

TESI DI LAUREA MAGISTRALE

**Meccaniche di gioco in realtà aumentata per la valorizzazione del
patrimonio culturale**

Relatore:

Umberto Castellani

Candidato:

De Vincenzi Marcello

Matricola VR474723

Indice

1	Introduzione	5
2	Realtà Aumentata: Tipologie e funzionamento	7
2.1	Realtà aumentata Marker Based	7
2.1.1	Realtà aumentata Marker Based: facilità di implementazione e usabilità	8
2.1.2	Limitazioni della realtà aumentata Marker Based: calibra- zione e vincoli tecnici	8
2.2	Realtà Aumentata Model Based	8
2.2.1	Vantaggi e svantaggi della tecnologia Model Based AR	9
2.3	Realtà Aumentata Markerless	10
2.3.1	Location Based AR	10
2.3.2	Overlay AR	11
2.3.3	Countur Based AR	11
3	RA: dalla Computer Vision alla Computer Grafica	13
3.1	Computer vision: definizione e metodi	13
3.1.1	Compiti principali della computer vision	13
3.2	Dalla computer vision alla realtà aumentata	14
3.3	La camera	15
3.3.1	Parametri estrinseci della camera	15
3.3.2	Parametri intrinseci della camera	16
3.4	Stima della posa della camera	17
3.4.1	Il metodo di Fiore per la stima della posa	17
4	Game Design e videogiochi	21
4.1	Definizione di un videogioco	21
4.2	Elementi chiave di un videogioco	22
4.3	Struttura di un videogioco	24
4.3.1	Estetica	25
4.3.2	Storia	25
4.3.3	Tecnologia	25
4.3.4	Meccaniche	26
4.4	Meccaniche di gioco	26
4.4.1	Spazio	26
4.4.2	Oggetti, attributi e stati	27
4.4.3	Azioni	27
4.4.4	Regole	28
4.4.5	Abilità	28

4.4.6	Chance	29
5	Studio delle meccaniche di gioco in realtà aumentata	31
5.1	Lancio di un oggetto con fisica	31
5.1.1	Versione Iniziale, lancio senza gravità	32
5.1.2	Implementazione della demo	32
5.1.3	Inserimento nel contesto del patrimonio culturale	33
5.1.4	Lancio con forza di gravità	33
5.1.5	Implementazione della demo	34
5.1.6	Studio delle meccaniche presenti	35
5.1.7	Inserimento nel contesto del patrimonio culturale	35
5.2	View Matching	35
5.2.1	Implementazione della demo	36
5.2.2	Studio delle meccaniche presenti	37
5.2.3	Inserimento nel contesto del patrimonio culturale	37
5.3	Ritagliare o Scontornare	37
5.3.1	Implementazione della Demo	38
5.3.2	Studio delle meccaniche presenti	39
5.3.3	Inserimento nel contesto del patrimonio culturale	40
5.4	Movimento di un avatar	40
5.4.1	Movimento tramite tasti	41
5.4.2	Implementazione della Demo	41
5.4.3	Avatar corrispondente al centro dello schermo	41
5.4.4	Implementazione della Demo	42
5.4.5	Studio delle meccaniche presenti	42
5.4.6	Inserimento nel contesto del patrimonio culturale	43
6	Valutazione delle meccaniche e User Study	45
6.1	Movimento di un avatar	46
6.2	View Matching	47
6.3	Lancio con fisica	49
6.4	Lancio senza fisica	50
6.5	Scontornare	52
7	Conclusioni	55
	Bibliografia & Sitografia	57

Capitolo 1

Introduzione

Nel vasto panorama dell'intrattenimento digitale, la Realtà Aumentata (RA) è recentemente emersa come tecnologia, trasformando il modo in cui interagiamo con il mondo circostante e aprendo le porte a nuovi tipi di esperienza.

Se da un lato il contesto della realtà aumentata apre le porte ad un nuovo mondo di possibilità ludiche e di conseguenza possibili giochi da implementare, dall'altro fa sorgere anche una serie di problemi e considerazioni da adottare per implementarli. Infatti, nel contesto della realtà aumentata, nello specifico in quella applicata ai dispositivi mobili, l'utente si trova a dover interagire con uno spazio misto attraverso la telecamera del proprio telefono e non attraverso uno schermo come accade nei casi classici di videogiochi.

Questo, come avviene per l'ambito del mobile gaming, richiede ai designer di studiare un modo per implementare le loro idee in forma di meccanica in modo che questa possa risultare fruibile anche in questo contesto.

Un altro problema riscontrabile nel contesto della realtà aumentata è quello della tecnologia, infatti il suo particolare funzionamento, ovvero il proiettare dei modelli sintetici su una scena reale, richiede che la tecnologia che la supporta, ovvero quella che si occupa di riconoscere piani, superfici, o addirittura oggetti nello scenario, funzioni adeguatamente in modo che non si vadano a creare dei fenomeni. L'obiettivo di questa tesi di Laurea è quello di analizzare alcune meccaniche di gioco ed effettuare uno user study per capire come queste possano essere implementate in realtà aumentata, prestando particolare attenzione a come quest'ultime possano andare ad arricchire il contesto del patrimonio culturale, nello specifico quello della città di Verona.

Capitolo 2

Realtà Aumentata: Tipologie e funzionamento

Definizione Con il termine realtà aumentata (abbreviato RA o AR dall'inglese *augmented reality*), si intende l'arricchimento della percezione sensoriale umana mediante informazioni, in genere convogliate elettronicamente, che non sarebbero percepibili con i cinque sensi. Essa si differenzia dalla realtà virtuale nel fatto che in questo caso la realtà già osservata dall'utente viene "aumentata" attraverso degli stimoli elettronici, generalmente visivi[1].

Tipi di realtà aumentata Come ogni altro tipo di tecnologia, sono presenti svariati modi di implementarla. Tra i molti tipi di classificazione disponibili[2], ne adottiamo una che distingue tre principali macro tipi, che differiscono per come essa andrà attivata e come riesce a posizionare i propri oggetti sintetici nella scena:

- realtà aumentata Marker Based
- realtà aumentata Model Based
- realtà aumentata Markerless

2.1 Realtà aumentata Marker Based

Questo tipo di realtà aumentata fa uso di codici visivi, detti marker o marcatori, per posizionare gli oggetti sintetici all'interno della scena. Questi marcatori sono spesso formati da pattern specifici che permettono al motore di realtà aumentata di capire dove posizionare gli oggetti sintetici in quest'ultima.

Di solito, un processo di realtà aumentata basato su marcatori segue una serie di fasi principali, le quali rimangono costanti indipendentemente dal tipo di software impiegato:

- riconoscimento del marcatore: in questa fase, la fotocamera del dispositivo deve riconoscere il marcatore presente nella scena.
- posizionamento dell'oggetto: qui, il software deve stimare la posa del marcatore sulla scena e quindi computare come il modello sintetico andrà a posizionarsi sulla scena reale.

La parte di stima della posa del marcatore e della scena reale è il processo di determinare con precisione la posizione e l'orientamento degli oggetti virtuali all'interno dello spazio fisico circostante. Questo viene solitamente fatto attraverso tecniche come la localizzazione dei marcatori, il tracciamento dei punti chiave nell'ambiente o l'uso di sensori come la fotocamera e gli accelerometri.

2.1.1 Realtà aumentata Marker Based: facilità di implementazione e usabilità

La realtà aumentata basata su marker si presenta come una soluzione più accessibile e pratica rispetto alla sua controparte markerless. La presenza di punti di ancoraggio fissi per gli oggetti virtuali nella scena rende significativamente più agevole lo sviluppo del software. Questo approccio fornisce un punto di riferimento stabile per sovrapporre gli elementi virtuali, semplificando notevolmente la gestione delle interazioni utente-oggetto.

Inoltre, la facilità di utilizzo è un vantaggio significativo del modello marker based. Gli utenti possono interagire direttamente con gli oggetti sintetici tramite il marker, stabilendo una connessione intuitiva tra l'esperienza fisica e quella virtuale. Questo approccio contribuisce a rendere l'interazione più immediata e comprensibile per il pubblico, soprattutto in contesti in cui la facilità d'uso è una priorità.

Inoltre, sono disponibili numerosi framework e strumenti di sviluppo specifici per la AR marker-based, fatto che semplifica ulteriormente il processo di creazione di applicazioni AR. Gli sviluppatori possono infatti usufruire di librerie e API dedicate che semplificano la gestione dei marker e la sovrapposizione degli oggetti virtuali, riducendo così i tempi di sviluppo e migliorando l'efficienza complessiva del processo di creazione di contenuti AR.

2.1.2 Limitazioni della realtà aumentata Marker Based: calibrazione e vincoli tecnici

Nonostante i suoi vantaggi, la realtà aumentata marker based presenta alcune limitazioni che ne influenzano l'efficacia complessiva. La sua operatività è circoscritta al riconoscimento e alla proiezione di oggetti nelle immediate vicinanze del marker, comportando la necessità di calibrare ogni marcatore separatamente. Questo processo può risultare oneroso, specialmente in scenari in cui la presenza di numerosi marcatori richiede un'attenzione dettagliata.

Ulteriori difficoltà sorgono nel caso in cui gli utenti dispongano di telecamere non ottimizzate per il modello marker based. La compatibilità tecnica è quindi una sfida, richiedendo una selezione attenta delle apparecchiature per garantire il corretto funzionamento del sistema.

2.2 Realtà Aumentata Model Based

Questo tipo di realtà aumentata risulta molto simile alla soluzione marker based: in questo caso invece di mappare gli oggetti sintetici tramite un pattern predefinito essi vengono mappati sopra un oggetto reale presente nella scena. Il modello su

cui essi andranno mappati dovrà essere quindi scelto in precedenza, e deve essere riconosciuto dal dispositivo.

Quindi, in questo caso non c'è un vero target specifico fisico che il dispositivo deve riconoscere, quanto piuttosto un oggetto, che può essere generico, come ad esempio un cestino da basket, arrivando a poter riconoscere in alcuni casi anche un'area intera.

Affinché il riconoscimento dell'oggetto venga eseguito, nel dispositivo deve essere fatto un training specifico su quest'ultimo tramite un motore apposito.

Ai fini di questo studio, è stata utilizzata questo tipo di realtà aumentata per ricreare le meccaniche di gioco nel contesto AR. In particolare, i modelli usati nella ricerca hanno una rilevanza nell'ambito dei beni culturali, tra cui figurano Porta Nuova o alcuni tratti importanti delle mura presenti a Verona, con lo scopo di poter ispirare un percorso didattico che fa uso delle meccaniche di gioco studiate per valorizzarle. Per la loro realizzazione, è stato utilizzato il programma di fotogrammetria 3D Zephyr per acquisire il modello 3D, mentre per il suo addestramento si è utilizzato il motore di Vuforia.



Figura 2.1: Logo di Vuforia



Figura 2.2: Logo di 3D Zephyr

2.2.1 Vantaggi e svantaggi della tecnologia Model Based AR

La realtà aumentata basata su modelli può essere un approccio innovativo con vantaggi significativi in varie applicazioni. Uno dei principali benefici di questa pratica risiede nella sua capacità di fornire un'esperienza utente più accurata e coinvolgente attraverso la sovrapposizione degli oggetti sintetici a quelli realmente presenti nella realtà.

Questa metodologia consente un livello di interazione avanzato, in cui gli utenti possono manipolare oggetti virtuali e ottenere informazioni dettagliate sui modelli visualizzati. Inoltre, la registrazione precisa e il tracking continuo migliorano l'allineamento del modello con l'ambiente fisico, garantendo una visualizzazione coerente e realistica.

Questo tipo di AR trova applicazioni significative in settori come la formazione, la progettazione industriale e la manutenzione, contribuendo a migliorare l'efficienza

operativa e la comprensione degli utenti.

D'altro canto, nonostante il potenziale miglioramento in termini di interattività e fornitura di informazioni rispetto alla tecnologia basata su marcatori, la realtà aumentata basata su modelli presenta una necessità di generare un database per l'addestramento del dispositivo. Questo processo è essenziale affinché gli oggetti fisici possano essere accuratamente riconosciuti e tracciati nell'ambiente. La creazione e la gestione di un database dettagliato richiedono risorse sostanziali, tra cui il lavoro di modellazione 3D e l'implementazione di algoritmi di addestramento. Pertanto, sebbene la tecnologia basata su modelli offra vantaggi distinti, la complessità associata alla creazione di database può rappresentare un aspetto da considerare nella valutazione della sua fattibilità e adozione.

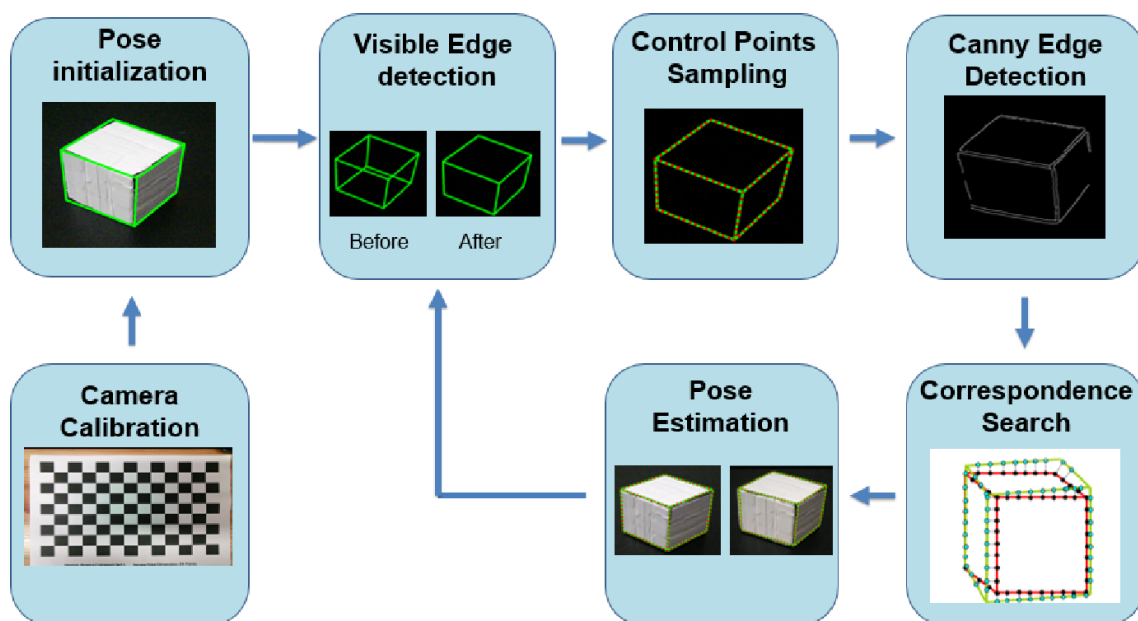


Figura 2.3: Schema dei vari passaggi utilizzati nella realtà aumentata Model Based[3].

2.3 Realtà Aumentata Markerless

Questo tipo di realtà aumentata non fa uso di Marker per posizionare gli oggetti sintetici della scena, ma offre più libertà all'utente e gli permette di posizionarli liberamente nelle scene reali. Essa fa molto più affidamento a strumenti del telefono come camera, sensori, e processore rispetto alla sua controparte basata sui marcatori. Molto spesso questo tipo di tecnologia richiede all'utente di avere a disposizione un piano solido che deve poi essere identificato dal sistema.

L'implementazione può avvenire in diverse modalità, tra queste principalmente distinguiamo: location based, overlay e contour based.

2.3.1 Location Based AR

Chiamata anche Geo-based oppure position-based, essa si basa sui sistemi di geolocalizzazione in tempo reale, che si trovano nella maggior parte di tutti i telefoni

venduti, e sfrutta questi dati per capire la posizione dell'utente rispetto ai punti di riferimento inseriti nel sistema, dove saranno proiettati gli oggetti sintetici.

In questo caso quindi l'ostacolo principale è quello di avere un algoritmo adeguato che sappia effettuare la proiezione degli oggetti sintetici nel luogo giusto avendo a disposizione la posizione del telefono rispetto a quella del punto di interesse.

Un possibile esempio di implementazione di questo metodo di realtà aumentata sono i giochi interattivi educativi, quelli più ludici come Pokèmon Go, oltre che quelli per fini più pratici anche la navigazione di Google Maps, che da poco ha implementato una funzione che utilizza la realtà aumentata[4].

2.3.2 Overlay AR

La realtà aumentata di tipo overlay, chiamata anche superimposition AR, rimpiazza la vista reale di un oggetto inquadrato con la sua versione sintetica, in modo che esso possa mostrare le proprie informazioni in tempo reale o in modo che l'utente vi possa interagire.

Un esempio pratico di questo tipo di realtà aumentata è il cosiddetto *Nextech Configuration Tool*: questo sistema, prodotto dalla azienda Nextech è presente in alcune applicazioni di shopping online di mobili, e permette all'utente di visualizzare questi prodotti in tempo reale proiettandoli nella loro abitazione, ed eventualmente modificarli tramite un menu presente in sovraimpressione allo schermo, cambiandone per esempio il colore o la taglia.

2.3.3 Countur Based AR

Questo tipo di realtà aumentata, come si può intuire dal nome, si basa sull'utilizzo di una telecamera apposita per riconoscere i contorni degli oggetti vicini e farli risaltare attraverso delle linee di contorno[5].

Ciò risulta utile in quanto permette agli utenti di vedere più chiaramente i contorni degli oggetti che hanno di fronte.

Può essere utilizzata ad esempio nel sistema di guida assistiti, in quanto permette ai guidatori di poter viaggiare più facilmente in strada nelle condizioni di scarsa visibilità.

Capitolo 3

RA: dalla Computer Vision alla Computer Grafica

3.1 Computer vision: definizione e metodi

La computer vision (in italiano visione computazionale) è un campo dell'informatica che si occupa dello sviluppo di sistemi in grado di interpretare e comprendere le informazioni visive provenienti dal mondo reale. L'obiettivo della computer vision è quello di consentire alle macchine di acquisire, analizzare e interpretare dati visivi in modo simile a come lo farebbe un essere umano[6]. Le attività della visione computazionale possono includere il riconoscimento di oggetti, la segmentazione dell'immagine, l'analisi del movimento, il riconoscimento facciale, la stima della profondità e molte altre.

Anche la tipologia di dati analizzati in questo ambito dell'informatica è molto eterogenea: sequenze video, prospettive da diverse telecamere, dati multi-dimensionali da scanner 3D, nuvole di punti tridimensionali provenienti da sensori LiDaR e immagini acquisite da dispositivi di scansione medica.

Le possibili applicazioni di queste metodologie includono: la medicina, dove si insegna alle macchine come diagnosticare un paziente attraverso le immagini delle scansioni, oppure l'ambito militare, dove la visione computazionale fa distinguere le macchine gli obiettivi nemici da colpire nel caso ad esempio di sistemi missilistici automatici. Altre applicazioni importanti della visione computazionale sono la machine vision, dove le informazioni date degli stimoli visivi vengono utilizzate per supportare una catena di produzione, per esempio come controllo qualità, dove i prodotti vengono visivamente ispezionati dalle macchine per trovare dei difetti di fabbrica, ma anche sistemi di produzione agricola che rimuovono il cibo guasto dal resto tramite un processo chiamato *optical sorting*. Questa branca dell'informatica viene utilizzata anche in altri settori che includono la sorveglianza, il camera tracking, l'identificazione di oggetti e l'ispezione automatica.

3.1.1 Compiti principali della computer vision

La visione computazionale si distingue per la sua capacità di estrarre e interpretare informazioni visive da immagini o dati visivi, cercando di replicare il processo di comprensione umana. In questo contesto, la computer vision si occupa di svariati task o compiti in diversi ambiti applicativi.

La *recognition* costituisce il primo pilastro della visione computazionale, impegnandosi nell'identificazione e riconoscimento di oggetti, attività o caratteristiche presenti nei dati visivi di input. Tra le diverse task all'interno di questa categoria, ne emergono alcune chiave:

- **Classificazione di immagini:** Questa task implica la capacità di analizzare il contenuto di un'immagine e di assegnarle un'etichetta o una categoria specifica. Algoritmi di classificazione di immagini sono fondamentali in applicazioni come la catalogazione automatica di foto, la diagnostica medica basata su immagini, e i sistemi di riconoscimento di oggetti.
- **Object detection:** L'object detection si concentra sul riconoscimento di specifici oggetti all'interno di una scena, determinandone anche la posizione attraverso bounding boxes, ovvero un rettangolo posto sulla scena dove all'interno vi sono i punti di interesse.
- **Image segmentation:** La segmentazione dell'immagine consiste nella suddivisione di un'immagine in regioni o segmenti, ciascuno associato a una specifica categoria o caratteristica. Questa task è essenziale in contesti come l'analisi medica, la mappatura ambientale e la comprensione dettagliata di scene complesse.
- **Editing di immagini:** La task di editing di immagini si concentra sulla manipolazione delle immagini, permettendo la modifica di alcuni loro aspetti come contrasto, luminosità e colore.

Per effettuare la task di object recognition si usano alcuni metodi come la stima della posa della camera.

Gli altri ambiti principali in cui viene applicata la computer vision sono quelli della analisi del movimento, che consiste nel saper stimare la velocità di tutti i punti in un'immagine di una sequenza video, oppure della camera stessa. Principalmente, questi ambiti sono l'egomotion, che determina il movimento rigido della camera durante la sequenza video, il tracking, che invece si occupa di seguire i movimenti di una serie ridotta di oggetti lungo la sequenza video, o l'optical flow, che si occupa, per ogni punto dell'immagine, come esso si muove rispetto al piano immagine, cioè di come esso si muove combinato col movimento della camera.

Gli ultimi due ambiti principali di questa materia sono la ricostruzione delle scene, ovvero la computazione di un modello 3D di una scena osservata, di cui fa parte il software già usato in questo studio ovvero *3D Zephyr* e la reastaurazione di un'immagine, cioè la capacità di recuperare l'immagine originale dopo che essa è stata degradata da alcuni fattori esterni dovuti alla camera, come per esempio il rumore.

3.2 Dalla computer vision alla realtà aumentata

La realtà aumentata ha radici negli ambiti della computer vision e della computer grafica. Questo processo infatti è realizzabile solo attraverso processi che coinvolgono queste ultime, quali la cosiddetta stima della posa.

La stima della posa è uno dei principali problemi della computer vision: esso consiste nel saper determinare la posizione di un oggetto o di un sistema di coordinate

in uno spazio tridimensionale.

Per fare in modo che questa avvenga, va prima effettuata la cosiddetta calibrazione della camera: ovvero determinare i parametri estrinseci ed intrinseci di una camera, in modo da compensare le distorsioni ottiche e geometriche introdotte dalla fotocamera stessa, consentendo di ottenere misurazioni accurate e una rappresentazione corretta della scena. Questo è particolarmente importante quando si utilizzano le immagini per misure quantitative o per applicazioni di realtà aumentata come nel nostro caso.

3.3 La camera

In questa sezione vengono elencati i parametri di una camera, elemento fondamentale nella computer vision.

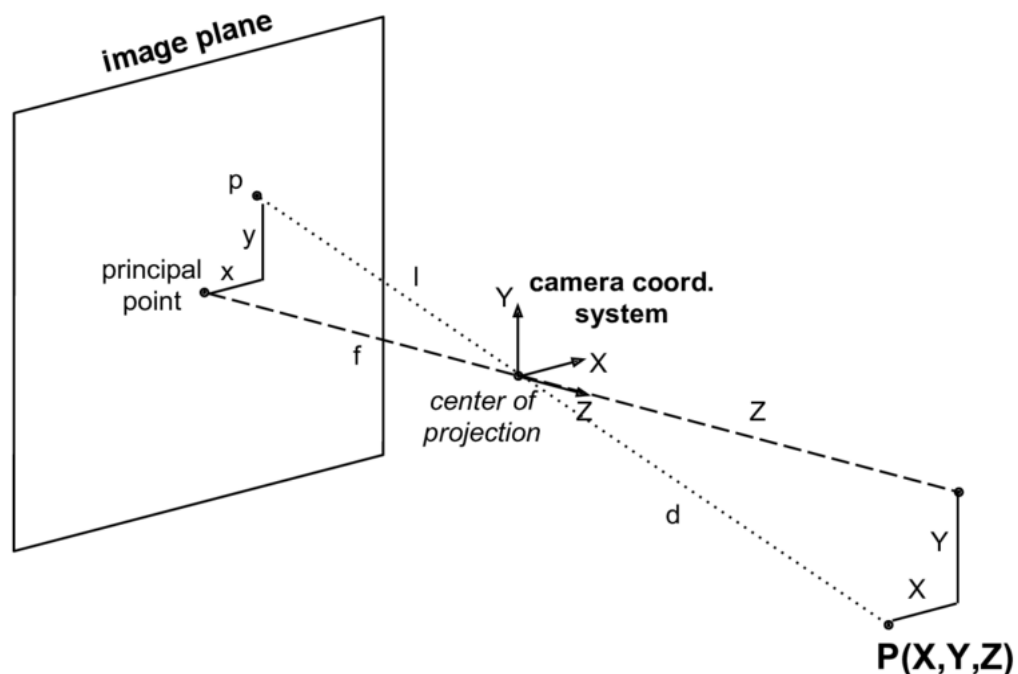


Figura 3.1: Immagine rappresentate il modello di una camera stenopeica[7].

3.3.1 Parametri estrinseci della camera

Una camera è composta da due tipi di parametri, quelli estrinseci, che dipendono dalla foto scattata, ed i parametri intrinseci, che invece dipendono da come la camera è fatta fisicamente.

I parametri estrinseci della camera descrivono dove la camera è posizionata nel mondo e in che direzione sta puntando.

In particolare, rappresentano una matrice di trasformazione che mappa punti dal sistema di riferimento mondo a punti nel sistema di riferimento camera. Questi parametri sono identificati dalla matrice

$$M = [R|T]$$

dove R (3x3) rappresenta la matrice di rotazione e T (3x1) il vettore di traslazione. Una matrice di trasformazione di questo tipo può essere vista anche come un cambio di sistema di riferimento, in cui le colonne rappresentano rispettivamente la direzione degli assi X, Y, Z e l'origine di questi ultimi visti dal nuovo sistema di arrivo rispetto a quello partenza. Ad esempio, nella matrice di trasformazione

$$transform = \begin{bmatrix} a_x & a_y & a_z & d_1 \\ b_x & b_y & b_z & d_2 \\ c_x & c_y & c_z & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

le colonne $[a_x, b_x, c_x]$, $[a_y, b_y, c_y]$, $[a_z, b_z, c_z]$ rappresentano rispettivamente gli assi X, Y e Z, ovvero la matrice di rotazione. Mentre l'ultima colonna $[d_1, d_2, d_3]$ rappresenta il punto di origine del sistema di partenza, ovvero il vettore di traslazione.

La matrice di rotazione della camera ha dimensione 3x3, ovvero è della forma:

$$R = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{bmatrix}$$

Mentre il vettore di traslazione è della forma:

$$T = [t_x \quad t_y \quad t_z]$$

3.3.2 Parametri intrinseci della camera

Una camera possiede anche dei parametri intrinseci, che dipendono da come essa è fatta fisicamente. Essi sono:

- Distanza focale (f): indica la distanza dalla lente della fotocamera al piano immagine, espressa in millimetri o pixel.
- Punto principale (c_x, c_y): le coordinate del punto principale, che rappresentano il punto di intersezione degli assi ottici della fotocamera con il piano immagine. Esse definiscono il centro dell'immagine.
- Coefficiente di distorsione radiale (k_1, k_2, k_3): i coefficienti che descrivono la distorsione radiale della lente della fotocamera.
- Coefficiente di distorsione tangenziale (p_1, p_2): i coefficienti che descrivono la distorsione tangenziale introdotta dalla posizione decentrata delle lenti.

Con questi parametri è possibile costruire la matrice degli intrinseci, ovvero una matrice di trasformazione che converte punti dal sistema di riferimento camera al sistema di riferimento pixel [8]. Ha dimensione 3x3 ed è nella forma:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Dove f_x ed f_y sono la distanza focale della camera in pixel nelle due dimensioni x e y, mentre c_x e c_y rappresentano la posizione del "principal point" sui medesimi assi.

3.4 Stima della posa della camera

La stima della posa di una camera è uno dei principali argomenti di cui si occupa la computer vision, esso è il processo di determinare la posizione e l'orientamento della fotocamera rispetto a un sistema di coordinate esterno. Questo processo è cruciale in molte applicazioni di computer vision, realtà aumentata, realtà virtuale, robotica e navigazione autonoma. Ci sono diversi modi per stimare la posa della fotocamera:

- Utilizzo di marcatori noti: si posizionano marcatori noti, come pattern a scacchiera o codici QR, nell'ambiente e si catturano immagini della scena con la fotocamera. Utilizzando le posizioni note dei marcatori e le corrispondenti posizioni rilevate nell'immagine, è possibile calcolare la posa della fotocamera rispetto al sistema di coordinate del mondo esterno[9].
- Corrispondenza di punti chiave: si trovano le corrispondenze tra caratteristiche distinte rilevate nell'immagine e punti noti nell'ambiente. Questo può essere fatto utilizzando algoritmi di corrispondenza di punti chiave come SIFT, SURF o ORB. Con le corrispondenze, è possibile stimare la posa della fotocamera attraverso tecniche di geometria epipolare.
- Sistemi di localizzazione basati su sensori: alcuni sistemi utilizzano una combinazione di sensori come giroscopi, accelerometri e magnetometri per stimare il movimento della fotocamera rispetto all'ambiente circostante. Questi sensori forniscono informazioni sulla velocità angolare, l'accelerazione lineare e la direzione magnetica che possono essere integrate per ottenere una stima della posa della fotocamera[10].
- Metodi basati sulla struttura del movimento: Questi metodi sfruttano le relazioni geometriche tra i punti nell'immagine per stimare la posa della fotocamera. Un esempio comune è l'utilizzo della geometria epipolare e della triangolazione per calcolare la posa[11].
- Approcci basati sul Deep Learning: dove si usano metodi di regressione lineare basati su training per riuscire ad elaborare la posa della camera basandosi sulle immagini date[12].

3.4.1 Il metodo di Fiore per la stima della posa

Per operare la realtà aumentata necessita di un motore in grado di capire come è posizionata la camera dato un oggetto reale, dato un set di punti tridimensionali e la loro proiezione sopra un'immagine, ovvero quella inquadrata.

Uno dei metodi per realizzare questa procedura è attraverso il metodo di Fiore [13], questo metodo infatti permette, in modo lineare, di avere questa stima.

Per prima cosa, si suppone di conoscere sia la matrice degli intrinseci K che le corrispondenze 3D e 2D dei punti sopra l'immagine $\{m_i, M_i\} I = 1, \dots, n$ dove tutti gli M_i sono nel sistema di riferimento globale. Quindi, possiamo scrivere che

$$p_i \propto [R|t]M_i$$

dove

$$p_i = [u_i, v_i, 1]^T = K^{-1}m_i$$

sono in coordinate normalizzate.

Successivamente si introducono dei fattori detti di scala, questi fattori ci fanno passare alla seguente equazione:

$$\zeta_i K^{-1} m_i = [R|t] M_i = R \widetilde{M}_i + t \quad \forall i$$

in cui \widetilde{M}_i indica che il vettore non è in coordinate omogenee.¹ Ovvero, possiamo dire che il fattore di scala fa sì che un punto nello spazio 2D sia uguale ad uno dello spazio 3D più una rotazione ed una traslazione. Trovando questi ζ_i , possiamo ricondurre questo problema ad uno di stima assoluta della posa, che è risolvibile.

In forma matriciale possiamo scrivere

$$K^{-1}[\zeta_1 m_1, \zeta_2 m_2, \dots, \zeta_n m_n] = [R|t][M_1, M_2, \dots, M_n]$$

in cui nominiamo la matrice $[\zeta_1 m_1, \zeta_2 m_2, \dots, \zeta_n m_n]$ W e la matrice $[M_1, M_2, \dots, M_n]$ M . Definiamo r come il rango della matrice M , ed effettuiamo sempre su M la scomposizione a valori singolari, ottenendo le matrici UDV^T . Infine, prendiamo la matrice V_r come le ultime $n - r$ colonne della matrice V .

Per definizione, V_r rappresenta il kernel² della matrice M per definizione e si può scrivere

$$MV_r = O$$

da questo possiamo ottenere la seguente equazione:

$$K^{-1}WV_r = [R|t]MV_r = \overline{O}$$

Tramite queste operazioni siamo riusciti ad eliminare i termini che avevano a che fare con la rotazione e la traslazione.

Per risolvere l'equazione, dobbiamo cercare di isolare il termine sconosciuto W e scomporlo in una maniera che ci permetta di stimarlo, per fare ciò, usiamo l'operatore di vettorizzazione. Possiamo quindi scrivere che:

$$vec(K^{-1}WV_r) = \overline{O} \rightarrow (V_r^T \otimes K^{-1})vec(W) = \overline{O}$$

Avendo isolato il vettore W , che è formato da:

$$vec(W) = \begin{bmatrix} \zeta_1 m_1 \\ \zeta_2 m_2 \\ \dots \\ \zeta_n m_n \end{bmatrix} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & \dots & m_n \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \dots \\ \zeta_n \end{bmatrix}$$

Da qui, chiamando D la matrice $\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & \dots & m_n \end{bmatrix}$ possiamo riscrivere l'equazione nel modo seguente:

$$((V_r^T \otimes K^{-1})D)\zeta = O$$

¹Sistema di coordinate che introduce una quarta coordinata, detta omogenea, che è settata ad uno per i punti e zero per le direzioni.

²Matrice formata da tutti i vettori che moltiplicati con essa danno il vettore nullo.

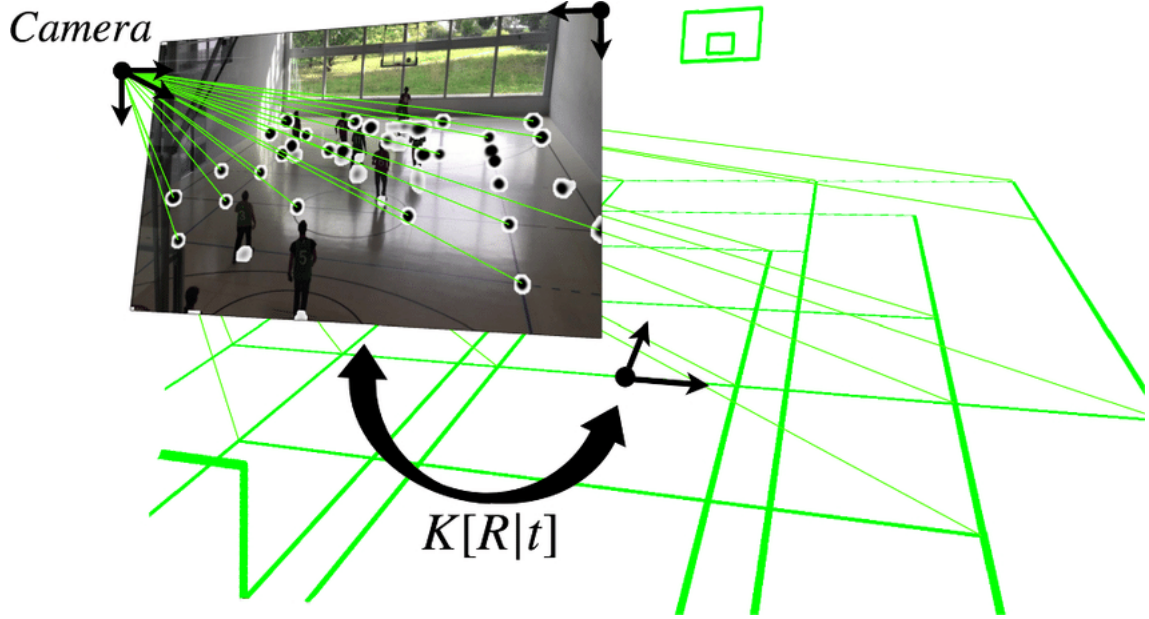


Figura 3.2: Visualizzazione della stima della posa che evidenzia la corrispondenza tra punti 2D e 3D[14].

Ora, conoscendo il resto, possiamo stimare la ζ .

Per computare i termini ζ ci serviranno un numero di equazioni che dipendano dalla grandezza di r , ovvero il rango della matrice M iniziale, in generale serviranno n equazioni, con n che deve essere

$$n \geq \frac{3r - 1}{2}$$

infine, una volta stimato ζ con tutte le equazioni necessarie, bisognerà andare a risolvere la seguente equazione:

$$\zeta_i K^{-1} m_i = R \tilde{M}_i + t$$

che è proprio un problema di orientazione assoluta della posa, facilmente risolvibile. Fortunatamente, il motore di Vuforia esegue in modo interno tutte queste operazioni per andare a mappare i punti, perciò non c'è stato bisogno di occuparcene per lo scopo della ricerca.

Capitolo 4

Game Design e videogiochi

Come osservato in precedenza, la realtà aumentata trova applicazione in una vasta gamma di contesti, che spaziano dalla medicina all'archeologia fino all'educazione. Tra questi, uno dei settori più significativi che ne fa uso è quello dei videogiochi, che continuamente cercano nuovi modi di integrarla. Basti pensare all'episodio rappresentato dall'uscita di Pokémon GO nel 2016, che ha catalizzato l'interesse globale per la realtà aumentata, portandola alla conoscenza delle masse[15]. Questo gioco è stato uno dei primi a sfruttare con successo la combinazione tra elementi fisici e virtuali, aprendo la strada ad un nuovo genere mobile, ovvero i giochi basati sulla location, che fanno uso del GPS del telefono per far avanzare il personaggio attraverso una mappa digitale che contiene però punti di interesse coincidenti con quelli esistenti nella realtà. Da questo, si può evincere che il contesto dei videogiochi cerchi sempre una continua evoluzione delle sue meccaniche e degli ambiti in cui si annovera.

In questa ricerca, si sono proposte varie meccaniche di gioco classiche implementate però nel contesto della realtà aumentata, cercando di capire come queste possano funzionare e come possano creare engagement nell'utente.

Tuttavia, prima di analizzare queste meccaniche, è necessario parlare di cosa effettivamente sia un videogioco e cosa siano le meccaniche al suo interno.

4.1 Definizione di un videogioco

Il videogioco è definito come un gioco gestito da un dispositivo elettronico che consente di interagire con le immagini di uno schermo[16]. Questa definizione tuttavia tratta il videogioco come un semplice software che si può avviare in un computer, quindi per definirlo veramente occorre capire quale sia la differenza tra software come può essere per esempio Blender e un videogioco come invece può essere Tetris. La definizione di videogioco è stata centro di un dibattito di molti teorici che hanno cercato di delinearne i confini.

Il dibattito ha portato alla formulazione di diverse prospettive e interpretazioni, ciascuna delle quali offre un'analisi unica del fenomeno del gioco. Una delle definizioni più significative proposte è quella che considera il gioco come un'espressione di dominio dove i giocatori si impegnano in una "lotta" contro gli avversari per conquistare l'obiettivo, limitata da regole e che alla fine conduce a un risultato che altera lo stato di equilibrio iniziale. Questo approccio mette in evidenza la natura competitiva del gioco e la sua capacità di generare tensione e suspense attraverso

il conflitto tra le parti coinvolte, che possono essere rappresentate anche da ostacoli o nemici comandati dal computer. Altre interpretazioni definiscono il gioco come una struttura interattiva, che richiede ai giocatori di impegnarsi attivamente nel perseguimento di obiettivi prestabiliti. In questa prospettiva, il gioco diventa un'opportunità per sperimentare all'interno di un contesto ludico, dove le azioni dei partecipanti contribuiscono a determinare il corso degli eventi e il risultato finale.

Invece, alcuni studiosi concepiscono il gioco come un sistema chiuso, in cui le regole stabilite definiscono chiaramente i parametri e i vincoli entro cui avviene l'interazione dei giocatori. Questo approccio sottolinea l'importanza delle regole nel fornire una struttura e un'organizzazione al gioco, che contribuiscono a definire i ruoli e le dinamiche di partecipazione al suo interno.

Infine, c'è un'ultima idea da prendere in considerazione, ovvero quella del valore interno, che sottolinea l'importanza delle esperienze e delle interazioni generate all'interno del contesto del gioco che possiede la capacità di generare un coinvolgimento emotivo e cognitivo significativo tra i partecipanti, che contribuisce a rendere l'esperienza ludica profondamente gratificante e memorabile[17].

Sebbene la definizione esatta di cosa sia un videogioco è dibattibile, le definizioni qui proposte hanno tutte come elementi comuni il fatto che il gioco debba avere un obiettivo da raggiungere, delle regole da rispettare per raggiungere quell'obiettivo, delle difficoltà od ostacoli che si frappongono tra l'utente e l'obiettivo da raggiungere, e soprattutto un mezzo che è capace di creare coinvolgimento e suscitare emozioni a chi lo utilizza. In conclusione, si può dire che il videogioco sia un'esperienza per l'utente.

4.2 Elementi chiave di un videogioco

Un videogioco è un tipo di software che pone all'utente un obiettivo da raggiungere, delle regole da rispettare, e soprattutto capace di suscitare emozioni e coinvolgimento in esso[17]. Proprio il fattore del coinvolgimento del giocatore nel gioco è quello che distingue un buon videogioco da uno meno buono, per ottenere questo coinvolgimento, bisogna avere alcuni fattori chiave che sono:

- Motivation
- Immersion
- Flow

La prima di queste, la *Motivation*, o motivazione, indica quanto i giocatori siano motivati a giocare il gioco in questione. La motivazione di giocare ad un gioco viene principalmente stimolata dal fatto che esso soddisfa alcuni bisogni primordiali dell'essere umano, come il bisogno di auto-realizzazione.

Diversi studi infatti hanno rilevato[18] che i primi videogiochi che hanno avuto successo sono quelli che hanno sfruttato il bisogno umano di realizzare le competenze, proponendo, come nel caso dei famosi Galaga, Donkey Kong e Pong, delle sfide di riflessi che andavano ad aumentare di difficoltà man mano che il giocatore avanzava nel gioco, e proprio questo bilancio tra difficoltà di gioco e abilità del giocatore è stata la chiave che ha permesso loro di avere successo.

Quindi, emerge come questo bilancio sia fondamentale per tenere la motivazione dell'utente, che può andare facilmente persa. I videogiochi devono andare a soddisfare

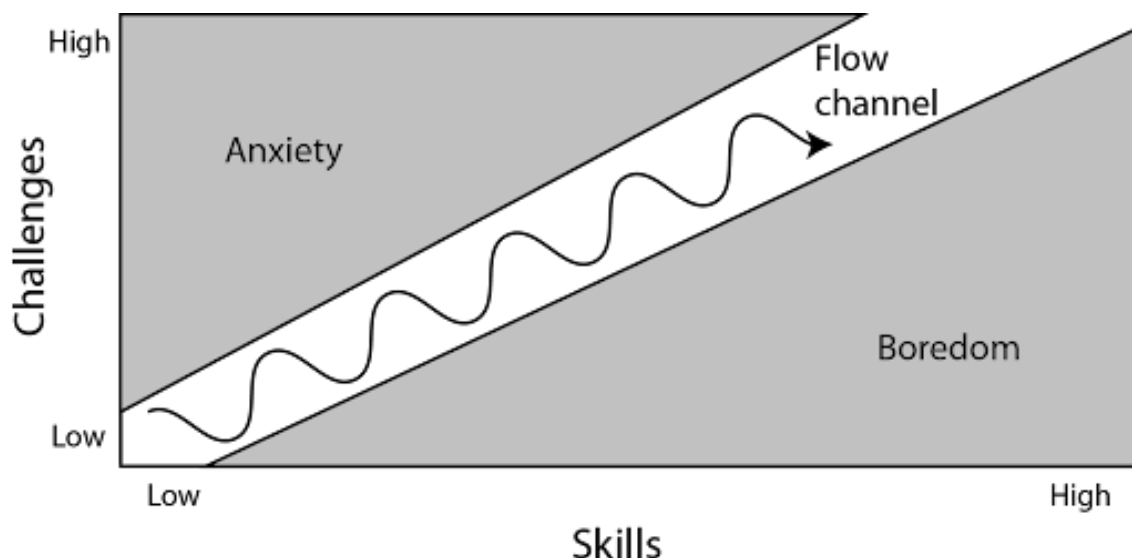


Figura 4.1: Diagramma che mostra il bilanciamento da tenere durante il gioco per creare engagement[19].

alcuni bisogni chiave dell'essere umano per risultare coinvolgenti, e spingere l'utente ad essere motivato a giocarli.

Come evidenziato in figura 4.1, la sfida proposta dal gioco deve essere in equilibrio con l'abilità richiesta al giocatore, dove un contesto di alta sfida ma poca abilità crea una sensazione di ansia, mentre un contesto di poca sfida ma troppa abilità richiesta crea invece un contesto di noia.

La soluzione ideale potrebbe variare a seconda del contesto e degli obiettivi del gioco. Tuttavia, come illustrato nella figura 4.1, è possibile considerare un approccio che preveda una progressione non lineare, con momenti di tensione e rilassamento alternati, mantenendo comunque un bilanciamento generale. È importante mantenere il bilanciamento affinché il gioco risulti coinvolgente e stimolante, evitando al contempo di generare frustrazione o noia nei giocatori. Va tenuto presente che il concetto di bilanciamento può variare notevolmente in base al tipo di gioco, al pubblico di riferimento e all'esperienza emotiva che si intende suscitare. Pertanto, è necessario valutare attentamente questi fattori durante lo sviluppo del gioco al fine di ottenere il risultato desiderato.

Sempre in questa figura, viene indicato il termine *Flow*, esso indica il sentimento di soddisfazione che occorre proprio quando c'è un bilancio tra abilità richiesta e sfida proposta [20], che crea come dice il nome un senso di concentrazione e coinvolgimento in cui il giocatore non vuole abbandonare il gioco, in cui sono immersi in esso e non vengono più distratti dagli stimoli esterni. Tuttavia, non è possibile misurare questo *Flow* nel breve periodo, quanto più va misurato durante tutto l'arco del gioco, fatto che rende complesso il suo bilanciamento.

Un' ultima caratteristica degna di nota dei videogiochi è la capacità di immersione, o immersion, che è definita come l'esperienza di diventare coinvolto all'interno del gioco ritenendo comunque sempre la coscienza dei propri dintorni, ma descrive anche la sensazione del giocatore di essere parte del mondo che sta giocando. Questo stadio di immersione del giocatore nel mondo di gioco si può raggiungere in svariati modi: si può ridurre al minimo l'interfaccia utente, per diminuire ciò che separa il giocatore dal suo personaggio, cambiare il tipo di inquadratura del proprio gioco da

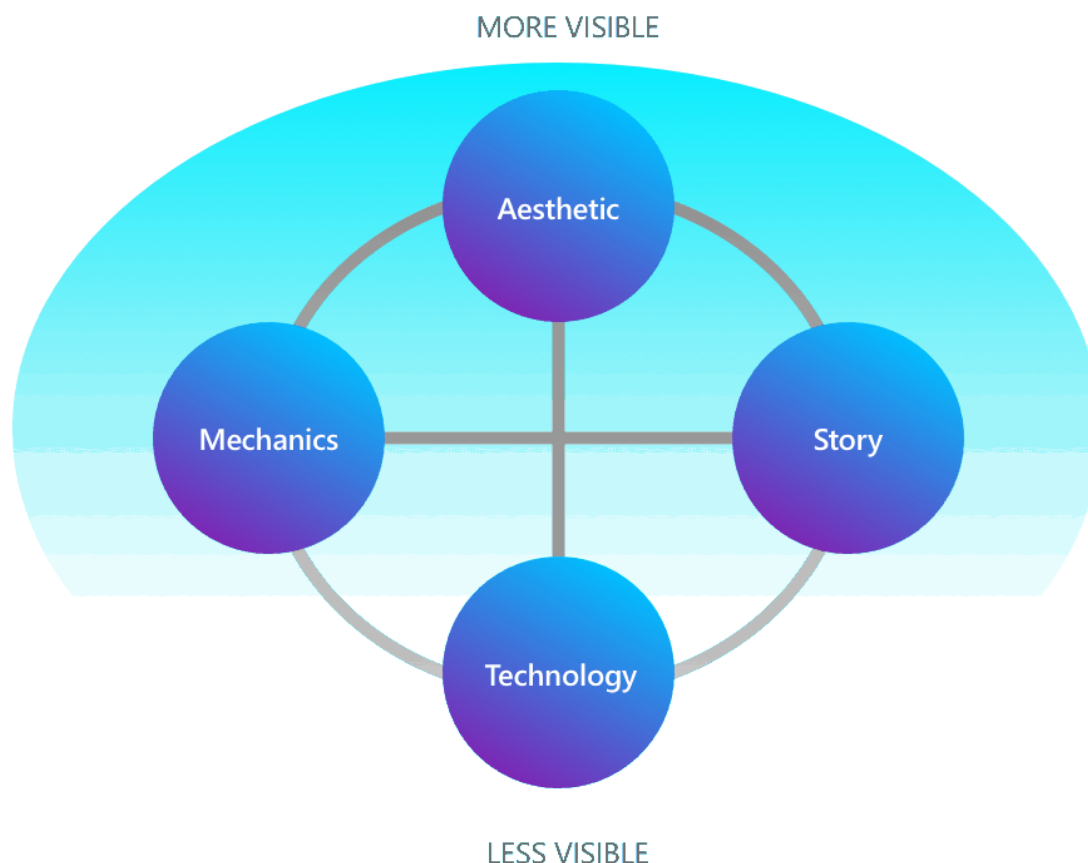


Figura 4.2: Schema che mostra le parti fondamentali di un videogioco in relazione alla loro visibilità[21].

terza a prima persona, oppure offrire la possibilità di crearne uno proprio. Tramite questi concetti fondamentali, un buon videogioco riesce quindi a coinvolgere i propri giocatori, creando un loop soddisfacente di gameplay che invoglia gli utenti a giocarlo.

4.3 Struttura di un videogioco

In questa sezione, verranno analizzati degli elementi che costituiscono un videogioco nella sua completezza. Tra le diverse metodologie di classificazione proposte, emergono quattro pilastri fondamentali: le Meccaniche di Gioco, la Narrazione, l'Estetica e la Tecnologia. Come è possibile osservare nella figura 4.2, questi elementi, pur essendo distinti, sono strettamente interconnessi e sono presenti in ogni videogioco esistente, indipendentemente dalla sua complessità o dal budget a disposizione. Tuttavia, come si può notare da questo schema, alcune di queste componenti potrebbero non essere immediatamente evidenti ad una prima occhiata, come ad esempio la tecnologia o le meccaniche di gioco. Ciononostante, è importante sottolineare che anche questi giocano un ruolo fondamentale nel determinare l'esperienza offerta dal videogioco e non vanno quindi sottovalutati.

4.3.1 Estetica

Con il termine estetica ci si riferisce a come un videogioco viene percepito, racchiude cioè tutte quelle sensazioni che sono percepite provandolo, ma anche come esso appare agli occhi, quindi il suo stile grafico o la responsività dei suoi controlli [22]. L'elemento dell'estetica è quello che appare più evidente agli occhi dei giocatori ed è una parte che va curata in quanto non solo dà il primo impatto a chi gioca, ma risulta uno di quelli determinanti perché i giocatori si immergano nella storia. Per avere una buona estetica però è necessario scegliere anche delle meccaniche che supportino il look del gioco, una tecnologia che sia in grado di visualizzare l'estetica desiderata e una storia che abbia un tono congruente a quest'ultima.

In sostanza, l'estetica è il risultato di una sinergia tra tutti gli elementi del gioco, che lavorano insieme per creare un'esperienza visiva coinvolgente e coerente. Quest'ultima quindi non solo migliora l'attrattiva del gioco, ma contribuisce anche a consolidare l'immersione dei giocatori nella sua trama e nell'universo di gioco.

4.3.2 Storia

Per storia di un videogioco si intende la sequenza degli eventi che si svolgono all'interno di esso, sia essa una storia lineare oppure scegliibile dal giocatore e quindi fatta a diramazioni. Il ruolo della storia è intrinsecamente connesso con quello delle meccaniche di gioco, in quanto quest'ultime, se scelte correttamente, possono dare maggior risalto e spessore alla storia e viceversa.

Per ottenere una buona storia, è necessario che l'estetica del gioco sia in linea con i temi trattati e ci sia una tecnologia adeguata per soddisfare le caratteristiche richieste da quest'ultima. Ogni videogioco, indipendentemente dalla complessità della sua trama, necessita di una storia, intesa però come sequenza di eventi che si svolgono al suo interno, che quindi può prendere anche la semplice definizione di contesto.

Ad esempio, anche un gioco che si concentra semplicemente sull'obiettivo di sparare a navicelle aliene ha una storia, poiché offre al giocatore il contesto necessario per comprendere il proprio ruolo e gli obiettivi da raggiungere.

4.3.3 Tecnologia

La tecnologia del videogioco, è l'insieme di strumenti utilizzati per la sua creazione e il suo funzionamento, è il mezzo per cui l'estetica del gioco prende vita. In questo elemento sono compresi quindi tutti gli strumenti utilizzati per la creazione del gioco, che siano le matite o le penne per prendere appunti o il motore di gioco usato per svilupparlo (ad esempio Unity o Unreal Engine).

La scelta della tecnologia è una delle più importanti, in quanto questa determina che tipo di estetica e meccaniche possono essere eseguite, e quali invece sono impossibili da realizzare.

Di conseguenza, le capacità e le limitazioni della tecnologia influenzano direttamente le possibilità creative degli sviluppatori e il prodotto finale. Prestare attenzione alla scelta della tecnologia è fondamentale per sviluppare un buon videogioco.

4.3.4 Meccaniche

Le meccaniche di gioco sono per definizione l'insieme di regole e procedure che determinano come l'utente deve raggiungere l'obiettivo finale posto dal gioco. Esse descrivono quindi come i giocatori si possono avvicinare all'obiettivo, cosa succede quando falliscono ad ottenerlo e cosa invece quando lo raggiungono. Le meccaniche rappresentano il cuore del videogioco ed è ciò che lo rende un'esperienza interattiva unica: a differenza di altre forme di intrattenimento come libri, film o spettacoli teatrali, che possono vantare una tecnologia, una storia e un'estetica definite, i videogiochi si distinguono per la presenza delle meccaniche di gioco.

La selezione delle meccaniche deve essere strettamente correlata agli altri elementi del videogioco: esse amplificano la narrazione e hanno senso solo se integrate in un contesto fornito dalla storia; devono riflettere un tono coerente con l'estetica del gioco e sono rese possibili grazie all'utilizzo della giusta tecnologia.

4.4 Meccaniche di gioco

Il fulcro della ricerca condotta per questa tesi si concentra sullo studio delle meccaniche di gioco applicate al contesto della realtà aumentata (AR). Le meccaniche di gioco delineano le regole, le interazioni e i sistemi che guidano l'esperienza del giocatore. Tuttavia, nel contesto dell'AR, queste meccaniche assumono una nuova dimensione, poiché si integrano direttamente con il mondo reale, sfruttando la tecnologia per sovrapporre elementi digitali alla visione del mondo fisico.

Prima di parlarne bisogna però definire cosa sia una meccanica di gioco. Le meccaniche di gioco, sono definite come quelle cose che delineano le regole di quest'ultimo, come esso viene controllato e cosa gli utenti possono fare al loro interno. Tuttavia, questa definizione risulta vaga e molti esperti hanno tentato di fornire una tassonomia definitiva[17], senza però giungere ad una vera conclusione unanime. Per questa ricerca, si è adottata la tassonomia presente nel libro *"The art of Game Design: A book of lenses"* di Jesse Schell, che racchiude le meccaniche in sei principali macrotipi:

- Spazio
- Oggetti, attributi e stati
- Azioni
- Regole
- Abilità
- Chance

4.4.1 Spazio

Le meccaniche di tipo spaziale sono quelle meccaniche che definiscono come lo spazio del gioco è organizzato. Esse definiscono tutti gli spazi esistenti nel gioco, e come questi spazi si connettono l'uno con l'altro, oltre che come essi siano fatti.

Di solito, uno spazio ha tre caratteristiche principali: è discreto o continuo, un

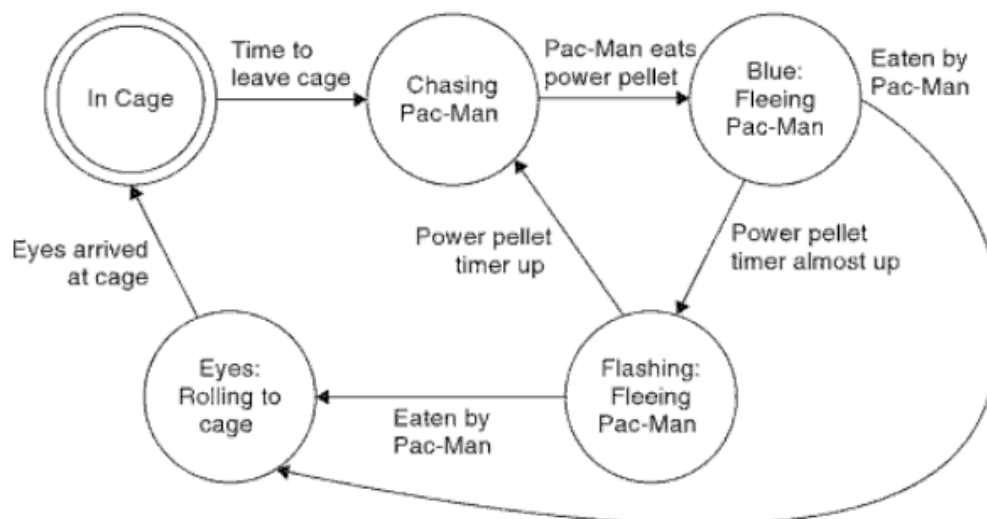


Figura 4.3: Immagine che visualizza gli stati di un oggetto di gioco[23].

numero di dimensioni e delle aree adiacenti a cui è connesso. Uno spazio discreto è uno spazio in cui il giocatore può muoversi solo in alcune posizioni prestabilite e non in tutta la sua totalità, mentre continuo è l'opposto; il numero di dimensioni indica in quante dimensioni il giocatore si può muovere, mentre l'ultimo attributo permette di visualizzare come queste aree si interconnettono tra loro.

Ci sono poi delle meccaniche che descrivono situazioni come spazi innestati, ovvero dove uno spazio grande ne nasconde uno al suo interno, come per esempio alcuni videogiochi dove il personaggio interagendo con una porta di una casa o con un'entrata di un luogo viene trasportato tramite caricamento ad uno spazio diverso che simboleggia la casa stessa, che però non è veramente presente nella mappa precedente.

4.4.2 Oggetti, attributi e stati

Come il nome fa intuire, queste meccaniche si occupano di regolare tutti gli oggetti presenti nel gioco, i loro attributi, che possono essere anche solo la loro posizione nel mondo di gioco, e lo stato di questi ultimi, definito come il valore del suddetto attributo, mostrato in figura 4.3.

Queste meccaniche hanno perciò il ruolo di regolare anche le informazioni date al giocatore riguardo a questi attributi, per esempio se visualizzarli e farli conoscere ad esso in fase di gioco, prospettiva che va quindi valutata in quanto non è sempre un bene che ciò avvenga, i giochi che infatti danno troppe informazioni ai propri giocatori possono risultare soverchianti per questi ultimi.

4.4.3 Azioni

Queste meccaniche si occupano di descrivere ciò che i giocatori possono fare, ovvero le azioni che possono compiere durante il gameplay. Ci sono due tipi di azioni:

- **Operative:** sono quelle azioni basilari che il giocatore può intraprendere in ogni momento.

- Risultanti: sono quel tipo di azioni che si manifestano nel contesto allargato del gioco, ovvero ciò che il giocatore può compiere nel lungo periodo, come le strategie che può utilizzare per esempio in un gioco di scacchi.

Idealmente, si dovrebbe ricercare un bilancio di questo tipo di azioni, senza avere poche azioni basilari, che porterebbero ad un gameplay ripetitivo e noioso, ma nemmeno troppe, il che causerebbe confusione nel giocatore relativamente a quelle risultanti.

Per creare questo tipo di bilanciamento, bisogna cercare di soffermarsi su come il giocatore può raggiungere il proprio obiettivo, e pensare ad altri modi in cui questo possa essere raggiunto.

4.4.4 Regole

Le meccaniche associate alle regole sono quelle che indicano gli oggetti, lo spazio, le azioni e le loro conseguenze, oltre che i loro limiti. Questo tipo di meccaniche possiede tantissimi sottotipi diversi, ricercati come evidenziati dallo schema di David Parlett[24], riportato in figura 4.4. Esse si possono suddividere in:

- Operative: queste regole definiscono cosa i giocatori fanno all'interno del gioco, come esso si svolge
- Fondamentali: rappresentano lo stato del gioco e come esso cambia in modo matematico.
- Comportamentali: le regole implicite, quelle che non sono scritte ma che si assume tutti conoscano.
- Scritte: il tipo di regole scritte che delineano come giocare, nel campo dei videogiochi per esempio può essere il tutorial iniziale.
- Leggi: le regole che si applicano quando il contesto è serio come nei tornei internazionali dei giochi competitivi, ma che non si applicano normalmente.
- Ufficiali: sono il tipo di leggi che, col tempo, sono diventate regole scritte, e quindi adottate in modo ufficiale dal gioco.
- Consultive: spesso chiamate "regole di strategia", sono solo suggerimenti per aiutarti a giocare meglio e non sono realmente "regole" dal punto di vista delle meccaniche di gioco.
- Regole della casa: indicate come *feedback* nello schema in figura 4.4, queste regole sono quelle che vengono utilizzate da un gruppo di giocatori informale per modificare il gameplay e renderlo più interessante per loro.

4.4.5 Abilità

Le meccaniche di gioco riguardanti le abilità sono quelle che vengono applicate al giocatore, ossia ciò che il gioco gli richiede: devono essere bilanciate con l'esperienza e la difficoltà del gioco per raggiungere un buon flow. Solitamente, i videogiochi non richiedono solo un tipo di abilità all'utente, ma svariati, ed esse si possono tassonomizzare in tre grandi macro-tipi:

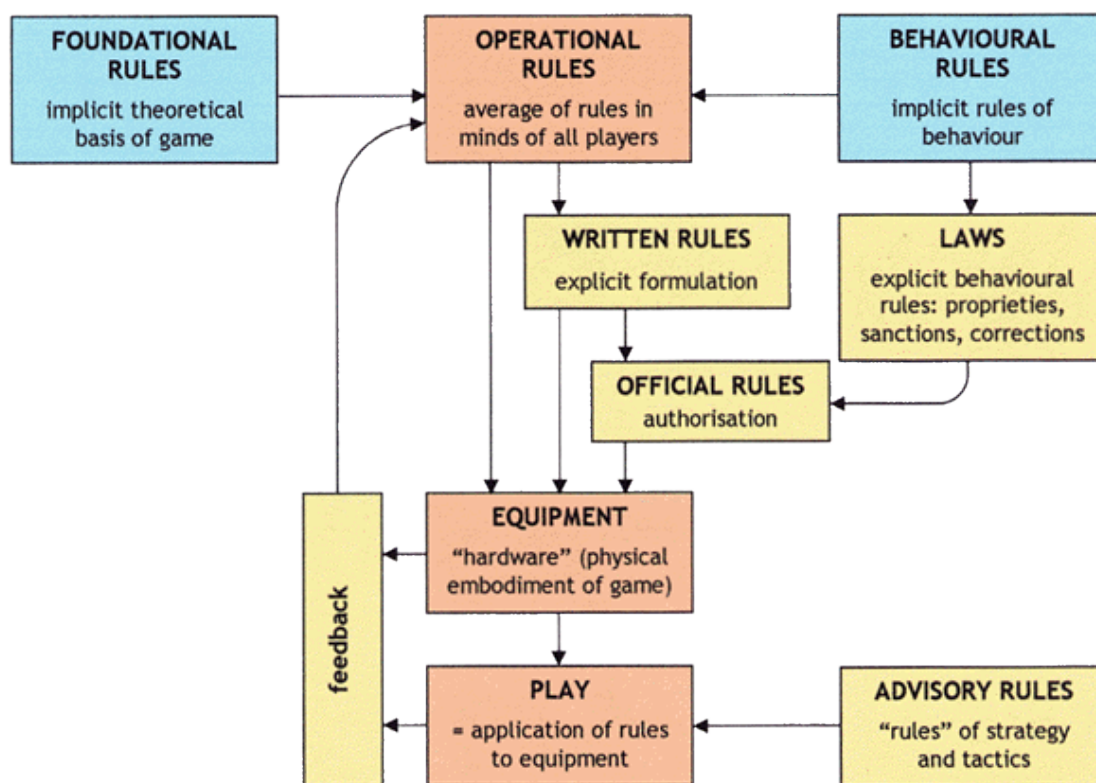


Figura 4.4: Schema delle meccaniche di tipo regola di Parlett[25].

- **Abilità fisiche:** abilità di manipolare oggetti che richiedono forza o destrezza, per esempio il manipolare il controller in un certo modo oppure per i casi di giochi come simulatori di guida dotati di volante, di saperlo maneggiare.
- **Abilità mentali:** questo tipo di abilità riguardano la memoria, l'osservazione e la capacità di risolvere enigmi. Vengono applicate soprattutto nei giochi a rompicapi.
- **Abilità sociali:** le abilità sociali si concentrano soprattutto sulla capacità di interagire in modo efficace con le altre persone, leggere le mosse dell'avversario nel caso dei giochi di combattimento e saper cooperare nei giochi di squadra.

Quindi queste meccaniche sono applicate alle vere abilità che vengono richieste al giocatore, mentre non si applicano alle abilità virtuali, ciò che il personaggio sa fare al suo interno.

4.4.6 Chance

Le meccaniche di questo tipo regolano le incertezze e le possibilità all'interno del gioco, vanno quindi a regolare con quanta probabilità le cose avvengono e cosa può avvenire in un determinato momento. La probabilità e l'incertezza sono un fattore molto importante nell'ambito videoludico, in quanto garantiscono che l'esperienza del giocatore sia sempre nuova ogni partita ed evita il sopraggiungere della noia, aumentando al contempo anche l'immersione dell'utente.

Per gestire queste incertezze, di solito si assegna a ciascuna scelta possibile ciò

che viene chiamato il *valore atteso*, che rappresenta una stima del guadagno medio generato da quella specifica opzione. Il design del gioco è progettato basandosi su questa media, consentendo al giocatore di ottenere vantaggi se supera tale valore atteso, oppure di subire svantaggi temporanei se va sotto di esso, contribuendo a mantenere il gameplay coinvolgente.

Questo bilanciamento tra rischi e opportunità offre una varietà di risultati possibili e incoraggia la sperimentazione e l'adattamento strategico durante il gameplay.

Capitolo 5

Studio delle meccaniche di gioco in realtà aumentata

Lo scopo della ricerca è analizzare come alcune meccaniche di gioco si possano adattare al contesto della realtà aumentata. A tale scopo sono state create alcune applicazioni demo che implementano in maniera molto semplice e differenziata le meccaniche analizzate. Le medesime, sono state provate da un gruppo di utenti ai quali, successivamente è stato somministrato un questionario su cui lasciare il loro feedback.

Ogni applicazione è stata realizzata con il modello di realtà aumentata di tipo *Model Based*, utilizzando il motore di gioco Unity, di realtà aumentata Vuforia e il software di fotogrammetria 3D Zephyr per acquisire i modelli 3D.

Una volta importati all'interno di Unity, i modelli creati tramite Zephyr dovranno subire un cambio di sistema di riferimento per essere correttamente posizionati all'interno dell'ambiente. Unity infatti utilizza un sistema di riferimento diverso da quello di Zephyr, in particolare Unity utilizza un sistema di riferimento sinistrorso con l'asse y che punta in alto [26], mentre Zephyr utilizza un sistema di riferimento destrorso con la X in alto, come mostrato in figura 5.1.

Un altro aspetto importante di questa ricerca è la valorizzazione del patrimonio culturale, in particolare quello del tratto di mura presente a Verona. La ricerca delle meccaniche di gioco infatti è stata svolta anche con l'obiettivo di poter aggiungere quest'ultime all'interno dell'applicazione non ancora rilasciata chiamata Mura AR, che si occuperà di valorizzare proprio questa zona di Verona utilizzando come tramite un videogioco, al cui interno sono inserite spiegazioni della storia e dell'importanza dei monumenti visitati mentre l'utente viene intrattenuto durante le meccaniche di gioco.

Qui di seguito verranno riportate le meccaniche con le loro relative analisi, tentando di inserirle nella tassonomia adottata nel capitolo precedente. E' importante tuttavia notare che essendo quest'ultima non univoca per tutti, è possibile classificare le meccaniche in molte categorie diverse. Saranno quindi fornite dove possibile alcune interpretazioni diverse di ciascuna di esse.

5.1 Lancio di un oggetto con fisica

La prima meccanica analizzata nella ricerca è il lancio di un oggetto sintetico che andrà ad obbedire alle leggi fisiche. Queste leggi possono variare a seconda del



Figura 5.1: Sistema di riferimento di Zephyr e Unity a confronto

contesto in cui l'oggetto viene lanciato, ma per i nostri scopi si utilizza la forza di gravità come forza agente sull'oggetto lanciato, aggiunta ad una forza iniziale di lancio.

5.1.1 Versione Iniziale, lancio senza gravità

Inizialmente, si è creata una demo di gioco di shooting, dove all'utente viene richiesto di colpire dei dischi volanti che compaiono sullo schermo e seguono delle traiettorie precise. Per questa versione quindi non viene utilizzata la forza gravitazionale sull'oggetto ma solo la forza derivante dalla spinta iniziale.

Le versioni proposte per questa implementazione sono state due, nella prima all'utente viene chiesto di cliccare qualsiasi punto dello schermo per lanciare il proiettile, che è sempre sparato al centro di dove la telecamera del smartphone sta puntando. La seconda versione invece richiede all'utente di premere il punto dello schermo che desidera colpire.

5.1.2 Implementazione della demo

Per realizzare lo sparo del proiettile nella prima versione, dove l'arma spara sempre al centro dello schermo, viene generato un Raycast con origine l'oggetto camera e direzione il vettore *forward* di quest'ultima, ovvero la sua asse Z locale. Questo Raycast va incontro a due possibilità: viene fermato da un collider¹ oppure si interrompe autonomamente dopo una certa distanza. In entrambi i casi, all'estremità del Raycast viene generato un punto che determinerà la direzione con cui sarà lanciato il proiettile.

Per l'implementazione del sistema di mira associato alla telecamera, presente nella variante dove l'utente può scegliere il punto dello schermo da colpire, è stato utilizzato nuovamente il sistema di Raycast. In particolare, quando l'utente preme sullo

¹confini modificabili di un oggetto in Unity, che causano collisioni.

schermo del telefono, un Raycast viene generato in quel punto fino ad una distanza predeterminata in fase di sviluppo (circa quattromila unità), e da questo viene generato un punto. Successivamente, dalla camera della scena verrà generato un Raycast che andrà dall'oggetto camera fino al punto generato dal touch dell'utente. Questa sarà la traiettoria del proiettile, che verrà lanciato in questa direzione, verificando che durante il suo volo incontri dei collider appartenenti agli UFO, distruggendoli.

5.1.3 Inserimento nel contesto del patrimonio culturale

Per iniziare la demo è richiesto di inquadrare uno dei cannoni presenti sopra le mura di Verona, gli UFO saranno quindi generati prendendo come riferimento quest'ultimo. Per questa applicazione tuttavia non si è per il momento ancora trovato un metodo per fare interagire gli utenti con il modello 3D che non fosse sparare a dei punti sopra quest'ultimo, il che sarebbe probabilmente risultato noioso o non appropriato per il fine richiesto.

5.1.4 Lancio con forza di gravità

Nella seconda versione del lancio di un oggetto con fisica viene aggiunta una componente gravitazionale al medesimo rendendolo un lancio parabolico. L'obiettivo del gioco è quello di colpire un punto specifico dell'oggetto acquisito, ovvero la canna del cannone per semplicità indicata da una capsula bianca, utilizzando come proiettile una palla di cannone.

Anche in questo caso, vengono presentate varie versioni del gioco agli utenti, i quali utilizzando un sistema simile a quello di una fionda, devono tenere conto della parabola del proiettile.

La differenza sostanziale tra le due versioni di lancio a gravità consiste, per una di esse, nella possibilità di visualizzare la traiettoria del lancio prima che questo avvenga: una linea blu si rende visibile, mentre l'utente sta mirando, ad indicare la traiettoria che il proiettile andrà ad eseguire, come mostrato in figura 5.2. In entrambe le versioni, rilasciando lo schermo la palla sarà lanciata in avanti descrivendo una traiettoria parabolica che avrà angolo di lancio tanto superiore quanto più l'utente trascina in basso il proiettile.



Figura 5.2: Demo del lancio parabolico con forza di gravità, nella variante che mostra la traiettoria

5.1.5 Implementazione della demo

L'implementazione di questa demo consiste in tre parti principali: il posizionamento della sfera prima del tiro, la predizione della traiettoria e il lancio della sfera.

Posizionamento della sfera Per gestire il posizionamento del proiettile nel contesto del lancio, si impiegano due funzioni di interpolazione lineare². La prima funzione agisce sulla profondità della sfera, rendendo la sua posizione sempre più prossima al punto di osservazione dell'utente man mano che questa viene spostata verso il basso dello schermo. Questo effetto è progettato per creare una sensazione di profondità dinamica: all'utente sembrerà di utilizzare una fionda per il lancio della sfera. La seconda funzione di interpolazione modifica l'angolo di lancio della sfera. Questo angolo parte da zero gradi quando la sfera si trova approssimativamente a metà dello schermo e aumenta gradualmente fino a 80 gradi quando la sfera viene posizionata nella parte inferiore di quest'ultimo, permettendo di regolare l'angolo di lancio.

Tiro della sfera Per gestire il lancio della sfera, è stata creata una funzione apposita che prendendo l'angolo di lancio (ricavato dal posizionamento dell'utente), suddivide la velocità iniziale, preimpostata in fase di design, nelle sue due componenti sugli assi X ed Y, e la setta al proiettile tramite il componente apposito di Unity *RigidBody.velocity*. La sfera conseguentemente andrà a compiere una parabola diversa a seconda dell'angolo a cui viene lanciata.

Simulazione della traiettoria Per simulare e visualizzare la traiettoria del lancio, viene attivato un componente ad ogni frame in cui l'utente posiziona la sfera. Questo componente, prendendo in input l'angolo ipotetico del lancio e la velocità

²Una funzione di interpolazione lineare è una funzione che stima il valore intermedio tra due dati in input.

iniziale, applica le leggi fisiche del moto parabolico per calcolare un insieme di punti che rappresentano la traiettoria prevista dell'oggetto. Successivamente, questi punti vengono passati a un oggetto *LineRenderer*, il quale ha il compito di mostrare la linea parabolica all'utente.

5.1.6 Studio delle meccaniche presenti

Possiamo eseguire una distinzione delle meccaniche presenti in queste due demo tramite la tassonomia vista nel capitolo precedente.

Un aspetto comune di entrambe le demo è l'uso di uno spazio continuo, il proiettile è libero di viaggiare nella totalità di quest'ultimo in entrambe le versioni, nonostante la maggior parte di esso sia vuoto, fatta eccezione per il modello dell'oggetto e gli obiettivi da colpire.

Nello specifico però, le meccaniche sottoposte a studio in queste due demo sono meccaniche classificabili come di tipo "oggetto", in quanto descrivono le caratteristiche del lancio dell'oggetto proiettile: la sua traiettoria, il suo attributo di massa e gravità (nel caso abbia fisica), la sua velocità ecc. Nella variante del lancio in cui è presente la fisica si vanno a testare altre meccaniche di questo tipo, per esempio se omettere al giocatore l'informazione su come sarà la traiettoria descritta dall'oggetto.

Vi sono poi altre meccaniche che compongono queste demo, nello specifico meccaniche di tipo "abilità", che descrivono come il giocatore debba avere le abilità fisiche non solo di saper ruotare in modo opportuno la telecamera del telefono al bisogno, quanto più che sappia predire la traiettoria dei bersagli e prendere la mira.

Dal punto di vista delle meccaniche di tipo "azioni", esse sono molto basilari: il giocatore infatti ha a disposizione solo un'azione disponibile in ogni dato momento, cioè sparare all'obiettivo proposto, non potendo formare molte strategie nel lungo termine.

Per quanto riguarda le meccaniche di tipo "regola", se ne prevedono di due tipi, all'inizio, una di tipo scritto, ovvero una breve schermata tutorial per far capire come funziona il gioco, la seconda di tipo operativo, ove il giocatore andrà a sparare agli obiettivi per completare il gioco.

Come ultimo tipo di meccaniche presenti, possiamo notare quelle di tipo "chance" nella applicazione del lancio senza fisica, alcuni UFO all'interno della demo infatti hanno un pattern di movimento che consiste nel muoversi in un punto all'interno di una circonferenza di raggio prestabilito, il che determina un certo grado di imprevedibilità a cui il giocatore si deve adeguare.

5.1.7 Inserimento nel contesto del patrimonio culturale

In questo caso, al giocatore è richiesto di inquadrare nuovamente il cannone sulle mura di Verona. Nel corso del gioco, i partecipanti devono colpire un obiettivo posizionato dentro la canna del cannone con un proiettile, questo metodo è stato selezionato in quanto il bersaglio non è comunque facile da colpire.

5.2 View Matching

Questo tipo di meccanica consiste nel dover ritrovare la posa della camera che ha scattato una particolare fotografia. La medesima è presente in moltissimi giochi,

anche non in realtà aumentata, pur se difficilmente come meccanica principale. Per implementare questa meccanica nella realtà aumentata, è stata creata anche in questo caso una demo apposita con diverse versioni. La demo in questione inizia una volta inquadrato un monumento specifico, in questo caso Porta Nuova. Una volta avviata, essa mostra all'utente l'immagine della foto di cui dovrà trovare la posizione, una volta trovato il punto di osservazione, comparirà la scritta "Congratulations!". Nel caso della prima variante della applicazione, la foto viene mostrata nell'angolo in basso a destra in modo perenne, mentre nella seconda variante la medesima viene messa in sovrapposizione sopra la telecamera.

5.2.1 Implementazione della demo

Per realizzare questa demo ci si è avvalsi di un Tool apposito, creato per un progetto di Computer Grafica, che permette di posizionare una camera virtuale di Unity nella posa di una reale usando il file di intrinseci ed estrinseci di una foto.



Figura 5.3: Schermata della demo sul View Matching: prima versione

Una volta impostata la camera virtuale nella posizione della foto, è stato creato uno script su Unity che controlla ad ogni frame la rotazione e la posizione della telecamera dell'utente rispetto a quella target, dando la vittoria nel caso che queste combacino.

Per entrambe le versioni, è stato creato un sistema di "tolleranza" che permette all'utente di non dover essere precisamente nella stessa posa della camera, ma in una approssimativa, in modo da evitare eccessiva frustrazione nel raggiungimento dell'obiettivo.

La schermata della demo sarà quindi composta dai seguenti elementi:

1. foto: questa immagine è quella che l'utente dovrà ricercare nell'ambiente circostante, posizionandosi nella stessa angolazione in cui è stata scattata.
2. suggerimento: questo quadrato ha un colore che indica all'utente quanto esso è vicino alla posizione corretta, esso passa dal blu, che indica che l'utente è lontano, al rosso, che indica la sua vicinanza.

3. indicatori di angolazione e posizione: questi indicatori mostrano a schermo quando l'utente è in una posizione corretta, facendo apparire le scritte "Abbastanza vicino" o "Angolazione corretta".

Per creare queste componenti, sono state usate varie funzioni di Unity. In particolare, per creare la componente numero tre è stata usata una funzione di interpolazione dei colori, usando il rosso e il blu come colori e la distanza tra giocatore e obiettivo come fattore di interpolazione.

Invece, per creare il fattore di tolleranza, sono state usate le funzioni "*Quaternion.Angle*" che restituisce l'angolo di Eulero³ tra due quaternioni⁴, verificando che esso sia minore di una certa soglia (in questo caso dieci gradi), e la funzione "*Vector3.Distance*" che invece fornisce la distanza tra due oggetti, anche questa verificata minore di una certa soglia (ovvero 0.15 unità).

5.2.2 Studio delle meccaniche presenti

La meccanica del view matching è difficile da classificare nelle categorie analizzate, una delle possibili è nel tipo "abilità", infatti questa meccanica possiamo dire rappresenti l'abilità richiesta al giocatore di capire ed interpretare una foto, sapendone ricercare la posizione in cui è stata scattata nell'ambiente reale. Essa è perciò di tipo mentale. Tuttavia può essere inserita anche nelle meccaniche di tipo "oggetto" poichè nel codice essa controlla come la camera del giocatore viene posta rispetto a quella obiettivo.

Notiamo che sono ancora presenti delle meccaniche che descrivono lo spazio come di tipo continuo, infatti in questo caso è l'utente che può muoversi liberamente all'interno di esso senza limitazioni. Vi sono inoltre meccaniche di tipo "regola", come il tutorial sempre presente prima di iniziare la demo; sono poi presenti meccaniche di tipo "oggetto", quelle cioè che controllano la rotazione e la posizione della camera, e quelle che informano l'utente se queste ultime sono quelle corrette per completare con successo il gioco. Anche in questo caso sono presenti meccaniche di tipo azione molto semplici, in cui il giocatore non ha possibilità di strategia e può solo limitarsi a posizionarsi nello spazio.

Non sono invece presenti meccaniche di tipo "chance", tutto il gioco quindi avviene in maniera deterministica.

5.2.3 Inserimento nel contesto del patrimonio culturale

Per questa implementazione, l'utente deve inquadrare il monumento di Porta Nuova e trovare il luogo in cui è stata scattata la foto. Tramite questa meccanica, si presuppone di stimolare l'interesse del giocatore verso il monumento storico andando a fargli ricercare angolazioni significative di queste ultime.

5.3 Ritagliare o Scontornare

Questa meccanica, che è stata nominata come *Ritagliare o Scontornare* indica quel tipo di meccanica di un gioco dove all'utente viene chiesto di seguire un tratto

³Sistema che descrive l'orientamento di un oggetto attraverso una serie di rotazioni a partire da un sistema di riferimento fisso xyz.

⁴Numero complesso a quattro unità, utilizzato per indicare una rotazione in Unity.

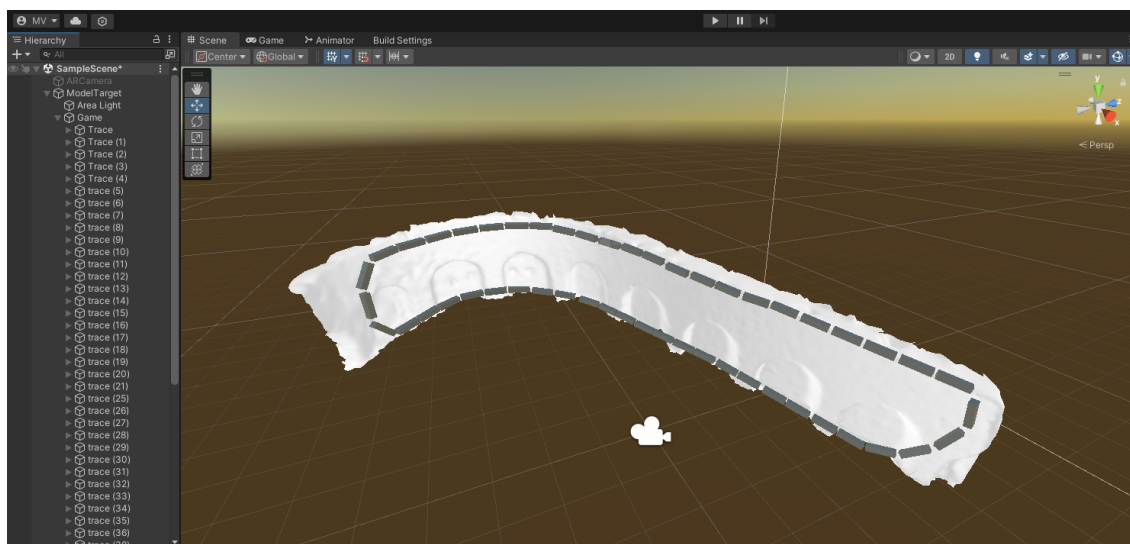


Figura 5.4: Contorno dell'oggetto (orecchio sulle mura) in Unity realizzato con i parallelepipedi mostrati in grigio

specifico per ritagliare o scontornare oggetti, si può dire che questa meccanica sia identificata dal seguire un sentiero prestabilito senza uscire dai suoi bordi.

Come ogni altra meccanica, per eseguire lo User Study è stata implementata una demo specifica con varie versioni.

Nella prima versione, una volta inquadrato il modello, in questo caso un tratto delle mura di Verona dalla forma di "orecchio di elefante"⁵ presente nel tratto tra Porta Palio e Porta Nuova, vengono mostrati sullo schermo dei parallelepipedi sintetici che lo circondano, questi costituiscono il "*sentiero*" da seguire per scontornare l'oggetto, l'utente li seguirà usando la modalità proposta dall'app. Le varianti di quest'app differiscono per quanti parallelepipedi appaiono contemporaneamente all'utente. E' stato richiesto di completare il percorso usando un numero variabile di parallelepipedi sempre maggiore: prima uno alla volta, poi quattro contemporaneamente, ed infine dieci, tenendo sempre il percorso e il numero totale di essi invariato, ovvero cinquanta.

I parallelepipedi vanno comunque acquisiti nell'ordine in cui vanno a formare il sentiero, anche se ce ne dovesse essere più di uno presente alla volta. Per acquisirli, l'utente deve inquadrarli con il centro della propria telecamera, provvisto di un apposito mirino; una volta evidenziati, i segmenti si illumineranno di giallo per indicare che sono stati completati.

Una volta completato il percorso, apparirà un'animazione del modello dell'orecchio di elefante che ruota al centro dello schermo e la scritta "Congratulations!".

5.3.1 Implementazione della Demo

Per implementare questa Demo, sono stati creati nell'editor di Unity come figli dell'oggetto da inquadrare dei parallelepipedi sintetici, che sono stati disposti nelle sue vicinanze.

La telecamera virtuale attraverso il quale l'utente osserva la scena emette costan-

⁵muro costituito da blocchi di pietra squadrati e regolarmente disposti in modo da formare una serie di protuberanze arrotondate lungo la superficie del muro, simili alle orecchie degli elefanti.

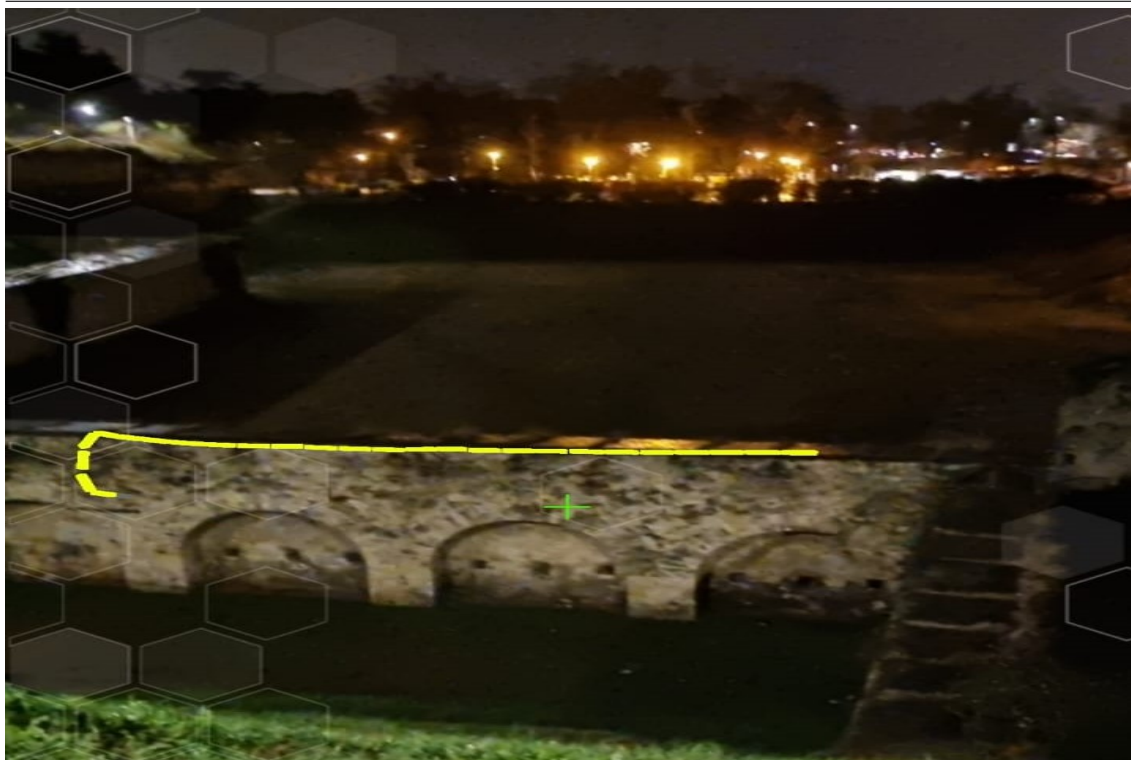


Figura 5.5: Immagine del gioco dello scontorno in azione sul tratto di mura

temente un *Raycast*, che però può generare collisioni con gli oggetti di una scena. I parallelepipedi, una volta ricevuto il *Raycast*, attivano l'apposita funzione che gli consente di illuminarsi di giallo, per indicare che sono stati completati. Una volta che un parallelepipedo è completato, verrà generato quello successivo, andando avanti così fino alla fine. Altrimenti, se sono presenti n segmenti sulla scena essi dovranno essere completati tutti prima che i prossimi n vengano generati. L'impostazione di quanti parallelepipedi vengono generati alla volta è controllabile dal menu di tutorial, presente all'inizio dell'applicazione.

Questo sistema permette una grande scalabilità nel caso si vogliano scontornare oggetti grandi, poichè il codice funziona è sufficiente inserire nuovi segmenti e collegarli alla lista di obiettivi da completare presente nell'apposito script.

5.3.2 Studio delle meccaniche presenti

Anche in questo caso, la meccanica dello scontornare può essere classificata in più modi, le più probabili sono la classificazione come meccanica di tipo "abilità", in cui si chiede al giocatore l'abilità fisica di manipolare la telecamera del telefono in modo che il centro di quest'ultimo vada ad inquadrare i segmenti, oppure come meccanica di tipo "regola", rientrando nel sottotipo operativo, in quanto ogni segmento può essere classificato come uno "stato" del gioco, e la meccanica dello scontornare regola come avviene il passaggio da uno di questi stati a quello successivo, ovvero il segmento seguente.

Per quanto riguarda sia le meccaniche di tipo "azione" che di tipo "oggetto", anche questa demo è in linea con le implementazioni precedenti trovate nelle altre applicazioni, ovvero l'utente non ha libertà di azione o di formare strategie, mentre le meccaniche di tipo "oggetto" sono quelle che descrivono lo stato dei segmenti e la

loro attivazione.

Una delle meccaniche che però differisce in questa demo sono quelle inerenti allo "spazio", infatti questa demo si può considerare avere uno spazio di tipo discreto, in quanto nonostante la telecamera e l'utente connesso ad essa si possano liberamente spostare nello spazio, i segmenti, che consistono nel vero obiettivo del gioco, non si muovono e occupano sempre la stessa posizione.

5.3.3 Inserimento nel contesto del patrimonio culturale

Per far funzionare la demo, il giocatore deve inquadrare il tratto delle mura dette a orecchio di elefante. Questa implementazione della meccanica vuole dare all'utente la sensazione di star "ritagliando" l'oggetto inquadrato, segmento dopo segmento, con l'obiettivo di generare interesse da parte di quest'ultimo verso il monumento o le sue parti ritagliate.

5.4 Movimento di un avatar

Il movimento di un avatar virtuale all'interno di un videogioco è una meccanica base che molti, se non tutti i giochi possiedono. Esistono molti modi di far muovere un avatar all'interno di un mondo sintetico, ci sono giochi che adottano per esempio un sistema di movimento libero, sia in 2D che in 3D, dove ad un tasto premuto dall'utente corrisponde una direzione di movimento, mentre altri possono adottare un sistema di movimento fisso, magari facendo spostare l'avatar tra un insieme di posizioni prestabilite in partenza, come in un gioco da tavolo, altri ancora potrebbero adottare un sistema *"punta e clicca"* dove all'utente è richiesto di cliccare il terreno nel luogo dove desidera che l'avatar si sposti, e il movimento di quest'ultimo avviene in maniera automatica creando un percorso dal punto A al punto B.

Tuttavia, quando ci si trova nel mondo della realtà aumentata, la meccanica di movimento è difficilmente implementata, in quanto si tende a inserire l'utente e la camera dello smartphone come il centro del mondo e a far compiere a quest'ultimo i movimenti attraverso la suddetta telecamera.

Ai fini di questa ricerca, si è voluto trovare il modo di far muovere l'avatar dell'utente in maniera indipendente da quest'ultimo, ricercando la maniera ideale per farlo, usando come sempre l'implementazione di una demo e le sue relative varianti.

La demo consiste in una serie di semplici parallelepipedi bianchi dove il personaggio, che in questo caso è rappresentato da un piccolo cubo, può muoversi liberamente, l'obiettivo del personaggio è prendere la capsula arancione presente su una piattaforma posta sopra di esso. Per raggiungere questa piattaforma, il cubo deve eseguire una serie di salti e spostamenti su altre piattaforme disposte a destra. Alla fine del percorso, il cubo raggiunge una piattaforma gialla che si sposta una volta che l'avatar è sopra di essa, consentendo al cubo di eseguire un ultimo salto per raggiungere la piattaforma contenente la capsula obiettivo. Una volta raggiunta, compare la scritta "Congratulations!" e il gioco termina.

Durante il percorso, il personaggio deve eseguire diverse azioni, tra cui saltare, muoversi e rimanere su una piattaforma in movimento. Sebbene queste tre azioni possano essere considerate come meccaniche di gioco distinte, sono state esaminate simultaneamente in quanto strettamente interconnesse. L'implementazione di una di queste

azioni può influenzare l'efficacia delle altre. Ad esempio, un sistema di salto ottimale in un determinato contesto potrebbe risultare inefficiente con un diverso tipo di sistema di movimento, e viceversa. E' importante quindi considerare l'interazione tra queste meccaniche e il loro impatto sul gameplay complessivo.

5.4.1 Movimento tramite tasti

In questa prima versione delle meccaniche, i movimenti dell' avatar vengono controllati tramite appositi tasti presenti sull'interfaccia del gioco, come si vede in alcuni casi di platform per dispositivi mobili. In questa implementazione, osservabile in figura 5.6, l'utente ha a disposizione sul lato destro dello schermo due pulsanti per muoversi, rispettivamente a sinistra e a destra; mentre nel lato sinistro è presente il pulsante che permette al cubo di effettuare un salto.

Il personaggio è liberamente controllabile durante l'atto del salto per consentire anche agli utenti meno esperti di coordinare meglio i movimenti. E' inoltre disponibile un pulsante denominato "reset", che consente di far ricomparire il cubo sopra la prima piattaforma, cosa che avverrà anche se il cubo dovesse cadere fuori dallo schermo. La piattaforma gialla comincerà a muoversi verso una seconda posizione quando il personaggio ci atterrà sopra, in questo caso il cubo verrà spostato con lei e l'utente non dovrà eseguire movimenti aggiuntivi per rimanerci sopra.

5.4.2 Implementazione della Demo

Nell'implementazione di questa demo, l'intero livello è figlio dell'oggetto da inquadrare, poiché questo può essere inquadrato da varie angolazioni, è stato sviluppato uno script dedicato che regola la gravità del mondo in modo dinamico, garantendo sempre un orientamento corretto del livello ed evitando comportamenti che potrebbero far cadere il cubo in modo anomalo.

Il movimento del cubo è gestito da uno script separato, il quale rileva le pressioni continue dei pulsanti sullo schermo per attivare delle flag che indicano le azioni da compiere. È importante notare che queste flag vengono lette ed eseguite nel metodo *FixedUpdate*. La lettura degli input avviene invece nel metodo *Update*: ciò è dovuto al fatto che mentre il secondo metodo viene chiamato ad ogni frame ed è quindi adatto per la lettura di eventi di input, il primo è chiamato con la frequenza del motore fisico all'interno di Unity ed è più adatto per applicare forze o modificare le componenti fisiche dell'oggetto[27].

La velocità del cubo è limitata da un valore massimo, ottenuto tramite una funzione di clamping⁶ tra la sua velocità attuale e quella massima preimpostata. Per implementare la piattaforma mobile invece, il cubo viene reso figlio di essa al contatto. Ciò consente al cubo di muoversi con la piattaforma. Quando il contatto viene perso, il cubo viene rimosso come figlio della piattaforma, consentendo al cubo di rimanere indipendente e continuare il suo movimento autonomamente.

5.4.3 Avatar corrispondente al centro dello schermo

In questa altra versione della meccanica, come si evince dal titolo della sezione, è stata proposta una implementazione dove l'avatar del giocatore segue sempre il

⁶Funzione che si occupa di limitare un valore dato in input in un intervallo specificato.



Figura 5.6: Implementazione del platform con pulsanti di movimento a schermo

centro della telecamera dell'utente, ma solo sull'asse X. In questo caso quindi l'utente dovrà per prima cosa inquadrare l'avatar per "attivarlo" e successivamente dovrà guidarlo sulla piattaforma tenendo presente che esso cercherà di andare sempre verso il centro del suo schermo. Per raggiungere il centro, l'avatar digitale utilizzerà una velocità fissa per rimanere ancorato a quest'ultimo e spostarsi verso esso quasi istantaneamente, pur essendo suscettibile alla gravità. Per eseguire un salto, in questa implementazione l'utente deve effettuare un rapido scatto col proprio telefono verso l'alto di una certa intensità.

5.4.4 Implementazione della Demo

Per creare questo tipo di meccanica, è stato generato un piano invisibile che interseca le piattaforme in maniera perpendicolare. Questo piano subisce le collisioni di un *Raycast* uscente dalla telecamera e con direzione ciò che le sta davanti. Ad ogni frame, il cubo acquisisce la coordinata di ascissa del punto di intersezione del raggio con il piano, ed utilizza la direzione data da quest'ultimo per compiere il movimento, assicurandosi di essere sempre ancorato al centro di dove l'utente sta puntando, pur risentendo l'effetto della gravità della scena e quindi potendo cadere.

Per implementare invece la meccanica del salto, lo script di movimento del personaggio è stato modificato collegando al salto non più la pressione del tasto presente nell'interfaccia, quanto il metodo di Unity chiamato *Input.Acceleration()* che utilizza l'accelerometro⁷ dello smartphone (presente ormai su tutti i dispositivi mobili) per rilevare una data accelerazione, in questo caso impostata a 10.0f sull'asse delle Y.

Il resto della demo è rimasto congruente con la variante analizzata precedentemente.

5.4.5 Studio delle meccaniche presenti

Anche per questa applicazione è possibile distinguere e studiare le meccaniche presenti dentro di essa. Per cominciare, possiamo classificare la meccanica del movimento dell'avatar e del salto come una meccaniche di tipo "azione" operativa, ovvero

⁷Dispositivo in grado di rilevare l'accelerazione in uno o più assi.

le scelte che il giocatore può compiere all'interno del gioco; è anche possibile classificarle come meccaniche di tipo "spazio". Lo spazio all'interno di questa demo è di tipo discreto, in quanto l'avatar dell'utente può effettivamente trovarsi solo sulle piattaforme presenti nella scena, uscendo da queste infatti esso viene resettato alla prima.

E' quindi possibile dare una interpretazione delle meccaniche di tipo salto e movimento come quelle che controllano il modo in cui l'avatar passa da uno spazio all'altro, ovvero di come cambia la piattaforma su cui è posizionato.

Si può considerare la meccanica della piattaforma mobile come una meccanica di tipo "attributo", in quanto va proprio a modificare il valore di un attributo di quest'ultima, ovvero la sua posizione nello spazio.

Le altre meccaniche poste all'interno del gioco sono congruenti alle precedenti applicazioni analizzate: gli oggetti e gli attributi, le regole supportate da un tutorial iniziale che spiega gli obiettivi del gioco, sono pressoché identici a ciò che si è spiegato precedentemente.

5.4.6 Inserimento nel contesto del patrimonio culturale

Per implementare questa demo nel contesto del patrimonio culturale, si è scelto di ambientarle a Porta Nuova. In questo caso, l'obiettivo è quello di generare l'interesse del giocatore al monumento inquadrato interagendo con quest'ultimo, creando un percorso dove l'avatar debba muoversi su di esso. Come scritto nella sezione successiva, questo non è stato implementato per ragioni tecniche.

Capitolo 6

Valutazione delle meccaniche e User Study

Uno degli obiettivi della ricerca consiste nell'analizzare le criticità e i punti di forza derivanti dall'integrazione delle meccaniche analizzate nella sezione precedente nel contesto della realtà aumentata, al fine di identificare quelle che risultano più promettenti e valutarne l'eventuale implementazione in un gioco basato su quest'ultima. A tal fine, è stato condotto uno studio sugli utenti coinvolgendo circa venti partecipanti, selezionati tra individui giovani e anziani, con un mix di esperienza per la tecnologia ed il mondo dei videogiochi che spazia da persone totalmente inesperte a quelle più esperte.

Lo studio consiste nella somministrazione di un questionario agli utenti, dopo che essi hanno svolto un tour tra le mura di Verona per provare tutte le demo di meccaniche generate. Il questionario è strutturato in dieci domande per ciascuna variante delle meccaniche studiate, seguite da due domande specifiche per valutare quale delle versioni complessive dell'applicazione è stata giudicata la migliore.

Esse vanno a valutare vari aspetti delle meccaniche sperimentate, concentrandosi principalmente su quanto siano state divertenti da giocare per gli utenti, la semplicità o la complessità dei controlli, la facilità o la difficoltà nel raggiungimento degli obiettivi. Vengono anche considerati altri aspetti delle meccaniche, come il loro potenziale nel valorizzare il patrimonio culturale di Verona e se il motore di realtà aumentata ha causato ostacoli durante il gioco a causa di eventuali malfunzionamenti.

Le domande del questionario comuni per tutte le applicazioni sono le seguenti:

1. Hai avuto problemi tecnici?
2. Hai avuto difficoltà con l'acquisizione del modello?
3. Com'è stata l'interazione con l'oggetto reale?
4. I comandi erano intuitivi?
5. L'applicazione ti ha divertito?
6. L'app ti ha suscitato interesse verso il monumento o l'oggetto inquadrato?
7. La difficoltà del gioco era bilanciata?

8. Quale versione ti è piaciuta di più?
9. Giocheresti un gioco che implementa la meccanica in questo modo?
10. Risposta aperta per ulteriori commenti

Per rispondere a queste domande, i partecipanti hanno utilizzato varie modalità: per la prima e la nona domanda, il questionario forniva opzioni di risposta chiuse, consentendo agli utenti di indicare se i problemi tecnici fossero gravi, moderati o assenti. Nella nona domanda, in particolare, i partecipanti potevano rispondere "sì", "no", oppure "sì, ma solo se la meccanica in questione non fosse predominante nel gioco".

Per le altre domande, è stata utilizzata una scala numerica da uno a cinque, dove uno indicava la risposta minima e cinque quella massima. L'ultima domanda era aperta, consentendo ai partecipanti di aggiungere eventuali commenti o osservazioni aggiuntive.

Di seguito saranno riportati i risultati di ciascuna applicazione sotto forma di tabella, seguiti da una relativa discussione su quest'ultimi.

6.1 Movimento di un avatar

Il monumento utilizzato per entrambe le versioni dell'applicazioni è Porta Nuova.

Movimento di un avatar		
Domanda	Comandi a schermo	Comandi della telecamera
Problemi tecnici	10%	20%
Difficoltà con l'acquisizione	5%	10%
Interazione con oggetto reale	non vi è stata	non vi è stata
Comandi intuitivi	4.3	3.9
Applicazione divertente	4.6	3.8
Interesse verso oggetto inquadrato	3.6	3.6
Difficoltà del gioco bilanciata	3.8	4.2
Giocheresti a questo gioco	66%	58%

Tabella 6.1: Tabella riassuntiva del sondaggio relativo all'applicazione del platform.

L'applicazione che tratta del movimento di un avatar è divisa nelle sue due varianti, come da figura 6.1: la prima con i comandi a schermo e la seconda con la telecamera che controlla la posizione dell'avatar.

Per quanto riguarda le prime due domande, che mostrano le percentuali di partecipanti che hanno fornito una risposta positiva, si può notare che non si sono riscontrati molti problemi su questo fronte, probabilmente poiché Porta Nuova è un monumento grande e ben inquadrabile da una telecamera, perciò il motore non ha avuto difficoltà a riconoscerlo.

Per quanto riguarda la sezione dei comandi, è emerso che i partecipanti hanno trovato

i comandi posti sullo schermo molto più intuitivi rispetto a quelli che coinvolgevano l'inquadratura e l'accelerometro per controllare il personaggio, nonostante questi comandi basati sull'inquadratura e sull'accelerometro non sono stati considerati frustranti. Questa differenza potrebbe essere attribuita al fatto che le meccaniche di gioco simili a quelle implementate nella prima versione ricordano molti giochi platform familiari alla maggior parte degli utenti, anche a coloro che non sono molto esperti di tecnologia, che di conseguenza potrebbe aver portato ad una certa difficoltà iniziale nell'abituarsi a un movimento legato esclusivamente all'inquadratura. Inoltre, sono state segnalate alcune difficoltà nell'uso del salto tramite l'accelerometro, con alcuni partecipanti che hanno avuto difficoltà a eseguire il comando correttamente. Questo suggerisce che tale meccanica potrebbe richiedere una revisione e un'implementazione diversa per migliorarne l'usabilità.

Dal punto di vista del divertimento, l'applicazione è risultata molto divertente per la maggior parte degli utenti, segnalando che però la prima è stata più divertente della seconda, nonostante il percorso da seguire fosse identico per entrambe. Questa differenza potrebbe essere nuovamente attribuita alla percezione dei comandi di gioco e alla loro maggiore difficoltà nella seconda versione rispetto alla prima. Tale conclusione è supportata anche dalla valutazione della difficoltà del gioco, dove la scala va da uno a cinque, con uno che indica un livello di difficoltà troppo facile e cinque troppo difficile: infatti per entrambe le varianti dell'applicazione si nota che la curva di difficoltà percepita è inclinata verso un valore di difficoltà superiore, che si nota nella seconda variante. Infine una domanda aggiuntiva rivolta agli utenti che hanno valutato la difficoltà come troppo elevata ha rilevato che il 60% di coloro che hanno assegnato un voto di quattro o più hanno indicato i controlli come una delle cause di questa percezione.

Per quanto riguarda l'interesse verso i beni culturali, i dati mostrano che l'app è riuscita a scatenare un buon interesse nel monumento inquadrato, essendo esso di sfondo al gioco, e potrebbe essere usata quindi per incentivare un percorso turistico. Per quanto riguarda l'interazione con quest'ultimo essa è purtroppo risultata assente: questo è dovuto al fatto che inizialmente il piano per questa demo era quello di dover far muovere il personaggio sul monumento per un tratto di percorso.

Tuttavia ciò non è risultato possibile poichè i modelli 3D usati per il riconoscimento sono pieni di imperfezioni che il modello del personaggio non sarebbe riuscito a superare; tuttavia questo potrebbe essere facilmente risolto tramite un modellatore 3D che riesca a pulire il modello e lisciare le parti interessate.

Complessivamente, tramite la domanda finale apposita, è evalso che la variante coi comandi a schermo è piaciuta di più rispetto alla seconda proposta, con il 64.3% degli utenti che hanno votato per la prima. Tuttavia, questo non implica che la variante con la telecamera sia meno valida, ma che probabilmente ha bisogno di altri studi e accorgimenti per poter funzionare al meglio, anche perchè alla domanda posta "ti interesserebbe giocare un gioco con queste meccaniche" circa il 70% degli utenti ha risposto in modo affermativo, mentre il restante ha risposto sì ma se non fosse la prevalente.

6.2 View Matching

Il monumento utilizzato per questa applicazione è stato Porta Nuova.

View Matching		
Domanda	Immagine nel riquadro	Immagine in sovraimpressione
Problemi tecnici	10%	0%
Difficoltà con l'acquisizione	10%	20%
Interazione con oggetto reale	3.5	3.5
Comandi intuitivi	4.5	4
Applicazione divertente	3	2.3
Interesse verso oggetto inquadrato	4.5	4.5
Difficoltà del gioco bilanciata	2.3	2.9
Giocheresti a questo gioco	62%	55%

Tabella 6.2: Tabella riassuntiva del sondaggio relativo all'applicazione di view matching.

I risultati di questa applicazione, come visto da figura 6.2, hanno fatto prevalere che dal lato dei controlli la demo risultava ben fatta: questo perché i giocatori probabilmente dovevano semplicemente muoversi con la telecamera verso una angolazione predefinita. La stessa osservazione vale per la difficoltà, che qui è risultata leggermente troppo facile rispetto alla media ideale di tre. Questo risultato era atteso in quanto è stata scelta appositamente una posizione della foto facilmente trovabile nell'ambiente per concentrarsi maggiormente sulle due versioni, ovvero alla posizione migliore della foto sopra l'interfaccia utente.

Per quanto riguarda questo ultimo punto, dai dati ottenuti è possibile capire che è stata preferita la versione con la foto da cercare nel riquadro piuttosto che quella in sovraimpressione, probabilmente poiché la prima versione risulta meno intrusiva nel posizionarsi con la fotocamera, come evidenziato dalla differenza dei voti dati sia ai comandi intuitivi, sia al divertimento dell'applicazione, oltre che alla domanda finale, dove nessuno ha preferito la seconda versione, mentre il 71,4% dei partecipanti non ha notato grandi differenze tra le due.

Il fattore del divertimento causato dall'applicazione in questo caso è molto basso: se da un lato la differenza tra le due versioni è data probabilmente dall'immagine in sovraimpressione, che rendeva la telecamera più difficile da osservare in quanto occupata con un'altra immagine, in generale il poco divertimento degli utenti potrebbe essere ricondotto alla eccessiva semplicità del gioco, è possibile ipotizzare infatti che, fornendo al giocatore degli scorci particolari o significativi ma più difficili da trovare, esso possa divertirsi di più in quanto sarà spinto ad esplorare l'ambiente che lo circonda.

Contrariamente a quanto avvenuto per la precedente applicazione, l'interazione con l'oggetto in questo caso è avvenuta registrando un voto identico per entrambe le versioni, questo voto è ipotizzabile dipenda dal fatto che l'interazione si fermi al solo dover osservare il monumento in alcune sue angolazioni.

Degno di nota però è l'interesse generato verso quest'ultimo da entrambe le versioni, anche qui registrato in modo uguale, probabilmente perchè gli utenti non hanno trovato grandi differenze tra esse. Questo dato fa capire che questa meccanica potrebbe essere in grado di suscitare un interesse per chi la utilizza su ciò che stanno ricercando visivamente. Se questo venisse magari accompagnato ad una spiegazione del monumento, magari con uno scorcio significativo di quest'ultimo, potrebbe essere utilizzata per valorizzare il patrimonio culturale.

Alla domanda che chiede se i partecipanti giocherebbero un gioco di questo tipo, solo il 62% e 55% hanno risposto di sì, dove la maggior parte però ha detto che lo farebbe solo se questa non fosse la meccanica principale. Dati questi rilevamenti, si può affermare che queste meccaniche di gioco creerebbero più divertimento se inserite in un gioco che ne contiene anche di diverse.

6.3 Lancio con fisica

Per questa demo, è stato utilizzato uno dei cannoni presenti tra le mura di Verona come modello da acquisire.

Lancio con fisica		
Domanda	Traiettoria mostrata	Traiettoria nascosta
Problemi tecnici	30%	20%
Difficoltà con l'acquisizione	10%	5%
Interazione con oggetto reale	3.8	3.5
Comandi intuitivi	4	3.2
Applicazione divertente	3	2.1
Interesse verso oggetto inquadrato	3	3
Difficoltà del gioco bilanciata	3.5	4.7
Giocheresti a questo gioco	60%	40%

Tabella 6.3: Tabella riassuntiva del sondaggio relativo all'applicazione di Lancio con fisica.

Per l'implementazione di queste meccaniche, ovvero il lancio di un oggetto avente la gravità nelle sue due versioni, il focus è stato quello di valutare se l'aggiunta della traiettoria visibile dell'oggetto prima del lancio potesse andare a costituire un elemento necessario, positivo oppure negativo per l'esperienza di gioco.

I dati raccolti relativi alla difficoltà mostrano che i partecipanti si sono trovati decisamente molto meglio con la versione che aveva la traiettoria a schermo, trovando di difficoltà medie quest'ultima, mentre per la versione senza la difficoltà è stata percepita molto difficile. Nella domanda che permetteva di inserire commenti aggiuntivi, alcuni partecipanti infatti hanno scritto che senza la traiettoria non riuscivano ad avere una percezione della profondità, e che sembrava di colpire l'obiettivo mentre la sfera gli passava dietro o di fronte. Questo dato è molto significativo per capire uno dei problemi di implementazione di questa meccanica, ovvero che vi è bisogno di una sensazione di profondità quando si lanciano degli oggetti avendo una fisica, e che senza l'esperienza può risultare disorientante per l'utente.

Gli altri dati raccolti da questa demo sono in linea con quanto appena detto, l'indice di divertimento è nella media per quanto riguarda la prima versione, forse perchè

la demo in sè non dura molto e non sono presenti molti obiettivi da colpire, mentre è significativamente più basso nella seconda, fattore probabilmente dovuto alla frustrazione degli utenti nello sbagliare incessantemente la mira con il proiettile a causa della mancanza di profondità.

Per quanto riguarda i controlli presentati, gli utenti sono riusciti facilmente a capire come si poteva lanciare la sfera tramite un adeguato tutorial: infatti questo tipo di lancio infatti è molto simile al lancio di una fionda, ovvero che più si porta in basso il proiettile, più questo andrà ad eseguire una parabola in alto. Anche in questo caso bisogna segnalare che la versione senza la traiettoria è stata più difficile da comprendere, probabilmente perché senza eseguire dei tiri di prova non si può capire come si modifica la traiettoria in base a come si posiziona inizialmente la sfera.

Inoltre, i dati mostrano che l'interesse generato da questo tipo di meccanica per il monumento è stato nella media, anche se in questo caso l'interazione con quest'ultimo è stata registrata molto più alta delle precedenti demo. Questo rilevamento può essere dovuto ad alcune cause principali: per prima cosa, la difficoltà percepita troppo elevata della demo potrebbe aver portato alcuni utenti a disinteressarsi al gioco, mentre potrebbe aver influito anche il monumento scelto: infatti questa demo infatti è stata svolta sui cannoni presenti sulle mura, che sicuramente possono suscitare un impatto visivo minore rispetto a Porta Nuova, che è stata visitata precedentemente. Ultimi dati da segnalare sono l'insorgenza maggiore di problemi tecnici e difficoltà di acquisizione del modello: alcuni utenti hanno infatti lamentato dei rallentamenti e scatti quando spostavano la sfera. Questi possono essere spiegati dal fatto che quest'app richiede un lavoro più intensivo del telefono per essere utilizzata rispetto alle altre due, in quanto deve simulare una traiettoria parabolica ad ogni frame del sistema, oltre che eseguire varie interpolazioni, e perciò potrebbe dare dei problemi su dei dispositivi meno potenti; problema però facilmente risolvibile con una implementazione più ottimizzata della demo. Le difficoltà nell'acquisire il modello potrebbero essere dovute al fatto che il modello scelto, cioè i cannoni, si presentano molto vicini alla telecamera e possono essere inquadrati da molte posizioni diverse, fatto che sperimentalmente ha causato problemi con il motore di Vuforia.

Come ultimo dato significativo, la ricerca ha mostrato che la percentuale di utenti che vorrebbe vedere questa meccanica implementata in un gioco è bassa, andando a sfiorare il 40% con la seconda versione. Quindi, questi dati fanno ipotizzare che la traiettoria del proiettile sia necessaria quando si eseguono questo tipo di meccaniche nella realtà aumentata, a meno che non si abbiano modi diversi per implementare una sensazione di profondità nel gioco.

6.4 Lancio senza fisica

Per implementare la demo del lancio, sono stati utilizzati ancora una volta i cannoni delle mura di Verona.

Lancio senza fisica		
Domanda	Lancio al centro	Lancio nel touch
Problemi tecnici	0%	10%
Difficoltà con l'acquisizione	0%	5%
Interazione con oggetto reale	non vi è stata	non vi è stata
Comandi intuitivi	4.7	3.9
Applicazione divertente	4.6	3.8
Interesse verso oggetto inquadrato	3	3
Difficoltà del gioco bilanciata	3.2	4.2

Tabella 6.4: Tabella riassuntiva del sondaggio relativo all'applicazione di Lancio senza fisica.

Per questa implementazione, la componente più grande da valutare sono stati i controlli, ovvero se lo shooting della prima versione, dove l'utente deve mirare con la telecamera in quanto il proiettile viene sparato al centro dello schermo sia migliore o peggiore della seconda, dove l'utente ha il controllo aggiuntivo di poter indicare dove andare a sparare nello schermo. A tale proposito, notiamo che la prima versione ha riscosso voti più positivi rispetto alla seconda su questo campo. Una possibile spiegazione per la differenza tra queste due valutazioni sono alcuni commenti lasciati nella risposta aperta e uditi durante la prova, ovvero che il dover cliccare sullo schermo da parte del giocatore il luogo in cui si vuole sparare genera dei proiettili non precisi, in quanto non si può ben interpretare il punto in cui il dito tocca lo schermo, oltre al fatto che il dito stesso copre la visuale verso l'obiettivo a cui si deve sparare. Questi controlli quindi possono risultare frustranti, in quanto il giocatore deve sia inquadrare gli obiettivi con la telecamera del telefono, sia cliccarci sopra con il dito usando l'altra mano, il che suggerisce che l'implementazione del proiettile al centro sia la variante più indicata quando si chiede al giocatore di spostare frequentemente il dispositivo.

Per quanto riguarda la componente del divertimento, questa meccanica è stata tra quelle che hanno intrattenuto di più l'utente, probabilmente poiché si trattava di una meccanica implementata in modo molto simile ad un gioco vero e proprio, il che sarebbe in linea con quanto analizzato per i risultati del movimento di un avatar. Un altro fattore che ha portato ad un altro grado di divertimento potrebbe essere stata la difficoltà, percepita in questo caso come bilanciata nella prima versione e leggermente difficile nella seconda. Gli utenti non hanno provato frustrazione, in quanto le navicelle aliene da colpire erano difficili da colpire all'inizio, ma dato che percorrono un percorso sempre uguale, ad eccezione di alcune che si muovono comunque entro un raggio prestabilito abbastanza ristretto, possono essere anticipate ed il gioco quindi superato anche da chi possiede meno riflessi.

La difficoltà maggiore della seconda versione si può attribuire ai controlli, come afferma circa l'80% degli utenti che ne ha dato un voto superiore, e anche dal fatto che le navi da colpire sono le stesse. Si potrebbe quindi ipotizzare di dover semplificare il gioco se si vuole adottare questo tipo di implementazione della meccanica.

Per quanto riguarda il contesto dei beni culturali, anche in questo caso non c'è stata un'interazione con l'oggetto inquadrato perché ai fini del gioco non si è creduto molto significativo far colpire ai giocatori dei punti fissi fermi sopra l'oggetto inquadrato, similmente a quanto avvenuto con il lancio provvisto di fisica. Segnaliamo anche che l'interesse generato da questa applicazione al monumento è congruente con quello della sua variante con fisica, questo è riconducibile sia al fatto che non vi è stata

interazione con quest'ultimo, sia che queste due applicazioni si sono svolte una di seguito all'altra per motivi di praticità, e perciò l'opinione dei partecipanti potrebbe essere rimasta invariata.

Lo studio ha rilevato che questa meccanica può risultare molto divertente se adattata con la giusta difficoltà e i metodi di controllo giusti, infatti circa il 70% degli utenti ha indicato che giocherebbe ad un gioco che implementa questa meccanica, e di questi solamente il 20% ha citato che lo farebbe solo se questa non fosse la predominante. Anche se la seconda versione è andata peggio della prima, non significa che con qualche aggiustatura o miglioramento non si possano trovare alternative valide alla prima. Tuttavia, per valorizzare il patrimonio culturale, si deve trovare un contesto in cui questo tipo di meccanica abbia senso e che metta in risalto il monumento, magari creando un'interazione con esso per fare in modo che i giocatori vi si interessino.

6.5 Scontornare

Per questa ultima demo, è stato usato come modello una sezione delle mura di Verona che possiede un tratto detto ad "orecchio di elefante".

Scontornare	
Domanda	Tutte le versioni
Problemi tecnici	10%
Difficoltà con l'acquisizione	30%
Interazione con oggetto reale	4.5
Comandi intuitivi	4.5
Applicazione divertente	2.8
Interesse verso oggetto inquadrato	4.5
Difficoltà del gioco bilanciata	1.8

Tabella 6.5: Tabella riassuntiva del sondaggio relativo all'applicazione dello scontornare.

Questa applicazione consiste nell'andare a seguire un tracciato di segmenti prestabiliti attorno all'orecchio in modo da generare una sensazione di "scontorno" di quest'ultimo. Come si nota dalla tabella riportata 6.5, vi sono stati diversi problemi nell'acquisizione di questo monumento: ciò potrebbe essere dato dal fatto che il muro si trova ad una distanza non poco rilevante dal punto di osservazione prestabilito, che ha generato alcune difficoltà nella creazione del modello durante la fase di design del gioco; ma tutti fattori facilmente risolvibili tramite il lavoro di un modellatore 3D più esperto.

Dal punto di vista dei controlli, questa applicazione ha riscontrato un feedback positivo dagli utenti, che si sono trovati a loro agio dopo i primi segmenti da inquadrare, fatto probabilmente dovuto alla assenza di movimenti complicati da compiere o tasti da premere. Altri feedback molto positivi si sono riscontrati con le interazioni sull'oggetto reale e l'interesse verso quest'ultimo prodotto; questa meccanica è forse quella che va maggiormente ad interagire con l'oggetto tra quelle proposte in quanto si svolge interamente su di esso, e l'utente quindi ha la possibilità di percepire davvero come poco a poco lo stesse scontornando, il che ha suscitato grande interesse verso ciò che si stava guardando. E' possibile affermare quindi che questa meccanica possa essere applicata facendo ritagliare ai giocatori alcune sue parti significative

per poi visualizzarle a schermo mentre ne viene raccontata la storia.

Uniche note negative di questa demo sono state il divertimento percepito e la difficoltà del gioco, probabilmente legate tra loro. Questa applicazione è risultata la più facile tra quelle proposte, in quanto probabilmente non presentava nessuna condizione di sconfitta o perdita, che invece erano presenti anche in minima parte nelle altre: ad esempio se l'utente nella prima demo riguardante il movimento dell'avatar faceva cadere quest'ultimo dalle piattaforme era costretto a ripetere il percorso, mentre nei lanci l'utente poteva sbagliare il colpo, che lo costringeva a riprovare. In questo caso invece, l'utente deve semplicemente seguire un percorso prestabilito, senza possibilità di commettere sbagli magari deviando o perdendo la traccia. Applicare questi accorgimenti e aggiungere degli stati di fallimento potrebbe aumentare il tasso di difficoltà percepito, e di conseguenza il divertimento dell'utente, che si può ipotizzare risulti molto basso per lo stesso motivo.

Questa meccanica di gioco può essere molto valida se inserita in una applicazione riguardante i beni culturali, in modo da far interagire il giocatore in modo diretto con i monumenti osservati, però se questa viene inserita senza possedere una vera difficoltà per il giocatore, rischia di causare noia risultando solo uno spreco di tempo per quest'ultimo.

Capitolo 7

Conclusioni

In conclusione, si può affermare che il contesto della realtà aumentata sia molto ampio. Questa nuova tecnologia, che fa affidamento a processi di computer vision come la stima della posa, viene utilizzata in moltissimi contesti ed ambiti diversi e possiede svariati modi in cui può essere implementata.

La ricerca si occupa di esplorare l'implementazione di tipo Model Based, cioè che fa affidamento ad un oggetto reale come marcatore per posizionare quelli sintetici nella scena, ed esplora la sua declinazione nel mondo dell'intrattenimento digitale, ovvero i videogiochi.

Essa ha come obiettivo quindi quello di studiare e ipotizzare come alcune meccaniche di gioco ritrovabili molto spesso in questi ultimi possono essere applicate nel contesto della realtà aumentata, implementandole in delle demo apposite con la funzione di testarle in alcune loro varianti implementative.

Tramite i risultati di questo studio, emergono alcune potenzialità e difetti in ciascuna di loro: il lancio di un oggetto, sia nella variante senza gravità che nella variante che la possiede, ha dimostrato di essere una meccanica potenzialmente divertente nelle sue implementazioni, dovendo però ancora ricercare il metodo corretto per integrarla al meglio in questo contesto, sapendo bilanciare la sua difficoltà in modo che essa non risulti troppo frustrante, come nel caso del lancio con gravità senza traiettoria, ma che riesca a mettere in luce l'oggetto inquadrato in qualche modo.

La meccanica dello scontornare è una di quelle più apprezzate dal pubblico che l'ha testata, infatti questa meccanica ha molto potenziale se inserita nell'ambito dei beni culturali, in quanto benché si tratti di una meccanica priva di una vera difficoltà, il suo uso è ottimo per far interagire l'utente con l'oggetto in modo diretto e suscitare interesse per quest'ultimo.

La meccanica del View Matching risulta estremamente valida nel contesto del patrimonio culturale, avendo la potenzialità di mettere in risalto scorci specifici o lati dei monumenti significativi, integrandoli in un gioco che sa divertire l'utente a condizione che le foto da ricercare siano poste in luoghi non ovvi o banali; tuttavia, essa manca di una vera interazione con l'oggetto che non sia inquadrarlo.

Infine, la meccanica di movimento e salto dell'avatar si sono rivelate estremamente valide, riuscendo a mantenere alto il divertimento durante la loro esecuzione. Con opportuni aggiustamenti e ricerche ulteriori sui controlli dell'avatar, poiché che quelli studiati non hanno riscontrato la giusta efficacia, ed un aggiustamento dei modelli per garantire che ci possa essere un effettiva interazione con gli oggetti inquadrati, questa applicazione potrebbe risultare estremamente valida per un gioco in realtà

aumentata.

Tramite i risultati della ricerca ottenuti, si può effettuare un lavoro di ulteriore studio delle meccaniche qui proposte, oltre che poterne ideare ed adattare di nuove, aiutandosi con la tassonomia riportata, con l'obiettivo di creare giochi veri e propri che le sappiano adottare in una misura che valorizzi il contesto in cui si trovano.

Tramite le meccaniche studiate, si potrebbero implementare alcuni tipi di giochi in AR, come per esempio un platform più completo, di cui l'implementazione del movimento studiata è già una demo, ma anche un Tower Defence, ottenuto combinando il lancio di un oggetto con fisica con nuove meccaniche svolte su un'area molto grande; oppure ancora si potrebbe pensare ad un gioco di shooting, integrando però gli oggetti inquadrati. Altri esempi di implementazione potrebbero essere giochi che permettono all'utente di disegnare in realtà aumentata, modificando la meccanica dello scontornare, oppure dei puzzle game usando quella del view matching, ecc.

Sicuramente uno degli aspetti che di più trae beneficio da questo studio è quello di una applicazione apposita pensata per il turismo, dove per incentivare l'interesse degli utenti per il percorso quest'ultimo viene presentato come una collezione di minigiochi, diversi per ogni tappa da visitare. Una delle applicazioni che mira a questo risultato è la già accennata MuraAR, che deve essere ancora rilasciata e che vuole occuparsi proprio di valorizzare il tratto di mura della città di Verona.

Bibliografia & Sitografia

- [1] Amanda Edwards-Stewart, Tim Hoyt e Greg M. Reger. «Classifying Different Types of Augmented Reality Technology». In: *National Center for Telehealth & Technology, U.S. Defense Health Agency b Veterans Health Administration, Puget Sound Health Care System* (2016).
- [2] Nextech3D.ai. *What Are The Different Types of Augmented Reality?* 25 Maggio 2022. URL: <https://www.nextechar.com/blog/what-are-the-different-types-of-augmented-reality>.
- [3] URL: <https://www.semanticscholar.org/paper/Model-Based-Tracking-for-Augmented-Reality-on-Raj-Lowney/0751d20c6f0f040dda44c5143a7643638a963fc5>.
- [4] Claudio Pomes. *Le Indicazioni a piedi in Realtà aumentata con Google Maps*. 18 Dicembre 2023. URL: <https://www.navigaweb.net/2019/08/le-indicazioni-piedi-in-realta.html>.
- [5] Rod Reynolds. *Augmented reality frameworks: types, elements, and trends*. 18 Dicembre 2023. URL: <https://www.ienhance.co/insights/augmented-reality-frameworks-types-elements-and-trends>.
- [6] Reinhard Klette. *Concise Computer Vision*. London Springer, 2014.
- [7] URL: https://www.researchgate.net/figure/Pinhole-camera-model-with-a-point-PX-Y-Z-according-to-the-camera-coordinate-system_fig2_326717734.
- [8] Richard Szeliski. *Computer Vision: Algorithms and Applications 2nd Edition*. 2022.
- [9] Alain Pagani, Johannes Koehler e Didier Stricker. «Circular Markers for camera pose estimation». In: (2011).
- [10] ShanshanWei, Zhiqiang He e Wei Xie3. «Relative Pose Estimation Algorithm with Gyroscope Sensor». In: (2017).
- [11] M.N. Alkhatib, A.V. Bobkov e N.M. Zadoroznaya. «Camera pose estimation based on structure from motion». In: (2021).
- [12] Yoli Shavit e Ron Ferens. «Introduction to Camera Pose Estimation with Deep Learning». In: (2019).
- [13] Fiore Paul B. *A Custom Computing Framework for Orientation and Photogrammetry*. 2000.
- [14] URL: https://www.researchgate.net/figure/Camera-pose-estimation-from-correspondences-Even-though-the-lighting-is-bad-we-can_fig1_340167758.
- [15] Wharton Staff. *How “Pokemon Go” Took Augmented Reality Mainstream*. July 21, 2016. URL: <https://knowledge.wharton.upenn.edu/article/how-pokemon-go-took-augmented-reality-mainstream/>.
- [16] *Enciclopedia Treccani*.
- [17] Jesse Schell. *The Art of Game Design: A book of lenses*. 24/11/2014.

- [18] Andrew K. Przybylski, Richard M. Ryan e C. Scott Rigby. «A Motivational Model of Video Game Engagement». In: *American Psychological Association* (2010).
- [19] URL: <https://cheesecakelabs.com/blog/educational-games-future-programming/>.
- [20] Jeanne H. Brockmyer, Christine M. Fox, Kathleen A. Curtiss, Evan McBroom, Kimberly M. Burkhart e Jacquelyn N. Pidruzny. «The development of the Game Engagement Questionnaire: A measure of engagement in video game-playing». In: *Journal of Experimental Social Psychology* (12/03/19).
- [21] URL: <https://developer.tobii.com/pc-gaming/design-guidelines/>.
- [22] Maksims Kornevs, Jannicke Baalsrud Hauge, Sebastiaan Meijer, Katherine Smith, Yuzhong Shen, Anthony Dean, Mirko Suznjevic, Maja Homen, Mifrah Ahmad, Zuzana Václavíková, Francesco De Pace, Andrea Sanna, Federico Manuri, Iñigo Lerga Valencia, Dionysios Manesis, Adrián Domínguez, Daniel Vallés, Luis de Marcos, Juan Aguado e Sergio Caro. *Game Design and Intelligent Interaction*. IntechOpen, 2020.
- [23] URL: <https://profs.info.uaic.ro/~evalica/hci2010/L13/>.
- [24] Parlett David. *RULES OK*. 2005. URL: <https://www.parlettgames.uk/gamester/rulesOK.html>.
- [25] URL: <https://www.parlettgames.uk/gamester/rulesOK.html>.
- [26] *Unity User Manual 2022.3*.
- [27] *Unity User Manual 2022.3*.