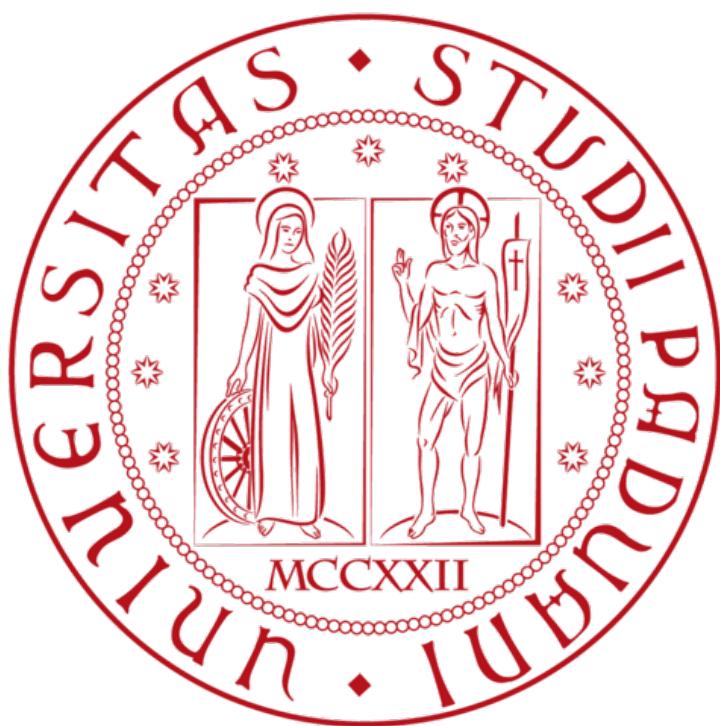


DIPARTIMENTO DI MATEMATICA

CORSO DI LAUREA IN INFORMATICA



Realizzazione di un configuratore di arredo in realtà aumentata

TESI DI LAUREA TRIENNALE

Davide Trivellato

Relatore
Prof. VARDANEGA TULLIO

8 ottobre 2015

Reality exists in the human mind, and nowhere else.

Ringraziamenti

Desidero ringraziare prima di tutti i miei genitori Emanuela e Sergio che hanno sempre creduto in me e che mi hanno sempre supportato in ogni mia scelta, lasciandomi libero di esprimere la mia creatività in ogni sua forma e permettendomi di coltivare le mie passioni. Spero di non avervi deluso e che siate orgogliosi di me come io lo sono di voi.

Ringrazio mio fratello Gianluca, mia sorella Silvia e i miei parenti per avermi sempre fatto sentire l'affetto e il calore della famiglia.

Ringrazio il mio nipotino Tommaso per la sua simpatia e per avermi permesso di fargli da guida. Spero di aver fatto un buon lavoro e di non averlo mai deluso. Ringrazio le mie nipotine Tania, Gaia e Greta che anche se sono ancora piccole riescono sempre a mettermi il sorriso e il buon umore.

Ringrazio i miei amici di Bertipaglia: Matteo, Nicola, Gianmarco, Federico, Alberto, Filippo, Andrea. Mi sono sempre stati vicini e mi hanno supportato anche nei momenti più bui e difficili ricordandomi di non essere mai solo. A voi devo alcuni dei momenti più belli vissuti.

Vorrei ringraziare anche i miei amici da Rovigo: Marco e Massimo. Sono stati i miei compagni di "giochi" e mi hanno fatto scoprire una città molto bella e molto diversa da Padova inserendomi nel loro gruppo e permettendomi di conoscere tante nuove persone. Un grazie anche ai miei compagni di università: Maria Giovanna, Filippo, Matteo, Samuele, Enrico, Francesco. Con loro ho passato uno dei periodi più belli della mia carriera universitaria, fatto di gioie e dolori, di risate ma anche di tensioni. Il miglior team che potessi mai desiderare, siete fantastici.

Un grazie speciale va a Mauro, colui che mi sopporta dai tempi delle superiori. Lui è il mio Samvise Gamgee, gli devo tutto. Penso non abbia mai capito quanto importante sia stato per me in tutti questi anni di università ma senza di lui non so se sarei arrivato dove sono ora. È stato un bellissimo viaggio in tua compagnia che rifarei altre mille volte ancora!

Ringrazio Elena e Giulia, due Amiche con la "A" maiuscola. Possiamo non sentirci e non vederci per mesi ma la nostra amicizia non vacilla mai. Siete state il mio punto di riferimento quando perdevo la retta via.

Un ringraziamento va anche a quanti mi dicevano che non ce l'avrei mai fatta a laurearmi, perché mi hanno dato la motivazione per impegnarmi ancora di più.

Ringrazio tutte le persone che ci sono state e che ora non ci sono più, i nomi sarebbero troppi, ma grazie anche a loro per aver fatto parte della mia vita anche se per poco. Ogni batosta, ogni delusione è servita a farmi crescere e maturare facendomi diventare la persona che sono oggi.

Ringrazio il mio relatore, il prof. Vardanega, per essere stato cordiale e disponibile nei momenti di necessità.

Grazie a tutto il team di Experenti: Amir, Barbara, Gioele, Andrea, Nicolas, Veronica, Lucia ed Elisa. Mi hanno fatto sentire a casa, mi hanno fatto ridere ed emozionare, ma soprattutto, hanno creduto in me. Loro sono la mia seconda bellissima famiglia.

Grazie, infine a tutti quelli che mi hanno permesso di raggiungere questo piccolo traguardo, a voi prometto che mi impegnerò al massimo per fare sì che non sia l'ultimo.

Indice

1 L’azienda	1
1.1 Presentazione	1
1.2 Organizzazione aziendale	1
1.3 Prodotti e soluzioni offerte	2
1.3.1 Lo scenario contemporaneo	2
1.3.2 Il trend	3
1.3.3 Il core business	3
1.3.4 Il prodotto	4
1.3.5 Tipologie di prodotto	5
1.4 Processi aziendali	6
1.4.1 Modello di ciclo di vita software	6
1.4.2 Strumenti a supporto dei processi	9
1.5 Tecnologie utilizzate	10
1.5.1 Unity 3D	10
1.5.2 Vuforia SDK	11
1.6 Propensione all’innovazione	13
2 Strategia Aziendale	14
2.1 Motivazione dello stage	14
2.2 Obiettivo dello stage	14
2.3 Vincoli imposti	15
2.3.1 Vincoli tecnologici	15
2.3.2 Vincoli metodologici	19
2.3.3 Vincoli temporali	19
2.4 Prospettive	20
3 Resoconto dello stage	21
3.1 Pianificazione di progetto	21
3.1.1 Descrizione generale	21
3.1.2 Dettaglio delle attività	22
3.2 Studio delle tecnologie e strumenti	23
3.2.1 Unity 3D	24
3.2.2 Vuforia SDK	25
3.2.3 Photon Unity Networking	25
3.2.4 Esempio di contenuto in realtà aumentata a tema libero	26
3.3 Svolgimento delle attività	29
3.3.1 Il cliente	29
3.3.2 Analisi dei requisiti	30
3.3.3 Progettazione	33
3.3.4 Implementazione	40
3.3.5 Verifica e validazione	46
3.4 Livello di completezza raggiunto	47

4 Valutazione Retrospettiva	48
4.1 Soddisfacimento obiettivi	48
4.2 Conoscenze acquisite	48
4.3 Distanza tra universita' e lavoro	48
4.4 Valutazione personale	48
4.5 Screenshot finali	49
A Realta' Aumentata	52
Glossario	54
A	54
B	54
C	54
G	55
P	55
R	55
S	55
T	55
Bibliografia	57

Elenco delle figure

1	Logo di Experenti srl	1
2	Grafico Hype Cycle sui principali trend (2014)	3
3	Esempio di realta' aumentata - Planisfero	4
4	Esempio di realta' aumentata - Motore	5
5	Esempio di app configuratore	6
6	Flusso dei processi interni secondo la metodologia Agile	8
7	Esempio di Kanban Board	10
8	Interfaccia di Unity 3D	11
9	Architettura di Vuforia SDK e flusso di funzionamento delle operazioni di tracciamento delle immagini	12
10	Esempio sviluppato su Google Cardboard di AR/VR	13
11	Utilizzo dei sistemi operativi mobile nel secondo quarto del 2014 (dato fornito da Net Applications)	16
12	Versione base di Unity	18
13	Diagramma di Gantt delle attivita'	21
14	Implementazione in Unity del tutorial riguardante l'animazione di un avatar 3D di cui successivamente è stato eseguito il porting su Android	24
15	Architettura generale del framework Photon Unity Networking	26
16	Schermata iniziale di Call Of Toony con Lobby	27
17	Call Of Toony - multiplayer su device diversi	28
18	Call Of Toony - schermata di gioco	29
19	Diagramma dei casi d'uso generico relativo all'applicazione Cora' Parquet Live	31
20	Diagramma delle classi generico	34
21	Layout con menu' info aperto	37
22	Layout con menu inferiore aperto	37
23	Animator Controller del pannello Info laterale	39
24	Animator Controller dell'avatar	40
25	Presentazione iniziale	43
26	Presentazione iniziale - Ventaglio di legno in apertura	49
27	Presentazione iniziale - Avatar	49
28	Tutorial - Istruzioni	50
29	Configuratore - Menu' inferiore aperto e Avatar in presentazione	50
30	Configuratore - Pannello info aperto	51
31	Configuratore - Menu' dei Credits aperto	51
32	Esempio di realta' aumentata	52

Elenco delle tabelle

1	Tabella relativa alle ore dedicate per ciascuna attività	23
---	--	----

1 L'azienda

1.1 Presentazione

EXPERENTI srl nasce dalla brillante idea di sfruttare la tecnologia avanzata della realtà aumentata integrandola in un'ottica di business e marketing esperienziale. La startup è nata nel 2012 da una collaborazione tra l'Università di Padova e Mentis, società di consulenza strategica. Dopo due anni, è riuscita a crescere diventando, nel 2014, una vera e propria realtà aziendale ottenendo importanti investimenti che ne hanno permesso un rapido sviluppo ed una veloce espansione al punto da riuscire ad aprire una filiale a New York.



Figura 1: Logo di Experenti srl

1.2 Organizzazione aziendale

La filiale produttiva dell'azienda ha sede presso Massanzago (PD) in Via de Faveri 16, e conta nel suo organico due diversi team:

- un team con competenze relative al contesto commerciale
- un team focalizzato sulla parte tecnica e di gestione di progetto.

Il primo team si occupa degli aspetti commerciali, di marketing e di immagine dell'azienda. L'obiettivo principale del team è la ricerca e l'aggancio di nuovi clienti, nonché la fidelizzazione dei clienti già acquisiti. Per fare questo sono presenti diverse figure:

- CMO (Chief Marketing Officer): una figura con una preparazione in comunicazione e marketing, che conosce molto bene gli aspetti psicologici. Questa persona si occupa di capire come l'azienda viene percepita dall'esterno e cerca di costruirne un'immagine solida che ispiri fiducia e sicurezza nei possibili clienti. Un ulteriore compito è quello di fornire all'azienda la possibilità di partecipare ad eventi e fiere in ambito tecnologico e innovativo e di pubblicizzare l'azienda stessa.
- CEO (Chief Executive Officer): è l'amministratore delegato dell'azienda. Ne definisce le scelte strategiche, dal modello di business all'approccio al mercato. Segue direttamente le relazioni con i clienti chiave che generano oltre 150000 euro di revenues. Sviluppa le relazioni con investitori e partner.

Il secondo team si occupa della parte produttiva e della parte di ricerca e sviluppo. L'obiettivo primario del team è quello di soddisfare il cliente realizzando l'applicazione che più si avvicina alle sue aspettative e nel più breve tempo possibile, in modo efficace ed efficiente. Obiettivi secondari ma non di minore importanza sono la ricerca di nuove tecnologie e l'implementazione di metodologie che aumentino l'efficacia e l'efficienza con cui viene portato a termine il lavoro. Le principali figure della squadra sono:

- PM (Project Manager): il suo obiettivo essenziale è quello di raggiungere gli obiettivi di progetto, assicurando il rispetto dei costi, dei tempi e della qualità concordati e soprattutto il raggiungimento della soddisfazione del committente. Ha una forte preparazione economica.
- CTO (Chief Technology Officer): il suo ruolo e' quello di monitorare le nuove tecnologie e valutarne il loro potenziale applicato ai prodotti e servizi; ma anche quello di supervisionare i progetti di ricerca per assicurare che portino valore aggiunto alla società'.
- sviluppatore software: si prende cura di piu' aspetti del ciclo di vita del software, partendo dall'analisi, passando poi per progettazione e codifica e terminando con il testing e la validazione dell'applicazione.

Al momento l'azienda si compone di una decina di persone, ma e' previsto un ampliamento dell'organico per fare fronte al crescente numero di progetti entranti.

1.3 Prodotti e soluzioni offerte

1.3.1 Lo scenario contemporaneo

Si puo' pensare che ormai e' gia' stato tutto scoperto e inventato, ma non e' cosi perche' ogni rivoluzione tecnologica apre un gigantesco universo di possibili applicazioni.

Quello in cui ci troviamo e' un secolo che vede la piu' rapida espansione ed evoluzione dal punto di vista tecnologico nella storia dell'umanita'. La crescita tecnologica si sviluppa ad una velocita' tale che quello che oggi esce come una novita' tra due mesi viene considerato vecchio e superato.

Inoltre, il modello capitalista adottato dai paesi piu' industrializzati e la globalizzazione impongono una continua ricerca ed aggiornamento dei propri prodotti e servizi per risultare piu' efficaci ed efficienti sul mercato. Per ottenere cio', soprattutto negli ultimi decenni si sono cercate nuove vie di comunicazione per raggiungere il maggior numero di possibili clienti, investendo moltissimi soldi in campagne pubblicitarie.

Ci troviamo in un periodo temporale in cui siamo assuefatti da pubblicita' di ogni genere, al punto da riconoscere un brand esclusivamente osservandone il packaging, oppure osservando il design di un prodotto. Ma la pubblicita' tradizionale sta perdendo efficacia, soprattutto per il fatto che molti mezzi di comunicazione stanno sempre piu' cadendo in disuso. Questo e', per esempio il caso di riviste e televisori, che si sono viste superate da quello che web e dispositivi mobile stanno sempre piu' offrendo.

Lo scopo dell' AR (Augmented Reality) e' quello di offrire nuove strade di comunicazione, da integrare ai dispositivi mobile (smartphone, tablet e visori).

1.3.2 Il trend

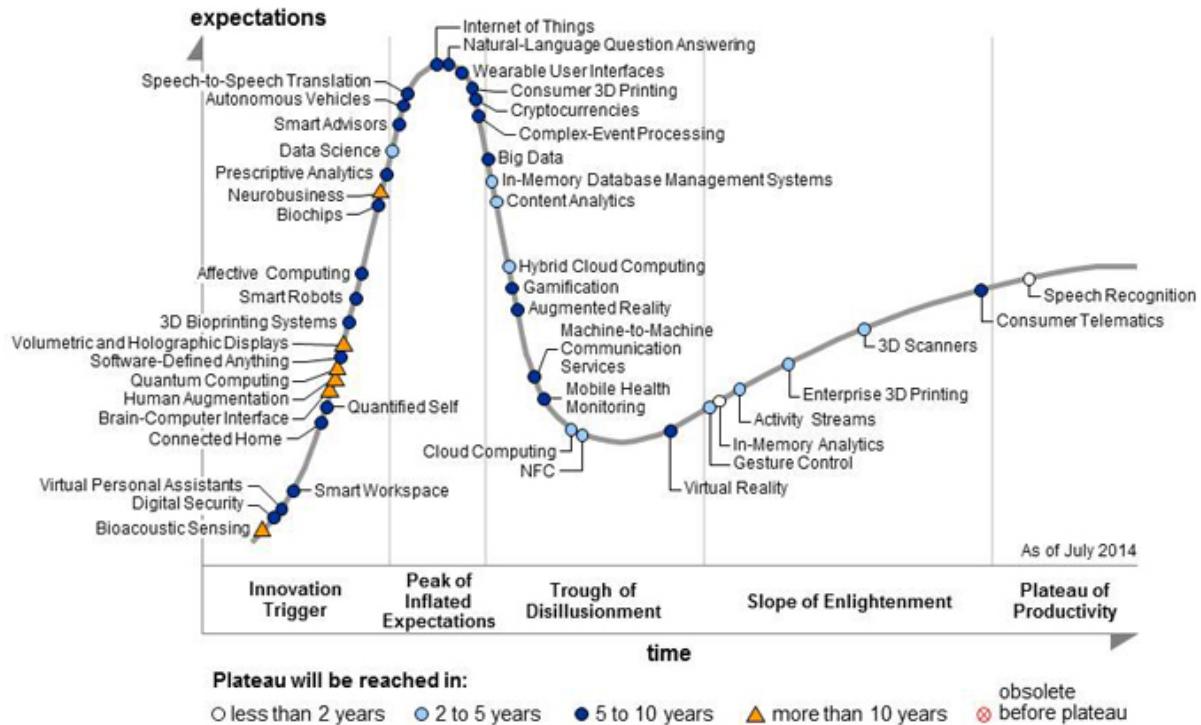


Figura 2: Grafico Hype Cycle sui principali trend (2014)

Quello che emerge da uno studio di Gartner Inc., multinazionale leader mondiale nel campo dell'Information Technology, è una interessante previsione sulla diffusione di alcune tecnologie emergenti. Lo strumento usato da Gartner è lo Hyper Cycle, che consiste in una rappresentazione grafica che mette in scena il ciclo di vita di una tecnologia, dal suo concepimento, alla maturità, alla sua diffusione. Questo, inoltre, viene spesso utilizzato come punto di riferimento nel marketing e nel reporting technology, essendo allegati al grafico i rischi e le opportunità di queste tecnologie.

Nello Hype Cycle possiamo per esempio notare che per la realta' aumentata i tempi per un'adozione di massa sono stimati in un periodo compreso tra i 5 ed i 10 anni, e che quindi un investimento nel settore puo' portare a una crescita esponenziale dei ricavi dell'azienda.

1.3.3 Il core business

L'obiettivo che vuole raggiungere Experenti e' quello di sfruttare la tecnologia della realta' aumentata per creare un servizio pubblicitario che faccia leva sulle emozioni dei consumatori e su quello che viene definito wow factor (espressione inglese che si riferisce a una qualita' o una caratteristica che sorprende; letteralmente elemento sorprendente, o fattore sorpresa). L'obiettivo e' quello di instaurare nell'utente finale un ricordo piacevole e associare quel ricordo a un marchio o a un prodotto.

1.3.4 Il prodotto

Il prodotto vero e proprio che viene creato dall'azienda e fornito ai committenti e' una applicazione mobile che permette la visualizzazione e l'interazione con alcuni contenuti multimediali, come ad esempio video o modelli 3D, applicati a un "tag" riconosciuto tramite l'ausilio della fotocamera integrata nel device. Il funzionamento dell'applicazione prevede che, inquadrando con la fotocamera del proprio dispositivo mobile una particolare immagine o un oggetto 3D chiamati Tag, sia possibile visualizzare i contenuti multimediali associati, "aumentando" così le informazioni percepite attraverso nuovi canali informativi. I prodotti sviluppati da Experenti coprono la richiesta di diversi settori, tra i principali:

- architettura e arredamento;
- comunicazione ed editoria;
- turismo e cultura;
- istruzione e formazione;
- smartcity ed eventi.

Sono settori tendenzialmente distanti dal mondo tecnologico, per cui e' di importanza cruciale un altissimo livello di usabilita' dei prodotti. Per cui il prodotto vero e proprio viene creato sulla base delle esperienze precedenti in termini di usabilita' e di prestazioni (primo su tutti il risparmio energetico della batteria dei device), e sulla base di chi sara' l'utente finale dell'applicazione.



Figura 3: Esempio di realta' aumentata - Planisfero



Figura 4: Esempio di realta' aumentata - Motore

1.3.5 Tipologie di prodotto

L'azienda prevede la scelta fra tre diversi tipi di pacchetto, ognuna studiata per venire incontro alle diverse disponibilita' di budget dei clienti:

- **inserimento di contenuti interni all'app Experenti:** la prima soluzione e quella piu' economica. Questa offerta consiste nell'inserimento di nuovi tag e nuovi contenuti associati all'interno di un visore di realta' aumentata gia' esistente avente come brand Experenti. Non e' prevista la modifica della struttura dell'applicazione che funge da contenitore di elementi provenienti da diverse fonti e destinati a diversi utenti. L'applicazione e' distribuita sia su dispositivi Android che su dispositivi iOS.
- **inserimento di contenuti interni ad un'app personalizzata:** e' la soluzione intermedia, che prevede la creazione di un'app personalizzata sulla base della struttura che possiede l'app di Experenti. E' quindi possibile cambiare nome, logo, palette di colori e contenuti andando a completare le fondamenta standard dell'app.
- **inserimento di contenuti interni ad un'app dedicata:** e' la soluzione meno economica ma piu' completa tra quelle presentate. Consiste nella realizzazione di un'app fornendo completa liberta' di personalizzazione e costruita su misura del cliente.

I prodotti tipicamente realizzati sono di due tipologie diverse:

- **app visore:** e' l'idea su cui si basa l'app Experenti e cioe' quella di fornire un semplice visualizzatore di contenuti, sia video che modelli 3D. E' data la possibilita' di interazione con i contenuti e vengono fornite le funzionalita' per scattare screenshot e abilitare il flash della fotocamera.
- **app configuratore:** e' un particolare tipo di applicazione che vede la sua migliore implementazione in ambito di architettura e arredo. Questo particolare tipo di app permette all'utente di "sfogliare" un catalogo di prodotti e di visualizzarli a grandezza naturale per avere una visione d'insieme all'interno del proprio locale o all'esterno. E' possibile vedere i prodotti nelle proprie varianti e di confrontarli tra di loro. L'applicazione prevede tipicamente un menu' inferiore per scorrere gli elementi e un pannello laterale per visualizzare le informazioni relative a ogni singolo oggetto. E' fornita la possibilita' di effettuare ricerche all'interno del catalogo tramite keyword.

E' prevista l'introduzione di una nuova tipologia di prodotto ancora in fase di sviluppo che e' il **visore di video configurabile da web**. Tutte le altre applicazioni che non rientrano in queste tipologie di prodotto non sono ancora state standardizzate e rientrano nella tipologia di **applicazioni custom**.



Figura 5: Esempio di app configuratore

1.4 Processi aziendali

1.4.1 Modello di ciclo di vita software

Come modello di ciclo di vita software, l'azienda ha deciso di adottare una metodologia AGILE. Si riferisce a un insieme di metodi di sviluppo del software emersi a partire dai

primi anni 2000 e fondati su insieme di principi comuni, direttamente o indirettamente derivati dai principi del "Manifesto per lo sviluppo agile del software". Experenti ha scelto questo modello perché, lavorando in un ambiente altamente innovativo, necessita di prediligere le iterazioni con gli individui esterni e la collaborazione con il cliente oltre a una rapida reazione al cambiamento. Questo tipo di modello si prefigge di ottenere software funzionante tralasciando aspetti importanti ma non essenziali quali, per esempio, una documentazione completa. I principi su cui si basa una metodologia agile che seguono i punti indicati dall'Agile Manifesto, sono quattro:

- le persone e le interazioni sono più importanti dei processi e degli strumenti (ossia le relazioni e la comunicazione tra gli attori di un progetto software sono la miglior risorsa del progetto);
- più importante avere software funzionante che documentazione (bisogna rilasciare nuove versioni del software ad intervalli frequenti, e bisogna mantenere il codice semplice e avanzato tecnicamente, riducendo la documentazione al minimo indispensabile);
- bisogna collaborare con i clienti oltre che rispettare il contratto (la collaborazione diretta offre risultati migliori dei rapporti contrattuali);
- bisogna essere pronti a rispondere ai cambiamenti oltre che aderire alla pianificazione (quindi il team di sviluppo dovrebbe essere pronto, in ogni momento, a modificare le priorità di lavoro nel rispetto dell'obiettivo finale).

La gran parte dei metodi agili tenta di ridurre il rischio di fallimento sviluppando il software in finestre di tempo limitate chiamate iterazioni che, in genere, durano qualche settimana. Ogni iterazione è un piccolo progetto a sé stante e deve contenere tutto ciò che è necessario per rilasciare un piccolo incremento nelle funzionalità del software: pianificazione, analisi dei requisiti, progettazione, implementazione, test e documentazione.

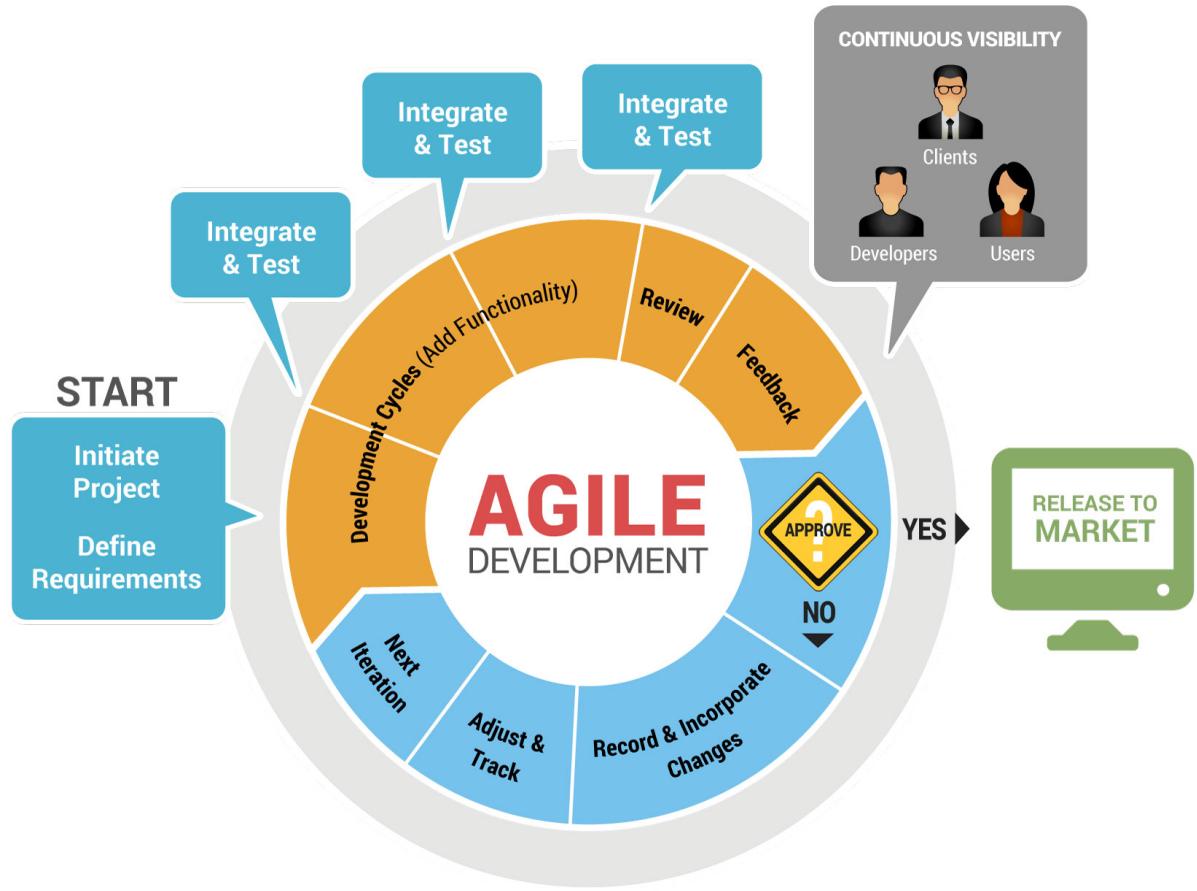


Figura 6: Flusso dei processi interni secondo la metodologia Agile

Il software, all'interno dell'azienda viene sviluppato in finestre di tempo limitate chiamate iterazioni che, in genere, durano dalle 2 alle 4 settimane. Ogni iterazione può essere considerata come un piccolo progetto a sé stante e deve contenere tutto ciò che è necessario per rilasciare un piccolo incremento nelle funzionalità del software: pianificazione (planning), analisi dei requisiti, progettazione, implementazione, test e documentazione. La comunicazione con il cliente avviene quotidianamente, fornendo da parte dell'azienda screenshot o video sulle funzionalità e ottenendo dal cliente feedback e nuove richieste. Anche se il risultato di ogni singola iterazione non ha sufficienti funzionalità da essere considerato completo deve essere rilasciato e, nel susseguirsi delle iterazioni, deve avvicinarsi sempre di più alle richieste del cliente. Alla fine di ogni iterazione il team rivaluta le priorità di progetto, viene eseguita una nuova pianificazione e una nuova progettazione in modo da ottenere un sostanziale incremento alla prossima iterazione, fino al completo soddisfacimento del cliente. Se in corso di progettazione in seguito a una richiesta di modifica dei requisiti ci si accorgere che alcune funzionalità richiedono un numero troppo elevato di risorse rispetto a quanto preventivato ci si accorda con il cliente per trovare un compromesso in modo tale che non resti deluso.

I metodi agili preferiscono la comunicazione in tempo reale, preferibilmente faccia a faccia, a quella scritta (documentazione). Il team agile è composto da tutte le persone necessarie

per terminare il progetto software. Come minimo il team deve includere i programmati ed i loro clienti (con clienti si intendono le persone che definiscono come il prodotto dovrà essere fatto: possono essere dei product manager, dei business analysts, o veramente dei clienti).

1.4.2 Strumenti a supporto dei processi

1.4.2.1 Sistemi operativi Il lavoro viene svolto in ambiente Microsoft Windows 8.1, anche se a fine agosto e' iniziato l'aggiornamento di alcune macchine a Windows 10. Una parte del team di sviluppo, invece, lavora in ambiente MacOS principalmente per la compilazione e pubblicazione di app per iOS che ne rendono l'utilizzo necessario. I programmi utilizzati quali Photoshop, Gimp, Unity, sono cross-platform quindi non e' un problema lo sviluppo su sistemi diversi. Per quanto riguarda il server aziendale vediamo la presenza di un sistema Linux così come per gli ambienti cloud, in quanto si presta molto bene a quello scopo. Per il testing delle applicazioni vengono utilizzati dispositivi Android aggiornati all'ultima versione 5.1.1 e dispositivi iOS aggiornati alla versione 8.4.1.

1.4.2.2 Gestione del versionamento Per la gestione del versionamento viene utilizzato internamente Tortoise SVN che e' un client grafico Subversion. Si e' scelto il suo utilizzo in quanto, oltre ad essere open source, e' stato scritto per girare come estensione di Microsoft Windows e quindi perfettamente integrabile nel sistema operativo usato per lo sviluppo software. I progetti realizzati sono contenuti in un repository che risiede nei server interni e gestito dal reparto tecnico dell'azienda.

1.4.2.3 Enterprise Resource Planning Come sistema di gestione per integrare tutti i processi di business rilevanti, l'azienda ha scelto di utilizzare **Odoo**, ossia un software ERP OpenSource maturo per la gestione di piccole e medie imprese. Odoo integra, tramite moduli, tutti i processi necessari all'impresa come:

- gestione della contabilità;
- gestione delle risorse umane;
- gestione di vendite e acquisti;
- gestione dei progetti;
- gestione documentale;

Odoo è noto per essere molto completo ed estremamente modulare, con più di 1000 moduli disponibili. È basato su una robusta architettura Model-View-Controller, con un server distribuito, workflow flessibili, una GUI dinamica e report personalizzabili. Le funzionalità principali per cui e' stato scelto Odoo sono:

- **Kanban Board:** utilizzata per organizzare in modo ottimale il lavoro e avere una visione generale sullo stato dei singoli progetti. E' molto usata soprattutto per il fatto che ci si trova ad agire seguendo un modello di sviluppo molto dinamico e soggetto a continui cambiamenti. L'utilizzo della Kanban Board porta ad eliminare

una classe di problemi e sprechi nell'attività produttiva attraverso un approccio sistematico ovvero creando un ambiente di lavoro che rende difficile commettere errori.



Figura 7: Esempio di Kanban Board

- **Gestione presenze e richiesta di permessi:** entrate e uscite sono gestite da Odoo così come la richiesta di permessi, in questo modo risulta semplice capire la disponibilità di personale a breve e lungo termine.
- **Calendarie note condivise:** grazie ai calendari e alle note condivise è possibile avere una visione di insieme altrimenti difficile da osservare.

L'utilizzo di Odoo si è rivelato di importanza fondamentale per gestire in modo efficiente il tempo del personale, ma anche per avere sempre sotto mano le priorità su un progetto piuttosto che un altro, oppure avere sempre un elenco descrittivo delle attività da svolgere durante la giornata.

L'utilizzo delle note condivise è usato soprattutto in ambito bug fixing, in quanto viene tenuta traccia della soluzione a un particolare bug riscontrato in una applicazione e quindi rintracciabile in futuro da altri che incontrano le stesse problematiche.

1.5 Tecnologie utilizzate

1.5.1 Unity 3D

Unity 3D è una potente e flessibile piattaforma di sviluppo utilizzata per la creazione di giochi 2D e 3D, oltre che per esperienze interattive. È un ecosistema completo per chiunque miri a costruire un business sulla creazione di contenuti di alto livello. I punti forti che hanno portato alla scelta di questa piattaforma sono essenzialmente:

- **Migliore motore di gioco mobile:** la caratteristica principale e' che Unity e' il migliore motore di gioco per lo sviluppo di applicazioni mobile. Permette migliaia di ottimizzazioni per ridurre il peso degli assets e opzioni dedicate per ottenere la migliore resa grafica sui vari dispositivi mobile.
- **Supporto al multiplatform:** Unity che permette la creazione dei contenuti una volta sola e la distribuzione su tutte le principali piattaforme mobile, desktop e console, semplicemente con un click.
- **Partnership di successo:** Unity trae molti benefici dalla forte e positiva partnership instaurata con colossi di piattaforme e costruttori di hardware come Microsoft, Sony, Qualcomm, Intel, Samsung, Oculus VR e Nintento.

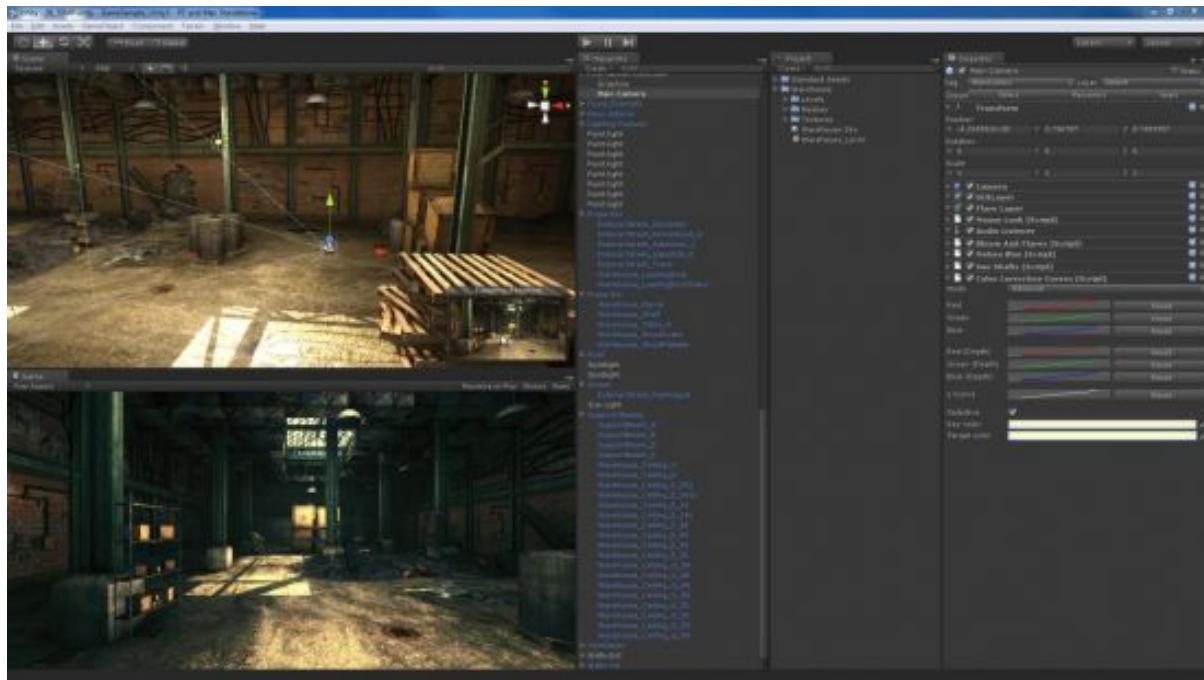


Figura 8: Interfaccia di Unity 3D

1.5.2 Vuforia SDK

Vuforia è la piattaforma software che consente le migliori e più creative esperienze di realtà aumentata attraverso gli ambienti del mondo reale, dando alle applicazioni mobile il potere di "vedere" attraverso la fotocamera del device. La piattaforma Vuforia utilizza un efficiente e stabile riconoscimento delle immagini basato sulla visione artificiale. La visione non e' intesa solo come acquisizione di una fotografia bidimensionale di un'area ma soprattutto come l'interpretazione del contenuto di quell'area. E' stato scelto l'utilizzo di Vuforia per il suo pieno supporto ad iOS, Android e Unity 3D.

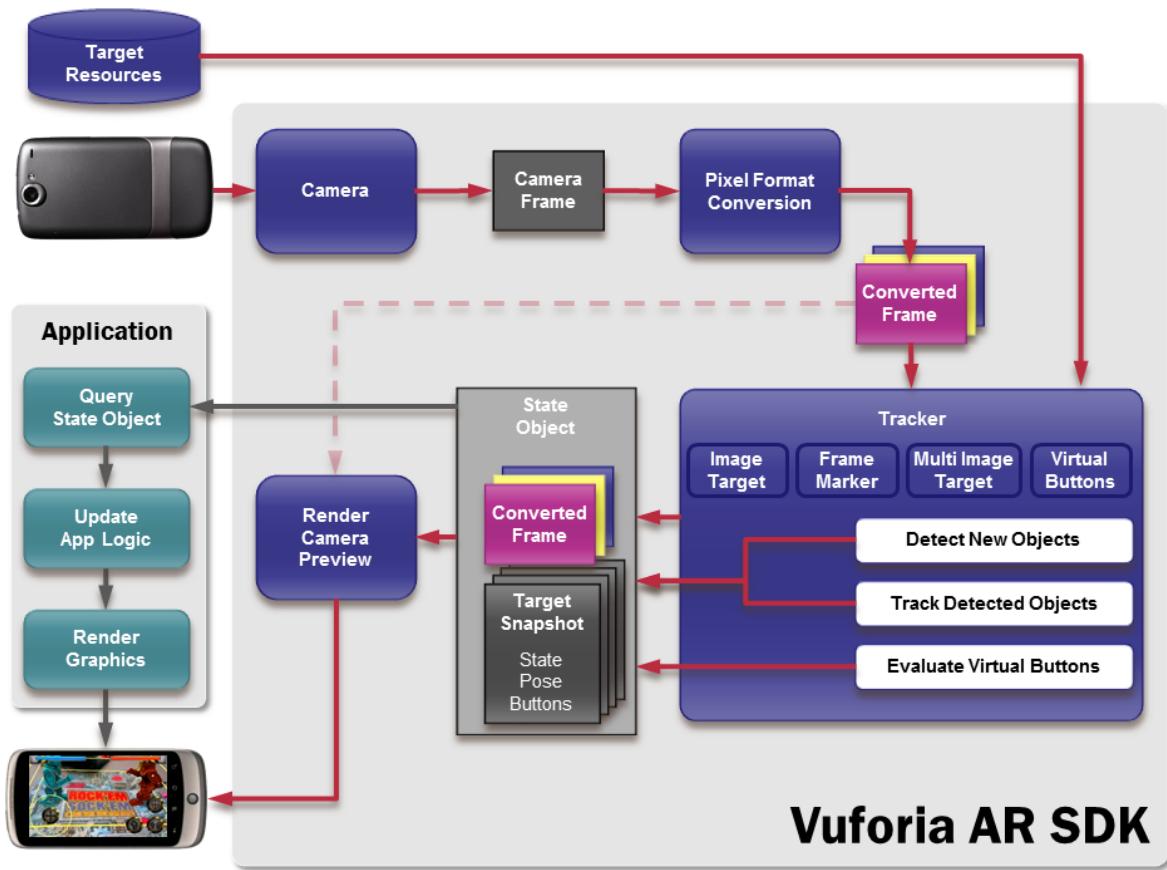


Figura 9: Architettura di Vuforia SDK e flusso di funzionamento delle operazioni di tracciamento delle immagini

Di seguito vengono spiegati i componenti dell'architettura di Vuforia SDK AR.

- **Application:** viene eseguita su device. In base ai dati di input avviene l'aggiornamento di stato di alcuni oggetti forniti da Vuforia che servono ad aggiornare l'App Logic e a renderizzare la grafica sullo schermo.
- **Camera:** assicura che tutti i frame della camera vengano passati al Pixel Format Conversion per ulteriori elaborazioni.
- **Pixel format Conversion:** converte i frame provenienti dal modulo Camera in modo da essere riconosciuti dal rendering e dal tracking dell'OpenGL ES. Questo modulo è necessario in quanto le fotocamere sono diverse da dispositivo a dispositivo e forniscono diversi formati di frame.
- **Target Detection:** Vuforia SDK fornisce il riconoscimento del tag in tre forme:
 - tag definiti dall'utente;
 - tag interni all'applicazione;

- tag gestiti in cloud.

I tag definiti dall'utente sono definiti usando un'algoritmo interno disponibile in Vuforia SDK. Per la seconda tipologia di tag, e' necessario il caricamento delle immagini sul portale di sviluppo di Vuforia, per poi scaricare un file da utilizzare in Unity. L'ultima tipologia usa il riconoscimento ricercando i tag tra quelli caricati sul portale di sviluppo di Vuforia.

- **Tracker:** e' il cuore di Vuforia SDK dove sono scritti tutti gli algoritmi di visione computerizzata per ogni tipologia di tag (immagini, cilindri, etc.). Il modulo si occupa di creare oggetti di stato, in base ai dati ricevuti, che verranno poi utilizzati dall'applicazione sviluppata.

1.6 Propensione all'innovazione

Come già accennato in precedenza, Experenti sta cavalcando l'onda di un trend molto caldo, e l'arrivo sul mercato dei visori di realtà virtuale e realtà aumentata sono un passo in avanti che l'azienda è pronta ad affrontare. Durante il mio periodo di stage ho potuto osservare come il team di sviluppo sia sempre aggiornato sulle nuove tecnologie e sulle release di nuove versioni di tecnologie già utilizzate. L'azienda ha già avviato ricerche e sperimentazioni per:

- l'utilizzo oggetti tridimensionali reali come tag;
- la creazione di applicazioni in realtà aumentata senza l'utilizzo di alcun tag;
- l'implementazione della realtà aumentata su visori AR/VR e quindi la creazione di applicazioni di realtà aumentata mista realtà virtuale.

L'azienda ha già portato avanti alcuni esempi di realtà aumentata sui Google Cardboard con discreti successi. Anche se l'innovazione è un aspetto molto importante per l'azienda, non può essere prioritaria per il fatto che le risorse sia umane che finanziarie sono ancora limitate e assegnate in primo luogo alla produzione.

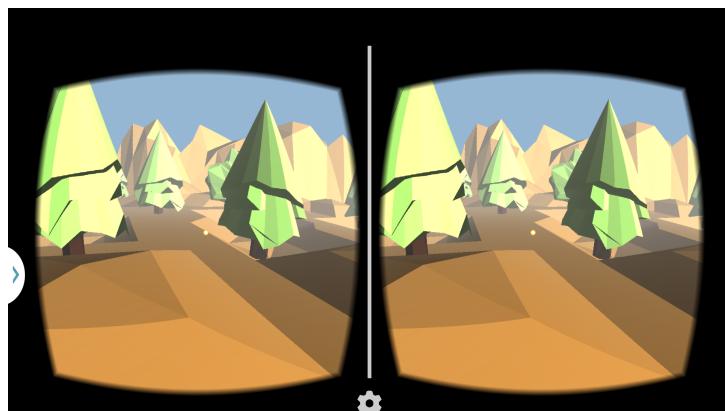


Figura 10: Esempio sviluppato su Google Cardboard di AR/VR

2 Strategia Aziendale

2.1 Motivazione dello stage

Lo stage ha potuto svolgersi grazie all'evento STAGE-IT 2015 che ha permesso l'incontro tra le imprese e gli studenti che sarebbero entrati a breve in stage nel mondo del lavoro con specifico riferimento al settore ICT. L'evento ha favorito un'occasione di conoscenza reciproca mediante colloqui individuali.

Experenti sta vivendo un momento di forte crescita, e ha visto nell'ultimo periodo un aumento del numero di progetti in ingresso. Per fare fronte alla richiesta, l'azienda ha deciso di espandere il suo organico anche in una possibile ottica di inserimento post-stage. Il team di Experenti richiedeva un laureando in Informatica che possedesse un'ottima capacità di programmazione ad oggetti, la conoscenza di C# e una propensione per la parte di progettazione propedeutica al coding vero e proprio. Inoltre, era apprezzata una qualche esperienza con modellazione, rendering 3D e con il motore grafico Unity 3D.

Per una azienda avviata da soli due anni, e' di fondamentale importanza gestire in modo ottimale le risorse, soprattutto quelle finanziarie. Per cui, l'azienda ha valutato positivamente il fatto di poter prendere uno stagiaire a tempo limitato senza obbligo di retribuzione, in modo da avere a disposizione ulteriori forze nell'immediato per gestire il notevole numero di progetti entranti in quel periodo.

Non e' stata una scelta dettata esclusivamente dalla necessita' di manodopera, pero', in quanto il tempo di formazione dello stagiaire comportava un dispendio iniziale di risorse, in quanto era necessario l'affidamento di un tutor aziendale per l'insegnamento delle metodologie, dell'utilizzo degli strumenti e delle best practises presenti in azienda.

2.2 Obiettivo dello stage

Lo stage prevedeva la suddivisione delle attivita' in due parti: la prima prettamente formativa, ha occupato circa il 60% del periodo di stage, mentre la seconda, che ha occupato il successivo tempo restante, si e' concentrata sulla parte produttiva dell'attività aziendale, in particolar modo sulla parte orientata alla realizzazione di progetti destinati ai clienti esterni.

Come **obiettivo minimo** era richiesto di sviluppare almeno un singolo contenuto complesso in realtà aumentata (ovvero: non video AR semplice e non 3D statico AR) da inserire all'interno di un'app commissionata da un cliente esterno.

Mentre, come **obiettivo massimo** era richiesto di sviluppare un'intera app visore di AR, completa di tutti i suoi contenuti semplici e complessi e della propria grafica, dalla fase di accettazione dei materiali in entrata fino alla fase di consegna della beta finale al cliente. Il progetto che avrei dovuto seguire non era stabilito sin da subito, ma e' stato concordato a stage gia' avviato, in seguito all'ingresso di un progetto commissionato da Cora' Divisione Parquet, di cui parlero' in seguito.

Il progetto consisteva nella realizzazione di un configuratore di arredo in realtà aumentata e nella gestione di un avatar 3D che effettuasse una presentazione iniziale e si occupasse di seguire l'utente durante l'utilizzo dell'app con una spiegazione sulle varie categorie di prodotto.

Entrando nel dettaglio, era richiesto di partire da un configuratore di prodotto di base, che consiste in un'applicazione tramite la quale gli utenti possono scegliere un modello di prodotto e le caratteristiche desiderate, e una volta definiti possono mandare una e-mail di richiesta preventivo oppure essere rimandati al sito web.

Il prodotto di base da cui bisognava partire era un configuratore, già realizzato, di stufe comprensivo di menu' inferiore per la selezione delle categorie e dei prodotti e un pannello laterale mostrante la descrizione di ogni prodotto. Nel configuratore, inoltre, era già presente uno script per gestire l'auto-focus della camera del device e un mirino con un piccolo pulsante per scaricare il tag nel caso non fosse già disponibile all'utente.

Definita la base di partenza, la prima parte del progetto, era la creazione della GUI personalizzata partendo da una grafica in formato PSD. Successivamente, bisognava inserire i primi prodotti all'interno del configuratore e quindi gestire i singoli dati riguardanti un prodotto in modo da fornire, in futuro, l'eventuale possibilità di effettuare ricerche tramite l'inserimento di keyword in un'apposita casella di input testuale.

Si richiedeva, inoltre, l'implementazione di una feature nuova, per fornire la gestione di pinch-to-zoom realizzata ad-hoc per una pavimentazione irregolare che è quella del parquet. Era richiesta un'estensione della superficie coperta dal parquet in concomitanza ad una particolare gestione (in questo caso un "pizzico" sullo schermo del device) applicata all'oggetto tridimensionale che si stava visualizzando.

La seconda parte del progetto consisteva nella gestione di un avatar umanoide in realtà aumentata e di un modello tridimensionale rappresentante un ventaglio in legno da usare come sipario per la comparsa dell'avatar. L'avatar doveva effettuare una presentazione iniziale dell'applicazione e delle tipologie di prodotto, e si voleva poterlo richiamare successivamente tramite un apposito pulsante per richiedere spiegazioni riguardanti particolari categorie di prodotto concordate con il committente.

2.3 Vincoli imposti

Per tutta la durata di svolgimento dello stage sono state imposte alcune condizioni da rispettare, spiegate di seguito, divise per tipologia.

2.3.1 Vincoli tecnologici

2.3.1.1 Android Android è un sistema operativo personalizzabile per dispositivi mobili sviluppato da Google Inc. basato su kernel Linux. È stato progettato principalmente per smartphone e tablet, con interfacce utente specializzate per televisori (Android TV), automobili (Android Auto), orologi da polso (Android Wear), occhiali (Google Glass), e altri. È per la quasi totalità Free and Open Source Software, ed è distribuito sotto i termini della licenza libera Apache 2.0.

Android dispone di una vasta comunità di sviluppatori che realizzano applicazioni con l'obiettivo di aumentare le funzionalità dei dispositivi. Queste applicazioni sono scritte soprattutto in linguaggio di programmazione Java.

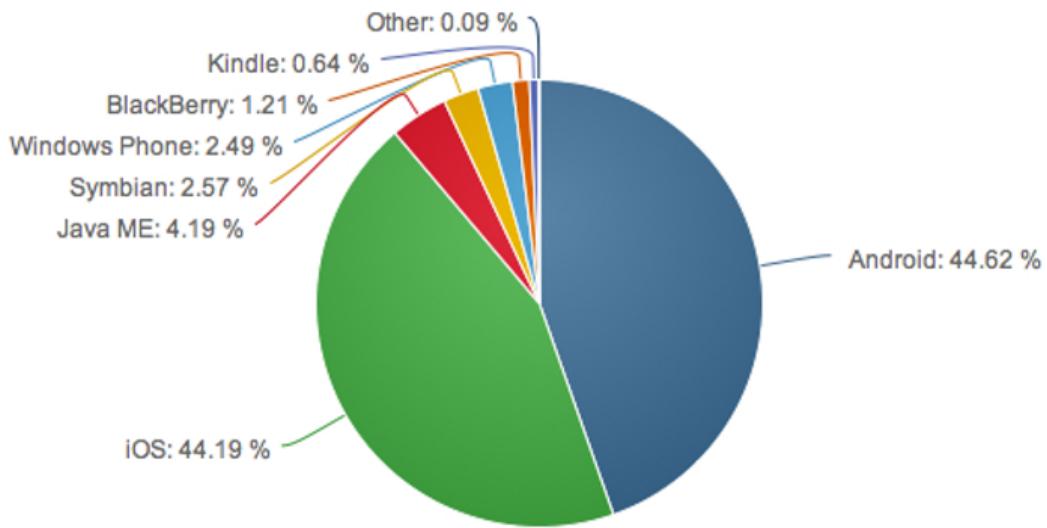


Figura 11: Utilizzo dei sistemi operativi mobile nel secondo quarto del 2014 (dato fornito da Net Applications)

Si e' scelto di sviluppare il progetto, in prima istanza, per i dispositivi Android su richiesta del committente (anche se la release dell'app e' avvenuta anche su iOS) e soprattutto perche', mentre Apple vive in un ambiente chiuso e ben definito, per quanto riguarda Android ci si trova a dover fare i conti con la diversita' di hardware dei dispositivi e diverse risoluzioni degli schermi. Per lo sviluppo dell'app si predilige Android, in quanto possono sorgere piu' problematiche, e dato che Unity, comunque, offre la possibilita' di sviluppare l'app solo una volta e distribuirla a piu' dispositivi diversi.

2.3.1.2 Unity 3D Come gia' spiegato in precedenza, Unity 3D e' un ambiente di sviluppo per la creazione di giochi. Come prima cosa fornisce un potente engine di supporto. Il motore in questione offre il supporto completo per tutti gli aspetti quali rendering grafico, effetti di luce, creazione di terrain, simulazioni fisiche, implementazione dell'audio, funzionalita' di rete e, cosa piu' importante, un sistema di scripting. Unity semplifica di molto la vita del programmatore con piccole feature di qualita', come per esempio la "live preview", che consente di vedere dal vivo quello che si sta creando, con un solo click. Inoltre, mette a disposizione un sistema di gestione delle risorse da usare nel progetto efficace ed intuitivo. I formati di file supportati dall'engine sono molteplici. Per i modelli 3D si ha il pieno supporto ai file generati da:

- Maya;
- 3D Studio Max;
- Cinema4D;
- Blender;
- SketchUp;
- Cheetah;
- Lightwave;
- XSI;
- Carrara;

- Wavefront Obj.

Mentre per quanto riguarda le immagini, i formati supportati sono:

- | | | |
|---------|--------|---------|
| • JPEG; | • BMP; | • PICT; |
| • PNG; | • TGA; | |
| • GIF; | • IFF; | |

Anche a livello di audio il sistema si difende piuttosto bene:

- | | |
|---------------|---------|
| • MP3; | • AIFF; |
| • Ogg Vorbis; | • WAV. |

Infine, per i video si ha supporto a:

- | | | |
|--------|--------|---------|
| • MP4; | • AVI; | • ASF; |
| • MPG; | • MOV; | • MPEG. |

Uno dei punti di forza di Unity 3D e' il fatto di essere gratuito in quasi tutte le sue features. Tuttavia, per particolari scopi o esigenze di pubblicazione, ci sono alcuni strumenti o parti del sistema che sono a pagamento. Unity e' la versione base che viene rilasciata gratuitamente con quasi tutte le funzionalita' piu' importanti a disposizione. Unity Pro, la versione apagamento, consente allo sviluppatore di usufruire di diverse features non presenti nella versione normale; su tutte l'assenza dello splash screen di Unity, cioe' della schermata iniziale mostrante il logo di Unity. Per usi commerciali, l'azienda necessita l'utilizzo della versione Pro, anche per il fatto che questa versione mette a disposizione utilissimi effetti di render su texture, di post processing e su luci e ombre.

Nella figura 12 si puo' osservare una tipica schermata di lavoro di Unity nella versione base (non Unity Pro) composta da cinque sezioni ben distinte:

1. **Scene:** La scena e' lo spazio di lavoro in cui vengono posizionati gli oggetti di gioco e in cui e' possibile avere una visione "grezza" di come apparira' la nostra applicazione. In questa zona e' possibile effettuare azioni sugli oggetti, quali modificare la scala, cambiare la loro posizione sugli assi, ruotarli o modificare le loro ancore e pivot.
2. **Game:** questa sezione e' dove viene mostrata la "live preview" dell'applicazione. Schiacciando il pulsante "play", infatti, sara' possibile avviare l'applicazione e interagire con essa, con la possibilita' di poter usufruire di tutte le gesture mobile tramite l'uso di tastiera e mouse. Per simulare la camera del dispositivo viene usata una webcam collegata al computer.
3. **Hierarchy:** e' la gerarchia di oggetti di gioco istanziati. Qui si possono tenere sotto controllo tutti gli oggetti presenti nella scena e modificarne le parentele.

4. **Project:** nella sezione Project vengono inserite tutte le risorse che si vuole rendere disponibili nel progetto come texture, script, modelli 3D, etc. Qui vengono inseriti anche gli oggetti non istanziati nella scena, ma che devono essere istanziati runtime.
5. **Inspector:** l'inspector è una sezione importantissima in cui si possono modificare tutti i parametri di ogni oggetto di gioco e agganciare nuove componenti come per esempio animator, collider, etc.

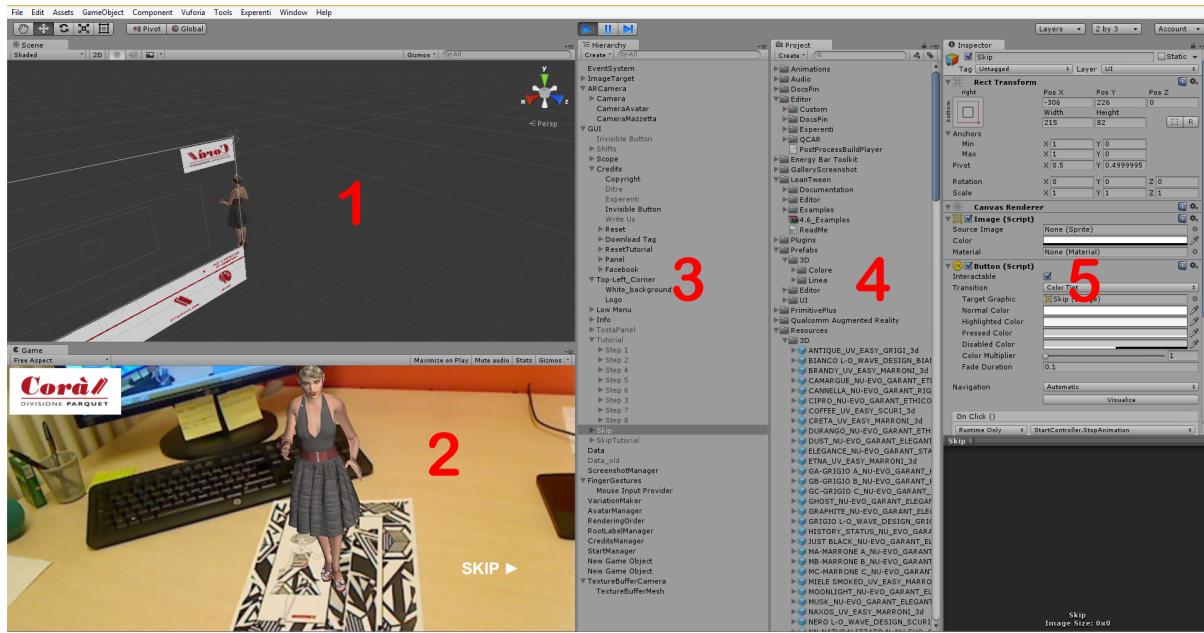


Figura 12: Versione base di Unity

Anche il layout delle sezioni da utilizzare in Unity mi è stato imposto, in quanto più persone lavorano sullo stesso progetto ed è necessario non rimanere disorientati da una diversa disposizione dell'interfaccia.

2.3.1.3 OpenGL ES OpenGL ES è uno standard industriale per la programmazione grafica 3D su dispositivi mobile. Khronos Group, un conglomerato che include marchi come ATI, NVIDIA ed Intel si preoccupa di definire ed estendere lo standard. Attualmente esistono molte versioni diverse delle specifiche OpenGL ES. La versione 1.0 venne ricalcata sulla versione 1.3 di OpenGL, mentre la versione 1.1 si basa su OpenGL 1.5 e la 2.0 è definita in relazione a OpenGL 2.0. La versione di OpenGL ES utilizzata era la 2.0 poiché, ormai, la gran parte dei dispositivi Android supporta quel tipo di libreria grafica.

2.3.1.4 Vuforia SDK Come già descritto in precedenza, è stato imposto il vincolo di utilizzo di Vuforia SDK, rispetto al suo principale competitor Metaio. La scelta è stata fatta alle origini dell'azienda, ed è stata fatta sulla base del supporto fornito in quanto documentazione, supporto e tutorial. Vuforia è una piattaforma completa che offre feature di spessore quali:

- Rilevamento istantaneo dei tag locali;
- Riconoscimento cloud fino a 1 milione di tag simultanei;
- Riconoscimento e tracking di testo stampato;
- Tracking contemporaneo fino a 5 tag;
- Risultati eccezionali con condizioni ambientali sfavorevoli come tag semi-coperti e carenza di luce.

Rispetto a Metaio, Vuforia SDK presenta un tracking migliore e una migliore integrazione con Unity 3D, che ne hanno dettato la scelta rispetto all'utilizzo di Metaio e altri competitor minori.

2.3.1.5 C# Per la scrittura del codice e' stato imposto l'utilizzo di C#, in quanto e' un linguaggio molto simile al Java, di cui avevo gia' una buona preparazione. L'alternativa all'utilizzo di C# sarebbe stata Javascript, un linguaggio molto potente ma che avrebbe richiesto ulteriore tempo di formazione. Essendo che piu' persone collaborano sullo stesso progetto, si e' ritenuto indispensabile la piena comprensione da parte di tutti i membri del team di sviluppo del linguaggio utilizzato.

2.3.2 Vincoli metodologici

Durante tutta l'attività di stage, e' stato imposta la gestione del versionamento tramite l'utilizzo di TortoiseSVN, inserendo le componenti in un repository interno al server locale dell'azienda. Non mi e' stato fornito alcun vincolo su quando effettuare i commit, la cui gestione e' stata lasciata a me.

2.3.3 Vincoli temporali

Il progetto prevedeva che lo stagiaire svolgesse un totale di 320 ore di attività presso l'azienda ospitante, suddivise in circa 40 ore settimanali soggette a possibili variazioni nel caso di scadenze aziendali o di impegni di varia natura da parte dello studente. Tali ore si dovevano svolgere internamente all'orario d'ufficio, dal lunedì a venerdì dalle 9:00 alle 13:00 e dalle 14:30 alle 18:30.

Le prime date concordate di inizio e fine stage sono state, rispettivamente, 2015-07-09 e 2015-09-09. In seguito a problematiche sorte da parte del tutor aziendale a ridosso della data di inizio stage si e' deciso di riconcordare nuovamente le date di inizio e fine stage, rispettivamente, 2015-07-13 e 2015-09-11.

Il periodo concordato e' stato suddiviso in due sezioni temporali della stessa dimensione:

- la prima parte prettamente formativa, consisteva nello studio delle tecnologie utilizzate e nella realizzazione di un piccolo progetto per l'assimilazione dei concetti appresi;
- la seconda parte, iniziata in conclusione della parte formativa, consisteva nella realizzazione del progetto vero e proprio.

2.4 Prospettive

Come descritto in precedenza, Experenti vuole ampliare il suo organico, inserendo nel team alcune nuove figure con una preparazione in ambito informatico e grafico. Il numero sempre crescente di progetti entranti rendono necessario l'inserimento di sviluppatori Android e iOS che seguano attivamente i progetti dalla fase di raccolta dei materiali, alla fase di progettazione, codifica e testing.

Il settore altamente innovativo e appena nato della realta' aumentata, comporta la possiblita' di lavorare su progetti all'avanguardia in ambito AR. Basta osservare il lavoro svolto nel mio stage per accorgersi delle continue migliorie che gli si possono applicare. Experenti presta molta attenzione nella ricerca di clienti che possano offrire motivazione nel cercare soluzioni per il raggiungimento di un grado di innovazione sempre maggiore e cercare, quindi, di essere sempre un passo avanti rispetto ai competitor.

3 Resoconto dello stage

3.1 Pianificazione di progetto

3.1.1 Descrizione generale

Come già detto in precedenza, lo stage è stato suddiviso in due parti: la prima orientata alla formazione su strumenti e tecnologie e la seconda parte orientata alla realizzazione di progetti destinati ai clienti esterni. Come tale, l'attività di formazione è stata opportunamente orientata all'apprendimento, da parte mia, delle meccaniche e delle norme vigenti internamente per lo sviluppo di tali progetti, oltre che alla normale parte di formazione tecnica prevista per portare a termine in maniera opportuna le attività dei progetti stessi. L'obiettivo finale dello stage è stato quindi quello di inserirmi come parte integrante del team di sviluppo per i progetti esterni, attribuendomi responsabilità e compiti adeguati al mio ruolo e orientati alle attività di produzione, testing e delivery di app mobile di Realtà Aumentata; la valutazione finale da parte del tutor aziendale è stata quindi effettuata sulla base sia della qualità sia della quantità delle attività portate a termine nella fase produttiva finale, oltre che alla capacità di lavorare correttamente in squadra con l'obiettivo comune di consegnare un prodotto finale nei tempi e nelle modalità stabilite.

Nel periodo antecedente l'inizio dello stage, insieme al tutor aziendale, sono state concordate le attività principali che avrei dovuto svolgere durante il periodo seguente della durata di 2 mesi. Nella descrizione delle attività, riportata nella sezione successiva, è stata fornita una descrizione molto generica per quanto riguarda il progetto principale che avrei dovuto seguire in quanto non era ancora chiaro a priori se ci sarebbe stata o meno la possibilità di seguire un progetto commissionato dall'esterno.

La dislocazione temporale delle attività è stata rappresentata graficamente in un Diagramma di Gantt che mi ha aiutato ad avere sempre una visione accurata sullo stato del mio stage, in particolare su eventuali ritardi. Rispetto al diagramma concordato nel piano di lavoro, il mio stage è partito dopo 2 giorni rispetto a quanto concordato a causa di un'indisponibilità del tutor aziendale, per cui è stata rifatta la pianificazione tenendo conto di questo ritardo.

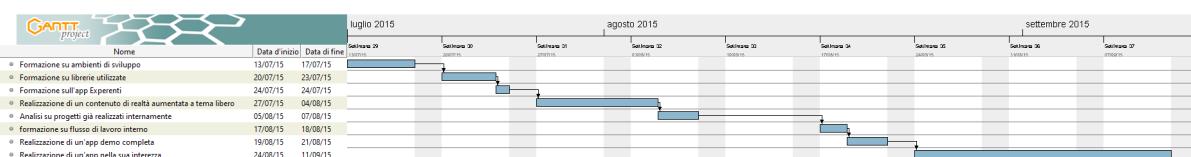


Figura 13: Diagramma di Gantt delle attività'

Il Diagramma di Gantt riportato in figura 13 mostra piuttosto fedelmente quanto svolto durante il periodo in azienda ed eventuali anticipi sulla tabella di marcia sono stati riempiti con approfondimenti sulle tecnologie e sessioni di ricerca e sviluppo su visori Google Cardboard. Non ci sono stati, invece, ritardi su quanto preventivato.

3.1.2 Dettaglio delle attività

Di seguito vengono elencate in dettaglio le attività svolte durante il periodo di stage svolto presso l'azienda ospitante Experenti. Un approfondimento per le principali.

1. Formazione sulle tecnologie utilizzate internamente per lo sviluppo, quali framework e SDK. In particolare:
 - (a) Ambiente di sviluppo (IDE) utilizzato (Unity3D) e fondamenti dei sistemi operativi mobile (Android e iOS);
 - (b) Formazione sulle librerie utilizzate internamente per l'elaborazione delle immagini per la realtà aumentata e per il successivo riconoscimento delle stesse in ambiente mobile;
 - (c) Formazione sull'app Experenti: nascita del progetto, funzionamento attuale, obiettivi di sviluppo. Formazione sulle procedure standard applicate internamente.
2. Realizzazione di un esempio di contenuto in Realtà Aumentata a tema libero. Questo contenuto, il cui sviluppo è stato necessario alla comprensione del flusso di lavoro interno e all'individuazione di determinate problematiche relative all'ambito AR mobile, ha particolari caratteristiche, quali animazioni e/o movimenti di parti specifiche, un certo grado di interattività e prevede parti semplici di grafica GUI (su schermo, in modalità HUD). E' stata richiesta, inoltre, l'individuazione di un tag adatto al riconoscimento dalle fotocamere mobile, possibilmente legato alla tematica che è stata sviluppata.
3. Analisi di casi di studio e app varie già realizzate internamente. Focus particolare sui progetti base già realizzati e sulla loro struttura: progetto base demo, progetto base visore AR, progetto base configuratore. In questa fase è avvenuta la formazione sul flusso di lavoro standard interno all'azienda e sul normale iter di un progetto commissionato da un cliente, dalla ricezione dei materiali fino alla fase di distribuzione (sia essa una distribuzione ad hoc o una distribuzione pubblica tramite Store mobile) ed è iniziato l'affiancamento al Project Manager nelle fasi di accettazione materiali.
4. Realizzazione di un'app demo completa. Per app demo si intende un'app a distribuzione solitamente ad hoc (non pubblicata sugli Store) resa disponibile dall'azienda per i propri clienti o reseller, comprendente un numero solitamente limitato di contenuti semplici (3D o video) fruibili dall'utente in realtà aumentata attraverso l'uso di un tag fornito dal cliente stesso. L'app possiede, inoltre, una GUI minimale ma personalizzata con il logo del cliente stesso, nonché un'icona e una splashscreen anch'esse personalizzate allo stesso modo. Richiesto l'affiancamento al Project Manager fin dalla fase iniziale di ricezione materiali, e prosecuzione poi in autonomia nella fase di sviluppo fino alla fase di rilascio e consegna (previa verifica del risultato prodotto da parte del Tutor Aziendale). L'entità dell'app demo è stata stabilita dal Project Manager aziendale alcuni giorni prima dell'inizio di questa fase e si è data preferenza, alla produzione di una demo per un cliente esterno.

3 RESOCONTO DELLO STAGE

5. Inserimento effettivo nel team di sviluppo per i progetti esterni. In questa fase, inizia l'affiancamento al team di sviluppo per i progetti commissionati dai clienti esterni; è iniziato quindi il coordinamento dal Project Manager aziendale nell'assegnazione di task appositi comprendenti le fasi di sviluppo e testing di intere app semplici o parti di app complesse; si è preferito assegnare la realizzazione di almeno un'app semplice nella sua interezza commissionata da un cliente esterno. L'assegnazione delle attività è stato effettuato attraverso il sistema di ticketing utilizzato internamente all'azienda, attraverso il quale è stato anche richiesto di rendicontare le proprie attività in termini di tempo utilizzato per ciascuna di esse, mentre l'assegnazione dei singoli task è stata effettuato dal Project Manager aziendale in collaborazione con il tutor aziendale. È stato valutato positivamente in questa fase la capacità di attenersi alle tempistiche date e il livello di dettaglio fornito nella successiva rendicontazione delle ore, oltre ovviamente alla qualità intrinseca del risultato prodotto.

Sezione	Descrizione	Ore di lavoro
1.1	Formazione su ambienti di sviluppo	40
1.2	Formazione su librerie utilizzate	28
1.3	Formazione sull'app Experenti	12
2	Realizzazione di un contenuto di realtà aumentata a tema libero	56
3	Analisi su progetti già realizzati internamente e formazione su flusso di lavoro interno	40
4	Realizzazione di un'app demo completa	24
5	Inserimento nel team di sviluppo e realizzazione di un'app nella sua interezza	120
TOTALE		320

Tabella 1: Tabella relativa alle ore dedicate per ciascuna attività

3.2 Studio delle tecnologie e strumenti

In questa sezione, vengono spiegate le attività di apprendimento svolte per imparare l'utilizzo delle nuove tecnologie e degli strumenti usati.

3.2.1 Unity 3D

Unity, come già detto in precedenza è un sistema cross-platform per lo sviluppo di giochi composto da un game engine e da un IDE integrato. Unity viene usato internamente all’azienda per lo sviluppo di app mobile distribuite su Android e iOS.

Unity nel suo sito fornisce un grosso supporto agli sviluppatori fornendo una documentazione completa e una sezione ben fornita di tutorial testuali e video suddivisi per categoria.

Inizialmente, ho dovuto seguire una parte di video tutorial riguardanti l’interfaccia di Unity, lo scripting, la gestione della fisica, animazioni e gestione della GUI. Questo primo periodo si è svolto integrando, oltre alla visione, anche la prova diretta sull’editor in modo da assimilare meglio i concetti appresi.

Nel caso in cui volessi approfondire un argomento oppure non lo ritenessi abbastanza chiaro, avevo sempre la possibilità di ottenere una spiegazione da parte del tutor aziendale, il quale si è dimostrato sempre molto disponibile anche nel ripetere più volte lo stesso concetto.

In questa parte di formazione, dopo aver seguito e implementato un tutorial riguardante l’animazione di un avatar, di mia iniziativa, ho effettuato il porting dell’applicazione su Android gestendo touch e multi-touch sullo schermo e impostando i movimenti dell’avatar basandoli sull’accelerometro del dispositivo mobile.

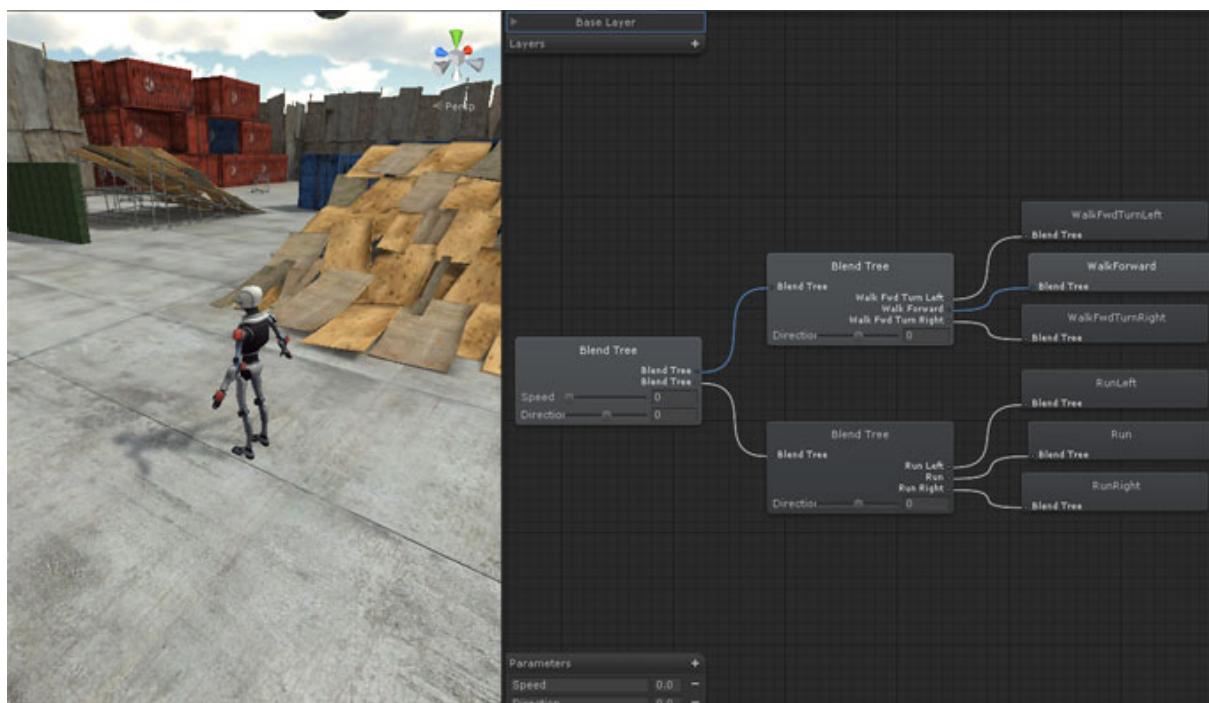


Figura 14: Implementazione in Unity del tutorial riguardante l’animazione di un avatar 3D di cui successivamente è stato eseguito il porting su Android

3.2.2 Vuforia SDK

Terminata la parte di formazione su Unity 3D, è iniziata la parte di preparazione relativa a Vuforia SDK, l'SDK utilizzato dall'azienda per l'implementazione della realtà aumentata. Nello specifico, il team tecnico si è occupato di spiegarmi come funziona l'SDK, e come funziona nello specifico il plugin di Unity, grazie alla quale è possibile operare all'interno di un unico ambiente di lavoro. Le attività principali svolte in questo lasso di tempo sono state l'implementazione di modelli 3D e di video associati ai tag, e lo studio sul riconoscimento e la creazione di tag ottimali.

3.2.3 Photon Unity Networking

Photon Unity Networking (PUN) è un framework di Unity per l'implementazione del multiplayer realtime nei giochi o nelle applicazioni sviluppate. Le applicazioni sviluppate con Photon vengono eseguite su un server cloud proprietario. Quindi, le operazioni di scaling e di service hosting sono gestite interamente da PUN, permettendo allo sviluppatore di concentrarsi puramente sulla costruzione dell'applicazione. Tutti i prodotti Photon Cloud sono basati su un'architettura client-to-server, che è la soluzione ottimale per il gaming online rispetto a una connessione peer-to-peer.

Photon è un package scaricabile dall'Asset Store (negozi di Unity online in cui comprare o scaricare gratuitamente asset come script, modelli 3D e 2D, etc.), e nella sua versione gratuita prevede l'accesso concorrente fino a 20 utenti sulla stessa stanza.

Di seguito vengono riportati esempi di codice per mostrare la semplicità di utilizzo del framework.

- **Connessione al server:** La connessione al server si basa sul passaggio di una stringa contenente la versione dell'applicazione. Può essere usata per dividere gruppi di client.

```
1 PhotonNetwork.ConnectUsingSettings("1.0");
```

- **Accesso a una stanza:** Per prendere parte a una partita esistente basta la seguente riga di codice specificando il nome della stanza in cui si vuole entrare.

```
1 PhotonNetwork.JoinRoom("RoomName");
```

- **Creare una stanza:** per creare una stanza basta fornire il nome, dare la possibilità o meno di essere trovata da altri utenti, fornire la possibilità agli altri di entrare, e il numero massimo di giocatori.

```
1 public void OnConnectedToMaster() {  
    PhotonNetwork.CreateRoom("RoomName", true, true, 4);  
3 }
```

Lo studio di Photon Unity Networking non mi è stato imposto dall'azienda, ma è stato un approfondimento che ho voluto fare di mia iniziativa per lo sviluppo dell'esempio di contenuto in realtà aumentata a tema libero di cui parlerò successivamente.

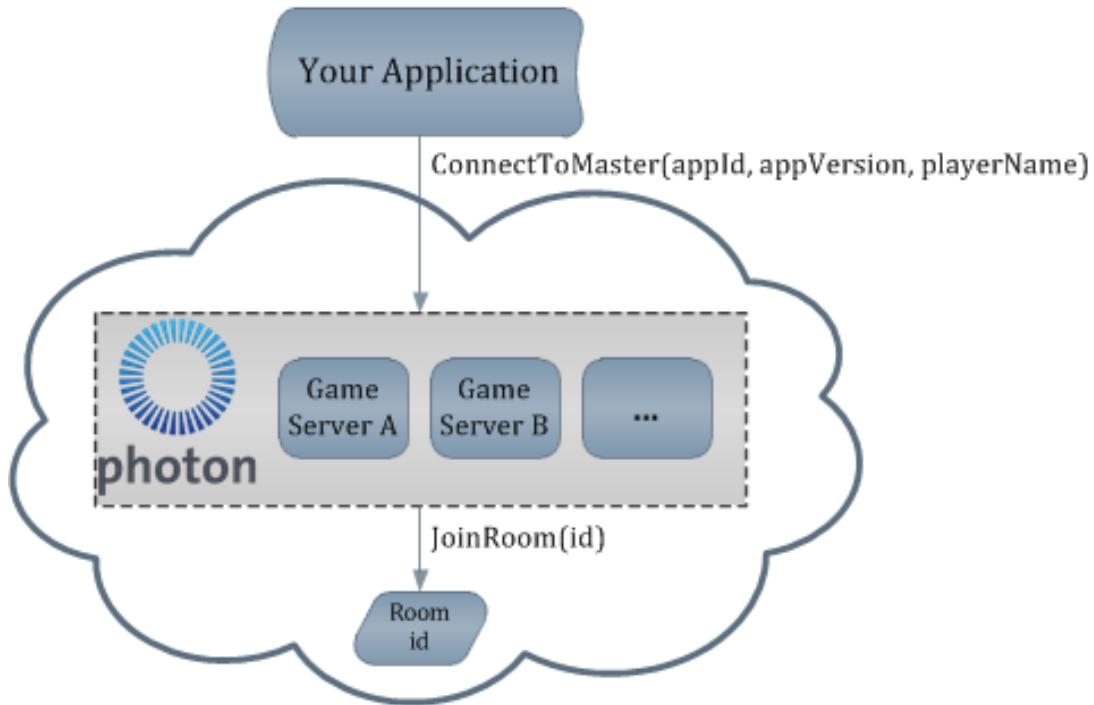


Figura 15: Architettura generale del framework Photon Unity Networking

PUN si è dimostrato uno strumento molto potente e relativamente di facile utilizzo. Creare un semplice scambio di dati tra diversi client è risultato piuttosto semplice. Il livello di difficoltà è salito quando ho cercato di aumentare il numero di informazioni passate e il numero di oggetti da "osservare". Essendo uno studio non richiesto dall'azienda, ho preferito non spendere troppo tempo in approfondimenti ma piuttosto avere un'idea chiara del funzionamento di base.

3.2.4 Esempio di contenuto in realtà aumentata a tema libero

Come ultimo passo a compimento del percorso di preparazione tecnologica e strumentale, e prima di iniziare la realizzazione del progetto principale di stage, mi è stato chiesto di sviluppare un'applicazione completa con dei contenuti non banali in realtà aumentata. Ciò che mi è stato reso disponibile per la realizzazione dell'app oltre a Unity 4.6.3 e Vuforia SDK 4, sono stati anche tutti i contenuti gratuiti disponibili sull'Asset Store. Ho quindi proceduto con un'accurata scansione dei contenuti scaricabili trovando ciò che fosse più utile per dare luce all'idea ins tato embrionale che avevo in mente.

Quello che stavo cercando erano dei modelli 3D in stile "cartoon" per lo sviluppo di un videogioco sparatutto multiplayer in realtà aumentata. L'idea che volevo implementare era dare la possibilità a più utenti di interagire sullo stesso tag, e rendere tali interazioni visibili ad utenti che si trovassero dall'altra parte

3 RESOCONTO DELLO STAGE

del mondo. La prima idea che mi era venuta, era quella di gestire un oggetto 3D e dare la possibilità di modificarne la struttura, il colore e la scala agli utenti, in modo che i cambiamenti fossero visibili a tutti gli utenti in osservazione su quel dato tag. Questo però non mi è bastato, in quanto volevo vedere fino a che punto si poteva spingere un device Android con il rendering di contenuti in realtà aumentata. Per cui la realizzazione di un gioco mi sembrava perfetta per testare questi due aspetti.

Come già detto ho effettuato uno studio di Photon Unity Networking per l'implementazione del multiplayer e successivamente ho proceduto con una semplice progettazione architettonale e con l'implementazione vera e propria.

La prima cosa fatta è stata la realizzazione della mappa di gioco, cercando di renderla il più simmetrica possibile in modo da non sfavorire nessuno dei due giocatori e creando delle barriere invisibili per non permettere ai giocatori di poter uscire dallo scenario.

Il passo successivo è stato quello di creare dei punti di respawn e di gestire la nascita dei giocatori in modo casuale sulla mappa. Avendo trovato sull'Asset Store un modello 3D già in possesso di animazioni e script che simulassero un soldato cartoon vero e proprio, non ho dovuto occuparmi della gestione dell'avatar.

Una volta creati e gestiti i punti di respawn, è iniziato il lavoro di configurazione della lobby (contenitore di stanze) in primo luogo, e delle stanze successivamente, fornendo la possibilità ai giocatori di creare la propria stanza e di vedere le stanze create dagli altri giocatori. Per la lobby è stata creata una schermata iniziale apposita di scelta dello username e di creazione nuova stanza o accesso a una stanza già creata.

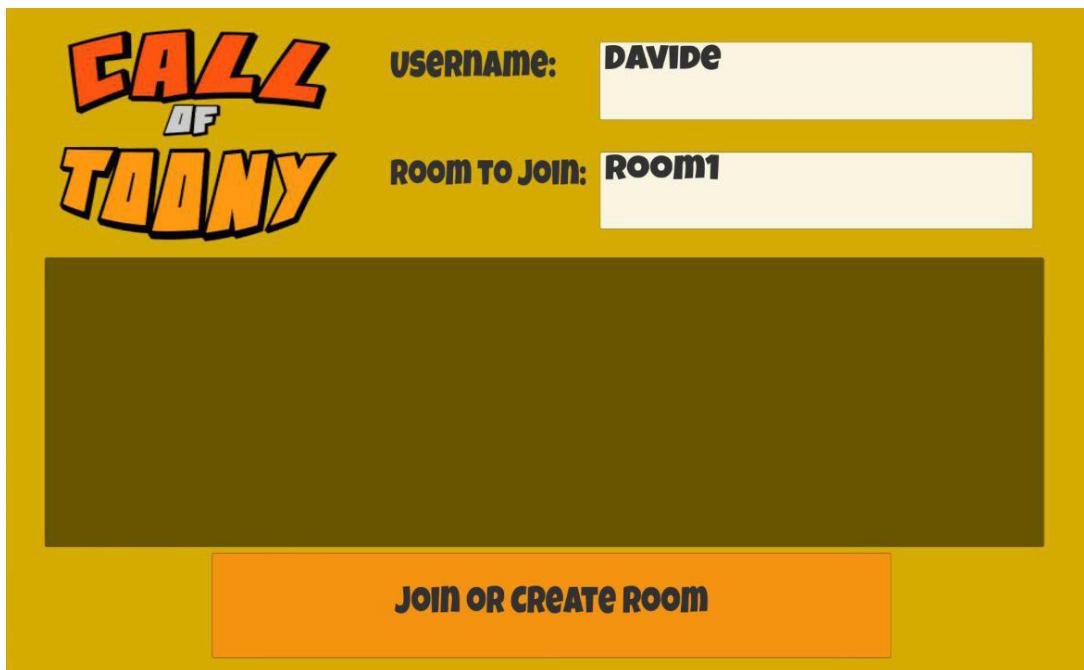


Figura 16: Schermata iniziale di Call Of Toony con Lobby

In seguito, sono stati gestiti gli script relativi al passaggio dei dati attraverso la rete. In particolare i dati passati sono:

- posizione e rotazione dei personaggi rispetto alla mappa;
- percentuale di salute rimasta sulla health bar;
- bersaglio colpito e chi ha colpito il bersaglio.

La gestione di questi dati ha permesso la gestione di un gioco basilare, in cui successivamente è stata applicata un'area in cui rappresentare i log della partita, quali: spawn dei giocatori e le uccisioni avvenute. Inoltre, localmente è stato reso disponibile il numero di morti del proprio personaggio, mentre non c'è stato abbastanza tempo per gestire anche il numero di uccisioni.

Il risultato è stato un gioco fluido e ben strutturato, in grado di gestire il multiplayer ad una latenza bassissima e in grado di coinvolgere il giocatore in un'esperienza nuova e unica. I problemi riscontrati sono dovuti unicamente ai device, i quali a lungo termine presentano surriscaldamento, consumo elevato di batteria e un calo di frame per secondo. Il progetto è piaciuto molto al team tecnico, che non ha escluso la possibilità, dopo un'accurata ottimizzazione, di un inserimento del contenuto in realtà aumentata all'interno dell'app Experenti.

Il nome del gioco è **Call Of Toony**, in richiamo a titoli videoludici del genere ben più noti.

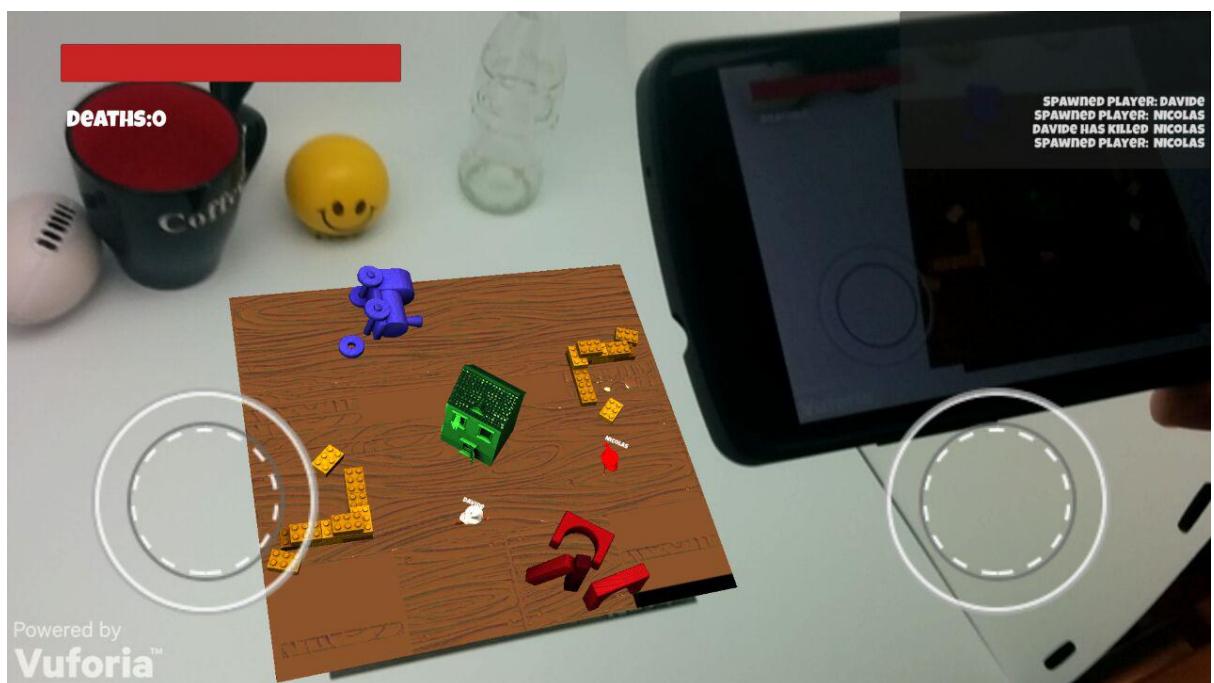


Figura 17: Call Of Toony - multiplayer su device diversi

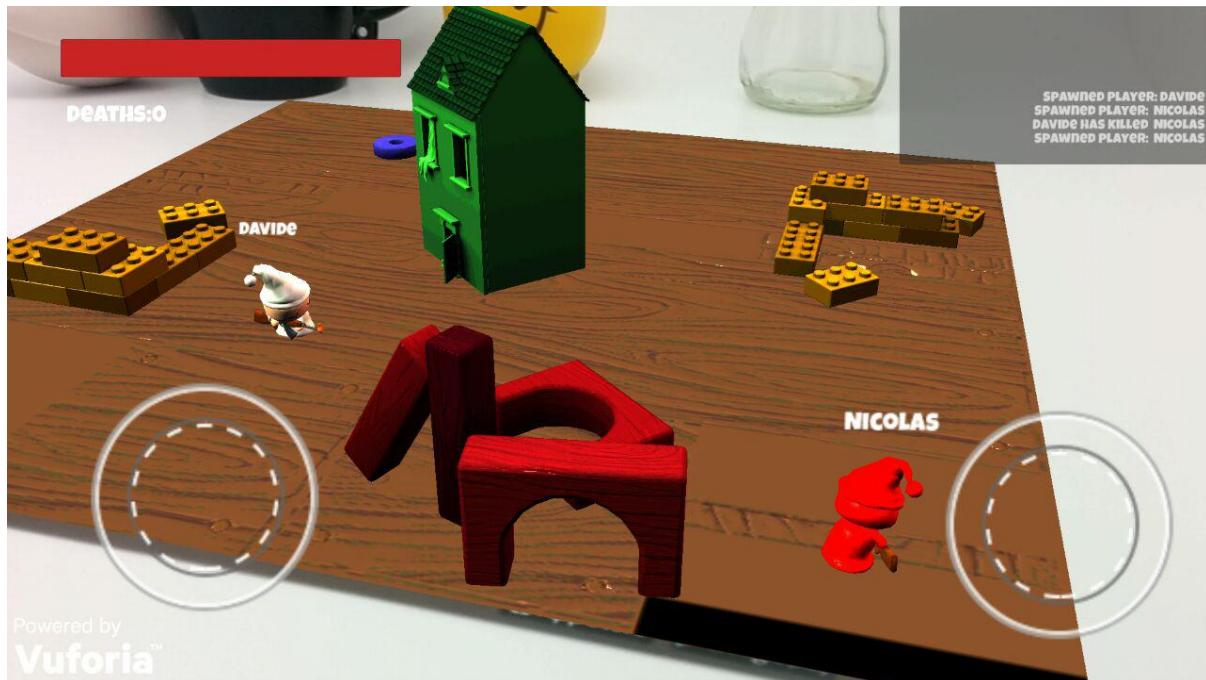


Figura 18: Call Of Toony - schermata di gioco

3.3 Svolgimento delle attività

Inizia ora la stesura di quanto svolto nel progetto vero e proprio, ripercorrendo tutte le fasi del ciclo di vita del software sviluppato fino al suo rilascio sui vari store. Per lo svolgimento del progetto sono state previste circa 120 ore inserite all'interno dell'orario di lavoro aziendale, dal lunedì al venerdì.

Il nome dell'applicazione da sviluppare è Corà Parquet Live.

Lo studio di fattibilità è stato fatto dal Project Manager insieme a un collega del reparto tecnico, ed io non ne ho potuto prendere parte.

3.3.1 Il cliente

Prima di procedere con gli aspetti tecnici è bene avere una panoramica su chi è il cliente che ha commissionato la realizzazione del progetto, quale target di pubblico punta a raggiungere e quali sono le sue aspettative.

Il cliente in questione è **Corà Divisione Parquet**, una divisione dell'azienda **Corà Legnami**, nata nel 1919. Corà Parquet è specializzata nella realizzazione di pavimenti in legno e realizza pavimentazioni per ambienti interni, ambienti esterni e spazi pubblici. L'azienda, leader nel settore, punta a raggiungere i suoi clienti attraverso canali innovativi grazie alla spinta al ringiovanimento voluta da Ettore Corà, amministratore delegato presso Corà Domenico & Figli SpA.

La richiesta del cliente era la realizzazione di un configuratore in realtà aumentata di pavimentazioni in legno, in grado di essere utilizzato inizialmente a scopo fieristico e suc-

cessivamente utilizzabile dai clienti finali dell’azienda in modo da avere uno strumento in grado di fare provare le varie linee di prodotto direttamente nelle case dei possibili clienti. Uno degli obiettivi era quello di fare scaturire nel possibile cliente ciò che viene chiamato “fattore wow”, lasciando sbalordito il cliente e consolidando nella sua mente la propensione dell’azienda verso l’innovazione e il miglioramento. Il secondo obiettivo è dato dal fatto che un’applicazione di un tale livello innovativo inevitabilmente produce un passaparola dagli utenti verso chi ancora non ha visto l’applicazione, generando quindi pubblicità e maggiore visibilità all’azienda.

Per generare questo ”effetto wow”, Corà ha deciso di farsi realizzare un modello tridimensionale di una pin-up con un vestito dilegno. Questo avatar avrebbe dovuto presentare le varie linee di prodotto ed eseguire una introduzione all’applicazione. Il modello e le animazioni non sono state realizzate internamente all’azienda ma sono state delegate a terzi.

Infine, per quanto riguarda il target dell’app, ci si voleva rivolgere all’utente medio, in grado di poter stampare autonomamente il tag a casa propria. Visto che l’utente finale avrebbe dovuto essere la cosiddetta ”signora Maria”, ossia un utente senza particolare propensione per tecnicismi informatici e tecnologia in generale, l’applicazione doveva avere un elevato grado di usabilità e doveva guidare l’utente passo passo.

3.3.2 Analisi dei requisiti

Nella metodologia Agile, l’insieme totale dei requisiti è ottenibile esclusivamente nelle ultime iterazioni, o comunque a progetto quasi ultimato. Questo perchè l’obiettivo principale del modello Agile è quello di ottenere il massimo soddisfacimento del cliente nei tempi e nei costi preventivati.

Alcuni aspetti di rifinitura sono stati quindi concordati in iterazioni intermedie, e sono stati accettati anche se cambiavano molto la struttura del progetto, purchè comunque non si avesse un totale stravolgimento di quanto fatto fino a quel punto.

Dopo un’attenta analisi iniziale, sono stati estratti i casi d’uso principali dalle richieste del committente, da uno studio sull’utenza finale e basandosi sull’user story.

L’analisi dei requisiti è stata svolta insieme alla Project Manager, la quale a mantenuto i contatti con il responsabile d’azienda esterno per tutta la durata del progetto. Insieme a lei abbiamo stilato una lista di funzionalità prioritarie e una lista, invece, di requisiti desiderabili e opzionali.

3 RESOCONTO DELLO STAGE

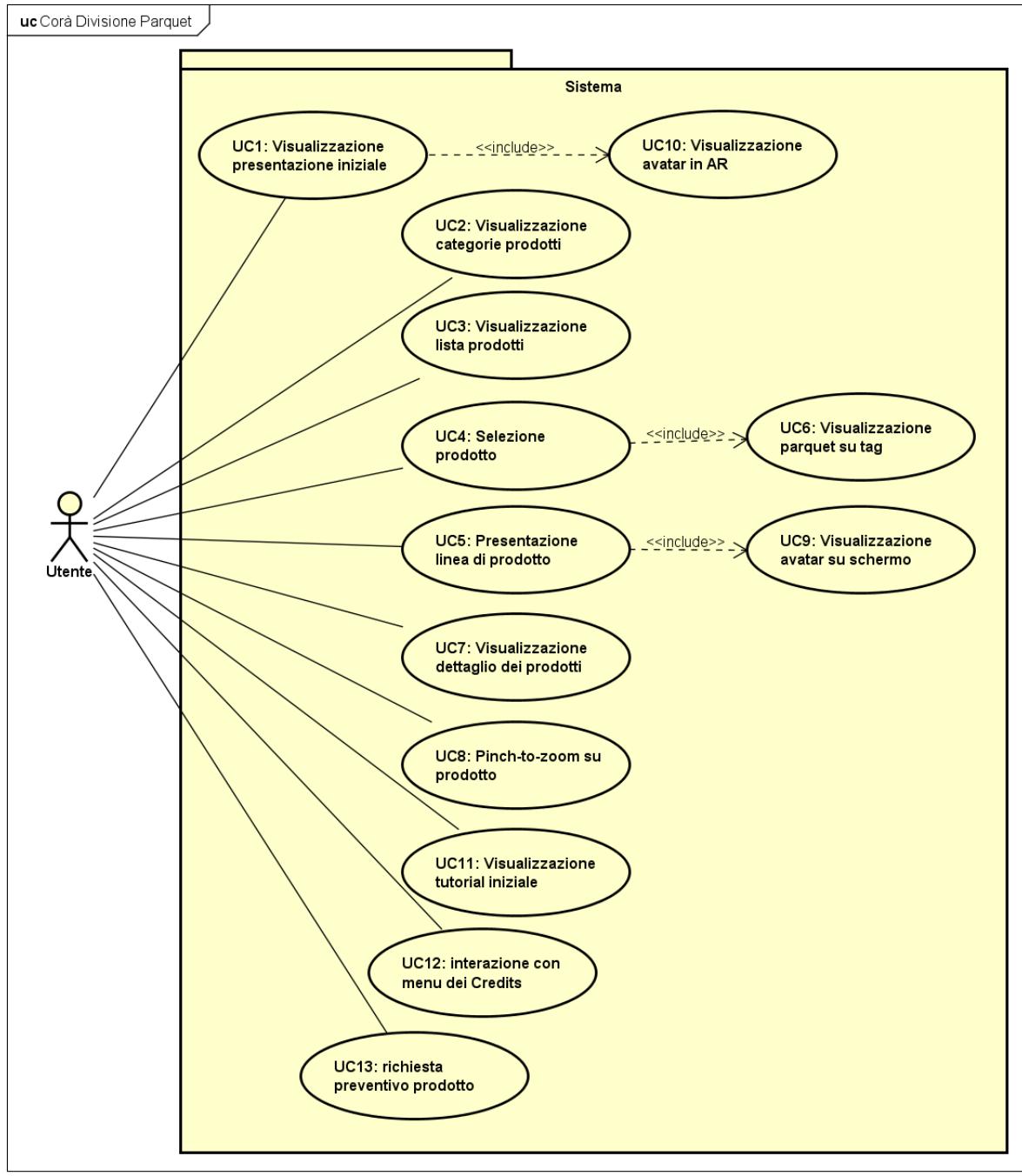


Figura 19: Diagramma dei casi d'uso generico relativo all'applicazione Cora' Parquet Live

Per quanto riguarda i requisiti, essi sono stati suddivisi in 3 categorie: requisiti obbligatori, requisiti opzionali e requisiti desiderabili.

Per quanto riguarda i requisiti opzionali, essi sono stati resi obbligatori dall'applicazione dei vincoli metodologici derivati dalle pratiche in uso aziendali.

Segue ora una lista descrittiva dei principali requisiti. Il presente documento vuole ga-

rantire un discreto livello di dettaglio senza annoiare il lettore, per cui verranno elencate solo le funzionalità principali.

Requisiti obbligatori:

- L'applicazione deve eseguire una presentazione iniziale dell'avatar, il quale deve comparire dietro un ventaglio di legno che si apre e deve fare un breve discorso di apertura.
- L'applicazione deve rendere disponibile un tutorial iniziale che spieghi tutte le funzionalità dell'app.
- L'applicazione deve dare la possibilità all'utente di sfogliare una lista di categorie di prodotto, suddivise alla radice in "Linea di prodotto" e "Colori", che permetteranno di accedere agli stessi prodotti in due modi diversi. La lista deve essere navigabile in entrambe le direzioni (da padre a figlio e da figlio a padre).
- Quando una linea di prodotto viene selezionata deve essere data la possibilità all'utente di avviare una breve presentazione eseguita dall'avatar sulla linea di prodotto scelta. Questo comporta l'apparizione dell'avatar a schermo.
- L'applicazione deve dare la possibilità all'utente di visualizzare i prodotti appartenenti a una certa categoria e deve dare la possibilità di selezionarne uno da visualizzare in realtà aumentata.
- L'utente deve avere la possibilità di selezionare un prodotto da una lista di prodotti e di visualizzarlo in realtà aumentata agganciato al tag. Alla selezione del prodotto deve, inoltre, essere disponibile la possibilità di visualizzare la descrizione del prodotto scelto in un pannello appositamente creato.
- L'applicazione deve permettere, quando un prodotto è selezionato e visibile in realtà aumentata, di effettuare modifiche sulla superficie coperta. In particolare, si vuole rendere disponibile all'utente la funzionalità di pinch-to-scale sull'oggetto parquet.
- L'applicazione rende disponibile un menù di "Credits" in cui è possibile riavviare la presentazione iniziale, rivedere il tutorial, oppure collegarsi alla pagina di download tag, o alla pagina Facebook ufficiale di Corà Divisione Parquet.
- L'applicazione deve dare la possibilità all'utente di richiedere un preventivo online e di scattare uno screenshot della schermata visualizzata.
- L'applicazione deve contenere almeno i primi 30 prodotti inviati dall'azienda.

Requisiti desiderabili:

- L'applicazione deve contenere i 21 prodotti inviati in seconda istanza.
- L'applicazione deve prevedere le gesture di "Swipe" per aprire e chiudere i menù.

3 RESOCONTO DELLO STAGE

- Le texture dei prodotti devono utilizzare uno shader speculare per ottenere un'effetto più lucido.
- La barra del menù di navigazione deve essere ingrandita rispetto alla versione base del configuratore.
- Gli screenshot devono catturare solo la schermata principale con il prodotto in realtà aumentata, il pannello descrittivo del prodotto e il logo di Corà Divisione Parquet, niente altro.

Come già descritto in precedenza, i requisiti hanno visto modifiche anche sostanziali ad ogni iterazione, questo perché la pratica adottata è stata un'analisi iniziale seguita da una breve progettazione e da un intenso periodo di implementazione. Al termine di tale periodo, avveniva prima una verifica completa e poi compilata una demo ed inviata al cliente. Il cliente visionava l'app e inviava di ritorno al Project Manager un feedback con le criticità e le migliorie da apportare. Seguiva, quindi un'ulteriore analisi dei requisiti ed iniziava un nuovo ciclo iterativo.

3.3.3 Progettazione

Nella metodologia Agile, la fase di progettazione è la più importante e difficile da realizzare. Serve infatti molta esperienza per lavorare con una metodologia di questo tipo, in quanto il progetto si ritrova soggetto a continui cambiamenti. Risulta quindi difficile una progettazione solida e duratura, e non si ha il tempo necessario per entrare nel dettaglio, in quanto le ore previste per questa attività sono poche e frequenti.

Come già detto in sezioni precedenti del documento, il configuratore da realizzare doveva partire da un modello di configuratore di base già realizzato e implementante un design pattern architettonico Model View Controller, per quanto Unity ne permetta un'implementazione limitata.

I design pattern sono strumenti che aiutano a risolvere un certo tipo di problema comune. Quindi, è necessaria la presenza di un problema per applicare un design pattern che lo risolva.

Bisogna sottolineare il fatto che Unity si basa pesantemente attorno alle componenti o ai più noti "GameObject", ovvero gli oggetti di gioco, non c'è modo di aggirare questa cosa.

Segue ora una descrizione generica di come è stato progettato il configuratore comprensivo delle funzionalità offerte di base da un configuratore già realizzato e delle parti da me costruite.

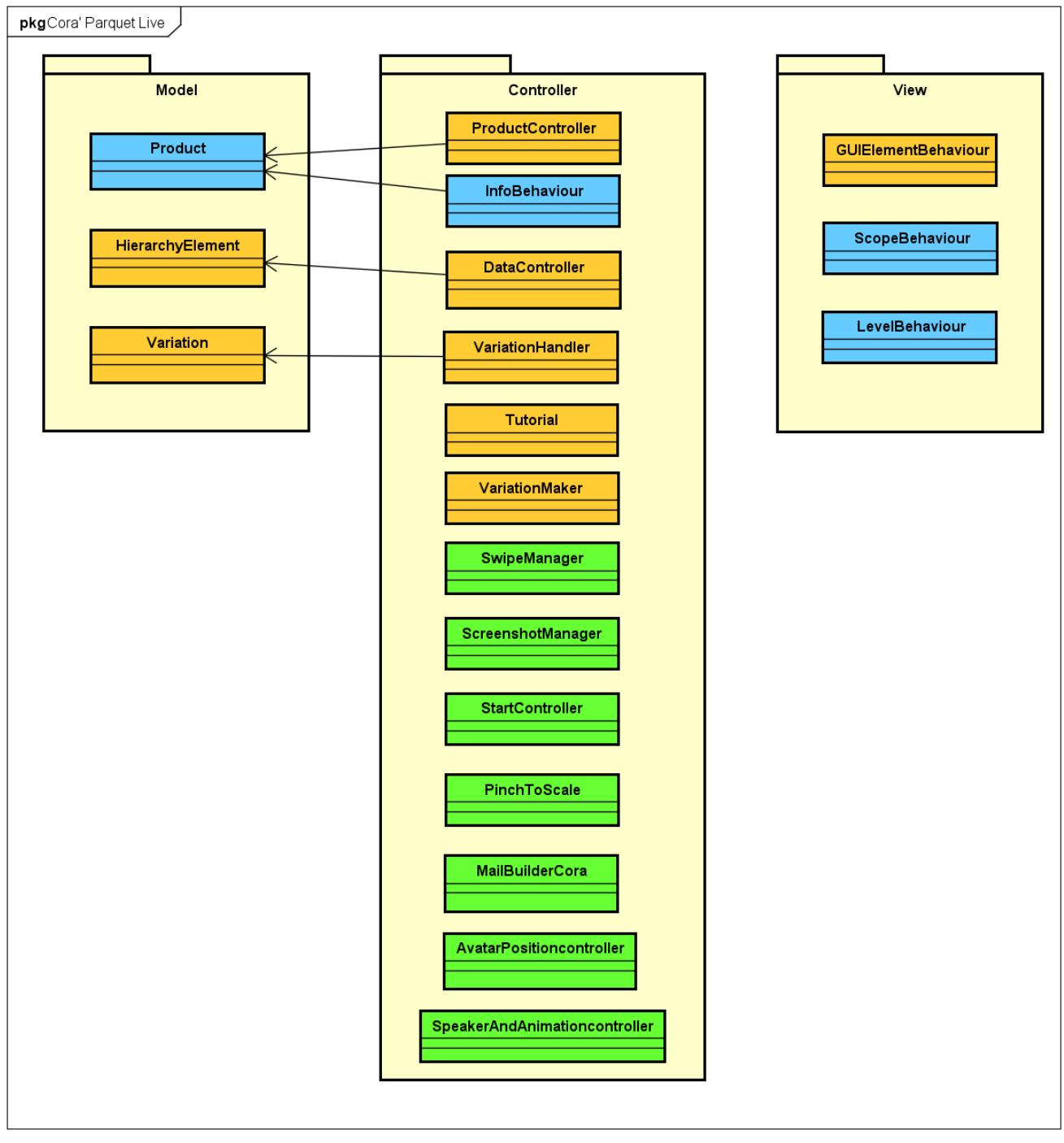


Figura 20: Diagramma delle classi generico

In figura 20 si può notare come sia stata organizzata l'architettura dell'applicazione sulla base di ciò che era già stato realizzato. In particolare, necessita una spiegazione l'etimologia dei colori del diagramma:

- **Giallo:** le classi colorate in giallo rappresentano classi già presenti e realizzate dai colleghi del reparto tecnico a cui non sono state apportate modifiche.
- **Azzurro:** Le classi colorate di azzurro rappresentano le classi già implementate a cui però è stata applicata una modifica non banale.
- **Verde:** in verde le classi create da zero.

Come è facile notare dal diagramma, il modello di gestione dei prodotti del configuratore era già implementato in una sorta di MVC. Il mio compito è stato quello di adattare il codice già sviluppato perché si prendesse carico dell'inserimento e della gestione di un nuovo tipo di contenuto.

Inoltre, il mio compito principale riguardava la gestione dell'avatar all'interno dell'applicazione e la gestione delle gesture.

3.3.3.1 Architettura generale Da subito sono state definite quelle che avrebbero dovuto essere le entità principali all'interno dell'applicazione, ovvero:

- **Product:** Classe rappresentante il modello dati di un prodotto, avente il compito di memorizzare tutte le informazioni relative a un parquet, quali: nome, linea, categoria, finitura, codice, descrizione, essenza, scelta, superficie e dimensioni. Inoltre contiene il nome del padre del prodotto e uno sprite di dimensioni 300x300(px) da usare come icona.
- **ProductController:** Questa classe si occupa di istanziare l'oggetto 3D rappresentante il prodotto vero e proprio una volta che un prodotto viene selezionato. Il modello 3D viene caricato e istanziato runtime dalle risorse disponibili in base al nome dell'oggetto scelto, che quindi dovrà essere univoco tra i vari prodotti e contenere la stringa ".3d" in coda.
- **HierarchyElement:** Classe che si occupa di gestire genitore e figli del prodotto preso in considerazione.
- **DataController:** Questa è la classe centrale su cui si basa tutto il funzionamento del configuratore. Il compito di questa classe è quello di gestire la gerarchia completa di prodotti, permettendo di impostare il nome e l'icona delle varie categorie, e di impostare i genitori e le icone di ogni prodotto. Tutto questo può essere gestito come plugin di Unity semplicemente attaccando lo script a un GameObject vuoto.
- **AvatarPositionController:** Classe principale di gestione e controllo dell'avatar. Qui viene gestito il posizionamento dell'avatar: all'avviamento dell'applicazione l'avatar viene posizionato sopra il tag ad una grandezza fissata, successivamente, una volta terminata la presentazione iniziale, l'avatar viene spostato e fissato a schermo in

una posizione laterale e non troppo ingombrante. Quando l’avatar è fissato a schermo ha una scala posta a zero, mentre viene ingrandito una volta richiamato tramite apposito pulsante. Questo script si occupa anche di gestire la posizione e l’apparizione del ventaglio iniziale per poi invocare il metodo dedicato alla presentazione vera e propria dell’avatar.

- **StartController:** E’ la classe che si occupa di gestire la presentazione iniziale dell’avatar. Qui vengono caricati i file audio e temporizzati con l’inizio delle animazioni. La classe si occupa di gestire sia l’avvio sia il termine (anche richiesto dall’utente) dell’applicazione.
- **SpeakerAndAnimationController:** Classe realizzata con lo scopo di gestire l’avatar in modo da fargli presentare le categorie tramite animazioni e file audio. Non era conosciuto a priori il numero di animazioni che si sarebbero dovute implementare per cui questa classe doveva essere progettata in modo da gestire un numero indefinito di animazioni e file audio.
- **PinchToScale:** Questa classe non si occupa di riconoscere la gestura, compito dedicato a una classe sviluppata da terzi, ma si occupa di definire il comportamento quando viene rilevata una particolare gestione sul modello 3D del prodotto.
- **SwipeManager:** Questa classe, come la precedente, non è un riconoscitore di gesture ma un gestore che si occupa di definirne il comportamento. Lo SwipeManager deve aprire e chiudere i menù in base al tipo di swipe effettuato.
- **ScreenshotManager:** Si occupa di gestire la cattura e il salvataggio della schermata visualizzata sul display. Lo script si affida all’utilizzo di classi dedicate sviluppate da terzi e ha il compito di gestire cosa deve essere visualizzato nello screenshot.
- **CoraMailBuilder:** E’ lo script che si occupa della creazione della mail di richiesta preventivo su un determinato prodotto
- **LevelBehaviour:** Questa classe si occupa di popolare il menù inferiore dell’applicazione, creando i pulsanti per ogni categoria e i pulsanti relativi ai prodotti.

Questi sono gli script principali utilizzati nella realizzazione del progetto. Come già detto, sono stati creati altri script più dedicati e di rifinitura, la cui spiegazione in dettaglio non farà parte di questo documento per non appesentirne la lettura.

3.3.3.2 Layout e GUI Per quanto riguarda la GUI dell’applicazione, essa è stata progettata da terzi per conto di Corà Divisione Parquet e mi è stata inviata tramite un file PSD contenente le grafiche da utilizzare e il layout a cui adeguarsi.

Il layout è stato scelto sulla base dei configuratori precedentemente utilizzati ed è stato adattato per il progetto specifico.



Figura 21: Layout con menu' info aperto



Figura 22: Layout con menu inferiore aperto

Di seguito vengono descritti i componenti principali della GUI.

- **Logo:** il logo è un pulsante cliccabile che apre il menù dei Credits nel centro dello schermo;
- **Menù inferiore:** contiene la lista delle categorie, con eventualmente la relativa icona, e la lista dei prodotti come figli della propria categoria madre. Il menù è composto da una "breadcrumb" navigabile, e da una sequenza di bottoni assegnati ognuno ad una categoria, e nei livelli più profondi ai prodotti.
- **Pannello Info:** è un pannello laterale che compare dopo la pressione del pulsante "i". Il pulsante in questione diventa visibile solo quando un prodotto è selezionato e permette di aprire la sezione relativa alla descrizione del prodotto. Il pannello si compone anche di un pulsante per l'acquisizione di screenshot e un pulsante per la richiesta di preventivo del prodotto visualizzato.
- **Pulsante animazione:** è un pulsante a forma di fumetto situato nella parte a est dello schermo, e diventa visibile ogni volta che l'avatar ha qualcosa da dire. Nello specifico, compare quando viene selezionata una categoria interna alle "linee di prodotto", e scompare negli altri casi. La comparsa è seguita da un'animazione di notifica per richiamare l'attenzione dell'utente. Una volta premuto il pulsante, compare l'avatar nella parte ovest dello schermo iniziando l'animazione dedicata.
- **Scope:** mirino posizionato al centro dello schermo e utilizzato per fornire brevi istruzioni all'utente su cosa fare e dare la possibilità all'utente di scaricare il tag, oltre al fatto di eseguire la sua funzionalità principale di supporto per mirare il tag.

L'apertura del menù, dei credits e del pannello laterale non è mutualmente esclusiva, in quanto tutte e tre le componenti possono essere visibili in stato aperto contemporaneamente senza interferire tra di loro.

In questo capitolo ho avuto voce, in quanto ho potuto esprimere il mio dissenso su alcune scelte relative all'usabilità dell'applicazione mantenendo un certo tipo di layout. Queste osservazioni e richieste di modifica della grafica sono state concordate prima con Project manager e successivamente con il cliente. Le modifiche osservate e accolte sono state:

- Grandezza del pannello delle informazioni laterali. Essendo troppo piccolo risultava inutilizzabile da smartphone.
- Stile dell'icona dell'applicazione che risultava poco leggibile.
- Posizione del pulsante animazione. Inizialmente posto nella parte superiore dello schermo, risultava difficile da raggiungere con un'impugnatura standard del device.

Inoltre, ho proposto l'utilizzo di un tag di grandezza A3, al posto del foglio A4 utilizzato per ottenere un miglior risultato di agganciamento del tag e di tracking. Questa richiesta non è stata accolta in quanto si presuppone la stampa domestica del tag in assenza di possibilità di stampare fogli A3.

3.3.3.3 Animator Controller Gli Animator Controller sono i gestori delle animazioni di un GameObject in Unity. Generalmente, ogni oggetto che si vuole animare ha attaccato un Animator con associato un Animator Controller che ne definisce il comportamento. In questo progetto, tali componenti hanno avuto un ruolo fondamentale, per cui è stata necessaria un'attenta progettazione di queste componenti.

In questo paragrafo verranno tralasciate le spiegazioni degli Animation Controller banali.

- **Info Panel Controller:** controller dedicato al pannello laterale per gestire le animazioni entrata, uscita e di oscurazione del pulsante "i". Esempio utile per capire come ogni oggetto della GUI è stato gestito in modo simile. Ogni animazione viene fatta partire al cambio di stato di un valore booleano oppure all'azionamento di un particolare trigger.

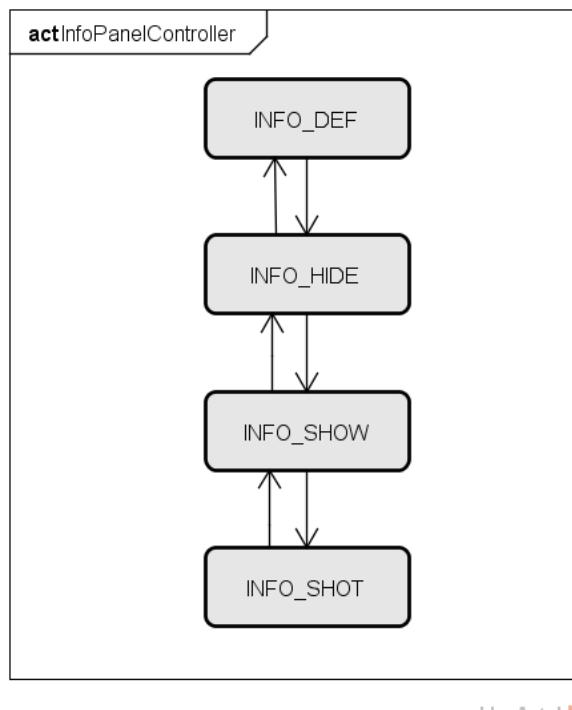


Figura 23: Animator Controller del pannello Info laterale

Nel dettaglio: INFO_DEF nasconde sia pannello che pulsante, INFO_HIDE mostra il pulsante "i", INFO_SHOW apre il pannello su schermo, INFO_SHOT, rimuove gli elementi della GUI indesiderati per l'esecuzione di uno screenshot.

- **Avatar Speaker Controller:** dedicato alla gestione della presentazione iniziale da parte dell'avatar e alle animazioni relative alla spiegazione delle categorie.

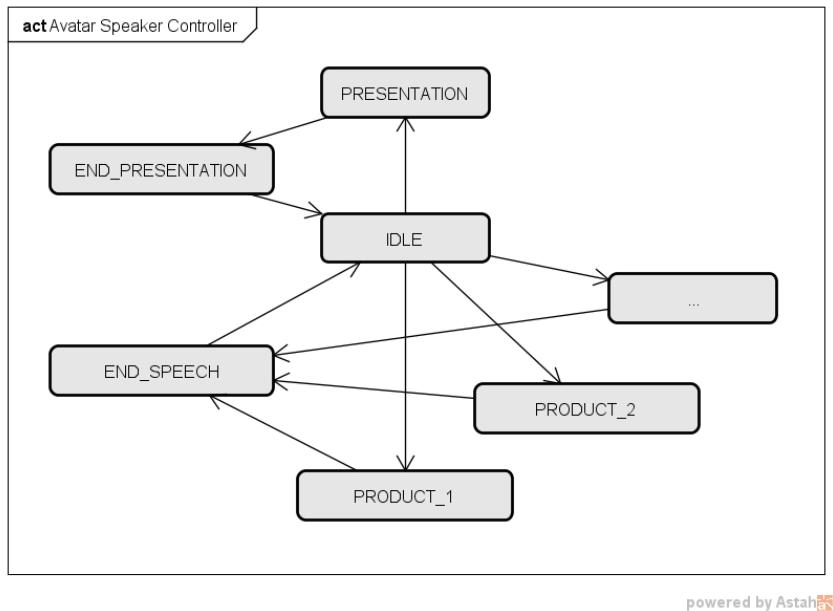


Figura 24: Animator Controller dell'avatar

Nel dettaglio: IDLE è lo stato iniziale, a cui l'avatar ritorna alla fine di ogni presentazione. Lo stato PRESENTATION è un particolare stato in cui l'animator viene portato nel momento in cui si vuole avviare la presentazione iniziale. Questo stato termina poi con il passaggio a END_PRESENTATION che è uno stato necessario per effettuare alcuni controlli da codice, come controllare che la presentazione sia terminata. Lo stesso viene fatto con le n categorie di prodotti di cui si ha la presentazione.

Questo animator poteva essere implementato in modo più efficiente caricando l'animazione da fare partire di volta in volta in base a una stringa passata. Avendo tempo limitato si è preferito procedere per una strada di facile implementazione come questa.

3.3.4 Implementazione

L'attività di implementazione e di codifica si è svolta secondo quanto previsto dalla metodologia Agile. Le fasi di codifica non sono avvenute in un unico round, ma in brevi e frequenti blocchi di tempo. Per prima cosa, sono state implementate le parti principali del sistema, cioè le parti che quasi sicuramente non avrebbero avuto modifiche nelle successive iterazioni.

Si è proceduto, in primo luogo, con l'adattamento di quanto disponibile nel configuratore base. In particolare è stata creata una classe Product in grado di contenere tutte le informazioni necessarie per ogni prodotto.

3.3.4.1 Pinch-To-Scale E' stata richiesta la possibilità di espandere il parquet mostrato in realtà aumentata direttamente runtime tramite l'applicazione di una gestione

3 RESOCONTO DELLO STAGE

dedicata.

Bisogna premettere che le texture dei parquet sono state recapitate simulandone la grandezza reale per cui, data anche l'irregolarità di alcuni parquet, era difficile fare il tiling (ripetizione della texture su uno o piùassi) della texture.

Su suggerimento del CTO si è deciso di procedere mantenedo la dimensione reale della texture, senza tiling, e applicando una maschera ridimensionabile all'oggetto in modo tale da visualizzarne solo una parte. La realizzazioen è stata affidata completamente a me. E' stata realizzata una maschera a forma di cornice, unendo più oggetti tridimensionali prefabbricati. A tale maschera è stato poi applicato un particolare shader in grado di rimuovere il rendering delle parti di texture coperte dalla maschera.

Di seguito è illustrata la codifica relativa allo script PinchToScale.

```
2 public class PinchToScale : MonoBehaviour
{
4     public Vector3 scaleWeights = Vector3.one;
5     public float minScaleAmount = 0.4f;
6     public float maxScaleAmount = 4.0f;
7     public float sensitivity = 0.3f;      // augment for a bigger scaling
8     public float smoothingSpeed = 12.0f;    // set to 0 to disable smoothing
9     public GameObject mask;
10
11    private float idealScaleAmount = 0.4f;
12    private float scaleAmount = 0.4f;
13    private Vector3 baseScale = Vector3.one;
14
15    public float ScaleAmount
16    {
17        get { return scaleAmount; }
18
19        set
20        {
21            value = Mathf.Clamp( value, minScaleAmount, maxScaleAmount ); //check
22            if the value is between limits
23
24            if( value != scaleAmount )
25            {
26                scaleAmount = value;
27
28                Vector3 s = scaleAmount * baseScale;
29                s.x *= scaleWeights.x;
30                s.y *= scaleWeights.y;
31                s.z *= scaleWeights.z;
32
33                mask.transform.localScale = s;          //resize the mask scale
34            }
35        }
36
37    public float IdealScaleAmount
38    {
39        get { return idealScaleAmount; }
40        set { idealScaleAmount = Mathf.Clamp( value, minScaleAmount,
```

```

        maxScaleAmount ) ;
    }

42 void Start()
{
    baseScale = transform.localScale ;
    IdealScaleAmount = ScaleAmount ;
}

48 void Update()
{
    if ( smoothingSpeed > 0 ) {
        ScaleAmount = Mathf.Lerp ( ScaleAmount , IdealScaleAmount , Time.deltaTime
        * smoothingSpeed ) ;
    } else {
        ScaleAmount = IdealScaleAmount ;
    }
}

56

58 void OnPinch( PinchGesture gesture )
{
    IdealScaleAmount += sensitivity * gesture.Delta.Centimeters() ;
}

```

Il codice, all'avvio dello script, memorizza la scala locale dell'oggetto, e imposta il valore di scaling ideale allo scaling di base. Ad ogni frame viene aggiornato, internamente al metodo Update, lo scaling attuale, sia nel caso sia presente un effetto di smooth sia altrimenti. Quando viene catturata una gestione di Pinch, ne viene presa l'ampiezza in centimetri e moltiplicata per la sensibilità che gli si vuole dare allo scaling rispetto alla gestione. Questo valore calcolato viene memorizzato in IdealScaleAmount che nel successivo Update sarà memorizzato come nuovo valore di ScaleAmount. ScaleAmount ogni volta che viene modificato si occupa di applicare lo scaling impostato alla maschera passata come parametro da Unity.

3.3.4.2 Inserimento prodotti Successivamente ho proceduto con l'inserimento dei primi 20 prodotti, in quanto il modello e la gestione delle categorie e dei prodotti era già implementata. Per ogni prodotto, prima si è dovuta importare negli assets la relativa texture e icona, poi, dopo aver attaccato uno script Product al prodotto comprensivo del modello 3D, questo veniva processato in uno script (VariationMaker.cs) che creava 2 oggetti: un GameObject contenente le informazioni del prodotto da utilizzare nello script di gestione dei dati (DataController.cs) e un GameObject con il modello 3D.

3.3.4.3 Gestione dell'avatar Come già accennato, il grosso del lavoro consisteva nell'implementazione di un avatar all'interno dell'applicazione. Per prima cosa è stata gestita la presentazione iniziale composta da 4 file di animazione e 4 file audio. E' stata

3 RESOCONTO DELLO STAGE

usata la tecnica del lip sync per ottenere un effetto realistico. Di ciò se ne è occupato lo StartController sviluppato appositamente.

Lo StartController è stato implementato utilizzando il design pattern Singleton in quanto era necessario avere al più un'istanza della classe.



Figura 25: Presentazione iniziale

All'avvio dello StartController vengono caricati in memoria gli audio file relativi alla presentazione. Durante ogni ciclo di Update, invece, viene controllato lo stato dell'animazione corrente e il numero contenuto all'interno di un particolare contatore. In questo modo è possibile eseguire nell'ordine corretto i vari pezzi ottenendo, così, l'animazione completa, senza che venga ciclata più volte essendo che il contatore procede verso un'unica direzione incrementale e viene ripristinato solo quando si vuole riavviare la presentazione.

Di seguito viene mostrata la codifica del metodo Awake() che si occupa del caricamento dei file audio.

```
1 void Awake() {
2     _instance = this;
3     pt1 = (AudioClip)Resources.Load ("Speakeraggio/01_Presentazione1",
4         typeof( AudioClip));
5     pt2 = (AudioClip)Resources.Load ("Speakeraggio/02_Presentazione2",
6         typeof( AudioClip));
```

```

5   pt3 = (AudioClip)Resources.Load ("Speakeraggio/03_Presentazione3",
6     typeof(AudioClip));
7   pt4 = (AudioClip)Resources.Load ("Speakeraggio/04_Presentazione4",
8     typeof(AudioClip));
9 }
```

E il metodo Update() che si occupa della gestione delle animazioni e della sincronizzazione con i file audio. Inoltre si può notare come venga reso disponibile un pulsante "Skip" per saltare la presentazione.

```

1 void Update(){
2   if (avatarController.GetCurrentAnimatorStateInfo (0).IsName ("01
3     _Presentazione1") && startLock == 1) {
4     presentationAudio.clip = pt1;
5     presentationAudio.Play();
6     skipButton.SetActive(true);
7     skipButton.GetComponent<Animator> ().SetBool ("Show", true);
8     startLock++;
9   }
10  if (avatarController.GetCurrentAnimatorStateInfo (0).IsName ("02
11    _Presentazione2") && startLock == 2) {
12    presentationAudio.clip = pt2;
13    presentationAudio.Play();
14    startLock++;
15  }
16  if (avatarController.GetCurrentAnimatorStateInfo (0).IsName ("03
17    _Presentazione3") && startLock == 3) {
18    presentationAudio.clip = pt3;
19    presentationAudio.Play();
20    startLock++;
21  }
22  if (avatarController.GetCurrentAnimatorStateInfo (0).IsName ("04
23    _Presentazione4") && startLock == 4) {
24    presentationAudio.clip = pt4;
25    presentationAudio.Play();
26    startLock++;
27  }
28  if (avatarController.GetCurrentAnimatorStateInfo (0).IsName ("EndPresentation")
29    && startLock == 5) {
30    EndPresentation ();
31    startLock = 1;
32  }
33 }
```

Il passo di implementazione successivo è stato la gestione della posizione dell'avatar. All'inizio doveva risultare agganciato al tag per poi spostarsi e rimanere agganciato alla parte ovest dello schermo, comparando solo su richiesta.

Questo script è quello che più ha usufruito della metodologia Agile, e in particolare quello che ha subito più incrementi nelle varie iterazioni.

Questo perchè il cliente, non essendo un esperto di realtà aumentata e, soprattutto di

3 RESOCONTO DELLO STAGE

applicazioni Android, non aveva un'idea precisa di come e dove dovesse apparire l'avatar. Per cui sono state richieste numerose prove per ottenere il risultato che più rendesse felice il cliente.

In iterazioni intermedie era stata prevista persino un'interessante funzionalità per cui il device calcolava l'altezza rispetto al tag e modificava la grandezza dell'avatar per una visione "da alzati" e una "da seduti". Questa funzionalità da me realizzata è piaciuta molto al team di sviluppo ma è stata bocciata dal cliente finale che, alla fine, è stato convinto del Project Manager della soluzione ottimale con avatar di grandezza fissa.

Questo script è fondamentale perché gestisce le animazioni dei vari componenti che agiscono nella presentazione iniziale e le sincronizza per ottenere il miglior effetto visivo.

L'ultimo script realizzato per il controllo dell'avatar è lo script che si occupa di gestire le presentazioni delle varie categorie, chiamato SpeakerAndAnimationController. Nel dettaglio, il pulsante di una categoria invoca, alla pressione, un metodo chiamato SetPresentation(string) che si occupa di caricare in memoria l'audio corrispondente effettuando una ricerca per nome negli assets. Nel caso la ricerca termini correttamente viene mostrato il pulsante animazione (pulsante a forma di fumetto), altrimenti viene nascosto, se già visibile, o non mostrato.

Alla pressione del pulsante animazione viene invocato un metodo dedicato allo start e allo stop della presentazione. In particolare, se esiste già un'audio e un'animazione in esecuzione da parte dell'avatar, e viene premuto tale pulsante, l'animazione viene fermata. Se invece non è presente alcuna presentazione in esecuzione, viene fatto partire l'audio in memoria con la relativa animazione, facendo comparire l'avatar con un effetto di scaling. E' utile osservare il funzionamento del metodo Update() per capire come avviene la sincronizzazione tra audio e animazione.

```
1 void Update(){
2     if (avatarController.GetCurrentAnimatorStateInfo (0).IsName ("EndSpeech"))
3     {
4         StartCoroutine(scaleIn( new Vector3 (100f, 100f, 100f), new Vector3(0f
5 ,0f,0f) , 1.5f));
6         hasAudioBeenPlayed = false;
7         instance.resetAnimation (active);
8     }
9     if (avatarController.GetCurrentAnimatorStateInfo (0).IsName ("Idle") &&
10        active != "") {
11         AudioClip audio = (AudioClip)Resources.Load ("Speakeraggio/" + active ,
12             typeof(AudioClip));
13         activeAudio.clip = audio;
14     }
15     if (!hasAudioBeenPlayed && !string.IsNullOrEmpty(active) &&
16         avatarController.GetCurrentAnimatorStateInfo (0).IsName (active)) {
17         hasAudioBeenPlayed = true;
18         activeAudio.Play ();
19     }
20 }
```

Come si può notare, viene fatto un controllo sullo stato in esecuzione dell'Animator controller. In particolare, se ci si trova in uno stato di fine presentazione, viene data la possibilità di far partire un nuovo audio e viene nascosto l'avatar. Se l'avatar si trova nello stato iniziale, viene caricato l'audio corrente, mentre se l'animazione relativa all'audio è attiva e l'audio può essere eseguito viene effettuato un play, che avviene in modo sincronizzato con l'animazione.

3.3.4.4 Gestione della GUI La GUI è l'aspetto del progetto che più ha subito modifiche. Inizialmente, per trovare la resa grafica migliore a livello estetico e, successivamente, per garantire il più alto livello possibile di usabilità. Il problema principale è stato identificare il contesto di utilizzo dell'applicazione e decidere se ottimizzare l'interfaccia per smartphone o tablet, in quanto la creazione di due interfacce ad-hoc avrebbe avuto un costo aggiuntivo per il cliente. Si è deciso di ottimizzare l'app per smartphone, in quanto più reperibile. Lo sviluppo della GUI è avvenuto seguendo come linea guida un file PSD contenente le varie schermate e da cui era possibile prendere le grafiche (icone, pulsanti, etc.)

3.3.4.5 Implementazione del tutorial Come ultimo passo di implementazione, ho proceduto con la creazione del tutorial, prendendo le grafiche da una sezione del file PSD dedicato e modificando del codice già creato per configuratori precedentemente realizzati. Il tutorial è stato pensato in modo tale da procedere con l'avanzamento grazie a un tap sullo schermo ed è stata prevista una funzionalità per saltare il tutorial ed andare direttamente alla parte del configuratore.

3.3.5 Verifica e validazione

Contestualmente all'applicazione sviluppata, i principali test svolti hanno riguardato il corretto funzionamento di tutte le funzionalità rese disponibili. L'attività di verifica e validazione è stata svolta quasi totalmente da alcuni colleghi del team tecnico e dal Project Manager, che hanno potuto valutare, durante tutto lo svolgimento del progetto, la correttezza dei componenti da me sviluppati.

I test avvenuti hanno avuto sempre esito positivo riguardo il corretto funzionamento dei componenti e hanno portato alla luce piccoli bug prontamente corretti.

Oltre ai test interni, ad ogni iterazione veniva fornita l'app a un responsabile esterno dell'azienda Corà Divisione Parquet, che aveva il compito di effettuare ulteriori test sulla qualità del prodotto e doveva verificare la corrispondenza di funzionalità tra le aspettative e l'applicazione reale.

I feedback da parte del cliente sono sempre stati positivi su quanto sviluppato, tranne per il lavoro svolto sull'avatar che, almeno inizialmente, non corrispondeva alle loro aspettative. Con successive iterazioni si è riuscito a ottenere un effetto ottimale dell'avatar che portasse soddisfazione al cliente.

I device utilizzati per il testing dell'applicazione sono stati:

- Google Nexus 5 (Android);

- Samsung Note 4 (Android);
- Samsung Galaxy S3 (Android);

Per quanto riguarda, invece, il testing dell'app durante l'attività di implementazione, è stato usato un plugin di Vuforia per Unity che permette l'utilizzo di una webcam per simulare la fotocamera dei device.

3.4 Livello di completezza raggiunto

Alla fine del tempo di stage si può stimare un completamento del progetto intorno circa al 98% rispetto al totale, in quanto l'ultima richiesta da parte del cliente prima della pubblicazione dell'app è stata quella di implementare il follow di occhi e testa dell'avatar rispetto alla telecamera. Questo ultimo aspetto è stato sviluppato interamente da un collega del reparto tecnico dopo la conclusione del mio stage.

4 Valutazione Retrospettiva

Segue ora una valutazione personale sul periodo di stage trascorso.

4.1 Soddisfacimento obiettivi

Lo stage si e' svolto nel completo rispetto di tutti i vincoli imposti, concludendosi in data 11/09/2015.

Per quanto riguarda gli obiettivi dello stage, essi sono variati durante lo svolgimento in quanto non era garantita la disponibilita' di seguire un progetto commissionato dall'esterno.

Come obiettivo massimo è richiesto di sviluppare un'intera app visore di AR, completa di tutti i suoi contenuti semplici e complessi e della propria grafica, dalla fase di accettazione dei materiali in entrata fino alla fase di consegna della beta finale al cliente.

Questo obiettivo e' stato pienamente raggiunto, anzi si e' quasi arrivati al release dell'applicazione sugli store, avvenuto pochi giorni dopo il termine del mio stage.

Tutti i requisiti dell'applicazione sono stati soddisfatti e i contenuti richiesti implementati al meglio delle mie attuali competenze.

L'applicazione realizzata e' piaciuta molto a Cora' Divisione Parquet e soprattutto all'amministratore delegato Ettore Cora', il quale si e' complimentato per l'ottimo lavoro svolto. E' certo che il progetto verra' ampliato con nuovi contenuti e nuovi prodotti continuando a usufruire dei servizi offerti da Experenti.

4.2 Conoscenze acquisite

E' difficile descrivere la totalita' delle cose che ho imparato nel periodo di due mesi di stage effettuato, in quanto e' stata un'esperienza completamente nuova e totalmente diversa da quanto gia' appreso nei vari progetti didattici di team.

A livello tecnologico ho potuto arricchirmi di strumenti e tecnologie innovative e all'avanguardia che sicuramente saranno sempre piu' richieste a livello curricolare con la prossima uscita di device portatili di ultima generazione e visori di AR/VR. In particolare:

- Unity:
- Vuforia:
- C#:

Ma non solo, in quanto la crescita piu' grande e' avvenuta nell'applicazione di processi aziendali e nell'applicazione del lavoro di squadra.

4.3 Distanza tra universita' e lavoro

4.4 Valutazione personale

4.5 Screenshot finali



Figura 26: Presentazione iniziale - Ventaglio di legno in apertura



Figura 27: Presentazione iniziale - Avatar



Figura 28: Tutorial - Istruzioni

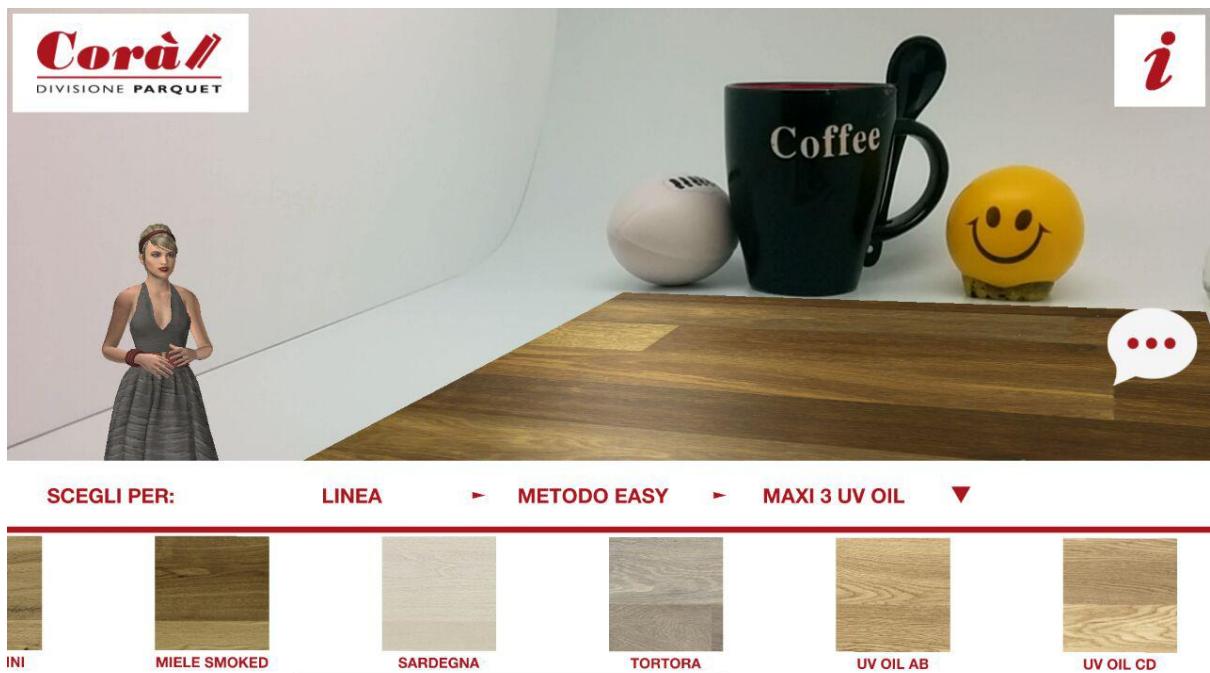


Figura 29: Configuratore - Menu' inferiore aperto e Avatar in presentazione



Figura 30: Configuratore - Pannello info aperto



Figura 31: Configuratore - Menu' dei Credits aperto

A Realta' Aumentata

La realtà aumentata consiste nell'arricchimento della percezione sensoriale umana mediante informazioni, in genere manipolate e convogliate elettronicamente, che non sarebbero percepibili con i cinque sensi. Gli elementi che "aumentano" la realtà possono essere aggiunti attraverso un dispositivo mobile, come uno smartphone, con l'uso di un PC dotato di webcam o altri sensori, con dispositivi di visione (per es. occhiali a proiezione sulla retina), di ascolto (auricolari) e di manipolazione (guanti) che aggiungono informazioni multimediali alla realtà già normalmente percepita.

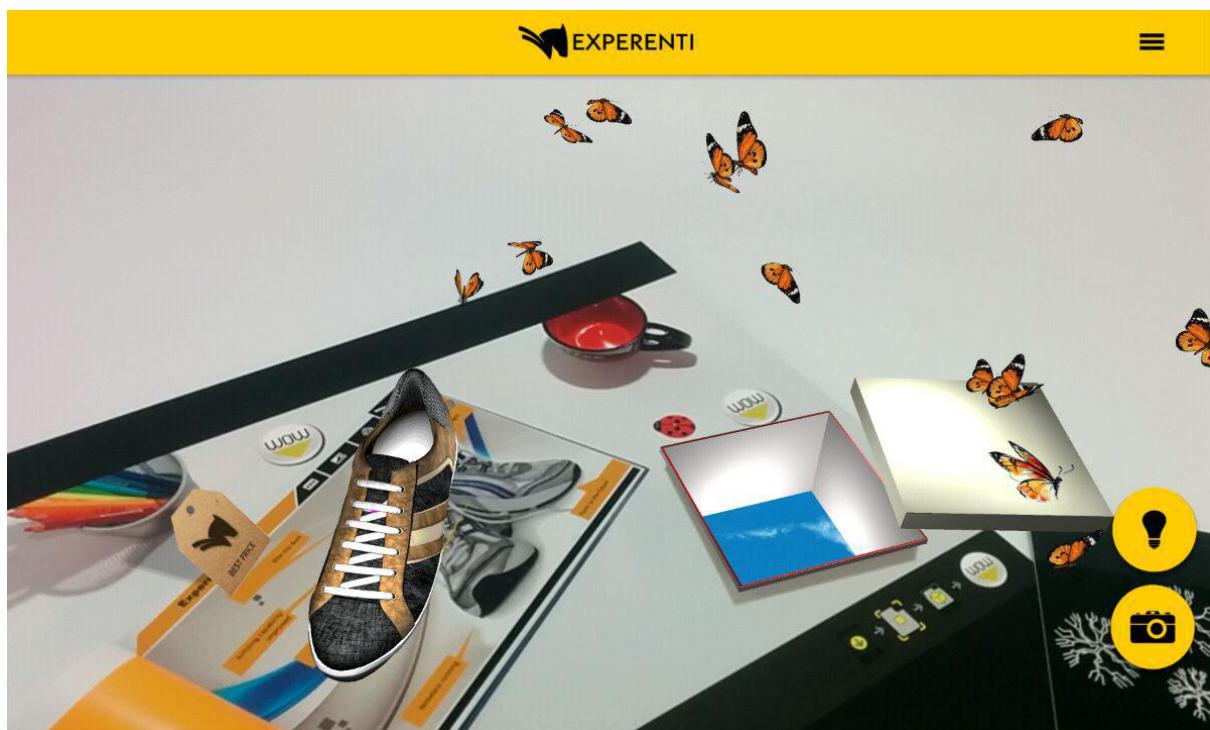


Figura 32: Esempio di realta' aumentata

Le informazioni "aggiuntive" possono, però, consistere anche in una diminuzione della quantità di informazioni normalmente percepibili per via sensoriale, sempre al fine di presentare una situazione più chiara o più utile o più divertente. Anche in questo caso si parla di AR.

Nella realtà virtuale (virtual reality, VR), le informazioni aggiunte o sottratte elettronicamente sono preponderanti, al punto che le persone si trovano immerse in una situazione nella quale le percezioni naturali di molti dei cinque sensi non sembrano neppure essere più presenti e sono sostituite da altre. Nella realtà aumentata (AR), invece, la persona continua a vivere la comune realtà fisica, ma usufruisce di informazioni aggiuntive o manipolate della realtà stessa.

Le informazioni circa il mondo reale che circonda l'utente possono diventare interattive e manipolabili digitalmente. Le informazioni che "aumentano" la realtà possono

A REALTA' AUMENTATA

essere presenti nella memoria del dispositivo utilizzato, oppure possono essere ricavate da internet in tempo reale.

Prima di essere impiegata in ambito mobile, con applicazioni per smartphone e tablet o visori da indossare, la realtà aumentata è stata introdotta in ambiti specifici come quello della ricerca, della medicina o nel settore militare. Basti pensare, ad esempio, agli head-up display (HUD) equipaggiati sugli aerei da combattimento, che mostrano al pilota informazioni come la distanza dall'obiettivo o l'inclinazione del velivolo, permettendogli di mantenere lo sguardo fisso su ciò che ha di fronte. In tempi recenti una delle prime app mobile a sfruttare questo approccio è stata Layar. Si tratta di un software che, sfruttando le informazioni di geolocalizzazione fornite dal modulo GPS del dispositivo, e accoppiandole con l'orientamento dello schermo individuato da accelerometro o giroscopio, permette all'utente di inquadrare attraverso la fotocamera l'ambiente circostante, visualizzando icone relative ai punti di interesse presenti nelle vicinanze, esattamente nella direzione in cui si trovano. Questo può risultare utile quando si cerca un ristorante, per capire che strada percorrere per raggiungerlo, oppure in modo da sapere in tempo reale la posizione di altre persone nei dintorni.

La realtà aumentata è una tecnologia applicabile a molti contesti diversi (contrariamente alla realta' virtuale che trova le sue principali applicazioni in ambito gaming e multimediale). I principali campi in cui puo' essere implementata spaziano dall'advertising al gaming, dall'edilizia all'arte e all'istruzione.

Glossario

A

Asset: E' una risorsa utilizzata in Unity. Un asset puo' essere, per esempio, un modello 3D, uno sprite, una texture, o qualsiasi oggetto di gioco (game object).

Avatar: E' un modello 3D o un'immagine scelta per rappresentare un'entita' in ambientazioni virtuali.

B

Business: Insieme delle attivita' che contribuiscono maggiormente alla produzione del fatturato.

C

Client-server: La presenza di un server permette ad un certo numero di client di condividerne le risorse, lasciando che sia il server a gestire gli accessi alle risorse per evitare conflitti di utilizzazione.

Collider: Componente di Unity che serve a gestire le collisioni dell'oggetto con gli altri oggetti di gioco. Collider è la classe base da cui ereditano tutti i vari tipi di collider, ognuno dei quali ha una forma diversa.

Cross-Platform: Si riferisce ad un linguaggio di programmazione, ad un'applicazione software o ad un dispositivo hardware che funziona su più di un sistema o appunto, piattaforma.

G

GUI: E' un tipo di interfaccia utente che consente all'utente di interagire con il device controllando oggetti grafici convenzionali.

P

PSD: E' un formato di file nativo di Adobe Photoshop per il salvataggio di immagini con le differenti caratteristiche gestite dal programma.

R

Revenue: introiti che l'azienda riceve dalla sua normale attivita' di business

S

Software: E' un termine generico che definisce programmi e procedure utilizzati per far eseguire al computer un determinato compito.

T

Texture:

Immagine di qualsiasi tipo utilizzata per rivestire la superficie di un oggetto virtuale, tridimensionale o bidimensionale, con un apposito programma di grafica.

Bibliografia

- <http://www.odoo-italia.org/>
- <http://www.math.unipd.it/~tullio/IS-1/2014/>
- <http://docs.unity3d.com/Manual/index.html>
- <https://developer.vuforia.com/>
- <http://doc.photonengine.com/en/pun/current/getting-started/>