

POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria del Cinema e dei
Mezzi di Comunicazione



Tesi di Laurea Magistrale

**MagicTurinAR: sviluppo di un serious
game location-based, intergenerazionale e
collaborativo in realtà aumentata**

Relatore

Prof. Andrea BOTTINO

Candidati

Flavio DELPIANO

Niccolò VIANELLO

Anno Accademico 2021/2022

Abstract

Con il lancio di PokémonGo nel 2016, il primo videogame in AR diffuso su larga scala, la realtà aumentata è entrata nell'immaginario del grande pubblico come una tecnologia alla portata di tutti e implementabile anche con logiche di geolocalizzazione.

In parallelo si stanno valutando le possibilità che offrono i videogiochi in ambiti diversi da quelli legati al puro intrattenimento: negli ultimi anni infatti i Serious Games sono soggetto di numerosi studi per comprendere al meglio quali siano le loro reali potenzialità e se sia possibile sviluppare esperienze videoludiche che possano migliorare o facilitare le attività in diversi settori.

Nel corso degli anni però i videogames hanno mutato le loro logiche di gioco: generazioni differenti hanno sviluppato gusti molto diversi in ambito videoludico e questo ha generato un gap a livello intergenerazionale tra i giocatori più giovani e i più anziani.

Si delinea quindi la necessità di trovare un modo per poter colmare questa distanza, creando videogiochi che tengano conto delle differenze di età, ma che attraverso gameplay studiati ad hoc possano portare generazioni distanti anagraficamente, come nonni e nipoti, a interagire e collaborare in esperienze che possano essere stimolanti e coinvolgenti.

In questo contesto si inserisce lo scopo di questa tesi, ovvero creare un serious game intergenerazionale location based e collaborativo in realtà aumentata.

È stato sviluppato un applicativo studiato per migliorare a nuclei familiari l'esplorazione della Torino esoterica, favorendo inoltre la cooperazione tra i vari membri della famiglia. Progettando un Game Design intergenerazionale, che tenesse conto delle abilità psicofisiche di bambini, adulti e anziani è stata creata un'esperienza asimmetrica dove le mansioni di tre diversi ruoli, ognuno studiato per una specifica fascia anagrafica, favoriscono il divertimento e la godibilità del tour.

Durante l'esperienza i giocatori saranno catapultati in un realtà in cui Torino sta per essere sopraffatta da un male imminente. Orientandosi per le strade del centro, troveranno collezionabili e svolgeranno mansioni in realtà aumentata che permetteranno loro di apprendere nozioni sui luoghi visitati.

Per valutare l'efficacia dell'esperienza e comprenderne le eventuali problematiche, sono stati analizzati gruppi da tre persone, composti sia da giocatori anagraficamente adatti per il design di gioco del ruolo a loro assegnato, sia da giocatori di età casuale. Sono state in seguito valutate la giocabilità, l'usabilità e l'efficacia della natura serious del videogioco.

Indice

1	Introduzione	1
1.1	Storia della Realtà Aumentata	1
1.2	Serious Games: come i videogiochi sono stati implementati in ambiti non ludici	4
1.3	Intergenerazionalità: un problema moderno	5
1.4	Organizzazione dei capitoli	6
2	Stato dell'arte	7
2.1	Serious games	7
2.1.1	Serious Game per la riabilitazione e l'assistenza a persone affette da disabilità	8
2.1.2	Serious Game per l'istruzione	8
2.1.3	Serious Game in ambito medico	10
2.2	Location-based games	11
2.2.1	Pattern ricorrenti	12
2.2.2	Ambiti di applicazione dei LBMG	13
2.2.3	Casi studio	14
2.3	Videogiochi intergenerazionali	17
2.3.1	Game design intergenerazionale	18
2.3.2	Age invaders	20
2.4	Giochi collaborativi in AR	21
2.4.1	Holoroyale	22
2.4.2	MathBuilder	24
2.4.3	Strumento per l'ispezione di edifici prefabbricati	25
2.5	Valutazione dei tools esistenti	28
2.5.1	AR Foundation	28
2.5.2	AR + GPS Location	29
2.5.3	Vuforia	30
2.5.4	Wikitude SDK	32
2.5.5	Lightship ARDK	34
2.5.6	Unity MARS	35

2.5.7	Mapbox - Maps SDK	36
2.5.8	ArcGis	37
3	Tecnologie utilizzate	39
3.1	Unity3D	39
3.1.1	L'interfaccia	39
3.1.2	Gestione della fisica	41
3.1.3	Scripting	41
3.2	Lightship ARDK	42
3.3	Mapbox	45
3.3.1	Mapbox Studio	46
3.3.2	Maps SDK for Unity	46
3.4	Mirror	47
3.4.1	Server e Clients	47
3.4.2	Authority	47
3.4.3	Network Identity	48
3.4.4	Network Behaviour	48
3.4.5	Network Manager	49
3.4.6	NoIP	49
3.5	Parrel Sync	50
3.6	Firebase SDK: Authentication e Real-time database	51
4	Game Design Document	53
4.1	Sinossi	53
4.2	Target	54
4.3	Game outline	55
4.3.1	Collaborazione tra team	58
4.3.2	Suddivisione dei ruoli	58
4.3.3	Assegnazione dei ruoli	59
4.4	Core game context	60
4.4.1	Storyline	60
4.4.2	Setting	60
4.4.3	Personaggi	61
4.4.4	Obiettivi	62
4.4.5	Risorse e collezionabili	62
4.5	Core gameplay	63
4.5.1	Meccaniche principali	63
4.5.2	Inventario e collezionabili	64
4.5.3	Sfida finale	65
4.5.4	Sfide collaborative	65
4.6	Game loop	66

4.7	Gameplay puzzles	66
4.8	L'esperienza di gioco prevista	67
4.9	Le ragioni dell'engagement	67
4.10	Reference grafiche	68
4.10.1	Medieval Resurrection (2005)	68
4.10.2	Nightmare before Christmas	68
5	Sviluppo e design	71
5.1	Autenticazione	71
5.2	Connessione al server e gestione della lobby	73
5.2.1	Offline	73
5.2.2	Lobby	73
5.3	L'interfaccia principale	76
5.3.1	GUI e tutorial	77
5.3.2	Network Player	78
5.4	Geolocalizzazione	78
5.4.1	Setup	78
5.4.2	Location Based Game	79
5.4.3	Session Manager	80
5.5	Inizializzazione e looping dell'esperienza	81
5.5.1	Game Manager	81
5.5.2	Missions Manager	82
5.6	Realtà Aumentata	83
5.6.1	AR_LookAtItem	85
5.6.2	AR_EnemyFight	86
5.6.3	AR_DestroyGem	87
5.6.4	AR_WisemanMemory	88
5.6.5	AR_MazeGame	88
5.6.6	AR_Collaborative	90
5.6.7	Statistiche	92
6	Risultati e analisi	93
6.1	Formulazione dei questionari	93
6.1.1	Questionario Pre-Test	93
6.1.2	Questionario Post-Test	94
6.2	Test	95
6.3	Valutazione dei risultati	95
6.3.1	Questionario Pre-Test	95
6.3.2	Questionario Post-Test	100
7	Conclusione e sviluppi futuri	111

Capitolo 1

Introduzione

1.1 Storia della Realtà Aumentata

Nel 1901 appare per la prima volta in campo letterario qualcosa di molto simile all'AR nel libro *The Master Key* di Frank L. Baum, dove un ragazzo, grazie ad un paio di occhiali, riesce identificare le persone buone o cattive grazie a delle lettere che appaiono sulla fronte della gente: questi occhiali sembrano la descrizione dei Google Glasses o degli Hololens che si pensa che in un futuro prossimo saranno in grado di proiettare in realtà aumentata l'identità degli individui inquadrati intercettando i dati esistenti nel web [1].

L'AR è considerata come quella tecnologia che permette l'arricchimento della percezione di oggetti reali, tramite l'inserimento di oggetti virtuali. Questa tecnologia, nel corso del tempo, è diventata sempre più immersiva aumentando la veridicità delle interazioni tra componenti virtuali e reali oltre che popolare e a prezzo consumer, basti pensare all'enorme successo avuto dai filtri di Snapchat o di Instagram. La realtà aumentata però non è una tecnologia recente, poiché la sua invenzione risale al 1968, quando Ivan Sutherland creò un dispositivo con due tubi catodici che proiettavano immagini 3D sovrapposte al mondo reale e che pesava così tanto da dover essere sostenuto da un braccio meccanico attaccato al soffitto, e per questo soprannominato Spada di Damocle [2](fig. 1.1).

Però è solo più tardi che questo tipo di tecnologia acquista il nome con il quale oggi è identificata. Infatti nel 1990 alla Boeing, celebre industria aeronautica statunitense, Tom Caudell e David Mizell idearono il primo prototipo di applicazione AR per uso industriale che consisteva in un Head Mounted Display che avrebbe dovuto proiettare agli operai la disposizione corretta dei cavi all'interno delle strutture e dunque permettere all'azienda di risparmiare sui costosi e complessi diagrammi cartacei utilizzati.



Figura 1.1

Questo progetto non venne mai realizzato, ma la documentazione fu presentata durante l'edizione di quell'anno dell'Hawaii International Conference on System Science, gettando così le fondamenta per futuri lavori che avrebbero sfruttato l'idea innovativa [3]. Due anni dopo infatti Louis B. Rosenberg sviluppò Virtual Fixtures per la Air Force statunitense, il primo sistema AR immersivo funzionante che usava due bracci robotici controllati a distanza da un esoscheletro mosso dall'utente.



Figura 1.2

Nel 1993 venne sviluppata KARMA (Knowledge-base Augmented Reality For Maintenance Assistance) da Feiner, MacIntyre e Seligmann, un programma AR per la manutenzione di stampanti che si componeva di un HMD combinato con appositi marker piazzati sulla stampante, che permettevano la proiezione sulle lenti del dispositivo di un'immagine che indicasse all'utente cosa fare.

Ma la realtà aumentata non permeò soltanto negli ambienti industriali: venne infatti implementata anche nell’ambito artistico quando nel 1994 Julie Martin, un direttore di teatro australiano, creò il primo spettacolo in realtà aumentata *Dancing in Cyberspace*. In questa rappresentazione i ballerini e gli acrobati interagivano con oggetti virtuali a grandezza umana [4].



Figura 1.3

La realtà aumentata iniziava a prendere piede, ma ancora non era stata implementata a livello consumer, ovvero si trattava di una tecnologia ancora non fruibile a basso costo.

Nel 1999 Hirokazu Kato del Nara Institute Of Science And Technology, pubblicò la prima libreria per lo sviluppo AR open source ToolKit: essa permetteva di generare immagini in real time tramite l’utilizzo di una telecamera, di cui veniva tracciata la posizione e l’orientamento, e di specifici marker quadrati. Fu il primo software di realtà aumentata per mobile su Symbian, iOS e Android.

Il 1999 rappresenta dunque la data in cui iniziarono ad essere sviluppate molte delle SDK che tutt’oggi utilizziamo per produrre applicativi in AR su mobile a livello consumer come Wikitude, oppure i più recenti e conosciuti ARCore e ARKit. Negli ultimi anni la ricerca ha fatto sì che fossero integrati degli algoritmi di computer vision affinché non fossero necessari marker per le immagini in AR, arrivando a mettere a disposizione dispositivi e sdk che potessero identificare piani, figure geometriche e addirittura generare mesh grazie allo scan dell’ambiente per poi successivamente avere un’interazione tra oggetti virtuali e ambiente totalmente credibile. Nel 2016 uscì *PokemonGo*, prodotto da Niantic, che diventò rapidamente il primo videogame AR diffuso su larga scala.

Per quanto riguarda la parte hardware nel 2013 ci fu una rivoluzione introdotta dai Google Glass, un dispositivo HMD venduto inizialmente come prototipo ad un prezzo di 1500 dollari negli Stati Uniti. Come ToolKit per le SDK, Google Glass diede il via alla produzione di dispositivi HMD sempre più sofisticati, fino ad arrivare ai moderni Google Glass Enterprise Edition 2 e i Microsoft HoloLens 2.



Figura 1.4: PokémonGO



Figura 1.5: Google Glass vs Microsoft HoloLens 2

1.2 Serious Games: come i videogiochi sono stati implementati in ambiti non ludici

Con l'espressione Serious Game si intende un gioco di cui lo scopo principale non è il puro intrattenimento e, nel caso si tratti di videogames, si parla di software creati ad hoc che utilizzano meccaniche ed estetiche dei videogames di puro divertimento, ma che presentano specifici contenuti che posseggano autorevolezza scientifica per ingaggiare tipi particolari di utenze in attività educative e informative [5].

L'origine di tali giochi a scopo formativo risale addirittura al 799 d.C quando pare che Alcuino da York raccolse giochi logico matematici per la formazione di studenti della Scuola Palatina per agevolare l'apprendimento. Si registrano inoltre anche altri esempi di Serious Game come le simulazioni di guerra dell'esercito prussiano agli inizi del XVIII secolo o le simulazioni militari su computer condotte negli anni cinquanta alla Johns Hopkins University. Il termine Serious Game venne coniato

però soltanto nel 1971 quando Clark C. Abt, uno sviluppatore di giochi militari per computer, pubblicò un libro che si intitolava in questa maniera, definendo questi giochi educativi come giochi con uno scopo educativo esplicito e primario, non pensati primariamente per il divertimento senza però escluderlo. All'inizio del nuovo millennio si è registrato un'impennata nella produzione di questa tipologia di videogiochi che si evolsero fino a incorporare al proprio interno logiche di economie reali, come in *Second Life* nel 2010 (gli utenti possono creare attività reali che forniscono beni e servizi virtuali in cambio di denaro). Negli ultimi anni la ricerca si è concentrata molto su questa categoria di videogames per comprendere se essi possano rappresentare un'effettiva svolta in diversi ambiti, da quello educativo a quello della riabilitazione, e analizzando risultati empiricamente molto incoraggianti [6].

Ma quali sono gli ambiti in cui questi videogiochi sono stati sviluppati? Che migliori portano a questi? Gli ambiti di applicazione dei Serious Games oramai sono numerosissimi: l'ambito medico ad esempio ha riportato risultati molto interessanti nella riduzione del dolore tramite la distrazione del paziente con un'esperienza in VR (Virtual Reality). *Snow World*, un videogioco con visore che proietta il giocatore in un mondo virtuale innevato, ha permesso di ridurre il dolore nella fase del cambio garze di ustionati dal 30% al 50%. Ma gli ambiti di applicazione sono molto numerosi: da quello educativo a quello di formazione aziendale, dagli *Advergame* (giochi creati con fine di advertising) all'ambito della sensibilizzazione riguardo a tematiche sociali, si sono ottenuti risultati rilevanti che sono oggetto di una continua ricerca al fine di implementare sempre maggiormente questa tipologia di gioco [7].

Nel seguente capitolo verranno analizzati più casi studio e i risultati ottenuti per comprendere al meglio quanto questa soluzione sia efficace e attuabile per gli scopi di questa tesi.

1.3 Intergenerazionalità: un problema moderno

Le ultime generazioni sono nate e cresciute in un mondo dove la tecnologia ha permeato la società in maniera profonda: sono sempre maggiori il numero di dispositivi che vengono utilizzati nel quotidiano e l'età in cui ci avviciniamo a queste tecnologie diminuisce di anno in anno generando di fatto una maggiore familiarità con nuovi dispositivi e applicativi.

In questo contesto però alcune generazioni sono rimaste riluttanti nell'adozione nel quotidiano di tali tecnologie, principalmente quelle più anziane, e si è venuta a creare una forbice intergenerazionale dove i più piccoli sono sempre più interessati e immersi nello sviluppo tecnologico mentre i più anziani ne rimangono insofferenti

ed esclusi. Questa forbice si riflette necessariamente sugli interessi comuni, che sono sempre minori, e ciò provoca un vero e proprio isolamento generazionale per gli anziani andando anche a peggiorare la qualità del rapporto tra queste fasce di età.

Per sopperire a questo problema moderno la ricerca si è mossa per poter formulare design di videogames che potessero, tramite opportuni accorgimenti, generare coinvolgimento per tutte le diverse generazioni rafforzandone il legame e la socializzazione. Il processo del gioco aiuta, come visto precedentemente, il trasferimento della conoscenza e ciò permette alle fasce di età più avanzate di entrare in confidenza con l'utilizzo di tecnologie prima da loro inutilizzate. Un quarto degli over 50 afferma di amare il tempo trascorso a giocare ai videogiochi e molti di questi ammettono di utilizzare i videogiochi per connettersi maggiormente ai nipoti [8]. Ovviamente il design di tali esperienze videoludiche deve essere studiato e tarato sulle potenzialità psicofisiche del target di riferimento, ma l'intergenerational gaming sembra essere una delle possibilità migliori per chiudere il gap intergenerazionale. Nel prosieguo di questa tesi verranno analizzate e formulate delle valide linee guida per la formulazione di un design di gioco funzionale a questo proposito.

1.4 Organizzazione dei capitoli

La tesi è strutturata in sette capitoli. In questo primo capitolo sono stati illustrati i concetti fondanti del gameplay dell'esperienza e la storia dello sviluppo della tecnologia su cui si basa applicativo finale.

Nel secondo capitolo sono stati analizzati e confrontati diversi casi studio al fine di comprendere al meglio le migliori pratiche e estrapolare delle linee guida per l'ideazione del design di gioco dell'applicativo, oltre che analizzare i vantaggi e svantaggi di determinate librerie utili allo sviluppo dello stesso.

Nel terzo capitolo è stato affrontato un approfondimento delle tecnologie scelte per lo sviluppo dell'applicazione e delle principali features implementate nella stessa. Nel quarto capitolo viene invece descritto il processo creativo e le motivazioni fondanti che hanno contribuito all'ideazione del design di gioco.

Nel quinto capitolo viene descritto lo sviluppo tecnico dell'applicativo e delle scelte intraprese per creare l'infrastruttura di gioco. Nel sesto capitolo sono illustrate le metodologie di test dell'applicazione, i risultati ottenuti e le analisi relative a questi ultimi.

Infine, nel settimo e ultimo capitolo sono state formulate le conclusioni del lavoro svolto e formulate ipotetiche migliorie da apportare, oltre a eventuali futuri impieghi dell'applicativo.

Capitolo 2

Stato dell'arte

Con lo sviluppo dei dispositivi mobili, la realtà aumentata è diventata di anno in anno una soluzione sempre più convincente e alla portata di tutti. La sua semplicità di utilizzo, unita alla portabilità dei dispositivi mobili (smartphones, tablet...) ne ha garantito un'enorme diffusione. Gli ambiti di applicazione sono infatti molteplici: dal gaming ad applicazioni educative, fino al training aziendale.

Vista l'enorme quantità di casi studio e articoli di potenziale interesse, è stato necessario restringere la ricerca, imponendo alcuni vincoli. La ricerca si è concentrata su quattro macro categorie:

- serious games,
- location-based games,
- giochi intergenerazionali,
- giochi collaborativi,
- giochi ed esperienze AR

È stato quindi necessario analizzare, parallelamente, anche tools e SDK presenti sul mercato, al fine di valutare le opzioni migliori per gli scopi di questo studio.

2.1 Serious games

L'espressione "Serious Games" indica generalmente i giochi con uno scopo che va oltre il puro intrattenimento, per offrire esperienze di apprendimento coinvolgenti in diverse gamme di settori [9]. Essendo questa una caratteristica importante per lo sviluppo del prodotto finale per questa tesi, sono stati analizzati diversi ambiti che sfruttano una dimensione di gioco per ottenere risultati significativi tramite la

realtà aumentata.

2.1.1 Serious Game per la riabilitazione e l'assistenza a persone affette da disabilità

L'utilizzo di Serious Games (SG) è sempre più considerata nell'assistenza a persone con disabilità: sono presenti sempre più studi e realtà che analizzano i risultati di terapie che utilizzano la dimensione del gioco confrontando i risultati con quelli ottenuti da procedure standard.

Le ricerche condotte sui SG hanno messo in evidenza vantaggi per quanto riguarda il coinvolgimento dei giocatori e l'efficienza dell'approccio, ma una costante sfida è rappresentata dall'implementazione congiunta di queste terapie con la tecnologia: essa risulta utile in queste dinamiche poiché permette, tramite movimenti nel mondo reale, di lavorare con oggetti virtuali catturando l'attenzione dell'assistito e migliorando il processo comunicativo [10].

Un altro elemento di motivazione è la ludicizzazione: un gioco infatti genera nelle persone un senso di realizzazione poiché sono presenti indicatori di progresso che sono importanti a mantenere alto l'interesse nell'attività, altrimenti tendono ad essere demotivate [11].

In questo contesto si inserisce un progetto come ATHYNOS [10], sviluppato su Unity 3D (codice in C) in combinazione con la libreria Vuforia e sviluppato per Kinect, che è stato al centro di uno studio sui risultati di tali terapie in realtà aumentata su bambini affetti da disprassia motoria o dalla sindrome da deficit dell'attenzione e iperattività. I bambini, divisi in due campioni basati sul disturbo di cui erano affetti, hanno testato diversi mini-giochi sviluppati grazie alla collaborazione di psicologi specialisti e seguendo logiche di ludicizzazione come ricompense, elementi di interazione piacevoli ai soggetti testati e obiettivi. Entrambi i due mini-giochi, che richiedevano una performance di difficoltà pari ad esercizi canonici, sono stati conclusi con risultati migliori sia in termini di tempo che in termini di correttezza, a dimostrazione di come la realtà aumentata, congiunta ad un processo di gamification, possa essere impiegata per l'assistenza e l'apprendimento di bambini affetti da determinati disturbi motori e cognitivi.

2.1.2 Serious Game per l'istruzione

Un interesse sempre maggiore della ricerca nell'ambito dei Serious Games viene riservato allo sviluppo di SG efficaci per la divulgazione e per l'apprendimento. Tramite l'utilizzo di nuove tecnologie, come la realtà aumentata, è possibile creare un nuovo metodo di insegnamento che possa essere più coinvolgente e interessante



Figura 2.1: Minigioco usato per i bambini affetti da disprassia motoria, in cui il soggetto deve associare l'immagine al contorno

sia per insegnanti, che per studenti [12]. Inoltre, alcuni studi hanno attirato l'attenzione sull'impatto sociale che la realtà aumentata può avere sugli studenti. L'uso di questa tecnologia, infatti, non solo è molto affascinante, ma può coinvolgere gli alunni, i loro genitori e gli insegnanti in interazioni sociali fornendo maggiori opportunità di comunicazione e collaborazione nel mondo reale [13].

In questo contesto si inseriscono due progetti molto interessanti: RoboTIC e pARabola. Entrambi gli applicativi sono stati sviluppati in Unity 3D, ma il primo è sviluppato per Microsoft Hololens, mentre il secondo per Android.

RoboTIC consiste nell'esplicazione di concetti chiave della programmazione con metafore di gioco accattivanti per i più giovani, con lo scopo di insegnare e interessare i ragazzi che, o non sono attratti, o non sono a conoscenza dell'ambito dello scripting.

Il gioco consiste nel far avanzare un robot utilizzando delle istruzioni che rappresentano concetti chiave della programmazione (cicli, condizioni, funzioni, ecc...) per farlo arrivare ad un punto di arrivo che rappresenta il completamento di un livello. Ovviamente i livelli sono molteplici e introducono, in ordine di difficoltà i concetti che si vogliono far apprendere.

Uno studio è stato condotto su un campione di ragazzi tra gli otto e i quindici anni, la cui maggior parte non era a conoscenza di cosa fosse la programmazione e nessuno aveva intenzione, in un futuro, che essa diventasse il fulcro del loro lavoro. A fine delle sessioni di gioco è risultato che più della metà dei ragazzi si fosse divertita durante l'esperienza e che il 62,5% dei partecipanti allo studio fosse ora interessata ad imparare a programmare nel prossimo futuro, a suggerire un interessante possibilità educativa [14].

Molto interessante è anche lo sviluppo di pARabola e il relativo studio condotto sulla sua efficacia. Tramite questa applicazione sviluppata utilizzando VuForia, alcuni studenti sono stati in grado di comprendere visivamente il concetto di

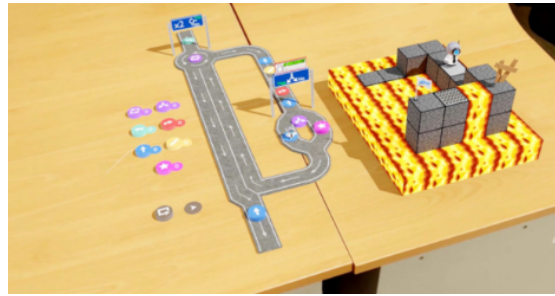


Figura 2.2: RoboTIC: sulla sinistra vengono selezionate le istruzioni e l'ordine con cui il robot le dovrà eseguire, sulla destra c'è il robot che mostrerà l'efficacia e la correttezza delle azioni selezionate e ordinate dall'utente

equazione quadratica, utilizzando un semplice dispositivo android ed un marker, e di comprendere anche il valore dei diversi parametri che la definiscono. Attraverso una GUI interagibile grazie al touchscreen del dispositivo, gli alunni che hanno testato pARabola, sono stati in grado di visualizzare visivamente i cambiamenti grafici che subisce una parabola al variare dei coefficienti dei termini di vario grado, segnalando nel questionario a cui sono stati sottoposti, che l'app li ha aiutati a comprendere meglio e più facilmente le diverse parti della funzione parabolica e il suo grafico [12].



Figura 2.3: A sinistra pARabola nella playmode di Unity, sulla destra l'effettivo funzionamento sul device

2.1.3 Serious Game in ambito medico

Nel corso della ricerca degli ambiti di applicazione dei Serious Game è risultato molto interessante un articolo di Boletsis e McCallum (2017) "The Smartkuber Case Study [15]: Lessons Learned from the Development of an Augmented Reality Serious Game for Cognitive Screening" che confronta un approccio di screening della demenza senile non canonico in realtà aumentata, come quello che viene descritto all'interno dell'articolo, con quello usuale (in carta e penna o computerizzato), cercando di comprendere che conseguenze possa avere "l'effetto wow" e l'esperienza

ludica sul coinvolgimento del paziente: si teme infatti, che le consuete modalità di screening non siano del tutto affidabili poichè il paziente potrebbe avere dei cali di concentrazione nel rispondere ai quesiti, dovuti alla noia e alla poca motivazione generata dal test cartaceo. L'esperienza è molto semplice, e consiste in cinque minigiochi dove il paziente deve:

- Memorizzare e ricostruire una bandiera con le facce colorate dei cubi
- Memorizzare le facce di due amici e usare i cubi per formare la corretta coppia di amici
- Memorizzare una sequenza di forme e ricomporla con i cubi
- Calcolare un'operazione matematica e rispondere componendo il risultato con i cubi
- Rispondere a una domanda formando una parola di risposta con i cubi

I risultati sono poi comprensibili grazie ai feedback in realtà aumentata. Attraverso diversi test e versioni di Smartkuber (sono state prodotte sia un'alfa sia una beta) si sono arrivati a correggere gli elementi di disagio che questo Serious Game provocava nei pazienti (ad esempio inizialmente le facce dei dadi erano disegnate in realtà aumentata, successivamente, a causa di difficoltà insorte nel tracking, si è preferito utilizzare dadi già disegnati e utilizzare solamente un'interfaccia e i feedback in realtà aumentata) e infine si è testato il prodotto finale su 13 persone anziane.

I risultati ottenuti sono stati confrontati con quelli registrati nel test di screening del MoCA [16] ed è stata notata una correlazione significativa tra le due tipologie di screening, evidenziando risultati molto simili, questo a conferma che elementi di ludicizzazione e il coinvolgimento di nuove tecnologie possono essere combinati anche in questo ambito.

2.2 Location-based games

Un servizio location based (LBS) è un tipo di servizio in grado di sfruttare i dati relativi alla localizzazione del dispositivo mobile sul quale è ospitato [17]. Sebbene alcuni LBS siano di uso comune dalla fine del secolo scorso (come il GPS), è solo con la pubblicazione di Ingress e PokémonGo, da parte di Niantic, che i sistemi location based iniziano a diventare mainstream anche nel mercato videoludico. Emerge quindi l'espressione Location Based Mobile Game (LBMG) a indicare proprio questo tipo di prodotti, che Lehman definisce come giochi pensati per essere utilizzati su dispositivi mobili, che adattano l'esperienza in base alla localizzazione del giocatore. I LBMGs sono assolutamente in grado di costruire un'intera esperienza di gioco,

rendendo il mondo reale parte della struttura narrativa e del gameplay stesso, sfumando moltissimo il confine tra gioco e realtà. Ciò porta inevitabilmente ad un cambiamento su più livelli del processo di game design (meccaniche, interazioni, flow) e di conseguenza del modo di giocare. È evidente che sia necessario tenere in considerazione anche i limiti di questa tecnologia: tra tutti, i principali riguardano la precisione limitata e variabile (da qualche metro fino anche ad alcune decine di metri) del GPS e i problemi derivanti dalla capacità di calcolo comunque contenuta dei dispositivi mobili contemporanei, cosa che rende assolutamente necessaria una particolare attenzione ai meccanismi di allocazione delle risorse.

2.2.1 Pattern ricorrenti

Nell'articolo dal titolo "Location-based Mobile Games", Lorenz Lehmann [17] evidenzia alcuni pattern strutturali ricorrenti all'interno dei LBMGs:

- Search-and-find: il giocatore ha il compito di raggiungere un luogo specifico, assegnato in qualche modo dal sistema. Le modalità di raggiungimento possono variare e utilizzare, ad esempio, un sistema di navigazione integrato.
- Follow-the-path: anche in questo caso il giocatore deve raggiungere una destinazione specifica ma, a differenza del pattern search-and-find, è importante che lo faccia seguendo una strada definita dal sistema. Per esempio, in Tourality il giocatore deve compiere il percorso dato entro un limite di tempo.
- Chase-and-catch: questo pattern prevede la presenza di un oggetto in movimento che il giocatore deve raggiungere e con lo scopo di interagire con esso (ottenere informazioni, collezionarlo, catturarlo, ...). Questo oggetto può essere sia appartenente al mondo reale, ad esempio un altro giocatore, sia parte del mondo virtuale. La core feature di questo sistema risiede nel cambiamento di posizione dell'oggetto da raggiungere, la cui frequenza determina il livello di difficoltà dell'esperienza. Nel gioco Shadow cities, ad esempio, gruppi di giocatori devono "cacciarsi" a vicenda: in questa dinamica, tutti sono allo stesso tempo cacciatori e prede.
- Change-of-distance: pattern certamente meno utilizzato rispetto agli altri, si concentra sulla distanza percorsa. Non contano infatti la direzione né il luogo raggiunto. Un esempio è costituito da The Journey, in cui le varie parti della storia si attivano in successione solo dopo aver percorso una determinata distanza dal momento in cui il pezzo precedente è stato sbloccato (Mopius, 2011).

2.2.2 Ambiti di applicazione dei LBMG

Tradizionalmente, lo scopo di un videogioco è sempre stato confinato all'intrattenimento puro. Con il tempo, sono nati alcuni sistemi che hanno consentito di integrare aspetti diversi all'interno dell'esperienza di gioco: la Nintendo, fra gli altri, nel 2006 ha lanciato Wii, una console che aveva l'obiettivo di unire il divertimento all'attività motoria.

Nel caso dei LBMGs, si è rivelato possibile integrare all'interno dell'esperienza funzionalità alternative in modi decisamente innovativi.

Attualmente, sempre secondo Lehmann [17], si possono individuare 5 principali campi di applicazione:

- **Intrattenimento:** Ogni gioco ha il compito primario e intrinseco di intrattenere. I titoli più commerciali presenti sul mercato si concentrano sulla massimizzazione del livello di intrattenimento, al fine di generare il profitto più alto possibile. Nell'ambito dei LBMG ce ne sono ancora pochi: degni di nota sono sicuramente Ingress, PokémonGo! e il meno conosciuto Shadow Cities.
- **Educazione:** Sempre secondo Lehmann [17], i LBMGs hanno una possibilità enorme in ambito educativo: connettere luoghi e storie, sfruttando ad esempio il pattern Search and find, particolarmente adatto nel caso in cui si vogliano integrare luoghi o elementi storici nell'environment di un gioco. La differenza sostanziale tra un videogioco tradizionale e un LBMG sta nella maggiore connessione tra realtà e finzione in quest'ultimo approccio: ciò garantisce un livello più alto di autenticità percepita e quindi un potere educativo superiore come diretta conseguenza.
- **Attività fisica:** Come accennato in precedenza sono state sviluppate piattaforme con lo scopo di incentivare l'attività fisica: Wii, PS Move e Kinect sono certamente le più note. Evidentemente, però, la ristretta gamma di movimenti/azioni possibili e lo spazio ridotto hanno sempre costituito un forte limite. Ecco quindi un ulteriore spazio in cui riescono a inserirsi i giochi location-based. Tourality, già citato, costituisce un perfetto esempio di location-based mobile game orientato all'attività fisica: il giocatore vince se corre più veloce degli altri.
- **Advertising:** La localizzazione è già ampiamente sfruttata per personalizzare le pubblicità in base alla localizzazione dell'utente target. Questo rende i LBMGs assolutamente appetibili per un imprenditore che vuole aumentare i propri guadagni. Nel suo articolo, Lehman [17] fa l'esempio di un supermercato in cui vengono fatti spawnare dei tokens in corrispondenza di alcuni prodotti (magari quelli in offerta, o quelli che vengono venduti con più fatica, per esempio). Se il cliente li raccoglie tutti ottiene un buono sconto da utilizzare nel supermercato

stesso. Un'app gratuita, per questi scopi, sarebbe sicuramente utilizzata dai clienti, pur di ottenere un prezzo più vantaggioso in fase di acquisto.

- **Acquisizione dati:** Un'ultima grande possibilità offerta dai LBMGs è relativa all'acquisizione di dati dal mondo reale, sfruttando il lavoro degli utenti. Attraverso la camera, per esempio, è possibile ottenere informazioni sull'ambiente, utilizzabili per rendere più dettagliate le mappe. L'approccio ludico chiaramente induce maggiormente l'utente a utilizzare l'applicativo, cosa che ha come effetto una maggiore, e quindi più utile, raccolta di dati. Un esempio di LBMG per la raccolta dati è rappresentato da CityExplorer [18]. Questa applicazione ha lo scopo di incrociare le informazioni provenienti dalla rete di comunicazione cellulare (celle GSM ma anche dati provenienti da access point WiFi) con quelle ricavate dal GPS. CityExplorer implementa anche una funzionalità per ricavare informazioni non-geografiche, come i POI. A questo scopo, l'approccio adottato è di tipo esplorativo: inizialmente la mappa è completamente vuota e i giocatori hanno il compito di ricercare, all'interno di un'area ristretta (denominata tile) un oggetto assegnato. Una volta raggiunto il luogo, corrispondente ad esempio ad un museo, un'opera architettonica, un parco, questo viene rivelato sulla mappa. Un secondo esempio è FIASCO [19], in cui i giocatori devono interpretare i luoghi realizzando delle scene create a partire da immagini raffiguranti i luoghi stessi. Questo favorisce una navigazione diversa sulla mappa, in cui emergono i diversi punti di vista attraverso i quali le persone vedono ciascun luogo.

2.2.3 Casi studio

Wikitude Drive

Sviluppata da Wikitude, Wikitude Drive è un'applicazione per smartphone dedicata alla navigazione, che implementa un'interfaccia in realtà aumentata: la fotocamera inquadra la strada, su cui compare una linea guida in AR che indica la direzione da seguire. L'app offre comunque la possibilità di passare ad una modalità di navigazione tradizionale, senza AR, toccando lo schermo.

Wikitude Drive implementa anche alcune features di tipo informativo: l'utente, muovendosi nello spazio, ha la possibilità di rivelare informazioni relative a ciò che ha intorno.

Purtroppo, l'applicazione non è più disponibile, ma a nostro parere costituisce un valido spunto per il prosieguo del lavoro su cui questa tesi si concentra.



Figura 2.4: Wikitude Drive

PokémonGo

PokémonGo rappresenta in assoluto il titolo che ha reso mainstream la realtà aumentata unita alle tecnologie di location detection nel mondo dei videogiochi. Sviluppato da Niantic, e pubblicato nel 2016, ha avuto uno smisurato successo in tempi brevissimi: si tratta infatti del primo gioco in AR per mobile in grado di raggiungere la vetta della classifica dei download. Inizialmente, le funzionalità legate alla realtà aumentata erano molto basilari e per questo venivano spesso disattivate dai giocatori. Nel 2018 Niantic rilascia un aggiornamento rinnovando queste features, proponendo AR+. Questa modalità di realtà aumentata è molto più immersiva grazie ad alcuni fattori: gli oggetti in realtà aumentata sono ancorati al suolo in maniera molto più precisa, consentendo al giocatore di muoversi intorno ad essi come se fossero realmente esistenti. Inoltre, è rilevato il movimento del giocatore, che, se è troppo vicino e si muove bruscamente, provoca la fuga del pokémon.

Nel gameplay di Pokémon Go! è presente una dinamica collaborativa sincrona, rappresentata dai cosiddetti raid: si tratta di un evento in cui più Allenatori devono unire le forze per sconfiggere un Boss Pokémon che ha conquistato una palestra. Più è alto il livello di difficoltà del raid, maggiore è il numero di Allenatori necessario per sconfiggere il Pokémon. Nel caso dei raid classici, la collaborazione è co-localizzata, ma esistono anche dei raid nei quali è possibile partecipare da remoto.

Per quanto riguarda la mappa, inizialmente il gioco sfruttava l'API di Google Maps, ma attualmente utilizza OpenStreetMap, un sistema totalmente open source.



Figura 2.5: PokémonGo

Viking Gost Hunt

Viking Gost Hunt è un gioco location based di avventura, basato su una ghost story gotica, ambientata durante l'occupazione vichinga di Dublino (800-1169 d. C.). Il giocatore veste i panni di un investigatore e ha il compito di esplorare la città con l'obiettivo di individuare tracce di attività paranormale, risolvendo enigmi e raccogliendo prove. Grazie al GPS, il sistema è in grado di collocare le attività paranormali in specifici punti e di confrontarne le coordinate con quelle del giocatore in movimento. Il giocatore utilizza dapprima il radar, per rilevare attività nelle vicinanze, e, non appena questo percepisce qualcosa, apre il Ghost Viewer per individuare il fantasma che ha originato l'attività paranormale. Il giocatore quindi deve scattare una foto per raccogliere la prova.

Particolarmente interessante è la struttura narrativa adottata all'interno del gioco: come accade spesso nei giochi di avventura, e in particolar modo in quelli a scopo investigativo, le azioni del giocatore sono cruciali e determinano l'andamento della storia. La narrazione, quindi, non è lineare e può svilupparsi in vari modi a seconda di come si muove il giocatore (sia nello spazio che dal punto di vista delle decisioni che sceglie di prendere).

Ingress

Sviluppato da Niantic, Ingress è un AR Location-based Mobile Game (LBMG), lanciato prima per Android, il 30 ottobre 2013, e in seguito per iOS, il 14 luglio 2014. I giocatori possono fare parte di una delle due fazioni in guerra tra loro: gli Illuminati e la Resistenza. Scopo del gioco è porre sotto il controllo della propria fazione le cosiddette Mind Units, corrispondenti ad aree del mondo reale. Ogni Mind Unit è rappresentata da un Control Field, ovvero l'insieme di tre portali collegati a triangolo. I portali sono collocati in luoghi reali di particolare interesse storico, artistico o culturale. Per acquisire il controllo dei portali, i giocatori



Figura 2.6: Viking Gost Hunt

devono recarsi fisicamente sul posto e interagire con essi: se un portale è neutro, ovvero non appartenente a nessuna delle due fazioni, il giocatore può conquistarlo posizionando i propri risonatori negli appositi slot; se invece il portale è sotto il controllo della fazione avversaria, questo può essere conquistato a seguito della distruzione dei dispositivi di controllo piazzati dagli avversari. Dal punto di vista del gameplay, è di fondamentale importanza la collaborazione tra agenti della stessa fazione: per costruire portali di livello n , ad esempio, è necessario che n giocatori posizionino un proprio risonatore sul portale; oppure, per realizzare un control field che copra un'area geografica molto vasta, servono tre giocatori in luoghi distanti che conquistino e allineino tre portali.

Ciò che emerge dall'analisi dei casi studio nei precedenti paragrafi è che spessissimo i sistemi che implementano tecnologie di tipo location based ricorrono all'utilizzo della realtà aumentata come soluzione in grado di arricchire la user experience. Per quanto riguarda le applicazioni videoludiche (Ingress, PokémonGo!) la realtà aumentata costituisce un elemento che incrementa l'immersività dell'esperienza ma che purtroppo, quando è possibile farlo, viene disattivata dall'utente. Nel caso di Wikitude Drive, invece, le funzionalità di realtà aumentata sono persino in grado di garantire un livello di sicurezza aggiuntivo rispetto all'interfaccia tradizionale. L'aspetto collaborativo esiste, ma non è quasi mai preponderante né per quanto concerne le app di utility né quelle che rientrano nell'ambito ludico.

2.3 Videogiochi intergenerazionali

Le attività intergenerazionali sono attività che puntano a riunire persone appartenenti a generazioni diverse per un obiettivo comune, basandosi sui diversi punti di forza che la diversa età offre, cercando di coltivare il rispetto reciproco e sfidare

la vecchiaia [20] , ma il contatto intergenerazionale sta diventando sempre più limitato a causa di una “segregazione per età” dovuta alle istituzioni, alla cultura, al cambiamento delle situazioni abitative e ad altri fattori [21].

Con il miglioramento tecnologico sta aumentando il gap generazionale e le giovani generazioni sono sempre meno attratte dalle attività ludiche di quelle più anziane (giochi da tavolo come gli scacchi), bensì sono attratte da prodotti videoludici [22] che raramente sono stati ideati e sviluppati tenendo conto delle fasce di età più avanzate e dei loro gusti.

Questo vuol dire però che c'è indifferenza verso un determinato target nel mercato composto, secondo il Dipartimento degli affari economici e sociali delle Nazioni Unite [23], dal 9.3% della popolazione mondiale di persone sopra i sessantacinque anni nel 2020, e che si stima possa crescere fino al 16% raggiungendo 1.5 miliardi di persone.

In questo capitolo sono stati analizzati diversi videogiochi intergenerazionali in realtà aumentata, che sfruttano le potenzialità di questa tecnologia, e le modalità per creare un design e un bilanciamento del videogioco utile a rafforzare la collaborazione e la comunicazione tra le fasce di età più distanti.

2.3.1 Game design intergenerazionale

La creazione di un design vincente per un gioco intergenerazionale non è scontata e ha bisogno di tenere in considerazione preferenze, abitudini, bisogni e limitazioni del target al quale si riferisce [24], dunque è necessario non solo indagare sulle fasce di età avanzate, ma anche su quelle più giovani, e trovare il punto di incontro che possa generare un'esperienza godibile per entrambe.

È importante considerare i cambiamenti dovuti all'invecchiamento, non solo dal punto di vista del degrado fisico e della coordinazione, ma anche dal punto di vista di una vasta e diffusa esperienza del mondo, una maggiore saggezza e una più grande sapienza [25], cercando di creare un'esperienza asimmetrica per i giocatori di diverse età, che possa sfruttare le loro diverse caratteristiche e bilanciarle in modo da non creare tensioni o situazioni di eccessiva difficoltà durante il gameplay. De Schutter e Abeele (2015) nel “Gerontoludic Manifesto” [26] suggeriscono che nell'ideazione di un gioco intergenerazionale ci sia necessità in primis di pensare alla giocabilità più che all'utilità, poiché l'attività ludica in sé ha un suo scopo e un'utilità, e in secundis di considerare la vecchiaia sia come un processo di declino che di crescita, andando a sottolineare quanto sia importante non ignorare le ridotte capacità fisiche, ma neanche sottovalutare capacità derivate dalla loro maggiore esperienza di vita e maggiore conoscenza.

Kian Razavi Satvati e Mojtaba Vahidi-Asl nel loro studio “Looking at Intergenerational Game Design through the Lens of Game Genres” [24] hanno stilato una

lista, basandosi sulle preferenze e le limitazioni di bambini e anziani, di 9 regole per creare un gioco intergenerazionale:

- Deve essere facile parteciparvi
- Avere obiettivi ben chiari
- Richiedere sessioni di gioco brevi e veloci
- Permettere diverse tipologie di sfida
- Utilizzare sia forme di interazioni competitiva (più affine ai bambini), sia cooperative (più affini agli anziani)
- Permettere interazioni sociali dirette tra i giocatori
- Avere un'interfaccia di gioco semplice e familiare
- Non essere troppo dipendente dalle capacità fisiche dei giocatori (gli anziani devono poter essere utili e competitivi in ogni tipologia di sfida)
- Applicare una strategia del tipo “Prima pensa e poi prendi decisioni”.

La ricerca dei sopracitati autori suggerisce che la dimensione che possa accogliere al meglio un'esperienza virtuale intergenerazionale sia un'esperienza in realtà mista, siccome le fasce di età più giovane preferiscono interagire con strumenti tangibili, mentre le fasce di età maggiore hanno difficoltà ad utilizzare diversi e nuovi dispositivi di input. Infine è stata stilata una classifica, in base alle caratteristiche dei generi videoludici, della compatibilità tra genere e possibilità di gioco intergenerazionale:

- Gioco d'azione: Interazioni dirette con i compagni garantite da chat vocali, la molteplicità di sfide, interfacce grafiche semplici con poche informazioni fondamentali e benefici derivanti dalle strategie adottate, rendono questa tipologia di videogame la più appetibile per il design di un gioco intergenerazionale sebbene richiedano un ingente uso dei riflessi e delle capacità fisiche. Questo ostacolo potrebbe essere aggirato con un gameplay asimmetrico che richieda mansioni più fisiche ad un player giovane, e più tattiche ad uno più anziano.
- Gioco di strategia: Ci sono un buon numero di interazioni con gli altri giocatori, sia competitive che collaborative, e per ottenere la vittoria può essere richiesto un diverso numero di sfide (vedi *risiko*) e tattiche complesse. La complessità del sistema di interfaccia, spesso molto articolato a causa dei molti fattori da tenere sotto controllo, e le lunghe sessioni di gioco (sia bambini che anziani tendono a preferire sessioni più brevi) potrebbero essere elementi di disagio a cui si dovrebbe rimediare.

- Giochi Puzzle: Meccaniche chiare, interfacce semplici e il non utilizzo di elevate capacità fisiche sono caratteristiche favorevoli alla creazione di un gioco intergenerazionale, ma la varietà dei task richiesti è minima e le sessioni possono essere anche molto lunghe al variare della complessità del puzzle.
- Giochi d'avventura/CMSs (Construction and Management Simulations)/RPGs/Simulazioni Sportive: Questi generi vengono classificati tutti allo stesso livello. Mentre i giochi d'avventura, i RPGs e i CMSs sono accomunati da sessioni troppo lunghe, le simulazioni sportive mostrano carenze dal punto di vista della varietà di obiettivi. Queste tipologie di gioco però presentano tutte comunque degli aspetti che potrebbero essere sfruttati nell'ambito della creazione di un gioco intergenerazionale: basti pensare che i giochi di avventura, come i RPG, possono presentare al loro interno una grande varietà di obiettivi e di generi di gioco (vedi risoluzione di enigmi tipica dei puzzle game), mentre i CMSs richiedono una pianificazione strategica e le simulazioni sportive propongono il miglior tipo di interazione sia collaborativa che competitiva. Questi generi dunque presentano delle potenzialità nell'ambito dei giochi intergenerazionali, ma sono rischiosi per alcune delle loro caratteristiche fondanti che vanno gestite al meglio per non portare a tensioni durante l'esperienza ludica.
- Simulazioni di guida: Non essendoci nè varietà di obiettivi o interfacce di gioco semplici, ed essendo che le uniche sfide proposte da questa tipologia di gioco sono fisiche, questo genere è quello che meno si adatta alle caratteristiche sopracitate per la creazione di un design vincente per un gioco intergenerazionale.

Dopo questa breve digressione sul design di gioco si procede ora all'analisi di un caso studio che è stato analizzato che segue i canoni sopra citati.

2.3.2 Age invaders

Il caso studio che verrà trattato è un gioco intergenerazionale che rispetta le regole per un game design funzionale per un gioco inter-generazionale citate nel capitolo precedente, e il suo nome è Age Invaders. Questo videogioco, di cui parlano Khoo e Cheok nel loro articolo “Age Invaders : Inter-generational Mixed Reality Family Game” nel 2006 [22], si pone come obiettivo non solo quello di rafforzare il legame e le interazioni tra le diverse generazioni che compongono una famiglia, ma anche quello di incoraggiare l'attività fisica, poiché richiede macromovimenti, e cognitiva, siccome comprende al suo interno dei concetti appartenenti ai puzzle-games.

Age Invaders si ispira al celebre videogioco Space Invaders di Toshihiro Nishikado del 1978, non soltanto riprendendo il gameplay e il titolo, ma adottando uno stile grafico simile a quello dei primi videogiochi degli anni '70 e '80 per generare un

sentimento di familiarità con le generazioni più anziane, mentre l'infrastruttura più moderna è stata pensata per catturare l'attenzione dei più giovani. Infrastruttura che si compone di una tavola composta da blocchi LED ad alta risoluzione, sui quali i giocatori si muovono; il movimento di questi ultimi è tracciato grazie a sensori RFID reader, presenti nelle scarpe in dotazione, mentre loro sono equipaggiati di una pistola bluetooth e di un display led su un'imbracatura.

Il gameplay consiste nel seguire un percorso tracciato sui led, evitando bombe e razzi scagliati dagli altri giocatori tramite le pistole bluetooth, cercando di ottenere un punteggio alto e cercando di non esaurire la vita. Il gioco inoltre è arricchito non solo tramite la collaborazione diretta tra i giocatori sulla pedana, ma anche grazie alla collaborazione da remoto, pensata per i genitori dei bambini, che consiste nella visualizzazione della pedana con i diversi giocatori tramite una rappresentazione 3D e permette di istanziare dei bonus, utili sia per il punteggio che per la vita, e di bilanciare alcune caratteristiche come la velocità dei razzi o la velocità con cui i percorsi si aggiornano. E' molto interessante come è stato affrontato il tema del bilanciamento delle risorse, poiché tiene in considerazione le caratteristiche fisiche dei giocatori non andando a creare situazioni di eccessivo stress: i razzi degli anziani sono più veloci, e i loro percorsi più semplici, tuttavia è possibile modificare queste impostazioni iniziali e adattarle per rendere il gioco sempre coinvolgente.

Analizzando Age Invaders dal punto del game design si può notare che le 9 regole prima descritte da Satvati e Vahidi-Asl [24] vengono rispettate, e anche i casi più complessi sono gestiti in maniera esemplare, come nel caso dell'ottava regola ("Non essere troppo dipendente dalle capacità fisiche dei giocatori (gli anziani devono poter essere utili e competitivi in ogni tipologia di sfida)") in cui gli ideatori hanno inserito un bilanciamento che tenesse conto delle varie capacità dei giocatori, inserendo comunque la possibilità di modificarlo in corso d'opera. Inoltre Age Invaders rientra nel genere dei giochi d'azione, genere consigliato dai sopracitati ricercatori come il migliore per i videogiochi inter-generazionali, adottando un sistema di realtà mista che viene da Stavati et al. (2021)[24] considerato come miglior soluzione per questo genere.

2.4 Giochi collaborativi in AR

Una caratteristica che vuole essere implementata nel progetto di questa tesi è la dimensione sociale dell'esperienza videoludica, ovvero sarà necessario che gli utenti possano collaborare nello svolgimento di mansioni con lo scopo di raggiungere obiettivi comuni. Affinché questo sia possibile, può essere necessario che i dati siano distribuiti, memorizzati e reperibili da tutte le persone coinvolte nell'esperienza e ciò comporta alcune considerazioni sugli strumenti da utilizzare.

In questo paragrafo analizzeremo tre casi studio, un videogioco, un serious game per



Figura 2.7: Age invaders

l'apprendimento e un'applicazione per l'ispezione di costruzioni prefabbricate, che implementano la collaborazione tra diversi utenti in maniera sincrona e asincrona.

2.4.1 Holoroyale

Questo videogioco è illustrato nell'articolo "Towards Large Scale High Fidelity Collaborative Augmented Reality" [27], che si pone come obiettivo quello di presentare il primo LSHF CAR (Large Scale High Fidelity Collaborative Augmented Reality) riuscendo ad affrontare tutte le sfide di design e tecnologiche che esso comporta: questo vuol dire che gli utenti devono poter collaborare in un'esperienza in realtà aumentata su larga scala (Holoroyale è pensato per un'area di 10, 000 metri quadrati) e che possa godere di interazioni verosimili e di consistenza spazio-temporale e visiva tramite l'uso di un dispositivo Hoololens. Dal punto di vista tecnologico questa è stata una vera e propria sfida considerando i limiti tecnici dei dispositivi, e anche usando algoritmi SLAM (Simoultaneous Localization And Mapping) ci sono vere e proprie limitazioni dovute alla sincronizzazione delle pose tra utenti e latenza del network.

Infrastruttura Software e Hardware

Per far fronte a queste problematiche il team di sviluppo ha pensato ad una fase di preparazione in cui scannerizzare tramite algoritmi di SLAM più mappe di minor dimensione (100 m), allinearle su un server e calcolare una mappa composta

da allineamenti non perfetti (ricerca del miglior allineamento possibile) per poi, in seguito, creare contenuti AR sul modello 3D calcolato. I contenuti in realtà aumentata vengono poi posizionati sul modello globale della mappa, mentre la posa viene calcolata rispetto all'origine della sotto-mappa più vicina. A questo punto l'Hololens indossato da un utente, quando invia una richiesta al server, viene localizzato rispetto alle coordinate del centro della mappa minore in cui si trova, permettendo al server di apprendere la relativa trasformazione e calcolare la sua posizione nella mappa globale, dunque lo "scene graph" e di tramettere quest'ultimo agli altri utenti. Holoroyale è stato sviluppato su Unity 3D per dispositivi Microsoft Hololens, utilizzando un controller Xbox One S e una rete wi-fi mobile. Il team di sviluppo ha scelto infine di utilizzare delle web api RESTful per interagire col Server e usare un database PostgreSQL per memorizzare le pose statiche.

Gameplay e limitazioni

Il gameplay del videogioco consiste nella collaborazione di molti utenti nel difendere quattro punti chiave all'interno di un'area urbana da un attacco di robot virtuali. Una volta sconfitte le varie orde, appaiono degli obiettivi da raggiungere, che sono delle zone di interesse che necessitano della permanenza dei giocatori al loro interno per 10 secondi per essere conquistate. La partita finisce se i robot distruggono uno dei punti chiave, che sono delle basi da difendere, o se i giocatori, dopo aver sconfitto le orde di robot e raggiunto le varie zone d'interesse, sconfiggono il boss finale.

Nel corso dello sviluppo del videogame si è andati incontro a diverse difficoltà che hanno comportato alcune modifiche al disegno iniziale del gameplay. I giocatori, anziché interagire direttamente con l'ambiente, sono seguiti da due droni virtuali che rispondono ai loro comandi: questa soluzione è stata trovata per ovviare al problema di sincronizzazione delle pose ancora presente nonostante le accortezze per migliorare il sistema. Inoltre per risolvere il problema di alcune zone in cui gli utenti non si devono recare, o per limiti della mappa, o per problemi con il funzionamento dell'attrezzatura in quel preciso spazio (dark areas), sono state introdotte delle barriere diegetiche che sono state soggetto di osservazioni particolari sugli effetti che causano sull'esperienza di gioco.

Collaborazione

La parte collaborativa in questo videogioco si declina in vari modi:

- Chat: è presente un sistema di chat vocale e di chat asincrona (emoticon) che servono per mettere in comunicazione i giocatori tra di loro. Queste, durante le sessioni di testing, sono servite ai giocatori per aiutarsi, ad esempio, con

la navigazione della mappa per raggiungere più velocemente delle posizioni chiave. La riflessione alla base della scelta di un sistema di comunicazione misto (testuale/vocale) è simile a quella intrapresa da molti giochi d'azione che per aumentare il coinvolgimento e la collaborazione optano per questo sistema di comunicazione.

- Sfide congiunte: le sfide affrontate durante le sessioni di gioco richiedono la collaborazione di gruppetti e l'organizzazione tra gli stessi per coprire efficacemente le zone da difendere. Inoltre la sfida finale è un grosso incentivo per la collaborazione di tutti i giocatori durante gli ultimi secondi della sfida.
- Aggiornamento della mappa: questa parte di collaborazione è utile a mantenere il sistema ed è sconosciuta ai giocatori. Dato che l'hololens è tracciato tramite l'origine della sotto-mappa più vicina, una volta che una mappa non riesce ad essere localizzata, essa viene contrassegnata dal singolo client sul server. Una volta che riceve tre segnalazioni, l'origine viene rimossa e in seguito verrà calcolato un nuovo modello e una nuova origine che verranno caricati al posto della mappa contrassegnata sul server.

La collaborazione dunque in un questo videogioco non solo è presente tra i giocatori, ma anche tra i dispositivi che compongono il sistema per mantenere coerenza ed efficienza all'interno dell'esperienza.

2.4.2 MathBuilder

MathBuilder è un serious game che viene esposto nell'articolo "MathBuilder: A Collaborative AR Math Game for Elementary School Students" [28] ed è un RPG disponibile per smartphone che punta, oltre che a insegnare la matematica ai bambini in maniera divertente e coinvolgente, anche a collaborare e ad aiutarsi reciprocamente nelle fasi di studio. Questo Serious Game è stato sviluppato su Unity tramite l'SDK VuForia e per la parte di networking è stato utilizzato UNet (o Unity networking ora deprecato in favore di MLAPI). Il gameplay consiste nella costruzione di edifici di una città tramite le risposte ad alcuni quiz matematici da risolvere singolarmente e in gruppo. Ci sono quattro tipologie di giocatori (muratore, carpentiere, decoratore e designer) che possono collezionare diverse tipologie di materiale, utili alla costruzione degli edifici, rispondendo alle proprie domande matematiche inerenti ad uno specifico argomento in base alla zona della città in cui si vuole costruire: la mappa della città è fisica e, tramite marker, permette la visualizzazione in realtà aumentata dei palazzi in costruzione, che saranno migliori a seconda della correttezza delle risposte date.

La caratteristica sociale di questo videogioco consiste nella domanda corale, che richiede il confronto di tutti i membri del team (composto dalle quattro tipologie

diverse di giocatore) e che consiste in una domanda più complessa di quelle dei singoli. Si è notato inoltre che, quando i bambini finivano prima le proprie domande, era propensi ad aiutare i compagni, stimolando la discussione matematica riguardo all'argomento.

Nelle interviste successive ai test svolti su una classe del quinto anno delle elementari sono sorte alcune considerazioni interessanti dai bambini riguardo il lato sociale e del design del gioco:

- Collaborazione tra compagni: gli studenti si sono ritrovati a discutere con i propri compagni riguardo argomenti scolastici, aiutandosi reciprocamente e non dovendo chiedere aiuto ai professori. Questo ha evitato situazioni di disagio, poichè dall'intervista risulta che i bambini pensavano di far sprecare tempo ai professori per far ripetere alcuni concetti non chiari, e uno studente ha fatto presente che non voleva chiedere aiuto per non sembrare meno intelligente e fare cattiva impressione sull'insegnante. La caratteristica sociale di questo gioco permette di creare un ambiente gradevole per il target che, oltre a imparare e approfondire argomenti solitamente ostici, permette anche di evitare situazioni che ne compromettono l'apprendimento o la comprensione di concetti chiave.
- Senso di novità riguardo al gioco: i bambini hanno apprezzato che il gioco, a differenza di altri serious game in realtà aumentata che avevano provato precedentemente, non sembrasse un "gioco riciclato" con l'implementazione della matematica, ma che fosse un'esperienza che appariva come nuova. Il gioco in realtà risulta essere un banale quiz game, che però riesce a trasmettere un senso di novità andando a utilizzare risorse, ricompense ed effetti visivi (grazie anche alle tecnologie implementate come la realtà aumentata) efficaci e studiati sul target di riferimento.
- Senso di collaborazione e ricompensa migliorati rispetto ad altri giochi: le ricompense sono state particolarmente apprezzate dagli studenti che hanno trovato questi collezionabili degli ottimi incentivi per progredire nel gioco. Inoltre gli studenti hanno molto apprezzato il fatto che potessero collaborare e confrontarsi mentre solitamente in classe non gli era permesso, stimolando anche interesse nella discussione grazie ai quiz di gruppo. Ne risulta un'esperienza videoludica che acquisisce molto valore grazie ad una giusta implementazione della dimensione sociale.

2.4.3 Strumento per l'ispezione di edifici prefabbricati

Questo tool, che viene presentato nell'articolo "A collaborative augmented reality annotation tool for the inspection of prefabricated buildings" [29], si pone come

obiettivo quello di poter permettere a coloro che ispezionano edifici pre-costruiti di lasciare annotazioni geolocalizzate in realtà aumentata tramite smartphone e di facilitare dunque la collaborazione tra vari ispettori nell'identificazione di eventuali difetti di costruzione.

Il funzionamento dell'applicativo permette in primis la sovrapposizione di un modello 3D delle dimensioni corrette sul prefabbricato tramite marker, in modo tale che la sovrapposizione sia sufficientemente precisa tra mondo reale e mondo virtuale e successivamente si possano individuare le discrepanze con maggiore semplicità. Gli utenti possono in seguito lasciare un'annotazione testuale, o grafica (immagine RGB, infrarossi), scannerizzare un modello 3D dell'imperfezione oppure disegnare linee utili a evidenziare le irregolarità. L'annotazione infine è visibile agli altri utenti come una sfera localizzata nei pressi della zona esaminata.

La parte collaborativa di questo progetto è asincrona e richiede dunque che le annotazioni vengano salvate su un server e che esse possano essere accessibili ai vari utenti, inoltre i modelli 3D con le dimensioni ideali dei modelli prefabbricati sono stati opportunamente salvati sul server prima dell'ispezione di test. L'applicativo è stato sviluppato su Unity 3D grazie all'utilizzo delle librerie per la realtà aumentata ARKit e Vuforia strumenti già analizzati in precedenza. La parte relativa al salvataggio su server invece è stata gestita tramite l'utilizzo di NodeJS e un database MongoDB: per salvare e accedere alle risorse, l'applicazione utilizza richieste tramite il protocollo di rete HTTP.

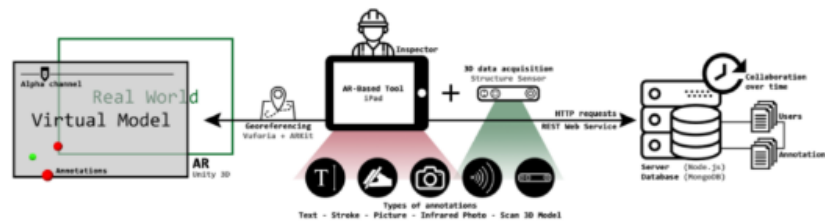


Figura 2.8: Schema rappresentante i componenti dell'applicativo

Considerazioni finali

Tutti e tre i casi studio analizzati implementano la collaborazione in tre diverse modalità, in diversi contesti e utilizzando diverse tecnologie. Per il design e la definizione di come sarà strutturato il progetto di questa tesi, è dunque necessario estrapolare delle considerazioni sui concetti fondanti delle esperienze collaborative per poi declinarli in maniera opportuna in seguito.

Tempo e Spazio

Le più comuni dimensioni utilizzate per classificare i sistemi CSCW (Computer Supported Collaborative Work) sono tempo e spazio: il tempo indica se l'applicativo è sincrono o asincrono, lo spazio se il sistema è in remoto o co-localizzato [30]. Come si è visto nei casi studio precedenti queste due dimensioni definiscono l'esperienza: un videogioco co-localizzato e sincrono come Holoroyale, richiede determinate caratteristiche come, ad esempio, la chat vocale o interazioni e coerenza verosimili all'interno dell'esperienza.

Simmetria

Come riportato da Sereno et al. (2020)[30] ci sono due tipologie di simmetria: tecnologica e di ruolo. La prima consiste nell'utilizzo di device uguali nell'esperienza nel caso di simmetria tecnologica, nel caso contrario l'esperienza si definisce tecnologicamente asimmetrica; la seconda simmetria riguarda i ruoli degli utenti che possono svolgere mansioni uguali (simmetria) o diverse (asimmetria). Come visto nei capitoli precedenti dello stato dell'arte, l'asimmetria di ruolo si sposa bene con quello che sono i giochi intergenerazionali, poichè permette di delineare esperienze più coinvolgenti tenendo conto dei gusti e della capacità derivanti dall'età. Nei casi studio analizzati in questo capitolo le esperienze sono tutte simmetriche (in MathBuilder[28]) ci sono quattro ruoli diversi ma l'esperienza e le mansioni sono le stesse), ma in Age Invaders [22] l'asimmetria di ruolo serve a coinvolgere anche i genitori dei bambini in remoto, creando un ulteriore livello di collaborazione. Sarà necessaria un'attenta valutazione in fase di ideazione del progetto di questa tesi per capire come gestire al meglio la simmetria di ruolo per creare una forte collaborazione tra i giocatori, cucendo su questi ultimi l'esperienza più coinvolgente possibile.

Conclusioni

È noto come, nelle esperienze collaborative, per una cooperazione che coinvolga più utenti, siano necessari i dati per la sincronizzazione e la comunicazione tra i componenti, poichè ogni singolo utente deve poter scambiare le informazioni necessarie all'interno del gruppo di gioco [31]. In questo capitolo sono stati individuati gli strumenti tecnologici e di design tramite i quali sono stati realizzati i progetti analizzati al fine di avere una visione di insieme che permettesse l'ideazione di un'esperienza collaborativa funzionale.

2.5 Valutazione dei tools esistenti

In questa sezione saranno analizzati sinteticamente i principali tools per la realtà aumentata e la gestione dei dati GPS presenti sul mercato.

2.5.1 AR Foundation

È un package che permette l'implementazione di applicazioni in realtà aumentata su Unity. Esso presenta un'interfaccia per gli sviluppatori e non implementa alcuna caratteristica aggiuntiva, ma definisce un'API multiplatforma che permette di lavorare contemporaneamente per diversi dispositivi di destinazione. Per utilizzare AR Foundation c'è la necessità di installare differenti packages a seconda del dispositivo di destinazione:

- ARCore XR Plugin per Android
- ARKit XR Plugin per iOS
- MagicLeap XR Plugin per MagicLeap
- Windows XR Plugin per HoloLens

Nell'immagine sono elencate le principali features che possono essere implementate grazie all'installazione dei diversi pacchetti sopracitati e grazie all'azione di AR Foundation. Si può notare un maggiore numero di features per il pacchetto ARKit, utile per la realizzazione di applicazioni IOS, mentre ARCore implementa meno funzionalità.

Limiti

AR Foundation presenta alcune diversità nelle funzionalità offerte per ARKit e ARCore, che potrebbero essere utili (per esempio funzionalità che permettono la localizzazione della posizione e dell'orientamento di altri dispositivi in sessioni condivise). Dunque questa soluzione, integrata con l'installazione dei relativi pacchetti per lo sviluppo sui diversi dispositivi di destinazione, presenta un buon set di funzionalità di base molto utili per lo sviluppo di un'applicazione AR, ma deve essere opportunamente integrata con altre librerie e tools per poter raggiungere lo scopo di questa tesi).

Licenza

AR Foundation non presenta una licenza a pagamento, ma è gratuitamente scaricabile da Unity, come per i pacchetti ARCore XR Plugin per Android, ARKit XR Plugin per iOS, Magic Leap XR Plugin per Magic Leap, Windows XR Plugin per HoloLens.

Feature Support Per Platform

You can refer to this table to understand which parts of AR Foundation are relevant on specific platforms:

	ARCore	ARKit	Magic Leap	HoloLens
Device tracking	✓	✓	✓	✓
Plane tracking	✓	✓	✓	
Point clouds	✓	✓		
Anchors	✓	✓	✓	✓
Light estimation	✓	✓		
Environment probes	✓	✓		
Face tracking	✓	✓		
2D Image tracking	✓	✓	✓	
3D Object tracking		✓		
Meshing		✓	✓	✓
2D & 3D body tracking		✓		
Collaborative participants		✓		
Human segmentation		✓		
Raycast	✓	✓	✓	
Pass-through video	✓	✓		
Session management	✓	✓	✓	✓
Occlusion	✓	✓		

Figura 2.9: Features implementabili attraverso i diversi plugin

2.5.2 AR + GPS Location

AR + GPS Location è un pacchetto, scaricabile sull'asset store di Unity, che permette di posizionare oggetti 3D in determinate coordinate geografiche nel mondo reale tramite GPS utilizzando Unity e la realtà aumentata. Questo pacchetto supporta sia AR Foundation, sia Vuforia, e permette di legare gli elementi di realtà aumentata con dati riguardanti la geolocalizzazione.

Caratteristiche principali

- Possibilità di posizionare nel mondo reale oggetti 3D, andando a specificare latitudine, longitudine e altitudine.
- Hotspot per la realtà aumentata che si attivano quando l'utente è vicino alla posizione data.
- Possibilità di posizionare informazioni testuali in realtà aumentata in posizioni di interesse.
- Posizione e direzione del dispositivo aggiornate in maniera fluida.

- Possibilità di muovere o piazzare oggetti lungo un percorso su una mappa.
- Implementazione delle ombre degli oggetti 3D in realtà aumentata.
- Implementazione di curve Spline e Catmull-rom general purpose.

Limiti

- L'informazione sull'altitudine solitamente è imprecisa, dunque è consigliato un utilizzo dell'altitudine relativa al device o ai piani individuati tramite tracking.
- Su molti device Android la modalità landscape non funziona o causa problemi riguardanti la compensazione dell'inclinazione sui dati del sensore magnetico.
- Non funziona bene in esperienze in interni.

Licenza

Questo pacchetto ha un costo di 70,57€ sull'Asset Store di Unity.

2.5.3 Vuforia

È un SDK utile per creare applicazioni in realtà aumentata. Gli sviluppatori possono facilmente implementare funzionalità avanzate di computer vision, permettendo all'applicazione sviluppata di riconoscere oggetti, immagini e interagire con lo spazio nel mondo reale. Vuforia Engine offre supporto per applicazioni per Android, iOS, Lumin e UWP.

Caratteristiche principali

Vuforia mette a disposizione dei tools per la creazione di target a cui poter legare contenuti 3D in realtà aumentata:

- Area Target Generator: è un'applicazione desktop che permette di creare degli Area Target da scan 3D di un ambiente.
- Model Target Generator : è un'applicazione desktop che permette di creare Model Targets da oggetti 3D preesistenti.
- Vuforia Area Target Creator: è un'applicazione completa per scannerizzare, generare e testare gli Area Targets.
- Area Target Test App: applicazione che aiuta a valutare rapidamente la qualità e la copertura di un set di dati dell'area Target.

- Model Target Test App: applicazione che assiste nella valutazione dei Model Target, delle sue Guide Views e del suo raggio di riconoscimento.
- Vuforia Object Scanner: applicazione che aiuta a scannerizzare piccoli oggetti 3D e a trasformarli dunque in file da caricare su Vuforia Target Manager che a sua volta crea gli Object Target.

I target calcolabili attraverso i tools sopraelencati vengono poi utilizzati per diverse metodologie di tracking:

- Tracking di immagini
- Tracking di oggetti
- Tracking di ambienti

Vuforia inoltre fornisce altre funzionalità:

- Vuforia Fusion: pensato per offrire un’esperienza di sviluppo univoca indipendentemente dal device di destinazione. Esso rileva le capacità del dispositivo (ARKit/ARCore) e le fonde con le funzionalità di Vuforia Engine, consentendo agli sviluppatori di fare affidamento su una singola API Vuforia.
- Device Tracking: tracking preciso e accurato grazie all’utilizzo di Positional Device Tracker che permette il tracciamento di un target anche quando lo stesso non rientra più nella vista della camera.
- Cloud Recognition Service: permette il riconoscimento di un ampio set di immagini che viene aggiornato frequentemente con nuove immagini.
- Recording and Playback: il Session Recorder permette di registrare e rivedere la sessione di realtà aumentata per permetterne il test e apportare migliorie anche una volta che i target e le loro posizioni non sono disponibili.
- Driver Framework: consente agli sviluppatori di fornire e usufruire dati da sistemi esterni tramite Vuforia Engine. Gli sviluppatori possono inoltre utilizzare Driver Framework per registrare e creare driver personalizzati per i loro dispositivi personalizzati.
- External Camera: possibilità di accedere ai dati video da una fotocamera esterna a quella di un telefono o un tablet durante la creazione di esperienze AR. La fotocamera esterna viene utilizzata in estensione del Driver Framework.

Licenza

Vuforia presenta una licenza gratuita di prova e diverse tipologie di licenze per diverse tipologie di applicazioni:

- Per grandi aziende
- Per piccole aziende
- Per istituti accademici

Il pacchetto per istituti accademici ha una licenza che costa oltre 3000 dollari americani. Vuforia concede anche una licenza gratuita che però non permette la pubblicazione di contenuti.

2.5.4 Wikitude SDK

Wikitude SDK è un libreria software per applicazioni mobile utile per creare esperienze di realtà aumentata. L'SDK nativa supporta casi d'uso che richiedono il riconoscimento delle immagini e tecnologie di tracking.

Caratteristiche principali

Wikitude presenta molte funzionalità che sono implementabili a seconda della licenza di cui si è in possesso:

- Studio Editor: Wikitude mette a disposizione una piattaforma che permette di creare, gestire, testare e pubblicare applicazioni in realtà aumentata senza dover implementare codice.
- Rendering Engine: motore di rendering interno che permette la creazione di oggetti 3D digitali.
- GEO AR: possibilità di implementare installazioni in realtà aumentata in punti geografici di interesse.
- Image Tracking: possibilità di implementare oggetti in realtà aumentata su specifiche immagini 2D (ad esempio un QR code o un volantino).
- Object Tracking: possibilità di implementare oggetti in realtà aumentata su specifici oggetti 3D.
- Scene Tracking: possibilità di implementare oggetti in realtà aumentata su strutture estese e complesse.

- Instant Tracking: possibilità di piazzare oggetti in realtà aumentata su superfici fisiche senza l'utilizzo di marker.
- Cloud Recognition: soluzione di archiviazione online per tracciamento delle immagini su larga scala.
- Cylinder Tracking: possibilità di implementare realtà aumentata su oggetti di forma cilindrica.
- URP support.
- Multiple Object Tracking: possibilità di aggiungere oggetti in realtà aumentata su oggetti reali multipli.
- Multiple Trackers: possibilità di combinare le diverse tipologie di trackers (object, image, positional, ecc...)
- Enterprise App: applicazioni non pubbliche utilizzate per una distribuzione interna all'azienda.
- 3D Model Object Tracking + Alignment Initializer: possibilità di usare modelli tridimensionali per permettere un potente object tracking e di inizializzare l'allineamento per guidare al meglio il punto di vista dell'utente.

Inoltre, sempre a seconda della licenza di cui si è in possesso, si può usufruire di diversi servizi:

- SDK Updates: aggiornamenti dell'SDK.
- Extensions: estensioni per Unity, Cordova, Xamarin e Flutter.
- Studio Hosting: possibilità di archiviare sui server di Wikitude i progetti AR a cui si lavora.
- Multiple Apps: è possibile integrare l'SDK di Wikitude i più applicazioni.
- Studio API: è possibile generare collezioni di target programmaticamente.
- Enterprise API/Script: possibilità di generare collezioni di target localmente.
- Self Hosting: possibilità di fare da host sulla propria infrastruttura server per progetti AR.

Licenza

Sono presenti quattro diversi pacchetti con prezzi diversi:

- PRO 3D(One-time fee): versione che costa 2490 euro ma non include gli aggiornamenti.
- PRO 3D(Subscription): versione che costa 2990 euro l'anno e che include gli aggiornamenti.
- CLOUD(cloud-hosting): permette il cloud hosting oltre a tutte le funzionalità di PRO3D Subscription e costa 4490 euro l'anno.
- ENTERPRISE : licenza custom richiesta per smart glasses, applicazioni aziendali o applicazioni che superano il milione di installazioni.

2.5.5 Lightship ARDK

È una piattaforma gratuita, disponibile sia per Android che per iOS, che permette la creazione di esperienze in realtà aumentata. ARDK (Augmented Reality Developer Kit) è un package installabile su Unity che permette di coniugare la tecnologia di mapping real-time sviluppata da Niantic, con esperienze singleplayer e multiplayer sia per sistemi operativi Android che iOS.

Caratteristiche principali

- Real-time mapping: generazione di una mesh derivata dalla scansione della fotocamera al fine di permettere interazioni realistiche tra l'ambiente e gli oggetti in realtà aumentata.
- Multiplayer Integrato: ARDK offre funzionalità utili anche alla creazione di esperienze multiplayer, offrendo la possibilità di agire sia ad alto livello, che a basso livello.
- Semantic Segmentation: possibilità di classificare oggetti noti (acqua, erba, cielo, ecc) inquadrandoli direttamente con la telecamera
- Possibilità di generare strade a partire dai dati reali

Limiti

- Scansione real time impossibile su larga scala e dunque esperienze limitate a spazi di dimensioni ridotte.

- Scansione difficoltosa se non vengono rispettati alcuni requisiti di illuminazione e contrasto
- L'esperienza multiplayer necessita l'allineamento di una mesh scannerizzata per permettere la coerenza spaziale tra dispositivi.
- Tool molto recente e quindi scarsa presenza di discussioni e forum online.
- Necessità di telefoni aggiornati per poter accedere a tutte le funzionalità (iphone 8 o più recenti, Android high-end usciti dopo il 2018).

Licenza

ARDK, essendo una novità rispetto ad altre librerie, offre una licenza molto conveniente che offre accesso a tutte le funzionalità gratuitamente, eccetto per le applicazioni che superano i cinquantamila utenti mensili per cui il prezzo è di appena 5 euro al mese.

2.5.6 Unity MARS

MARS è un'estensione di Unity che fornisce la possibilità di considerare oggetti ed eventi del mondo reale come GameObject. Si tratta di un tool ad alto livello, che quindi fornisce anche un'interfaccia di controllo che rende il tool utilizzabile anche senza occuparsi del codice. MARS inoltre fornisce una modalità di simulazione che consente di testare i contenuti prodotti dall'utente in diversi contesti del mondo reale e in un breve tempo di interazione.

Caratteristiche principali

MARS introduce come nuove funzionalità:

- AR WYSIWYG: editor che permette di creare applicazioni visivamente complesse e data-oriented. Ha una finestra personalizzata per i creatori include strumenti e interfacce che permettono di vedere, prototipare e testare le applicazioni di realtà aumentata come se fossero provate in un ambiente reale.
- Riconoscimento intelligente del mondo reale: sistema di query che permette alle applicazioni di estrarre dati da qualsiasi fonte per includerli nell'esperienza AR (es: riconoscimento piani di certe dimensioni).
- Test in the Unity Editor: possibilità di testare direttamente nell'editor di Unity senza dover aver accesso a dati del mondo reale o senza dover testare fisicamente l'esperienza in ambienti forniti direttamente in MARS.

- Fuzzy Authoring: possibilità di usare gizmos per definire misurazioni minime e massime per oggetti del mondo reale piuttosto che codificare misurazioni precise.
- Multiplatform AR dev framework: possibilità di creare un'esperienza distribuita su più dispositivi (quelli supportati da AR Foundation).
- Facilitazione sviluppo AR: MARS permette anche un'esemplificazione della procedura per lo sviluppo di applicazioni in AR poiché permette di scrivere meno codice e gestire molti comportamenti dall'interfaccia aggiuntiva.

Licenza

La licenza richiede una subscription annuale di 600 dollari americani. C'è la possibilità di provare Unity MARS per una prova gratuita di quarantacinque giorni.

2.5.7 Mapbox - Maps SDK

Maps è un SDK per unity che permette l'utilizzo di mappe e dati annessi (traffico, punti di interesse, ecc) al fine di legare l'esperienza di gioco alla posizione geografica dell'utente.

Caratteristiche principali

- Editor Mappe a nodi per poter creare mappe visivamente personalizzabili
- Possibilità di generazione di mappe virtuali basate su dati reali allo scopo di garantire un'esperienza coinvolgente
- Possibilità di integrare nell'esperienza virtuale punti d'interesse già noti a MapBox e di poterne aggiungere di nuovi
- Map texturing: possibilità di texturizzare le mappe (sono modificabili le texture, i colori e le proporzioni)

Limiti

Allo scopo della nostra tesi lo strumento risulta molto utile ma da implementare con logiche di Spatial Augmented Reality che mancano nella descrizione dell'SDK. L'implementazione di tale strumento in tali applicazioni è dimostrato nella sezione degli esempi pratici. Un limite di Mapbox è rappresentato dall'impossibilità di utilizzo delle features in realtà aumentata incluse nella libreria nelle ultime versioni di Unity (2020 e 2021).

Licenza

L'integrazione di Maps SDK in unity è gratuita fino al raggiungimento di 25000 utenti attivi mensilmente.

2.5.8 ArcGis

Si tratta di un SDK ancora in beta, (attualmente alla versione 0.3.0). È possibile accedere al beta program e scaricare, con lo scopo di testarla e contribuire al suo miglioramento, l'ultima versione disponibile. ArcGis può essere integrato con Unity e con Unreal Engine: poiché in questa tesi verrà usato Unity, valuteremo ArcGis rispetto a questo game engine.

- Cross platform: è possibile realizzare app location based su WIndows e MacOS e destinarle sia a PC (Windows o MacOS) che a dispositivi mobili (Android e iOS)
- Possibilità di generazione di mappe virtuali basate su dati reali
- Possibilità di integrare nell'esperienza virtuale punti d'interesse (utilizzando ad esempio il database di OpenStreetMap).
- Possibilità di aggiungere livelli creati da altri utenti (edifici, strade, informazioni meteo) e di creare i propri.
- Utilizzabile solo con HDRP e URP.

Limiti

Trattandosi di una beta, l'utilizzo di ArcGis non garantisce il 100

Licenza

La versione beta di ArcGis è gratuita.

Capitolo 3

Tecnologie utilizzate

In questo capitolo verranno discussi e analizzati in modo approfondito gli strumenti adottati per lo sviluppo del progetto di tesi, ponendo l'accento sulle caratteristiche principali di ciascun tool e sulle motivazioni che hanno portato al loro utilizzo.

3.1 Unity3D

Unity3D [32] è uno tra i più popolari motori grafici che permettono lo sviluppo di videogiochi e contenuti interattivi per più piattaforme. Presenta sia una versione gratuita che una versione Pro a pagamento. Unity3D inoltre è disponibile per i sistemi operativi più utilizzati (Windows, MacOS e Linux) e permette la creazione di applicazioni funzionanti per MacOS, Windows, Linux, Web Browser (questa funzionalità è presente solo per Windows e MacOS), device mobili Android e iOS, console Nintendo, Playstation e Microsoft. Questo specifico motore grafico è stato programmato in C/C++ e richiede che la componente di codice di ogni applicativo sia scritta in c e Javascript. Unity3D presenta al suo interno due sottosistemi: l'Editor e l'Engine. Il primo, Unity Editor, rappresenta l'interfaccia con il quale lo sviluppatore interagisce direttamente, mentre l'Engine è ciò che gestisce la fisica, il rendering e l'interazione tra editor e meccanismo di scripting.

3.1.1 L'interfaccia

L'interfaccia di Unity è composta da più sottofinestre, ognuna responsabile di una specifica funzionalità. Essendo l'interfaccia customizzabile, di seguito verrà illustrata l'interfaccia di default di un progetto.

- Inspector: Sottofinestra in cui è possibile andare a modificare, assegnare o rimuovere le proprietà, dette component, che mette a disposizione Unity o le SDK installate, di un singolo GameObject come ad esempio la Transform,

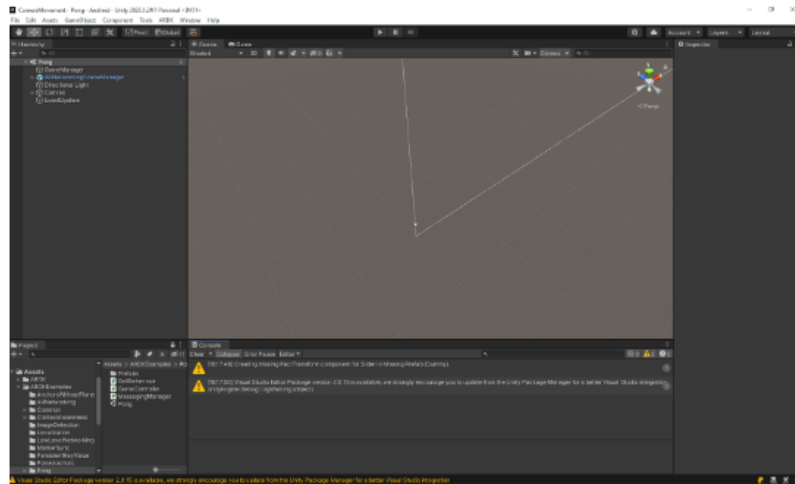


Figura 3.1: La schermata di default di Unity

responsabile della gestione della posizione e rotazione dell'oggetto nel sistema di riferimento dell'ambiente virtuale. L'inspector permette di aggiungere inoltre degli script custom scritti dallo sviluppatore al fine di controllare tramite codice i comportamenti dell'oggetto stesso e anche dell'ambiente circostante.

- **Hierarchy:** Unity rappresenta gli oggetti presenti nell'ambiente virtuale tramite un sistema gerarchico; questa gerarchia è visibile nella sottofinestra della Hierarchy, dove possono essere manipolati i legami di parentela tra oggetti. È importante ricordare che oggetti figli ereditano dal padre le trasformazioni che vengono applicate a quest'ultimo.
- **Project:** sottofinestra dedicata alla rappresentazione delle cartelle in cui vengono salvati tutti i file presenti nel progetto (oggetti 3d, file audio, immagini, ecc...) e dalla quale è possibile selezionarli per utilizzarli nello sviluppo del prodotto finale.
- **Scene view:** in questa sottofinestra è possibile manipolare l'ambiente virtuale delle scene selezionate: oltre ad aggiungere gameobject, è possibile disporli nel sistema di riferimento dell'ambiente.
- **Game view:** è la sottofinestra dedicata alla simulazione del progetto: qui è possibile eseguire l'applicativo per testarlo e verificare la correttezza del funzionamento dello stesso.
- **Console:** è la sottofinestra in cui vengono mostrate le stampe di errore, warning e di log e risulta molto utile per la fase di debug dell'applicativo.

L'interfaccia di Unity è personalizzabile: è infatti possibile modificarla aggiungendo altre sottofinestre e spostandole per configurare lo spazio di lavoro più adatto ad ogni esigenza.

3.1.2 Gestione della fisica

Unity3D per la gestione della fisica utilizza un motore fisico sviluppato da NVIDIA chiamato PhysX. Esso è responsabile di rilevare collisioni e calcolare le forze risultanti fornendo la possibilità di adoperare alcuni component per facilitare il rilevamento e la gestione di questi fenomeni da parte degli sviluppatori. Tra i più noti strumenti forniti dal PhysX ci sono:

- **Rigidbody**: component che gestisce un oggetto come un corpo rigido, e dunque gli permette di essere soggetto a forze o imprimerle su altri oggetti.
- **Collider**: component che consiste in un bounding box che rileva le collisioni di altri GameObject all'interno del suo volume. Esso può avere varie forme primitive (sferico, prismatico, a forma di capsula), ma può anche assumere la forma di specifiche mesh (Mesh Collider) per rilevare collisioni in maniera più precisa ma aumentando in maniera ingente il costo computazionale.
- **Raycasting**: funzionalità che permette di programmare dei raggi a partire da una sorgente, utili a rilevare la collisione del raggio stesso con altri oggetti. Questi raggi vengono definiti da codice.

Unity dà la possibilità però di modificare o personalizzare le regole fisiche dell'ambiente e molte altre caratteristiche attraverso lo scripting.

3.1.3 Scripting

La possibilità di scrivere frammenti di codice per poter controllare i comportamenti degli oggetti, o più in generale dell'applicativo, è fondamentale per Unity, che permette di utilizzare come linguaggi C# oppure JavaScript. Quando uno script viene creato su Unity, eredita di default dalla classe `MonoBehaviour`, e al suo interno potranno essere implementate funzionalità che andranno a comporre la logica di gioco. Di fondamentale importanza è sicuramente la conoscenza e la comprensione di quello che viene chiamato **Lifecycle** di Unity, ovvero ciò scandisce l'invocazione di alcuni eventi della classe `MonoBehaviour`:

- **Awake()**: funzione che viene chiamata appena il GameObject al quale lo script è legato viene creato;
- **Start()**: funzione invocata subito dopo **Awake()**, nel preciso frame in cui lo script viene attivato;

- `Update()`: funzione invocata ad ogni frame, se lo script è attivato all'interno dell'Inspector. La sua frequenza dipende dalla velocità alla quale il computer su cui è eseguito l'applicativo renderizza le immagini;
- `FixedUpdate()`: funzione invocata ad intervalli prefissati, non dipende dalle prestazioni del computer;
- `LateUpdate()`: funzione invocata una volta completate tutte le chiamate alle funzioni `Update()`;

Per la scrittura del codice sono stati utilizzati gli IDE Visual Studio e JetBrains Rider e il linguaggio utilizzato è stato C#, un linguaggio orientato ad oggetti sviluppato nei primi anni 2000 da Microsoft.

Unity3D rappresenta il software di riferimento, insieme ad Unreal Engine, per lo sviluppo di videogiochi o applicazioni virtuali indipendenti e presenta una documentazione molto approfondita oltre che una community vasta, fattori che sono stati decisivi per la scelta di questo software e in seguito nello sviluppo dell'applicativo.

3.2 Lightship ARDK

Lightship ARDK [33] è un toolkit gratuito, disponibile sia per il deployment su Android che su iOS, che permette la creazione di esperienze in realtà aumentata senza necessità di marker. Disponibile da novembre 2021, ARDK (Augmented Reality Developer Kit) è un sdk integrabile in Unity che permette di coniugare insieme la tecnologia di mapping real-time sviluppata di Niantic, con esperienze multiplayer sia per sistemi operativi Android che iOS. ARDK offre, tra le sue caratteristiche principali:

- **Realtime Mapping: Depth Occlusion e Meshing** L'operazione di mapping avviene in real-time, generando una mesh composta di facce triangolari derivata dai colori inquadrati dalla camera del dispositivo mobile utilizzato e interpretati da una rete neurale. Il modello che ne risulta è molto utile per aumentare il grado di realistica delle interazioni tra oggetti virtuali e l'ambiente circostante. Infatti la mesh generata sarà utile per generare occlusioni tra oggetti su diversi gradi di profondità, permetterà di avere delle interazioni fisiche realistiche come ad esempio le collisioni, inoltre permetterà di popolare l'ambiente circostante di oggetti virtuali con features più flessibili rispetto alla classica plane detection che viene utilizzata in altri tools come ad esempio AR foundation, senza necessitare di marker. La mesh generata può contenere informazioni per un massimo di un raggio d'azione di 100 metri intorno all'utente ed è importante considerare che il sistema di riferimento di questa differisce da quello di Unity,

essendo right-handed e allineato con la gravità sull'asse Y. Per facilitare la manipolazione della mesh l'algoritmo la suddivide in cubi di grandezza uguale (lato di 1,4 metri circa) e tiene conto di ogni cubo che contiene almeno un triangolo della mesh generata. L'aggiornamento della mesh poi avviene andando a modificare i cubi nei quali variano il numero di vertici e di facce ma, per evitare grossi costi computazionali, l'applicazione deve limitare al massimo questi aggiornamenti.

- **Networking:** ARDK offre funzionalità utili anche alla creazione di esperienze multiplayer, fornendo la possibilità agli sviluppatori di andare a lavorare sia con componenti di alto livello HLAPI (High Level API), che a basso livello (IMultipeer Networking features). Il multiplayer è supportato da un server stateless che fornisce la gestione delle sessioni, routing dei messaggi e il rilevamento dei peer. Il concetto di sessione è fondamentale per il networking ARDK poiché permette di rendere noto ai dispositivi con quali peer è necessario scambiarsi i messaggi. Ogni sessione è identificata da un id che permette di crearne una o di parteciparvi. Il problema che viene affrontato dal Networking è quello di garantire la coerenza tra i diversi ambienti sui diversi dispositivi e ARDK utilizza una soluzione host-oriented per ovviare a questo problema. Il primo peer che si connette ad una sessione ne diventa automaticamente host: il suo compito sarà quello di scannerizzare l'ambiente circostante e la sua posizione iniziale, dall'inizio della scannerizzazione, verrà considerata come origine (0,0,0) del sistema di riferimento comune. I peer che si connettono successivamente alla sessione, dovranno anche loro scannerizzare lo stesso ambiente, ma le loro posizioni saranno rilocalizzate nelle nuove posizioni corrispondenti al sistema di riferimento dell'host: questo è possibile grazie ad un confronto tra le diverse mesh ottenute dalle scannerizzazioni dei dispositivi. Nelle diverse fasi di questo processo di sincronizzazione i peer assumono diversi stati:
 - **Unknown:** quando il client non è connesso al Niantic Backend, ma è in procinto di farlo;
 - **Waiting For Localization Data:** quando l'host non ha ancora completato il caricamento della mappa iniziale e del suo sistema di riferimento;
 - **Localizing:** il client ha ricevuto la mappa dell'host e deve iniziare a scannerizzare l'ambiente;
 - **Stable:** ARDK è sincronizzato e la mappa su tutti i dispositivi stable e connessi alla stessa sessione è coerente;
 - **Failed:** l'host non ha mai caricato uno scan utilizzabile della mappa e si è disconnesso dalla sessione.

È possibile sincronizzarsi anche tramite markers, includendo in essi sia informazioni utili per localizzare un'origine del sistema di riferimento comune, sia informazioni necessarie per connettersi alla sessione, ma questa metodologia risulta meno precisa e può provocare drift nella posizione e nell'orientamento reciproci degli oggetti virtuali. La soluzione migliore è rappresentata dalla combinazione di localizzazione tramite marker e tramite scan: così facendo, tutti i peer, mentre si occupano della scannerizzazione, assumono le informazioni utili dal marker per iniziare l'esperienza condivisa nel caso alcuni dispositivi fossero più lenti a sincronizzarsi con lo scan e a raggiungere lo stato stabile. Per la gestione del networking, ARDK offre la possibilità di utilizzare un approccio sia ad alto livello (HLAPI), che a basso livello (IMultipeer Networking).

HLAPI

Sistema che lavora out-of-the-box, però generico abbastanza da permettere implementazioni custom a qualunque livello. Introduce uno schema di indirizzamento gerarchico dove ogni oggetto è rappresentato da un NetworkGroup con uno specifico tag NewtorkID: ciascun NetworkGroup contiene dei NetworkedDataHandlers, identificati da stringhe e che permettono la gestione dei dati (come ad esempio la posizione di un oggetto che si muove). HLAPI inoltre mette a disposizione dello sviluppatore infrastrutture utili per velocizzare la pipeline di sviluppo come ad esempio la network transform (componente che invia automaticamente informazioni agli altri peer riguardo la posizione dell'oggetto a cui è attaccato) o i Generic Network Fields (campi che propagano automaticamente aggiornamenti riguardo il valore di una certa variabile mantenendo coerente il suo valore tra i diversi peer) e gestisce, tramite classi intermedie (come HlapiSession e NetworkGroup) i problemi classici del networking come l'addressing, il caching, e la serializzazione di dati.

IMultipeer Networking

Consigliato quando si hanno applicazioni dove non è necessaria una grande varietà di tipologie di messaggi per provvedere alla coerenza dell'ambiente condiviso, l'IMultipeer Networking è un sistema più grezzo, meno raffinato, che permette però allo sviluppatore di lavorare a basso livello. I messaggi di scambio tra i peer risultano più snelli poiché l'intestazione non deve contenere informazioni su NetworkGroup, NetworkID e DataHandlers, ma sono identificati da un intero che indica la tipologia di messaggio (per questo è consigliato avere poche tipologie di messaggio), dalla reliability (UnreliableUnordered, UnreliableOrdered, ReliableUnordered e ReliableOrdered), e dal destinatario. L'IMultipeer Networking offre anche la possibilità di usare coppie chiave e valore persistenti che garantiscono la condivisione di dati e la notifica di eventuali cambiamenti a tutti i client appartenenti ad una sessione, e offre la possibilità ai peers di sincronizzare il proprio clock con il clock del server:

ciò garantisce che tutti i clients abbiano un massimo di 30 ms di discrepanza su `CurrentCorrectTime`.

Purtroppo, trattandosi di uno strumento nuovo, ARDK presenta alcune limitazioni segnalate da Niantic:

- Limiti legati alle sessioni: Niantic consiglia sessioni di gioco per un massimo di 5 giocatori per una durata massima di 5 minuti.
- Affidabilità Messaggi: Ci sono alcuni casi in cui, nonostante il protocollo di trasporto usato sia il TCP, i messaggi possono essere persi ed è dunque importante salvare le informazioni sullo stato in coppie chiave valore.
- Visibilità Peer: E' possibile che un peer connesso non sia immediatamente notificato agli altri, con conseguente perdita di messaggi trasmessi in questo periodo refrattario.
- Context Awareness (Semantic Segmentation): È possibile inoltre sfruttare la segmentazione semantica di ARDK per arricchire l'esperienza e per permettere di classificare oggetti reali che vengono inquadrati dall'utente con la telecamera, come ad esempio cielo o prato. Questo può essere utile per permettere interazioni particolari con superfici specifiche.
- Tools per testing: ARDK offre la possibilità inoltre di testare le funzionalità di realtà aumentata, sia condivisa che non, direttamente nella playmode di unity offrendo strumenti come Virtual Studio (nel caso specifico del networking) o delle scene di mockup dove testare funzionalità come il meshing, plane detection o altro. Inoltre procura con un package aggiuntivo (ARDK Examples) un vasta gamma di scene esempio per aiutare i neofiti, oltre a fornire, sulla pagina principale del sito, la possibilità di consultare un forum.

Lightship ARDK ad oggi rappresenta una realtà all'avanguardia nell'ambito della realtà aumentata accessibile a tutti ma, essendo un toolkit molto recente (ricordiamo che è stato pubblicato nel Novembre 2021), presenta ancora qualche imperfezione ed è difficoltoso reperire documentazione che non sia quella presente sul sito ufficiale.

3.3 Mapbox

Mapbox [34] è un provider Americano di mappe online, nato nel 2010 e utilizzato da applicazioni mobile e siti web come Foursquare, Snapchat e Lonely Planet. Dal 2010 si è affermato sempre di più grazie alla possibilità per gli sviluppatori di customizzare ampiamente le mappe, rispondendo quindi alla poca flessibilità offerta

da altri provider esistenti come Google Maps. Mapbox ha inoltre sviluppato delle librerie open source destinate all'integrazione delle proprie mappe su applicazioni web, iOS e Android.

3.3.1 Mapbox Studio

Mapbox Studio è l'applicativo online necessario per creare, customizzare e gestire le mappe che si desidera implementare. Questo tool consente sia di accedere ai dati di Mapbox, sia di importare i propri, attraverso file .json o .csv, e di personalizzarne la visualizzazione sulla mappa. Il processo che, partendo dalla definizione dei dati, porta alla visualizzazione della mappa finita è diviso in tre sezioni:

- dati geospaziali: Mapbox utilizza i tilesets, ovvero contenitori di dati molto leggeri in grado di gestire sia dati raster che informazioni vettoriali. Ogni volta che i dati vengono caricati su Mapbox Studio, questi vengono tramutati in tilesets per poter essere visualizzati su una mappa.
- stile: una volta raccolti i dati, è necessario definirne la visualizzazione attraverso font e colori. Lo stile viene gestito attraverso lo Style Editor, un'interfaccia interattiva ad alto livello che archivia le informazioni in un documento Json.
- visualizzazione: per renderizzare i contenuti, Mapbox fa uso di una libreria grafica basata su OpenGL, chiamata Mapbox GL, in grado di mettere insieme i dati raster o vettoriali dei tileset con le corrispondenti specifiche relative allo stile.

3.3.2 Maps SDK for Unity

Tra tutti, lo strumento di Mapbox che più si è rivelato utile per il progetto descritto in questa tesi è sicuramente il sdk per Unity, che racchiude tutte le features principali di Mapbox in un'unica grande libreria utilizzabile sul game engine. Oltre alle funzionalità prettamente legate all'implementazione delle mappe, il sdk contiene anche una sezione dedicata all'integrazione dell'AR, il cui funzionamento è basato su AR Foundation. Si tratta di un file .unitypackage scaricabile direttamente dal sito ufficiale di Mapbox e che può essere installato su Unity come custom package. La versione utilizzata in particolare è la 2.1.1, rilasciata il 15 ottobre 2019. Il problema principale riscontrato nell'installazione del pacchetto riguarda proprio la sezione relativa alla realtà aumentata, che abbiamo dovuto escludere poiché in conflitto con le versioni più recenti di AR Foundation, contenute nella versione di Unity che abbiamo deciso di adottare (2020.3.24f1).

3.4 Mirror

Mirror [35] è un API high level per il networking open source implementata per Unity3D grazie alla quale è stato possibile sviluppare la parte collaborativa del progetto di questa tesi. Si colloca sul livello di trasporto per la comunicazione real-time, e gestisce molte delle tasks che sono richieste in un videogioco multiplayer. Mirror è inoltre un sistema server authoritative ovvero permette a uno dei partecipanti di svolgere il ruolo sia di client che di server allo stesso tempo, in modo tale che non ci sia la necessità di avere un server dedicato. In seguito verranno illustrati i concetti chiave e i componenti fondamentali per il funzionamento di questa libreria.

Mirror offre dei vantaggi nella comodità d'uso poiché permette di sviluppare videogiochi multiplayer con una contenuta quantità di lavoro da parte degli sviluppatori, essendo che esso funziona in congiunzione coi servizi internet ed essendo che permette un tipo di workflow iterativo.

3.4.1 Server e Clients

Sono le due classi base per il funzionamento di un'applicazione sviluppata con Mirror. Il server è un'istanza dell'applicativo alla quale tutti i giocatori si connettono quando vogliono giocare insieme, e tiene conto di diversi aspetti di tale applicativo, mantenendo la coerenza dell'ambiente virtuale condiviso (ad esempio il punteggio), mentre i client sono ogni singola istanza del gioco che si connette al server. Questi possono connettersi in locale, ma anche online (di seguito la procedura). Il server può essere sia dedicato, quando c'è un'istanza del gioco che è dedicata soltanto alle funzionalità proprie del server, sia Host server, ovvero uno dei client agisce sia da Server che da client. Ora per comprendere come queste due classi comunichino bisogna introdurre i concetti cardine di Mirror

3.4.2 Authority

Come detto precedentemente, Mirror è un sistema server authoritative e ciò significa che dipende dall'authority, ovvero come e dove un oggetto di gioco sia gestito, e questa solitamente è di appartenenza del server; ciò fa sì che sull'istanza di gioco del server l'ambiente condiviso venga modificato e poi propagato ai rispettivi client mantenendo la coerenza tra le diverse istanze. Mirror però offre anche la possibilità di affidare la local authority ad alcuni gameobject.

3.4.3 Network Identity

Componente che controlla l'identità unica di un oggetto nel gioco sul network, e che è utile affinché il gioco mantenga la coerenza sullo stato di quell'oggetto su tutti i client connessi.

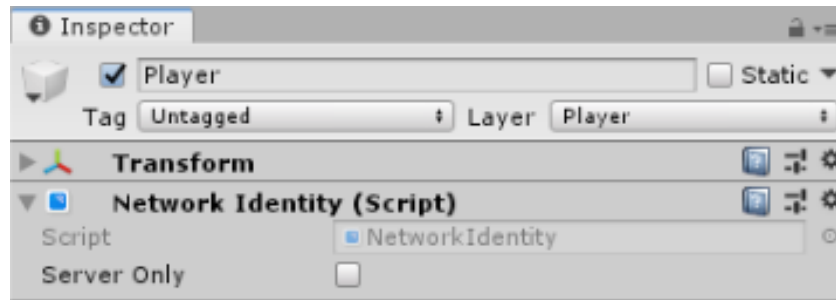


Figura 3.2: Il component Network Identity nell'Inspector di Unity

3.4.4 Network Behaviour

Si tratta di script che funzionano per tutti gli oggetti con una network identity. Questi script permettono di utilizzare funzioni ad alto livello che permettono di comunicare col server per aggiornarlo sulle azioni dei singoli giocatori sulle istanze locali, oltre a fornire numerose informazioni sulle proprietà dell'oggetto (netID, isServer, isClient, isLocalPlayer, ecc.).

Le funzioni che permettono di comunicare col server sono chiamate Commands e sono l'unico modo per i client di attivare del codice sul Server, che è l'unica istanza a possedere l'autorità sugli oggetti. Quando un gameobject di un client esegue una command, questa automaticamente sarà letta dal server come proveniente da quel determinato client e dunque sarà eseguita sul gameobject corrispondente al mandante della command ma sull'istanza del server. Questo fa sì che sia impossibile chiamare una command da un client per un giocatore differente.

Il server risponde a queste funzioni con due diverse tipologie di funzioni di alto livello:

- Client RPC Calls: possono essere mandate da ogni oggetto presente sull'istanza server con una NetworkIdentity, e il contenuto di tale funzione si riflette sul gameobject corrispondente di ogni istanza client collegata al server.
- Target RPC Calls: funzioni ad alto livello chiamate dal server e invocate sull'oggetto del client corrispondente alla NetworkConnection specificata nei parametri della funzione. Se nel primo parametro è presente la NetworkConnection allora il client corrispondente a quella connessione riceverà il messaggio

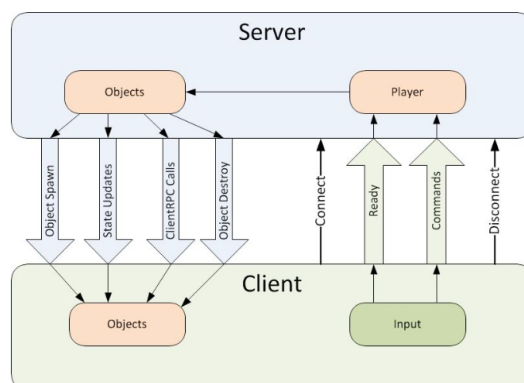


Figura 3.3: Client RPC

senza preoccuparsi del contesto, invece se il primo parametro è di un altro tipo, allora il client padrone del gameobject con lo script contenente la `TargetRpc` riceverà il messaggio.

3.4.5 Network Manager

È il componente che permette il funzionamento dell'intera infrastruttura Mirror, e si occupa della gestione di gran parte degli aspetti di un gioco multiplayer. Esso necessita di un riferimento ad un protocollo di trasporto (nella figura viene inserito quello di default per Mirror, ovvero il `Thelepathy Transport`). Gli sviluppatori di Mirror hanno voluto lasciare la possibilità di scegliere quale protocollo fosse più utile per il proprio videogioco. Vengono supportati trasporti di tipo UDP, TCP, WebGL e Steam e questi possono essere combinati poiché Mirror propone un componente che permette al server di utilizzare due diversi protocolli insieme.

Il `NetworkManager` si occupa anche della gestione degli spawn dei networked objects tramite l'istanziamento di prefab specificati nell'inspector del componente. Il `Player` prefab è il prefab che rappresenta il giocatore, come ad esempio un avatar personalizzato, mentre è possibile dichiarare una lista di altri oggetti che potranno essere spawnati dinamicamente dal server, oppure iscrivere altri prefab direttamente da codice. Inoltre la natura open source di Mirror permette di creare il proprio `Network Manager` con funzionalità additive ereditando dalla classe `NetworkManager`.

3.4.6 NoIP

Precedentemente è stata accennata la possibilità di connettersi a Mirror sia tramite una rete locale ("localhost"), che però comporta che tutti i dispositivi siano collegati

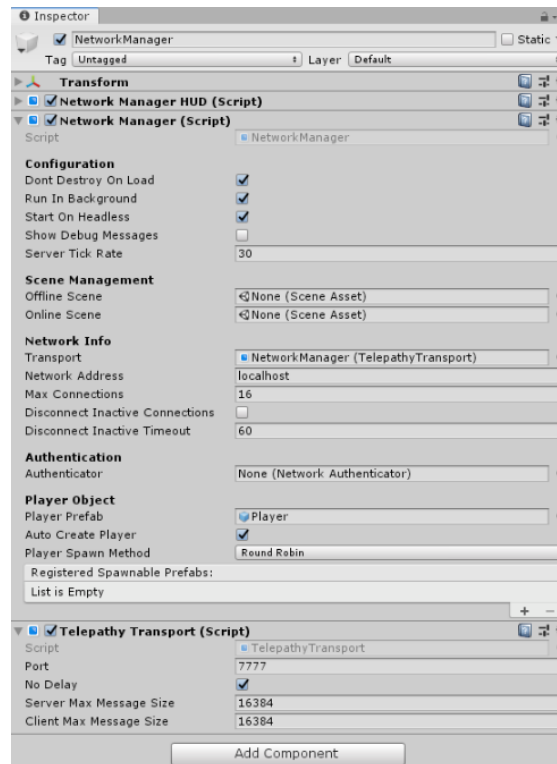


Figura 3.4: I settaggi del NewtworkManager nell'Inspector di Unity

alla stessa sottorete per poter comunicare, sia online. Questa seconda metodologia comporta però che il server sia accessibile dalla rete e questo comporta la necessità di un indirizzo IP statico, oppure di un servizio che assegni un hostname all'indirizzo IP dinamico che si desidera raggiungere, ovvero quello a cui è collegato il server. Per far ciò è stato scelto di sfruttare i servizi messi a disposizione da No-IP, che permette appunto di creare degli hostname custom da assegnare alla propria rete di casa affinché sia accessibile anche dall'esterno, facendo riferimento ad una stringa anziché ad un indirizzo IP. Questo servizio offre la possibilità di ottenere fino a tre hostname gratuiti, che però per rimanere attivi devono essere aggiornati una volta al mese (questo per evitare di avere numerosi hostname che non vengono utilizzati). Ovviamente sono presenti anche pacchetti a pagamento che non richiedono alcun aggiornamento periodico e permettono di gestire più di tre hostname.

3.5 Parrel Sync

L'utilizzo di Mirror ha comportato la necessità di testare il progetto su più istanze contemporaneamente, e questo vuol dire dover buildare ogni versione del videogioco

per verificare il corretto funzionamento. Questo per un workflow rapido e ottimizzato non è funzionale, e dunque è stato utilizzato un tool che permette di aprire più istanze di Unity, indipendenti l'una dall'altra: Parrel Sync. Questo tool è un'estensione di Unity3D che permette di aprire multiple finestre di Unity Editor e che permette che i cambiamenti nella finestra originale si propaghino anche nelle altre finestre, mantenendo così la coerenza tra le diverse istanze. Il funzionamento del tool è molto semplice perché genera dei semplici cloni del progetto, e ciò è possibile direttamente dall'interfaccia di Unity Editor che, una volta installato il package, si aggiorna con la voce ParrelSync. Una volta generato il clone del progetto, questo sarà aggiornato costantemente con le modifiche del progetto originale ogni volta che quest'ultimo verrà salvato. Purtroppo l'utilizzo di questo tool è ingombrante a livello della memoria poiché ogni clone verrà memorizzato e salvato, andando a raddoppiare lo spazio totale occupato dal progetto.

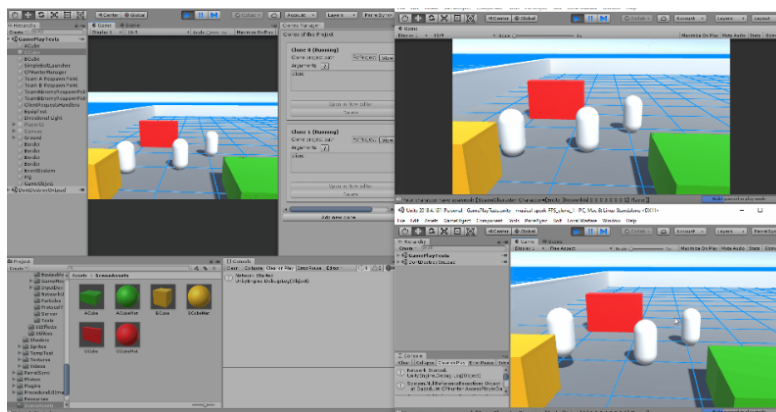


Figura 3.5: Due istanze di Unity in esecuzione tramite Parrel Sync

3.6 Firebase SDK: Authentication e Real-time database

Nello sviluppo del progetto di tesi è stato necessario considerare la possibilità di poter salvare dati relativi alle singole partite per poterli successivamente analizzare. Questa procedura può essere approssciata in diversi modi, sia salvando le informazioni utili in locale, e dunque sul terminale su cui è in esecuzione il server, sia il salvataggio in cloud, ovvero su un server esterno. Siccome è stata scelta la seconda metodologia per rendere l'applicazione più scalabile si è scelto di utilizzare l'SDK per Unity3D proposta da Firebase [36].

Firebase è una piattaforma per la creazione di applicazioni per dispositivi mobili e web acquistata da Google nel 2014 (precedentemente indipendente e fondata nel

2011). In questo progetto sono stati utilizzati due tool forniti da questa piattaforma che ci hanno permesso di implementare un processo di autenticazione, attraverso il canonico metodo di e-mail e password, e un processo di salvataggio su cloud grazie al real-time database. Entrambi i package per Unity3D sono di semplice implementazione e quest'ultima verrà approfondita nel capitolo Design e Sviluppo.

Capitolo 4

Game Design Document

Al fine di comprendere meglio il capitolo successivo, riguardante le scelte e le modalità che hanno guidato lo sviluppo, è necessario approfondire il design dell'esperienza e i vincoli che questo impone sul lato tecnico del progetto.

4.1 Sinossi

MagicTurinAR è un serious RPG location-based intergenerazionale multiplayer in realtà aumentata per iOS e Android, pensato per nuclei familiari in visita a Torino. I giocatori saranno proiettati in una Torino che rischia di tornare sotto il controllo di un Oscuro Sovrano che mille anni prima controllava la città, tormentando e schiavizzando i cittadini attraverso il caos e il suo esercito di demoni. L'Ordine Bianco, garante della pace e del quieto vivere, invierà una squadra specializzata per indagare e trovare il luogo in cui gli adepti dell'Ordine Oscuro, un'organizzazione che nel corso dei secoli ha tramato il ritorno della loro guida, tenteranno di compiere il rituale di evocazione del loro Sovrano. I giocatori verranno proiettati nei panni dei membri del team da cui dipendono le sorti della città di Torino, e attraverso enigmi, l'esplorazione della città e il combattimento con i demoni che la infestano, riusciranno a comprendere i lati più esoterici della storia di Torino e ad evitare che questa ricada nel caos.

La squadra sarà suddivisa in tre ruoli chiave: il saggio, grande conoscitore della magia in tutte le sue forme e risolutore di enigmi; l'esploratore, in grado di orientarsi grazie alle sue abilità magiche, e il cacciatore, il braccio armato del gruppo in grado di fronteggiare qualsiasi creatura. Solo tramite la collaborazione di tutti i membri sarà possibile affrontare le sfide e le difficoltà che i giocatori troveranno davanti a sé.

Le famiglie coinvolte nel gioco verranno guidate tra le strade di Torino attraverso una mappa e potranno raccogliere oggetti e visualizzarli in realtà aumentata:

tramite opportune ricompense riusciranno a raggiungere diversi punti di interesse, corrispondenti ai luoghi esoterici di Torino, e a comprenderne la storia. Durante l'avventura, oltre agli indizi utili a scoprire dove sia il Grande Buio (così si chiama il luogo segreto dell'evocazione), i giocatori saranno in grado di trovare gemme e collezionabili, interagire con altre famiglie per sconfiggere demoni potenti che infestano determinate zone della città, oltre che con numerosi bizzarri personaggi, venendo immersi in un nuovo magico mondo.

Il gameplay di MagicTurinAR porterà le famiglie a spasso per Torino mostrando loro la storia della Torino Magica e i maggiori punti d'interesse della città, migliorando le loro capacità di orientamento e rendendo l'esperienza e la visita turistica piacevole e divertente anche per i più piccoli.

4.2 Target

Il target è costituito dalle famiglie in visita a Torino, interessate a conoscere i misteri della Torino Magica. L'esperienza è pensata principalmente per famiglie comprendenti membri appartenenti ad almeno due generazioni diverse (genitori e figli, nonni e nipoti, ...) e mira a coinvolgere il nucleo familiare all'interno di un'avventura che possa rafforzare i rapporti interpersonali tra i membri, insegnando anche la storia dell'esoterismo torinese.

Il serious game è funzionale ad un maggiore coinvolgimento dei bambini, spesso annoiati durante le visite turistiche. Attraverso meccaniche, storyline e elementi puramente legati alla dimensione ludica, l'esperienza mira a migliorare sia il processo comunicativo tramite il quale viene insegnata la storia di Torino, sia a migliorare le doti di orientamento e ragionamento degli altri giocatori, tramite la proposta di quiz e sessioni di navigazione per le vie della città, facendo leva sul senso di realizzazione che scaturisce dal completamento delle missioni e dalle ricompense. La famiglia ideale a cui il gioco fa riferimento è stata pensata come composta da due persone più anziane (un genitore e un nonno, due genitori, due nonni) e una giovane (un bambino) e di conseguenza i ruoli sono stati cuciti sulle esigenze e sui gusti di queste categorie di età.

Secondo alcune teorie della psicologia, diventando più anziani, le persone tendono a preferire le mansioni a loro note [37], come possibilmente possono essere i Puzzle Games, mentre il lato sociale dei giochi è l'aspetto che si avvicina di più alla loro attitudine [38]. Inoltre essere in grado non solo di giocare con i coetanei, ma anche con gli altri membri della famiglia, soddisfa gli anziani dando un senso di adempimento dei loro obblighi sociali [39]. Per questa fascia di età si è pensato dunque di sviluppare dei ruoli che svolgessero mansioni tramite meccaniche di gioco familiari (come può essere l'enigma), non dimenticando l'importanza della collaborazione tra i vari componenti del team, implementando una complementarietà

tra obiettivi e skills, facendo sì che la collaborazione, e dunque il lato sociale, fossero centrali nell'esperienza di gioco.

I bambini però, come descritto nel paragrafo 2.3.1, sono più inclini ad un'esperienza competitiva, e prediligono videogiochi dove le capacità fisiche come i riflessi e i movimenti complessi compongono le skills fondamentali. Per questa fascia di età si è dunque pensato di inserire un ruolo che si identificasse con il killer nella matrice di Bartle, e che agisse in un contesto d'azione.

Guardando la matrice di Bartle si può dunque profilare le tipologie di giocatori che più si avvicinano al target di questa esperienza.

I bambini, come detto in precedenza, si avvicinano a una dimensione simile a quelli dei giocatori killer, ovvero giocatori ai quali piace competere e che mirano alla vittoria di sfide competitive.

Gli anziani in media si avvicinano più alla sfera dei socializers, in quanto reputano importante la collaborazione con altre persone, che esse siano coetanee o no.

Essendo il target rappresentato da famiglie in vacanza, e quindi interessato alla visita di Torino e dei suoi luoghi culturali, si è pensato poi di considerare la porzione destra della matrice di Bartle nella profilazione. Ciò è stato fatto in quanto il contesto (mondo) in un videogioco in realtà aumentata coincide col mondo reale, e dunque l'interesse turistico per Torino si riflette in quello per il mondo ludico.



Figura 4.1: La matrice di Bartle

4.3 Game outline

MagicTurinAR è un location-based mobile serious game in AR, intergenerazionale e collaborativo, ambientato nei luoghi appartenenti alla Torino Magica. L'esperienza è pensata per famiglie in visita a Torino che cercano un modo alternativo e divertente per esplorarne il centro storico e i luoghi di interesse principali. Il team sarà composto da tre ruoli diversi, con abilità e mansioni differenti, pensati per differenti

fasce di età in base alle conclusioni tratte nel paragrafo 2.3.1 (che però non è strettamente necessario rispettare):

- **saggio:** è il membro della famiglia che dovrà far valere il proprio intelletto per risolvere gli enigmi e interpretare gli indizi al fine di consegnare all'esploratore le coordinate per trovare il luogo di accesso alla missione successiva. Inoltre dovrà studiare i frammenti di storia e combinarli opportunamente per aiutare i compagni a completare la missione finale. Il saggio ha il compito di gestire l'inventario e distribuire le risorse raccolte dai compagni oltre che a collezionare frammenti di magia bianca. Questo ruolo è pensato per giocatori appartenenti ad una fascia d'età più anziana, poiché comprende mansioni legate al ragionamento più che ad azioni fisiche complesse: le mansioni rientrano nella logica dei Puzzle Games genere particolarmente affine al target a cui si riferisce il ruolo.
- **esploratore:** è colui che dovrà condurre la famiglia nei luoghi individuati dal saggio. Ha il potere di vedere e raccogliere le Gemme Oscure nell'ambiente e di attivare il Navigatore nel caso perda l'orientamento. Questo ruolo è particolarmente indicato per una persona adulta poiché richiede di svolgere mansioni che combinano ragionamento e decision making, mansioni difficilmente adatte ad un bambino. Il tipo di gameplay rientra nel genere dei Puzzle Games e dei giochi d'avventura. Tra i pattern individuati da Lehman (par. 2.2.1), l'esploratore rientra indubbiamente nella logica Search and Find.
- **cacciatore:** il cacciatore dovrà interfacciarsi con le attività più fisiche come la distruzione di Scigni e la caccia ai Demoni. È l'unico in grado di raffinare le Gemme Oscure, distruggendole e ricavandone frammenti di Magia Bianca. Il ruolo del cacciatore è pensato per i più giovani, siccome prevede meccaniche che presuppongono reattività e mobilità fisica. Il genere inoltre rientra nei giochi d'azione, caratterizzati da sessioni di gioco corte e obiettivi chiari. Facendo sempre riferimento alla classificazione di Lehman, il cacciatore rientrerebbe nel pattern Chase and Catch.

I tre ruoli, tra loro complementari, costruiscono un gameplay intergenerazionale che, grazie alla sua asimmetria, permette ad ogni singola fascia di età di esprimere i propri gusti videoludici e di esaltarne le qualità al fine di trasmettere una sensazione di appagamento. Come descritto nella quinta delle nove regole per un game design intergenerazionale funzionale, è stato deciso di puntare sia su una dimensione di cooperazione nelle relazioni tra i giocatori, più affine alle fasce di età maggiore, sia su una dimensione di competitività nelle relazioni con gli NPC, più incline ai gusti dei giovani. Infine, si è deciso di non comprendere oggetti fisici all'interno del gameplay, sebbene nel paragrafo 2.3.1 si suggerisca tale pratica nel design di un gioco

intergenerazionale. La ragione di questa scelta risiede nel fatto che, trattandosi di un gioco pensato per turisti, la distribuzione di oggetti fisici rappresenterebbe una complicazione non necessaria.

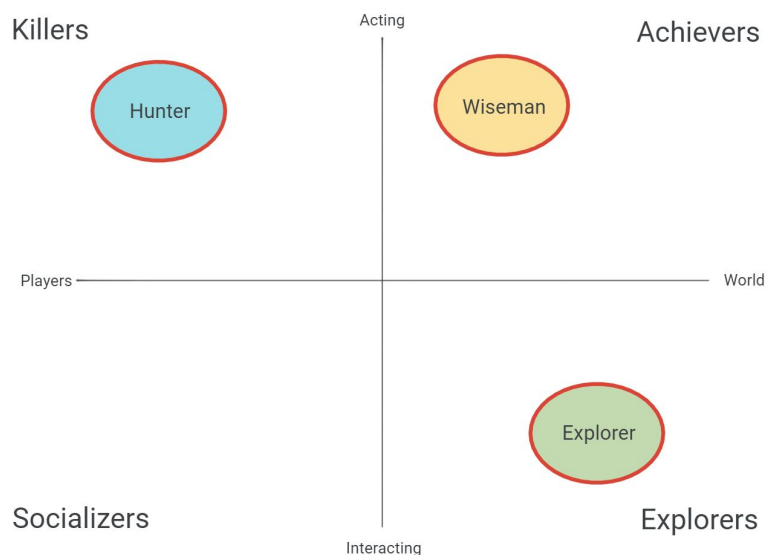


Figura 4.2: Collocazione dei tre ruoli all'interno della matrice di Bartle

Come si evince dalla matrice di Bartle in figura 4.2, i tre ruoli si collocano in una posizione ben precisa: un killer si identifica facilmente nell'hunter, il saggio è più adatto alla personalità di un achiever e l'esploratore, ovviamente, è più vicino all'explorer. Data la dimensione collaborativa dell'esperienza, la figura del socializer è in realtà presente in tutti i personaggi nel momento in cui devono interagire tra loro, sia all'interno dell'esperienza (ad esempio scambiandosi le risorse) che nel mondo reale, mentre esplorano la città insieme.

La collaborazione tra i diversi personaggi è incentivata da un controllo sulla loro posizione reciproca: se un membro è troppo distante dagli altri due, è possibile che le skill principali siano disabilitate. Ad esempio, se un cacciatore si reca in solitudine sul luogo individuato dall'esploratore non può vedere né attivare la missione. Gli scopi dell'esperienza sono molteplici:

- fornire un modo divertente e alternativo per esplorare il centro di Torino e imparare a conoscere i lati più esoterici della sua storia.
- coinvolgere diverse generazioni all'interno di un'esperienza videoludica condivisa, catalizzando l'interesse dei più piccoli (spesso annoiati durante le visite turistiche) ma senza trascurare il divertimento dei più anziani.

- incentivare la collaborazione tra i giocatori attraverso meccaniche di scambio e la complementarità delle skills, sia dei personaggi che dei giocatori che li controllano.
- generare divertimento tarando le mansioni da svolgere in base all'età e ai gusti dei giocatori (per esempio, un genitore sarà facilmente più interessato al ruolo del saggio più che a quello del cacciatore, più fisico e che meglio si avvicina agli interessi dei bambini).

4.3.1 Collaborazione tra team

C'è la possibilità di prendere parte a missioni secondarie, non necessarie al completamento della storia principale. Nello specifico, prendendo spunto da ciò che accade nei raid di PokémonGo (paragrafo 2.2.3), si tratta di demoni particolarmente potenti che hanno conquistato un luogo d'interesse in città e che non possono essere sconfitti da un solo cacciatore. Ognuna di queste missioni richiede infatti la partecipazione di un certo numero di team, direttamente proporzionale alla potenza del demone. I cacciatori devono lanciare attacchi coordinati per abbattere l'avversario, al fine di sbloccare tanti enigmi quanti sono saggi presenti sul posto. Una volta risolti tutti gli enigmi, i saggi devono capire come mettere in ordine le soluzioni e ottenere un insieme di coordinate da consegnare agli esploratori. Queste definiscono un'area che gli esploratori devono setacciare usando la realtà aumentata per trovare il bottino di Gemme che il demone ha nascosto (maggiore è il numero di famiglie e maggiore sarà il bottino relativo, poiché la sfida richiederà una collaborazione maggiore e quindi avrà un livello di difficoltà più elevato).

4.3.2 Suddivisione dei ruoli

In caso le famiglie fossero composte da $n > 3$ membri, si è pensato di distribuire i ruoli secondo questo schema:

- se $n \% 3 = 0$ allora i ruoli saranno distribuiti equamente (ad es. se $n = 6$ ci saranno due esploratori, due saggi e due cacciatori)
- se $n \% 3 = 1$ allora ci potrà essere un cacciatore o un saggio aggiuntivo, in base alle caratteristiche del giocatore in surplus. Nel ruolo che coinvolge il giocatore aggiuntivo, le prove saranno caratterizzate da un livello di difficoltà maggiore e richiederanno la collaborazione tra i membri che ricoprono lo stesso ruolo.
- se $n \% 3 = 2$ allora ci saranno sia un cacciatore che un saggio aggiuntivo e le missioni saranno gestite come nel caso 2.

Se le famiglie sono numerose a tal punto da consentire la generazione di almeno due team indipendenti, il gioco sarà in grado di parallelizzare l'esperienza, gestendo i gruppi come entità distinte, aumentando il numero di luoghi visitabili e dunque dei relativi task. L'obiettivo finale richiede comunque la partecipazione di tutti i membri della famiglia, e dunque la vicinanza di tutti i membri.

4.3.3 Assegnazione dei ruoli

Le tre modalità che sono state pensate per l'assegnazione dei ruoli sono le seguenti:

- Questionario che serve a delineare i gusti, come giocatore, di ciascuna persona e che permette al sistema di comprendere e assegnare il ruolo più adatto
 - Pro: Ruoli adattati ai gusti del giocatore.
 - Contro: Necessità di avere i tre ruoli distinti per completare l'esperienza. Nel caso due persone rispondano nello stesso modo, il gioco dovrebbe imporre i ruoli e questo potrebbe generare malcontento.
- Scelta dei ruoli lasciata ai giocatori a seguito della presentazione delle caratteristiche di ciascun ruolo e delle mansioni che si devono svolgere.
 - Pro: La scelta è lasciata ai giocatori in base ai loro gusti
 - Pro: I possibili conflitti vengono risolti dai giocatori e non dagli sviluppatori, evitando imposizioni da parte del gioco.
 - Contro: viene imposta l'eterogeneità dei ruoli e se due persone desiderano ricoprire lo stesso ruolo devono scendere a compromessi generando possibili attriti.
- Scelta imposta dal gioco in base all'età
 - Pro: Ruoli dei giocatori disegnati per i gusti medi dei giocatori, quindi adatti al caso generico ma non necessariamente a quello specifico.
 - Pro: Non viene lasciata scelta ai giocatori e dunque si evitano attriti iniziali interni al team.
 - Contro: non si tiene conto di casi particolari (ad esempio un bambino che vuole fare l'esploratore) e questo potrebbe generare malcontento nei confronti dell'esperienza.
 - Contro: in caso di famiglie caratterizzate da una composizione disomogenea (ad esempio un genitore e due bambini), difficoltà ad assegnare ruoli.

Si è deciso di optare per la seconda modalità di scelta per dare una sensazione di customizzazione ulteriore dell'esperienza: i giocatori non si dovrebbero sentire vincolati a scelte imposte dal gioco e in caso di conflitto d'interesse per il singolo ruolo si è ritenuta migliore la possibilità di discussione tra i giocatori anziché una scelta randomica fatta dall'applicazione.

4.4 Core game context

4.4.1 Storyline

Dopo mille anni di pace garantita dalla supremazia delle forze della Magia Bianca, l'allineamento perfetto dei pianeti del Sistema Solare rappresenta l'occasione per alcuni esponenti dell'Ordine Oscuro di riunirsi per rievocare l'Oscuro Sovrano, che fino a mille anni prima dominava la città, seminando caos e disgrazia attraverso il suo esercito di Demoni e seguaci. Per salvare la situazione, l'Ordine Bianco invia in città i suoi adepti: il loro compito sarà quello di esplorare la città per trovare e distruggere il Grande Buio, ovvero il luogo dove le Forze Oscure si riuniranno per evocare l'Oscuro Sovrano. Affinché questo sia possibile, i seguaci dell'Ordine Bianco dovranno trovare manufatti, risolvere enigmi e completare missioni per ottenere indizi relativi alla posizione del Grande Buio e impedire che il rituale venga compiuto.

4.4.2 Setting

Il gioco si concentra all'interno del centro storico di Torino e principalmente nei 9 luoghi carichi di Magia Bianca o Nera:

- Magia Bianca
 - Gran Madre
 - Mole Antonelliana
 - Fontana delle Nereidi e dei Tritoni
 - Fontana Angelica
 - Museo Egizio
- Magia Nera
 - Occhi del Diavolo
 - Portone del Diavolo
 - Rondò della Forca

- Obelisco di Piazza Statuto
- Museo Egizio

Come si evince dall'elenco, il Museo Egizio compare due volte perché è considerato il cuore dell'esoterismo torinese e sembrerebbe contenere numerosi artefatti oscuri, ma anche molti reperti associabili alla Magia Bianca.

4.4.3 Personaggi

- Magia Bianca
 - Gran Maestro dell'Ordine Bianco: il coordinatore della missione. Invierà i suoi adepti per salvare Torino.
 - Esploratore (adepto): ha strumenti per orientarsi all'interno della mappa. È in grado di utilizzare un Navigatore in caso perda l'orientamento, ma ciò gli sottrae Energia Bianca. Può raccogliere gli Antichi Manufatti e le Gemme Oscure, ma non è in grado di decifrare gli enigmi.
 - Cacciatore (adepto): esperto di magia distruttiva, può forzare gli Scrigni per trovare chiavi e indizi e combattere i Demoni. È in grado di raffinare le Gemme per ricavare frammenti di Magia Bianca.
 - Saggio (adepto): conoscitore esperto della Magia in tutte le sue forme, è in grado di decifrare gli enigmi e quindi di fornire all'Esploratore le informazioni di cui ha bisogno. Possiede e gestisce l'inventario e decide come distribuire i frammenti di Magia Bianca tra i membri.
- Magia Nera
 - Oscuro Sovrano: il Signore della Magia Oscura, esiste nel limbo tra vita e morte. Da mille anni brama di tornare al potere e ha ancora molti sostenitori che aspettano il suo ritorno.
 - Demone (NPC): figura oscura, serva dell'Oscuro Sovrano.
 - Seguaci dell'Ordine Oscuro (NPCs): sostenitori che cercano di rievocare il loro padrone. Hanno il compito di posizionare gli Antichi Manufatti in luoghi specifici per consentire la riuscita del rituale di evocazione.
 - Traghettoniere (NPC): personaggio da raggiungere per poter attraversare il Fiume Po e raggiungere la zona della Gran Madre. Profondamente avido di Gemme Oscure, le richiede per svolgere il suo compito.

4.4.4 Obiettivi

- Obiettivi Principali Maggiori
 - capire dove si trova il Grande Buio
 - evitare che il rituale venga compiuto
- Obiettivi Principali Minori
 - trovare la posizione di determinati punti d'interesse
 - ottenere indizi uccidendo demoni e risolvendo enigmi
 - trovare e gestire le risorse
- Obiettivi Secondari
 - raccogliere i manufatti collezionabili
 - ottenere magia bianca rispondendo ad altri quiz
 - liberare la città dai potenti demoni che la infestano insieme ad altre famiglie
 - trovare la posizione dei tesori dei demoni insieme ad altre famiglie tramite la risoluzione di enigmi e l'esplorazione.

4.4.5 Risorse e collezionabili

- Antichi Manufatti: oggetti di finissima fattura realizzati da esperti artigiani. Possono essere trovati e raccolti solo dall'esploratore.
- Vecchi Manoscritti: volumi carichi di antica conoscenza, che possono essere trovati solo dal saggio.
- Rune: simboli di un alfabeto dimenticato. Ogni runa è incisa su un frammento di materiale differente, selezionato dalla lista dei materiali presenti nelle costruzioni di Torino [40]. Possono essere individuate e raccolte solo dal cacciatore.
- Gemme Oscure: pietre di inestimabile valore, individuabili solo dall'esploratore, cariche di magia nera. Possono essere distrutte per ricavare frammenti di Magia Bianca e rappresentano la moneta di scambio per accedere ad alcune aree o servizi. I luoghi di accesso alle missioni non possono essere sbloccati in presenza di una Gemma nelle vicinanze.
- Scrigni: contengono indizi, enigmi o chiavi necessari per scoprire la posizione del Grande Buio.

- Frammenti di Magia Bianca: necessari per incrementare l'Energia Bianca del Cacciatore e dell'Esploratore. Possono essere ottenuti distruggendo le Gemme Oscure o raccolte dal Saggio. Se il Cacciatore non ha Energia sufficiente, non è in grado di distruggere gli Scrigni o combattere i Demoni.

4.5 Core gameplay

Come specificato in precedenza, ognuno dei tre ruoli presenti all'interno dell'esperienza ha le proprie abilità ed è necessario affinché il percorso possa essere portato a compimento. In questa sezione verranno dettagliate le modalità di interazione tra i giocatori stessi, con il mondo (incluse altre famiglie, nemici e NPCs) e con i collezionabili.

4.5.1 Meccaniche principali

Il gameplay comporta diverse meccaniche di gioco per ciascun ruolo scelto dal giocatore.

L'esploratore deve cercare un portale di accesso per dare via alle missioni degli altri due componenti del team; dovrà trovare questo portale che sarà visibile soltanto se, esplorando una zona visibile sulla mappa della città, il giocatore si avvicinerà fisicamente alla sua geolocalizzazione. In caso l'esploratore si senta disorientato, oppure voglia arrivare celermente al portale nascosto, potrà attivare il potere di Navigazione consumando energia bianca e visualizzando il percorso più rapido per l'obiettivo sulla mappa. Nell'ambito delle missioni ordinarie, l'esploratore riceve le coordinate dal saggio e ha il compito di guidare il gruppo verso un punto preciso sulla mappa (secondo la logica Search-and-find [17]). In questo caso, il percorso seguito dal team non ha alcuna importanza. Nell'ambito della missione finale, invece, quando il saggio mette insieme i pezzi di storia di fatto costruisce una mappa finale, che indica non solo l'ultima location del gioco, ma anche il percorso da seguire per raggiungerla. Questa meccanica fa riferimento invece al pattern Follow-the-path [17].

Una volta che l'esploratore riesce a trovare il luogo d'interesse, inizia la missione del cacciatore. Questo ruolo è il più dinamico dei tre e il suo obiettivo è quello di combattere dei demoni che cercheranno di colpirlo e dunque fuggire. Compito del cacciatore sarà quello di distruggere i demoni lanciando degli incantesimi, prima che questi riescano a esaurire la vita del giocatore. Gli incantesimi consumano energia bianca e si dividono tra l'incantesimo comune, che consuma poca energia e infligge danni lievi, e quelli potenziati che consumano più energia, ma risultano 5 volte più potenti di un incantesimo base. La ricompensa sarà un enigma che dovrà essere passato al saggio. Nel caso in cui il cacciatore non riuscisse a sconfiggere il demon, dovrà ripetere la missione. Come accennato nell'Outline, il ruolo del

cacciatore segue la logica Chase-and-catch [17].

Il ruolo del saggio prende le redini del gameplay nel momento in cui gli vengono consegnati gli enigmi dal cacciatore. Questo ruolo si rifà al genere dei Puzzle Games e richiede un ingente sforzo mentale per riuscire a comprenderne la soluzione. Gli enigmi consistono in testi con dei messaggi in codice da individuare, risoluzione di labirinti, decifrazioni di testi in codice, memory e puzzles. La ricompensa sono le coordinate della prossima missione da consegnare all'esploratore. Compito del saggio però, oltre a ricavare informazioni preziose grazie al suo intuito, è anche quello di distribuire in modo coscienzioso ed equilibrato la magia bianca di cui il team è riuscito ad entrare in possesso tra le altre due tipologie di compagni, gestendo la parte dell'inventario dedicata a gemme oscure e frammenti di Magia Bianca.

4.5.2 Inventario e collezionabili

Gli oggetti collezionabili sono visibili ai giocatori in base al ruolo ricoperto, ma la meccanica di raccolta è la medesima. In particolare, dopo averlo visualizzato sulla mappa, il giocatore dovrà avvicinarsi fisicamente all'oggetto collezionabile e, non appena rientrerà in un raggio di azione di circa una trentina di metri, potrà prenderlo cliccandoci sopra. Nel caso in cui si tratti di Frammenti di Magia Bianca o di Gemme, visualizzabili solo dall'esploratore e dal saggio, contestualmente alla raccolta si verificherà l'invio del collezionabile all'inventario del saggio.

Ogni giocatore ha il proprio inventario, contenente oggetti collezionabili. Il saggio gestisce le gemme, i frammenti di storia e i Vecchi Manufatti. Il cacciatore conserva le Rune e l'esploratore possiede la mappa e gli Antichi Manufatti. Ogni personaggio è in grado di visualizzare nel dettaglio ciascun elemento, manipolandolo in una vista 3D in AR. Per quanto riguarda la meccanica dello scambio di gemme, il saggio ha la possibilità di consegnarle al cacciatore, lasciandole nel punto in cui si trova e consentendogli di vederle e distruggerle in AR, con l'obiettivo di produrre Frammenti di Magia Bianca, automaticamente consegnati al saggio.

Il saggio può poi inviare frammenti di energia bianca, cliccando sugli avatar degli altri due giocatori. Un frammento di magia bianca comporterà l'aumento di un quarto di energia bianca sul totale di quella immagazzinabile da un giocatore.

Il valore delle Gemme non riguarda solo l'energia: esse rappresentano infatti anche una moneta di scambio. È possibile infatti che sia richiesto un pedaggio per completare un task: per sbloccare l'area oltre il fiume Po, ad esempio, è necessario consegnare alcune gemme al Traghettoniere.

4.5.3 Sfida finale

L'ultima sfida del gioco consiste nell'individuare il Grande Buio e impedire la resurrezione dell'Oscuro Sovrano. La missione è composta da tre diversi task, che seguiranno il game loop canonico, ma richiederanno sforzi maggiori da parte dei giocatori.

- Saggio: deve mettere insieme i Frammenti di Storia che ha raccolto durante il viaggio. Il testo contiene un enigma (alcune lettere evidenziate che con la giusta combinazione compongono una frase), la cui soluzione invita a capovolgere il foglio manipolandolo attraverso l'inventario. Il retro del foglio conterrà una percorso da seguire per raggiungere il luogo della sfida finale che verrà inviato all'esploratore.
- L'esploratore deve guidare il team attraverso tutti i checkpoint del percorso svelatogli dal saggio. A differenza di una canonica missione, non è lasciata libertà ai giocatori su come raggiungere la posizione del Grande Buio, ma sarà necessario passare per dei punti specifici seguendo la logica follow the path [17]. Il luogo finale è costituito da un'area contenuta e pedonale.
- Nell'area finale si trovano alcuni generatori che devono essere scovati dal cacciatore e attivati tramite una nuova meccanica. Il cacciatore deve toccare una sequenza di bottoni seguendo il ritmo che gli viene imposto (Guitar Hero, Piano Tiles) e, una volta attivati i generatori, la Magia Bianca sarà tale da non permettere il completamento del rituale.

4.5.4 Sfide collaborative

Occasionalmente, sulla mappa dell'esploratore appaiono degli indicatori che segnalano la presenza di una sfida collaborativa. Come spiegato nel paragrafo 4.3.1 (Collaborazione tra team) ognuna di queste sfide richiede un certo numero di famiglie per essere portata a termine.

La missione consiste nell'abbattere un certo numero di demoni entro un limite di tempo. I cacciatori devono quindi muoversi all'interno dell'area identificata e sconfiggere tutti i demoni. Una volta completato il lavoro dei cacciatori, ciascun saggio deve risolvere il proprio enigma, ma è chiaro che esiste la possibilità di aiutarsi a vicenda, trattandosi di un'esperienza co-localizzata. Ogni enigma, una volta risolto, genera un frammento di mappa: gli esploratori hanno il compito di lavorare insieme per unire correttamente i vari pezzi e generare una mappa unica, che segna un'area da raggiungere per arrivare al bottino del demone. La logica applicata è quella del pattern Search-and-find [17].

4.6 Game loop

Durante le missioni individuali, i giocatori non direttamente coinvolti hanno la possibilità di leggere curiosità, guardare video e poi rispondere a un quiz a scelta multipla inerente l'argomento precedentemente illustrato. In caso di risposta corretta, il team guadagna una piccola quantità di Frammenti di Magia Bianca. Questa eventualità si verifica solo durante la risoluzione degli enigmi da parte del saggio e durante le mansioni del cacciatore, poiché, durante l'esplorazione, si vuole incentivare tutti i membri del team a partecipare attivamente all'esplorazione stessa e alla visita della città.

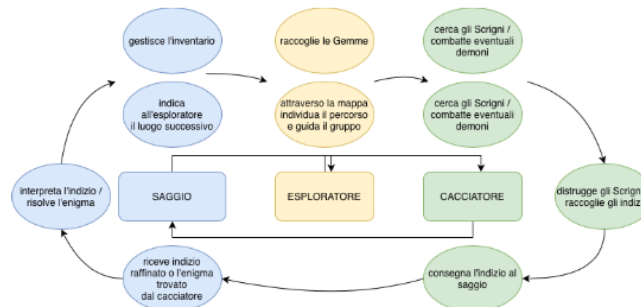


Figura 4.3: Il game loop dell'esperienza

4.7 Gameplay puzzles

RUOLO	SAGGIO		ESPLORATORE		CACCIATORE	
DIFFICOLTÀ	SEMPLICE	COMPLESSO	SEMPLICE	COMPLESSO	SEMPLICE	COMPLESSO
ATTIVITÀ	Interpretare indizi e risolvere enigmi. Gestisce l'inventario.	Studiare i frammenti di storia, utili per l'enigma finale	Navigare la mappa, raccogliere le Gemme e i luoghi di accesso	Ricavare le coordinate per raggiungere l'enigma finale	Distruggere gli scigni e uccidere i demoni	Attivare i generatori
PROGRESSI	Trova le coordinate per la missione successiva	Comprende le mosse da fare per evitare l'evocazione dell'Oscuro Sovrano	Raggiungere il luogo di accesso alla missione	Trovare il luogo di accesso alla missione finale	Ottenere l'indizio	Evitare l'evocazione dell'Oscuro Sovrano
ATTIVATORI	Indizi ricevuti dal cacciatore	Frammenti di storia (Realtà aumentata per unire i pezzi del foglio)	Mappa (eventualmente navigatore) + AR per individuare oggetti	Mappa (no navigatore) e AR per individuare il GB	AR per vedere e distruggere scigni e demoni	AR
OSTACOLI	Complessità dell'enigma	Memorizzare le informazioni	Orientamento	Decrittazione delle coordinate finali	Dosaggio dell'energia, tempo limite	Velocità dei tiles da toccare

Figura 4.4: Obiettivi primari e secondari di ciascun ruolo

4.8 L'esperienza di gioco prevista

I giocatori, tramite il gameplay, avranno la possibilità di esplorare la città di Torino e approfondire gli aspetti legati alla magia bianca e nera. Il Saggio eserciterà la propria memoria e la propria logica, attraverso la risoluzione degli enigmi e lo studio dei frammenti di storia. L'esploratore migliorerà le proprie capacità di orientamento e imparerà a conoscere meglio le strade e i luoghi di interesse di Torino. Il Cacciatore, invece, avrà compiti più fisici, che richiederanno lo svolgimento di task entro limiti di tempo specifici o movimenti complessi. Globalmente, i giocatori avranno la possibilità di vivere un'esperienza ad ampio respiro, articolata su diversi livelli e che risponde ad esigenze di tipo diverso:

- in primo luogo, ovviamente, la dimensione ludica dell'esperienza garantirà momenti di svago, spensieratezza e divertimento, utili soprattutto per quanto riguarda il coinvolgimento dei più piccoli;
- secondariamente, la natura serious rende il percorso accattivante per i giocatori adulti, il cui obiettivo principale resta quello di scoprire i misteri di Torino e la storia della città;
- gli elementi di collaborazione hanno lo scopo di incentivare la comunicazione tra i membri di una famiglia, che spesso risulta poco spontanea a causa degli ovvi salti generazionali.

4.9 Le ragioni dell'engagement

- Gli obiettivi di ciascun giocatore sono vincolati al raggiungimento degli obiettivi dei compagni: ciò favorisce la collaborazione tra i membri della famiglia e l'aiuto reciproco.
- La gestione delle risorse è fondamentale per svolgere determinati task e la possibilità di scambiare oggetti con i compagni aumenta le possibilità di collaborazione e interazione.
- La sfida finale è un ulteriore elemento di coinvolgimento dei giocatori in un'esperienza fondata primariamente sulla collaborazione e la condivisione.
- Ipoteticamente, il gioco potrebbe essere utilizzato dalle attività locali come forma di investimento, per esempio chiedendo di posizionare task o Gemme all'indirizzo in cui si trovano (un ristorante).
- L'esplorazione della città come core mechanic incentiva la curiosità e l'approfondimento della storia di Torino, primariamente quella legata alla magia.

- La collaborazione tra diversi team in occasione delle missioni secondarie introduce un ulteriore livello di engagement e di rigiocabilità.
- La fase di attribuzione dei ruoli consente di cucire l'esperienza attorno ai gusti dei giocatori, assegnando task che più si confanno alla loro attitudine.

4.10 Reference grafiche

Si è pensato allo stile grafico come ad un mix tra fotorealismo, per quanto riguarda i collezionabili da visualizzare in AR, e una grafica cartoon per quanto riguarda invece l'interfaccia della mappa e i personaggi. In questo caso, lo stile cartoon è stato adottato poiché per gli anziani, la fetta del target più complessa da raggiungere, in termini di estetica, i giochi preferiti sono quelli con questo tipo di grafica rispetto a quelli fotorealistici [41]. In questa sezione sono raccolte alcune reference e schizzi di concept art di videogiochi celebri e film che hanno ispirato lo stile grafico del gioco.

4.10.1 Medieval Resurrection (2005)

Remake del celebre gioco Medieval, uscito nel 1998 per PlayStation.

4.10.2 Nightmare before Christmas

Film d'animazione del 1993 diretto da Henry Selick, ideato e co-prodotto da Tim Burton.



Figura 4.5: Medievil Resurrection



Figura 4.6: Nightmare before Christmas

Capitolo 5

Sviluppo e design

In questo capitolo verranno analizzate le caratteristiche e le funzionalità del sistema, con particolare attenzione alle strategie adottate per la realizzazione finale dell'applicativo. L'approccio adottato per la descrizione dello sviluppo segue l'andamento delle scene assemblate su Unity: dopo aver analizzato la fase di login e di connessione al server, verranno trattati la creazione della lobby e i meccanismi di partecipazione dei giocatori; si passerà in seguito ad esaminare la scena Game Main, ovvero l'ambiente in cui si svolge la maggior parte dell'esperienza, ed infine si analizzeranno le scene in realtà aumentata.

5.1 Autenticazione

La prima scena dell'applicativo è la scena di log in, nella quale si è deciso di permettere all'utente di poter creare un account con cui accedere all'esperienza, in modo tale da consentire il successivo salvataggio dei dati della partita. È stato deciso di utilizzare il sdk di Firebase, in particolare i pacchetti riguardanti l'autenticazione e l'utilizzo del Realtime Database fornito dal toolkit.

I componenti principali in questa scena sono il Firebase Initializer (in scena `FirebaseInit`) che permette l'inizializzazione delle funzionalità Firebase, il `VibrationInitializer` che permette l'inizializzazione delle funzionalità che permettono di produrre feedback aptici (vibrazioni) su dispositivo mobile, e infine il `Firebase Manager`, ovvero l'oggetto responsabile della gestione dell'interfaccia della scena e della comunicazione tra applicativo e piattaforma Firebase.

Il `Firebase Manager`, nello specifico, possiede coroutine come:

- `private IEnumerator Register(string __email, string __password, string __username)`
- `private IEnumerator Login(string __email, string __password)`

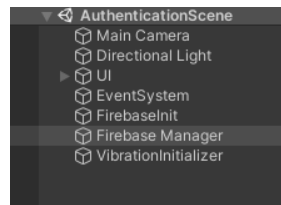


Figura 5.1: La hierarchy dell'Authentication Scene

responsabili della registrazione e del login. Gli attributi differiscono poiché la stringa definita come username una volta che l'account è registrato sarà legata a quel singolo account e sarà accessibile una volta effettuato il login. Ovviamente il sistema di login e di registrazione è gestito lato utente tramite un'interfaccia che permette l'inserimento di mail, password e username.

Nel Firebase Manager, la cui istanza non verrà distrutta al caricamento delle scene successive, sono presenti anche i metodi relativi al salvataggio dei dati relativi alla singola partita e di aggiornamento dello username.

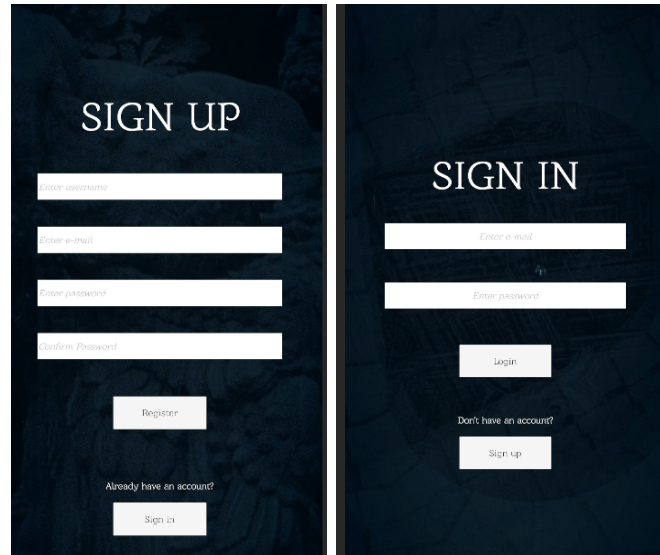


Figura 5.2: Le schermate di sign in e sign up

5.2 Connessione al server e gestione della lobby

5.2.1 Offline

La scena *Offline* è una scena molto semplice ma che presenta al suo interno il Network Manager, GameObject responsabile del funzionamento dell'intera infrastruttura di Mirror e dunque della dimensione collaborativa dell'applicativo. Questo dispone di due campi, controllabili tramite l'Inspector, dedicati alle scene che devono essere caricate quando il client è connesso al server oppure non riesce a stabilire la connessione. Un GameObject altrettanto utile è l'AutoConnect, un oggetto che permette la connessione automatica del client al server, nel caso il server sia acceso e accessibile, in modo da snellire la procedura di connessione lato client.

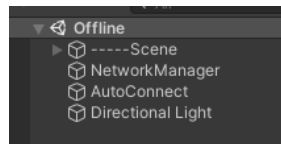


Figura 5.3: La hierarchy dell'Offline Scene

5.2.2 Lobby

Una volta connesso al server, il client carica la scena Lobby, in cui il server istanzia i Prefab rappresentativi dei singoli giocatori. In questa scena sono presenti tanti player prefab quanti sono i client connessi, e ciascun prefab è identificato sul network grazie ad una network identity che permette l'instradamento dei pacchetti. La hierarchy della lobby presenta al suo interno diversi GameObjects di grande importanza per il funzionamento dell'applicativo.

MatchMaker

Il MatchMaker è un GameObject responsabile della generazione di lobby identificate da un GUID (Globally Unique Identifier) e della loro relativa gestione: sono presenti infatti funzioni che permettono l'ingresso in una determinata stanza, di hostare una lobby e ovviamente di dare il via all'esperienza per tutti i partecipanti. Il MatchMaker conserva al suo interno due liste di oggetti di tipo Match, classe che contiene al suo interno i giocatori e l'ID della stanza, e di ID dei match, in modo tale da poter gestire un semplice sistema multilobby. Lo script permette inoltre lo spawn del TurnManager, un prefab che mantiene il riferimento della lista di giocatori della partita durante l'esperienza di gioco.

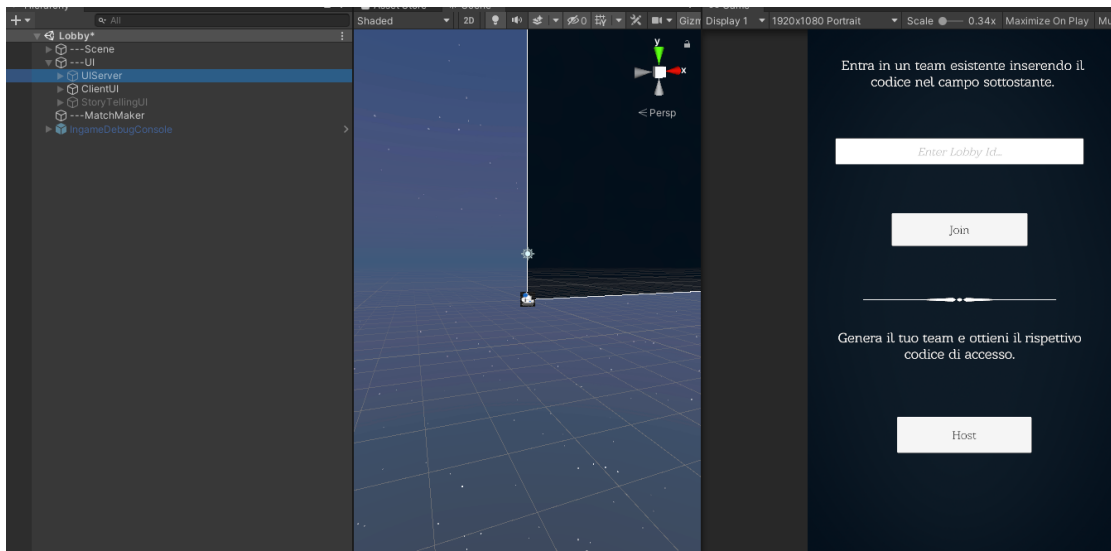


Figura 5.4: La hierarchy della Lobby

LobbyNetworkPlayer

Nella lobby sono presenti inoltre i prefab rappresentativi dei singoli giocatori, i quali possiedono lo script `LobbyNetworkPlayer`. Questo script è responsabile delle command incaricate della comunicazione al server per quanto riguarda le operazioni di join e host. La comunicazione tra il `MatchMaker` e i `LobbyNetworkPlayer` permette dunque l'intero sistema multilobby e la sua coerenza sul server.

ItemAssets

All'interno della scena Lobby è presente inoltre il prefab *ItemAssets*, che colleziona, all'interno di uno script omonimo a lui legato, diversi prefab e Scriptable Objects tramite i quali è possibile comporre l'esperienza videoludica. Tra gli altri, il `MagicInventorySO` è uno dei principali: si tratta di uno Scriptable Object che contiene al suo interno una lista di Scriptable Objects rappresentativi di tutti i collezionabili dell'esperienza. All'interno dell'*ItemAssets* sono definiti anche i tre prefab relativi ai ruoli interpretabili durante la sessione di gioco. Durante il gameplay, sono molti gli script che accedono all'*ItemAssets* per ottenere il riferimento di cui necessitano.

InGameDebugConsole

Un elemento utile durante la fase di debug è stata sicuramente la *InGameDebugConsole* (nell'immagine del setup è disabilitato), che ha permesso di visualizzare

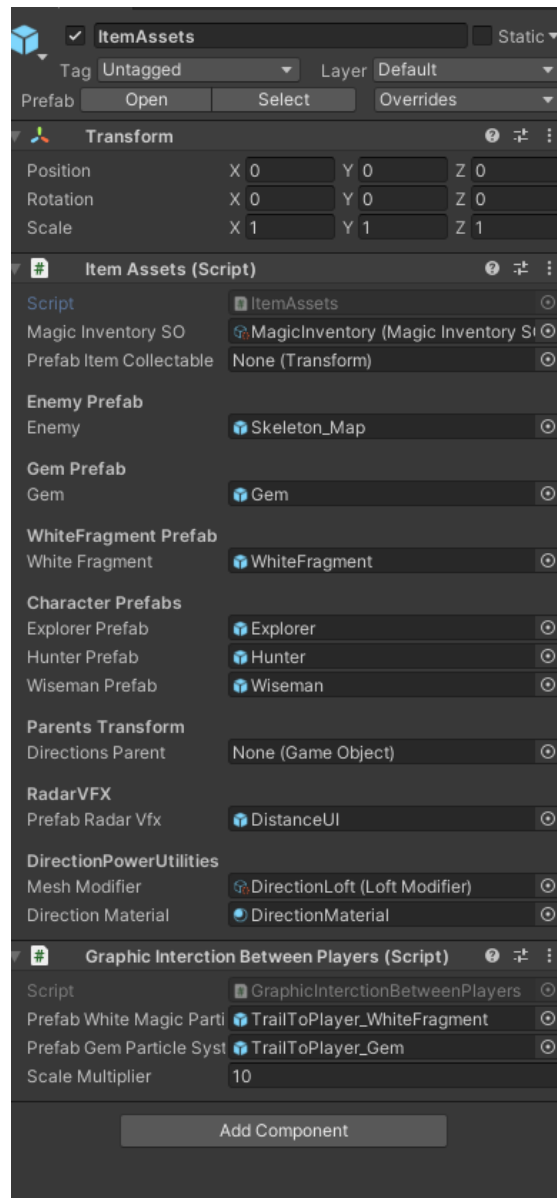


Figura 5.5: La vista di ItemAssets dall'Inspector

stampe di log, warning ed errore durante la simulazione su dispositivo mobile. Questa feature si è rivelata cruciale per comprendere al meglio determinati bug, soprattutto quelli impossibili da riprodurre in Play Mode, e correggerli.

Interfaccia

L'interfaccia si compone di tre diverse sezioni:

- **UIServer**: l'interfaccia che viene visualizzata quando un dispositivo opera solamente da server. Essa è molto semplice e illustra solamente il numero di client connessi.
- **ClientUI**: l'interfaccia con cui interagisce il giocatore. Presenta la possibilità di hostare una partita, oppure di entrare in una stanza tramite un codice alfanumerico univoco. Una volta all'interno della lobby, ogni giocatore è identificato da un riquadro che presenta lo username e il ruolo scelto, con l'immagine relativa all'avatar corrispondente. In questa schermata è inoltre possibile scegliere la tipologia di personaggio andando a leggere una breve descrizione delle mansioni e delle abilità.
- **StoryTellingUI**: Interfaccia di gioco che appare una volta che l'host inizia la partita. Ha il compito di introdurre i giocatori nel contesto immaginario in cui è ambientata l'esperienza e consiste in una serie di messaggi, messi in successione attraverso uno `ScriptableObject`.

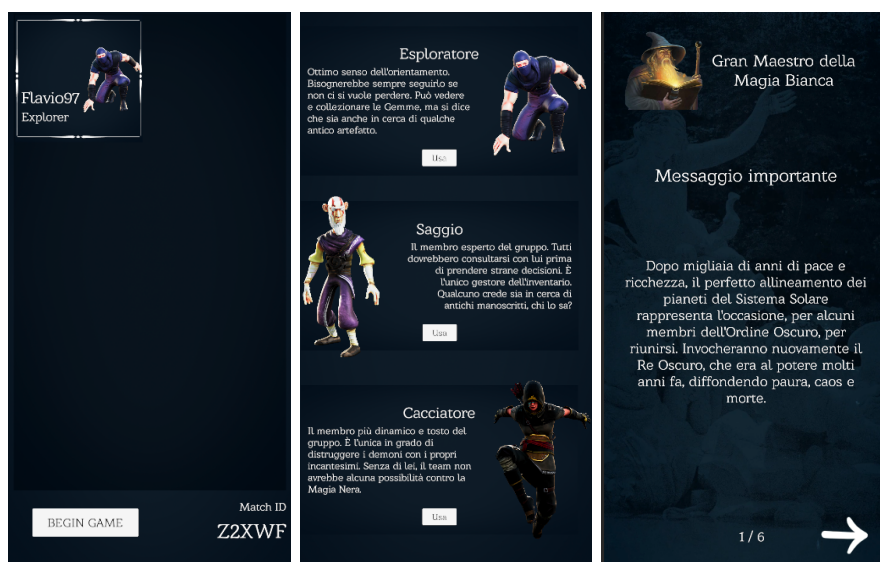


Figura 5.6: UIServer, ClientUI e StorytellingUI

5.3 L'interfaccia principale

La scena principale di tutta l'esperienza è *Game Main*. Da qui è possibile accedere alle missioni, all'inventario ed è visibile la mappa con i relativi dati di geolocalizzazione. Le missioni, eccezion fatta per quelle dell'Esploratore, che richiedono la

mappa per potersi orientare, consistono in scene che vengono caricate additivamente al momento dell'attivazione delle stesse, ma fin quando l'esperienza ludica non giunge all'ultima scena riguardante le statistiche, il Game Main rimane sempre attivo, per garantire la persistenza dei dati. Sulla mappa vengono localizzati gli avatar dei giocatori, le cui posizioni sono aggiornate al variare delle coordinate GPS del dispositivo: è possibile, inoltre, raccogliere i collezionabili e i consumabili presenti sulla mappa, ma gli utenti devono essere necessariamente vicini, sia tra di loro, sia all'oggetto da raccogliere. L'interfaccia per ciascun ruolo presenta dei punti in comune, come gli indicatori della qualità del segnale del GPS e della vicinanza dei membri del gruppo, ma si differenzia per altre caratteristiche, come ad esempio l'inventario, a causa del design asimmetrico dei tre diversi ruoli.

Di seguito sono illustrati i GameObject, gli script e i componenti più importanti della scena, per comprendere il funzionamento della stessa e il metodo con cui sono state affrontate le criticità in fase di sviluppo.

5.3.1 GUI e tutorial

Per quanto riguarda il design della GUI all'interno di questa scena, si è cercato di inserire il minor numero possibile di elementi:

Icone e slider

- L'icona in alto a sinistra indica la vicinanza tra i giocatori: se è verde significa che essi sono ad una distanza reciproca adeguata e che quindi possono procedere con l'esperienza. Nel caso in cui anche solo uno di loro fosse troppo lontano, l'icona diventerebbe rossa, a segnalare che è impossibile avanzare nel gioco: in questa situazione, i giocatori non possono raccogliere i consumabili (Gemme e Frammenti di Magia Bianca), né attivare le rispettive missioni.
- In alto a destra, invece, è presente un indicatore molto semplice da interpretare, che indica la qualità del segnale GPS, ottenuta con una Coroutine ad intervalli regolari.
- Sempre in alto a destra è possibile notare due slider, che riportano rispettivamente il Mana e la Salute disponibili. La Salute viene visualizzata solamente dal Cacciatore, in quanto è l'unico che può perderla durante il combattimento con i Demoni, mentre il Saggio non visualizza nessuno di questi due indicatori.

Bottoni

- Sulla sinistra, al centro dello schermo, sono stati posizionati due bottoni necessari per i controlli touch della mappa: il primo, visualizzabile solo

dall'Esploratore, consente di vincolare o svincolare la camera dal player prefab, mentre il secondo attiva o disattiva la rotazione basata sul GPS.

- Nella parte bassa dello schermo, invece, compare un bottone, comune a tutti, necessario per accedere allo zaino e visualizzare i collezionabili raccolti.
- Solo per l'Esploratore è stato predisposto un secondo bottone, necessario per attivare e disattivare il Navigatore.

Appena viene lanciata l'esperienza, si è deciso di mostrare un breve tutorial per illustrare il significato e le funzionalità dei vari componenti dell'interfaccia. Essendo le interfacce dei tre giocatori leggermente differenti tra di loro, è stato necessario utilizzare tre `GameObject` indipendenti, che comunque implementano lo stesso script, *Tutorial*. Il funzionamento di questo componente è molto semplice: tramite Inspector è possibile passare una lista di `GameObject`, i quali vengono letti in ordine e mostrati uno per volta. Ognuno di questi `GameObject` è costituito da una finestra contenente un campo di testo e da un bottone che consente di procedere nel tutorial.

5.3.2 Network Player

NetworkPlayer è una classe figlia di *LobbyNetworkPlayer*, vista sopra, in cui sono dichiarate le funzioni responsabili di alcune delle funzionalità di gioco, come lo scambio di risorse tra giocatori oppure l'avanzamento delle missioni. Essa è attribuita al player prefab spawnato dal Network Manager e permette di mantenere la coerenza nell'avanzare dell'esperienza comunicando col server tramite Command, Target RPC e Client RPC, particolari tipologie di funzioni messe a disposizione da Mirror.

5.4 Geolocalizzazione

Dal punto di vista della gestione delle mappe e dei dati GPS dei dispositivi, il sdk di Mapbox per Unity si è rivelato una carta fondamentale. Di seguito, sono brevemente descritti il setup iniziale e i componenti principali.

5.4.1 Setup

Una volta installato il package, si apre la finestra riportata nell'immagine: qui è possibile inserire il token di accesso ottenuto in fase di creazione dell'account su Mapbox.com e necessario per attivare alcuni servizi come geolocalizzazione e navigazione. Inoltre è possibile aprire alcune scene di default predisposte dal team di Mapbox.



Figura 5.7: La finestra di setup di Mapbox

5.4.2 Location Based Game

Un prefab di fondamentale importanza nello sviluppo del progetto è Location-BasedGame. Questo è infatti un prefab predisposto dal sdk che contiene due GameObjects importantissimi: il primo è l'AbstractMap, mentre il secondo è il LocationProvider. Questi due componenti sono strettamente interconnessi e hanno a che fare rispettivamente con la gestione dei settaggi della mappa e con il fetching della posizione del dispositivo mobile tramite il GPS.

Abstract Map

Si tratta dello script più complesso e ricco dell'intera suite. Implementa l'interfaccia IMap e contiene la logica per configurare la mappa e gestire la visualizzazione dei relativi dati. Lo script infatti incapsula il controllo dello stile grafico, del terreno e delle features che la mappa deve mostrare. In particolare, questo *Monobehaviour* costituisce un ponte tra Unity e Mapbox Studio, poiché è qui che dev'essere inserito l'URL relativo allo stile della mappa, pubblicato tramite lo Style Editor. È sempre attraverso questo script che, inserendo l'URL di un tileset custom, è possibile ricavarne i relativi dati geospaziali e utilizzarli su Unity, ad esempio per spawnare GameObjects in posizioni specifiche.

All'interno di questo script è definito un metodo largamente utilizzato in fase di sviluppo:

```
public virtual Vector3 GeoToWorldPosition(Vector2d latitudeLongitude, bool query-Height = true)
```

Come si può facilmente intuire dal nome, questo metodo è utilissimo in tutte le situazioni in cui è necessario passare dalle coordinate geografiche alle coordinate spaziali relative al mondo di Unity. Nel nostro caso, il metodo è stato chiamato

nello script *SpawnOnMapCustom()*, che abbiamo concepito per poter posizionare i target dell'Esploratore nelle coordinate specificate all'interno dello Scriptable Object relativo alla missione corrispondente.

È stato utilizzato anche il metodo *SetExtentOptions()*, per impostare a runtime i settaggi di visualizzazione della mappa. Il bisogno di settare questi dati a runtime deriva dalla necessità di utilizzare come riferimento il GameObject del giocatore, spawnato durante il gioco, in modo che la mappa venisse renderizzata attorno ad esso.

```
public virtual void SetExtentOptions(ExtentOptions extentOptions)
```

In particolare, è stato opportuno utilizzare la classe *RangeAroundTransformTileProviderOptions*, che eredita da *ExtentOptions*, per impostare il field *targetTransform* utilizzando proprio il giocatore come GameObject, come precedentemente detto.

Location Provider

Il comportamento di questo GameObject è definito dallo script *LocationProviderFactory*, il quale predispone quattro campi che ospitano due tipi di oggetti: *AbstractLocationProvider* e *DeviceLocationProviderAndroidNative*: il primo utilizza il servizio *Input.Location* di Unity, mentre il secondo, che eredita comunque da *AbstractLocationProvider*, rappresenta una versione ad hoc per i dispositivi Android. Nel caso quest'ultimo non venga specificato, i dispositivi Android possono comunque utilizzare l'*AbstractLocationProvider*.

Questi oggetti sono responsabili della localizzazione del giocatore e hanno fondamentalmente due comportamenti e scopi differenti: gli oggetti *DefaultDeviceLocationProvider* e *AndroidDeviceLocationProvider* utilizzano l'antenna GPS del dispositivo e funzionano solo nel caso in cui l'applicativo venga eseguito su dispositivo mobile, mentre gli altri sono necessari per testare il comportamento tramite l'editor. A questo proposito, si è rivelato particolarmente utile l'oggetto *EditorLocationProviderLocationLog*, in grado di leggere un file di testo contenente una serie di coordinate e di muovere il giocatore sul percorso corrispondente. In termini di ottimizzazione del workflow, questa feature si è rivelata fondamentale.

5.4.3 Session Manager

Session Manager è una classe fondamentale per quanto riguarda la localizzazione degli avatar dei giocatori presenti nella lobby. Esso viene aggiunto runtime al player prefab, ed è responsabile di inviare al server le coordinate GPS del dispositivo con una frequenza variabile a seconda delle impostazioni scelte. Al suo interno è presente una coroutine che, ad intervalli regolari (Figura 5.8), manda una command

al server per fornire le coordinate del dispositivo, oltre ad aggiornare l’interfaccia relativa alla qualità del segnale GPS. Una volta ricevuta la command, il server risponde inviando a tutti i dispositivi le coordinate ricevute, in modo tale che tutti i giocatori visualizzino nello stesso posto l’avatar relativo al dispositivo che ha comunicato la sua posizione, mantenendo quindi la coerenza spaziale dell’esperienza in tutte e tre le istanze del gioco.

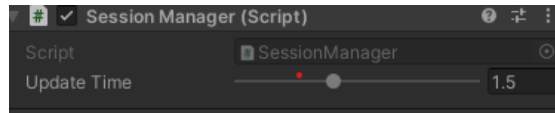


Figura 5.8: Il Session Manager nell’Inspector

5.5 Inizializzazione e looping dell’esperienza

5.5.1 Game Manager

Il *Loader* è un *GameObject* che possiede come attributo lo script *GameManager*, responsabile di alcune funzioni di caricamento e settaggio della scena *Game Main* e dell’attivazione e disattivazione di determinati oggetti durante il gioco. Una volta caricata la scena *Game Main*, il *Loader* è responsabile di:

- Impostare correttamente i componenti da aggiungere nella scena tramite il metodo *SetUpPlayer()*, come ad esempio i provider necessari alla localizzazione e all’orientamento del dispositivo;
- Istanziamento di componenti come *GemFactory* e *WhiteFragmentFactory*, che permettono lo spawn di Gemme e Frammenti di Magia Bianca, oppure l’aggiunta del componente *MagicPlayer* al giocatore locale, script responsabile ad esempio della funzione di raccolta dei collezionabili.
- Ottenere i riferimenti ad eventuali oggetti da disabilitare durante il corso dell’esperienza, come ad esempio la camera che, ogni volta che viene caricata una scena in AR, viene disattivata.
- Caricare nella scena l’avatar corretto in base al ruolo interpretato dal giocatore
- Impostare i comandi riguardanti lo zoom e, nel caso dell’Esploratore, la possibilità di sganciare la telecamera dal giocatore per poter visitare la mappa.

Queste responsabilità fanno del *Game Manager* un componente fondamentale sia durante l’impostazione iniziale della scena, sia durante l’esperienza poichè permette di non visualizzare oggetti virtuali non desiderati durante le scene AR.

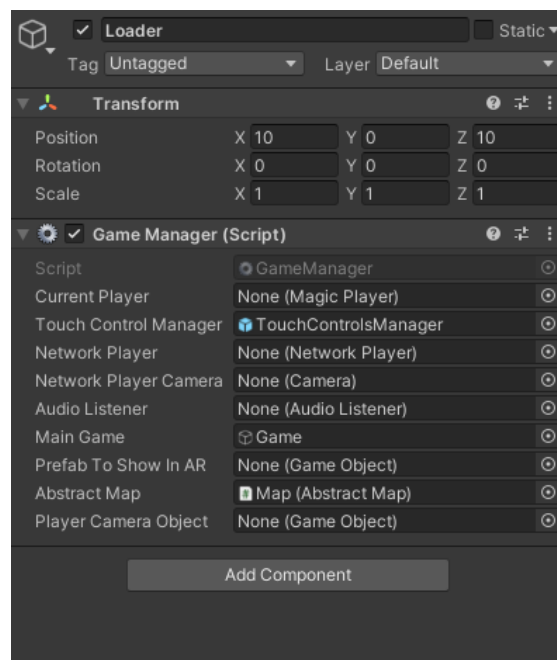


Figura 5.9: Il Game Manager

5.5.2 Missions Manager

Il *Missions Manager* è il *GameObject* responsabile del level design dell'esperienza di gioco. Esso possiede uno script omonimo nei quali sono presenti i riferimenti allo *ScriptableObject* *MagicTurinLevelDesign*: quest'ultimo, a sua volta, possiede una lista di *ScriptableObject* rappresentativi di ogni singola missione. Lo scopo di questo *GameObject* è molto semplice: ogni volta che una missione viene completata da un giocatore, il server viene informato ed incrementa l'indice della missione corrente, in modo tale che su tutti i dispositivi si progredisca nella lista dei task. Ogni *ScriptableObject* relativo a una missione ha un riferimento alla scena che dovrà essere caricata in maniera additiva e alle caratteristiche proprie della missione, come ad esempio il giocatore per cui deve essere caricata. Il *Missions Manager*, infine, gestisce l'interfaccia riguardante l'inizio e la conclusione delle missioni: quando un giocatore deve affrontare una missione o concluderla, vengono mostrate alcune finestre che chiedono conferma per proseguire nell'esperienza. I campi di testo di queste finestre vengono modificati in base a stringhe salvate negli *ScriptableObject* delle missioni: ciò fa sì che ad ogni iterazione l'interfaccia sia sempre differente, seguendo i progressi del giocatore.

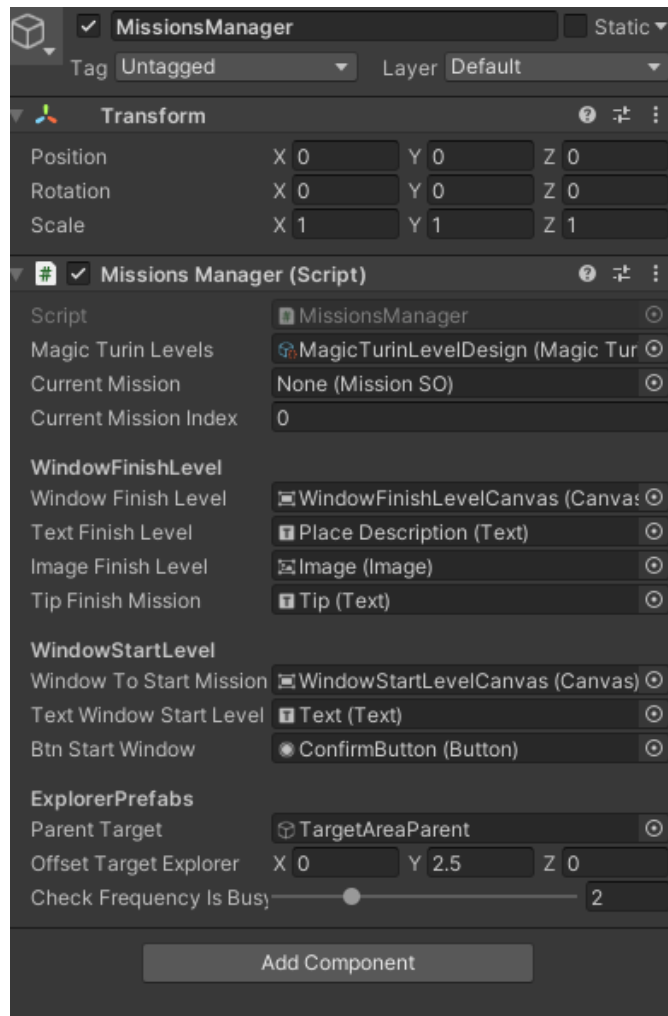


Figura 5.10: Il Missions Manager

5.6 Realtà Aumentata

Le scene che contemplano l'uso della Realtà Aumentata sono sei: due di esse riguardano le missioni del Saggio (il labirinto e il memory), altre due invece sono dedicate al Cacciatore (la distruzione della Gemma e il combattimento con i Demoni) e una, quella che consente di osservare i collezionabili, che è accessibile da tutti i giocatori. La sesta scena, invece, è quella collaborativa, in cui i tre giocatori devono risolvere un piccolo enigma per trovare una combinazione necessaria ad aprire un forziere. Queste scene hanno molti punti in comune dal punto di vista dello sviluppo. Il principale punto di contatto è chiaramente il tool utilizzato, ovvero *Lightship ARDK*, ma ci sono anche numerosi GameObject e componenti in comune.

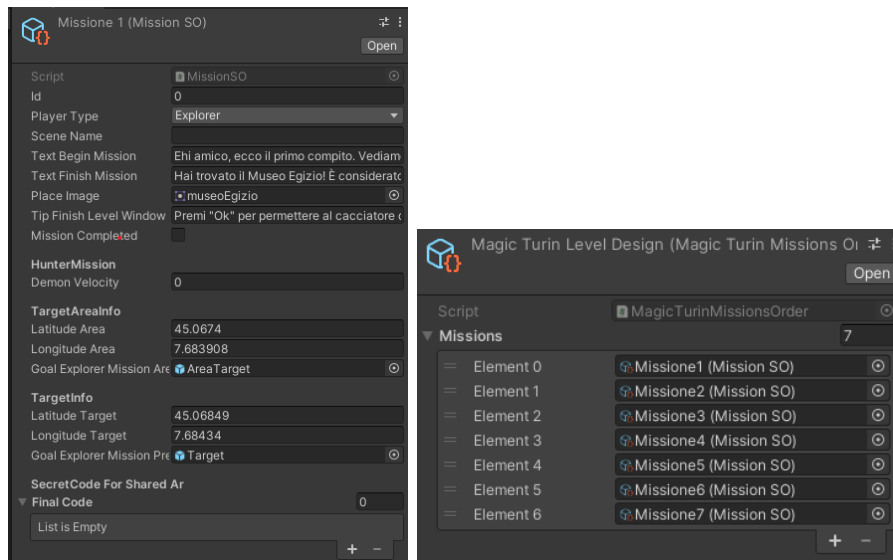


Figura 5.11: I dettagli di una singola missione (a sinistra) e il Level Design (a destra)

La gestione della realtà aumentata passa attraverso lo script *ARSessionManager*, che controlla il ciclo di vita della sessione AR: questo script può sia seguire il ciclo di eventi di Unity, sia utilizzarne uno proprio, gestito dallo sviluppatore. Nel nostro caso questa seconda opzione non si è rivelata necessaria. È consigliabile, anche se non strettamente necessario al funzionamento del sistema, affiancare all'*ARSessionManager* il componente *CapabilityChecker*, che ha il compito di verificare se il dispositivo è in grado di supportare l'AR. ARDK predispone, a questo scopo, il prefab *ARSceneManager*. Oltre agli script precedentemente descritti, questo *GameObject* ha un oggetto di tipo *Camera* come figlio: a questo oggetto, che rappresenta la *MainCamera*, sono attribuiti una serie di componenti che specificano la logica della scena. Gli script ricorrenti sono:

- *ARCameraPositionHelper*, che gestisce la coerenza spaziale tra il dispositivo e la camera di Unity;
- *ARPlaneManager*, in grado di rilevare superfici piane sia orizzontali che verticali e di mostrare, eventualmente, un prefab in corrispondenza di ciascuna superficie;
- *ARDepthManager*, necessario per calcolare e visualizzare le occlusioni;
- *ARHitTester*, per posizionare oggetti nel mondo reale attraverso il tocco sullo schermo. Questo componente non è però stato utilizzato nella versione fornita da ARDK, ma è stato customizzato (*CustomARHitTest*) per ottimizzarne il

funzionamento nel contesto dell'esperienza. Oltre alla logica dello script di partenza, è stato aggiunto un flag che consente o meno di posizionare oggetti multipli (se è false l'oggetto si sposta nella nuova posizione), e una variabile di tipo `Vector3`, settabile da inspector, utilizzabile come eventuale offset.

5.6.1 AR_LookAtItem

È la scena in realtà aumentata più semplice e viene caricata additivamente quando l'utente preme l'immagine di un collezionabile raccolto nell'inventario. Essa permette di visualizzare in realtà aumentata il collezionabile selezionato, posizionandolo nell'ambiente circostante tramite la scannerizzazione di quest'ultimo e della generazione di una mesh su cui individuare i punti di ancoraggio utili per mantenere costante la posizione dell'oggetto virtuale. L'utente è in grado di poter visualizzare i dettagli del collezionabile muovendosi intorno allo stesso nello spazio e di riposizionarlo nelle posizioni a lui più congeniali.

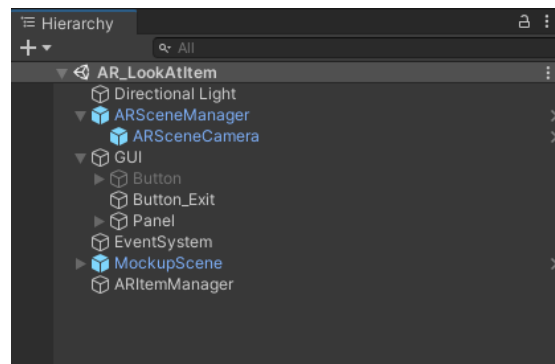


Figura 5.12: La hierarchy della scena

Nella Hierarchy sono presenti pochi componenti in cui il responsabile delle funzionalità in AR è la `ARSceneCamera` che presenta il `CustomARHitTest` per il posizionamento del collezionabile nell'ambiente, l'`ARDepthManager` responsabile delle occlusioni e l'`ARCursorRenderer` che permette di visualizzare il punto dove il collezionabile può essere spawnato oppure spostato. `MockupScene` è un prefab utile nella fase di test su Unity, poiché permette di testare le funzionalità nell'ambiente virtuale senza necessità di fare una build: questo prefab si disattiva automaticamente nell'apk. L'interfaccia infine si compone solamente di un bottone per tornare alla schermata principale.

5.6.2 AR_EnemyFight

In questa scena è implementato il combattimento in realtà aumentata con il Demone. Questa scena può essere caricata solo dal Cacciatore, in due casi ben precisi:

- quando è il suo turno all'interno del ciclo di missioni;
- quando visualizza un Demone sulla mappa e sceglie di combatterlo.

Fatti salvi i componenti descritti in precedenza, necessari per l'allestimento della scena in AR, le logiche più significative sono sicuramente quelle che descrivono la gestione degli incantesimi e l'intelligenza artificiale del nemico. Per quanto riguarda gli incantesimi, si è adottata la logica degli Scriptable Objects. Questo ha consentito di avere degli oggetti parametrizzabili e facilmente controllabili tramite codice. È stato sviluppato lo script *CastSpell*: questo componente è associato ad un bottone di tipo *LongClickButton*, e in particolare ad un evento di tipo *OnLongClick*. Questo evento è stato implementato appositamente per questo scopo, e consente di ricavare la durata della pressione del bottone. Per comodità, si è scelto di normalizzare questo valore per ottenere un valore compreso tra 0 e 1. Questa percentuale viene poi moltiplicata per la forza, una costante di tipo float specificata nello Scriptable Object corrispondente all'incantesimo lanciato, e attribuita all'incantesimo istanziato in scena. Gli incantesimi sono infatti oggetti dotati di fisica: oltre allo Sphere Collider, necessario per calcolare le collisioni con altri oggetti presenti in scena, sono dotati di un Rigidbody che consente di calcolarne la traiettoria sfruttando la gravità.

Ogni volta che si cerca di lanciare un incantesimo viene inoltre controllata la disponibilità di Mana del Cacciatore: se l'incantesimo che si desidera lanciare ha un costo (parametro specificato all'interno dello ScriptableObject corrispondente) maggiore del Mana disponibile viene istanziato un testo di avviso, in caso contrario viene sottratto il costo dell'incantesimo al Mana corrente. Per differenziare l'esperienza, si è pensato di inserire sia un incantesimo base, con un costo energetico contenuto e un danno arrecato non eccessivo, sia un incantesimo avanzato, che arreca un danno molto maggiore ma che allo stesso tempo sottrae una quantità di Mana considerevole.

Relativamente al comportamento del Demone, nel corrispondente MonoBehaviour vengono specificati i meccanismi alla base del suo movimento e viene gestita la macchina a stati che ne controlla le animazioni. Per quel che riguarda il movimento, si è deciso di implementare una dinamica che stimolasse la "fisicità" dell'esperienza: il demone, infatti, una volta fatto spawnare sul terreno tramite l'ARHitTester, ha come target il dispositivo mobile del giocatore, rappresentato nella scena dall'ARSceneCamera. Questo rende possibile la configurazione di un vero e proprio inseguimento. Per rendere più interessante questa missione, si è pensato di inserire anche dei parametri di controllo, come la distanza Demone-Cacciatore e il campo

visivo del Demone, finalizzati a stabilire se quest'ultimo è in grado di vedere, e quindi inseguire, o meno il Cacciatore.

La gestione delle animazioni, come anticipato, passa attraverso un AnimationController dotato di una macchina a stati: ogni animazione viene infatti controllata tramite una serie di parametri di tipo Trigger o Bool, settati dinamicamente a runtime in seguito al verificarsi di opportune situazioni. Per fare un esempio, nel momento in cui il demone viene colpito da un incantesimo, condizione verificata tramite l'evento OnTriggerEnter(), il parametro "Damage" viene azionato e viene quindi lanciata la relativa animazione.

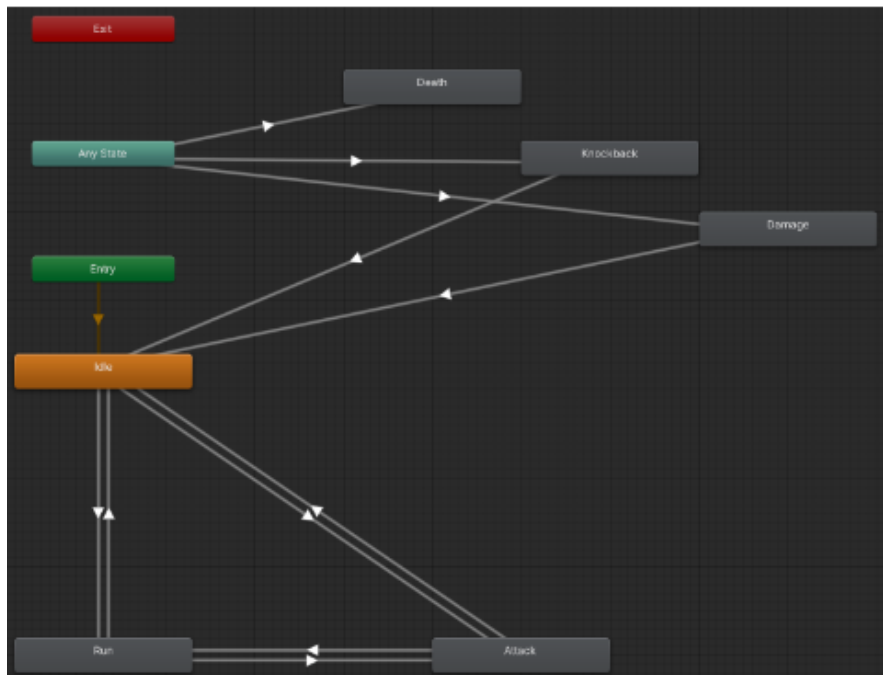


Figura 5.13: La macchina a stati che controlla le animazioni del demone

Per aumentare gradualmente la difficoltà, la velocità di movimento del demone aumenta in base all'indice della missione corrente all'interno del MissionsManager.

5.6.3 AR_DestroyGem

Questa scena è stata sviluppata per consentire al Cacciatore di distruggere una Gemma e generare di conseguenza un certo numero di Frammenti di Magia Bianca. La caratteristica principale alla base di questa sezione è sicuramente legata alla fisica che si è deciso di attribuire alla Gemma stessa. Si è optato per sfruttare le feature di scanning in AR proposte da ARDK per consentire al dispositivo mobile di raggiungere un buon livello di Context Awareness e di capire quindi

com'è fatto l'ambiente circostante, con lo scopo di usare i dati spaziali ottenuti per generare interazioni tra gli oggetti virtuali (in questo caso la Gemma) e quelli reali (la strada e gli edifici). La scena si apre con un breve tutorial, necessario per dare agli utenti le informazioni necessarie ad effettuare una buona scansione dell'ambiente e a spawnare correttamente la Gemma nello spazio. Successivamente, il giocatore può dare inizio al processo di scanning e, non appena il dispositivo riconosce l'ambiente, utilizzare l'ARHitTester per istanziare la Gemma. Una volta spawnato il GameObject, il Cacciatore ha la possibilità di colpirlo utilizzando il sistema di incantesimi precedentemente descritto e, dopo un numero variabile di tentativi andati a buon fine, di distruggerlo. È previsto, nel caso in cui la Gemma cadesse nel vuoto a causa di una scansione non accurata, il reloading della scena.

5.6.4 AR__WisemanMemory

Il memory rappresenta uno degli enigmi pensati per il Saggio. La scena, dopo un breve tutorial illustrativo, prevede l'utilizzo dell'ARHitTester per far spawnare un oggetto di tipo MemoryManager. Questo GameObject possiede i 18 spawn points necessari a indicare la posizione in cui istanziare, secondo la logica del Memory Manager, le 9 carte e i rispettivi duplicati. Ogni carta è dotata di uno script molto semplice, Card: quando viene toccata sullo schermo, viene scatenato l'evento OnMouseDown, che triggerà l'animazione scoprendo la carta. In questo momento parte una coroutine che, dopo un intervallo di tempo di 2 secondi, controlla se esiste una seconda carta scoperta. Se questa esiste, lo script verifica un eventuale match, altrimenti rimane in attesa di una seconda carta. Nel caso le due carte scoperte risultino uguali, vengono distrutte. Una volta scoperte tutte le carte, il memory è completato e la missione termina.

5.6.5 AR__MazeGame

È la prima missione del saggio e consiste nel risolvere un labirinto in realtà aumentata muovendo una pallina fino all'obiettivo attraverso un labirinto. La scena presenta un'interfaccia con quattro bottoni, necessari per muovere la pallina, e un bottone per poter riposizionare il labirinto in caso quest'ultimo sia occluso da oggetti reali rendendo quindi difficoltoso il gameplay. La scena presenta l'ARSceneCamera, utile per la riproduzione di tutte le features in AR: essa possiede come attributi i sopracitati ARCursorRender e ARDepthManager, oltre all'ARPlaneManager, componente utile a identificare i piani tramite gli algoritmi di computer vision di ARDK, e il PlaceDynamicARObject, componente utile allo spawn del labirinto. Il labirinto è generato attraverso uno script MazeLoader che crea una griglia composta da un numero specificato nell'inspector di righe e colonne. Successivamente, a questa griglia, viene applicato un algoritmo di Hunt Kill, il

quale permette di generare i corridoi del labirinto: partendo da una cella specifica (0,0), l'algoritmo cerca iterativamente una cella vicina in modo tale da generare un percorso, che viene interrotto soltanto quando si trova un vicolo cieco, ovvero una cella che non ha vicini già esplorati (ovvero celle che non sono parte di percorsi già tracciati).

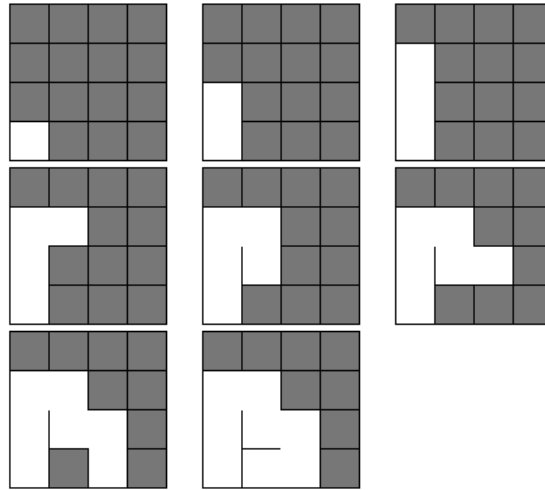


Figura 5.14: Genesi procedurale del labirinto (1)

Una volta arrivati alla dead-end di un percorso, l'algoritmo entra in modalità “caccia”, in cui esegue la scansione della griglia alla ricerca di celle inesplorate adiacenti a quelle già visitate. Se trovata, considera questa cella inesplorata come nuovo punto di partenza, genera un percorso tra la cella visitata e quella adiacente inesplorata e ricomincia a generare un percorso randomico come in precedenza. L'algoritmo si ferma quando non ci sono più celle inesplorate.

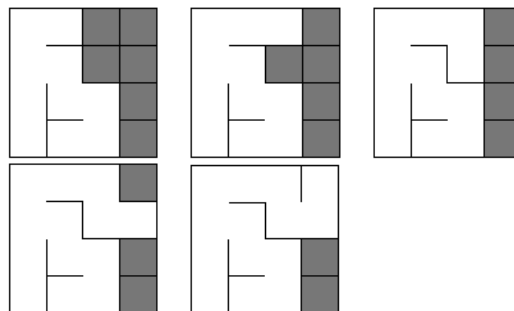


Figura 5.15: Genesi procedurale del labirinto (2)

Questo algoritmo permette che tutte le celle della griglia infine siano in comunicazione e dunque ogni parte del labirinto sia esplorabile.

La missione consiste nel muovere grazie ai quattro bottoni inferiori dell'interfaccia una pallina fino al punto di uscita del labirinto rappresentato da un cubo bianco. I punti di spawn di questi due oggetti è sempre fisso e si basano ovviamente sulla posizione del labirinto: la pallina ha spawn corrispondente al centro della cella (0,0), mentre l'arrivo al centro della cella (n-1,m-1), dove n e m rappresentano rispettivamente il numero di righe e colonne del labirinto.

5.6.6 AR_Collaborative

La missione collaborativa CollaborativeAR sfrutta le funzionalità di networking di ARDK descritte precedentemente nel capitolo delle Tecnologie.

In una prima fase iniziale viene chiesto ai tre giocatori di scannerizzare insieme un oggetto che abbia determinate caratteristiche (asimmetrico, non riflettente, ecc. . .) e ad un solo giocatore viene chiesto di decidere dove localizzare il punto di spawn del baule. Quando tutti e tre i dispositivi hanno scannerizzato l'oggetto e il luogo di spawn è stato selezionato, viene generato un baule in realtà aumentata in una posizione univoca per tutti e tre i dispositivi. Ai giocatori viene chiesto dunque di cercare ai lati del baule due fogli, di comprendere quale sia la combinazione scoprendo i numeri in comune, e di inserirla nel lucchetto per riuscire ad aprire la cassa e dunque terminare la missione.

Nella hierarchy della scena, oltre all'ARSceneCamera, che possiede i componenti ARHitTest, ARCursorRender e ARPlaneManager, è presente il GameObject SharedARManager responsabile della gestione della realtà aumentata condivisa.

La gestione della realtà aumentata condivisa in ARDK comporta che il primo dispositivo a collegarsi ad una sessione sia identificato come host: il manager responsabile controlla quale dispositivo sia host e disattiva agli altri due i componenti ARPlaneManager, ARCursorRender e ARHitTest, poiché essi servono solamente all'host per identificare il punto in cui verrà posizionato l'indicatore per lo spawn della cassa. Quando i tre dispositivi confrontano le tre scansioni dell'oggetto preso in esame dai giocatori, queste tre scansioni si allineano andando a generare uno spazio di riferimento univoco per tutti. La posizione dell'indicatore di spawn viene passata allo SharedARManager, il quale si occupa di trasmettere a tutti i dispositivi quella che sarà la localizzazione del baule, tramite un MessagingManager che sfrutta il sistema di comunicazione proposto dalle features di network ARDK. La rotazione delle rotelle numerate del lucchetto è stata resa coerente grazie all'utilizzo di Mirror, più stabile rispetto al servizio di Networking proposto da ARDK, in modo tale da rendere la comunicazione tra i dispositivi più affidabile.

La GUI presente è minimale e si compone, nella prima fase di scannerizzazione, di un testo che indica ai giocatori se si è Leader (Host) o Fellow (non host), e di un ulteriore testo, che dà invece consigli su come effettuare la scansione dell'oggetto.



Figura 5.16: La fase di scanning vista in Play Mode su Unity

Dopo lo spawn del baule si mantiene solamente il secondo campo di testo, che diventa responsabile di fornire suggerimenti ai giocatori per risolvere l'enigma.



Figura 5.17: Lo spawn del baule testato in Play Mode su Unity

L'interfaccia in questa scena svolge un ruolo fondamentale a causa della meccanica molto complessa di scannerizzazione al fine di generare uno spazio di riferimento univoco. Tutti i componenti della GUI sono gestiti dallo UIManagerSharedAR,

script presente sull'oggetto GUI.

5.6.7 Statistiche

Al termine del ciclo di missioni viene caricata, su tutti i dispositivi, una scena che riassume i collezionabili raccolti da ciascun giocatore. L'interfaccia contiene una scroll view orizzontale che permette di passare da un giocatore all'altro e, per ogni giocatore, predispone una seconda scroll view, questa volta verticale, per visualizzare i contenuti. Ciascun utente può scegliere di salvare i dati della partita: tramite il toolkit di Firebase, infatti, è possibile caricare sul database un file Json contenente l'indice della partita, la data e le informazioni riguardanti i collezionabili raccolti.

Capitolo 6

Risultati e analisi

In questo capitolo verrà illustrata la fase di sperimentazione dell'applicativo, dalla formulazione dei due questionari sottoposti ai tester all'analisi dei risultati e la valutazione di possibili miglioramenti che potrebbero essere implementati sia dal punto di vista tecnico, sia dal punto di vista del game design.

6.1 Formulazione dei questionari

Si è deciso di strutturare l'indagine in due questionari distinti: uno preliminare, fatto compilare prima del test, e un questionario finale, sottoposto invece al termine dell'esperienza per valutarne più nello specifico gli aspetti.

6.1.1 Questionario Pre-Test

Oltre a raccogliere dati relativi ai tratti anagrafici dei tester, al loro ambito di studio o lavoro e ai rispettivi gusti ludici, l'obiettivo di questa survey è quello di indagare sulla conoscenza delle principali features, sia tecniche che di design, dell'applicativo. Sono state poste dunque domande che mirano a quantificare il livello di consapevolezza, e di eventuale esperienza pregressa, relativa a quattro delle cinque principali caratteristiche.

- Serious Games
- Realtà aumentata
- Giochi Location-based
- Giochi intergenerazionali

Nella parte finale del questionario viene posta una domanda per valutare anche un possibile interesse a provare un videogame che racchiuda al suo interno queste caratteristiche, per cercare di comprendere il grado di curiosità che l'esperienza è in grado di stimolare.

6.1.2 Questionario Post-Test

Il questionario post-test è composto da una parte riguardante l'usabilità dell'applicazione e da una serie di sezioni per valutare le funzionalità, sia da un punto di vista tecnico che di design. Sebbene l'applicativo sviluppato sia un prototipo, e dunque necessiti di ovvi miglioramenti per essere considerabile un prodotto destinabile al mercato, è stato reputato opportuno valutare tutte le caratteristiche, usabilità compresa, per comprendere il livello raggiunto e dove intervenire maggiormente per ottenere miglioramenti sensibili. Il test è diviso in tredici sezioni, ciascuna con un obiettivo specifico:

- **Usabilità dell'applicazione (SUS):** per valutare l'usabilità è stato utilizzato il SUS (System Usability Scale). Trattandosi di un questionario standard, oltre a fornire un calcolo diretto per ricavare uno score su una scala da 0 a 100, dà la possibilità di confrontare questo punteggio con alcuni valori standard (es. score minimo per considerare un applicativo sufficientemente usabile).
- **Valutazione del gameplay (intrattenimento, bilanciamento, ecc...):** sono state poste sia domande specifiche per ogni ruolo, relative a bilanciamento, alla facilità di apprendimento delle meccaniche e alla godibilità dell'esperienza, sia domande riguardanti aspetti comuni, come il tutorial e la missione in condivisa in realtà aumentata. Nell'indice di godibilità si tiene conto non solo del gameplay ma anche del fattore anagrafico e di quanto questo possa essere incisivo sulla scelta della tipologia di gameplay.
- **Valutazione della possibile utilità dell'applicativo per un tour cittadino e dunque della sua natura serious:** sono presenti domande riguardanti l'interesse stimolato nei confronti della città e delle location in cui è ambientato il gioco, in modo tale da comprendere quanto il focus del giocatore sia concentrato sull'esperienza e quanto sull'ambiente circostante. Queste osservazioni sono importanti per stabilire se questo videogioco possa avere anche un'utilità tipica dei Serious Games, affiancando al divertimento informazioni relative ai luoghi esoterici di Torino.
- **Valutazione dell'intergenerazionalità dell'applicazione e del design dei 3 ruoli:** nel terzo paragrafo dello stato dell'arte sono state identificate delle linee guida che permettessero l'ideazione di un gameplay asimmetrico

dell'esperienza e di creare dunque i tre ruoli interpretabili dai giocatori. Questa sezione del questionario aiuta a comprendere quanto le missioni caratterizzanti, per ogni ruolo, siano piaciute al giocatore rispetto alle mansioni svolte dagli altri giocatori, in modo da confermare o confutare le linee guida estrapolate nello stato dell'arte. È stato chiesto ai tester se l'esperienza sia stata di loro gradimento o se avrebbero preferito interpretare un altro ruolo, più adatto ai loro gusti videoludici personali.

- **Valutazione del gradimento della collaborazione:** le domande di questa sezione mirano a valutare la tipologia di collaborazione preferita dai giocatori. Nel gameplay, infatti, si alternano missioni single player, ma che comunque richiedono l'aiuto degli altri membri (nella risoluzione di enigmi, il cacciatore necessita di energia dal saggio, l'esploratore deve raccogliere le gemme utili a ottenere energia), a missioni collaborative in realtà aumentata, il che rende necessario indagare su quale delle due sia la modalità preferita. Sono presenti, inoltre, anche domande riguardanti l'affiatamento del gruppo di tester, al fine di valutare l'efficacia della collaborazione come collante dell'esperienza.

6.2 Test

I test sono stati condotti tra lunedì 27 giugno 2022 e lunedì 4 luglio 2022 e hanno coinvolto 26 persone suddivise in 10 gruppi (in quattro casi i gruppi erano formati da due utenti, quindi il terzo componente è stato uno dei due sviluppatori). I ruoli sono stati sia assegnati a persone appartenenti a fasce anagrafiche per cui il design di gioco del ruolo che hanno interpretato era stato studiato, sia assegnati in base ai gusti ludici dei tester, per comprendere al meglio se le mansioni pensate per una certa fascia d'età potessero risultare interessanti anche per gli altri componenti del team. L'applicativo è stato sperimentato in una porzione del centro storico di Torino, più precisamente tra Piazza San Carlo e i Giardini Reali. Ad ogni slot sono stati dedicati 90 minuti, una stima che si è rivelata adeguata: in media l'esperienza è durata per ogni gruppo circa quarantacinque minuti, ai quali si sommano il tempo necessario alla compilazione dei due questionari, una breve introduzione all'esperienza e i tempi refrattari di spostamento per raggiungere il punto di ritrovo una volta finito il test.

6.3 Valutazione dei risultati

6.3.1 Questionario Pre-Test

Dal questionario pre-test è stato possibile valutare le conoscenze pregresse dei tester riguardo le principali features dell'esperienza. Anagraficamente, i tester risultano

appartenenti per la maggior parte ad una fascia d'età giovane e adulta, affine a ruoli come il cacciatore e l'esploratore. Circa un decimo dei tester risulta over 60, età per cui è stato pensato il ruolo del Saggio. La popolazione in esame mostra che

Qual è la tua età?

26 risposte

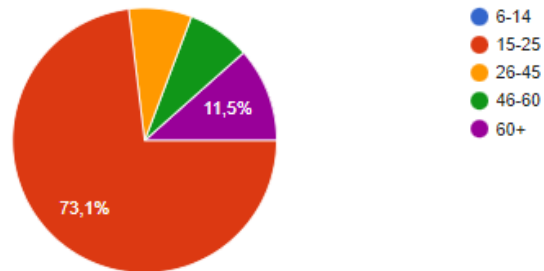


Figura 6.1: Distribuzione anagrafica percentuale della popolazione.

solamente il 61.5% dei partecipanti ai test gioca ai videogiochi, e solamente il 50% di questi è un giocatore assiduo.

Giochi ai videogame?

26 risposte

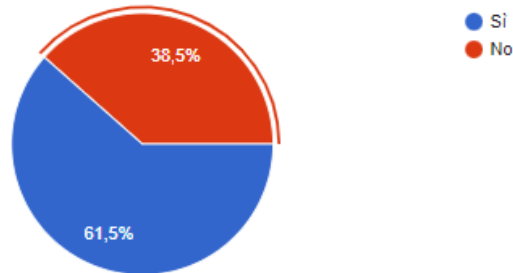


Figura 6.2: Distribuzione percentuale dei videogiocatori nella popolazione in esame.

Per quanto riguarda le conoscenze relative alle caratteristiche tecniche e di design sono state poste alcune domande per comprendere se la popolazione in esame conoscesse o meno la definizione di tali caratteristiche e se avesse mai avuto esperienza con applicativi che le comprendessero.

Quanto spesso?

16 risposte

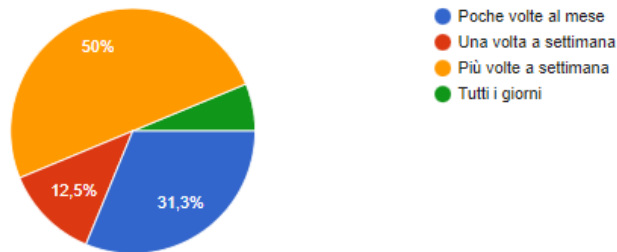


Figura 6.3: Distribuzione percentuale della frequenza di gioco dei videogiocatori della popolazione in esame.

Serious Games

Solamente l'11.5% ha affermato di conoscere il significato di "Serious Game". Successivamente, dopo aver fornito la definizione, è risultato che il 42.3% avesse già giocato ad un videogame con queste caratteristiche e dunque che i risultati inerenti alla prima domanda fossero correlati ad una scarsa consapevolezza nella classificazione del gioco. Inoltre il 96.2% del campione ripone fiducia nei serious game come possibile mezzo di valore per il raggiungimento di obiettivi al di fuori del puro intrattenimento.

Sai cos'è un serious game?

26 risposte

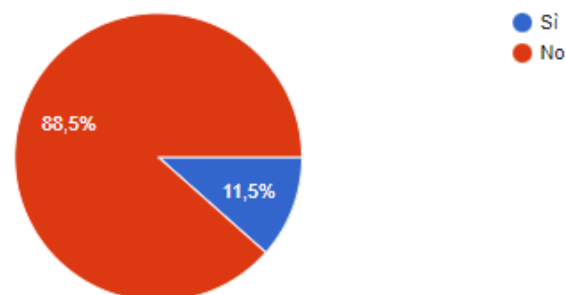


Figura 6.4: Percentuale della popolazione che conosce il termine "Serious Game".

Realtà aumentata

L'88.5% della popolazione esaminata risulta conoscere la definizione di realtà aumentata, ma solamente il 65.4% afferma di averne avuto esperienza diretta, mentre l'11.5% afferma di utilizzarla frequentemente. A coloro che hanno utilizzato l'AR è stata posta un'ulteriore domanda per indagare quale fosse il dispositivo maggiormente utilizzato in queste esperienze, ed è emerso che la netta maggioranza dei fruitori avesse utilizzato un dispositivo mobile come tablet o smartphone.

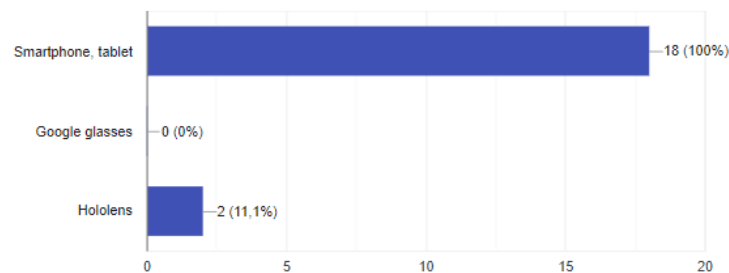


Figura 6.5: Dispositivi utilizzati dal campione nelle precedenti esperienze in AR.

Location-based Games

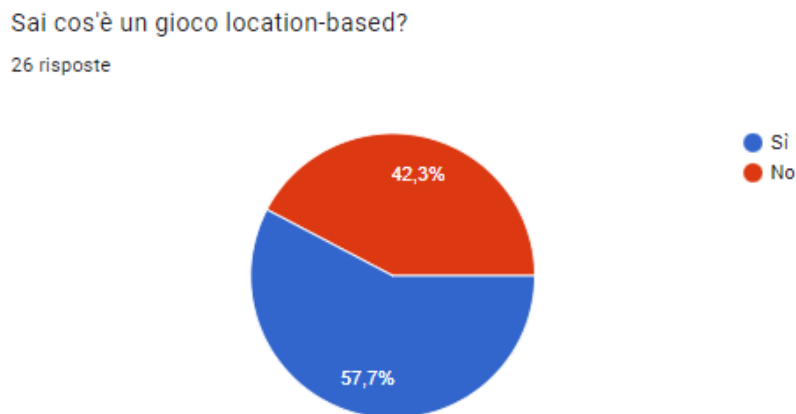


Figura 6.6: Percentuale della popolazione che conosce l'espressione "Location-based Game".

Il 57.7% dei tester conosce l'espressione "Location-based game", e solamente il 53.8% della popolazione ne ha avuto esperienza diretta. Il gioco location-based maggiormente utilizzato e conosciuto tra la popolazione è risultato essere PokémonGo, ma sono stati giocati anche i meno celebri Ingress e Harry Potter: Wizards Unite. L'84.6% della popolazione infine si è rivelata ottimista per quanto

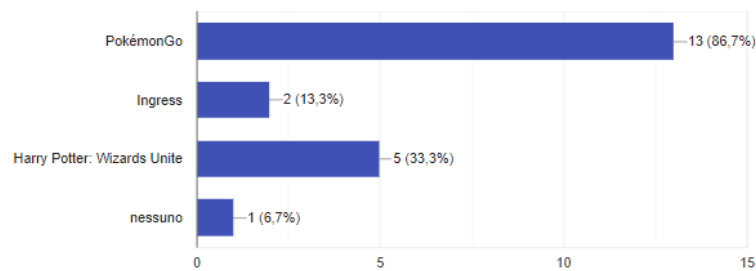


Figura 6.7: Videogiochi location-based giocati dalla popolazione in esame.

riguarda il possibile impiego di un location-based game in un contesto turistico.

Giochi Intergenerazionali

Il 61.5% dei tester ha dichiarato di non conoscere la definizione di gioco intergenerazionale e solamente il 30.8% ha dichiarato di averne avuto esperienza. Quando è stato chiesto se questo potesse essere valutato come un possibile mezzo per avvicinare generazioni anagraficamente distanti, il 65.4% ha risposto positivamente.

Sai cos'è un gioco intergenerazionale?

26 risposte

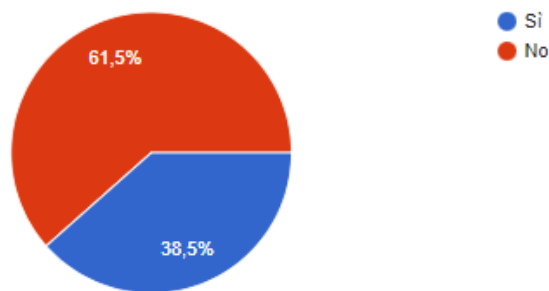


Figura 6.8: Percentuale della popolazione che conosce il termine "videogioco intergenerazionale".

È stato infine chiesto se un videogame che racchiudesse all'interno tutte queste caratteristiche potesse risultare interessante, raccogliendo la totalità delle risposte positive.

6.3.2 Questionario Post-Test

Il questionario post-test è stato valutato per sezioni differenti con diversi criteri, sottoponendo una serie di affermazioni valutabili secondo una scala da 1 a 5 (1: per niente d'accordo, 5: completamente d'accordo):

Usabilità (SUS)

Sebbene l'applicativo sia ancora in una forma embrionale è stato ritenuto opportuno valutare il livello di usabilità raggiunto dallo stesso. È stato utilizzato, come metodo valutativo, il SUS (System Usability Scale) considerando sia il risultato sulla popolazione totale, sia sulle frange over 50 e under 30. I risultati sono stati incoraggianti poiché i punteggi conseguiti superano la soglia media di 68 punti. Con una media sulla popolazione totale di 72.02 punti su 100, sulla popolazione under 50 di 72.62 e sulla popolazione over 50 di 69.5 su 100, l'applicativo risulta già utilizzabile sebbene di molto migliorabile. La differenza di punteggio tra le due fasce di età analizzate sottolinea una maggior predisposizione alle nuove generazioni nel comprendere più facilmente determinate procedure e funzionalità.

SUS GLOBALE	72,02
SUS OVER 50	69,5
SUS UNDER 30	72,62

Tabella 6.1: Valutazione dell'usabilità

Sono stati rilasciati inoltre alcuni commenti riguardanti alcuni aspetti che hanno generato alcune difficoltà o perplessità nei tester. È stata riportata da alcuni utenti la necessità di consultare nuovamente il tutorial o una guida durante il gameplay che potesse permettere ai giocatori di rivedere le innumerevoli funzionalità. Inoltre alcune meccaniche sono risultate poco intuitive, come ad esempio il lancio degli incantesimi del cacciatore, per cui molti tester, anziché tenere premuto il bottone, scorrevano con il dito sullo schermo. Sono stati individuate inoltre alcune difficoltà nello smistamento delle risorse: la dinamica di invio dei Frammenti di Magia Bianca andrebbe migliorata, poiché alcune volte il Saggio sprecava risorse inviando energia al personaggio sbagliato. Inoltre, la modalità di invio, differente per Frammenti di Magia Bianca e Gemme, ha generato confusione in alcuni tester.

Valutazione del raggiungimento di un design di gioco “Serious” funzionale

In generale la popolazione testata ha affermato di voler avere più informazioni riguardanti i punti di interesse presenti sul percorso intrapreso durante il test e i luoghi turistici visitati, questo ad indicare un interesse diffuso verso il contesto cittadino ed esoterico di Torino. Inoltre, i tester hanno manifestato interesse nell'utilizzare un'applicazione simile in un contesto turistico, affermando inoltre di essere incuriositi riguardo ai luoghi visitati grazie all'applicazione e confermando quindi la potenziale utilità di tale applicativo per migliorare un tour cittadino. Osservando le risposte illustrate in Tabella 6.2, si nota che la popolazione over 50 preferirebbe avere informazioni riguardo ai luoghi turistici, approfondendo le informazioni rilasciate durante il gioco, mentre la popolazione under 30 riferisce di essere interessata maggiormente a ricevere informazioni riguardanti punti di interesse come ristoranti o negozi. In alcuni commenti viene suggerito di stringere partnership coi negozi del centro di Torino: le attività potrebbero, per esempio, rilasciare Gemme agli utenti in seguito ad acquisti o aumentare la propria visibilità richiedendo il posizionamento di collezionabili nelle immediate vicinanze. Inoltre, per aumentare il grado di immersione all'interno del contesto cittadino, alcuni utenti suggeriscono di inserire elementi interattivi per la città, oltre a quelli delle missioni, che spingano ulteriormente i giocatori a guardarsi intorno e ad apprezzare l'ambiente circostante.

Valutazione dell'efficacia della collaborazione

La valutazione dell'efficacia della collaborazione è stata nel complesso molto positiva: il gioco risulta stimolare i giocatori a collaborare tra di loro, grazie anche alla differenza tra i gameplay e alla loro complementarietà. Come è possibile osservare dai dati in Tabella 6.3, i tester affermano di sentirsi parte di una squadra e di aiutarsi a vicenda durante il completamento dei propri task riconoscendo un ruolo centrale alla collaborazione durante l'arco dell'esperienza. Infine nella valutazione delle due modalità di collaborazione presente all'interno del gameplay (diretta, come nella missione del baule, o indiretta, come nelle missioni canoniche) gli utenti non manifestano una particolare preferenza apprezzando entrambe le modalità in egual modo. Le risposte mediamente risultano avere un valore medio maggiore per i tester che superano la soglia dei cinquant'anni, sebbene le valutazioni della maggioranza delle domande sia simile tra le due fasce anagrafiche: è interessante la differenza rilevata nella nona domanda, in cui i tester over 50 dichiarano di essersi fatti aiutare maggiormente nello svolgimento delle proprie mansioni (3,8 su 5 in confronto al 2.33 su 5 degli under 30) a indicare probabilmente la necessità di dover comunicare e confrontarsi col team per comprendere al meglio come approcciare il gameplay.

DOMANDE	MEDIA UNDER 30	MEDIA OVER 50	MEDIA GLOBA- LE
Mi sarebbe piaciuto ricevere suggerimenti rispetto a punti di interesse nelle vicinanze (ristoranti, musei, negozi...).	4,10	3,60	4,00
Avrei gradito ricevere più spunti e dettagli relativi ai luoghi che ho visitato.	3,76	4,20	3,85
Se mi trovassi in vacanza (in famiglia o con amici) in una città che non ho mai visitato, un'esperienza come questa stimolerebbe la mia curiosità.	4,38	4,80	4,46
Durante l'esperienza mi sono concentrato più sul contenuto virtuale che sull'ambiente intorno a me.	3,67	3,60	3,65
L'esperienza mi ha incuriosito relativamente ai luoghi che ho visitato.	3,81	4,60	3,96

Tabella 6.2: Valutazione raggiungimento di un design di gioco "Serious" funzionale.

Intergenerazionalità e gameplay

Si è deciso di valutare complessivamente l'efficacia delle caratteristiche intergenerazionali per ogni tipologia di gameplay. La popolazione di indagine è stata suddivisa tra giocatori assegnati a ruoli pensati coerentemente con la loro età e giocatori assegnati in base al proprio gusto ludico o casualmente in modo tale da poter valutare anche la godibilità dei diversi gameplay su diverse fasce anagrafiche. In una prima parte del questionario, uguale per tutti i tester, sono state valutate le risposte alle seguenti domande. Dai risultati indicati in Tabella 6.4, risulta che i tester che hanno intrapreso un gameplay pensato coerentemente con la loro età hanno apprezzato maggiormente la tipologia di mansioni a loro richieste, dichiarando di non tendere a voler provare un ruolo differente da quello interpretato. Questi hanno inoltre espresso di aver trovato meno interessanti le mansioni degli altri giocatori in confronto agli altri tester, che hanno dichiarato di aver aiutato gli altri giocatori poiché trovavano più intrattenenti i loro task. In entrambe le divisioni in cui è stata divisa la popolazione le abilità psico-fisiche richieste dalle mansioni sono state ritenute adeguate.

Il questionario sul ruolo dell'Esploratore (dati in Tabella 6.5) presenta valutazioni che si discostano poco tra giocatori anagraficamente coerenti e quelli non coerenti,

DOMANDE	MEDIA UNDER 30	MEDIA OVER 50	MEDIA GLOBA- LE
Il gameplay stimola la collaborazione tra gli utenti.	4,19	4,60	4,27
La collaborazione ha un ruolo centrale nel gameplay.	4,05	4,60	4,15
La collaborazione tra i miei compagni mi ha permesso di procedere in maniera più spedita nel gameplay.	3,86	4,20	3,92
La collaborazione è stata incentivata dal fatto di avere mansioni e abilità differenti.	4,33	4,40	4,35
Ho preferito la collaborazione indiretta (mansioni differenti) a quella diretta (il baule in realtà aumentata).	2,19	2,60	2,27
Mi sono sentito parte di una squadra.	4,33	4,80	4,42
Il mio ruolo è fondamentale per gli altri membri.	4,24	4,40	4,27
Gli altri membri sono stati fondamentali per il completamento dei miei task.	4,48	4,80	3,73
Mi sono fatto aiutare nello svolgimento delle mie mansioni.	2,33	3,80	2,62
Ho aiutato gli altri membri nello svolgimento delle loro mansioni.	2,71	2,60	2,69

Tabella 6.3: Valutazione dell'efficacia della collaborazione

a indicare una maggiore versatilità del ruolo che è pensato per un'ampia fascia d'età (23 - 55 anni).

Il questionario riguardante il gameplay del Cacciatore sottolinea come alcune features e mansioni richieste al suddetto ruolo siano state apprezzate maggiormente dalle persone appartenenti alla fascia anagrafica per cui esso è stato pensato. In Tabella 6.6 si osserva che giocatori più giovani hanno trovato la battaglia col demone più avvincente e hanno apprezzato maggiormente i task di distruzione delle gemme al fine di produrre magia bianca.

Il questionario relativo al Saggio sottolinea come il ruolo sia stato apprezzato da giocatori di età coerente e non con il ruolo, mostrando buone valutazioni nella media globale (Tabella 6.7). Purtroppo la scarsa reperibilità di tester over 60,

DOMANDE	Anagrafic. COERENTI	Anagrafic. INCOERENTI	MEDIA GLOBALE
Il ruolo che ho interpretato ha le mansioni che più si addicono al mio gusto ludico.	4,31	3,77	4,04
Avrei voluto interpretare un altro ruolo rispetto a quello che mi è stato assegnato.	1,69	1,85	1,77
Ho aiutato i miei compagni poiché il loro gameplay era più interessante.	1,62	2,30	1,96
I task che ho dovuto affrontare si sono rivelati meno accattivanti di quello che mi aspettavo.	2,08	1,85	1,96
Ho avuto difficoltà fisiche nel completare le mie mansioni.	1,08	1,23	1,15
Le mie abilità mentali sono risultate sufficienti al fine di eseguire i miei task.	4,62	4,46	4,54
Ho avuto difficoltà a comprendere le meccaniche di funzionamento del mio ruolo.	1,85	1,92	1,88
Le mansioni che ho svolto hanno una durata adeguata.	3,08	3,31	3,19

Tabella 6.4: Valutazione del gameplay intergenerazionale

DOMANDE	Anagrafic. COERENTI	Anagrafic. INCOERENTI	MEDIA GLOBALE
Ho trovato il sistema di navigazione efficace.	3,37	4,00	3,50
L'area di ricerca dell'obiettivo è troppo vasta.	1,62	1,50	1,60
La meccanica riguardante la ricerca dell'obiettivo è chiara.	3,25	4,00	3,40
Orientarmi è stato facile.	3,50	4,00	3,60
I meccanismi per controllare il panning e lo zoom mi hanno aiutato ad orientarmi sulla mappa.	3,25	4,00	3,40
Nel guidare il gruppo, ho preferito prendermi la libertà di esplorare il più possibile piuttosto che scegliere il percorso più breve.	2,25	3,00	2,40
Ho avuto spesso bisogno di attivare il navigatore.	2,50	3,50	2,70
Il navigatore consumava Mana troppo velocemente.	1,87	3,00	2,10

Tabella 6.5: Valutazione delle features dell'Esploratore

ovvero appartenenti alla fascia anagrafica di riferimento per il ruolo, non permette di avere risultati rappresentativi. È risultato molto interessante un commento che sostiene che il ruolo del saggio debba comprendere meno task al fine di far concentrare l'anziano più sulla collaborazione e meno sulla risoluzione di enigmi per semplificare al massimo il ruolo e permettere così di renderlo più fruibile per le generazioni più anziane. Questo è molto interessante poiché permetterebbe all'anziano di partecipare ad un'attività senza porlo sotto eccessivo stress per fargli imparare le meccaniche di gioco.

DOMANDE	Anagrafic. COERENTI	Anagrafic. INCOERENTI	MEDIA GLOBALE
La battaglia con i demoni è avvincente.	4,67	3,67	4,17
La difficoltà aumenta adeguatamente con il crescere della velocità del demone.	4,33	4,00	4,00
Ho trovato efficace il meccanismo scelto per lanciare gli incantesimi.	4,00	4,33	4,17
Il consumo del Mana era adeguato al danno procurato al demone.	4,00	4,00	4,00
Ho dovuto chiedere spesso l'energia al saggio.	4,00	3,67	3,83
Sono riuscito a capire in fretta come dosare il caricamento dell'incantesimo.	4,33	2,33	3,33
Ho apprezzato la meccanica relativa alla distruzione delle gemme al fine di produrre energia.	4,67	2,67	3,67
Ho trovato difficile distruggere la gemma perché cadeva spesso nel vuoto.	1,67	2,33	2,00

Tabella 6.6: Valutazione delle features del Cacciatore

Sono stati inoltre valutati, all'interno del questionario, la missione collaborativa, che non differiva per task richiesti a seconda del ruolo, e il tutorial. Questi, siccome sono uguali per i tre differenti gameplay, sono stati valutati per efficacia e interesse stimolato in base alla suddivisione under 30 e over 50.

I risultati della Tabella 6.8 confermano in parte ciò che è stato anticipato precedentemente quando si è parlato di usabilità, ovvero che è stata palesata la

DOMANDE	Anagrafic. COERENTI	Anagrafic. INCOERENTI	MEDIA GLOBALE
Ho trovato chiare le meccaniche del labirinto.	5,00	4,62	4,70
Ho avuto difficoltà nel piazzare il labirinto dove volevo.	2,00	2,00	2,00
Ho trovato chiare le meccaniche del memory.	5,00	4,25	4,40
Ho avuto difficoltà nel piazzare il memory dove volevo.	1,00	2,00	1,80
La meccanica relativa alla gestione delle risorse del gruppo mi è risultata chiara.	5,00	4,13	4,30
Mi è piaciuto gestire l'inventario.	5,00	4,00	4,20
Avevo sempre energia a disposizione quando i miei compagni ne avevano bisogno.	5,00	4,25	4,40
Inviare energia ai miei compagni è stato semplice.	5,00	3,87	4,10

Tabella 6.7: Valutazione delle features del Saggio

necessità di inserire la possibilità di rileggere il tutorial, specie dalla popolazione over 50 che afferma di aver letto attentamente il tutorial e successivamente aver avuto difficoltà a ricordare certe meccaniche. Valutazioni molto simili tra fasce anagrafiche sono state fatte sulla qualità del tutorial e la sua durata, mostrano valutazioni molto simili.

La missione collaborativa risulta essere stata apprezzata dai tester che complessivamente affermano che sia stata divertente e con un livello di difficoltà adeguato (Tabella 6.9). La differenza più rilevante tra le valutazioni degli under 30 e over 50 riguarda la domanda numero sei, dove gli over 50 affermano di aver trovato i suggerimenti fondamentali nella risoluzione dell'enigma, probabilmente perché

DOMANDE	UNDER 30	OVER 50	MEDIA GLOBALE
Il tutorial è spiegato in modo chiaro.	4,19	4,00	4,15
Il tutorial è esauriente e spiega tutte le funzionalità.	3,81	3,80	3,81
Il tutorial è troppo lungo e spiega anche funzionalità banali.	1,62	1,80	1,65
Le funzionalità spiegate sono molte, quindi subito dopo il tutorial non me le ricordavo tutte.	2,29	3,40	2,50
Vorrei aver avuto la possibilità di consultare nuovamente il tutorial.	3,05	3,60	3,15

Tabella 6.8: Valutazione dell'efficacia del tutorial

meno esperti di quella specifica tipologia di enigma. Generalmente la missione ha mostrato qualche falla nella coerenza dell'ambiente in AR condivisa per via della differente qualità delle videocamere dei dispositivi che comportava spesso incongruenze nel piano di riferimento condiviso creato tramite ARDK.

Sono state poste infine alcune domande generiche sul gameplay che sono state utili a comprenderne e valutarne l'efficacia. Dalle valutazioni dei tester, visibili in Tabella 6.10, risulta che la realtà aumentata abbia favorito un ottimo grado di immersività, destano più entusiasmo nelle generazioni più anziane: inoltre questi ultimi hanno manifestato una maggiore curiosità nei confronti dei collezionabili distribuiti per la mappa, mentre hanno percepito meno l'incremento della difficoltà del gameplay con i proseguire dell'esperienza. Riguardo i collezionabili, nella sezione commenti, è emersa la necessità di contestualizzarli all'interno della trama e di renderli più manipolabili: ad esempio è stato richiesto di poter leggere i libri, o interagire con manufatti. Questo in una versione finale dell'applicativo dovrebbe essere considerato al fine di incentivare ulteriormente l'esplorazione della città.

DOMANDE	UNDER 30	OVER 50	MEDIA GLOBALE
La missione collaborativa è stata divertente.	4,71	5,00	4,77
Il livello di difficoltà è adeguato.	4,38	3,80	4,27
La collaborazione sincrona rende l'esperienza più avvincente.	4,71	4,20	4,62
Se fossi stato da solo, avrei trovato la combinazione nello stesso tempo.	1,81	2,40	1,92
I suggerimenti sono risultati utili.	4,00	4,60	4,12
Senza suggerimenti non sarei riuscito a completare il task.	2,52	4,40	2,88
Nonostante i suggerimenti, abbiamo dovuto chiedere aiuto esterno per risolvere l'enigma.	1,67	2,00	1,73
Gli oggetti virtuali erano nello stesso posto per tutti i giocatori.	3,57	3,20	3,50
I dispositivi hanno comunicato correttamente durante tutta la missione.	3,24	4,60	3,50
Ho avuto difficoltà ad interagire con il baule.	2,62	3,60	2,81

Tabella 6.9: Valutazione della missione collaborativa in realtà aumentata

DOMANDE	UNDER 30	OVER 50	MEDIA GLOBALE
La realtà aumentata ha favorito l'immersività.	4,38	5,00	4,50
L'esperienza è stimolante anche durante le missioni degli altri giocatori.	3,43	4,40	3,62
Ho trovato interessanti i collezionabili e ho cercato di raccoglierne il più possibile.	3,52	4,40	3,69
Ho notato un incremento della difficoltà con l'avanzare del gioco.	2,62	1,40	2,38

Tabella 6.10: Valutazione delle funzionalità comuni nel gameplay

Capitolo 7

Conclusione e sviluppi futuri

Il presente elaborato si inserisce in un contesto in cui la Realtà Aumentata si sta affermando come tecnologia cardine in svariati ambiti applicativi e le esperienze che fanno uso dei dati GPS sono sempre più all'ordine del giorno. A seguito di un'attenta analisi dello stato dell'arte, approfondito dal punto di vista della letteratura ma anche prendendo in esame reali casi studio, si è delineata la necessità di porre le basi per una filosofia di sviluppo che, partendo sempre dalla valutazione delle tecnologie più all'avanguardia, abbia come obiettivo quello di fare dell'intergenerazionalità e della collaborazione due dei suoi punti chiave. Troppo spesso infatti si parla delle innovazioni come di qualcosa che allontana le generazioni più distanti e che ne rende difficoltoso il dialogo. Se da un lato, quindi, puntare allo sviluppo di un prodotto innovativo e originale ha rappresentato un obiettivo primario, dall'altro si è scelto di operare tenendo sempre a mente il target di riferimento (un target, appunto, *intergenerazionale*), sin dalle prime fasi di design. È stato quindi adottato un approccio trasversale, che si focalizzasse da una parte sull'usabilità e sull'accessibilità dell'applicativo, dall'altra sulla godibilità dell'esperienza per tutte le fasce d'età coinvolte. Attraverso i test e la valutazione dei risultati provenienti dai questionari sottoposti, è emerso che l'applicativo, pur essendo un prototipo, rispetta i requisiti di usabilità e pone le basi per la realizzazione di un'esperienza assolutamente godibile e stimolante, nonostante presenti notevoli margini di miglioramento.

A questo proposito, l'applicativo è stato concepito con l'obiettivo di renderlo il più possibile scalabile: la logica degli Scriptable Objects, frequentemente adottata, consente di avere un'interfaccia di controllo ad alto livello e totalmente personalizzabile che può essere utilizzata per ampliare o modificare l'esperienza (ad es. il setting di gioco, la storia o i contenuti). Allo stesso modo, l'organizzazione modulare del codice dà allo sviluppatore la possibilità sia di comprenderne facilmente le sezioni esistenti, sia di aggiungere nuove funzionalità o i propri contenuti virtuali.

Per quel che riguarda invece aspetti più legati allo sviluppo di un prodotto destinabile al mercato, sarebbe sicuramente interessante dare la possibilità a imprenditori

ed esercenti di pubblicizzarsi all'interno dell'esperienza, implementando un sistema di notifiche che, sulla base dei dati GPS, avvisi l'utente relativamente alla presenza di attività locali nelle immediate vicinanze.

Bibliografia

- [1] *Marco Pucci*. <https://www.marcopucci.it/storia-della-realta-aumentata/> (cit. a p. 1).
- [2] *Pikkart*. https://www.pikkart.com/servizi/notizie/notizie_fase02.aspx?ID=3464# (cit. a p. 1).
- [3] *UniPV*. https://vision.unipv.it/corsi/InformationTechnology/IT_AR.pdf (cit. a p. 2).
- [4] *Datafloq*. <https://datafloq.com/read/when-augmented-virtual-reality-hit-theater/> (cit. a p. 3).
- [5] *Frameblog*. <https://frameblog.unibo.it/index.php/2021/06/09/serious-games-il-videogioco-al-servizio-della-cultura/> (cit. a p. 4).
- [6] *Wikipedia - Serious Games*. <https://frameblog.unibo.it/index.php/2021/06/09/serious-games-il-videogioco-al-servizio-della-cultura/> (cit. a p. 5).
- [7] *Projectfun - Guida ai serious games*. <https://www.projectfun.it/serious-game/guida/> (cit. a p. 5).
- [8] *Ita Sciences*. <https://ita.sciences-world.com/intergenerational-gaming-helps-families-connect-72921> (cit. a p. 6).
- [9] Chek Tien Tan e Donny Soh. «Augmented Reality Games: A Review». In: apr. 2011, pp. 212–8 (cit. a p. 7).
- [10] Diego Avila Pesantez, Luis Rivera, Leticia Vaca-Cardenas, Stefano Aguayo e Lourdes Zuniga. «Towards the improvement of ADHD children through augmented reality serious games: Preliminary results». In: apr. 2018, pp. 843–848. DOI: 10.1109/EDUCON.2018.8363318 (cit. a p. 8).
- [11] Hong-Wei Wong. «Leap Motion-Enhanced Augmented Reality Video Games to Encourage Intergenerational Engagement and Wrist Exercise». In: *International Journal of Advanced Trends in Computer Science and Engineering* 9 (giu. 2020), pp. 78–82. DOI: 10.30534/ijatcse/2020/1191.32020 (cit. a p. 8).

- [12] Ramon Barraza, Vianey Sánchez e Osslán Vergara. «A Pilot Study on the Use of Mobile Augmented Reality for Interactive Experimentation in Quadratic Equations». In: *Mathematical Problems in Engineering* 2015 (feb. 2015), pp. 1–13. DOI: 10.1155/2015/946034 (cit. alle pp. 9, 10).
- [13] Jingya Li, Erik Spek, Loe Feijs, Feng Wang e Jun Hu. «Augmented Reality Games for Learning: A Literature Review». In: mag. 2017, pp. 612–626. ISBN: 978-3-319-58696-0. DOI: 10.1007/978-3-319-58697-7_46 (cit. a p. 9).
- [14] Santiago Schez-Sobrino, David Vallejo, Carlos Glez-Morcillo, Miguel Redondo e Jose Castro-Schez. «RoboTIC: A serious game based on augmented reality for learning programming». In: *Multimedia Tools and Applications* 79 (dic. 2020). DOI: 10.1007/s11042-020-09202-z (cit. a p. 9).
- [15] Costas Boletsis e Simon Mccallum. «The Smartkuber Case Study: Lessons Learned from the Development of an Augmented Reality Serious Game for Cognitive Screening». In: vol. 10324. Giu. 2017, pp. 457–472. ISBN: 978-3-319-60921-8. DOI: 10.1007/978-3-319-60922-5_36 (cit. a p. 10).
- [16] Ziad Nasreddine, Natalie Phillips, Valérie Bédirian, Simon Charbonneau, Victor Whitehead, Isabelle Collin, Jeffrey Cummings e Howard Chertkow. «The Montreal Cognitive Assessment, MoCA: A Brief Screening Tool For Mild Cognitive Impairment». In: *Journal of the American Geriatrics Society* 53 (mag. 2005), pp. 695–9. DOI: 10.1111/j.1532-5415.2005.53221.x (cit. a p. 11).
- [17] Lorenz Lehmann. «Location-based Mobile Games». In: 2012 (cit. alle pp. 11–13, 63–65).
- [18] Sebastian Matyas. «Playful Geospatial Data Acquisition by Locaton-based Gaming Communities». In: *IJVR* 6 (gen. 2007), pp. 1–10 (cit. a p. 14).
- [19] Michele Chang e Elizabeth Goodman. «FIASCO: Game Interface for Location-Based Play». In: *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*. DIS '04. Cambridge, MA, USA: Association for Computing Machinery, 2004, pp. 329–332. ISBN: 1581137877. DOI: 10.1145/1013115.1013172. URL: <https://doi.org/10.1145/1013115.1013172> (cit. a p. 14).
- [20] *Generations Working Together*. <https://generationsworkingtogether.org/downloads/5bebf57b90d25-Intergenerational-Activity-Guide-2018-St-Monica-Trust.pdf> (cit. a p. 18).
- [21] Simone Hausknecht, Carman Neustaedter e David Kaufman. «Blurring the Lines of Age: Intergenerational Collaboration in Alternate Reality Games». In: ott. 2017, pp. 47–64. ISBN: 978-3-319-41795-0. DOI: 10.1007/978-3-319-41797-4_4 (cit. a p. 18).

- [22] Khoo Tat e Adrian Cheok. «Age Invaders: Inter-generational Mixed Reality Family Game». In: *IJVR* 5 (gen. 2006), pp. 45–50. DOI: 10.20870/IJVR.2006.5.2.2688 (cit. alle pp. 18, 20, 27).
- [23] *World Population Ageing 2020 Highlights*. https://www.un.org/development/desa/pd/sites/www.un.org.development.desa.pd/files/files/documents/2020/Sep/un_pop_2020_pf_ageing_10_key_messages.pdf (cit. a p. 18).
- [24] Kian Razavi Satvati e Mojtaba Vahidi-Asl. «Looking at Intergenerational Game Design through the Lens of Game Genres». In: feb. 2021 (cit. alle pp. 18, 21).
- [25] Mary Hummert, Teri Garstka, Jaye Atkinson e Sharon Strahm. «Stereotypes of the Elderly Held by Young, Middle-Aged, and Elderly Adults». In: *Journal of gerontology* 49 (ott. 1994), P240–9. DOI: 10.1093/geronj/49.5.P240 (cit. a p. 18).
- [26] Bob De Schutter e Vero Vanden Abeele. «Towards a Gerontoludic Manifesto». In: *Anthropology Aging* 36 (nov. 2015), p. 112. DOI: 10.5195/aa.2015.104 (cit. a p. 18).
- [27] Damien Rompapas, Christian Sandor, Alexander Plopski, Daniel Saakes, Joon Gi Shin, Takafumi Taketomi e Hirokazu Kato. «Towards Large Scale High Fidelity Collaborative Augmented Reality». In: *Computers Graphics* 84 (ago. 2019). DOI: 10.1016/j.cag.2019.08.007 (cit. a p. 22).
- [28] Almar van der Stappen, Yunjie Liu, Jiangxue Xu, Xiaoyu Yu, Jingya Li e Erik Spek. «MathBuilder: A Collaborative AR Math Game for Elementary School Students». In: ott. 2019, pp. 731–738. ISBN: 978-1-4503-6871-1. DOI: 10.1145/3341215.3356295 (cit. alle pp. 24, 27).
- [29] Inma García-Pereira, Cristina Portalés, Jesús Gimeno e Sergio Casas-Yrurzum. «A collaborative augmented reality annotation tool for the inspection of prefabricated buildings». In: *Multimedia Tools and Applications* 79 (mar. 2020), pp. 1–19. DOI: 10.1007/s11042-019-08419-x (cit. a p. 25).
- [30] Mickael Sereno, Xiyao Wang, Lonni Besançon, Michael McGuffin e Tobias Isenberg. «Collaborative Work in Augmented Reality: A Survey». In: *IEEE Transactions on Visualization and Computer Graphics* PP (ott. 2020), pp. 1–1. DOI: 10.1109/TVCG.2020.3032761 (cit. a p. 27).
- [31] Reiner Wichert e Matthias Finke. «An Event-Based Communication Mechanism to Realize a Mobile Collaborative AR Environment». In: (gen. 2003) (cit. a p. 27).
- [32] *Unity*. <https://unity.com/> (cit. a p. 39).
- [33] *Lightship ARDK*. <https://lightship.dev/products/ardk/> (cit. a p. 42).

- [34] *Mapbox*. <https://www.mapbox.com> (cit. a p. 45).
- [35] *Mirror Networking*. <https://mirror-networking.com/> (cit. a p. 47).
- [36] *Firebase*. <https://firebase.google.com/> (cit. a p. 51).
- [37] Max Birk, Maximilian Friehs e Regan Mandryk. «Age-Based Preferences and Player Experience: A Crowdsourced Cross-sectional Study». In: ott. 2017, pp. 157–170. DOI: 10.1145/3116595.3116608 (cit. a p. 54).
- [38] Bob De Schutter. «Never Too Old to Play: The Appeal of Digital Games to an Older Audience». In: *Games and Culture - Game Cult* 6 (mar. 2011), pp. 155–170. DOI: 10.1177/1555412010364978 (cit. a p. 54).
- [39] Abdullah Mahmud, Omar Mubin, Suleman Shahid e Jean-bernard Martens. «Designing social games for children and older adults: Two related case studies». In: *Entertainment Computing* 1 (dic. 2010), pp. 147–156. DOI: 10.1016/j.entcom.2010.09.001 (cit. a p. 54).
- [40] *Desa Development*. <http://www.comune.torino.it/arredourbano/pianocolore/pietrelocali/index.shtml> (cit. a p. 62).
- [41] Celia Pearce. «The Truth About Baby Boomer GamersA Study of Over-Forty Computer Game Players». In: *Games and Culture - Game Cult* 3 (feb. 2008), pp. 142–174. DOI: 10.1177/1555412008314132 (cit. a p. 68).