vende e performante nei risultati visivi per chi compra.

In particolare, inquadrando verso il basso con lo smartphone, per prima cosa deve essere in grado di riconoscere la superficie del pavimento reale e, una volta ottenuto il tracking, l'utente può posizionare nell'ambiente dei punti di interesse sfruttando un

apposito mirino presente sull'interfaccia grafica. Questi punti verranno uniti da linee che si aggiornano in tempo reale in base all'ordine in cui l'utente posiziona i punti nel mondo reale, e non appena le linee generate formano il perimetro desiderato, alla pressione di un apposito pulsante verrà generato un pavimento virtuale ma visibile come se fosse presente realmente.

Una volta che l'utente ha creato il pavimento, ci si può girare intorno, avvicinare o allontanare, cambiare il tipo di piastrelle o superficie, cambiare i riflessi di luce per simulare un ambiente reale e ruotare il senso delle piastrelle per adattarle a superfici già presenti nella realtà.

1.3.3 Considerazioni Progettuali

Come per ogni tipo di software che si sviluppi, vanno valutati in fase di progettazione anche tutti quegli aspetti che potrebbero incidere sulle prestazioni dell'applicazione in maniera negativa. Poiché si tratta di un'applicazione che sfrutta la realtà aumentata attraverso la fotocamera dello smartphone, può capitare che il tracking delle superfici non sia del tutto preciso a volte. Ad esempio, può succedere che un determinato piano non venga rilevato alla giusta altezza: infatti, generando un pavimento su una superficie rilevata ad un'altezza superiore rispetto a quella reale, l'utente vedrà il pavimento sospeso nell'aria, il che non rispetta la realtà che il cliente si aspettava, poiché avvicinando lo smartphone al suolo, il pavimento virtuale deve rimanere ancorato a terra permettendo una realistica prospettiva.

Un altro caso di tracking errato può essere dovuto alla scarsa luminosità dell'ambiente, oppure a una grafica delle superfici reali troppo uniformi e senza punti di riconoscimento che quindi il tracciamento non riesce a riconoscere come tali. In questi casi può essere che una volta generato il pavimento, muovendo lo smartphone intorno all'ambiente, si può verificare un leggero scostamento della posizione del pavimento rispetto a quella nella quale è stato generato, poiché il software non riesce a calcolare i giusti punti di ancoraggio non riuscendo a riconoscere le superfici.

Tutti questi possibili errori che rendono l'applicazione poco efficiente possono essere evitati con un tracking preciso dell'ambiente: infatti, ad applicazione lanciata, è buona norma inquadrare l'ambiente circostante per un buon numero di secondi prima di

agganciare i punti di interesse, in modo da permettere al software di eseguire i suoi calcoli in modo corretto.

2 Progettazione del Sistema

In questo capitolo verrà presa in considerazione la progettazione dell'intero sistema, suddividendo tale analisi in due parti: la prima sezione tratterà del design dell'applicazione, attraverso l'utilizzo di alcuni *mockup* per avere una prima idea di come l'interfaccia utente verrà implementata. Mentre nella seconda parte si parlerà delle tecnologie che sono state utilizzate per lo sviluppo dell'applicazione, cercando di illustrarne i principali punti di forza e i motivi per i quali sono state scelte come pilastri per la realizzazione dell'applicativo.

2.1 Design

In questo paragrafo si analizzerà il processo di design dell'applicazione, facendo sempre riferimento ad un contesto di marketing e design industriale.

Nelle due sottosezioni successive verranno presi in considerazione i mockup delle schermate utente, ovvero una prima proposta dell'interfaccia grafica che l'applicazione proporrà al cliente una volta che sarà in commercio e, successivamente,

verranno analizzate quali possano essere i requisiti di cui un eventuale cliente potrebbe aver bisogno utilizzando l’applicazione.

2.1.1 Mockup

I mockup delle schermate utente sono stati realizzati tramite l’ausilio del sito web *mediamodifier.com*, uno strumento online dedicato alla creazione di layout per la creazione di elementi visivi di marketing stimolanti [MED20].

Come viene illustrato successivamente, le schermate dell’interfaccia utente non sono numerose, ma sono state realizzate utilizzando un approccio funzionale piuttosto che estetico. È essenziale dare priorità ad un aspetto funzionale piuttosto che grafico in un contesto simile, poiché l’applicazione è pensata per essere efficace in ciò per cui è stata progettata. Ovviamente anche una buona presentazione dell’interfaccia è essenziale per un buon utilizzo del prodotto da parte dell’utente, che deve essere semplice da comprendere e facile da utilizzare.

Di seguito verranno presentate le schermate che presenta l’applicazione:

- *Schermata principale*: questa è la prima interfaccia che l’utente vede una volta lanciata l’applicazione. È qui che l’utente può avviare il processo di tracking dell’ambiente circostante e far sì che l’applicazione cominci a selezionare i punti di ancoraggio per il riconoscimento delle superfici. Questa schermata non presenta un’interfaccia troppo complessa, ma ha tutto ciò di cui l’utente necessita per creare il pavimento, il tutto tramite l’utilizzo di elementi intuitivi presenti sullo schermo.

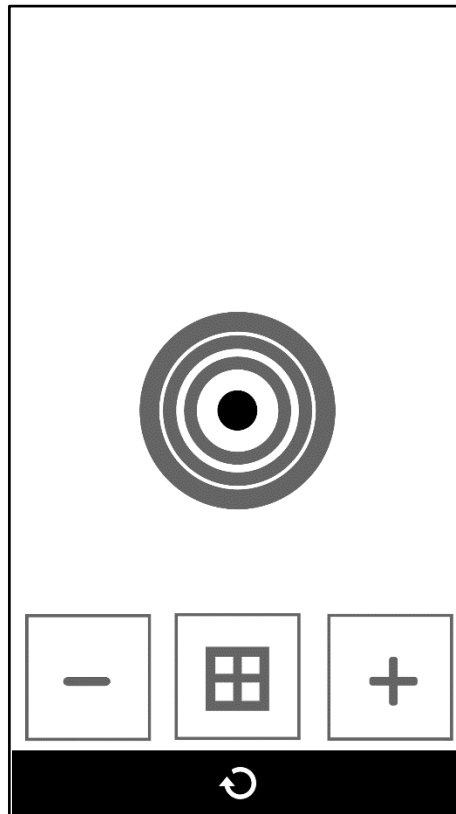


Figura 2.1: Mockup relativo alla schermata principale dell'applicazione.

- *Schermata visualizzazione:* questa è l'interfaccia che si presenta all'utente appena viene creato il pavimento. Anche in questo caso si è scelto di rispettare un approccio funzionale piuttosto che estetico per la schermata, poiché le funzionalità disponibili all'utente sono poche, ma chi utilizza il prodotto deve subito capire cosa può o non può fare con tale interfaccia.

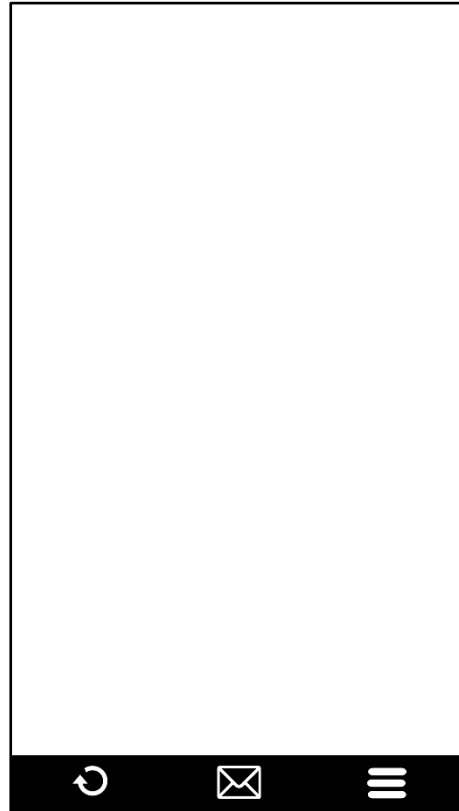


Figura 2.2: Mockup relativo alla schermata di visualizzazione del pavimento.

- *Schermata modifica*: questa è la schermata più importante in fase di visualizzazione da parte del cliente poiché, tramite questa interfaccia, l'utente può modificare il pavimento creato in base a parametri specifici e osservare se le modifiche effettuate possono produrre o meno il risultato atteso.

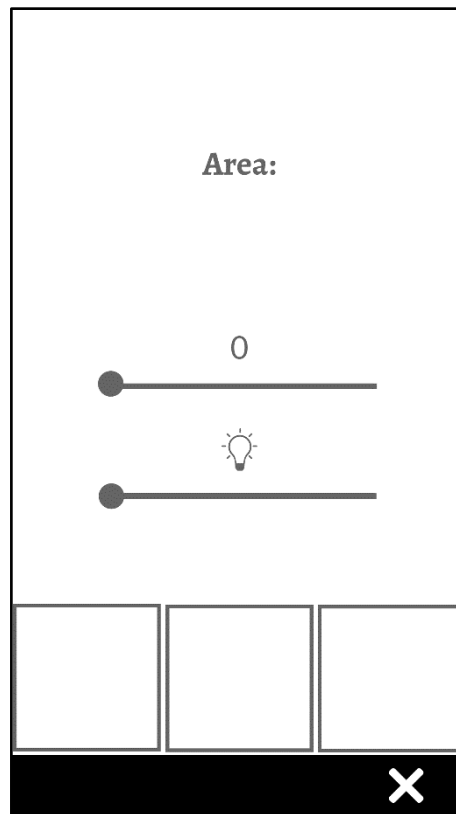


Figura 2.3: Mockup relativo alla schermata di modifica dei parametri del pavimento creato.

Se a primo impatto l'interfaccia dell'applicazione può sembrare semplice, in realtà è stata studiata per essere funzionale e intuitiva, in modo da catturare l'attenzione dell'utente finale e contemporaneamente metterlo a proprio agio.

Le schermate pensate ovviamente potranno subire leggere variazioni grafiche in base all'esigenze del cliente o a sviluppi futuri, ma comunque si è cercato di mantenere il focus sull'efficienza e la funzionalità.

2.1.2 User Group

Poiché si tratta di un'applicazione destinata alla vendita, contestualizzandola sempre in ambito del prodotto industriale, non è servito organizzare un focus group per capire quali possano essere le richieste degli utenti, cosa vorrebbero migliorare, quali funzionalità aggiungere, oppure quali interfacce rendere più funzionali e intuitive migliorandone la grafica. Piuttosto è servito calarsi in un contesto orientato al cliente, chiedendosi cioè: chi userà l'applicazione, di quali funzionalità potrà usufruire in maniera intuitiva e veloce? Cosa può servire a chi vende un pavimento e che cosa invece è

superfluo alla vendita? Chi potrebbe acquistare un prodotto simile e come fare in modo che il cliente sia sempre soddisfatto?

Tali quesiti, sono tutti capisaldi del progettista che sviluppa un software destinato alla vendita. Senza queste domande, il prodotto non può avere successo poiché non sarebbe stato analizzato il contesto in cui potrebbe tornare utile.

Per concludere, sviluppando un'applicazione destinata al commercio, chi progetta il software deve immaginarsi che cosa il cliente può richiedere fin da subito e cercare di anticipare tutte le funzionalità essenziali che servono a chi utilizza il prodotto. Per creare un'applicazione a scopo ludico, di solito, chi progetta il software, prima di iniziare lo sviluppo vero e proprio cerca proposte e feedback da un limitato numero di persone che secondo lui possono utilizzare l'app una volta terminata, mentre in quest'altro caso il processo è esattamente l'inverso: il focus group di persone che dovrebbero recensire e valutare l'applicazione viene decontestualizzato ed è il progettista a immaginarsi come il suo prodotto potrà essere utilizzato e quali funzionalità deve garantire per creare interesse da parte del compratore.

2.2 Tecnologie Utilizzate

In questa sezione verranno analizzate le tecnologie utilizzate per la realizzazione dell'applicazione, ovvero:

- Unity: motore grafico per lo sviluppo utilizzato in ambito gaming per realizzare applicazioni ad alto livello.
- AR Foundation: interfaccia che espone delle API (Application Programming Interface) che permettono di gestire funzionalità di realtà aumentata a prescindere dalla piattaforma che si sta utilizzando.

2.2.1 Unity

Unity è un motore grafico che permette lo sviluppo di videogiochi e altri contenuti interattivi, come visualizzazioni architettoniche o animazioni 3D in tempo reale. Consente di esportare i progetti sviluppati su più piattaforme, desktop, Web, e dispositivi portatili.

Unity nasce con l'intento di semplificare lo sviluppo applicativo, fornendo all'utente un editor visuale che consente di lavorare sugli oggetti di gioco in maniera più interattiva, e la logica dei comportamenti degli oggetti presenti in scena può essere scritta in C#, linguaggio di programmazione orientato agli oggetti facile da utilizzare, flessibile e completo. In altre parole, si tratta di un insieme di strumenti completo per la creazione di contenuti interattivi, semplificando il processo di sviluppo e rendendolo più veloce, soprattutto grazie alla finestra di anteprima del gioco, che mostra in tempo reale le modifiche apportate al progetto durante le operazioni di programmazione [HTM20].

I punti di forza di Unity sono senza dubbio estendibilità e flessibilità: infatti viene definito “multiplatforma” proprio per la sua capacità di sviluppare il codice una sola volta e poi, integrando poche istruzioni specifiche, compilare il progetto per sistemi operativi differenti. Il programmatore deve solo preoccuparsi di far funzionare il suo programma nella maniera corretta, e sarà Unity che in fase di compilazione riuscirà a “tradurre” i metodi utilizzati in modo da renderli compatibili con la piattaforma bersaglio; utilizzando le stesse API quindi, è possibile sviluppare lo stesso progetto per più piattaforme [INN14].

Si è scelto di utilizzare Unity come ambiente di sviluppo, poiché fornisce alcune funzionalità chiave per la creazione di un'applicazione in un contesto industriale, che andremo ad analizzare:

- Modellazione 3D: oltre che a tutto il contesto logico-implementativo, Unity fornisce anche un aiuto sostanziale allo sviluppatore per quanto riguarda la modellazione 3D. Infatti, il progettista, può generare primitive 3D standard direttamente da editor grafico del programma, o importare oggetti creati tramite l'utilizzo di altri software. Una volta importate le geometrie nel progetto, Unity ci permette di agire su alcune caratteristiche come posizione, scala, rotazione, materiale, texture, luci o ombre [TUT20].

- Gerarchie di Oggetti: Unity mette a disposizione dello sviluppatore l'utilizzo di gerarchie di oggetti attraverso l'approccio *parent-children*. Tale approccio permette quindi al progettista di avere pieno controllo su quei componenti complessi che altrimenti sarebbero difficili da gestire.
- Classe MonoBehaviour: è la classe base da cui deriva ogni script C# di Unity. È fondamentale che ogni script che descriva il comportamento di ogni oggetto in scena, faccia riferimento a tale classe poiché offre alcune funzioni per il ciclo di vita degli oggetti (Figura 2.4). Infatti, ogni oggetto in scena possiede di default due metodi principali che questa classe espone, Start e Update; il primo viene richiamato tramite call-back al momento della creazione dell'oggetto, mentre il secondo viene richiamato ad ogni frame di gioco per aggiornare il suo stato e il suo comportamento, al verificarsi di certe condizioni o eventi.

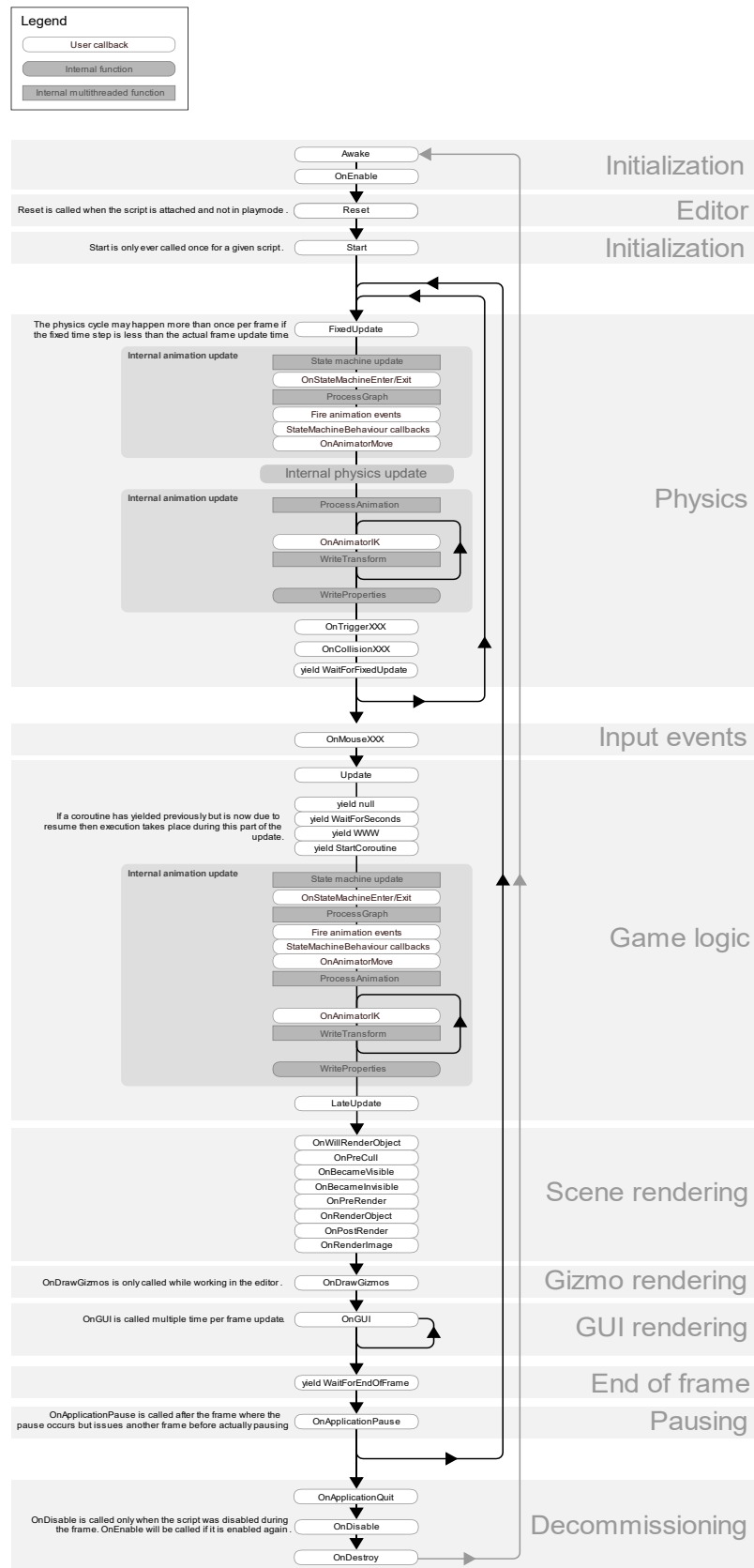


Figura 2.4: Diagramma di flusso del ciclo di vita degli oggetti della classe MonoBehaviour [MON20].

- **Multiplatforma:** aspetto fondamentale per il quale si è scelto Unity come strumento di sviluppo, è, come già accennato, la sua capacità di essere versatile in base al contesto di progettazione. Grazie a Unity infatti, è stato possibile creare un software universale, utilizzando API standard, e poi, in fase di compilazione, specificare la piattaforma di destinazione in modo da permettere a Unity di generare l’eseguibile corretto che rispetti i requisiti di tale dispositivo. L’applicazione industriale deve essere sviluppata per funzionare su più piattaforme, in modo da ricoprire più target possibili ed essere competitiva sul mercato, altrimenti, focalizzarsi sullo sviluppo per una piattaforma specifica, significherebbe per l’azienda perdita di eventuali clienti e di conseguenza riduzione della vendita del proprio prodotto.
- **Interfaccia:** grazie al supporto grafico fornito dall’IDE, è stato possibile realizzare l’interfaccia utente direttamente da editor tramite l’utilizzo di un componente chiamato Canvas, sul quale è possibile aggiungere gli elementi 2D che fanno parte delle schermate dell’applicazione quali pulsanti, slider o pannelli di testo. Una volta creato l’oggetto *Canvas*, Unity genera in automatico un oggetto *EventSystem*, che si preoccupa di rilevare gli eventi di interazione con l’utente come ad esempio i touch sullo schermo. Tutto ciò che viene inserito in scena, e fa parte dell’aspetto grafico dell’applicazione, viene visto attraverso il Canvas, che rappresenta lo schermo fisico del dispositivo. Inoltre, è possibile personalizzare gli elementi grafici attraverso l’*Inspector* dell’editor (finestra visuale in cui è possibile modificare i parametri dei componenti di scena), cambiandone posizione, scala, rotazione e pivot, ossia il punto di ancoraggio sullo schermo che funge da riferimento per l’oggetto e per le sue proprietà.

2.2.2 AR Foundation

AR Foundation è un framework creato appositamente per lo sviluppo di applicazioni che utilizzano la realtà aumentata. Consente di sviluppare la propria app una volta sola, e successivamente distribuirla su più piattaforme o dispositivi mobili che supportano AR, come viene illustrato in Figura 2.5.

Questo pacchetto offre quindi un'interfaccia per gli sviluppatori, ma non implementa direttamente alcuna funzionalità AR. Per utilizzare gli appositi metodi per la gestione della realtà aumentata su un particolare dispositivo di destinazione, occorre installare dei pacchetti aggiuntivi specifici per la piattaforma bersaglio, in particolare ARCore XR Plugin per Android e ARKit XR Plugin per iOS [ARF20].

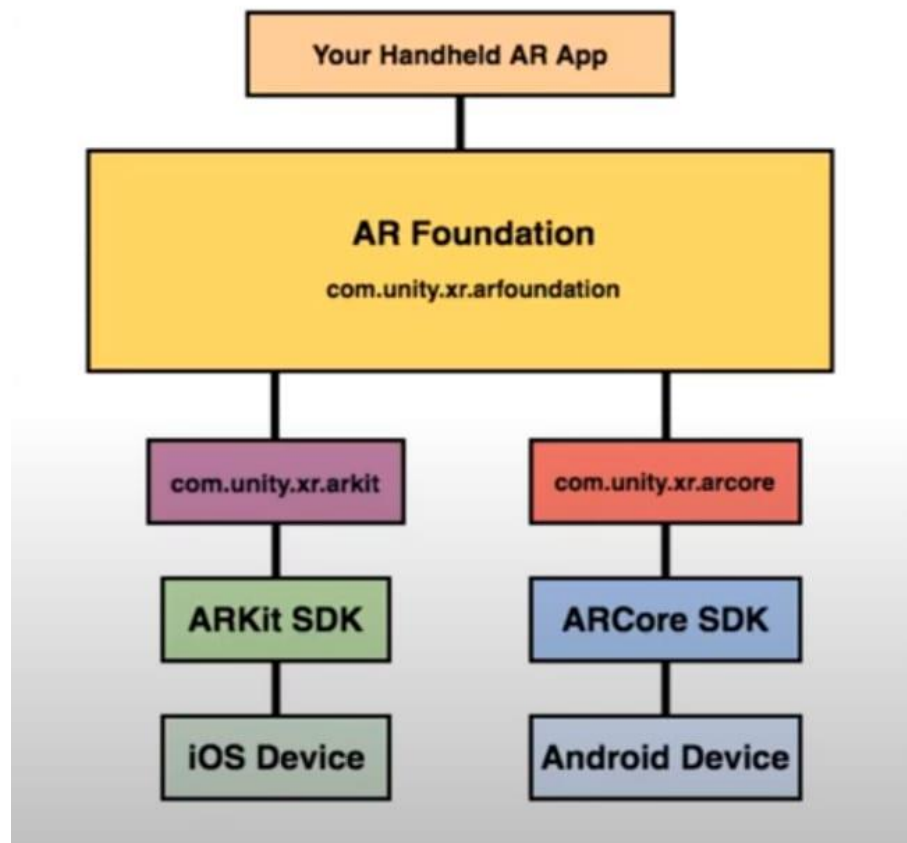


Figura 2.5: Schema esplicativo di come AR Foundation si compone [LAN20].

AR Foundation è quindi un insieme di API per gestire i dispositivi che supportano la realtà aumentata attraverso i seguenti concetti:

Device Tracking	Traccia la posizione e l'orientamento del dispositivo nello spazio fisico.
Plane Detection	Rileva superfici orizzontali e verticali.

Point Cloud	Nota anche come <i>feature points</i> , la “nuvola” di punti rappresenta l’insieme dei punti che il dispositivo riconosce nell’ambiente circostante.
Anchor	Una posizione e un orientamento arbitrari di cui il dispositivo tiene traccia.
Light Estimation	Stima della temperatura media del colore e dell’intensità di luce nello spazio fisico.
Environment Probe	Uno strumento per generare una mappa cubica con lo scopo di rappresentare una particolare area dell’ambiente fisico.
Face Tracking	Rileva e traccia i volti umani.
2D Image Tracking	Rileva e tiene traccia di immagini 2D.
3D Object Tracking	Rileva oggetti 3D presenti nell’ambiente reale.
Meshing	Genera mesh triangolari che corrispondono alle forme degli oggetti nello spazio fisico.
Body Tracking	Rappresentazioni 2D o 3D di esseri umani riconosciuti nello spazio fisico.
Collaborative Participants	Traccia la posizione e l’orientamento di altri dispositivi in un’esperienza AR condivisa.
Human Segmentation & Occlusion	Applica la distanza e la profondità agli oggetti nel mondo fisico rispetto al contenuto 3D renderizzato, ottenendo una fusione realistica tra oggetti fisici e virtuali.
Raycasting	Permette di rilevare e ricevere le proprietà di un oggetto virtuale richiesto dall’utente, selezionandolo nell’ambiente.
Pass-Through Video	Rendering ottimizzato dell’immagine della telecamera mobile sul touch screen come sfondo per i contenuti AR.
Session Management	Manipolazione automatica della configurazione a livello di piattaforma quando le funzionalità AR sono abilitate o disabilitate.

Tabella 2.1: Principali funzionalità di supporto che offre AR Foundation [ARF20].

Si è scelto di utilizzare questo framework di sviluppo, poiché dotato appunto di una forte estendibilità, perché il primo requisito che l'applicazione in questione deve possedere è quello di essere funzionante e performante su molteplici piattaforme.

In particolar modo, in fase di implementazione, sono stati di cruciale importanza i seguenti concetti per realizzare le funzionalità AR dell'applicazione:

- **Plane Detection:** in fase di tracking dell'ambiente circostante attraverso la fotocamera, tale tecnologia permette all'applicazione di riconoscere immediatamente le superfici orizzontali nelle quali è possibile istanziare un pavimento virtuale. È una fase molto delicata dell'utilizzo dell'applicativo poiché il dispositivo ha bisogno di tempo per scansionare in maniera accurata l'ambiente e garantire così una maggiore precisione nel rilevare, nel nostro caso, i piani orizzontali. Ruolo fondamentale in questo processo, è svolto dalla *points cloud*; il dispositivo individua dei punti sensibili presenti nell'ambiente, come un cambio di colore improvviso piuttosto che un cambiamento della forma di un oggetto. Successivamente fissa dei punti virtuali nelle posizioni che ha rilevato e infine, dopo aver ottenuto un buon numero di punti scansionati (*function points*), attraverso la loro analisi e grazie all'ausilio dei sensori del dispositivo, è capace di rilevare oggetti e piani presenti nel mondo fisico.
- **Raycast:** il *raycast* è un raggio virtuale ortogonale che parte dalla *main camera* e attraversa lo spazio 3D. Nel progetto in analisi, questo metodo è stato utilizzato per consentire di trovare i punti di intersezione tra il piano virtuale, individuato tramite il *plane detection*, e il raycast generato dalla telecamera. Questi punti (*anchor points*) calcolati dal dispositivo, che determinano l'area del pavimento virtuale, corrispondono a riferimenti fisici del mondo reale.

- **Anchor:** il concetto di anchor è stato utilizzato per mantenere le posizioni dei punti che l'utente ha selezionato come limiti del pavimento. Attraverso raycast, l'utente posiziona dei punti virtuali nell'ambiente fisico, e tali punti rimangono ancorati al mondo reale. L'utente quindi può anche spostare la visuale in un'altra direzione, ma quando guarderà nuovamente i punti da lui stesso istanziati, essi saranno ancora nella stessa posizione, che il software si preoccupa di calcolare ad ogni frame e di garantire la maggior accuratezza possibile. In caso di tracking dell'ambiente troppo rapido o troppo complicato poiché la luce non è quella giusta o i materiali delle superfici non sono facilmente riconoscibili dal dispositivo, anche i punti ancorati possono subire degli scostamenti portando inevitabilmente a degli errori visivi. Questi metodi SLAM (*Simultaneous Localization and Mapping*) hanno il limite di essere molto dipendenti dall'ambiente circostante e il tracking sarà influenzato positivamente o negativamente a seconda di quello che vede la camera del dispositivo mobile.
- **Session Management:** la gestione della sessione AR gioca un ruolo fondamentale nel processo di funzionamento dell'applicazione. L'intero software è stato progettato suddividendolo in stati di esecuzione ben precisi: inizializzazione e tracciamento, utilizzo e reset. La fase di reset è stata particolarmente importante poiché nel caso in cui l'utente voglia ricreare nuovamente il pavimento virtuale in un'altra area oppure voglia ricominciare una scansione precedentemente errata, l'intera sessione AR verrà resettata in modo corretto: tutti gli elementi virtuali creati in fase di calcolo come la nuvola di punti e le superfici orizzontali trovate dal dispositivo durante il tracking, verranno eliminati prima di poter iniziare nuovamente la fase di tracciamento (oltre a ripulire la memoria da eventuali oggetti creati dall'utente quali il pavimento virtuale e i punti di ancoraggio).

2.2.3 ARKit

ARKit è la piattaforma di realtà aumentata di Apple per dispositivi che possiedono un sistema operativo iOS. Consente ai progettisti quindi, di sviluppare applicazioni che interagiscono con il mondo intorno a loro utilizzando la camera e i sensori di movimento del dispositivo.

ARKit utilizza una tecnologia chiamata Visual Inertial Odometry per tracciare il mondo intorno al dispositivo, consentendo al device di capire il suo movimento all'interno di una stanza. Con tale termine, inoltre, si indica il processo con il quale un dispositivo riesce a determinare la propria posizione e il proprio orientamento analizzando le immagini provenienti dalla camera associata [TIL19].

Utilizzando la fotocamera del dispositivo iOS, gli accelerometri, il giroscopio e la consapevolezza del contesto, ARKit esegue la mappatura dell'ambiente circostante mentre il dispositivo viene spostato. La fusione dei dati dei sensori con i dati della telecamera consente una mappatura e un rilevamento della posizione estremamente accurati. Il software individua le caratteristiche visive nell'ambiente come le superfici piane o verticali, e tiene traccia del movimento insieme alle informazioni dai sensori. La fotocamera viene utilizzata anche per determinare le sorgenti luminose mediante le quali vengono illuminati gli oggetti AR. La soluzione di Apple per aumentare i dettagli e la precisione di tracking e quindi l'utilizzo della memoria è una mappa scorrevole in cui i vecchi dati scompaiono per dar spazio ai nuovi. Gli utenti possono posizionare dei marker per contrassegnare le creazioni che desiderano salvare [ROU17].



Figura 2.6: Apple ARKit [HELI7].

2.2.4 ARCore

ARCore è la piattaforma di Google per la creazione di esperienze di realtà aumentata. Utilizzando apposite API, ARCore consente al telefono di rilevare l’ambiente circostante, comprendere il mondo reale e interagire con le informazioni ottenute [JAI18].

ARCore utilizza tre funzionalità chiave per integrare i contenuti virtuali con il mondo reale osservato attraverso la fotocamera del telefono:

- Rilevamento del movimento: consente al telefono di rilevare e tenere traccia della sua posizione geografica.
- Comprensione ambientale: consente al telefono di rilevare la dimensione e la posizione di tutti i tipi di superfici orizzontali, verticali o inclinate.
- Stima della luce: consente al telefono di stimare le condizioni di illuminazione attuali dell’ambiente.

Principalmente ARCore si occupa di tracciare la posizione del dispositivo mobile mentre si muove e costruire una propria interpretazione del mondo reale. La tecnologia di tracciamento del movimento di ARCore utilizza la fotocamera del telefono per identificare punti di interesse (*cloud points*) e tenere traccia di come questi punti si spostano nel tempo. Con una combinazione del movimento di questi punti e letture dai sensori del telefono, ARCore determina sia la posizione che l’orientamento del telefono mentre si muove attraverso lo spazio [JAI18].

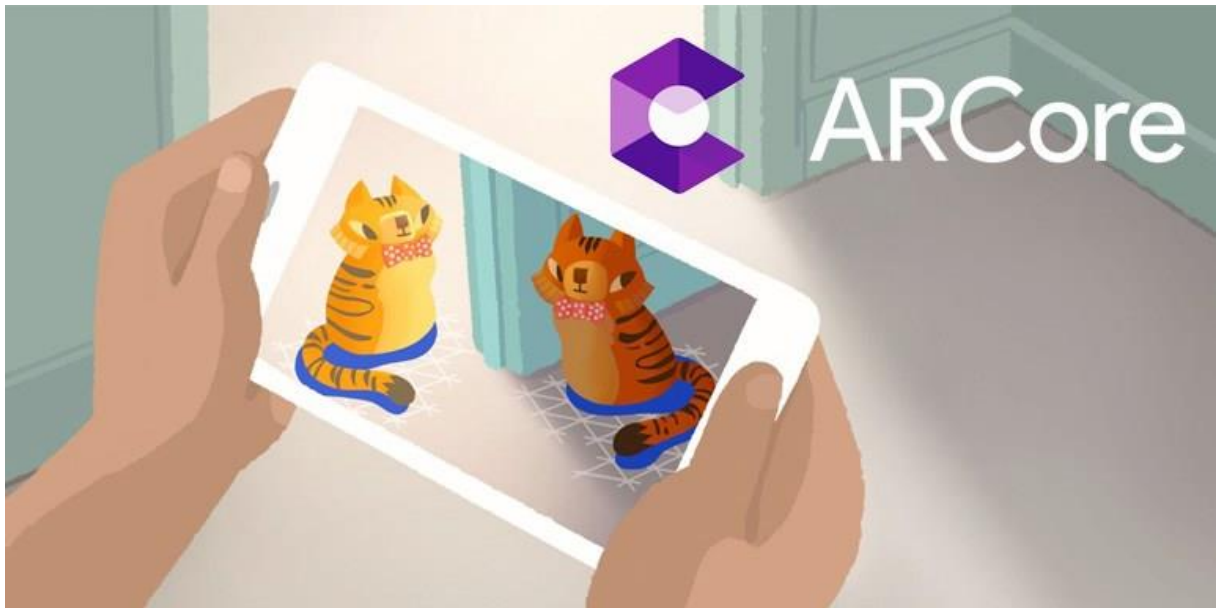


Figura 2.7: Google ARCore [TEC19].

3 App Sviluppata

In questo capitolo verrà presentato il risultato finale dell'applicazione sviluppata, spiegandone le funzionalità, le interfacce e le scelte implementative e progettuali effettuate.

In fondo al capitolo, inoltre, verrà presentata una sezione riguardante gli sviluppi futuri di tale progetto, quindi quali funzionalità potranno essere aggiunte in base alle richieste dei clienti, quali interfacce migliorare, oppure aggiornamenti del software grazie all'utilizzo di nuove tecnologie.

3.1 Scelte Implementative

In questa sezione vengono analizzate le scelte implementative effettuate in fase di sviluppo, spiegando i motivi per i quali sono state adoperate alcune soluzioni piuttosto che altre. Inoltre, in questo contesto, verranno anche descritti i principali algoritmi di funzionamento dell'applicazione.

3.1.1 Lista Materiali

Si è scelto di gestire i materiali che possono comporre il pavimento virtuale, mettendo a disposizione dell'utente una lista di texture grafiche già preimpostate all'interno dell'applicazione, che a seconda delle preferenze di chi usa l'app, potranno essere selezionate per cambiare aspetto al pavimento creato. Questa scelta deriva dal fatto che l'applicazione nasce come una demo, perciò offre al venditore un servizio puramente dimostrativo, e di conseguenza ha bisogno che le risorse di cui necessita, siano caricate in maniera veloce ed efficace (Figura 3.1) per garantire un corretto funzionamento del prodotto, inoltre, senza la necessità di possedere una connessione ad Internet.

```
var materiali = Resources.LoadAll<Material>(path);
float valoreRect = materiali.Length * ((Screen.currentResolution.width / resolutionDivision) / materialiVisibili);
materiallist = materiali.ToList<Material>();
foreach (Material materiale in materiali)
{
    GameObject buttone = Instantiate(buttonPrefab);
    buttone.transform.SetParent(pannelloMateriali.transform);
    buttone.GetComponent<ApplicaMaterial>().materialeDaApplicare = materiale;
    buttone.GetComponent<ApplicaMaterial>().ApplicaMatButton();
    buttone.transform.localScale = new Vector3(1, 1, 1);
    toggles.Add(buttone.GetComponent<Toggle>());
}
```

Figura 3.1: Caricamento dei materiali disponibili per il pavimento virtuale.

Le icone con le texture grafiche modificano a run-time il materiale che è assegnato alla *mesh* del pavimento. Questo materiale, o per meglio dire, lo *shader* utilizzato per visualizzare il materiale standard, prende il nome di “custom shader” poiché è stato appunto modificato successivamente alla selezione. La modifica dello shader base ha permesso di esporre alcune proprietà per avere il controllo della rotazione e della scala del pavimento.

3.1.2 Occlusion Walls

Gli *occlusion walls* sono stati oggetto di grande ricerca e profondo studio per risolvere un problema che si sarebbe potuto presentare in caso di creazione di un pavimento complesso: se ci si trova in una stanza dove sono presenti muri rientranti ma si vuole comunque generare un pavimento attorno ad essi, come gestire il fatto che, se l'utente si trova dietro il muro, il pavimento virtuale non deve vedersi.



Figura 3.2: Pavimento virtuale creato dietro l'angolo di un muro.

Poiché il dispositivo calcola soltanto la posizione degli oggetti virtuali ma non riconosce gli ostacoli fisici reali presenti nell'ambiente, si è dovuto ricorrere ad una soluzione semplice ma efficace; per ogni lato del pavimento, è stato innalzato un muro virtuale trasparente, formato da un materiale chiamato “*see-through*”, ovvero “guarda attraverso”. Questo materiale speciale permette di avere un effetto di occlusione di ciò

che si trova dietro di esso: sfruttando le proprietà geometriche dei piani, infatti, se la camera si trova nella direzione della normale dell'oggetto muro (vettore uscente perpendicolare al piano), allora il materiale renderizzerà a video ciò che vedrebbe la camera se quest'ultimo non ci fosse, dando l'impressione all'utente che dietro di lui non ci sia niente.

In questo modo, l'esperienza di utilizzo da parte dell'utente diventa ancora più immersiva, poiché il pavimento è creato in maniera del tutto realistica, nascondendo le parti non visibili come se fosse veramente parte del mondo fisico.



Figura 3.3: Visualizzazione del pavimento sfruttando i muri occludenti, che quindi permettono un'esperienza visiva più vicina alla realtà nascondendo il pavimento virtuale se nascosto da oggetti fisici presenti nel mondo reale.

3.1.3 Trasformazione Coordinate

La funzione di trasformazione dalle coordinate 2D sullo schermo in coordinate 3D reali nel mondo fisico è stata di cruciale importanza in questo progetto. Ha permesso, infatti, di stabilire con cura la corretta posizione in cui l'utente istanzierà i punti di ancoraggio che determineranno poi i vertici del pavimento. Attraverso tale metodo di mappatura e trasformazione delle coordinate, e grazie all'aiuto grafico di un mirino presente nell'interfaccia, l'utente decide il punto in cui istanziare il pin virtuale posizionandolo attraverso lo schermo 2D, e lo vedrà renderizzato nel mondo reale 3D rispettando le giuste proporzioni e distanze rispetto a dove si trova e a quanto lontano sta puntando la camera dello smartphone.

3.1.4 Triangolazione Superficie

Una volta posizionati nel piano 2D i vertici, si è utilizzato l'algoritmo Triangulator trovato nella documentazione di Unity, che ha permesso di creare la superficie del pavimento virtuale. Tale procedura prende in input una lista di Vector2 corrispondenti ai punti 2D di un piano e restituisce un'area geometrica bidimensionale (mesh). Questo algoritmo è stato leggermente modificato poiché i pin virtuali sono elementi tridimensionali e quindi oggetti di tipo Vector3, mentre il piano è un oggetto di tipo Mesh generato da vertici tridimensionali. È stata quindi implementata una conversione di coordinate da Vector3 a Vector2 rimuovendo l'asse z, dato che si tratta di superfici orizzontali, sono stati processati i punti fornendoli come input all'algoritmo, e infine è stata eseguita una conversione inversa da Vector2 a Vector3 per permettere di generare una mesh tridimensionale.

3.2 Schermate e Funzionalità

Questa parte tratterà delle funzionalità che l'applicazione offre direttamente all'utente, mostrandone qualche porzione di codice e descrivendo l'aspetto delle schermate attraverso degli screenshot.

3.2.1 Tracciamento

```
if (ARSession.state == ARSessionState.Unsupported)
{
    testoSupported.enabled = true;
    testoSupported.text = "Il dispositivo non è compatibile con le funzioni AR richieste!";
}
else if (ARSession.state != ARSessionState.SessionTracking)
{
    testoSupported.enabled = true;
    testoSupported.text = "Inquadra il pavimento e muovi leggermente il cellulare per avviare il tracking";
}
```

Figura 3.4: Codice di controllo per verificare compatibilità del dispositivo con le funzioni AR.

Nella fase di avvio dell'applicazione, dopo aver verificato che il dispositivo sia compatibile con le funzioni di realtà aumentata, come mostrato in Figura 3.4, viene, per prima cosa, presentata all'utente un'interfaccia in cui sono spiegate le istruzioni di funzionamento, in modo che venga subito capito come bisogna procedere in maniera rapida e intuitiva. Tali indicazioni (Figura 3.5) servono inoltre a far capire all'utente che per avere una migliore precisione in fase di tracking, deve visualizzare l'ambiente attraverso più angolazioni.

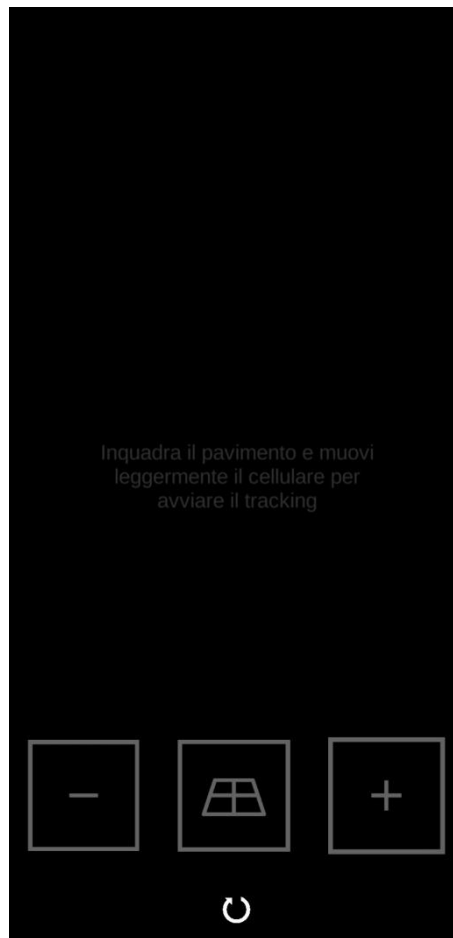


Figura 3.5: Schermata d'avvio in cui vengono visualizzate le istruzioni iniziali.

Una volta che il dispositivo rileva l'ambiente che lo circonda e individua delle superfici piane, l'interfaccia utente cambia; i bottoni diventano interattivi ed appare un mirino al centro dello schermo. A questo punto l'applicazione entra nella fase di creazione del pavimento virtuale in cui l'utente può cominciare a interagire direttamente con essa.

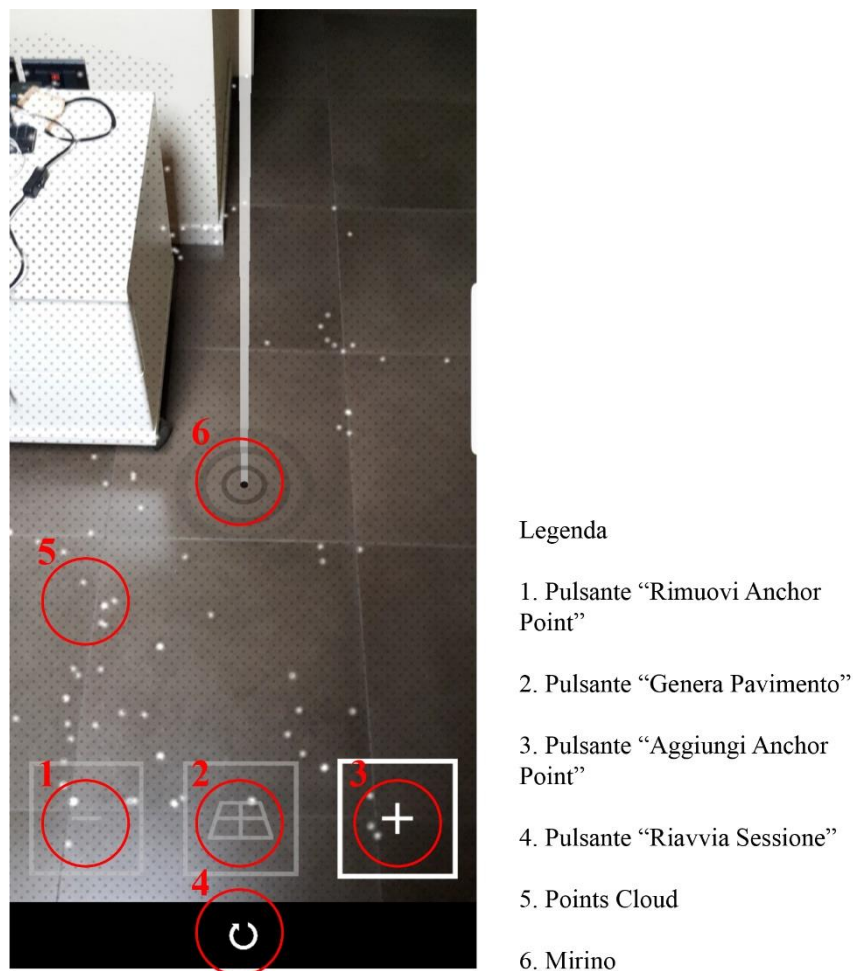


Figura 3.6: Interfaccia utente durante la fase di tracciamento.

Come già citato nel capitolo precedente, l'interfaccia appare all'utente come una semplice schermata con elementi grafici 2D, mentre invece ogni suo singolo elemento è stato studiato per un preciso scopo. Ad esempio, i bottoni sono auto esplicativi, ovvero non c'è bisogno che l'utente si chieda a cosa servano ma lo intuisce da solo osservando l'icona contenuta al loro interno. Inoltre, sono di notevoli dimensioni per facilitarne la pressione da parte dell'utente, ma allo stesso tempo, permettono di vedere ciò che cattura la camera al loro interno per non occupare spazio sullo schermo.

Andiamo quindi ad analizzare gli elementi che presenta tale schermata ed analizziamo le sue funzionalità principali:

- Una volta che il dispositivo inizia a rilevare gli elementi presenti nell'ambiente, l'applicazione entra in fase di tracciamento mostrando all'utente la point cloud nei punti di elevato contrasto (Figura 3.6, elemento 5). Una volta che il software si accorge che la nuvola di punti di una zona precisa può rappresentare una superficie orizzontale attendibile, all'utente viene mostrata una griglia di punti neri in cui è possibile istanziare i pin per la creazione del pavimento virtuale.
- Il cuore di questa interfaccia è il mirino presente al centro dello schermo (Figura 3.6, elemento 6). Grazie a tale elemento, studiato per essere graficamente semplice e intuitivo, l'utente può muoversi nell'ambiente intorno a sé e puntare la camera del dispositivo verso le superfici orizzontali rilevate, e generare i pin virtuali in modo più accurato (Figura 3.7).

```

public IEnumerator MandaRaycast()
{
    while (true)
    {
        if (managerRaycast.Raycast(cam.ViewportPointToRay(new Vector3(0.5f, 0.5f, 0f)), hits, tipoDiTrackableDaIndividuare))
        {
            if (pins.Count < 1)
            {
                hitPose = hits[hits.Count - 1].pose;
            }
            else
            {
                foreach (ARRaycastHit hit in hits)
                {
                    if (hit.trackableId.Equals(pianoIniziale))
                    {
                        hitPose = hit.pose;
                    }
                }
            }
            mirino.transform.position = hitPose.position;
        }
        else
        {
            inserisciPuntoButton.interactable = false;
        }
        if (pins.Count > 0)
        {
            for (int i = 0; i < pins.Count; i++)
            {
                linea.SetPosition(i, pins[i].transform.position);
            }
        }
        yield return null;
    }
}

```

Figura 3.7: Porzione di codice che, tramite raycast, permette al mirino di muoversi in base a dove si sta puntando la camera del dispositivo.

- I pulsanti presenti nella parte inferiore dello schermo (Figura 3.6, elementi 1, 2, 3) rappresentano l'unica parte di interazione con l'utente di tale interfaccia. Infatti, con il bottone nella parte destra dello schermo, l'utente può istanziare un pin di ancoraggio, che rappresenta un vertice del pavimento virtuale, nel punto in cui sta mirando in quel momento. Una volta istanziato il primo punto del perimetro del pavimento, l'interfaccia mostrerà una linea che collega il pin appena creato con il mirino stesso, ovvero con l'eventuale prossimo punto che l'utente fisserà nell'ambiente. Grazie a queste linee di collegamento, l'utente può subito rendersi conto se il pavimento virtuale che verrà creato, soddisferà o meno le sue aspettative. Quando l'utente ha istanziato punti sufficienti alla creazione del pavimento, può scegliere tra due alternative: istanziare il pavimento virtuale facendo pressione sul pulsante centrale, oppure premere il tasto di riavvio della sessione per rifare tutto da capo, poiché non soddisfatto dei punti che ha ancorato nell'ambiente (Figura 3.6, elemento 4). Se l'utente si accorge, durante la creazione del perimetro del pavimento, che l'ultimo punto inserito non è nella posizione desiderata, può toccare il pulsante sulla sinistra dello schermo per annullare l'ultimo inserimento del pin e riposizionarlo dove meglio crede.

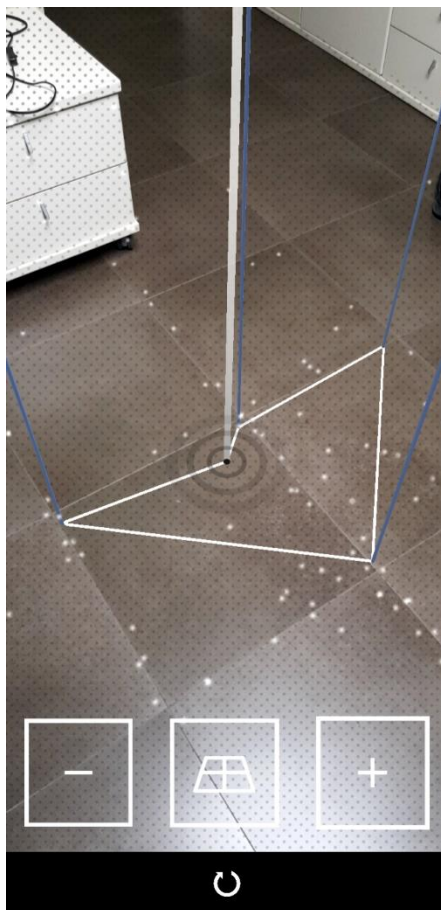


Figura 3.8: Screenshot della schermata durante l'inserimento dei pin utilizzati per delimitare il pavimento virtuale.

L'inserimento di un pin nell'interfaccia grafica avviene mediante raycast, come mostrato in Figura 3.9: grazie al mirino presente al centro dello schermo, dalla camera del dispositivo esce un raggio virtuale che viaggia fino ad intersecare una superficie orizzontale individuata durante la fase di tracciamento. Le coordinate di intersezione tra la linea del raggio e il primo piano orizzontale incontrato rappresentano la posizione del mondo fisico in cui verrà istanziato il pin.

```

public void PosizionaPunto()
{
    Vector3 appoPos = new Vector3();
    referenceManager.GetReferencePoint(referencePoint);
    if (pins.Count < 1)
    {
        if(punti.Count < 1)
        {
            pianoIniziale = hits[hits.Count - 1].trackableId;
            punto = referenceManager.AttachReferencePoint(planeManager.GetPlane(pianoIniziale), hitPose);
        }
        else
        {
            punto = punti[0];
            punto.transform.position = hitPose.position;
        }
        appoPos = hitPose.position;
    }
    else
    {
        appoPos = hitPose.position;
    }
    var pinIstanziato = Instantiate(pin, appoPos, Quaternion.identity);
    pinIstanziato.transform.SetParent(punto.transform);
    pins.Add(pinIstanziato);
    linea.positionCount++;
    linea.SetPosition(linea.positionCount - 1, pins[pins.Count-1].transform.position);
    if(pins.Count == 1)
    {
        linea.positionCount++;
    }
}

```

Figura 3.9: Screenshot di codice che mostra l'implementazione dell'inserimento del pin.

3.2.2 Visualizzazione

Questa schermata si presenta all'utente una volta che ha deciso di istanziare il pavimento virtuale. A questo punto l'interfaccia si semplifica ulteriormente, e scompaiono tutti quegli elementi che servivano alla creazione del pavimento, come i bottoni, il mirino e la nuvola di punti, in modo da far visualizzare all'utente solo il pavimento creato. Se la fase di tracking è avvenuta con successo, il pavimento virtuale sembrerà realmente ancorato a terra con estrema precisione; infatti, avvicinando il dispositivo al pavimento, i suoi punti di ancoraggio rimangono ben saldi nelle posizioni precedentemente selezionate, garantendo così all'utente un'esperienza visiva davvero vicina alla realtà.



Figura 3.10: Schermata di visualizzazione del pavimento virtuale creato.

Una volta creato il pavimento virtuale, oltre ad osservare come si rapporta con l'ambiente circostante, l'utente ha tre possibilità di scelta, selezionabili dai pulsanti presenti nella barra di stato nella parte bassa dell'interfaccia: è possibile resettare l'intera sessione AR e procedere con la creazione di un nuovo pavimento, oppure catturare uno screenshot della schermata, salvarlo e condividerlo tramite apposite applicazioni, o altro ancora è possibile modificare alcuni parametri del pavimento virtuale entrando in modalità "modifica" premendo il pulsante corrispondente.

3.2.3 Modifica

L'ultima schermata che può presentarsi all'utente durante l'esecuzione dell'applicazione è l'interfaccia di modifica dei parametri del pavimento virtuale. Questa schermata è molto importante perché consente una completa personalizzazione del

pavimento da parte dell'utente, permettendo a quest'ultimo di cambiarne i parametri a proprio piacimento.

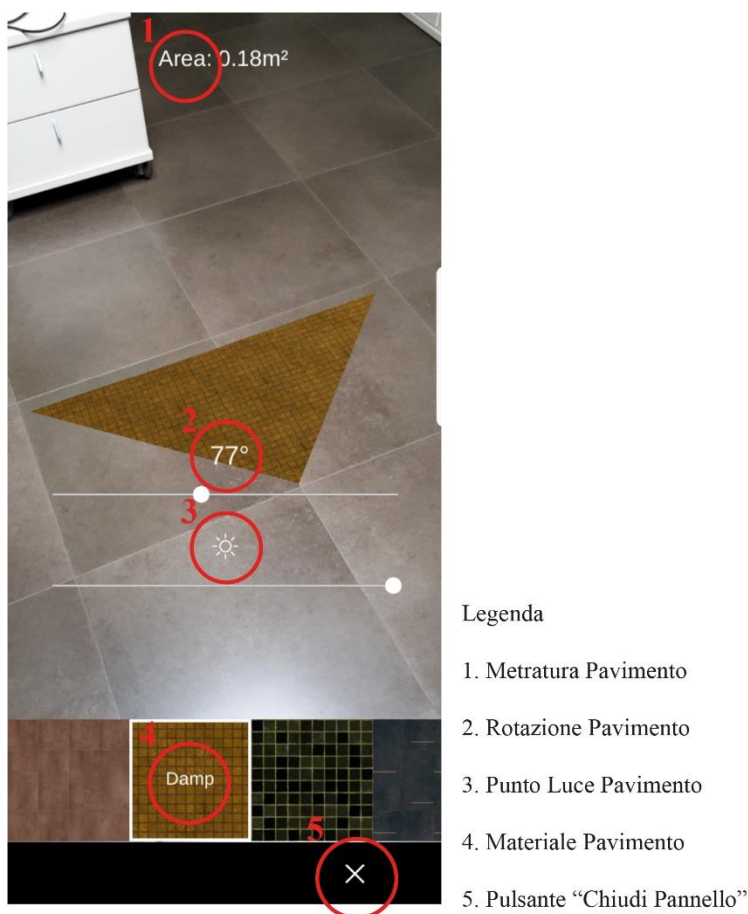


Figura 3.11: Schermata di modifica dei parametri del pavimento virtuale.

Anche questa interfaccia presenta degli elementi semplici e funzionali con i quali l'utente può interagire:

- Il primo elemento che risalta alla vista è la lista dei materiali disponibili. L'utente può scorrere la lista attraverso un movimento di swipe laterale e selezionare la grafica che più soddisfa le sue esigenze (Figura 3.11, elemento 4). Una volta selezionato il materiale desiderato, tramite tocco sullo schermo, questo viene istanziato sul pavimento virtuale permettendo all'utente di avere immediatamente un riscontro visivo.

- Gli *slider* presenti al centro della schermata hanno due funzioni ben precise: la prima è legata alla rotazione della texture del pavimento (Figura 3.11, elemento 2), mentre la seconda invece è legata alla rotazione della finta luce diretta ambientale (Figura 3.11, elemento 3). Infatti, con il primo elemento, si è gestito il caso in cui l'utente voglia allineare il pavimento virtuale a una porzione di pavimento presente nell'ambiente reale. Mentre, nel secondo caso, ci si è immaginato di dover gestire e simulare un punto luce reale, come una finestra, che genera un riflesso nel pavimento. Le Figure 3.12 e 3.13 mostrano l'implementazione di queste funzionalità.

```
public void RotateMaterial()
{
    if(render != null)
    {
        render.sharedMaterial.SetFloat("_Degrees", sliderRotazione.value);
    }

    rotationText.text = " " + sliderRotazione.value + "°";
}
```

Figura 3.12: Codice che regola l'angolazione del pavimento virtuale.

```
public void SetLuce()
{
    var rotazione = new Vector3(0, sliderLuce.value, 0);
    controlluce.lightObject.transform.rotation = Quaternion.Euler(rotazione);
}
```

Figura 3.13: Codice per la regolazione del riflesso di luce del pavimento virtuale.

- L'elemento presente nella parte superiore dell'interfaccia mostra la metratura del pavimento virtuale creato (Figura 3.11, elemento 1). Anche in questo caso, si è ragionato verso un'ottica di utilità poiché a chi vende o compra un pavimento potrebbe essere utile conoscere quanti metri quadrati occupa la superficie del pavimento per stipularne, ad esempio, un prezzo di mercato.

- L’ultimo componente della schermata che andremo a discutere è il pulsante di chiusura del pannello di modifica, situato nella parte inferiore dell’interfaccia (Figura 3.11, elemento 5). Alla pressione di tale bottone, le modifiche effettuate da parte dell’utente vengono salvate e viene chiuso il pannello di modifica dei parametri. A questo punto l’utente si ritrova nuovamente nella schermata di visualizzazione del pavimento per osservare se le modifiche effettuate possono o meno soddisfare le sue esigenze.

3.3 Sviluppi Futuri

In questa sezione si parlerà di quali potranno essere gli sviluppi futuri dell’applicazione. In base alle richieste dei clienti oppure grazie all’innovazione tecnologica, che in questo campo non smette mai di progredire, il software potrà subire dei miglioramenti sostanziali. Tale progetto infatti, è stato sviluppato, sin dall’inizio, garantendo una forte estendibilità, e per questo motivo, il software dell’applicazione, è sempre predisposto ad “accogliere” degli aggiustamenti che mantengano l’applicazione innovativa, performante e al passo con i tempi.

3.3.1 Database

L’azienda che acquista l’applicazione perché interessata alla vendita di pavimenti, potrebbe possedere un gran numero di modelli di piastrelle, marmi, parquet, insomma una notevole quantità di materiali applicabili al pavimento virtuale. Oltre ad occupare troppo spazio in memoria in caso di installazione dell’applicazione, caricare in locale un gran numero di elementi sarebbe un approccio del tutto sbagliato per quanto riguarda l’aggiornamento dei materiali disponibili: possedere una lista numerosa, infatti, potrebbe rallentare il processo di inserimento o eliminazione di un elemento dalla stessa, poiché ogni volta che l’azienda proporrà un materiale nuovo o avrà bisogno di rimuoverne uno obsoleto dal catalogo di vendita, bisognerà mettere mano al codice per aggiornare la lista, rischiando di commettere errori.

Una valida soluzione a questo problema è l'implementazione di un database online che contenga tutti i modelli di materiali di cui un'azienda ha bisogno. L'applicazione quindi dovrà preoccuparsi di connettersi al database e caricare in memoria solo il materiale di cui si necessita in quel momento, risparmiando spazio di archiviazione locale. Valore aggiunto in questo contesto è che l'azienda stessa può aggiornare il catalogo di materiali tramite il proprio database, mentre il codice dell'applicazione non verrà modificato in alcun modo, limitando di gran lunga tutti quei possibili errori derivanti da una continua manipolazione del software.

3.3.2 Tracciamento Verticale

L'idea di base per il tracciamento delle superfici verticali nasce dalla richiesta di chi vende pavimenti: chi vende piastrelle per pavimenti, potrebbe avere bisogno di vendere anche rivestimenti per i muri. In questo contesto, l'applicazione deve permettere all'utente di scegliere per quale tipo di superficie generare la copertura virtuale.

Siccome il tracking verticale dipende dall'ambiente circostante, ovvero se ci sono abbastanza punti di contrasto, attualmente il tracciamento di pareti non è ancora ottimizzato poiché spesso le pareti non possiedono punti di riferimento che il dispositivo riesca a riconoscere.

Esistono metodi per selezionare le pareti verticali, ad esempio fissando punti sul pavimento orizzontale e innalzando dei piani perpendicolari ad esso, ma non avranno mai la stessa precisione con la quale lo smartphone riesce a tracciarle automaticamente attraverso le apposite tecnologie e l'utilizzo dei propri sensori.



Figura 3.14: Esempio di utilizzo di tracciamento verticale [VER18].

3.3.3 Condivisione Screenshot

Una volta creato il pavimento virtuale, all'utente potrebbe tornare utile effettuare uno screenshot alla schermata di visualizzazione e condividerlo. Infatti, adottando un approccio orientato all'industrial design, uno strumento di condivisione simile potrebbe senza dubbio tornare utile all'utente che ha la possibilità quindi di salvare una bozza del pavimento virtuale nella galleria, salvarlo sul proprio cloud privato, oppure mandarlo come allegato utilizzando altre applicazioni di comunicazione. Tutto questo aggiungendo all'applicazione la possibilità di processare un *intent* Android capace di gestire tale funzionalità, come si può osservare in Figura 3.17.

```

public void ShareScreenshotWithText(string text)
{
    string screenShotPath = Application.persistentDataPath + "/" + ScreenshotName;
    if (File.Exists(screenShotPath)) File.Delete(screenShotPath);

    ScreenCapture.CaptureScreenshot(ScreenshotName);
    string testo = "";
    StartCoroutine(DelayedShare(screenShotPath, (testo)));
    SystemManager.instance.animCanvas.SetTrigger("View");
    StopCoroutine("NascondiEScatta");
}

IEnumerator DelayedShare(string screenShotPath, string text)
{
    while (!File.Exists(screenShotPath))
    {
        yield return new WaitForSeconds(.05f);
    }

    NativeShare.Share(text, screenShotPath, "", "", "image/png", true, "");
}

```

Figura 3.17: Prototipo di implementazione della condivisione di uno screenshot tramite intent Android.

Conclusione

Questa tesi è nata con l'intenzione di fornire, ad un eventuale venditore di superfici per pavimenti, uno strumento all'avanguardia e innovativo per far colpo sul cliente, mostrando in tempo reale come sarebbe stato il risultato direttamente sul pavimento di casa sua. Durante lo sviluppo ci si è voluto concentrare sugli aspetti più funzionali che l'applicazione potesse offrire, presentando un prototipo che contenesse tutti gli elementi di base sui quali si potrà successivamente costruire un sistema più complesso, poiché dotato di una forte estendibilità. In modo particolare, l'applicazione mobile, sviluppata sfruttando il supporto del motore grafico Unity e le tecnologie del framework AR Foundation, implementa il concetto di realtà aumentata impiegato nel campo dell'industrial design, quindi calato in un contesto commerciale e finalizzato alla vendita del prodotto. Per questo motivo, l'applicazione ha subito, in fase di progettazione, un processo di design *human-centered*, proprio perché deve suscitare un senso di curiosità in chi la utilizza per avere successo nel mondo del mercato. Per raggiungere tale scopo, infatti, prima di iniziare lo sviluppo e l'implementazione dell'applicazione, ci si è dovuto domandare che cosa un eventuale acquirente potrebbe richiedere a livello funzionale, e come rendere le interfacce utente graficamente intuitive per essere utilizzate al meglio.

Essendo una demo, sono diversi gli aspetti che possono essere migliorati e le funzionalità che potrebbero essere aggiunte per rendere più completa l'applicazione mobile.

Per quanto riguarda le interfacce utente, esse sono state studiate per essere semplici ma del tutto funzionali e intuitive, senza la necessità di aggiungere colori e forme agli elementi, ma concentrando l'attenzione su come far capire immediatamente all'utente il compito di un determinato componente grafico.

Poiché l'evoluzione tecnologica è in continuo progredire, le funzionalità che l'applicazione offre potranno essere migliorate in futuro, sfruttando nuove tecniche innovative che attualmente non sono disponibili o al massimo non del tutto ottimizzate. Ad esempio, i dispositivi saranno in grado di eseguire un tracciamento dell'ambiente più preciso e meticoloso, riuscendo anche a distinguere gli oggetti presenti nel mondo reale e riconoscerli nel contesto in cui sono inseriti (pareti, arredi, superfici). Inoltre, sarebbe una grande evoluzione per l'applicazione, che subirebbe una forte spinta nel mondo del mercato, l'ottimizzazione del tracciamento verticale per il riconoscimento di pareti per istanziare superfici virtuali anche sui muri di casa.

Nel contesto per cui l'applicazione è stata pensata, le funzionalità che essa stessa offre all'utente sono complete, ma con i giusti mezzi tecnologici, potranno essere ottimizzate al meglio anche per i dispositivi che tra qualche anno subiranno notevoli miglioramenti.

L'applicazione in questione ha raggiunto lo scopo per cui è stata ideata, perché forte delle funzionalità che possiede e del supporto che può fornire come strumento di vendita, potrebbe essere acquistata da un'azienda che si occupa di commercio di superfici, e quindi entrare nel mondo del mercato.

Bibliografia

- [DIB05] V. Di Bari, P. Magrassi, *2015 weekend nel futuro*, Edizioni Il Sole 24 Ore, 2005
- [LAT18] F. La Trofa, *#VR Developer. Il creatore di mondi in realtà virtuale e aumentata*, Franco Angeli, Italia, 2018
- [BAT19] G. Battista Tateo, *Gli albori della e breve storia della realtà aumentata*, <http://tateoblog.blogspot.com/p/ar11-gli-albori-della-realta-aumentata.html>, 2019
- [NIB20] Nibes AR Team, *Gli utilizzi ed esempi della Realtà Aumentata*, <https://www.nibescomputing.com/utilizzi-ed-esempi-della-realta-aumentata-2/>, 2020
- [REA20] Wikipedia, *Realtà aumentata*, https://it.wikipedia.org/wiki/Realt%C3%A0_aumentata, 2020

- [VET20] A. Vettese, *Il design industriale*,
<https://www.edatlas.it/documents/db95b296-89eb-4a10-b8f7-714b9846dce2>, 2020
- [BAI10] A. Bai, *Sensorama, la realtà virtuale degli anni '60*,
<https://www.appuntidigitali.it/6895/sensorama-la-realta-virtuale-degli-anni-60/>, 2010
- [ZUG20] Zugara, *Webcam Social Shopper for Web – Virtual Design Room Interface*, <https://zugara.com/wss-for-web-large>, 2020
- [KOH17] C. Kohles, *Top 5 travel apps to enjoy this summer*,
<https://www.wikitude.com/blog-top-travel-apps-enjoy-summer/>, 2017
- [LAH18] S. Lahoti, *Niantic, of the Pokemon Go fame, releases a preview of its AR platform*, <https://hub.packtpub.com/niantic-of-the-pokemon-go-fame-releases-a-preview-of-its-ar-platform/>, 2018
- [LIM18] S. Limberti, *Ikea Place sbarca su Android, arredare casa con la realtà aumentata*, <https://www.hdblog.it/2018/03/20/ikea-place-arredamento-realta-aumentata-arcore/>, 2018
- [ZUC17] A. Zucchiatti, *IKEA Place: l'app di realtà aumentata che ti fa vedere come starebbero i mobili in casa*,
<https://www.ninjamarketing.it/2017/10/09/ikea-place-app-realta-aumentata/>, 2017
- [CAZ20] F. Cazzadore, *Il caso dell'app Ikea Place*,
<https://www.omniaweb.it/blog/ikea-place>, 2020
- [THE20] TheIntellify, *Lifelike and immersive virtual experience*,
<https://theintellify.com/services/augmented-reality/>, 2020

- [CLE19] Cleveland Clinic, *HoloLens to Assess the Resectability of Ovarian Cancer*, <https://consultqd.clevelandclinic.org/hololens-to-assess-the-resectability-of-ovarian-cancer/>, 2019
- [MOR18] A. Morozova, *Augmented Reality in education*, https://medium.com/@info_35021/augmented-reality-in-education-708a0dd7ae71, 2018
- [SUW18] T. Suwannawong, *Tipi di AR*, <http://finalproject2561.blogspot.com/2018/08/types-of-ar.html?m=1>, 2018
- [GRE19] M. Grenacher, *Die neue Sicht für den Kundenservice*, <https://www.it-production.com/produktionsmanagement/die-neue-sicht-fuer-den-kundenservice/>, 2019
- [KRO18] Krol – Hersh, *Uso de la realidad aumentada*, <https://krolhershblog.wordpress.com/2018/06/01/uso-de-la-realidad-aumentada/>, 2019
- [MED20] Mediamodifier, <https://mediamodifier.com/mockups/all>, 2020
- [HTM20] HTML, *Cos'è Unity*, <https://www.html.it/pag/46816/lambiente-di-sviluppo-unity-e-visual-studio/>, 2020
- [INN14] innovaformazioneblog, *Cos'è Unity 3D*, <https://innovaformazioneblog.altervista.org/cose-unity-3d/>, 2014
- [TUT20] unity3dtutorials, *La grafica tridimensionale nel realtime*, <https://www.unity3dtutorials.it/2018/02/06/la-grafica-tridimensionale/>, 2020

- [MON20] Unity Manual, *Script Lifecycle Flowchart*,
<https://docs.unity3d.com/Manual/ExecutionOrder.html>, 2020
- [ARF20] Unity Manual, *About AR Foundation*,
<https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.0/manual/index.html>, 2020
- [TIL19] M. Tillman, *Apple ARKit explained: Everything you need to know about Apple's augmented reality platform*, <https://www.pocket-lint.com/ar-vr/news/apple/141615-apple-ar-kit-explained>, 2019
- [ROU17] M. Rouse, *ARKit*,
[https://whatis.techtarget.com/definition/ARKit#:~:text=ARKit%20\(Apple%20ARKit\)%20is%20Apple's,experiences%20for%20iPad%20and%20iPhone.&text=AR%20scenes%20made%20by%20one,others%20visiting%20the%20location%20later.](https://whatis.techtarget.com/definition/ARKit#:~:text=ARKit%20(Apple%20ARKit)%20is%20Apple's,experiences%20for%20iPad%20and%20iPhone.&text=AR%20scenes%20made%20by%20one,others%20visiting%20the%20location%20later.), 2017
- [JAI18] A. Jain, *What is ARCore by Google? All you need to know to get started*, <https://medium.com/mindorks/what-is-arcore-by-all-you-need-to-know-to-get-started-b3715734cdd3>, 2018
- [TEC19] G. La Terza, *Google ARCore ora supporta nuovi dispositivi*, <https://www.teeech.it/2019/04/11/google-arcore-supporta-nuovi-dispositivi/>, 2019
- [HEL17] Helen, *A look at the newest augmented reality apps coming to ARKit*, <https://applemagazine.com/look-newest-augmented-reality-apps-coming-arkit/35550>, 2017
- [LAN20] Lankinen, *All resources to learn AR Foundation Unity*,
<https://dev.to/lankinen/all-resources-to-learn-ar-foundation-unity-3goi>, 2020

[VER18] *Augmented Startup, Vertical Plane Detection ARCore 1.2 & Gesture
Touch Control – Pinch to zoom touch rotate – Unity,*
<https://www.youtube.com/watch?v=VSMmoebJ8ag&app=desktop>, 2018