

POLITECNICO DI TORINO

---

Corso di Laurea Magistrale  
in Ingegneria del Cinema e dei Mezzi di Comunicazione

Tesi di Laurea Magistrale

ANALISI E SVILUPPO PRATICO DELLA PIPELINE DI LAVORO  
PER IL GAMING VR



Relatore:  
prof. Riccardo Antonio Silvio Antonino

Candidato:  
Luca Maglione

---

ANNO ACCADEMICO 2017–2018



*"Con i videogiochi puoi esprimere quello che vuoi, esattamente come con il cinema.*

*Ci sono film che hanno solo bombe ed esplosioni, e ci sono film che hanno grandi storie e che ti spingono a riflettere su te stesso e sulla vita.*

*Dipende da quello che i creatori vogliono esprimere e sperimentare, da ciò che vogliono dare alla gente."*

---

Hideo Kojima  
da un'intervista a Wired (2014)



# Indice

<b>Introduzione</b>	<b>9</b>
<b>I Nuovi mezzi narrativi</b>	<b>11</b>
<b>1 Il videogioco come narrazione</b>	<b>13</b>
1.1 L'evoluzione della tecnologia . . . . .	14
1.1.1 Gli albori . . . . .	14
1.1.2 Le Home Console . . . . .	15
<b>2 Un nuovo modo di raccontare</b>	<b>19</b>
2.1 Esempio di narrazione complessa nel videogioco: <i>Metal Gear Solid 3</i> . . . . .	20
2.2 Esempio di arte nell'estetica del videogioco: <i>Okami</i> . . . . .	27
2.3 Esempio di metanarrazione nel videogioco: <i>Undertale</i> . . . . .	28
2.4 Il gaming di massa . . . . .	30
2.5 Semplificare le esperienze . . . . .	34
<b>3 La realtà virtuale</b>	<b>35</b>
3.1 L'evoluzione della VR . . . . .	36
3.2 La tecnologia della VR . . . . .	39
3.2.1 Rendering stereoscopico . . . . .	43
3.2.2 Audio stereoscopico . . . . .	44
3.3 <i>Samsung Gear VR</i> . . . . .	45
3.4 Gaming VR . . . . .	46
3.4.1 La lenta diffusione . . . . .	46
3.4.2 Un nuovo livello d'immersione . . . . .	48
3.4.3 Le problematiche . . . . .	50
3.4.4 Differenze d'approccio rispetto al gaming tradizionale . . . . .	53
3.5 Il futuro del gaming VR . . . . .	54

---

<b>II</b>	<b>Il videogioco "made in Italy"</b>	<b>57</b>
<b>4</b>	<b>Il panorama italiano</b>	<b>59</b>
4.1	Le migliori produzioni italiane . . . . .	60
4.2	Sviluppatori Indie in Italia . . . . .	63
4.3	KR Games . . . . .	64
4.3.1	Intervista a Kabir Ferro e Roberto Comella . . . . .	65
<b>III</b>	<b>Lo sviluppo di un videogioco in VR presso KR Games</b>	<b>73</b>
<b>5</b>	<b>Un'introduzione alla modellazione 3D</b>	<b>75</b>
5.1	Modellazione 3D manuale . . . . .	80
5.1.1	Modellazione "per spostamento di elementi" . . . . .	81
5.1.2	Metodo della "texture piana" . . . . .	82
5.1.3	Metodo "a tela di ragno" . . . . .	82
5.1.4	Metodo "a rifinitura progressiva" . . . . .	83
5.1.5	Metodo "per <i>displacement map</i> " . . . . .	84
5.1.6	Metodo "per scultura 3D" . . . . .	86
5.2	L'importanza della funzione del modello 3D . . . . .	86
<b>6</b>	<b>Un'introduzione all'animazione</b>	<b>87</b>
6.1	L'animazione convenzionale . . . . .	87
6.1.1	Differenti tecniche . . . . .	89
6.2	L'animazione 3D . . . . .	91
6.3	I dodici principi dell'animazione . . . . .	92
6.4	Un approfondimento sull'animazione 3D . . . . .	94
6.4.1	Animazione con armature . . . . .	96
<b>7</b>	<b>La mia esperienza presso KR Games: organizzazione e tecnologia usata</b>	<b>103</b>
7.1	Il <i>workflow</i> . . . . .	104
7.2	Comuni criticità delle applicazioni VR . . . . .	107
7.3	I software utilizzati . . . . .	110
7.3.1	<i>Blender</i> : le caratteristiche che ne hanno decretato la scelta . . . . .	110
7.3.2	<i>Unity 3D</i> : le caratteristiche che ne hanno decretato la scelta . . . . .	120
<b>8</b>	<b>Lo sviluppo di <i>The Villa: Allison's diary</i> presso KR Games</b>	<b>125</b>
8.1	La trama e l'atmosfera . . . . .	125
8.2	Creazione degli ambienti . . . . .	130
8.3	Creazione dei personaggi . . . . .	137
8.3.1	Modellazione . . . . .	137
8.3.2	Applicazione delle armature . . . . .	140
8.3.3	Animazione . . . . .	143
8.4	Implementazione dei modelli . . . . .	145

## INDICE

---

8.5 La distribuzione . . . . .	145
<b>Conclusioni</b>	<b>147</b>
<b>Ringraziamenti</b>	<b>149</b>



# Introduzione

Alla base di questo studio vi è l'analisi del mondo videoludico in *realtà virtuale* e del suo sviluppo nel tempo come mezzo narrativo.

In particolare, si pone l'attenzione su come vengono sviluppati i moderni videogiochi in VR<sup>1</sup> e le tecniche che si utilizzano nell'ambito della modellazione e dell'animazione 3D, analizzandone le differenze rispetto a quanto visto per videogames fruibili su un normale schermo.

Si vuole anche capire perché le grandi potenzialità narrative della realtà virtuale non riescano ancora ad emergere a confronto di quelle del tradizionale videogame.

A spingermi ad approfondire questo tema è il fatto di essere da sempre un grande appassionato del mondo videoludico che seguo non come semplice hobby, ma come una vera passione, tanto che questo è diventato il mio ambito lavorativo.

Userò, quindi, la mia esperienza nel mondo del lavoro come animatore e modellatore 3D, presso la casa di sviluppo software per realtà virtuale KR Games, come mezzo per approfondire ulteriormente la mia analisi e mostrare come il mondo della VR viene visto da chi ci lavora a stretto contatto.

L'obiettivo di questa tesi non è solo analizzare lo sviluppo di un prodotto per VR, in special modo nelle fasi di modellazione e animazione delle componenti digitali, ma vedere se questo possa integrarsi come medium narrativo al fianco del videogioco classico.

La tesi si sviluppa in tre grandi sezioni: nella prima percorreremo brevemente la storia del videogioco e della realtà virtuale, spiegandone le rispettive caratteristiche e analizzando qualche esempio concreto delle possibilità narrative offerte da questi mezzi.

Nella seconda guarderemo al mondo videoludico italiano, analizzando alcune sue peculiarità grazie anche ad un'intervista ai fondatori di KR Games.

Proprio il mio lavoro presso di questa azienda sarà fulcro della terza sezione, in cui analizzeremo tecniche d'animazione e software usati oltre ad analizzare passo dopo passo come viene sviluppato un modello 3D per applicazioni in VR.

---

<sup>1</sup>Virtual Reality

## INTRODUZIONE

---

Grazie a questa esperienza lavorativa a stretto contatto con chi conosce bene questa nuova tecnologia è stato possibile analizzare i vari fattori che definiscono il concetto di *gaming* in VR e il suo rapporto con il videogioco classico; questi risultati saranno ampiamente discussi nelle conclusioni finali di questa tesi.

# Parte I

## Nuovi mezzi narrativi



# Capitolo 1

## Il videogioco come narrazione

Fin dalle prime arcaiche forme di linguaggio umano, passando per la rivoluzione del testo scritto e giungendo ai mezzi di comunicazione di massa moderni, l'informazione è stata prevalentemente trasmessa in forma di narrazione, ovvero riorganizzando il modo di esprimere la propria esperienza ricavandone un racconto che segua le relazioni logiche di causa ed effetto per essere correttamente interpretato dal fruitore. Questa scelta continua tutt'oggi nella quasi totalità delle produzioni cinematografiche, per poter avere la maggior diffusione possibile presso le masse, e, per lo stesso motivo, nelle più recenti produzioni videoludiche.

Con l'avvento di questo nuovo mezzo di comunicazione si ha un'enorme innovazione: se con il cinema si era riusciti a trasferire una storia in un'immagine dinamica reale, con il videogame si aggiunge il fattore dell'interattività, passando così, da un "medium caldo", che può essere vissuto passivamente perché trasmette autonomamente il messaggio voluto, ad uno "freddo", che necessita di una maggior interazione dell'utente, secondo l'analisi del sociologo Marshall McLuhan (1911-1980), come vediamo in una delle sue opere: *Gli strumenti del comunicare*.

Con il videogioco lo spettatore passivo del cinema diventa ora un fruitore attivo in quanto il game designer pone le basi del racconto dell'opera videoludica, ma sta al giocatore muovere la camera come "regista", personalizzare il proprio avatar come se scegliesse i "costumi di scena" e portare avanti la trama agendo in prima persona come "attore", diventando nel complesso un "co-creatore" del prodotto.

# 1.1 L'evoluzione della tecnologia

## 1.1.1 Gli albori

La prima sperimentale forma di "gioco elettronico" risale al 1947 con il *Cathode-Ray Tube Amusement Device* di Thomas T. Goldsmith Jr. e Estle Ray Mann, che permetteva all'utente di comandare la traiettoria di un missile verso un bersaglio incollato sullo schermo con pellicole trasparenti, dato che non poteva essere rappresentato elettronicamente. Il fatto che il dispositivo usasse solo tecnologia analogica non gli permette di rientrare a pieno titolo nella categoria di "videogame", ma ne è comunque il precursore.

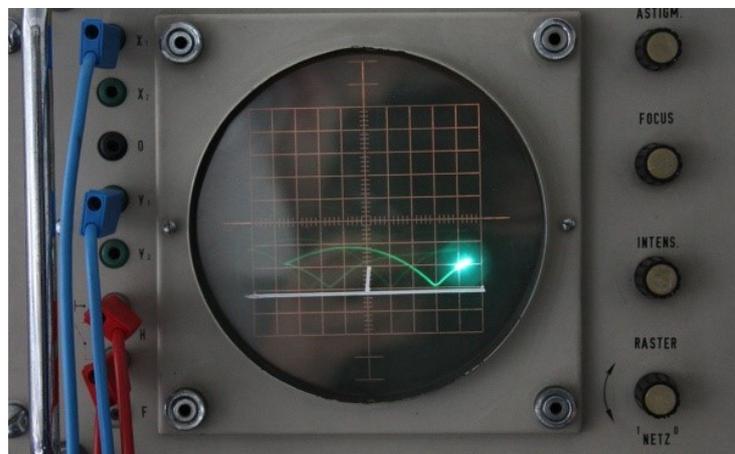


Figura 1.1: Il display del *Cathode Ray Amusement Device*.

Lungo il corso degli anni '50 vari videogame furono sviluppati per scopo di studio delle tecnologie presso vari istituti, ma è solo nel 1962 con *Spacewar!* che vediamo un videogame venire diffuso al di fuori di un singolo istituto.

Dal 1970 questi progetti per uso di studio accademico hanno incominciato a diffondersi al di fuori delle università dando il via al fenomeno degli **Arcade**, ovvero le Sale Giochi (Figura 1.2 (a)), diventate veri e propri luoghi di ritrovo e fenomeni sociali tra gli anni '80 e inizio anni '90.

Al giorno d'oggi questa concezione del videogioco continua a perdurare in Giappone dove i moderni *Club Sega* (Figura 1.2 (b)) e *Taito Station* continuano a proporre nuovi *Coin Op*, abbreviazione di *Coin Operated* (gioco a gettoni) per un pubblico sempre stabile, mentre nel resto del mondo sono state soppiantate dalle piattaforme di gioco casalinghe, ovvero le *Home Console*, e spesso trasformate in Sale Slot, erroneamente accomunate alle precedenti Sale Giochi.

---

Bibliografia [2]



(a) Sala giochi occidentale di fine anni '80.



(b) Un moderno Club Sega giapponese.

Figura 1.2: Due esempi di Arcade.

### 1.1.2 Le Home Console

Ad oggi distinguiamo nove diverse generazioni, o ere, di videogame legate al tipo di piattaforma su cui sono fruibili; ne segue una breve presentazione.

Il veloce evolversi della tecnologia ha portato ad un'altrettanta veloce evoluzione del videogame che, dopo vari esperimenti, vede nel 1972 la nascita della prima *Home Console* di concezione moderna, ovvero una macchina di dimensioni ridotte da poter utilizzare nella propria abitazione e capace di leggere diversi giochi, il *Magnavox Odyssey* ideato da Ralph H. Baer.

Questa prima era delle Home Console raccoglie molte decine di diversi prototipi che non usavano microprocessori, ma circuiti a logica discreta i quali implementavano tutti gli aspetti del gioco stesso, che era quindi direttamente integrato nella console.

Questo tipo di tecnologia continuò a essere usata fino al 1977 e poi abbandonata in favore delle cartucce, ma, di recente, Nintendo ha sfruttato il fattore nostalgia dell'utenza più matura per proporre versioni moderne e compatte di vecchie console da gioco con un nutrito parco di titoli pre-installati, riutilizzando, di fatto, questa arcaica concezione di macchina da gioco.

Parte quindi la seconda era delle console, caratterizzata dall'avvento dei microprocessori economici che permettevano di programmare i giochi su ROM montate a cartucce rimovibili invece di avere i giochi fisicamente presenti nella console stessa, come accadeva nella generazione precedente.

Di questa generazione fa parte la prima celebre console di Atari, ovvero *Atari 2600* che, potendo contare sui diritti per le conversioni dei famosi titoli per sale giochi come *Pac-Man* (sviluppato da Namco) e *Space Invaders* (di Taito), divenne la piattaforma più diffusa del periodo.

---

Bibliografia [2]

## 1.1. L'EVOLUZIONE DELLA TECNOLOGIA

---

Si passa alla terza era nel 1983 con l'avvento del *Famicom*, conosciuto come *NES* in occidente, e del *Sega Master System*. Questa generazione è caratterizzata dall'elevata qualità dei nuovi titoli proposti, non a caso sono saghe che, pur avendo subito pesanti modifiche per adattarsi alle più fluide meccaniche di gioco odierne, continuano ancora oggi, come: *Metal Gear Solid* (di Konami), *Super Mario Bros* (di Nintendo), *Final Fantasy* (di Square Co.) e *Bomberman* (di Hudson Soft).

La quarta generazione è data dalla nascita delle console a 16 bit nel 1987/88 con il *Sega Mega Drive*, o *Genesis*, come conosciuto in occidente, e con lo *SNES* (successore del *NES*) con cui si sono visti i primi esperimenti di giochi 3D, tecnologia che oggi è alla base della grande maggioranza dei videogiochi moderni, spesso relegando i titoli 2D al concetto di "operazione nostalgia".

La rivoluzione della grafica 3D segna anche la nascita delle produzioni che puntano al realismo, ma complice la mancanza di tecnologie adeguate al tempo, non tutti i titoli 3D hanno riscosso immediato successo.

Si passa alla quinta era nel 1993 con l'arrivo del *3DO*, la prima console a 32 bit con supporto ottico. Questo nuovo tipo di hardware ha permesso uno sviluppo massiccio della grafica 3D come vediamo sulla prima *Playstation* di Sony, su *Sega Saturn* e *Nintendo 64*.

Oltre ad una mera innovazione estetica, il passaggio al 3D ha portato grandi novità alle meccaniche di gioco per generi che stavano ormai iniziando a sentire il peso degli anni; su tutti, il genere *Platform* (o gioco di piattaforme) ne uscì revitalizzato, tanto che ancora oggi *Super Mario 64*, gioco per *Nintendo 64*, è riconosciuto come uno dei migliori esponenti del genere.

L'introduzione della grafica 3D ha permesso anche la nascita di nuovi generi di gioco che prima non potevano essere rappresentati per limitazioni tecniche, come ad esempio gli fps<sup>1</sup>, che è oggi il genere più diffuso in occidente.

La sesta generazione non porta modifiche strutturali, ma l'avanzamento delle tecnologie consente di raggiungere dettagli grafici mai visti prima, inoltre si inizia a delineare il panorama di mercato che perdura ancora oggi, ovvero l'oligopolio di Sony e Nintendo, e successivamente Microsoft, a cui dobbiamo le successive introduzioni dei servizi on-line per le Home Console (come multiplayer online, DLC<sup>2</sup>, microtransazioni, shop on-line) e, in particolare grazie a Nintendo, dei controlli di movimento con il *Wiimote*, un particolare pad per il *Nintendo Wii* che riceve gli input del giocatore tramite una combinazione di giroscopi che rilevano il movimento e le rotazioni della periferica.

---

<sup>1</sup>Fotogrammi per secondo

<sup>2</sup>Contenuto scaricabile (*downloadable content*)

## 1.1. L'EVOLUZIONE DELLA TECNOLOGIA

---

Le successive generazioni sono state caratterizzate da una rincorsa al progresso tecnologico per cercare di avvicinare le home console alle prestazioni dei più potenti PC, che da sempre settano picchi qualitativi per definizione di immagine e framerate.

Al contrario delle dirette concorrenti di mercato Nintendo ancora oggi preferisce la via dell'originalità a discapito del crudo avanzamento tecnologico.

Nel 2017 siamo alle battute finali dell'ottava era, periodo che vede il mercato console conteso tra *PS4* e *XBox One*, lanciate nel 2013 in occidente e l'anno successivo in Giappone, e la recente *Nintendo Switch* disponibile da aprile 2017.

Questa era ha portato grandi innovazioni tecnologiche tra cui il supporto allo standard di risoluzione 4K con *PlayStation 4 Pro* e *XBox One X* per poter soddisfare la richiesta del pubblico sempre più attento al lato tecnico della produzione rispetto al passato e l'introduzione di supporti per la realtà virtuale con *Playstation VR*, che cerca di rendere disponibile questo tipo di esperienze alla massa, ancora scettica sulle effettive potenzialità di questa tecnologia "relativamente" nuova.



Figura 1.3: Le tre principali console odierne: da sinistra *Nintendo Switch*, *XBox One* e *PlayStation 4* rispettivamente commercializzate da Nintendo, Microsoft e Sony.

## 1.1. L'EVOLUZIONE DELLA TECNOLOGIA

---

Parallelamente al mondo home console si sono sviluppati anche il *PC gaming*, che ad oggi è il punto di riferimento per il lato tecnico del videogame, raggiungendo vette di qualità grafica impensabili per altre piattaforme, e il *mobile gaming*, ovvero il gaming su cellulari e tablet che, con l'introduzione del modello "Free to Play", ovvero proponendo gratuitamente un gioco in cui si possono acquistare facoltativamente oggetti *in app*, ha letteralmente dominato il mercato Giapponese negli ultimi anni, tanto che questo tipo di meccaniche di gioco è stato poi introdotto anche nei più classici videogame da console. La contaminazione non ha solo intaccato il modello commerciale proposto nel mondo dei videogame, ma anche le strutture di molti giochi, portando le esperienze ad avvicinarsi ai prodotti fruibili anche con sessioni più rapide, tipiche del gaming mobile.

Il più recente prodotto dato da questa commistione di stili è la recente piattaforma di Nintendo, *Nintendo Switch* che risulta utilizzabile sia in mobilità come un tablet che come una classica console fissa casalinga.

Anche, e soprattutto, in queste due branche di mercato si è fatta strada la realtà virtuale con vari dispositivi che variano tra loro per qualità della resa visiva, prezzo e sensazione di immersione. Tra i maggiori esponenti ricordiamo *HTC Vive* e *Oculus Rift*, dispositivi ad alta qualità per PC entrati in commercio nel 2016, e vari *head set* per cellulari quali il famoso *Gear VR*.

Proprio l'introduzione e la seguente evoluzione di questo nuovo paradigma nel modo di fruire le esperienze videoludiche ha permesso lo sviluppo di svariate nuove *software house* che puntano unicamente ad offrire esperienze esclusive per VR, tra cui ovviamente KR Games, con i suoi giochi/esperienze per *Gear VR* e *Oculus Rift*, che analizzeremo in seguito.



(a) Magnavox Odyssey (1972).



(b) Gear VR (2016).

Figura 1.4: Confronto visivo tra la prima Home console di concezione moderna, il *Magnavox Odyssey* (a) e uno dei più diffusi headset per realtà virtuale, *Gear VR* (b).

## Capitolo 2

# Un nuovo modo di raccontare

Dopo questa rapidissima panoramica sull'evoluzione tecnica di questo medium dobbiamo discutere di alcuni punti chiave inerenti al contenuto di quest'ultimo piuttosto che al come esso viene trasmesso: i videogiochi moderni possono avere un reale un messaggio o le loro trame sono solo pretesti per giocare?

Sono una forma d'arte o semplici passatempo?

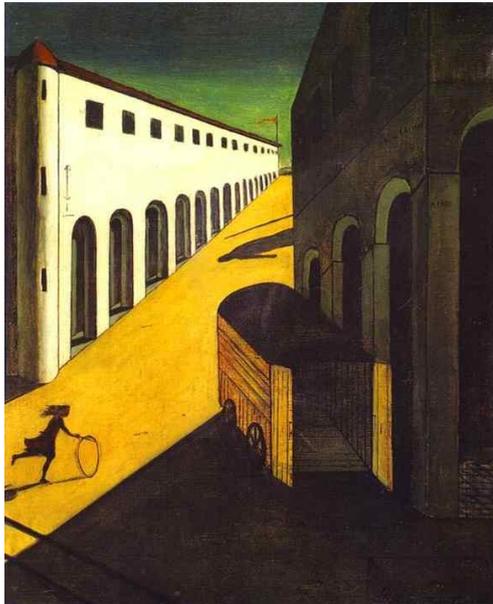
È giusto definirli semplici film interattivi o il concetto stesso di essere avventure plasmabili dall'utente nasconde qualcosa in più?

Difficile se non impossibile dare una risposta esaustiva a queste domande, sia per il fatto che giochi diversi pensati per un pubblico differente presentano una narrazione e meccaniche di *gameplay* non paragonabili tra loro; sia perché il concetto di arte in un gioco può essere molteplice: lo si può individuare nel lato estetico, nel sonoro, nella scrittura dei dialoghi o nel *level design* per fare qualche esempio, ma quando un videogioco può ottenere lo status di opera artistica, elevandosi dal semplice concetto di giocattolo elettronico?

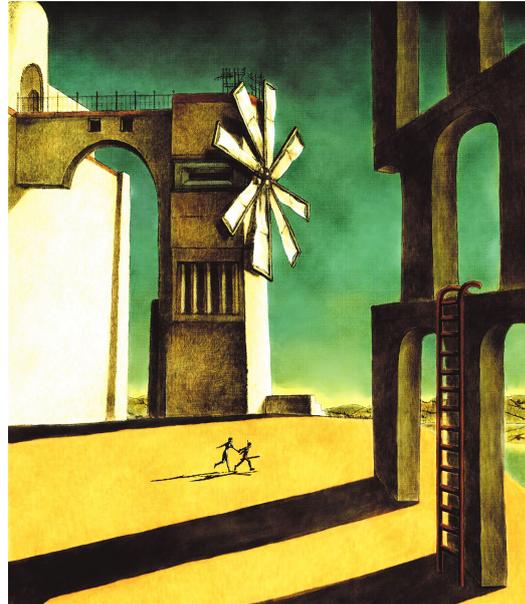
Essendo molto più giovane degli altri media, solo ora il videogioco sta iniziando ad attraversare quella fase di accettazione dell'opinione pubblica che, ad esempio, il cinema ha superato decine di anni fa, per essere visto non solo come un giocattolo, ma come un mezzo artistico a tutti gli effetti. L'industria del gaming si è evoluta al pari con le tecnologie che da sole non bastano più a creare un buon titolo, ma si rende necessario uno studio curato per *character design*, una sceneggiatura scritta con attenzione ai minimi particolari, *sound design* per avere una buona immersione, pianificazioni per level e game design, tanto che le maggiori produzioni richiedono team formati da centinaia di persone altamente qualificate in ambiti molto diversi tra loro.

## 2.1. ESEMPIO DI NARRAZIONE COMPLESSA NEL VIDEOGIOCO: *METAL GEAR SOLID 3*

---



(a) Mistero e malinconia di una strada (1914), Giorgio de Chirico.



(b) Cover art di ICO.

Figura 2.1: Confronto tra un quadro di Giorgio de Chirico e la cover art di un videogioco.

## 2.1 Esempio di narrazione complessa nel videogioco: *Metal Gear Solid 3*

Per capire quanto possa essere oneroso il lavoro che sta dietro allo sviluppo narrativo di un singolo gioco prendiamo in esame il caso della saga fantapolitica di Konami, *Metal Gear Solid* creata da Hideo Kojima e, nello specifico, parliamo del suo terzo capitolo uscito su PS2 nel 2004, *Metal Gear Solid 3: Snake Eater*.

Il gioco sembra essere un prequel dei due precedenti capitoli, ambientati negli anni 2000; ci troviamo infatti nel 1964, durante la Guerra Fredda, e l'unità FOX di cui fa parte il protagonista dal nome in codice "Naked Snake", deve recuperare lo scienziato Nikolai Stepanovich Sokolov dai territori dell'Unione Sovietica.

Durante questa missione il giocatore controlla Naked Snake e ha una libertà d'azione incredibile per il tempo: può affrontare le guardie sparando ad armi spianate come nei migliori film d'azione; può pianificare con cura la scelta del suo equipaggiamento, da silenziatori a tute mimetiche per passare inosservato; può improvvisare ad esempio fingendosi morto in mezzo ad altri cadaveri per poi alzarsi all'improvviso sorprendendo i nemici o, se catturato, può letteralmente far girare su se stesso il proprio avatar fino a farlo vomitare fingendo un

## 2.1. ESEMPIO DI NARRAZIONE COMPLESSA NEL VIDEOGIOCO: *METAL GEAR SOLID 3*

---

malore, così, i carcerieri, vedendoci ammalati apriranno la cella per controllare rendendo possibile la nostra fuga, oppure ancora possiamo rifiutarci più volte di mangiare le razioni della prigione, il che porta ad un dialogo segreto con i nostri aguzzini che ci fa capire come anche i nostri crudeli nemici detestino il loro lavoro.

Queste interazioni avvengono in modo naturale ed amalgamato con la narrazione: non compare a schermo un indicatore "a tasto x corrisponde azione y", tutto segue una logica di azione e reazione estremamente realistica nonché avveniristica per un videogioco dell'epoca, cercando di eliminare dallo schermo tutte le informazioni di HUD non necessarie, ovvero cercando di rendere trasparente il più possibile la barriera tra gioco e giocatore. Uno dei migliori esempi di questa scelta è la scena finale del gioco, quando, sconfitta la ex mentore di Naked Snake, lei rivela di non aver mai realmente tradito il suo paese e di aver fatto il doppio gioco con il nemico sapendo che sarebbe poi stata uccisa dai suoi precedenti alleati, per permetterne la vittoria; sta al giocatore premere il grilletto per l'ultima volta senza che questo venga esplicito da alcun input extra diegetico: il giocatore durante il corso del gioco ha appreso che il tasto R1 è adibito allo sparo e non servono altre spiegazioni che vadano a inficiare il momento particolarmente emozionante della trama (Figura 2.2 (b)).

Al contrario esistono molti esempi negativi di situazioni simili in cui l'immersione viene totalmente spezzata per spiegare cosa deve fare il giocatore: un caso emblematico è la famosa scritta a schermo "Press F to pay respects" (Figura 2.2 (a)) del videogioco *Call of Duty: Advanced Warfare* (di Sledgehammer Games) che cerca di ricondurre alla pressione di un pulsante una reazione emotiva del personaggio controllato, ignorando ciò che possa effettivamente provare il giocatore.

2.1. ESEMPIO DI NARRAZIONE COMPLESSA NEL VIDEOGIOCO:  
*METAL GEAR SOLID 3*

---



(a) *Fotogramma dal videogioco Call of Duty: Advanced Warfare, Sledgehammer Games.*



(b) *Fotogramma dal videogioco Metal Gear Solid 3: Snake Eater, Konami.*

Figura 2.2: Confronto tra l'utilizzo delle "call to action" nei videogiochi: nella Figura (a) ne vediamo l'esempio scorretto di *Call of Duty: Advanced Warfare*, nella (b) vi è il suo corretto utilizzo con *Metal Gear Solid 3: Snake Eater*.

## 2.1. ESEMPIO DI NARRAZIONE COMPLESSA NEL VIDEOGIOCO: *METAL GEAR SOLID 3*

---

Tornando al titolo di Konami, lo stesso livello di ricercatezza lo si trova nei dialoghi, ottimamente scritti e facenti spesso riferimento ad importanti eventi storici, raccontando dettagli reali sapientemente miscelati ad avvenimenti frutto della fantasia dell'autore.

Tutto questo per sottolineare quanta attenzione è stata riservata ai minimi dettagli dell'opera e per far notare quanto fuori posto sembrano alcune piccolezze del gioco: nel briefing della "Missione Virtuosa", Naked Snake mal capisce il nome in codice, strano dettaglio per un personaggio che appare sempre il perfetto soldato vigile, attento e scrupoloso.

MAJOR ZERO : Jack, I've got some important news. The Head of the CIA has finally given us the green light for the "Virtuous Mission".

NAKED SNAKE : Virtual mission?

MAJOR ZERO : No, the "Virtuous Mission".

In seguito incontreremo un personaggio che riveste il ruolo di uno dei maggiori antagonisti di Snake nei due titoli precedenti della serie, di cui questo videogioco è prequel, ovvero Revolver Ocelot, ed il gioco, sempre fedele alla sua linea di lasciare libertà al giocatore, ci permette di ucciderlo istantaneamente, ma qualora lo si faccia si arriva ad una schermata di Game Over.

Viene spontanea la domanda: perché uccidere un nemico dovrebbe portare alla sconfitta?

Attendendo qualche minuto alla schermata di Game Over senza dare input ai comandi si ottiene il messaggio scritto "Time Paradox" e si sente una voce che urla: "What have you done? You changed the future!" (Figura 2.3). Questa breve frase rivela un'informazione preziosa, ovvero che, nel mondo di gioco, qualcuno già conosce il futuro della storia e il ruolo che Ocelot avrà in questa.

Il tutto è ancor più strano poiché la voce che sentiamo è quella del Colonnello Campbell, personaggio che non compare in Metal Gear Solid 3, pur essendo il nome del suo doppiatore inserito tra i titoli di coda, ma appare nei giochi successivi nella timeline della saga. La presenza di questo personaggio nel 1964 è possibile solo se in realtà non ci si trovi nel 1964, ma anni nel futuro e la missione sia, come suggerisce Naked Snake nel suo primo dialogo di gioco, una missione virtuale.

## 2.1. ESEMPIO DI NARRAZIONE COMPLESSA NEL VIDEOGIOCO: *METAL GEAR SOLID 3*

---



Figura 2.3: La curiosa schermata di *Game Over* che si ottiene eliminando anzitempo Revolver Ocelot.

Come rafforzativo di questa tesi, nel gioco sono reperibili alcuni oggetti e vengono citati eventi che non potevano esistere nell'anno in cui si dovrebbe svolgere la storia di *Metal Gear Solid 3*: troviamo come oggetto di cura una sorta di barretta energetica realmente esistente chiamata *Calorie Mate* che viene prodotta dal 1983, sentiamo riferimenti al *JFK Special Operations Center* (una struttura militare) che viene costruita in onore di John F. Kennedy solo nel 1982 e infine sentiamo un dialogo via radio tra Para-Medic, altra alleata di Naked Snake, e Major Zero, grande fan di James Bond, in cui parlano con grande ardore dei suoi film e dei suoi famosi gadget da agente segreto.

MAJOR ZERO : Snake, wouldn't you like to have a gun shaped like a pen?

PARA-MEDIC : Snake, have you seen *From Russia with love*?

Il film è del 1963, quindi compatibile con la timeline, ma la penna-pistola appare solo nel 1983 nel film *Never say never again*.

Questi anacronismi potrebbero essere semplici errori di produzione, ma in un'opera che punta molto sulla grande coerenza della trama fantapolitica e che fa della mole di dettagli usati per conferire credibilità al mondo di gioco un punto di forza, queste leggerezze stonano. Si può notare come questi elementi anacronistici vengano tutti dal futuro, più precisamente dal 1982, 1983 e probabilmente non è un caso.

In *Metal Gear Solid V: The Phantom Pain*, Naked Snake, dopo aver ucciso The Boss, ne eredita il nome in codice diventando "Big Boss", continuando ad accettare missioni pericolose sui campi di battaglia come mercenario finché viene ferito gravemente in un' esplosione e resta in coma per nove anni.

Nello stesso tempo Major Zero scopre un altro sopravvissuto coinvolto anch'esso nella stessa esplosione e ne fa alterare l'aspetto tramite chirurgia per farlo somigliare a Big Boss e gli dà il nome in codice "Venom Snake"; al suo risveglio non solo somiglia al Big Boss originale, ma ne ha memoria e capacità.

## 2.1. ESEMPIO DI NARRAZIONE COMPLESSA NEL VIDEOGIOCO: *METAL GEAR SOLID 3*

---

In *MGS V* non viene chiaramente spiegato come ciò sia possibile, ma se assumiamo che gli eventi di *MGS 3* siano in realtà una missione virtuale vissuta da Venom Snake per allenarlo e renderlo il nuovo Big Boss, tutti i paradossi temporali trovano un senso, poiché l'anno in cui Venom Snake diventa un agente operativo è il 1984. Le due storie dei titoli presi in esame si completano a vicenda perfettamente pur non dicendo mai espressamente al giocatore se questa sia effettivamente la giusta interpretazione; non solo, questa particolare interpretazione pur essendo di grande importanza per la comprensione del mondo di questa saga, è nascosta per tutti gli utenti che subiscono passivamente il videogioco senza esplorarlo in tutte le sue possibilità, ovvero per quegli utenti che vivono un videogioco come un film o come un altro medium non interattivo.

Questa è una delle grandi armi a disposizione del videogame: lasciare all'utente la possibilità di scoprire o meno i risvolti importanti della storia a seconda di come questo vada ad interagire con l'ambiente di gioco; per un'esperienza diversa da quella cinematografica va sfruttata al massimo la peculiare interattività che offre questo mezzo, non basta più intervallare sezioni filmate a sezioni di gioco, ma, per un prodotto di alto livello, serve un avanzamento omogeneo e naturale di narrazione e gameplay.

Ovviamente la bontà di queste osservazioni è fortemente legata alla tipologia di gioco che si vuole sviluppare.

Per capire meglio l'importanza che ha l'interattività nella narrazione di un videogioco, segue un piccolo stralcio di un'intervista di *Wired* ad Hideo Kojima, creatore della saga di *MGS*:

**Wired:** Cosa ti attirava dei videogame?

**Hideo Kojima:** Da ragazzo volevo lavorare nel cinema e creare i miei film, e intanto avevo cominciato a scrivere alcuni romanzi. [...] Proprio in quel periodo scoprii il Nes e un nuovo approccio all'interattività [...] Era una cosa molto diversa dal cinema, ma pensavo che questa forma di interattività forse mi avrebbe permesso di esprimere me stesso e comunicare qualcosa alle altre persone.

**Wired:** Cosa cambia in *Metal Gear Solid V*, in termini di narrazione?

**Hideo Kojima:** [...] puoi giocarlo più volte e trovare sempre qualcosa che non avevi visto. [...] scopri mano a mano varie parti della storia, le ricolleggi insieme e ottieni il quadro generale, la 'grande storia' che sta dietro a tutto.

## 2.1. ESEMPIO DI NARRAZIONE COMPLESSA NEL VIDEOGIOCO: *METAL GEAR SOLID 3*

---



Figura 2.4: Hideo Kojima mentre ritira il premio alla carriera al *Brazil Game Show 2017*.

Ad oggi Hideo Kojima ha lasciato Konami, la casa di produzione a cui è stato lungamente legato e che ha contribuito a renderlo un nome di spicco dell'industria, per lavorare ad una nuova proprietà intellettuale, *Death Stranding* di cui si sono visti pochi enigmatici trailer ricchi di simbolismi, ma senza il minimo accenno di gameplay, altro indizio che, per questo autore, il messaggio che vuole portare va oltre il gioco stesso. A rafforzare l'idea che si stia cercando di creare un'opera che possa far parlare di se anche al di fuori del mondo del gaming, il progetto vede la collaborazione del famoso regista Guillermo Del Toro e dell'attore Norman Reedus.



Figura 2.5: Un'immagine dal trailer di *Death Stranding*, nuovo progetto di Hideo Kojima, in cui vediamo il personaggio interpretato dall'attore Norman Reedus.

## 2.2 Esempio di arte nell'estetica del videogioco: *Okami*

Andiamo ora a presentare un esempio della complessa ricerca stilistica ed artistica che è a monte del processo di creazione di alcuni videogiochi.

Dal vocabolario:

**Arte:** attività umana volta a creare opere a cui si riconosce un valore estetico, per mezzo di forme, colori, parole o suoni.

Esistono particolari titoli in cui la ricerca stilistica per grafica e sonoro ha occupato la maggior parte del tempo di produzione generando non solo un grande impatto visivo, ma assimilando queste peculiari scelte estetiche all'interno del gameplay.

Vediamo rapidamente l'esempio di *Okami, adventure game* (Figura 2.6) uscito nel 2006 su *Play Station 2* e sviluppato da Clover Studio; il gioco racconta di come la Dea Scintoista del sole Amaterasu scese nel mondo degli uomini sotto forma di lupo bianco, per salvarli dal male.

Tutto il gioco è rappresentato in *cel-shading*, ovvero mediante una tecnica di rendering non fotorealistica che va ad imitare il disegno a mano aggiungendo, ad esempio, linee nere lungo i bordi degli oggetti ed ombreggiature che simulano l'effetto di una matita da disegno.

La variante di cel-shading usata per il titolo in questione risulta particolarmente interessante poiché imita lo stile pittorico *Sumi-e*, una tecnica pittorica nata in Cina durante la dinastia Tang (618-907) e diffusasi in Giappone, dove è ambientata la storia del gioco, nel XIV secolo; questa tecnica prevede l'utilizzo del solo inchiostro nero in varie concentrazioni e sfrutta forme per lo più astratte. Questo stile visivo si fonde completamente alle meccaniche di gioco, infatti, il giocatore può a suo piacimento trasformare il mondo virtuale in un dipinto *Sumi-e* ed aggiungere oggetti dipingendoli direttamente nello scenario con il "pennello celestiale" che funge da "arma" per il protagonista.

In questo caso dunque lo stile visivo è stato scelto non per la mera ricerca di un'estetica ispirata, ma per renderlo una componente portante dell'opera sfruttando la peculiare interattività che solo i videogiochi possono offrire, creando una sorta di quadro interattivo. Il giocatore in questo caso più che in molti altri non è solo spettatore, ma un vero cocreatore dell'opera.

### 2.3. ESEMPIO DI METANARRAZIONE NEL VIDEOGIOCO: *UNDERTALE*

---



Figura 2.6: Schermata di gioco di *Okami*.

### 2.3 Esempio di metanarrazione nel videogioco: *Undertale*

Negli ultimi anni si è affermato il fenomeno del videogioco indipendente, o *Indie*, cioè un videogame sviluppato da un piccolo team o addirittura da una sola persona, pubblicato senza l'ausilio economico di un editore.

In particolare dagli anni 2000, grazie alle nuove piattaforme di distribuzione digitale come *PSN*, *Steam* e gli store per dispositivi mobile, e grazie a nuovi strumenti di sviluppo più user friendly come *Adobe Flash* e *Game Maker*, alcuni di questi videogiochi sono diventati veri successi commerciali, complice il fatto di potersi prendere grandi rischi a livello creativo, cosa non possibile per produzioni da milioni di dollari che necessitano di minimizzare i rischi per massimizzare i profitti per poter sostenere gli enormi investimenti fatti.

### 2.3. ESEMPIO DI METANARRAZIONE NEL VIDEOGIOCO: *UNDERTALE*

---

Proprio queste produzioni ad alto budget, nonostante a livello puramente tecnico siano spesso lo stato dell'arte, hanno iniziato ad uniformarsi tra di loro, proponendo sempre più di frequente meccaniche di gioco ben collaudate, tematiche scomode o non adatte al grande pubblico vengono presentate di rado, così come di rado si vedono metodi di narrazione che non seguano i classici punti di svolgimento del testo narrativo:

- 1) Situazione iniziale.
- 2) Esordio, un avvenimento che modifica la situazione iniziale.
- 3) Le peripezie, i vari avvenimenti che coinvolgono i personaggi.
- 4) La *Spannung*, il momento di massima tensione.
- 5) Lo scioglimento o conclusione in cui si ricompone l'equilibrio.

Le produzioni Indie hanno quindi iniziato a sperimentare per poter superare i "big" dell'industria spingendo sull'innovazione e sulle particolarità di meccaniche e narrazioni offerte. Tra i molti esempi che si possono fare spicca, soprattutto per la qualità del messaggio che trasmette e del modo con cui l'opera "parla" al giocatore, *Undertale*, gioco per PC sviluppato interamente da Toby Fox e Temmie Chang, e pubblicato il 15 settembre 2015.

Il gioco si presenta come un classico *RPG*<sup>1</sup> dalla grafica 8 bit in cui il protagonista deve fuggire dal mondo dei mostri combattendoli mediante una particolare meccanica *bullet hell* (letteralmente, inferno di proiettili); ma la vera trama del gioco la si può esplorare solo qualora si cerchi di oltrepassare i mostri senza ucciderli, sfruttando un sistema di dialogo a scelte multiple (Figura 2.7), scoprendo che queste creature non sono poi così mostruose, infatti ogni personaggio presenta una bizzarra *back story*, spesso comica, ma dai risvolti talvolta amari: lo scheletro spadaccino vuole solo fare bella figura con le guardie reali del palazzo, il robot killer cerca di ucciderci solo per via della sua programmazione, ma in realtà vorrebbe fare lo showman, e così via.

Nelle fasi più avanzate, scopriamo che la EXP guadagnata sconfiggendo i nemici e i LV accumulati (solitamente sigle classiche del mondo dei videogames che stanno per "esperienza accumulata" e "livelli") in questo titolo significano *Execution Point* e *Level of Violence* (punti uccisione e livello di violenza) trasformando il giocatore che ne ha guadagnati molti, abituato alle solite meccaniche di gioco, nel vero mostro della storia.

Uccidendo anche solo un mostro gli altri personaggi reagiscono all'avvenimento cambiando le loro linee di dialogo e diventando più ostili.

Anche ricaricando la partita dopo aver ucciso un qualsiasi personaggio e, questa volta, risparmiandolo si ottiene un effetto simile.

---

<sup>1</sup>Role Play Game (gioco di ruolo)

FLOWY : [...] In this world, it's kill or be killed. You spared the life of a single person, but don't act so cocky. I Know what you did. You murdered her. And then you went back because you regretted it [...] The ability to play God. The ability to "SAVE" [...] Enjoy that power while you can. I'll be watching.



Figura 2.7: Schermata di combattimento di *Undertale*; invece di attaccare i mostri si può raccontare loro una barzelletta per diventare amici.

Il gioco rompe la quarta parete e parla direttamente al giocatore che ha compiuto la data azione, piuttosto che al suo avatar, e questo serve ad aumentare la sensazione di immersione nell'opera in modo impensabile per altri media, poiché il gioco reagisce ad una scelta fatta anche quando questa, secondo la logica dello "strumento videogioco", uscendo senza salvare e ricaricando la partita, non dovrebbe essere mai accaduta.

Inoltre, sul piano di lettura della narrazione, ci dice che Flowy si rende conto delle nostre azioni reali, come uscire dal gioco senza salvare per modificare una nostra azione e che comprende questo potere, che diventa quindi parte integrante della storia raccontata.

Il concetto astratto di "potere di scegliere" di una persona reale diventa qui una componente concreta della trama dell'opera sfruttando le diverse chiavi di lettura che le si può dare.

## 2.4 Il gaming di massa

Seppur il lato del gaming analizzato nel paragrafo precedente sia estremamente affascinante, i giochi creati con l'effettivo intento di portare un'opera artistica al pubblico sono relativamente pochi e spesso provenienti dal solo mercato giapponese, come dimostrano ad esempio le opere di Fumito Ueda (Figura 2.8) quali *Ico* e *The last Guardian*, che fanno delle atmosfere sognanti e dei lunghi silenzi contemplativi il loro cavallo di battaglia, o la saga di *Persona* (Figura 2.9) di Atlus, che in ogni iterazione basa la propria narrazione su

## 2.4. IL GAMING DI MASSA

---

questioni filosofiche importanti come "Il concetto di libertà nella società moderna", "ciò che vediamo con i nostri occhi rappresenta la verità per tutti?" o "la ricerca del senso della nostra vita".



Figura 2.8: Lo sconfinato e silenzioso mondo di *Shadow of the Colossus*, altra opera di Fumito Ueda.

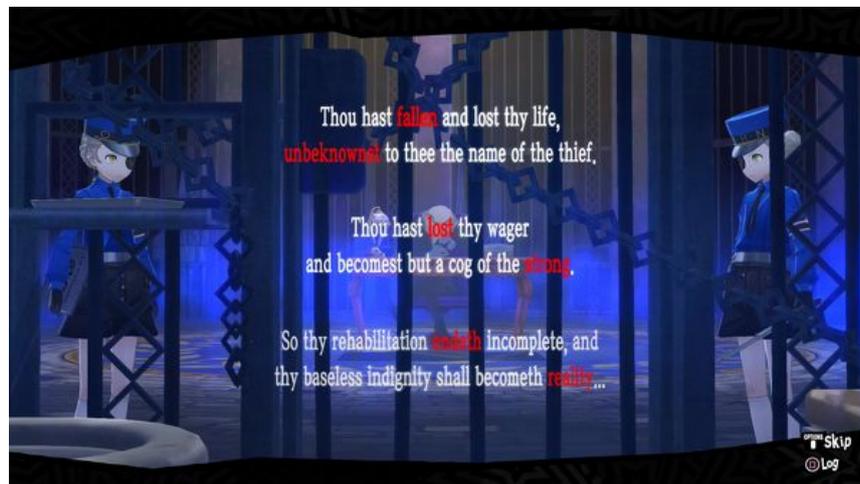


Figura 2.9: Poetica schermata di Game Over del videogioco *Persona 5*.

Soprattutto in occidente, invece, si tende a creare giochi seguendo un più classico modello commerciale che prevede un'uniformità generale nelle meccaniche e nelle trame dei giochi per poter recuperare con più sicurezza il capitale investito. Questo modello commerciale occidentale prevede il focalizzare il gioco sulla componente multiplayer competitiva che permette ai giocatori di sfidarsi tra di loro on line, spesso tralasciando completamente l'implementazione di una trama single player; questo particolare trend ha avvicinato molto il mondo dei videogame al gioco competitivo tanto che è notizia del 28 ottobre 2017 che il CIO, Comitato Internazionale Olimpico, si è espresso favorevolmente all'introduzione dei cosiddetti *eSport*, ovvero i videogiochi competitivi, tra le discipline olimpiche.

«Gli eSports competitivi - è scritto in un comunicato del CIO - possono essere considerati un'attività sportiva, e i giocatori coinvolti si preparano e allenano con un'intensità che può essere paragonata a quella degli atleti delle discipline tradizionali. Gli eSports sono in forte crescita, in particolare fra i giovani dei vari paesi, e ciò può essere la piattaforma per un coinvolgimento nel movimento olimpico»

La notizia non solo ha causato il prevedibile malcontento del mondo degli sportivi che vedono il loro mondo invaso da un elemento estraneo, ma anche una buona parte di videogiocatori si trova contraria a queste dichiarazioni, non volendo vedere perdersi ulteriormente l'idea di videogame come opera artistica da vivere in solitaria contemplazione, come fosse un quadro, in favore di più moderne, ma talvolta sterili, meccaniche di gioco.

Questa omogeneizzazione delle esperienze non è solo legata alle meccaniche di gioco, ma anche all'impostazione generale del prodotto, tant'è che, presentando più scatti di diversi giochi occidentali di grande successo, è molto difficile, per chi non conosce in prima persona questi titoli, capire che si tratta effettivamente di prodotti differenti (Figura 2.10).

Vediamo un'interfaccia utente molto simile in tutte e quattro le proposte, l'impostazione della visuale pressoché identica e la stessa scelta di puntare al realismo visivo di ambientazioni ed oggettistica. Il giocatore non avanza spinto da una motivazione di particolare, ma solo dal puro piacere di giocare, rendendo il prodotto più simile ad un giocattolo che non ad una esperienza narrativa.

## 2.4. IL GAMING DI MASSA

---



(a) *Fotogramma dal videogioco Call of Duty.*



(b) *Fotogramma dal videogioco Player Unknown Battlegrounds.*



(c) *Fotogramma dal videogioco Battlefield One.*



(d) *Fotogramma dal videogioco Far Cry 4.*

Figura 2.10: Immagini di quattro famosissimi videogiochi diversi, del genere fps sviluppati da software house occidentali differenti dimostrano la grande somiglianza tra le proposte di gioco.

Al contrario questo accade molto meno nelle più artistiche e creative esperienze proposte dal mercato orientale.

Propongo come contro prova quattro immagini di giochi sviluppati da software house giapponesi che, anche se appartenenti allo stesso genere, difficilmente possono sembrare provenienti dalla stessa opera (Figura 2.11).

Sia per scelte estetiche che per impostazione della UI, i quattro titoli presi in esame sono, come possiamo vedere, molto diversi tra loro, pur facendo parte dello stesso genere di esperienze, ovvero RPG, questo perché spesso, le produzioni orientali hanno come fulcro un messaggio da trasmettere, una storia da raccontare attraverso delle meccaniche studiate appositamente ed usate come strumento della narrazione.

Al contrario nei prodotti occidentali le storie e gli spunti narrativi diventano pretesti per godere delle meccaniche di gameplay create.

## 2.5. SEMPLIFICARE LE ESPERIENZE



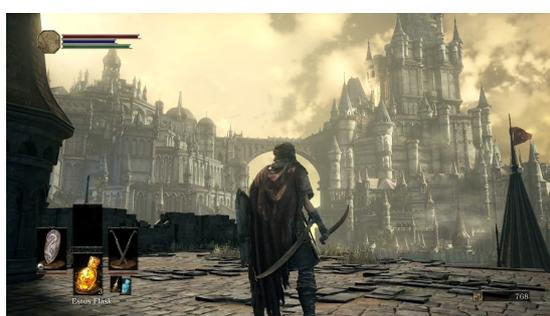
(a) Fotogramma dal videogioco Dragon Quest XI



(b) Fotogramma dal videogioco Final Fantasy XV.



(c) Fotogramma dal videogioco Persona 5 .



(d) Fotogramma dal videogioco Dark Souls 3.

Figura 2.11: Immagini di quattro famosissimi videogiochi diversi, del genere RPG, sviluppati da software house orientali differenti dimostrano la grande originalità tra le proposte di gioco.

## 2.5 Semplificare le esperienze

Le prime esperienze videoludiche di spessore in VR sono sicuramente di stampo prettamente occidentale, anche perché la visuale in prima persona di questo tipo di prodotti, ben si presta alle produzioni fortemente cinematografiche e "su binari" tipiche di questo mercato. Come vedremo nell'analisi dell'applicazione creata per questa tesi, il grande senso di immersione del giocatore va a colmare quelle lacune al gameplay lasciate da meccaniche di gioco spesso molto semplificate o, semplicemente, utilizzate più volte da altri prodotti visti in precedenza. Per ora è proprio la voglia di scoprire la tecnologia di per sé, piuttosto che vivere l'avventura proposta dall'applicazione scelta, a spingere nuova utenza verso la VR; questo permette anche a software house con minore esperienza nella creazione di universi narrativi complessi, di trovare un loro spazio nel mercato attuale. Solo quando ci sarà un pubblico fidelizzato si potrà iniziare a vedere qualche produzione che sia al contempo di alto livello tecnico e che proponga un gameplay più profondo ed appagante.

## Capitolo 3

### La realtà virtuale

Concentriamoci ora sul tipo tecnologia che è alla base della fruizione del prodotto sviluppato durante il periodo lavorativo presso KR Games, ovvero la realtà virtuale.

Il termine *realtà virtuale* viene usato per la prima volta nel 1982 dallo scrittore di fantascienza Damien Broderick, nel libro *The Judas Mandala*, ed indica una simulazione della realtà che coinvolga i sensi dell'uomo mediante dispositivi tecnologici. Attualmente questa definizione può essere usata per indicare non solo a simulazioni virtuali complete, ma anche videogiochi o semplicemente il World Wide Web.

Nei capitoli successivi indicherò con "realtà virtuale" solo la *VR immersiva*, ovvero quella che crea un ambiente esplorabile attorno all'utente, pur ricordando che sarebbe impossibile ricreare completamente la realtà con le sue infinite variabili con le attuali risorse tecnologiche.

La realtà virtuale si basa su tre concetti chiave: **immersione**, **interazione** e **presenza**.

Il primo indica quanto l'utente possa percepire come credibile la simulazione in cui si trova inserito, indica quella che è conosciuta come "sospensione all'incredulità"; il secondo indica la varietà di interazioni possibili nella simulazione, ovviamente una simulazione più complessa prevede molte interazioni, mentre ci possono essere simulazioni dedicate ad ambienti e fenomeni circoscritti con meno interazioni previste; il terzo è definibile come "sensazione di appartenenza" che viene data a sua volta da tre fattori distinti: *qualità delle informazioni sensoriali*, ovvero la qualità di modellazione e di rendering delle componenti che formano mondo virtuale, siano esse visive, sonore o legate agli altri sensi; *mobilità e controllo dei sensori*, cioè quanto i sensori usati dall'utente risultano naturali e comodi all'utilizzo, avendo come obiettivo ultimo quello di offrire un'interazione col mondo virtuale senza dare la sensazione di interagire con una macchina e, infine, il *controllo sull'ambiente*, ovvero quanto l'utente può effettivamente interagire su ciò che lo circonda nel mondo virtuale con un'effettiva relazione di azione e reazione.

## 3.1 L'evoluzione della VR

I primi esperimenti di VR risalgono al 1962 quando Morton Heiling diede vita alla sua visione di un cinema che potesse coinvolgere altri sensi oltre alla vista, realizzando il *Sensorama*, un dispositivo meccanico privo di tecnologia digitale. Questo dispositivo era in grado di proiettare film ricreando odori e vibrazioni coerenti con le scene mostrate ed utilizzava la visione stereoscopica, concetto chiave della moderna VR.

Lo sviluppo della tecnologia si è però allontanato da questo concetto che prevedeva il coinvolgimento della totalità dei sensi a favore di macchine dedicate ai sensi più facilmente stimolabili: udito e vista.

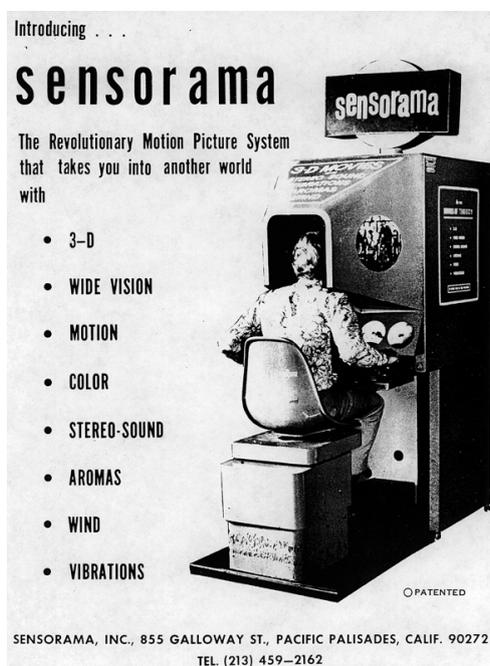


Figura 3.1: Originale locandina di presentazione del *Sensorama*.

Nel 1968 nasce il concetto di *HMD*, *head mounted display*, con la creazione di Ivan Sutherland, chiamata *Spada di Damocle* (Figura 3.2) dalla forma del dispositivo che risultava talmente pesante da essere indossato da dover venire fissato al soffitto della stanza. Le immagini consistevano nella versione *wireframe* di varie stanze navigabili generate da un comune computer e inviate su di un display stereoscopico inserito nell'headset. Proprio questa idea di display da indossare è ciò che è ancora oggi alla base dei più famosi e diffusi dispositivi VR.

### 3.1. L'EVOLUZIONE DELLA VR

---

Ovviamente sia la qualità delle immagini che la scomodità di utilizzo dei dispositivi sono migliorate di molto, ma restano comunque i due parametri che ancora necessitano di ulteriori migliorie per ampliare il bacino d'utenza di questi dispositivi.

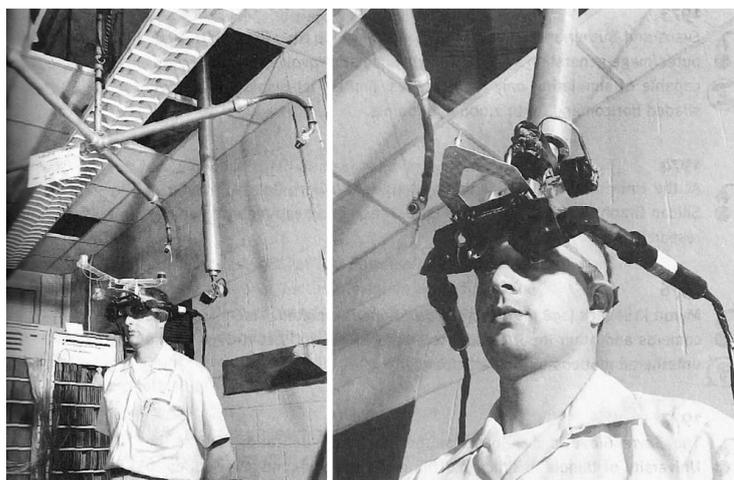


Figura 3.2: L' head mounted display *Spada di Damocle*.

Per due decenni queste tecnologie vennero sviluppate presso varie università, affinandone sempre più le qualità e, soprattutto, riducendone le dimensioni permettendo il lancio sul mercato dei primi dispositivi VR commerciali tra cui il *Mattel Power Glove* (Figura 3.3) di Nintendo. Questa periferica consiste in un guanto indossabile che riconosce i gesti della mano.

L'azienda giapponese si è sempre dimostrata grande innovatrice dell'industria e cercò di sfruttare il crescente desiderio del pubblico verso questa tecnologia, che però si dimostrò incapace all'epoca di stare al passo con le alte aspettative dei consumatori. Il *Mattel Power Glove* infatti, per rientrare in una fascia di prezzo da dispositivo *consumer*, rinunciava alle alte prestazioni; gli iniziali prototipi prevedevano l'utilizzo di un guanto cablato a fibre ottiche, ma si optò per una tecnologia basata su resistenze variabili e ultrasuoni, più robusta ed economica. Sulle dita del guanto sono montate delle strisce che poggiano su superfici con diversa intensità elettrica, di modo che, piegando le dita, la striscia si possa muovere facendo variare la quantità di energia ricevuta dal controllore; ogni dito è fornito di quattro superfici per un totale di due bit d'informazione disponibile per ogni dito. Per il rilevamento spaziale sono montati sul guanto due emettitori di ultrasuoni e un ricevitore, per ricavare la posizione del guanto per triangolazione, che va posizionato sopra al televisore.

### 3.1. L'EVOLUZIONE DELLA VR

---



Figura 3.3: *Mattel Power Glove* di Nintendo.

Il pubblico abbandonò rapidamente l'utilizzo di questa periferica, complice la presenza di pochi titoli che la supportassero e la scarsa efficacia dei controlli.

Tra la fine degli anni '80 e l'inizio dei '90 l'interesse generale per questa nuova frontiera della tecnologia era molto elevato, ma finì, come anticipato, per scemare rapidamente portando a un blocco per lo sviluppo di dispositivi commerciali; questo avvenne per tre principali motivi: la necessità di workstation di fascia alta (molto costose) per gestire i calcoli computazionali volti a creare la simulazione; la mancanza di interfacce ad alta qualità come display ad alta risoluzione o strumenti di input di precisione e la mancanza di software di qualità dedicati allo sviluppo, progettazione e gestione della simulazione virtuale creata, oggi comunemente chiamati *game engine*.

Il cofondatore di Intel, Gordon Moore, nel 1965 scrisse quella che divenne successivamente nota come la *prima legge di Moore*:

«La complessità di un microcircuito, misurata ad esempio tramite il numero di transistori per chip, raddoppia ogni 18 mesi (e quadruplica quindi ogni 3 anni).»

Questa legge rimase rispettata fino agli anni duemila, quando l'avanzamento tecnologico portò le GPU a superare gli aumenti delle performance previsti da Moore dando la possibilità di gestire milioni di poligoni texturizzati al secondo ed effetti grafici complessi come riflessioni e rifrazioni, il tutto con apparecchi dal costo di poche centinaia di Euro. La causa di questa spinta al progresso è da ricercarsi nell'industria dei videogiochi che, già all'inizio degli anni duemila, risultava essere un mercato dal grande valore economico.

Nascono così diversi dispositivi low-cost, ma dalla buona precisione, affiancati da nuovi dispositivi di visualizzazione di grandi dimensioni, su cui era più semplice lavorare grazie a vari *software open source* dedicati alla creazione di ambienti, all'audio processing ed alla gestione delle simulazioni complesse, ovvero i *Game Engines*.

Oggi abbiamo sul mercato diversi visori che differiscono per prezzi e prestazioni andando incontro a diverse fasce d'utenza.

Tra i principali ricordiamo **Play Station VR**, il visore di Sony lanciato il 16 ottobre 2016 ed usato come periferica per *Play Station 4* in combinazione con *Play Station Camera* e *Play Station Move* che ha come punto di forza la qualità delle esperienze esclusive per questa piattaforma; **Oculus Rift**, visore lanciato sul mercato il 14 marzo 2014 grazie ad un finanziamento su *Kickstarter*, è un device molto costoso, ma dalle alte performance, infatti offre un campo visivo doppio rispetto a quello proposto dalla concorrenza; e, infine, **Gear VR**, particolare visore che sarà descritto approfonditamente in seguito in quanto è stato il device scelto per la pubblicazione del videogioco create in azienda per questa tesi.

Quello che accomuna questi prodotti odierni è l'idea di abbandonare la ricerca di una simulazione virtuale completa per andare a focalizzarsi sui sensi più facilmente stimolabili, ovvero vista e udito, cercando di essere il meno ingombrati possibile e facilmente utilizzabili da chiunque, anche da chi non è avvezzo ad usare dispositivi tecnologici complessi.

Nonostante tutto la VR non è ancora fenomeno di massa soprattutto perché l'utente, diventato ormai abituato ai più alti livelli di fotorealismo portati dai videogame fruibili su classici display, tende a notare più facilmente ogni incoerenza con la realtà e i suoi sensi risultano più difficili da ingannare; inoltre, la moderna VR si concentra maggiormente sui sensi più facili da stimolare, vista e udito, ma poco è stato fatto per coinvolgere gli altri e, spesso, alcune interfacce risultano invasive, provocando nausea e fastidi dopo usi prolungati in special modo agli utenti che non hanno mai avuto esperienza con questo tipo di tecnologia.

## 3.2 La tecnologia della VR

Qualunque sia il device usato per la VR, questo dovrà obbligatoriamente essere composto da dispositivi per l'input e per l'output delle informazioni che possono essere debitamente classificati. I dispositivi di input si occupano di raccogliere le informazioni ottenute dall'utente, come ad esempio, la sua posizione nello spazio; mentre starà ai dispositivi di output dare un feedback percepibile da uno dei cinque sensi dell'utente a seconda dell'informazione precedentemente ottenuta ed elaborata.

Per i primi si possono attuare tre diverse classificazioni: li si può catalogare secondo le caratteristiche dei dati che trasmettono, secondo i gradi di libertà che concedono o secondo il tipo di sensore che utilizzano; nel primo caso si possono suddividere in

- **Discreti:** questi generano un solo evento alla volta in risposta ad azioni dell'utente come il premere o rilasciare un pulsante.

### 3.2. LA TECNOLOGIA DELLA VR

---

- **Continui (o campionati):** generano informazioni autonomamente (passivi) o in risposta alle azioni dell'utente (attivi) come i *tracker* posizionali.
- **Ibridi:** che, come suggerisce il nome, uniscono le due tipologie di dispositivi per l'input precedenti; ad esempio il mouse che rileva la posizione come un dispositivo continuo, ma può misurare l'evento del click come discreto.

Con la catalogazione in gradi di libertà, ovvero il numero di parametri che controllano, si suddividono in:

**1 DOF:** un solo grado di libertà, ad esempio un pulsante che può essere premuto o a riposo.

**2 DOF:** due gradi di libertà, come la posizione su di un piano nelle coordinate (x, y).

**3 DOF:** tre gradi di libertà, posizionamento su di un piano più l'eventuale evento di click, classico esempio del mouse.

**6 DOF:** sei gradi di libertà, posizione nello spazio 3D nelle coordinate (x, y, z) più l'orientamento lungo i tre assi.

**6+N DOF:** oltre alla posizione e all'orientamento viene tenuto conto di un numero variabile di gradi di libertà, controllabili, ad esempio, da una tastiera.

Infine, con la catalogazione secondo il tipo di tecnologia usato dai dispositivi di input abbiamo:

- **Dispositivi magnetici:** tre sorgenti fisse generano altrettanti campi magnetici ortogonali tra loro all'interno dei quali si muove un *receiver* di cui possiamo sempre avere la posizione. Subiscono interferenze da campi magnetici esterni quindi vanno usati in condizioni ottimali per evitare rumore nelle misurazioni.
- **Dispositivi acustici:** usano suoni a bassa frequenza emessi da una sorgente triangolare composta da tre altoparlanti di modo che il ricevitore possa avere tre diverse misurazioni in ogni istante per poter triangolare la posizione della sorgente. Il dispositivo risulta economico, ma la sua precisione varia a seconda delle condizioni dell'ambiente come temperatura e umidità, o semplicemente dalla quantità di rumori esterni.
- **Dispositivi ottici:** ve ne sono di due tipologie, la prima, denominata *outside looking in*, usa sensori che emettono segnali luminosi nell'ambiente, che vengono catturati da appositi *marker* posti sul soggetto, dei quali posso calcolare la posizione per triangolazione; la seconda tipologia chiamata *inside looking out* si ha posizionando i sensori sul soggetto in movimento, questi misurano il movimento del soggetto rispetto all'ambiente fisso circostante.

### 3.2. LA TECNOLOGIA DELLA VR

---

- **Dispositivi inerziali:** utilizzano sistemi microelettromeccanici (MEMS) che, attraverso tre giroscopi perpendicolari tra loro, calcolano le velocità angolari dei movimenti; stessa cosa fanno tre accelerometri per le accelerazioni sugli stessi assi. Posso quindi ottenere l'orientamento lungo gli assi, le accelerazioni e la posizione dell'oggetto. Essendo che le misure si ottengono per integrazione, il rumore su di esse non si elimina, ma si somma portando all'errore di *drift*; per eliminarlo si deve utilizzare un altro tipo di tracker per resettare le misure.
- **Dispositivi meccanici:** le strutture cinematiche che compiono i movimenti hanno dei sensori ai giunti che misurano gli angoli tra i segmenti mobili di cui le strutture sono composte. Precisi e privi di interferenze, ma ingombranti e scomodi da utilizzare per lunghi periodi. Utilizzati soprattutto per compiti che richiedono estrema precisione.
- **Dispositivi ibridi:** sfruttano due o più delle precedenti tecnologie per coprire le criticità di una singola tipologia di dispositivo, ma sono spesso più complessi rispetto agli altri nel loro utilizzo.



Figura 3.4: Mouse e tastiera, classici esempi di dispositivi di input usati quotidianamente.

I dispositivi di output si possono semplicemente catalogare a seconda di quale dei cinque sensi portano le informazioni; ovviamente un sistema VR idealmente perfetto dovrebbe avere informazioni su tutti questi canali, ma nella maggior parte dei casi ci si limita ad avere output video e audio, quindi ci concentreremo su questi.

Per i display usati nelle applicazioni VR dei recenti headset, va ricordato che lo spettatore non guarda direttamente lo schermo, bensì la sua immagine viene filtrata da due lenti, una per occhio, che ricoprono molteplici ruoli: concentrano la luce per far sembrare gli schermi infinitamente lontani, ingrandiscono il display in modo da nascondere i lati dello schermo allo spettatore, regolano la messa a fuoco che è variabile da utente a utente e, in alcuni casi come con *Gear VR*, creano l'effetto 3D vero e proprio.

### 3.2. LA TECNOLOGIA DELLA VR

---

Per la scelta dei corretti display visuali bisogna tenere sempre conto che l'uomo vede la realtà attraverso la visione binoculare che coinvolge due processi distinti: **la percezione simultanea**, ovvero la percezione da parte dei due occhi di due immagini leggermente diverse e la **stereopsi**, cioè l'elaborazione da parte del cervello delle due immagini ricevute per avere un risultato visivo finale da cui possiamo estrapolare le informazioni legate al concetto di profondità nello spazio.

Una delle più famose tecniche per visualizzare immagini visibili in 3D è il cosiddetto **anaglifo**: l'immagine veniva ripresa con due camere spostate orizzontalmente di pochi centimetri, per simulare la distanza tra gli occhi dell'osservatore, e le due immagini prevedevano un filtraggio cromatico per essere viste con la corretta simulazione del 3D tramite particolari occhialini a lenti colorate che filtrano il colore ad esse complementare. Questa semplice tecnica a basso costo ha il suo periodo di massimo utilizzo tra gli anni '50 e '60, ma le sue basse prestazioni non gli hanno permesso di trovare utilizzi in apparecchi moderni.

Un'altra tecnica consiste nell'utilizzo di lenti polarizzate montate su appositi occhiali; anche in questo caso le immagini che vediamo sempre due, ma vengono alternate per ogni fotogramma ad una frequenza tale da non essere di disturbo all'osservatore. Le due immagini mono polarizzate in modo da essere viste solo attraverso la rispettiva lente, mantenendo la divisione tra ciò che percepisce l'occhio destro e l'occhio sinistro. La qualità dell'immagine è migliore rispetto ad altre tecnologie, soprattutto perché lascia inalterati i colori naturali.

Oppure si può ricorrere alla tecnica dell'**alternate image**, o **shutter glasses**, che prevede l'utilizzo di particolari occhiali che sono dotati di otturatori mobili che bloccano e permettono la vista in alternanza; nello stesso tempo, su schermo, vengono alternate le immagini dirette all'occhio destro e sinistro in sincronia con l'azione degli occhiali, dando l'illusione della visione stereoscopica senza affaticare l'osservatore.

Per ricreare l'effetto artificialmente nelle applicazioni VR si usano per lo più display stereoscopici che generano due immagini diverse, una per occhio, da mostrare su altrettanti display oppure su un unico display, ma separabili tramite particolari occhialini o lenti che, sfruttando tecniche differenti, mostrano immagini separate che saranno poi unite tramite il processo della stereopsi per avere il risultato desiderato. L'effetto può essere dato con la stereoscopia attiva, cioè rendendo opaca una lente alla volta e alternare le immagini per l'occhio che, di volta in volta, vede lo schermo; o con la stereoscopia passiva che usa particolari filtri polarizzati per separare le componenti della luce per i due occhi, tecnica a basso costo, ma dalla qualità inferiore.

### 3.2.1 Rendering stereoscopico

Per creare rendering stereoscopici (dei quali vediamo un esempio nella Figura 3.5) oggi possiamo usare molti dei comuni software per modellazione e animazione; in molti di questi è infatti possibile settare appositi parametri per creare due rendering da posizioni parallele con distanza di camera uguale alla distanza fra gli occhi dell'osservatore e poi sovrapporli nel modo corretto in una sola immagine.

Tra i vari parametri settabili c'è anche la scelta del filtro che si vuole usare per visualizzare l'immagine finale, di default questo sarà il filtro cromatico rosso-ciano degli anaglifi.

Il risultato ottenuto si può visualizzare semplicemente utilizzando gli appositi occhiali.

I rendering vanno comunque creati tenendo conto di particolari regole che servono a garantire la corretta visione dell'effetto 3D.

**Regola 1** - Vanno evitati i colori troppo saturi per ovviare ad eventuali aberrazioni cromatiche.

**Regola 2** - Va posta attenzione a quegli elementi che ci danno indizi sulla profondità degli oggetti come ombreggiature ed occlusioni per aumentare il realismo della scena.

**Regola 3** - Gli oggetti non devono essere tagliati ai bordi dell'immagine poiché la cosa toglierebbe l'illusione di tridimensionalità alla scena.

Vengono di seguito presentati i passi computazionali per la realizzazione di un rendering grafico stereoscopico.

- 1) **Application:** questa parte avviene a livello software e comprende la lettura degli input e il calcolo delle conseguenze sulle geometrie come modifiche alla posizione, cambi di vista e collisioni.
- 2) **Geometry:** avviene a livello hardware e comprende l'applicazione di tutte le trasformazioni agli oggetti virtuali precedentemente calcolate (spostamenti, rotazioni e scalamenti); in questa fase vengono anche calcolate illuminazione ed il mapping delle texture.
- 3) **Rasterizer:** avviene a livello hardware come il precedente step ed in questa fase vengono colorati i singoli pixel a seconda delle informazioni di texture, illuminazione e alla proiezione 2D dei modelli tridimensionali; inoltre vengono determinate le superfici visibili tramite la tecnica dello *z-buffer*.

Fondamentale per la riduzione della complessità di calcoli e mantenere un'alta fluidità per ingannare l'occhio umano è ridurre il numero di poligoni delle singole mesh ed il numero delle luci; inoltre bisogna evitare l'uso eccessivo di effetti grafici come il *motion blur*, preferendo cercare soluzioni più "rozze" seppur con risultato simile, ma meno onerose in termini di calcolo computazionale.

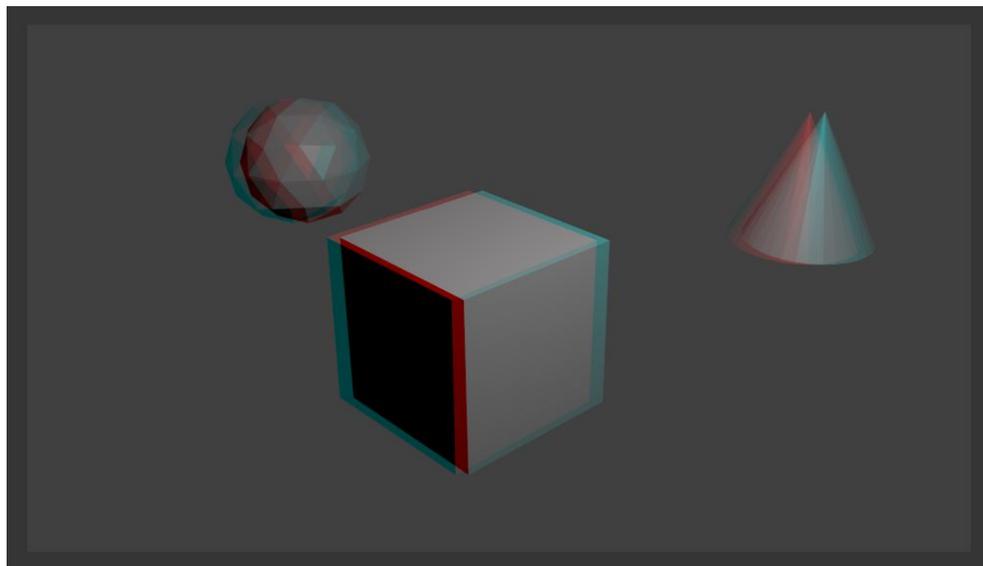


Figura 3.5: Semplice esempio di rendering stereoscopico ottenuto con *Blender*, pronto per essere visualizzato con occhiali a filtri cromatici rosso-ciano.

### 3.2.2 Audio stereoscopico

Per avere un alto senso di immersione risulta necessario un buon 3D audio display che può portare suono monoaurale, ovvero lo stesso suono ad entrambe le orecchie, o binaurale, se si porta un suono diverso per ogni orecchio. Vi sono diversi sistemi di audio display, i più comuni sono i sistemi a due canali (o stereo) che hanno un canale dedicato per ogni orecchio ed il sistema multicanale, che, aumentando il numero di canali disponibili, fa aumentare il range di possibili punti di piazzamento della sorgente sonora virtuale nello spazio.

In una corretta simulazione è necessario non solo posizionare la sorgente sonora nello spazio 3D, ma anche calcolare tutti quei fenomeni acustici che si verificano quando il suono interagisce con l'ambiente, come riflessioni o echi, gestendone l'intensità a seconda di parametri come il materiale delle superfici dell'ambiente o le caratteristiche del mezzo di trasmissione del suono, ovvero dell'aria, nella simulazione.

Vediamo brevemente i passi fondamentali del rendering audio.

- 1) **Calcolo della posizione e della distanza delle sorgenti:** si riduce il volume con la distanza dalla sorgente dividendo lo spazio in aree ad ugual volume per semplificare i calcoli.
- 2) **Culling e clustering delle sorgenti:** sempre per semplificare i calcoli computazionali si annullano le sorgenti audio che generano suoni sotto la soglia dell'udibile o che sono coperte da altre sorgenti. In più si rappresenta l'insieme delle sorgenti come un'unica sorgente e ne si calcola la posizione tenendo conto dei volumi delle altre.

### 3.3. SAMSUNG GEAR VR

---

- 3) **Calcolo delle oclusioni:** si calcola il percorso diretto, riflessioni, ostruzioni ed esclusioni dei suoni.
- 4) **Auralizzazione:** è il processo che sintetizza i calcoli fatti per simulare l'ascolto binaurale.

L'audio risulta estremamente importante per creare quell'immersione ricercata dalle applicazioni VR poiché, il suo corretto utilizzo, oltre ad essere importante di per sé, va a direzionare lo sguardo dell'utente, che, sentendo un suono provenire da una determinata direzione, sarà portato a voltarsi verso la sorgente del suono, o ad allontanarsene, a seconda della tipologia di suono riprodotta. Un abile designer di questo tipo di esperienze può prevedere le reazioni dello spettatore usando il suono per guidarlo in una determinata direzione, facendolo avanzare nel mondo virtuale.

### 3.3 *Samsung Gear VR*

Analizziamo ora in particolare il *Samsung Gear VR*, dispositivo VR di riferimento per il prodotto sviluppato presso KR Games per questa tesi.



Figura 3.6: *Samsung Gear VR* versione SR 325.

Samsung Gear VR è un headset per realtà virtuale sviluppato da Samsung Electronics, in collaborazione con Oculus.

L'headset, rilasciato il 27 novembre 2015, necessita di uno smartphone *Samsung Galaxy* compatibile (*Galaxy Note 5*, *Galaxy S6/S6 Edge/S6 Edge+*, *Galaxy S7/S7 Edge*, o *Galaxy S8/S8+*) che va inserito nell'apposito supporto collegandosi via micro-USB e va a fungere

## 3.4. GAMING VR

---

da display visibile attraverso due lenti con *field of view* di 96° (poi 101° con la versione *R323* e seguenti), mentre il dispositivo in sé è adibito a controller e gestisce il tracking tramite IMU (unità di misura inerziale) ovvero un sistema basato sull'utilizzo di sensori inerziali quali accelerometri e giroscopi per calcolare la posizione dell'utente.

Inoltre, l'headset include un touchpad, due pulsanti adibiti alla funzione di *back* e *home* sul lato destro dello stesso, e una rotellina per la calibrazione della messa a fuoco posta sulla cima della mascherina.

L'ultimo modello di *Gear VR* denominato *SM-R325* è stato rilasciato il 15 settembre 2017; questo garantisce il supporto a *Samsung Note 8* e include un controller retrocompatibile con precedenti versioni del device. L'intuitiva interfaccia è basata sulla tecnologia *Oculus*: ci si orienta semplicemente ruotando la testa e si accede ad un contenuto con un tocco sul touchpad laterale, oppure usando il controller dell'ultima versione del device.

Oltre al classico utilizzo personale, *Gear VR* si presta ottimamente anche ad essere usato in altri ambiti come la didattica o il turismo. Ad esempio è usato nel parco divertimenti *Bobbejaanland*, in Belgio, per rendere ancora più adrenaliniche le montagne russe, ricreando l'eruzione di un vulcano durante la corsa della giostra; ma potrebbe ad esempio essere usato in musei o mostre per aumentare l'interattività con le opere esposte coinvolgendo il pubblico in modo più profondo di quanto è normalmente possibile fare.

Il successo di questo visore è da ricollegare a due fattori principali: la semplicità di utilizzo, dato che non ha necessità di cavi o altri strumenti ingombranti, risultando leggero e comodo; e il prezzo competitivo rispetto ai concorrenti, pur mantenendo una buona qualità generale sia per il visore in sé, come struttura e materiali usati, sia per la resa visiva offerta, che, naturalmente, varia a seconda dello smartphone usato come display.

## 3.4 Gaming VR

In questa sezione guarderemo alla questione VR non sul lato tecnico, ma analizzeremo come questo mezzo va ad agire sul mondo del videogame ricordando che oggi siamo nel periodo che in futuro sarà probabilmente visto come la nascita di quello che sarà lo standard, ovvero il *gaming VR*.

### 3.4.1 La lenta diffusione

In passato vi sono stati diversi tentativi di introdurre la VR nel mondo dei videogiochi, ma tutti si sono rivelati infruttuosi ed hanno contribuito a far accrescere quello scetticismo verso questa nuova tecnologia che ancora adesso perdura; tra questi ricordiamo lo sfortunato caso del *Virtual Boy* (Figura 3.7), console prodotta e sviluppata da Nintendo e distribuita in Giappone il 21 luglio 1995 (14 agosto negli USA), la sua produzione è terminata dopo un solo anno a causa del grande insuccesso commerciale.

---

Bibliografia [13]

### 3.4. GAMING VR

---

Questo particolare headset sfruttava un macchinoso sistema di specchi e lenti oscillanti per proiettare immagini 3D dei giochi, che però potevano essere colorate solo con rosso o nero, un enorme passo indietro rispetto al miglior impatto visivo dei giochi 16 bit dell'epoca; a questo si aggiunge poi un marketing discutibile che descriveva la console come "portatile", quando le dimensioni generose e il peso ne rendevano difficile l'utilizzo fuori casa ed elencava una serie di effetti collaterali dovuti al *motion sickness* che scoraggiava all'acquisto i genitori più preoccupati.

La console vendette pochissimo quindi i giochi che la supportavano smisero velocemente di venire prodotti e il progetto *Virtual Boy* venne rapidamente abbandonato a favore del più fortunato *Nintendo 64*. Fu quindi la scarsa qualità permessa dalle tecnologie di livello consumer del tempo a decretarne l'insuccesso commerciale.



Figura 3.7: L'headset di Nintendo: *Virtual Boy*.

Altro esempio della difficile situazione di questa tecnologia è il ben più recente *Kinect One* (Figura 3.8). Nel 2010, per approfittare della scia del successo incredibile ottenuto da Nintendo con il controller di movimento *Wii Mote*, che riuscì a vendere oltre 8 milioni e mezzo di unità nei soli Stati Uniti imponendosi nella fascia di pubblico denominata "*casual gamers*", venne commercializzato il primo *Kinect*, una periferica sviluppata da Microsoft per la console *XBox 360*, che legge il movimento del corpo usando un emettitore di raggi infrarossi che colpisce persone e oggetti presenti davanti allo schermo; un ricevitore integrato vede i punti d'impatto dei raggi infrarossi e ne calcola le posizioni permettendogli di capire dove si trovino i giocatori per utilizzare questa informazione come input per le azioni di gioco, rimuovendo ogni tipo di device fisico per il controllo.

Il successo iniziale di questa periferica venne meno quando, con l'avvento di *XBox One*, nuova versione della console Microsoft, il *Kinect 2.0* venne venduto insieme alla console aumentandone il prezzo di vendita di 100 dollari, costringendo anche i videogiocatori

### 3.4. GAMING VR

---

tradizionalisti ad acquistare una costosa periferica vista come "superflua", mentre per i giocatori occasionali non vi era un reale motivo per acquistare una costosa nuova console avendo a disposizione il modello precedente.

Il malcontento generale degli acquirenti ha portato all'allontanamento forzato da Microsoft di Don Mattrick, capo dell'Entertainment Business e uomo chiave della divisione *XBox*.



Figura 3.8: La periferica per *XBox One* di Microsoft: *Kinect One*.

#### 3.4.2 Un nuovo livello d'immersione

Va però detto che la VR porta in questo ambiente un livello di immersione che mai prima d'ora si era visto e il valore potenziale di questa tecnologia è inimmaginabile, ma appunto, ancora non sfruttato appieno; il suo sviluppo avverrà solo qualora le grandi aziende vi investiranno tempo e denaro per produrre esperienze di gioco di qualità con meccaniche pensate appositamente per queste nuove piattaforme.

Importante ricordare che ora come ora e per i prossimi anni, il gaming VR sia solamente una componente accessoria, un graditissimo surplus del gaming tradizionale e non una sostituzione o una proposta universale come fu per *Nintendo Wii* con il *motion control*.

La miglior caratteristica di questo nuovo modo di giocare che già possiamo vedere è appunto l'immersione: non si tratta di una mera illusione, i più moderni dispositivi VR ingannano i sensi e l'utente si sente davvero dentro al mondo virtuale e, soprattutto per i giochi che conservano i classici controller per l'interazione, non scompare la complessità di gameplay a favore di una sola esperienza visiva.

### 3.4. GAMING VR

Aumenta quindi l'emozione anche per le più semplici interazioni con l'ambiente: anche il banale raccogliere un oggetto appoggiato su di un tavolo semplicemente avvicinando la mano a questo e chiudendo le dita si rivela una nuova esperienza e l'utente, da giocatore, diventa il protagonista dell'azione.

Questo nuovo paradigma può porre l'utente al centro di situazioni che non potrebbe, o non vorrebbe, vivere nella vita vera, facendogli vivere un'esperienza percepita come reale, ma in un mondo di simulazione virtuale.

Bastano poche immagini per capire la diversa profondità delle meccaniche e la differenza nella complessità delle azioni richieste nella maggior parte dei titoli pensati esclusivamente per il VR: da una parte le esperienze VR ci richiedono di interagire semplicemente con l'ambiente imitando in molti casi le azioni reali, mentre le produzioni classiche spingono il giocatore a navigare attraverso complessi menù di statistiche per evolvere il proprio personaggio o a compiere complesse combinazioni di tasti molto poco intuitive.



(a) La schermata di un'esperienza VR priva d'interfaccia a schermo per rendere intuitive le azioni.



(b) La complessità dei menù e delle interfacce dei videogame classici.

Figura 3.9: Confronto tra le interfacce di un'esperienza in VR, (a), e quella di un videogame tradizionale, (b).

### 3.4. GAMING VR

---

Questa naturalezza nelle interazioni porta, anche per i giocatori più esperti, alla diminuzione della *situation awareness*, rendendo più complesso e caotico gestire gameplay particolarmente frenetici, scoraggiando il giocatore che vive il gioco come un meccanismo che deve essere preciso, efficiente e completamente controllabile ad avvicinarsi a questo nuovo tipo di videogame.

Ci sono però particolari generi di giochi che traggono solo vantaggi da questo nuovo mezzo, in particolare parliamo dei **giochi su rotaia** e **simulatori di volo o di guida**, questo perché il poter muovere la testa durante un movimento di traslazione e poter quindi guardare l'abitacolo o l'interno del proprio mezzo durante una corsa è una possibilità non ottenibile con altri mezzi senza sacrificare troppo il controllo che il player ha sull'azione, allo stesso modo il poter mirare muovendo lo sguardo verso il bersaglio si fonde perfettamente alle meccaniche di un classico sparatutto su rotaie.

In più, è il realismo a guadagnare maggiormente dall'utilizzo della VR per questo genere di esperienze, poiché, il movimento del nostro avatar è gestito esattamente come lo sarebbe nella vita reale, ovvero premendo un pedale, o in questo caso un tasto, adibito all'acceleratore, e sterzando tramite un volante od una cloche, rispettando quelle che sono le nostre esperienze di vita reale nel mondo virtuale; nel caso invece si controlli un personaggio umano in un videogioco VR, subentrerà subito una sensazione di estraniamento dovuta al fatto che controlliamo i passi del nostro avatar tramite input dati dalle mani.

#### 3.4.3 Le problematiche

Vanno però evidenziati alcuni contro in questo tipo di esperienze, in particolare due: la mancanza di riscontro fisico sui movimenti e le sollecitazioni che il proprio avatar subisce e la qualità della risoluzione delle immagini. Il primo può portare in alcuni casi una sensazione di disagio nell'utente poiché, ad un dato movimento del personaggio, il nostro cervello si aspetta una sollecitazione fisica che crede istintivamente di ricevere pur non esistendo portando, in alcuni casi, al *motion sickness*, ovvero una sensazione di nausea e malessere. Il secondo problema è senza dubbio il più grave, poiché ad oggi, le immagini di questi giochi risultano spesso sfocate laddove si voglia ricreare geometrie complesse, questo perché la tecnologia ancora non consente di avere una risoluzione che sia comparabile a quella raggiungibile dal gaming tradizionale, pur mantenendo ragionevoli i prezzi del device e stabile il framerate dell'applicazione.

Quest'ultima è una condizione fondamentale per la buona riuscita di un'applicazione VR dato che sotto i 60 fps queste perdono di realismo poiché l'occhio ne riconosce l'artificialità. Inoltre, questa "sfocatura" crea rapidamente una certa fatica negli occhi dell'utente anche con brevi sessioni da 40 minuti circa, in special modo per chi non è abituato all'utilizzo di visori.

### 3.4. GAMING VR

---

Vi è poi un motivo meno tecnico, ma insito nella natura del videogiocatore sul perché il VR ancora non sia di massa, per comprenderlo analizziamo brevemente il pubblico videogiocante delineando tre macro categorie: *hardcore* (o semplicemente *Core gamers*), *casual gamers* e *conscious gamers*.

Queste tre tipologie di utenza, pur non avendo una definizione ufficiale, sono comunemente adottate da appassionati ed esperti del settore.

La prima categoria identifica la classica concezione del videogiocatore: questi acquista svariate decine di titoli all'anno, resta sempre informato sulle novità e sulle nuove tecnologie, è molto critico rispetto ai contenuti di un titolo privilegiando esperienze di gioco complesse e poco guidate ed è tipicamente diffidente dei nuovi sistemi di controllo e, in generale, dei cambiamenti in questo ambiente che non vive solo come hobby, ma come una passione in cui investire tempo e denaro.

La seconda categoria, ad oggi numericamente maggiore rispetto alle altre, accoglie tutti i giocatori che usufruiscono di pochi titoli all'anno, scegliendoli non per qualità intrinseche, ma spesso per moda o grazie alle influenze delle campagne marketing; spesso ignorano totalmente l'esistenza di intere saghe storiche prediligendo esperienze più rapide, fruibili su mobile e vedono il gaming come un semplice passatempo. Questa categoria è nata più di recente, con il passaggio del videogame da fenomeno di nicchia per pochi "nerd" a fenomeno di massa.

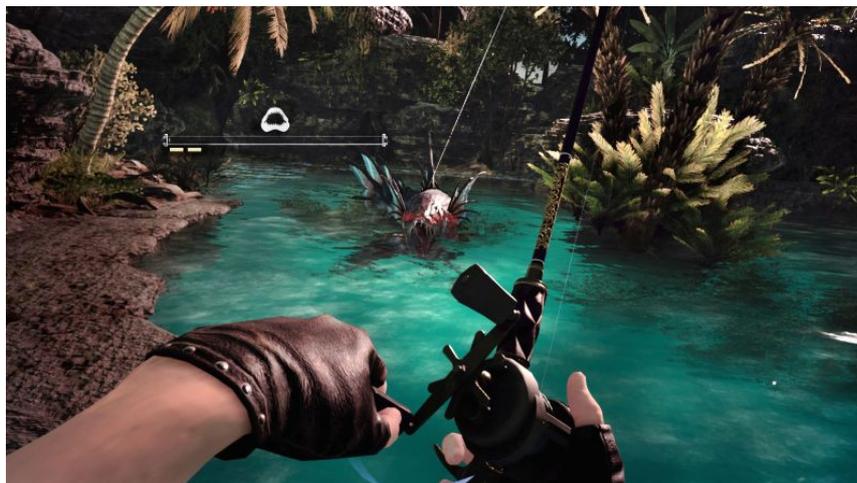
La terza categoria pur se probabilmente esistente da diverso tempo, ha ricevuto un nome ed una effettiva definizione ancora più recente poiché è nata dal bisogno di identificare una parte dei *Core Gamer*, ovvero quella parte che non si limita a videogiocare, ma che va ad analizzare il medium in profondità, informandosi di tutto ciò che sta dietro al prodotto finito, dalle strategie di mercato, alle differenze culturali che portano alla modifica di alcune esperienze in certi paesi, dalle diverse fasi di creazione del gioco, alle influenze artistiche presenti nel gioco stesso, ed è un pubblico che, in generale, è conscio di cosa gioca e della qualità del prodotto che ha tra le mani.

Il VR ora come ora risulta troppo giovane per avere titoli dalla qualità tale da attirare gli *hardcore gamer* ed è troppo costoso, almeno per piattaforme VR ad alta qualità, per grande parte dei *casual gamers*, che non desiderano spendere svariate centinaia di euro per un semplice passatempo.

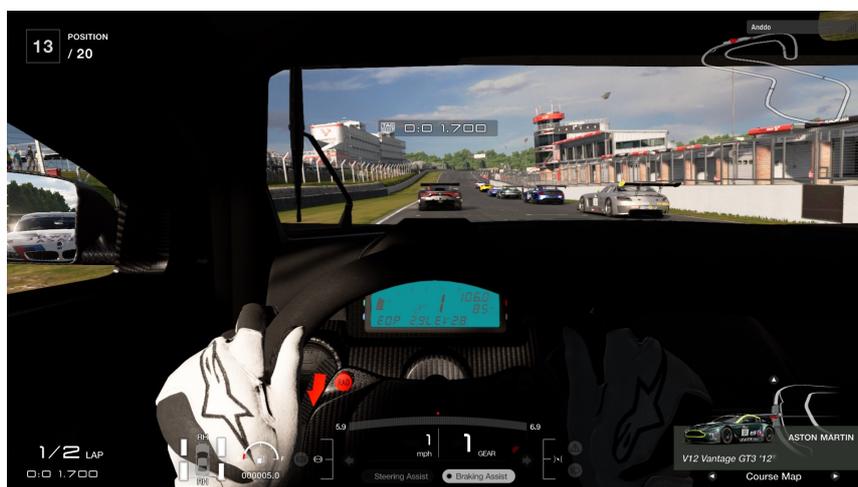
Il gaming VR rispetto a quello tradizionale si conferma quindi un gradevole extra, ma per l'appassionato in cerca di grandi esperienze videoludiche occuperà una quantità di tempo infinitesimale rispetto al gioco tradizionale e questa tesi è avvalorata dal fatto che i videogiochi di alto profilo si limitino ad integrare la componente VR con piccole modalità dedicate separate dal resto del gioco: tra questi vale la pena nominare *Gran Turismo* (Figura 3.10 (b)) e *Final Fantasy XV* (Figura 3.10 (a)) essendo titoli di reale richiamo mondiale.

### 3.4. GAMING VR

---



(a) *Fotogramma da Final Fantasy XV.*



(b) *Fotogramma da Gran Turismo.*

Figura 3.10: Le sezioni in VR di *Final Fantasy XV*, (a), e *Gran Turismo*, (b), su *Play Station VR*, inerte come componenti accessorie dei giochi originali tramite patch, dopo la loro pubblicazione.

Per gli anni a venire è dunque improbabile che la VR diventi di massa, ma arriverà il momento in cui gli ormai classici schermi piatti *Full HD*, verranno considerati obsoleti e saranno soppiantati dai più moderni visori.

### 3.4.4 Differenze d'approccio rispetto al gaming tradizionale

Oltre alle ovvie differenze tecnologiche ci sono altre piccole, ma altrettanto importanti differenze d'approccio a questo nuovo modo di videogiocare rispetto a quello tradizionale che vanno tenute in considerazione qualora si decida di esplorare in prima persona questa nuova tecnologia.

- **Lo spazio necessario:** con i classici videogame una TV, un controller wireless e la console sono sufficienti a giocare, ma per esperienze VR serve spesso una stanza dedicata, priva di ostacoli o altri oggetti ingombranti, dato che molti titoli richiedono di stare in piedi e muoversi per controllare il proprio avatar. Questa apparentemente semplice condizione non è sempre facile da soddisfare.

- **Il tempo:** uno dei fattori che ha portato alla fortuna del gaming su console tradizionale è il cosiddetto *plug & play* ovvero il fatto che, accesa la console, si può giocare senza ulteriori perdite di tempo in vari tipi di settaggi; il VR necessita invece di una fase di vestizione dell'headset e di calibrazione che, talvolta, è scomoda eseguire da soli.

Va detto che con le ultime generazioni di console, il fattore *plug & play* sta man mano venendo meno dato che queste, sempre più simili a PC, spesso necessitano di scaricare aggiornamenti o patch ed i giochi richiedono di installare grandi quantità di dati sulla piattaforma utilizzata prima di poter iniziare ad utilizzarli.

- **La differenza delle esperienze:** come già anticipato, la VR per ora tende a sfruttare la sua grande capacità di far immergere l'utente nel mondo virtuale tralasciando, però, la componente di interattività avanzata propria dei videogame più complessi, ergo molti giocatori di vecchia data snobbano queste nuove esperienze vedendole come qualcosa di alieno rispetto ai canoni classici dei videogame. Questo problema sussiste comunque da diversi anni anche per alcuni dei videogame fruibili classicamente su una TV, come ad esempio i recenti giochi sviluppati da *Quantic Dream*, da molti identificati come "film interattivi" piuttosto che videogiochi.

La discussione su questo argomento è ancora aperta e sarà probabilmente impossibile trovare una linea netta per definire il limite tra videogame classico e esperienza interattiva moderna.

- **I costi:** ovviamente va tenuto conto del fatto che visori di alta qualità possono venire a costare oltre i 400€ ed offrono poi un parco titoli numericamente inferiore rispetto a piattaforme di gioco tradizionali, le quali possono vantare centinaia di giochi di qualità. Ad oggi, l'acquisto di un visore VR per videogiochi è più da intendersi come un investimento per qualcosa che si avrà in futuro.

- **L'assenza dei grandi nomi:** il mondo dei videogame è formato da numerosissime saghe storiche a cui gli utenti sono molto affezionati, tanto che, al minimo cambiamento nelle strutture di gioco, c'è sempre una parte d'utenza pronta alla critica pesante sui social.

Come ovvio, trasportare classiche saghe su VR comporterebbe cambiamenti drastici alle loro meccaniche e grossi rischi finanziari, visto che i fans storici di queste produzioni si potrebbero sentire "traditi" piuttosto che invogliati all'acquisto, ecco perché ancora sono pochissime le esclusive VR con un nome di spicco.

## 3.5 Il futuro del gaming VR

Su quest'ultimo punto si sta però iniziando a mobilitare il mercato intero, basta guardare le recenti fiere e conferenze sul mondo dei videogame per notare quanto tempo viene dedicato per presentare i titoli esclusivi per VR per potersi rendere conto che questa parte di mercato, oggi ancora poco sfruttata, sarà molto più rilevante in futuro.

Tra gli esempi recenti ricordiamo la famosissima saga horror di Capcom, *Resident Evil*, di cui il settimo capitolo, disponibile da inizio 2017 per *Play Station 4* e *XBox One*, è anche interamente giocabile su *PlayStation VR* oltre che in modalità classica. Per rendere questa cosa possibile, il titolo è passato dalla classica visuale in terza persona che caratterizzava la serie, ad una visuale in prima persona proprio per garantire la totale immersione del giocatore nel mondo di gioco grazie all'utilizzo del visore.



Figura 3.11: Schermata di *Resident Evil 7* in VR.

### 3.5. IL FUTURO DEL GAMING VR

---

Inoltre, all'*E3 2017*, ovvero l'*Electronic Entertainment Expo*, fiera annuale del mondo gaming che si tiene a Los Angeles, sono stati mostrati una versione VR di *Skryim* (Bethesda), titolo di culto della saga *GDR The Elder Scrolls*, uscito originariamente nell'11 novembre 2011, poi rimasterizzato nel corso degli anni per molte piattaforme, come le recenti *Play Station 4* e *XBox One*, che ha venduto oltre venti milioni di copie in tutto il mondo nel corso degli anni.



Figura 3.12: Schermata di *Skryim* giocato con *Play Station VR*.

Altro titolo annunciato nella medesima occasione è *Transference* (Figura 3.13), thriller psicologico di Ubisoft che si pone l'obbiettivo di amalgamare quanto più possibile film e videogames, avvalendosi della collaborazione della casa di produzione cinematografica *Spectre Vision* fondata dal noto attore Elijah Wood, che interpreterà uno dei personaggi di gioco.

L'esperienza che Ubisoft vuole proporre è l'immersione del giocatore in quello che sembra un episodio di una serie TV ad alto budget reso interattivo grazie alla tecnologia della realtà virtuale.

### 3.5. IL FUTURO DEL GAMING VR

---



Figura 3.13: Una delle immagini promozionali di *Transference*, nuovo progetto di Ubisoft per VR.

L'idea di far vivere proprio un thriller, un'esperienza scioccante che nessuno vorrebbe vivere nella realtà, ma che può essere provata per gioco, è una delle basi per cui molti prodotti fruibili su visori sono esperienze a tinte horror, o, in generale, dal forte impatto visivo.

È questo il caso dell'applicazione sviluppata presso KR Games per questa tesi, che, come vedremo, si rifà al classico genere horror.

## Parte II

### Il videogioco "made in Italy"



# Capitolo 4

## Il panorama italiano

Quando si parla del settore videoludico vengono subito alla mente i colossi del settore come Nintendo, Sony e Microsoft, rispettivamente con sede in Giappone e USA, con i loro vari titoli blockbuster capaci di raggiungere milioni di copie vendute.

Alcuni di questi titoli, con il passare del tempo, raggiungono uno status di vero e proprio simbolo del settore come lo è stato e lo è tuttora *Super Mario* di Nintendo, diventato una vera icona nell'immaginario mondiale, tanto che chiunque, anche chi non ha mai preso un gamepad alla mano, sa riconoscere la sua immagine.

Per capire la grande portata di questo settore basti pensare che nel 2017 il suo fatturato ha superato per la prima volta i 100 miliardi di Dollari e, secondo uno studio della società SuperData, questo valore è destinato ad aumentare dell'80% entro il 2020, toccando il valore di 168,8 miliardi di dollari. Sempre secondo l'analisi di questa società, i titoli più profittevoli del 2017 sono stati *League of Legends* (di Riot Games), *FIFA 17* ( di Electronic Arts) e *Honour of Kings* (di Timi Studio); questi titoli, pur appartenendo a generi diversi, hanno in comune il modello di gioco "occidentale" analizzato nel capitolo precedente, infatti propongono una competizione tra giocatori, piuttosto che un'esperienza da vivere.

Anche in Italia il mercato dei videogiochi è in continua espansione: l'AESVI (Associazione Editori Sviluppatori Videogiochi Italiani) ha fornito i risultati di una ricerca effettuata da *GfK* per mostrare lo stato della situazione nel nostro paese; il fatturato del 2017 ha superato il miliardo, e si prospetta un incremento del 7,5% per il 2018. Dall'analisi dei consumatori emerge che il pubblico è sempre più maturo, 6 videogiocatori su 10 hanno tra i 25 e i 65 anni e gli over 65 risultano più numerosi della fascia tra i 14 e 17 anni, con rispettivamente una percentuale di 7,9% e 7,2%.

#### 4.1. LE MIGLIORI PRODUZIONI ITALIANE

---

Quello che non si evince da questi numeri è che oltre ai "big" dell'ambiente, si sono formati molti team indipendenti in tutto il mondo, capaci di sviluppare titoli che propongono storie e meccaniche innovative senza dover rischiare grandi capitali come succede per le case di sviluppo maggiori.

Questi particolari titoli creati e pubblicati senza l'ausilio di un grande investitore, vengono chiamati *Indie*.

In particolare negli ultimi anni il panorama italiano si è sviluppato enormemente, come dimostrano i numeri della settima edizione della *Milan Games Week*, evento all'insegna del videogame che si tiene annualmente a Milano nel mese di settembre, che ha registrato più di 150 mila spettatori e ha mostrato anteprime di titoli attesi a livello mondiale di Sony, Microsoft e Nintendo, oltre alle sempre più promettenti produzioni "made in Italy" con 56 diversi prodotti realizzati da 48 case di produzione nostrane.

### 4.1 Le migliori produzioni italiane

Tra i titoli mostrati a questo evento, alcuni meritano un piccolo approfondimento per le grandi qualità mostrate e per l'essere riusciti ad imporsi anche nel resto del mondo. Tra tutti spicca sicuramente il blockbuster *Mario+Rabbids Kingdom Battle* (Figura 4.1), gioco sviluppato dalla sede lombarda di Ubisoft, **Ubisoft Milan**, per *Nintendo Switch* e pubblicato ad agosto 2017.

Nel gioco vediamo fondersi il mondo di Super Mario (di cui Nintendo ha permesso di utilizzare nomi, personaggi iconici e locations) con quello dei *Rabbids*, personaggi creati dalla stessa Ubisoft di cui esistono già svariati giochi, per creare un particolare crossover che utilizza meccaniche da "strategico a turni", rinnovando la classica formula dei precedenti titoli che vedono Super Mario protagonista.

#### 4.1. LE MIGLIORI PRODUZIONI ITALIANE

---



Figura 4.1: Cover di *Mario+Rabbids Kingdom Battle*.

Questo titolo si è confermato una delle migliori esclusive per *Nintendo Switch*, ottenendo grandi responsi da critica e pubblico come dimostrano i punteggi ricevuti da siti e riviste specializzati nel settore:

*Nintendo Life* - 9/10

*God is a Geek* - 8/10

*IGN.com* - 7.7/10

*Slant Magazine* - 4.5/5

*USgamer* - 4.5/5

*COGconnected* - 88/100

Questo è un caso di estremo interesse perché, per la prima volta, Nintendo ha ceduto il controllo creativo del suo più iconico personaggio ad un team di sviluppo italiano, fidandosi delle capacità dimostrate dai nostri connazionali che non hanno tradito le aspettative degli storici fan dell'idraulico baffuto.

All'annuncio del gioco durante l'*E3*<sup>1</sup> del 2017, a Los Angeles, una delle personalità più influenti di questo campo Shigeru Miyamoto, direttore creativo presso Nintendo e ideatore della serie di *Super Mario* oltre che di molti altri capolavori come *Donkey Kong* e *The Legend of Zelda*, ha ringraziato live dal palco della conferenza il direttore creativo di *Mario+Rabbids* presso lo studio milanese di Ubisoft, Davide Soliani, presente alla conferenza e visibilmente commosso, elogiando il lavoro creativo del team italiano (Figura 4.2).

---

<sup>1</sup>*Electronic Entertainment Expo*, famoso evento annuale legato al mondo del videogioco

#### 4.1. LE MIGLIORI PRODUZIONI ITALIANE

---



(a) La commozione di Davide Soliani, direttore creativo di Ubisoft Milan, per i complimenti ricevuti da Shigeru Miyamoto.



(b) La presentazione di Shigeru Miyamoto e Yves Guillemot per Mario+Rabbids, gioco sviluppato da Ubisoft Milan.

Figura 4.2: Annuncio di *Mario+Rabbids Kingdom Battle* all'E3 del 2017.

Oltre a questo titolo estremamente famoso, abbiamo avuto molti altri esempi di videogiochi nostrani presentati all'ultima *Games Week* tra cui il capolavoro di **Ovosonico** (uno studio di Varese) ovvero il toccante *The last day of June*, un gioco che racconta di come si possa affrontare la perdita di una persona cara e lo fa usando immagini delicate, ma al contempo evocative.

Il gioco, disponibile su PC e *PS4* è "un'avventura grafica" con enigmi ambientali che narra di come Carl, il protagonista, debba rivivere più volte l'ultimo giorno di vita dell'amata cercando di salvarla da un fatale incidente.

A colpire è la direzione artistica che contrappone la tristezza dell'argomento trattato alla scelta dei filtri grafici usati per ottenere un effetto cel-shading dalle vivaci tinte acquarello (Figura 4.3) per sottolineare la dolcezza dei sentimenti di Carl, cambiando lentamente in colori più cupi man mano ci si rende conto che non è possibile cambiare il passato per salvare la sua amata.

Anche il modo in cui sono rappresentate le interazioni tra i personaggi risulta atipico, infatti, pur non parlando mai e non utilizzando l'animazione del volto, questi comunicano tra loro con gesti e suoni tanto buffi quanto esplicativi del loro stato d'animo, con una soluzione utile per evitare l'oneroso compito dell'animazione facciale e del doppiaggio, ma al contempo coerente con l'estetica proposta.

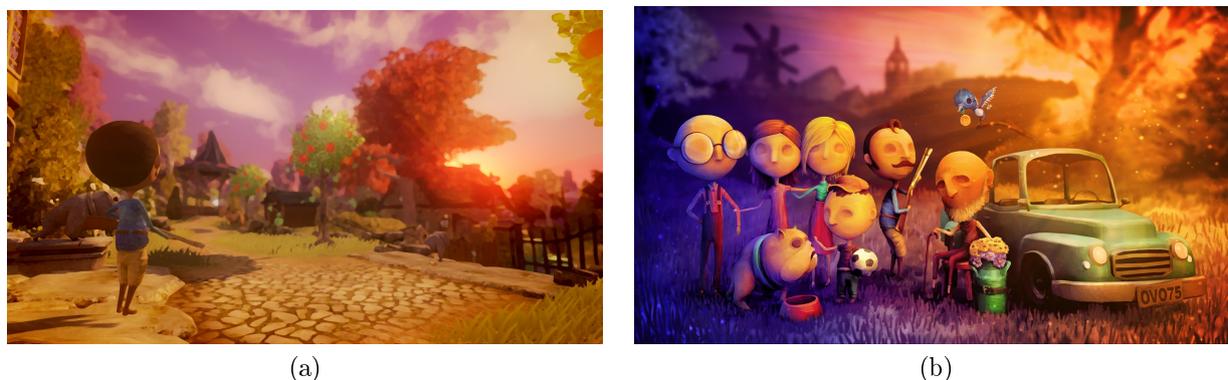


Figura 4.3: Gli scenari a tinte acquarello e i personaggi di *The Last Dat of June*, avventura grafica sviluppata a Varese da Ovosonico.

Altra importante realtà del videogioco italiano è rappresentata da **Milestone**, casa di sviluppo di videogiochi con sede a Milano, da sempre fortemente legata al mondo dei motori; è infatti detiene le licenze per lo sviluppo e la pubblicazione dei maggiori titoli dell'ambiente sportivo motoristico, tra cui *Moto GP*, *Supercross*, *WRC* e *Superbike* oltre ad aver ottenuto i diritti per pubblicare giochi legati alle carriere di grandi personalità dello sport come Valentino Rossi e Sébastien Loeb.

La particolarità di Milestone è proprio l'essersi specializzata in questa nicchia del mercato fin dalla sua fondazione, nel 1993, e di aver avuto modo di perfezionare tutti gli ambiti della simulazione sportiva, in particolar modo nella fedeltà dei modelli 3D proposti, che rispecchiano con grande cura le controparti reali.

Potendo sfruttare queste licenze, Milestone è uno dei marchi italiani in ambito videogames più famoso all'estero.

## 4.2 Sviluppatori Indie in Italia

Non solo al *Games Week*, ma anche alle altre fiere italiane dedicate al fumetto e al mondo dell'entertainment come il *Lucca Comics* o il *Torino Comics*, il mondo dei videogames riesce a ritagliarsi uno spazio sempre più importante, forte del fatto che, in Italia, hanno iniziato a nascere accademie dedicate a questo mezzo come *Vigamus Academy* (Roma) o *Digital Bros Game Academy* (Milano).

Queste offrono corsi mirati ai vari ambiti dello sviluppo di un videogame: dal *level design*, alla modellazione passando per il *game design*, alternando sessioni pratiche allo studio.

L'apertura di queste accademie è sintomo di quanto il settore si stia espandendo anche nel nostro paese e si rende necessaria la presenza di figure professionali specifiche per questo ambito lavorativo che necessita di conoscenze approfondite in vari diversi rami.

### 4.3. KR GAMES

---

In questa situazione di fermento, traggono vantaggio tutti quei piccoli team di sviluppo che, non avendo a disposizione grandi somme di denaro per promuovere progetti eccessivamente complessi, possono sperimentare con la tecnologia VR per la quale, dato che la sua espansione è avvenuta solo di recente, è facile trovare generi di gioco non ancora presenti a cui potersi dedicare, andando così a muoversi su un terreno con minore concorrenza.

Altro elemento che giova al successo di questi team è sicuramente il fattore novità di questa tecnologia, che può spingere anche utenti non esperti in ambito videoludico (*casual gamers*) a provare questi nuovi videogame che risulterebbero, in alcuni casi, meno appetibili ad un pubblico più smaliziato.

Come il passato ci insegna, con la nascita di un nuovo medium, le prime applicazioni per quest' ultimo andranno sempre a far leva sulle emozioni umane più semplici da suscitare come, ad esempio, la paura, come già successo per il cinema, per sopperire alla scarsa esperienza con il mezzo sfruttando i "punti deboli" dell'animo umano.

Ecco perché una gran parte dei giochi VR attuali sono iterazioni del genere horror: dai "*survival*" ai "*walking simulator*" alle "*adventure action*".

### 4.3 KR Games

Proprio sfruttando i fattori appena analizzati è potuta nascere e crescere **KR Games**, un giovane studio Indie con sede a Vercelli, che crede fermamente nella VR come forma di intrattenimento, e non solo, e la supporta con costanza con varie sue creazioni.



Figura 4.4: Logo ufficiale di KR Games.

Ad oggi KR Games conta cinque diversi videogiochi all'attivo, di questi, quattro sono esperienze dedicate alla realtà virtuale, ed in particolare, tre di queste sono proprio *walking simulator* a tema horror. La scelta di puntare su questo genere per nascondere alcune piccole mancanze sulla realizzazione tecnica e sul game design, dovute soprattutto alle dimensioni modeste del budget a disposizione, si è rivelata vincente: KR Games è tuttora in attivo, e il suo *The hospital: Allison's diary*, è stato inserito da più siti nella classifica delle migliori app per *Gear VR*.

### 4.3.1 Intervista a Kabir Ferro e Roberto Comella

Per sviluppare questa tesi, e meglio comprendere il percorso di sviluppo di un'applicazione VR, ho collaborato per un periodo di sei mesi presso quest'azienda, lavorando a stretto contatto con i suoi due membri fondatori, **Kabir Ferro** e **Roberto Comella**, che ho sottoposto ad una breve intervista per capire come venga percepita la VR da chi lavora nel settore.

**Domanda:** Cosa rappresenta per voi KR Games?

**K. Ferro:** Vorrei fare una premessa: per noi questa è una messa in gioco, abbiamo iniziato per gioco, per provare, e ci siamo buttati. Lui (Roberto Comella) in quel periodo, quando abbiamo iniziato, aveva importanti colloqui in corso; io avevo un lavoro a tempo indeterminato, quindi abbiamo entrambi lasciato una carriera esterna per buttarci in questo progetto insieme credendo nella tecnologia. Capisco bene la diffidenza verso la realtà virtuale, perché lo eravamo anche noi, ma provandola ci ha molto colpiti e abbiamo fatto un piccolo investimento rispetto altre aziende [...] ad oggi siamo avanti a molti competitors, siamo in attivo. KR Games è qualcosa in cui crediamo, vogliamo fare giochi per realtà virtuale e magari in un futuro espanderci sul mercato più tradizionale.

**R. Comella:** Sì, l'idea è nata due anni fa quando in America ho provato il visore, una volta tornato l'ho comprato e mi son detto "Qui si può far qualcosa di interessante" e l'ho fatto provare a lui (Kabir Ferro).

**K. Ferro:** Sì, me l'ha fatto provare e mi ha proposto quest'idea di metterci in proprio e ci ho creduto anch'io.

**R. Comella:** Ad oggi KR Games è un'azienda di software in rampa di lancio, con molti progetti per il futuro.

**Domanda:** Perché avete deciso di lanciarvi proprio nell'industria del gaming VR?

**R. Comella:** Io sono sempre stato un videogiocatore, quindi la cosa mi ha sempre appassionato e ho studiato informatica anche per questo. Quando a suo tempo

provai il visore, non lo feci con l'idea di dover poi sviluppare qualcosa, ma proprio per poterci giocare, per vedere cos'era. Si è scelto il gaming anche perché oggi, nell'informatica, è un settore di massa che può generare rendita più facilmente di altri settori che sono già coperti.

**K. Ferro:** Sì, se ad esempio avessimo scelto di creare un nostro social network, oggi forse non saremmo qui a parlarne, non tanto per questione di complessità o di tempi, ma proprio perché è difficile contrastare la concorrenza. Qui, nel mondo VR, c'è più spazio anche per le nuove piccole imprese, è ancora un settore poco battuto.

**R. Comella:** In più l'utente medio di giochi ne prova tanti, ad esempio il social, tranne casi particolari, quando ne hai due o tre è già tanto.

**K. Ferro:** Probabilmente, se ti metti a pensarci, sarebbe anche fattibile avere un'idea migliore di Facebook, che ha comunque diversi problemi [...] ma il difficile non è tanto il realizzarlo, quanto il promuoverlo: cambiare l'idea che il ragazzo medio ha e fargli capire che può trovare un social migliore. Nel gaming(VR) invece, essendo un mercato in espansione proprio in questi anni, abbiamo molta più visibilità; basti vedere che su Gear VR siamo in "Top Selling".

**Domanda:** L'Italia è un paese florido per questo particolare mercato?

*Si guardano tra loro ridendo.*

**R. Comella:** Qualche download arriva anche dall'Italia, ma se dovessimo vivere solo grazie a quelli non si riuscirebbe a resistere.

**K. Ferro:** È per questo che abbiamo deciso di lasciare tutti i testi e dialoghi sempre e solo in Inglese, per poter raggiungere il maggior numero di utenti possibile.

**R. Comella:** Fai conto che in questo ambiente che è ancora agli inizi sopravvivi a malapena con il mercato mondiale. I maggiori download arrivano da USA, Brasile, Germania e Gran Bretagna. Uno si aspetterebbe di avere un bon riscontro in Cina vista la propensione al videogioco, ma loro hanno i loro mercati particolari, ad esempio ora HTC vuole fare un visore pensato solo per il mercato asiatico per dire.

**Domanda:** Come nasce e cresce un progetto presso KR Games?

**R. Comella:** Essendo per ora piccoli progetti e il numero di persone che ci lavora anche, non c'è un iter particolare; si parla, si fa brainstorming assieme e quando

si trova l'idea che piace e che può funzionare mettiamo giù un piccolo planning su carta.

**K. Ferro:** Ecco, non so se ricordi, qui c'è un primo schizzo della Villa, un po' della planimetria. Poi in realtà non abbiamo molto, essendo in pochi farebbe ridere; usiamo più che altro il blocco note come "To Do". Tra l'altro, essendo prodotti nostri, non abbiamo altri col fiato sul collo per la pubblicazione, cerchiamo di gestirci per seguire l'andamento di mercato facendo uscire i giochi in un dato periodo o con il lancio di un dispositivo; per dire, *The Hospital* (altro titolo horror) è uscito nel periodo di Halloween per spingerlo maggiormente; *The Cathedral* l'abbiamo pubblicato nel periodo di lancio dell'S8 per sfruttare un po' l'effetto calamita del nuovo dispositivo.

**R. Comella:** Lo sviluppo in sé del gioco segue le linee guida standard: fai la modellazione e inizi a creare il *framework* del gioco; lasciamo l'audio alla fine per abbinarlo meglio alle animazioni, come tempi.

**K. Ferro:** Sì, l'audio è molto importante per questa tipologia di esperienze horror, per questo dobbiamo aggiungerlo quando abbiamo l'animazione finale, per avere sincro perfetto. Poi quando escono bug o problemi in revisione è normale dover fare qualche modifica all'ultimo; ad esempio facendo testare *The Villa*, la scala che c'è per andare al secondo piano non la vedeva nessuno, nessuno capiva che doveva salire di lì e abbiamo cercato nel tempo di ottimizzare la cosa al massimo per far capire al giocatore che doveva andare lì. Molte cose le vedi solo durante il gameplay.

**R. Comella:** Anche la storia della trilogia di Allison si è sviluppata man mano; ai tempi del primo capitolo non sapevamo ancora come sarebbe poi evoluta la trama. Ma probabilmente faremo qualcosa con un mondo più scritto e vincolante per un nostro nuovo progetto del prossimo anno.

**Domanda:** C'è un motivo particolare per cui state dando molta importanza al genere horror?

**R. Comella:** Sì, risposta secca, perché è molto efficace con la VR, ci sono moltissime esperienze horror infatti, ultimamente, è arrivata un po' di concorrenza, ce n'erano molti meno l'anno scorso quando abbiamo iniziato *The hospital*: al tempo ce n'erano cinque o sei, ora saranno attorno alle venti. Spaventano molto, fanno un buon effetto, perché comunque lo spavento non è qualcosa che tu vuoi vivere nella realtà, quindi lo vivi in VR.

**Domanda:** Nello sviluppo dei vostri giochi avete qualche influenza particolare?

- K. Ferro:** Sicuramente i competitors sono un punto di riferimento, ovviamente i migliori. Devi prendere spunto da quello che fanno, migliorarti e superarli. Ad esempio *Affected* per VR è stato un nostro punto di riferimento.
- R. Comella:** Sì, o almeno lo era, visto che lo abbiamo superato con i nostri nuovi giochi per qualità grafica e audio.
- K. Ferro:** Prendiamo comunque molta ispirazione dai film, soprattutto per l'audio, perché i film, non avendo l'immersione che hai in VR, giocano molto sull'audio che incide tanto emotivamente: ti può mettere tensione, tristezza o paura se usato bene. Poi noi tendiamo a recuperare reference o trarre ispirazione da luoghi veri, ad esempio per *The Hospital* ci siamo ispirati ad un ospedale che c'è stato veramente a Vercelli, ormai è abbandonato, ma da ragazzino ci son stato con gli amici, così per gioco. O anche *The Villa* trae spunto da una vera Villa del Bergamasco, come location.
- Domanda:** Si può considerare il videogame, che sia in VR o tradizionale, come forma d'arte?
- R. Comella:** Assolutamente sì, nel videogame conta molto la creatività, e dove c'è creatività si può parlare di arte. Conta molto anche la tecnica, ma questo vale anche per le forme d'arte più classiche.
- K. Ferro:** Anche perché di solito l'arte più sperimentale nasce dopo che si hanno delle basi solide. Comunque sì, il videogioco non solo può essere arte, ma è fatto da una somma di arti diverse: visive per le grafiche, poi audio, letterarie se devi scrivere una trama complessa, doppiaggio per le produzioni importanti; è un'unione di tante arti diverse. Magari in generale non siamo il tipo di artisti che vogliono far passare un messaggio profondo, ma vogliamo farti vivere un'esperienza; quando sviluppiamo un gioco, noi pensiamo all'utente che ci giocherà. In fondo una forma d'arte è qualcosa che ti passa delle emozioni, e noi lo facciamo.
- R. Comella:** Una società di videogiochi è un'azienda e punta comunque a produrre reddito; ammetto che non facciamo arte solo per amore dell'arte, ecco.
- Domanda:** Come vedete il rapporto tra gaming tradizionale e gaming in VR? Cosa resta uguale a se stesso e quali sono le differenze?
- R. Comella:** Cambia sicuramente il livello di immersione, ma l'utenza può essere la stessa: abbiamo provato di recente un fps per *Rift*, una cosa molto classica e ben fatta che può piacere molto a chi già gioca su console o pc, bisogna solo convincere questi giocatori classici a mettersi il visore, ma se la tecnologia

si conferma, si impone bene, allora riesci a convincerlo. Ad esempio negli ultimi anni gli schermi son diventati sempre più grandi, sempre più ad alta definizione eccetera, adesso siamo saturi, non si può avere uno schermo più grande di una parete; il passo successivo è quello di metterti dentro al gioco, di farti vivere il gioco. Per me l'utenza può essere la stessa, devi solo convincerla che la tecnologia può funzionare, che può essere migliore.

**K. Ferro:** Bisogna aumentare il bacino d'utenza perché attualmente, la VR, è più incentrata sulle piccole esperienze, come quelle che proponiamo noi; compri una di queste esperienze a poco, fai che di media sono sui 5€, la provi e poi al massimo la fai provare agli amici, non è che la ripeti tutti i giorni.

**R. Comella:** È come andare al luna Park, non è che ci vai tutti i giorni: vai una volta ogni tanto, provi la giostra e torni a casa, la stessa cosa è per la VR adesso, provi l'esperienza che hai scelto e in un paio d'ore o anche meno, l'hai finita, ma perché alle giostre, al cinema eccetera vanno tutti e pochi usano la VR? Eh, bisogna far capire le sue potenzialità.

**Domanda:** Perché la VR non è ancora un fenomeno di massa? Cosa serve perché lo diventi?

**R. Comella:** Non è di massa per tre motivi: costi, accessibilità e facilità.

**K. Ferro:** In più c'è sempre un po' di diffidenza non solo sulla tecnologia, ma a volte anche sull'etica: può fare paura vedere un mondo diverso in cui entrare e staccarsi dal mondo reale. Anche noi che ci lavoriamo se dovessimo immaginare un mondo futuro in cui i nostri figli invece di uscire giocare con gli amici, stanno tutto il giorno con un ipotetico visore VR fa un po' paura. Tutto va usato col giusto peso.

**R. Comella:** Sì, per me l'utilizzo ideale del *Gear VR*. almeno per adesso, è magari la domenica mattina, invece di guardarti la replica di un film ti vivi un'oretta e mezza di esperienza VR a due o tre euro.

**K. Ferro:** Ad esempio guardando al mercato americano che è molto più attivo, vedi su *YouTube* molti video di gente che si trova con gli amici e giocano alla VR passandosi il visore, lo vedono non come un qualcosa che ti isola o ti allontana dalla realtà, ma come un qualcosa da vivere assieme.

**R. Comella:** Sì, una sorta di gioco di società moderno, ecco.

**K. Ferro:** Ecco, un altro fattore negativo è la poca promozione; ad esempio se entri in un qualunque negozio di elettronica puoi provare la *Playstation* o *Xbox* con il nuovo gioco del momento, e poi se vuoi la compri, ma la VR se vuoi

### 4.3. KR GAMES

---

provarla devi avere un amico che ce l'abbia, è raro poterla provare ai negozi, almeno in Italia.

**Domanda:** È notizia di qualche settimana fa che si stia pensando di inserire il videogame competitivo come disciplina sportiva alle Olimpiadi. Non c'è il rischio che si tenda a dimenticare la parte narrativa del videogioco? Voi di KR Games cosa ne pensate di questa diatriba?

**K. Ferro:** Io per sport vedo un qualcosa legato al movimento fisico. è vero che qui il cervello lavora di più rispetto ad altre attività, però per me lo sport resta legato alla competizione fisica. tra due atleti che corrono, vince il più veloce, quello che si è allenato di più, ma è vero anche che l'allenamento vale anche per i videogame. . . Dipende dalla definizione di sport, perché in qualunque cosa chi si allena di più diventa più bravo, ma non tutto può essere sport.

**R. Comella:** Gli scacchi sono considerati sport dal comitato olimpico internazionale, quindi, non vedo perché il videogioco non possa esserlo dato che sono un tipo di competizione vagamente simile, anzi, il gioco aggiunge alla parte strategica anche il dover usare i riflessi per essere più veloci degli avversari.

**Domanda:** Quali sono le sfide più impegnative nel realizzare un gioco in VR?

**R. Comella:** Probabilmente è strutturare il progetto, mettersi d'accordo su come iniziare insomma. Poi di solito il resto viene da sé procedendo.

**K. Ferro:** Ci è capitato spesso di perdere molto tempo ad organizzare un planning, preparare un sacco di cose che sembrano non finire mai e poi il lavoro riusciamo comunque a concluderlo settimane prima della data prevista. Quando parti a lavorare hai tutto molto più chiaro e riesci a semplificare cose che all'inizio in fase di progettazione ti sembrano difficilissime.

**R. Comella:** Esatto, il difficile è proprio iniziare e pianificare bene i tempi. Una volta che sei partito riesci a capire come gestire tutto.

**Domanda:** Come vedete il futuro della VR? Riuscirà a integrarsi meglio al gaming tradizionale rispetto ad ora o sarà solo una moda di passaggio?

**R. Comella:** Eh, ad oggi non possiamo saperlo; noi lo speriamo. Se i costi diventano accessibili può diventare il futuro; cioè, se a uno piacciono gli sparatutto e si mette a provare uno sparatutto in VR con tutta l'immersione che ne deriva, secondo me non si stanca tanto presto.

### 4.3. KR GAMES

---

- K. Ferro:** Bene o male tutti quelli che adesso provano la VR per la prima volta rimangono soddisfatti.
- R. Comella:** Esatto, e come è piaciuto a te che sei un videogiocatore vecchia scuola, se la provassero la VR piacerebbe anche a tanti altri, anche perché, fondamentalmente, chi oggi gioca in modo hardcore già si isola con cuffie da gaming, stanza apposita per giocare eccetera, col VR non cambia molto, anzi migliora, perché se ti giri non vedi la porta della stanza, ma sei ancora dentro al gioco. Quindi potenzialmente per chi ama giocare seriamente può diventare una tecnologia interessante.
- K. Ferro:** Ecco, sostituire il gaming tradizionale magari no, ma integrarsi può farlo sicuramente.
- R. Comella:** Devono però scendere i costi, quello sì.



## Parte III

# Lo sviluppo di un videogioco in VR presso KR Games



# Capitolo 5

## Un'introduzione alla modellazione 3D

Partiamo definendo una grande distinzione tra due categorie di modellazione date da cosa andiamo a ricreare: la modellazione *organica* e quella *geometrica*.

La prima si riferisce alla modellazione di esseri umani od umanoidi, animali e creature di vario tipo oltre a vegetazione e tutti quegli elementi del territorio definibili come "naturali".

Questi sono infatti caratterizzati da un alto numero di dettagli e irregolarità dovuti al fatto di non essere formati da geometrie nette come lo sono gli oggetti più regolari creati dall'uomo; per questo motivo la modellazione organica necessita di un elevato numero di poligoni per simulare le curve morbide e le irregolarità di cui sono composti gli elementi che vogliamo ricreare (Figura 5.1).

Al contrario la modellazione geometrica è quella in cui ci si trova a creare in 3D oggetti artificiali, siano essi case, armi, oggetti di uso comune, arredo, elettrodomestici e così via. Per questi oggetti si avrà una complessità del modello molto inferiore per via del minor numero di facce e dettagli, ma il fatto di dover rappresentare degli oggetti artificiali porta questi ad aver bisogno di grande attenzione e precisione sulle corrispondenze di vertici, parallelismi e angoli tra le singole facce, in quanto un'imperfezione, ad esempio, sul tronco di un albero può essere funzionale al realismo del modello, ma ottiene un effetto opposto per il modello di un oggetto creato dall'uomo.

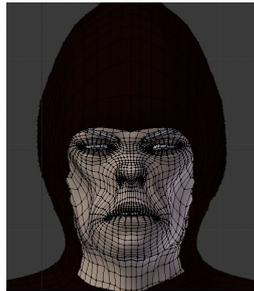
Vi sono ovviamente casi di oggetti che possono essere ricondotti in una zona intermedia tra i due tipi di modellazione, basti pensare ai capi d'abbigliamento che, pur creati dall'uomo, risultano più consoni ad essere definiti come modellazione organica in quanto seguono le forme morbide del corpo umano tranne particolari parti come cinture, fibbie o bottoni (Figura 5.2, pagina 77).



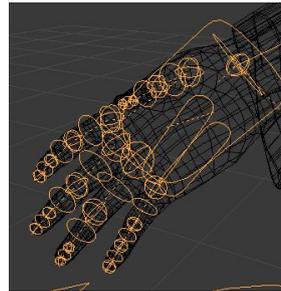
(a)



(b)

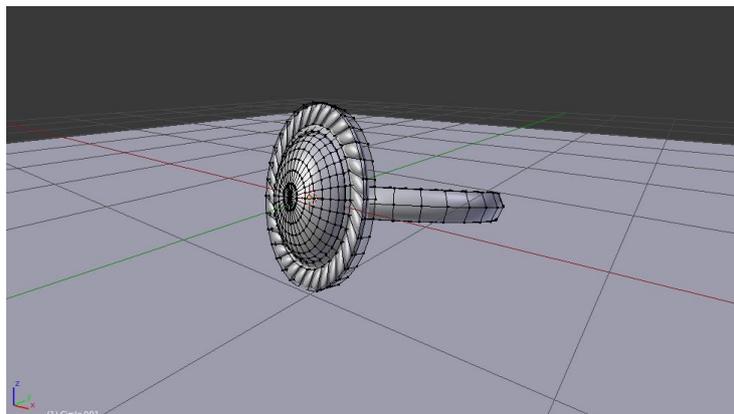


(c)



(d)

Figura 5.1: Immagini di Guglielmo, un modello utilizzato per *The Cathedral: Allison's Diary*, esempio di modellazione 3D organica creato su *Blender*.



(a)



(b)

Figura 5.2: Il modello di un anello creato su *Blender*, esempio di modellazione geometrica per oggetti di uso comune.

Dividiamo poi la modellazione 3D in 3 macro categorie in base al modo in cui questa viene effettuata e non a cosa andiamo a rappresentare: abbiamo la **modellazione procedurale**, quella **tramite dati provenienti da modelli reali** e la **modellazione manuale** per la quale ci sarà un approfondimento dedicato.

La prima sfrutta un insieme di algoritmi per creare una struttura complessa partendo da un singolo blocco fondamentale che, una volta ripetuto, con eventuali cambi di parametri inseribili a posteriori come ampiezza di angoli o dimensioni, crea la struttura finale evitando di dover modellare l'intero oggetto in ogni sua singola parte.

Questa tecnica si divide in varie sottocategorie a seconda di cosa l'algoritmo è progettato per creare: abbiamo simulatori fluidodinamici, di tessuti, generatori di vegetazione, capelli o di frattali.



Figura 5.3: Alcuni dei parametri modificabili per l'algoritmo di generazione procedurale di vegetazione in *Blender*.

Su *Blender* è ad esempio possibile utilizzare un algoritmo per la modellazione procedurale della vegetazione, scaricabile tramite *add-on*, chiamato *Sapling Tree Generator*; utilizzandolo è possibile creare modelli di piante e arbusti semplicemente assegnando valori ai parametri presenti in un'apposita tabella (Figura 5.3): numero dei rami, presenza o meno delle foglie, suddivisioni e spaccature nel tronco, dimensione del fusto ed altri valori. Una volta ottenuto un risultato convincente basta confermare i parametri inseriti e verrà generato il modello 3D desiderato.

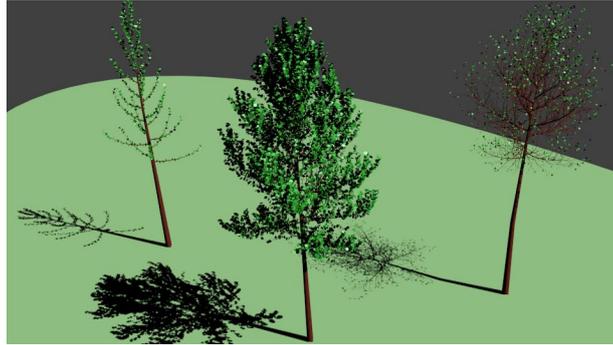


Figura 5.4: Alcuni possibili risultati della generazione procedurale di vegetazione in *Blender*.

La seconda tipologia, la modellazione per acquisizione, punta a realizzare modelli 3D servendosi di dati relativi all'oggetto reale ottenibili attraverso vari metodi che sfruttano diverse tecnologie.

Tra le principali tecniche di acquisizione ricordiamo:

**Acquisizione per Fotogrammetria:** i dati vengono ottenuti tramite varie fotografie dell'oggetto reale scattate da punti diversi, a volte con l'ausilio di marker, poi sarà compito del software costruire un modello 3D; questa tecnica fornisce risultati a bassa fedeltà, ma è anche piuttosto economica e semplice da realizzare.

**Acquisizione per sonda 3D a contatto:** una tecnica che si basa sull'utilizzo di un braccio meccanico che muove una sonda lungo la superficie dell'oggetto di modo che il software possa copiare le corrispondenze di questi punti nello spazio 3D; tecnica ideale per oggetti rigidi di piccole dimensioni, ma necessita di strumentazione hardware apposita.

**Acquisizione per scansione laser:** questa tecnica sfrutta apparecchi di dimensione variabile che proiettano laser lungo la superficie dell'oggetto da scansionare, il software andrà a creare nuvole di punti nello spazio 3D seguendo le corrispondenze con la superficie ottenuta con il laser, ottenendo un livello di dettaglio elevato anche per oggetti flessibili e morbidi. Anche in questo caso, il maggior problema risulta essere la necessità di avere il costoso apparecchio necessario per la scansione.

Infine abbiamo la modellazione manuale che indica tutta la modellazione che avviene per intero, in tutte le sue parti, eseguita dal modellatore senza sfruttare algoritmi per la duplicazione in massa.

Ora analizzeremo in dettaglio le varie tecniche ad essa riconducibili, in quanto sono state quelle più usate durante il lavoro di modellazione presso KR Games.

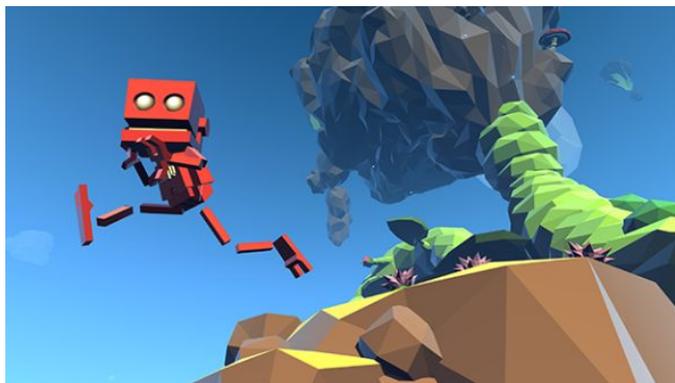
### 5.1 Modellazione 3D manuale

Innanzitutto ricordiamo che la modellazione poligonale lavora su superfici divise in una o più parti, dette facce poligonali il cui insieme va ad approssimare l'oggetto desiderato aumentando di realismo man mano che si aumenta la suddivisione in facce; avendo ipoteticamente a disposizione una macchina capace di suddividere infinitamente in facce una superficie, si potrebbe creare un oggetto 3D identico all'originale.

A seconda della fedeltà rispetto all'oggetto reale il modello sarà definito *low poly*, se composto da pochi poligoni, o *high poly* altrimenti; ma attenzione, *low poly* non è sinonimo di scarsa qualità o precisione, spesso è una scelta stilistica adottata con sapienza dei creatori di contenuti per dare uno stile volutamente *cartoon* alla propria opera. Sono infatti molti i videogiochi dallo stile visivo minimalista che nascondono trame e scelte di gameplay molto ricercate.



(a) Fotogramma da *Uncharted 4: La Fine di un Ladro (Naughty Dog)*.



(b) Fotogramma da *Grow Home (Ubisoft)*.

Figura 5.5: La differenza tra modellazione *High Poly*, (a), e *Low Poly*, (b).

Nelle prossime pagine si andranno a spiegare brevemente le più importanti tecniche di modellazione manuale, focalizzandoci su quelle usate a KR Games.

### 5.1.1 Modellazione "per spostamento di elementi"

Ogni primitiva geometrica disponibile sui vari software per la modellazione è composta da tre elementi fondamentali: **vertici**, **lati** e **facce**.

Uno o più di questi elementi può essere modificato con le tre azioni di base della modellazione: **scalamento**, **rotazione** e **spostamento** (tranne il caso del vertice che, ovviamente, essendo "nullo" a livello di spazio occupato, non ha senso di essere ruotato o scalato).

Si procede quindi a lavorare su gruppi di queste entità fondamentali combinando le tre trasformazioni di base; questa tecnica consente di ottenere velocemente risultati poco complessi, ma questi sono vincolati dalla suddivisione della primitiva originale.

Pur non potendo garantire risultati ad alto impatto visivo, viene usata perché molto semplice da utilizzare.

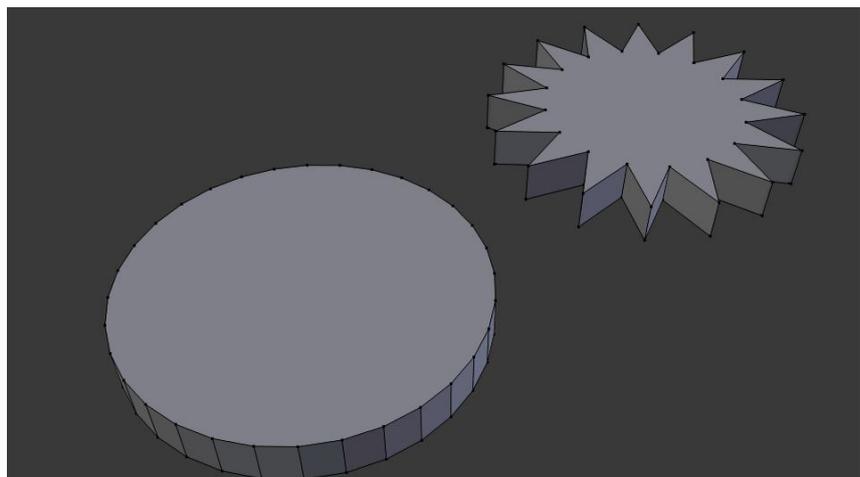


Figura 5.6: Semplice esempio di modellazione "per spostamento di elementi" su *Blender*.

### 5.1.2 Metodo della "texture piana"

Si può creare ogni singolo poligono della texture e sistemarlo su di un piano creando il profilo in 2D del modello 3D che voglio ottenere; questo si può ottenere creando una superficie piatta e suddividendola più volte.

Arrivati a questo punto si possono spostare i vertici centrali oppure utilizzare vari metodi di estrusione per ottenere la tridimensionalità dell'oggetto.

Anche in questo caso la suddivisione di partenza influenza molto la qualità del prodotto finale.

Questa tecnica risulta buona per oggetti semplici o con molte simmetrie, ma è inadatta per sviluppare modelli complessi o irregolari, poiché la superficie di partenza dovrebbe avere troppe suddivisioni per essere gestita correttamente.

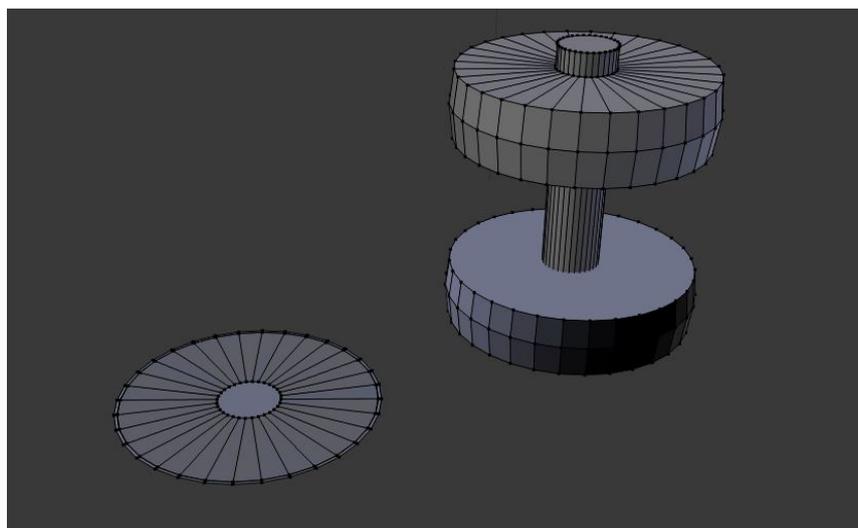


Figura 5.7: Semplice esempio di modellazione con il metodo della "texture piana" su *Blender*.

### 5.1.3 Metodo "a tela di ragno"

In questo caso si parte creando le facce di una zona centrale del modello e si va all'esterno estrudendo "a cerchio" e andando a modellare ogni singolo poligono, spostando vertici e lati a seconda della forma che vogliamo dare.

Questa tecnica, pur essendo molto precisa se usata nel modo corretto, risulta molto dispendiosa in termini di tempo ed è quindi usata solo per modelli che richiedono un alto grado di precisione.

È molto utile per la modellazione del viso dei personaggi, in particolare partendo dal contorno occhi. Con questa tecnica si può adattare il modello in corso d'opera, in quanto non si dipende da una scelta di suddivisioni iniziale, ma il fatto di dover partire da zero, creando di volta in volta ogni poligono come se si stesse disegnando il personaggio, la rende una tecnica lenta da utilizzare.

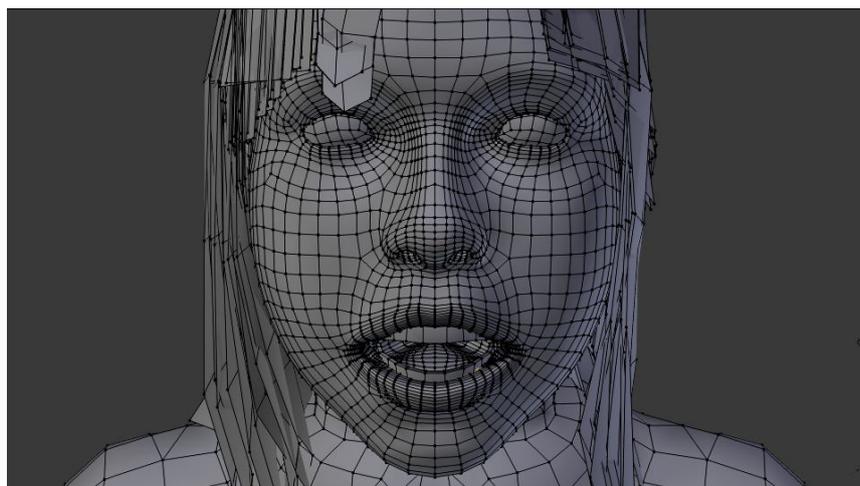


Figura 5.8: Esempio di modellazione del viso con la tecnica "a tela di ragno" su *Blender*.

### 5.1.4 Metodo "a rifinitura progressiva"

Si può creare una texture a bassa definizione con il metodo desiderato facendo attenzione a tenere basso il numero di poligoni.

Fatto questo si va a generare un certo quantitativo di poligoni aggiuntivi per creare quei dettagli mancanti dalla mesh *low poly*, partendo da quelli più visibili e successivamente scendendo nel particolare aggiungendo sempre più divisioni.

Bisogna ricordare che tanto è minore il numero di poligoni, tanto è più facile modificare proporzioni e morfologia generale della texture, ergo la fase iniziale di creazione della texture grezza richiede un'attenta pianificazione in modo da potersi dedicare ai successivi dettagli senza dover usare metodi complessi per ritoccare la forma originale.

Una buona pratica per evitare eventuali errori è salvare la texture ad ogni passo di definizione aggiuntiva, in modo da avere la possibilità di tornare comodamente ad una situazione precedente.

Questa metodologia (Figura 5.9) di modellazione molto evoluta è oggi considerata uno dei massimi paradigmi della modellazione 3D.

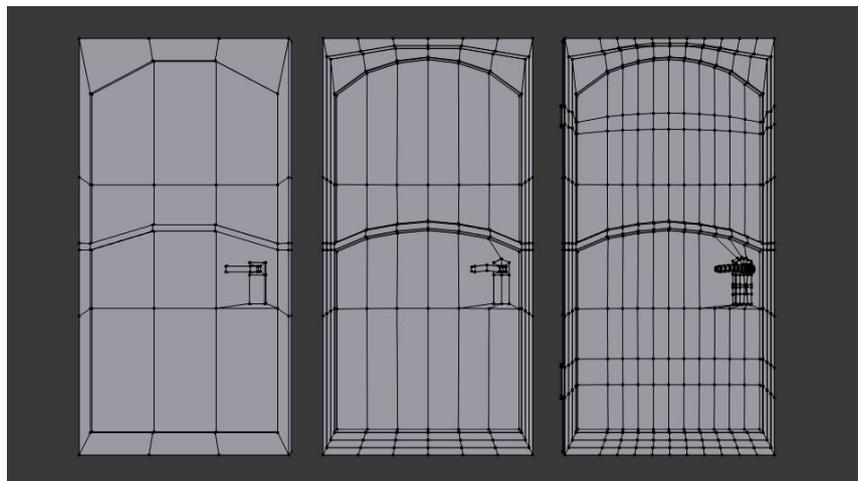
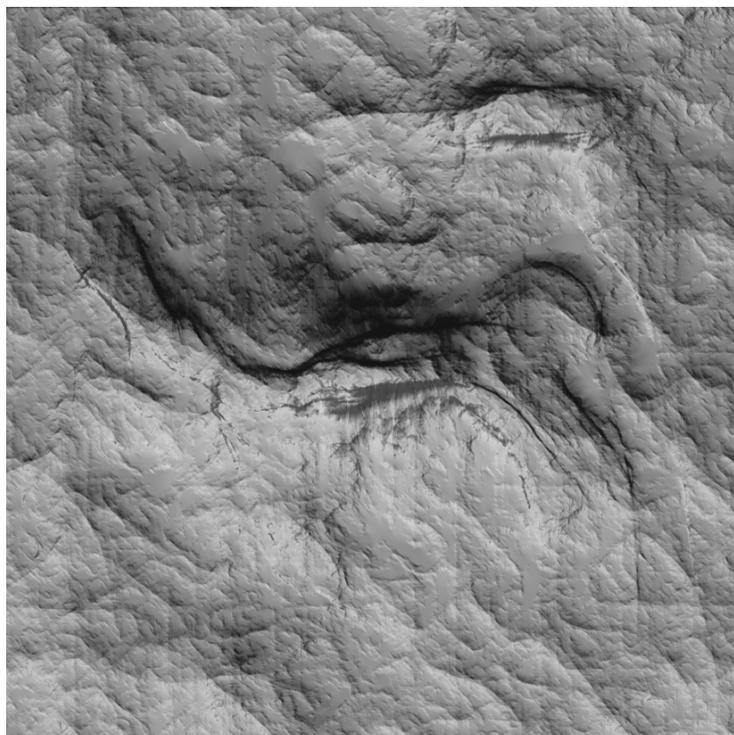


Figura 5.9: Esempio di "modellazione progressiva" in *Blender*.

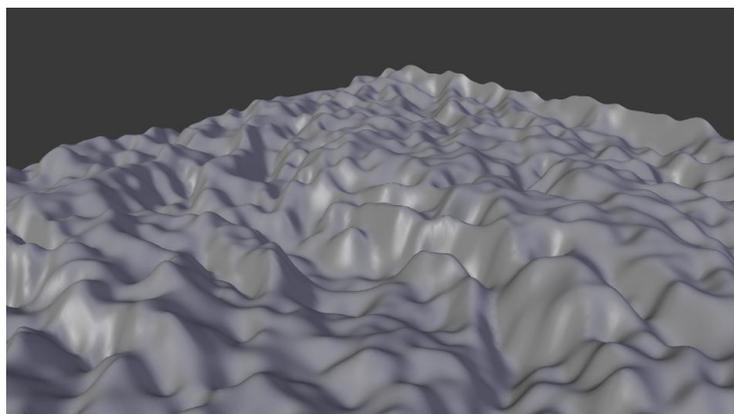
### 5.1.5 Metodo "per *displacement map*"

Questa tecnica differisce da quelle elencate finora poiché non si va a lavorare direttamente sulla texture, ma si utilizzano immagini in scale di grigio che indicano la profondità. Si crea un piano con una suddivisione in facce tanto più fitta quanto più si desidera un risultato qualitativamente alto; la mappatura a *displacement* va ad agire sulla texture spostando lungo la normale positiva i vertici che si trovano in corrispondenza di zone chiare dell'immagine, lungo la normale negativa altrimenti a seconda dell'intensità del colore. Dividiamo due diverse tipologie di questa tecnica: il ***displacement geometrico*** che va ad agire sui punti già esistenti della mesh iniziale e che quindi ne necessita molti per un risultato di qualità, e il ***displacement per micropoligoni*** che funziona allo stesso modo, ma va ad aggiungere, laddove servono, suddivisioni in facce triangolari aggiuntive arrivando all'ordine dei milioni, ma questa suddivisione viene applicata solo in fase di rendering, lasciando inalterata la geometria iniziale durante tutta la fase di lavoro.

Queste particolari tecniche sono solitamente usate per la modellazione di territori estesi piuttosto che di piccoli oggetti realizzabili a mano (Figura 5.10).



(a)



(b)

Figura 5.10: Una "*displacement map*", (a), ed il risultato dell'applicazione della "*displacement map*", (b), su di un piano suddiviso in molteplici facce tramite il modificatore *Subdivision Surface*, per creare una zona montuosa in *Blender*.

### 5.1.6 Metodo "per scultura 3D"

Questo metodo ricorda le tecniche di scultura tradizionale, infatti utilizza una serie di diversi scalpelli virtuali ognuno con dimensioni e proprietà differenti per modificare la geometria di una texture in tempo reale creando solchi, incisioni e avvallamenti come se la texture modificata fosse un pezzo di argilla.

Questa tecnica è soprattutto usata per modellazione organica, specialmente per rifiniture ai modelli di personaggi per ottenere risultati ad alta definizione, ma poiché questo metodo può generare anche molti milioni di poligoni, si tende a trasferire i dati a *displacement map* per poterli utilizzare in modelli più leggeri.

Infatti, anche utilizzando lo *sculpting* su modelli con un basso numero di poligoni, questo va a creare autonomamente ulteriori suddivisioni in base al livello di dettaglio da noi creato.

## 5.2 L'importanza della funzione del modello 3D

È importante capire che non sempre un modello 3D qualitativamente superiore a livello di dettaglio sia necessariamente il modello corretto da utilizzare; a seconda dell'utilizzo che se ne dovrà fare, bisogna creare un modello con determinate tecniche e criteri.

Ad esempio un modello 3D di un orologio a muro formato da decine di migliaia di poligoni per andare a creare i dettagli del decoro e delle intarsiature e texture ad alta definizione dedicate al quadrante dell'orologio, seppur eccezionalmente bello da vedere, non è indicato per essere inserito in un'applicazione real time come elemento di sfondo, in quanto andrebbe ad appesantire l'applicazione nel calcolo di elementi come luce o collisioni, pur non essendo fondamentale per l'azione; oppure al contrario, creare un modello di un'automobile gestendo i poligoni in modo da avere le ruote anteriori perfettamente ruotabili e le portiere apribili lungo dei perni risulta essere un'inutile dispendio di forze se a noi interessa un rendering statico per una locandina.

Avere sempre ben chiaro il fine ultimo del prodotto che vogliamo creare e capire quali altri passaggi dovranno essere eseguiti sul modello 3D a modellazione conclusa prima di iniziare a creare l'oggetto, sono passaggi di grande importanza, pena ritrovarsi con un modello 3D inutilizzabile e dover spendere tempo ed energie per ricrearlo nel modo corretto, cosa inaccettabile in ambito lavorativo.

Questa banale osservazione sul fine ultimo del modello che andremo a creare è stato un'importante insegnamento ottenuto durante i mesi di lavoro a KR Games, dove tutti i modelli creati sono stati pensati esplicitamente per applicazioni in realtà virtuale real time.

# Capitolo 6

## Un'introduzione all'animazione

Partiamo col definire correttamente il concetto di animazione. In generale questa è un'illusione data da una serie di immagini statiche che si susseguono ad alta velocità, ogni singola immagine lascia nell'occhio umano una sorta di impronta che perdura per pochissimi istanti, quindi, se la velocità con cui si susseguono è abbastanza elevata, nelle aree corticali del nostro cervello si attiva un processo che fonde le immagini dando all'osservatore l'illusione della continuità.

Nel caso non sia raggiunta la velocità necessaria si ha il fenomeno del *flickering*, ovvero la percezione di fastidiosi scatti nel movimento, al caso opposto, ovvero se gli oggetti si muovono troppo velocemente rispetto alla soglia di campionamento dell'occhio umano si avrà il fenomeno del *motion blur*.

La frequenza di immagini al secondo (fps) oggi è divisa in diversi standard a seconda della tecnologia video usata: il cinema utilizza 24 fps, la tecnologia PAL della televisione analogica, usata in Europa e medio oriente, sfrutta 25 fps e la NTSC comune in Nord America e Giappone ne utilizza 30.

### 6.1 L'animazione convenzionale

Prima di addentrarci a discutere dell'animazione 3D, spendiamo qualche parola sull'animazione cosiddetta convenzionale a cui si rifanno quasi tutti i film d'animazione del XX secolo.

Questa prima fase dell'animazione nasce nel 1908 con il primo film d'animazione della storia, *Fantasmagorie* di Emile Cohl; in questo film per ogni fotogramma si è realizzato un disegno su carta, copiati ad uno ad uno su fogli di acetato trasparente, colorati a mano ed infine fotografati per essere trasferiti su pellicola proiettabile al cinema.

---

Bibliografia [4], [17], [18], [19]

## 6.1. L'ANIMAZIONE CONVENZIONALE

---

Ogni progetto "classico" si sviluppa secondo una ben preciso progetto di produzione diviso in step validi ancora oggi per le grandi creazioni cinematografiche, semplicemente oggi, i frame non vengono disegnati a mano, ma tramite appositi software:

- Fase 0 - Idea di produzione:** viene decisa l'idea che sarà alla base del progetto, delineando le caratteristiche dei personaggi principali e i punti chiave della trama.
- Fase 1 - *Storyboard*:** un insieme di schizzi accompagnati da testo utile per descrivere scene e personaggi. La creazione di uno *storyboard* completo può richiedere molto tempo per le grandi produzioni, ma risulta fondamentale per avere chiarezza su come devono essere girate le varie scene. Gli schizzi non devono brillare per precisione, ma devono indicare le azioni che vengono compiute dai personaggi che vanno ritratti da quella che poi sarà l'inquadratura della scena.
- Fase 2 - *SoundTrack* preliminare:** registrazione temporanea di suoni e voci da usare per calcolare tempistiche e sincronizzare correttamente l'animazione relativa all'effetto sonoro.
- Fase 3 - *Animatic*:** animazione molto basilare in cui solo i personaggi principali risultano in movimento; viene creata aiutandosi con le immagini dello storyboard da utilizzare come base per stendere la sceneggiatura.
- Fase 4 - *Design*:** definizione di tutti i dettagli più fini su personaggi e sfondi, quali scelta dei colori, stile, espressioni dei personaggi, particolarità nell'abbigliamento, postura eccetera, tramite disegni molto dettagliati.
- Fase 5 - *Timing*:** definizione dei keyframe principali e del numero di frame necessari per ogni singola animazione, così da avere le tempistiche esatte per ogni azione da far compiere al personaggio.
- Fase 6 - *Layout*:** in questa fase vengono definite ufficialmente inquadrature, background, pose dei personaggi secondari ed eventuali animazioni degli sfondi.
- Fase 7 - *Animazione*:** processo di disegno di tutti i singoli frame delle animazioni partendo dai frame principali di ogni sequenza e andando poi a legarli creando altri frame detti *in-betweenner*. Questa fase comprende il solo disegno dei personaggi principali; sfondi e animazioni di personaggi secondari possono essere elaborati in seguito.
- Fase 8 - *Sfondi*:** vengono disegnati gli sfondi seguendo le scelte stilistiche prese nei passi precedenti.
- Fase 9 - *Inking e painting*:** i disegni vengono trasferiti su fogli di acetato trasparente, colorati a mano e poi ulteriormente trasferiti su pellicola.

Ovviamente per l'animazione 2D moderna tutto è realizzato tramite appositi software, tranne per poche case di produzione che continuano ad usare tecniche di disegno classiche per una questione stilistica, infatti, l'animazione manuale, se fatta correttamente, conferisce un mood facilmente riconoscibile alla propria opera.

### 6.1.1 Differenti tecniche

Seppure i passi usati per la loro creazione fossero sempre gli stessi, vi è grande differenza tra le animazioni ad alto budget e quelle a basso budget: nelle prime si ha un'animazione piena, ovvero sia personaggi che sfondi risultano animati con molti frame con un misto di *shooting on one* e *shooting on two*, ovvero l'utilizzo di uno o due disegni per frame, mentre per le animazioni con budget minore vengono spesso riciclati gli sfondi e si animano solo i personaggi principali usando meno frame (*shooting on three, four...*).

Senza entrare nel dettaglio, ricordiamo che l'animazione tradizionale ha potuto fruire di varie nuove tecniche per merito dell'avanzare delle tecnologie, tra queste ricordiamo la combinazione tra animazione e *live action* con esempi importanti come *Chi ha incastrato Roger Rabbit?* nel 1988 o *Space Jam* nel 1996; il *rotoscoping*, ovvero una tecnica che prevede un girato in *live action* su cui vengono ricalcati i disegni, utilissimo per avere movimenti più naturali come vediamo in alcuni suoi utilizzi come per le spade laser di *Star Wars* del 1977; la *stop motion*, una particolare tecnica che sfrutta il posizionamento manuale di modellini reali che vengono fotografati e successivamente spostati di poco e ancora fotografati in modo che le immagini diventino i frame dell'animazione, l'effetto che ne si ottiene è peculiare, ma difficile da gestire correttamente visto l'enorme quantità di tempo e precisione richiesto anche per scene brevi; tra gli esempi di maggior successo ricordiamo *Nightmare before Christmas* e *Wallace e Gromit*.

Anche nel mondo dei videogame esistono produzioni particolari o sperimentali che utilizzano l'animazione classica per mettere in scena il gioco.

In alcuni casi questi risultano essere incredibilmente simili a cartoni animati interattivi come le classiche opere di Donald Virgil Bluth, regista, animatore e produttore cinematografico, che nel 1983 ebbe l'idea di produrre *Dragon's Lair* (Figura 6.1), un videogioco formato da spezzoni di cartone animato disegnato completamente a mano che procedevano in modo diverso a seconda degli input del giocatore creando quello che è considerato l'antenato dei *Quick time event*, ovvero un particolare concetto di gioco riassumibile con "premi il tasto indicato su schermo nel momento indicato", meccanica usata (ed a volte abusata) nei videogiochi moderni.

## 6.1. L'ANIMAZIONE CONVENZIONALE

---



Figura 6.1: Schermata di gioco di *Dragon's Lair*.

Al primo *Dragon's Lair* ne seguì un secondo nel 1991 ed un nuovo gioco che ne manteneva inalterate le meccaniche, *Space Ace* nel 1984.

Vi sono anche esempi più recenti di utilizzo dell'animazione classica nei videogames, come il fortunato *Cuphead*, gioco *Run 'n' Gun* per *XBox One* e PC, sviluppato da Studio MDHR e disponibile da settembre 2017.

In questo gioco, ogni singola animazione di personaggi, armi ed eventi è stata disegnata a mano ispirandosi ai corti animati anni trenta; proprio questa fase di ricerca sul design e di animazione ha richiesto molto tempo e risorse, ma ha portato ad uno stupendo risultato tanto che il gioco ha ricevuto il riconoscimento ai Game Awards 2017 di "Miglior Action Game dell'anno".



Figura 6.2: Schermata di gioco di *Cuphead*.

## 6.2 L'animazione 3D

Proseguendo sulla scia degli avanzamenti tecnologici tecnici visti prima, dagli anni novanta si decide di sfruttare la *Computer Grafica*, o CG, prima relegata ad ambiti di studio e di simulazione per videogiochi, anche per la produzione video e nel 1997 vediamo il primo lungometraggio completamente girato in CG, *Toy Story*, realizzato da Pixar.

Dopo questa prima opera ne sono seguite molte altre, e oggi.

La maggioranza dei film d'animazione risulta essere completamente in CG, ma anche l'animazione in computer grafica, proprio come quella tradizionale, deve seguire un proprio preciso progetto di produzione diviso in varie fasi che andiamo ora ad analizzare:

- Fase 1** - Come già visto, si traduce la versione orale della storia in una cartacea, ovvero lo *storyboard* in cui avremo schizzi delle varie scene corredati da testo per meglio spiegare inquadrature e movimenti di camera.
- Fase 2** - Si descrivono minuziosamente scenari e illuminazione degli stessi in modo da sapere con precisione quanti e quali saranno gli elementi in scena e come saranno illuminati tenendo dell'inquadratura scelta.
- Fase 3** - Vengono modellati i personaggi e il mondo in cui questi si muoveranno seguendo bozzetti e disegni fatti in fase di ideazione del prodotto cercando di replicare il più fedelmente possibile i *concept art*.
- Fase 4** - Tutti gli aspetti descritti durante la seconda fase vengono tradotti in texture, materiali ed effetti grafici che vengono applicati ad ambienti e personaggi.
- Fase 5** - Vengono realizzate le animazioni dei personaggi precedentemente modellati utilizzando una o più delle tante possibili tecniche di animazione 3D.
- Fase 6** - Viene creata l'illuminazione degli ambienti tenendo conto sia delle fonti di luce naturale che delle fonti di luce artificiali come possono essere neon, lampadari o esplosioni.
- Fase 7** - In ultimo saranno renderizzati ad uno ad uno tutti i fotogrammi che realizzano la scena; spesso questo procedimento è molto oneroso in termini di tempo, specialmente per scene complesse con molte fonti di luce ed effetti particellari, quindi si preferisce renderizzare separatamente personaggi e sfondi utilizzando diversi layer di rendering per snellire quest'operazione.

### 6.3 I dodici principi dell'animazione

Al di là della tecnologia utilizzata va detto che per un'animazione che risulti piacevole al pubblico, vanno rispettati i **12 principi** introdotti nel 1981 dagli storici animatori Disney, Ollie Johnston e Frank Thomas che risultano validi ancora oggi tanto per l'animazione tradizionale, quanto per la moderna *computer animation*:

- 1) ***Squash & Stretch***: principio secondo cui il volume di un corpo animato debba rimanere lo stesso, ma può cambiare di forma come per gli stiramenti o schiacciamenti di un pallone, questo perché il corpo umano e vari oggetti presentano, in realtà, una notevole elasticità.
- 2) ***Anticipation***: per guidare il pubblico nella comprensione di un'azione di una scena che altrimenti risulterebbe troppo improvvisa da capire, si deve antecedere a quest'azione un piccolo movimento che anticipi ciò che sta per accadere, come il caricare il braccio prima di sferrare un pugno; l'effetto opposto è la "gag a sorpresa" che ha effetto proprio perché non è possibile prevederla.
- 3) ***Staging***: con questo concetto ampio si intende tutto ciò che fa in modo che l'azione principale di una scena sia effettivamente in armonia con il resto e, di conseguenza, di senso facilmente comprensibile; inoltre l'azione va svolta in modo che lo sguardo dello spettatore sia portato a vederla e non distratto da azioni di contorno.
- 4) ***Straight Ahead & Pose to Pose***: questi concetti vanno ad indicare i due diversi approcci nel creare un'animazione. Nel primo caso tutta l'animazione viene creata in sequenza da inizio a fine, cosa che crea un risultato finale dal mood molto spontaneo; nel secondo caso l'animazione è creata definendo prima alcuni frame chiave e poi creare i frame intermedi, questo metodo dà più controllo sull'animazione in generale e i risultati sono più chiari e fluidi. Va quindi usato un metodo piuttosto che l'altro a seconda delle situazioni.
- 5) ***Follow Through & Overlapping Action***: sotto questo principio stanno tutti quei movimenti che corredano un'azione principale quando questa cambia di velocità o direzione, ad esempio, quando un personaggio si ferma di colpo da una corsa, questo non si fermerà istantaneamente in modo rigido, ma per avere naturalezza, tutte le sue parti non legate alla sua struttura ossea (come capelli, vestiti, ed eventuali altre appendici). Oltre che per dare naturalezza questi movimenti servono a conferire carattere ai movimenti, ad esempio per far sembrare atletico o, al contrario scoordinato, un personaggio.
- 6) ***Slow In & Slow Out***: per simulare la fisica reale bisogna tener conto delle accelerazioni dei movimenti ricordando che ogni azione parte lenta, accelera e rallenta nuovamente, non tiene una velocità costante da inizio a fine e infatti, una cosa del genere in un'animazione risulterebbe innaturale e robotica.

### 6.3. I DODICI PRINCIPI DELL'ANIMAZIONE

---

- 7) **Arcs**: i movimenti umani e animali molto raramente seguono percorsi in linea retta da un punto A ad un punto B, ma solitamente si segue un arco di circonferenza per ogni più piccola azione; per rappresentare ciò in fase di animazione è comodo creare un frame centrale all'animazione in cui l'oggetto in movimento sia al culmine di questo arco di circonferenza tra partenza e arrivo.
- 8) **Secondary Actions**: indica tutte quelle azioni secondarie che servono a rafforzare la credibilità della scena accompagnando un'azione principale senza però nasconderla, quindi questa va pianificata attentamente nei tempi per non oscurare dettagli importanti dell'azione principale. L'azione secondaria può anche essere semplicemente il cambio d'espressione del personaggio mentre questo agisce.
- 9) **Timing**: indica la temporizzazione della scena. Anche se i singoli frame risultassero chiari e puliti, un cattivo timing rovinerebbe completamente la scena, poiché senza un chiaro studio dei tempi non si può far capire l'attitudine e la personalità di un personaggio tramite i movimenti. Legato al timing vi è anche il concetto di "Passo uno" o "Passo due", cioè utilizzare lo stesso disegno una volta o ripeterlo due volte consecutivamente; il passo due, come visto in precedenza, permette una mole di lavoro decisamente minore ed è quindi usato per produzioni a basso budget a differenza dell'animazione a "Passo uno". Si può comunque usare un misto tra le due tecniche usando il "Passo uno" per quelle azioni che risultano più importanti su ogni altra e rimanere al "Passo due" il resto.
- 10) **Exaggeration**: indica il dover esagerare alcune caratteristiche di un personaggio per renderle visibili allo spettatore, cioè, se si vuole far vedere un personaggio "pensieroso", questo dovrà essere esageratamente "pensieroso" perché questa sua caratteristica passi allo spettatore.
- 11) **Solid drawing**: i disegni devono avere volume e peso, non devono apparire piatti, ma profondi e al contempo immersi realisticamente nell'ambiente. Ciò nonostante deve rimanere una certa flessibilità nel personaggio poiché nessuna creatura è totalmente rigida.
- 12) **Appeal**: indica l'insieme di qualità del personaggio che fanno in modo che lo spettatore voglia vederlo in azione: queste comprendono sia bellezza, fascino, lineamenti puliti che la caratterizzazione, ad esempio anche gli antagonisti crudeli e spietati, devono avere *appeal* dato dalla personalità che esprimono tramite il design e i movimenti, altrimenti lo spettatore perderebbe interesse nelle vicende.

## 6.4 Un approfondimento sull'animazione 3D

Ma a livello pratico, come vengono create le animazioni tramite i software appositi? Abbiamo visto prima che queste sono generate dal veloce susseguirsi di frame statici, ma le posizioni dell'oggetto in sé, come vengono calcolate?

Si utilizzano varie diverse forme di interpolazione che possono essere gestite mediante curve parametriche che determinano la posizione dell'oggetto per ogni frame rispetto al tempo, quindi la pendenza della curva andrà a indicare quanto velocemente l'oggetto si avvicina alla posizione finale. Questa traiettoria viene parametrizzata secondo la lunghezza d'arco e otterremo che a tratti di lunghezza d'arco costante corrisponde velocità costante.

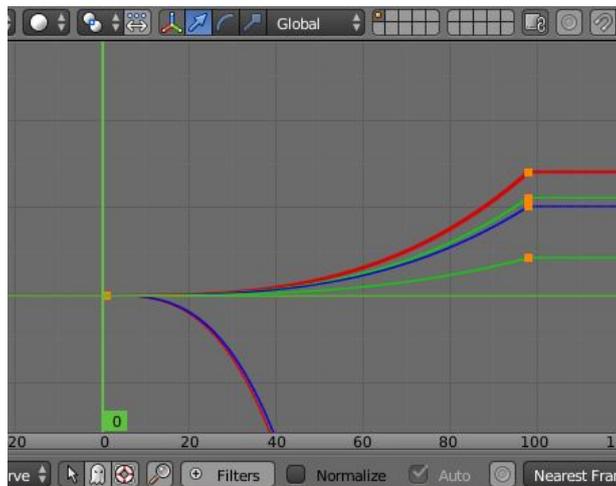
Esistono apposite funzioni di controllo della velocità applicabili per ottenere effetti particolari senza dover agire manualmente sulle curve, tra le principali ricordiamo:

- **Funzione *Ease-in***: crea un'accelerazione regolabile in partenza da fermo.
- **Funzione *Ease-out***: crea una decelerazione regolabile fino allo stop.
- **Funzione *quadratica***: crea un'accelerazione che segue una funzione quadratica.
- **Funzione *cubica***: crea un'accelerazione che segue una funzione cubica (Figura 6.3 (a)).
- **Funzione *back***: fa proseguire l'oggetto oltre il punto d'arrivo predefinito per poi farlo tornare indietro; molto utile per simulare un movimento fatto con forza eccessiva (Figura 6.3 (c)).
- **Funzione *bounce***: particolare funzione di interpolazione che simula l'effetto della forza di gravità con accelerazione costante verso l'arrivo, ma giunto a destinazione l'oggetto rimbalza indietro più volte lungo la sua traiettoria fino a fermarsi (Figura 6.3 (b)).

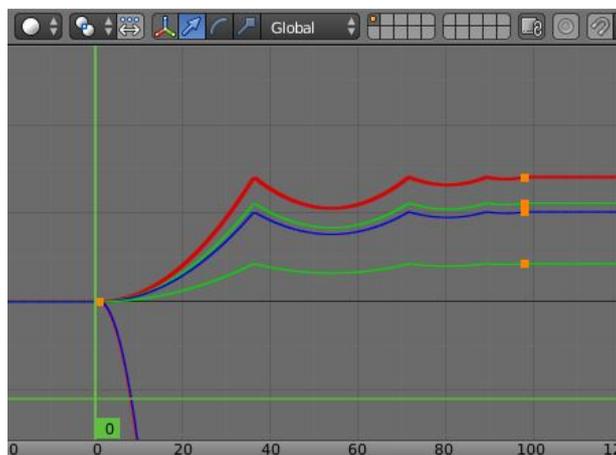
Ovviamente queste considerazioni valgono non solo per la posizione, ma anche per le altre trasformazioni fondamentali, ovvero rotazione e scalamento.

Ogni oggetto può quindi essere animato semplicemente indicando in più keyframe i suoi valori di rotazione, scalamento e posizione, ma questo metodo di animazione è l'imitante, in quanto ogni oggetto sarà animato senza la possibilità di trasformare solo parti della texture che lo compone.

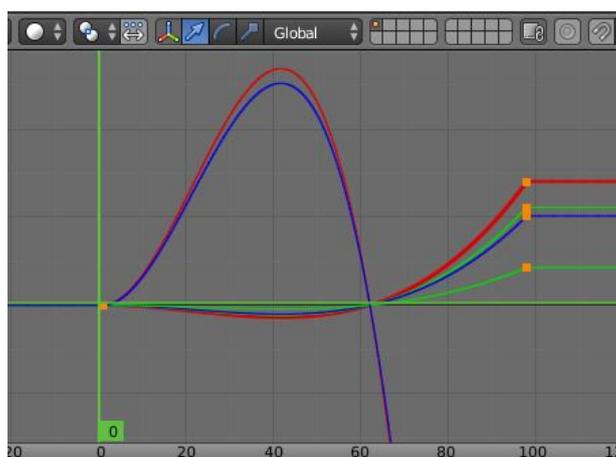
## 6.4. UN APPROFONDIMENTO SULL'ANIMAZIONE 3D



(a) *Funzione cubica.*



(b) *Funzione bounce.*



(c) *Funzione back.*

Figura 6.3: Grafici su *Graph editor* di *Blender* mostrano l'interpolazione due keyframe fissi con tre tipi di funzione diversi.

### 6.4.1 Animazione con armature

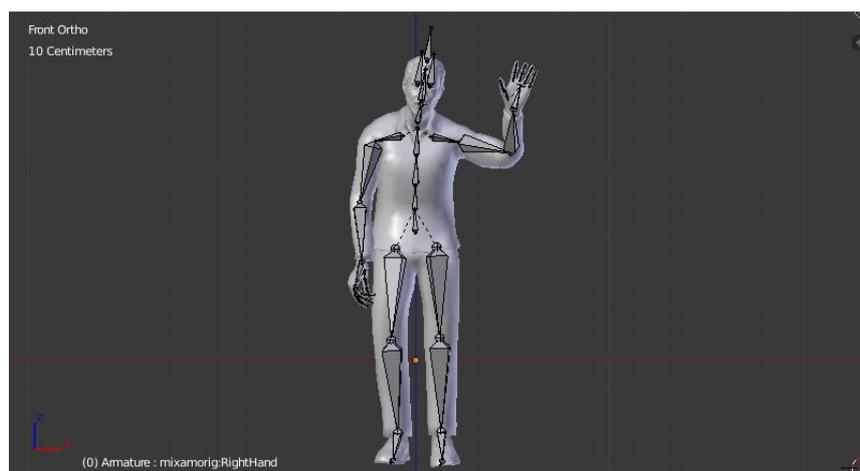
Per le animazioni di modelli complessi come possono essere forme umanoidi, è però impossibile lavorare direttamente sulla mesh stessa, ma si procede applicando un particolare modello gerarchico al personaggio ed è su questo che andremo ad agire.

Questo modello gerarchico è chiamato *armatura* ed è composto da una serie di componenti definite "ossa" che sono legate tra loro tramite vincoli di connettività.

Questi vincoli, se applicati con cura, permettono all'armatura di simulare correttamente i movimenti dello scheletro umano senza la possibilità di compiere torsioni o spostamenti impossibili per la nostra anatomia.



(a) *Posizione di riposo.*



(b) *Posa costruita con cinematica diretta.*

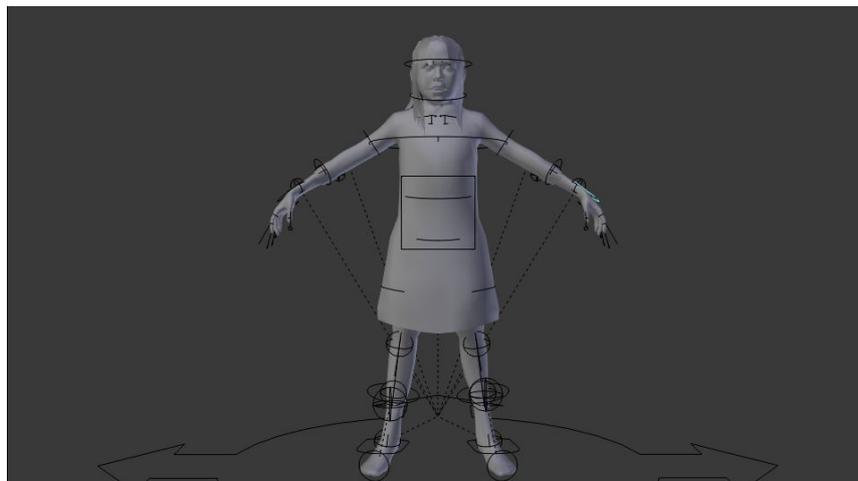
Figura 6.4: Armatura applicata ad un personaggio 3D in *Blender*.

Ogni *osso* dell'armatura è legato ad altri tramite un *giunto* che può avere uno o più gradi di libertà: un grado di libertà permette una rotazione attorno ad un solo asse, con due gradi di libertà avremo rotazione su due assi e tre gradi di libertà permettono ovviamente rotazioni libere sui tre assi.

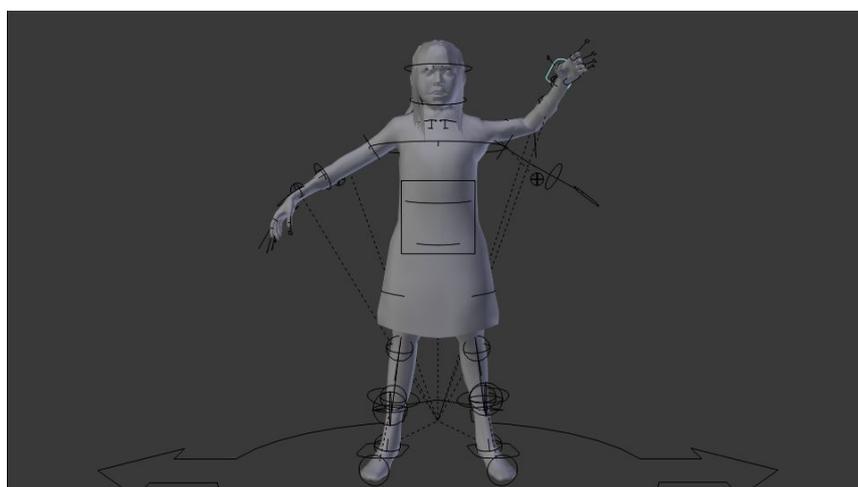
La gerarchia che vige tra le singole ossa prevede l'esistenza di un osso radice che funge da centro relativo per ogni altra; dalla radice si diramano le ossa più alte nella gerarchia che a loro volta possono avere dei figli e così via; questo è fondamentale poiché qualunque trasformazione applicata ad un osso viene applicata anche ai suoi figli relativamente al giunto di livello superiore nella gerarchia.

Lavorando sugli scheletri è possibile utilizzare due approcci differenti per animarli e posizionarli nella posa desiderata: il primo, molto lento e complesso, è detto *cinematica diretta* (Figura 6.4) e consiste nello specificare ad uno a uno tutti gli angoli dei giunti presenti nella nostra armatura; fortunatamente questo lungo processo è stato sostituito quasi totalmente dalla *cinematica inversa* (Figura 6.5), processo con cui basta decidere la posizione finale di un dato osso dell'armatura per fare in modo che le altre ossa di quell'arto si orientino di conseguenza lungo i rispettivi giunti per permettere la posa dell'osso spostato inizialmente rispettando tutti i loro vincoli, ad esempio basta muovere l'osso che gestisce una mano per fare in modo che avambraccio, braccio e spalla si muovano correttamente seguendo la posa scelta per la mano.

Ovviamente, quando si dovrà animare uno scheletro, non sempre l'interpolazione tra due pose create con la cinematica inversa sarà soddisfacente, quindi si dovrà procedere a modificare manualmente le pose per i frame tra keyframe iniziale e finale usando anche, qualora necessario, la cinematica diretta per avere più controllo.



(a) *Posizione di riposo.*



(b) *Posa ottenuta spostando il controllore della mano con cinematica inversa che influenza l'intero il braccio.*

Figura 6.5: Il modello di Allison nel videogioco *The Villa: Allison's Diary*.

Scendiamo nel dettaglio analizzando brevemente alcuni esempi comuni di animazione di un'armatura: uno dei casi più frequenti è l'animazione degli arti superiori di un personaggio che devono risultare realistici e fluidi nei movimenti, per ottenere ciò si segue un modello che prevede il braccio come un'entità a sé stante.

Se ignoriamo momentaneamente la mano, il braccio risulta essere rappresentabile come un'entità formata da tre giunti in corrispondenza di spalla, gomito e polso, e due ossa, ovvero braccio e avambraccio a sette gradi di libertà complessivi: tre per il giunto in corrispondenza della spalla che può compiere rotazioni sui tre assi, uno per il giunto che si riferisce al gomito che può ruotare lungo un solo asse e di nuovo tre per il giunto che si trova all'altezza del polso.

Ogni posa diversa assumibile dal braccio è data semplicemente dall'insieme del valore degli angoli di ciascuno dei sette gradi di libertà. Questi angoli però, per permettere il corretto funzionamento della cinematica inversa, devono avere dei vincoli legati all'anatomia umana: il gomito non può ruotare più di  $160^\circ$  circa, il polso ha un limite massimo di rotazione di circa  $180^\circ$  su ogni asse e così via, pena realizzare un'animazione visibilmente innaturale. Se questi vincoli sono corretti, allora sfruttando la cinematica inversa basterà posizionare il polso nel punto voluto per avere la corretta posa dell'intero braccio.

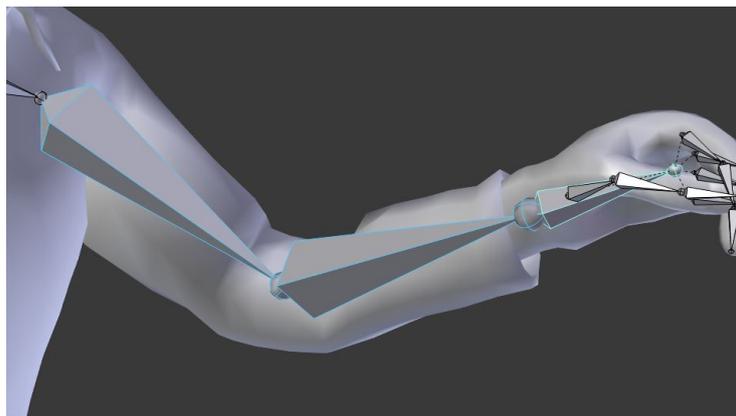


Figura 6.6: Particolare dell'ossatura di un braccio di un personaggio 3D con armatura in *Blender*.

Altro classico esempio è quello dell'animazione della camminata di un personaggio: azione apparentemente semplice e comune che presenta però una grossa difficoltà, ovvero il fatto che la camminata è dinamicamente stabile, ma non lo è staticamente.

Per gestire bene la camminata vanno tenuti in conto attributi che variano da personaggio a personaggio, come lunghezza del passo, rotazione dell'anca e posizione dei piedi; la singola gamba è rappresentabile nuovamente come un'entità a sé stante formata da tre ossa in corrispondenza di femore, tibia e piede e quattro giunti per otto gradi di libertà complessivi: un giunto per l'anca avente 3 gradi di libertà, uno per il ginocchio ad un grado di libertà, uno per la caviglia a tre gradi di libertà e l'ultimo per le dita, viste complessivamente, ad un grado di libertà.

Il ciclo di camminata è diviso in fasi a seconda della posizione dei piedi rispetto al terreno.

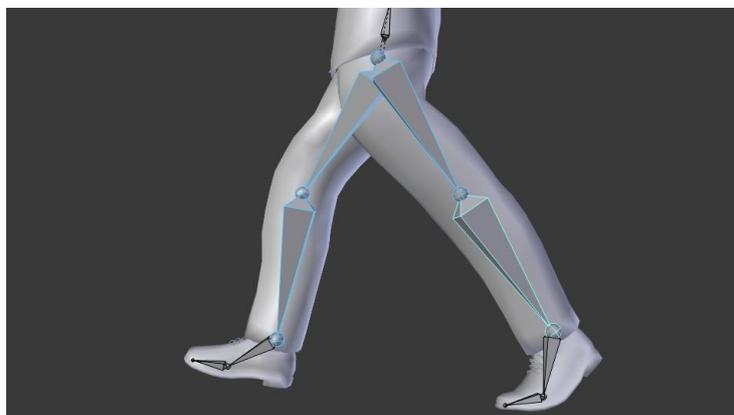


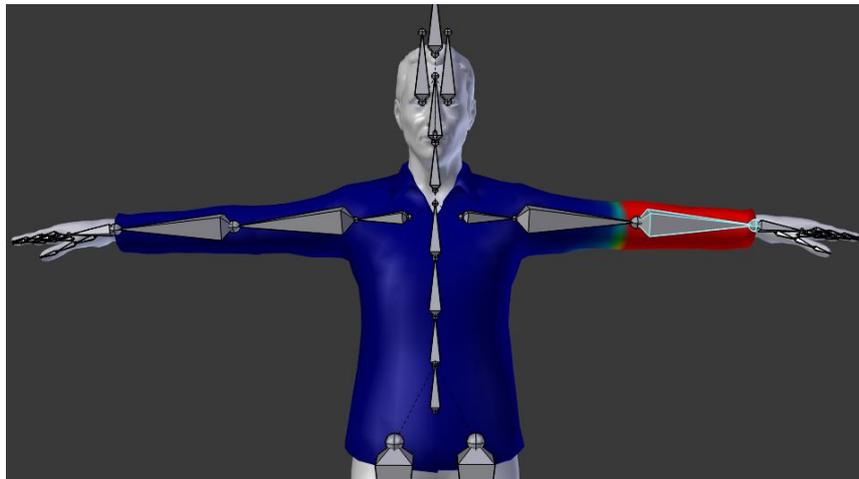
Figura 6.7: Particolare dell'ossatura delle gambe di un personaggio 3D con armatura in *Blender*.

Ma come può l'armatura influenzare il movimento del personaggio precedentemente modellato?

Vi è un processo chiamato *skinning* che consiste nell'associare ad ogni osso dello scheletro un'area di influenza sulla texture, di modo che, al movimento dell'osso, la parte di modello selezionata si muova tanto più il peso dell'associazione al dato osso è alta.

Per determinare questi "pesi d'associazione" ci si può affidare ad algoritmi matematici che calcolano automaticamente le aree di influenza delle ossa scegliendo i vertici del modello più vicini ad esse; questo è un buon punto di partenza, ma siccome spesso i personaggi presentano vestiti complessi, oggetti legati ad esempio alla cintola o alle spalle, capelli lunghi o altre appendici, il calcolo automatico dei pesi avrà sempre qualche problema.

Per ovviare a queste mancanze sarà necessario agire manualmente andando a definire i pesi per i singoli vertici (o gruppi di essi) che non hanno risposto a dovere alla pesatura automatica; per fare questo si può semplicemente associare i vertici scelti al gruppo controllato dall'osso voluto dando così peso pari a uno, cioè la massima responsività rispetto il movimento dell'osso, oppure utilizzare una tecnica chiamata *weight painting* (Figura 6.8) che sfrutta una serie di strumenti che agiscono come pennelli sulla mesh, e, nelle zone in cui agiscono, possono aumentare o diminuire il peso rispetto all'osso scelto. Questa particolare tecnica è utile soprattutto per creare un effetto "morbido" nelle zone di intersezione tra le aree controllate da due diverse ossa.



(a) Weight painting *relativo all'avambraccio sinistro.*



(b) Weight painting *relativo alla vita.*

Figura 6.8: Semplice esempio in *Blender* di *weight painting* per un personaggio 3D: le zone rosse sono le più influenzate dall'osso selezionato, mentre quelle in blu non lo sono affatto.

Proprio la tecnica del *weight painting* in combinazione con l'animazione per cinematica inversa, sono state le più usate durante la creazione delle animazioni per KR Games, e nei prossimi capitoli analizzeremo nel dettaglio alcune di queste creazioni.



## Capitolo 7

# La mia esperienza presso KR Games: organizzazione e tecnologia usata

Si passa ora ad illustrare il lavoro svolto presso KR Games lungo un arco temporale di sei mesi, da fine maggio 2017 a inizio dicembre dello stesso anno, periodo durante il quale ho lavorato come **3D modeler** e **3D animator** per il videogioco horror in VR *The Villa: Allison's Diary*: è stato mio compito creare ambientazioni di gioco funzionali e credibili, alcuni personaggi e relative animazioni sempre seguendo precise direttive volte ad ottimizzare il mio lavoro creativo per la fruizione in VR del prodotto.



Figura 7.1: Immagine promozionale di *The Villa: Allison's Diary*.

## 7.1. IL WORKFLOW

---

Per la creazione dei suddetti elementi ho potuto lavorare da remoto, sfruttando *Dropbox* per la condivisione dei contenuti, archiviati in una gerarchia organizzata di cartelle, per poter sempre avere la versione più aggiornata di un dato contenuto; per la comunicazione diretta, invece, si è scelto di utilizzare il sistema di chiamata *Skype* con condivisione dello schermo, in modo da avere feedback visivo real time durante il processo creativo di modellazione e animazione.

Nei periodi di poco antecedenti a importanti scadenze di progetto, si è preferito avermi direttamente sul posto, in modo da potermi confrontare direttamente con il resto del team di sviluppo per migliorare eventuali criticità riscontrate.

### 7.1 Il workflow

Prima di scendere nei dettagli tecnici della creazione dei vari elementi, analizziamo brevemente i passi usati nel *workflow* di KR Games per impostare il lavoro, gestire i tempi e creare l'applicazione VR completa.

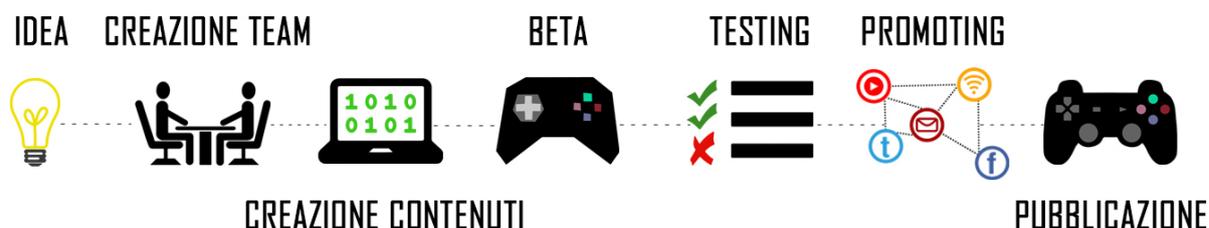


Figura 7.2: Schema di una pipeline di lavoro per la realizzazione di un videogioco, in generale. Di seguito vengono espone nel dettaglio le singole fasi.

#### Fase 1 - Preparazione

- (a) **L'idea e le meccaniche base:** questa prima fase consiste in un *brainstorming* creativo volto a trovare un'idea vincente che stia alla base del titolo che andremo a creare; viene deciso il genere del videogioco (soprattutto, action, survival horror, gdr...) e le meccaniche di base che saranno al centro del suo gameplay e attorno cui sarà costruito il resto; si decide la trama che deve essere adattata alle meccaniche scelte, e non il contrario.

Viene inoltre decisa la piattaforma per cui il gioco sarà sviluppato in modo da avere subito chiari eventuali limiti tecnologici e logici da rispettare.

Da qui si inizia a pianificare quante risorse economiche e temporali investire nel progetto.

- (b) **Progettazione della struttura:** in questa fase si pianifica come il giocatore può progredire nella sua avventura decidendo con cosa e in quali modi può interagire con ambiente e personaggi; inoltre va pensato come introdurre al giocatore le meccaniche che dovrà usare e i corretti input da utilizzare (tutorial situazionale, schermata dedicata, suggerimenti nascosti...).

### Fase 2 - Creazione del team di sviluppo

- (a) **Organizzazione del team:** un videogame complesso richiede diversi tipi di conoscenze e abilità diverse per essere creato con successo, ed ovviamente servono più persone per avere tutte le competenze necessarie. In base al tipo di esperienza che si vuol creare si dividono i compiti in modo che ad ogni membro del team tocchino le mansioni a lui più congeniali (modellazione, animazione, grafica, programmazione, *level design*, ecc...), organizzando i tempi di lavoro e scadenze in modo da poter proseguire in parallelo per potersi sempre confrontare con gli altri membri del team in caso di bisogno.
- (b) **Stesura del documento di progetto:** il documento di progetto è composto da una dettagliata descrizione di tutti gli elementi che comporranno il prodotto finito: storia, meccaniche, stile grafico, personaggi, controlli, design dell'interfaccia, *concept art* e campioni audio. Il documento di progetto non ha una formattazione fissa, ma anzi è consigliato variarla in base al gioco presentato in modo da rispettarne lo stile anche in formato cartaceo.  
Questo documento ha due funzioni principali: serve da cartina di tornasole per chi lavora sul gioco descritto, di modo che non si possa perdere traccia delle idee iniziali durante le fasi avanzate di sviluppo; inoltre, un documento di progetto ben scritto può far interessare potenziali investitori convincendoli della bontà del nostro lavoro.

### Fase 3 - Lavorazione

- (a) **Scripting:** scelto il motore su cui girerà il gioco (*Unity 3D*, *Unreal*, *Blender Game*, ecc...) è necessario iniziare ad impostare gli script principali con il linguaggio di programmazione scelto.  
Questi script sono quelli legati alle meccaniche base del gioco come il controllo dei requisiti necessari per passare da un livello ad un altro o il "restart" in caso di "game over".  
Per ora viene tutto testato su texture di personaggi e ambienti temporanee, per semplificare il lavoro di debugging e controllo. Sostanzialmente viene implementata la logica di gioco.
- (b) **Creazione del contenuto:** inizia parallelamente allo scripting la fase di modellazione 3D di ambienti e personaggi (o creazione di *sprite* qualora si sia optato per un gioco 2D vecchia scuola).  
Per comodità si inizia a modellare gli elementi richiesti da chi gestisce lo scripting in modo che questi possano essere immediatamente testati per valutarne

l'effettiva efficacia in game. Oltre alla creazione del modello semplice questo va poi texturizzato, gli va applicato un materiale adeguato e deve essere retopologizzato nel caso vi siano particolari esigenze sul fronte dell'animazione.

Si ricorda che va cercato un compromesso tra il numero di poligoni della texture, quindi la sua qualità visiva, e il peso a livello di calcolo computazionale che questa avrà sul progetto.

Rientra in questa fase anche la creazione delle animazioni che saranno attivate in game tramite script.

- (c) **Implementazione del contenuto nel mondo di gioco:** il contenuto creato alla fase precedente viene sostituito alle texture di prova in via definitiva e si crea una versione finale del mondo di gioco aggiungendo un'illuminazione precalcolata, oppure real time, ed effetti sonori oltre ad eventuali filtri grafici ed elementi estetici supplementari come la realizzazione del cielo mediante uno *skybox*.

Si decide, inoltre, come gestire i calcoli computazionali ad esempio nascondendo gli oggetti non visibili per non appesantire l'uso della CPU, oppure dividendo il mondo di gioco in sezioni che vanno caricate man mano che si attraversano cancellando quelle attraversate in precedenza.

**Fase 4 - Beta:** vengono create delle copie del gioco in versione *Beta* (non definitiva) in modo che possano essere testate da persone esterne al team di sviluppo poiché, elementi scontati per chi ha lavorato al progetto, possono non esserlo per chi non vi ha preso parte attivamente.

Grazie ai feedback ricevuti si possono modificare particolari valori per gestire meglio difficoltà, velocità di movimento, altezza di salto e tutti quei parametri legati alle meccaniche di gioco; in più si può capire quali elementi sia estetici che di gameplay risultino meno apprezzati per migliorarli.

Questa fase è utile anche per scoprire *bug* di programmazione che peggiorano l'esperienza di gioco e poterli quindi rimuovere prima della pubblicazione ufficiale.

**Fase 5 - Testing:** è buona abitudine riprovare più volte l'intero gioco in ogni sua parte per individuare anche il più piccolo bug o difetto che possa esser finora sfuggito.

È importante dedicare molto tempo ed attenzione a questa fase vista l'importanza del compito: un gioco difettoso non è mai un buon biglietto da visita.

**Fase 6 - Promoting:** quando si ha una *build* definitiva del gioco (ma anche già durante lo sviluppo), è necessario dare visibilità al proprio prodotto sfruttando vari media: si può creare una pagina sui social per condividere schermate di gioco o video, creare un sito web del gioco con descrizioni e dettagli sulle tecnologie utilizzate che possa reindirizzare al sito della casa di sviluppo o creare teaser trailer per incuriosire il pubblico.

Suscitare l'interesse dei potenziali acquirenti o futuri investitori è estremamente importante.

## 7.1. IL WORKFLOW

**Fase 7 - Pubblicazione:** si può passare alla pubblicazione del proprio gioco sulla piattaforma (o piattaforme) scelta considerando che, ad oggi, gli *app Store* e *Steam* sono i servizi più indicati per sviluppatori indipendenti come nel nostro caso. Si potrebbe optare per una pubblicazione del gioco su di un sito personale, ma si perde in visibilità ed i costi di hosting sono talvolta molto elevati.

È utile ascoltare eventuali recensioni o commenti dell'utenza per individuare eventuali problemi e procedere a correggerli con un aggiornamento, oppure inserire contenuti aggiuntivi come nuovi livelli nel caso si tratti di un gioco *arcade*.

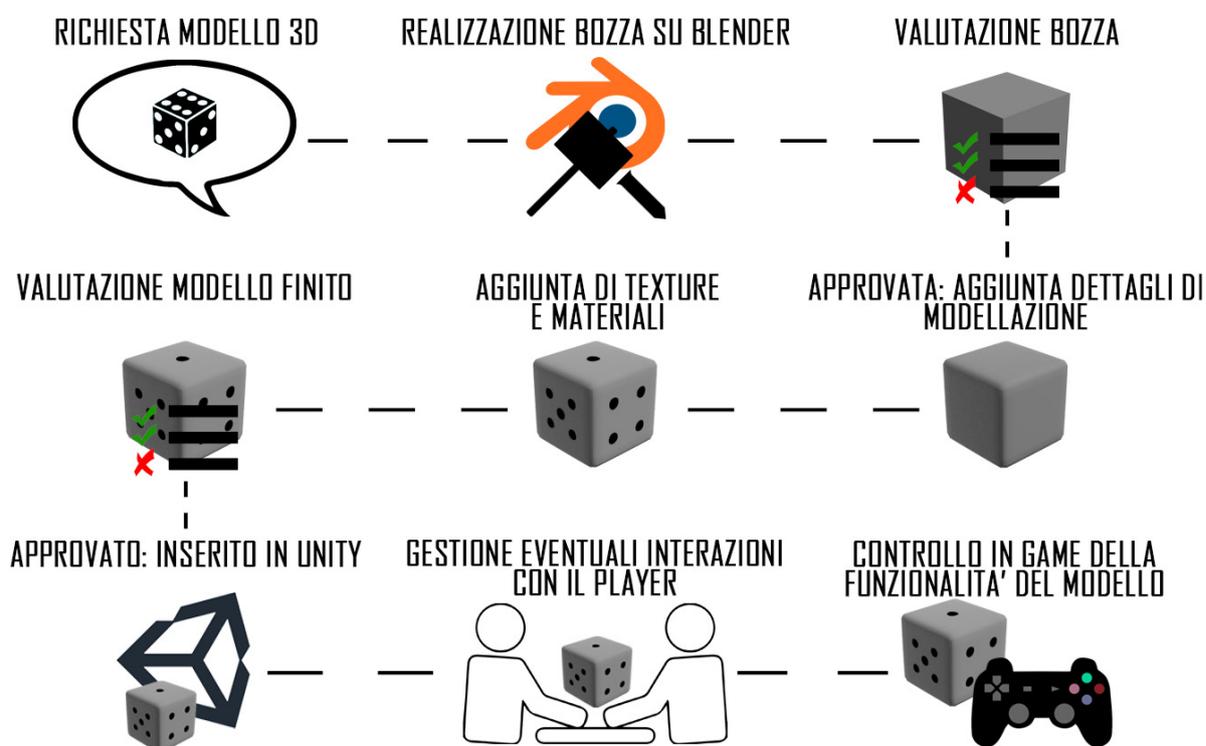


Figura 7.3: Schema di una pipeline di lavoro per la realizzazione di un videogioco, dal punto di vista del modellatore. Nel capitolo 8 viene, in particolare, analizzato questo processo nel caso specifico di *The Villa: Allison's diary*.

### 7.2 Comuni criticità delle applicazioni VR

Rispetto al *workflow* precedentemente illustrato vanno però tenute in conto delle problematiche specifiche che si palesano sviluppando giochi per dispositivi VR, queste sono soprattutto tecniche e legate alla natura complessa dei visori per realtà virtuale.

La prima problematica è data dal bisogno di **mantenere il framerate del gioco stabile**, in una finestra che va da 60 frame a 90 frame al secondo, poiché, con frequenti oscillazioni di questo valore o un valore inferiore al limite minimo, non si riesce a simulare correttamente la fluidità di visione dell'occhio umano e questo può portare, insieme ad altri elementi come la mancata sincronia tra azione ed audio, al *motion sickness*, una sensazione di malessere nel giocatore data da stimoli visivi che il nostro cervello interpreta come "sbagliati" rispetto i feedback ottenuti con gli altri sensi.

Questa sensazione, se di durata prolungata, può portare a nausea e perdita di equilibrio per i soggetti più sensibili; generalmente si raccomanda una pausa ogni 40 minuti con indosso un visore.

Per evitare questo problema è necessario pianificare con cautela il peso dell'applicazione sulla piattaforma su cui gira, ad esempio alleggerendo pesanti effetti grafici, usando texture ad alta definizione solo quando necessarie, evitando di usare un alto numero di poligoni per oggetti poco visibili e usando con moderazione simulazioni fisiche come il calcolo del movimento di fluidi, fumo e sistemi particellari con relative collisioni, cercando invece sistemi alternativi quando possibile.

Altra grande problematica consiste nel **progettare una UI<sup>1</sup> pulita ed efficace**, compito già non facile per un videogame classico, ma che nel caso dei giochi VR aumenta esponenzialmente di difficoltà poiché serve un'interfaccia non invasiva per evitare di perdere l'effetto di immersione nel mondo di gioco, ma che al contempo sia intuitiva dato che spesso, il giocatore, non usa un controller, ma dà i suoi input solo tramite i movimenti della testa.

---

<sup>1</sup>Interfaccia Utente

## 7.2. COMUNI CRITICITÀ DELLE APPLICAZIONI VR



(a) L'HUD di *The Villa: Allison Diary*.



(b) L'HUD di *Xenoblade Chronicles X*.

Figura 7.4: Confronto tra due HUD: quello di *The Villa: Allison Diary*, (a), è sostanzialmente nullo per la massima immersione, mentre in *Xenoblade Chronicles X*, (b), gioco per *Wii U* sviluppato da Monolith Soft, è molto presente e complesso.

Va infine considerato il problema rappresentato dal **prezzo dell'attrezzatura necessaria** per sviluppare e testare adeguatamente un videogame in VR, soprattutto per gli studi indipendenti.

Servono infatti i costosi visori, da integrare a volte con cellulari di fascia alta, ed ovviamente uno o più pc ad alte prestazioni in grado di reggere i vari software richiesti per la creazione dell'applicazione con la possibilità di usarne più di uno contemporaneamente senza causare rallentamenti alla macchina oltre alla necessità di mantenere stabile il framerate richiesto dall'applicazione VR (60-90fps) in fase di *testing*. Basterebbero invece strumenti dalla minore potenza di calcolo, e quindi dai costi minori, per creare applicazioni classiche.

## 7.3 I software utilizzati

In questa sezione analizzerò più nel dettaglio i software ed i mezzi tecnologici usati durante il mio periodo lavorativo presso KR Games.

Come modellatore ed animatore 3D il primo passo è stato scegliere quale software di modellazione utilizzare tenendo conto di vari fattori come la compatibilità dei formati dei file con il programma usato per creare l'applicazione finale, la familiarità con il programma, la quantità di funzioni disponibili (animazione, texture, effetti grafici, simulazioni fisiche, ecc. . . , oltre alla modellazione) ed eventuali costi nel caso si opti per un software a pagamento.

### 7.3.1 *Blender*: le caratteristiche che ne hanno decretato la scelta

La scelta è ricaduta su *Blender*, software libero multiplatforma per modellazione, *rigging*, animazione, *rendering* e *compositing*; l'insieme di queste funzioni lo rende paragonabile ai più famosi e richiesti *3D Studio Max*, *Cinema 4D* e *Maya*.

L'autore principale, Ton Roosendaal, fondò la società **Not a Number Technologies (NaN)** nel 1998 per distribuire il programma che inizialmente fu rilasciato come software proprietario a costo zero (freeware) fino alla bancarotta di NaN nel 2002.

I creditori acconsentirono a pubblicare *Blender* come software libero, sotto i termini della licenza GNU, General Public License, per il pagamento una-tantum di 100 000€.

Il 18 giugno 2002 fu iniziata da Roosendaal una campagna di raccolta fondi e il 7 settembre 2002 fu annunciato che l'obiettivo era stato raggiunto e il codice sorgente di Blender fu pubblicato in ottobre.

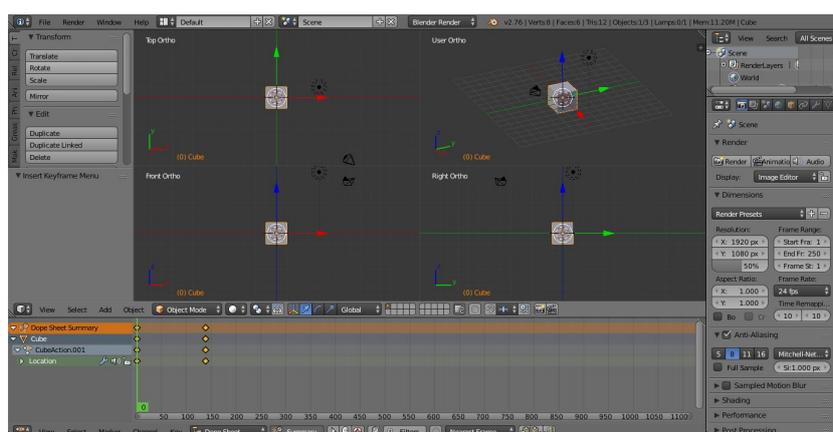


Figura 7.5: L'ambiente di sviluppo *Blender* nella sua variante *quad-view* dedicata all'animazione.

*Blender*, pur essendo un software relativamente leggero, ha molte features avanzate nel campo della modellazione ed animazione, grazie ai continui aggiornamenti rilasciati gratuitamente lungo il corso degli anni.

Segue una carrellata delle sue principali caratteristiche.

### Primitive geometriche e *add-ons*

*Blender* può contare sul supporto di una grande varietà di primitive geometriche tra cui: texture poligonali, *NURBS*, curve di bezier, *metaball*, *path* e font vettoriali oltre ad un grande numero di elementi aggiuntivi scaricabili come *add-ons* tra cui alcuni molto utili per creare componenti organiche come la vegetazione e elementi ripetuti come mattoni che compongono un muro.

### Esportabilità

La possibilità di esportare il proprio file in vari formati per permetterne l'utilizzo su vari altri programmi. Tra questi abbiamo: *Collada* (.dae), *3D Studio* (3ds), *FBX* (.fbx), *Motion Capture* (.bvh) e *Wavefront* (.obj).

La scelta di un diverso formato permette l'utilizzo del file su altri software come *Unity 3D*, *3D Studio Max*, *Maya*, *Unreal* e altri, lasciando quindi molta libertà d'azione per poter implementare nel proprio file effetti non utilizzabili direttamente in *Blender*.

### Il supporto all'animazione

*Blender* dispone di una comoda *timeline* e di appositi pannelli chiamati *Dope Sheet* e *Graph Editor* per gestire le animazioni tramite l'inserimento di *keyframe*.

Possiamo quindi animare un oggetto inserendo un keyframe di partenza e uno di arrivo modificando posizione, rotazione e scala dell'oggetto, *Blender* farà un'interpolazione tra i due frame per creare l'animazione; anche l'interpolazione fra questi keyframe può essere modificata scegliendo tra diversi tipi di *easing* (sinusoidale, quadratico, cubico, esponenziale...) oppure modificandola manualmente tramite un grafico a curve di bezier (Figura 7.6).

## 7.3. I SOFTWARE UTILIZZATI

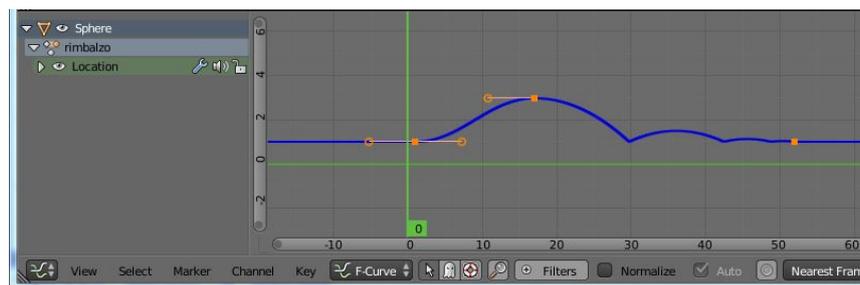


Figura 7.6: L'andamento del grafico ottenuto al pannello *Graph editor* di *Blender* che indica lo spostamento sull'asse *z* rispetto al tempo per l'animazione di una palla che rimbalza utilizzando l'interpolazione *Bounce*.

Oltre alle animazioni applicate ad un oggetto, *Blender* prevede l'utilizzo di armature (già citate nel Capitolo 6.4.1) applicabili a texture per muovere parzialmente l'oggetto secondo la struttura dell'armatura che può essere creata da zero ed applicata alla texture scelta, oppure si possono sfruttare armature umanoidi fornite in partenza.

Le armature possono sia utilizzare la *forward kinematic*, che prevede il posizionamento di ogni singolo osso dell'armatura, che la *inverted kinematic*, con la quale basta posizionare delle speciali ossa di controllo poste alla fine di ogni arto per condizionare, tramite un insieme di *constraint* (vincoli) regolabili, le posizioni e rotazioni di ogni osso dell'arto.

Si può inoltre deformare una texture tramite *Shape Keys*, ovvero creare differenti forme assumibili dalla mesh scelta interpolabili tra loro nel tempo.

Quest'ultima è una tecnica molto utile poiché permette la deformazione di una singola parte della texture gestibile tramite modellazione in *edit mode*, ma, nonostante la sua indubbia utilità non è stata usata per questo progetto in quanto la gestione su *Unity* di questo tipo di animazione risulta essere particolarmente macchinoso.

### Il supporto alle simulazioni fisiche

*Blender* dispone di un pannello dedicato alle simulazioni fisiche di *Fumo*, *Fuoco* (Figura 7.7), *Fluidi*, *Collisioni* e *Campi di forza*.

Ognuna di queste diverse simulazioni dispone di vari parametri regolabili per adattare la simulazione all'effetto visivo ricercato; tra i vari la densità del liquido o dell'aria, la temperatura del fuoco, l'elasticità di un corpo rigido eccetera. Una volta settati i parametri è possibile salvare a parte il risultato della simulazione tramite il comando *Bake* per snellire i successivi tempi di rendering.

Per la creazione di queste simulazioni si utilizzano sempre un dominio, ovvero una zona in cui viene effettivamente calcolata la simulazione; la corretta impostazione del dominio è importante per snellire i tempi di calcolo evitando di calcolare le interazioni per zone non visibili della scena e dare effetti particolari alla simulazione scegliendo la quantità di passi di rendering.



Figura 7.7: Una semplice torcia ottenuta simulando la fisica di un fuoco in un ambiente a pressione, umidità e temperatura comuni.

#### Gestione dei *Particles System*

Su *Blender* è possibile selezionare un oggetto e utilizzarlo come emettitore di particelle; le particelle possono essere texture più o meno complesse, gruppi di oggetti o *billboards* e di queste si può decidere il tempo di vita, la velocità di emissione, il numero, la dimensione nonché modificare impostazioni per randomizzare i parametri della singola particella.

Oltre a questo anche le particelle possono essere impostate per subire gli effetti delle simulazioni fisiche seguendo parametri preimpostati ad esempio per la fisica Newtoniana o per una più efficiente simulazione dei liquidi.

I sistemi particellari sono usati per conferire realistica ad esplosioni, fuochi, per creare effetti atmosferici o simulare pulviscolo. Un altro metodo per utilizzare i sistemi particellari è impostarli come generatori di "*hair*" per simulare, appunto, capelli e pellicce, ma non solo, possono anche essere usati per creare una fitta vegetazione partendo da un piccolo numero di oggetti "albero" che vengono ripetuti e randomizzati in rotazione e dimensione.



Figura 7.8: Pioggia di cristalli luminosi ottenuta con un emettitore di particelle e un piano semi trasparente perpendicolare alla vista per ottenere l'effetto *glow*.

## Gestione dei materiali

Con *Blender* è possibile utilizzare un complesso **editor a nodi per la creazione dei materiali** capace di generare effetti estremamente realistici tenendo conto di riflessi, trasparenze, rifrazioni ed effetti volumetrici partendo da una lista di *shader* combinabili tra loro: *transparent*, *hair*, *volume absorption*, *glossy*, *glass*, *diffuse*, *subsurface scattering*, *emission*, *toon*, *velvet*, *translucent*, *mix shader*, *ambient occlusion*, *volume scatter*, *background*, *holdout*, *refraction* e *anisotropic*.

I nodi dell'editor sono raggruppati in varie categorie che comprendono: *converter*, *vector*, *color*, *texture*, *shader*, *output* ed *input*.

Le particolarità intrinseche nell'utilizzo degli *shader* e dei nodi non viene discussa in modo approfondito, in quanto non mi sono occupato in prima persona della creazione dei materiali per lo sviluppo della nostra app in VR.

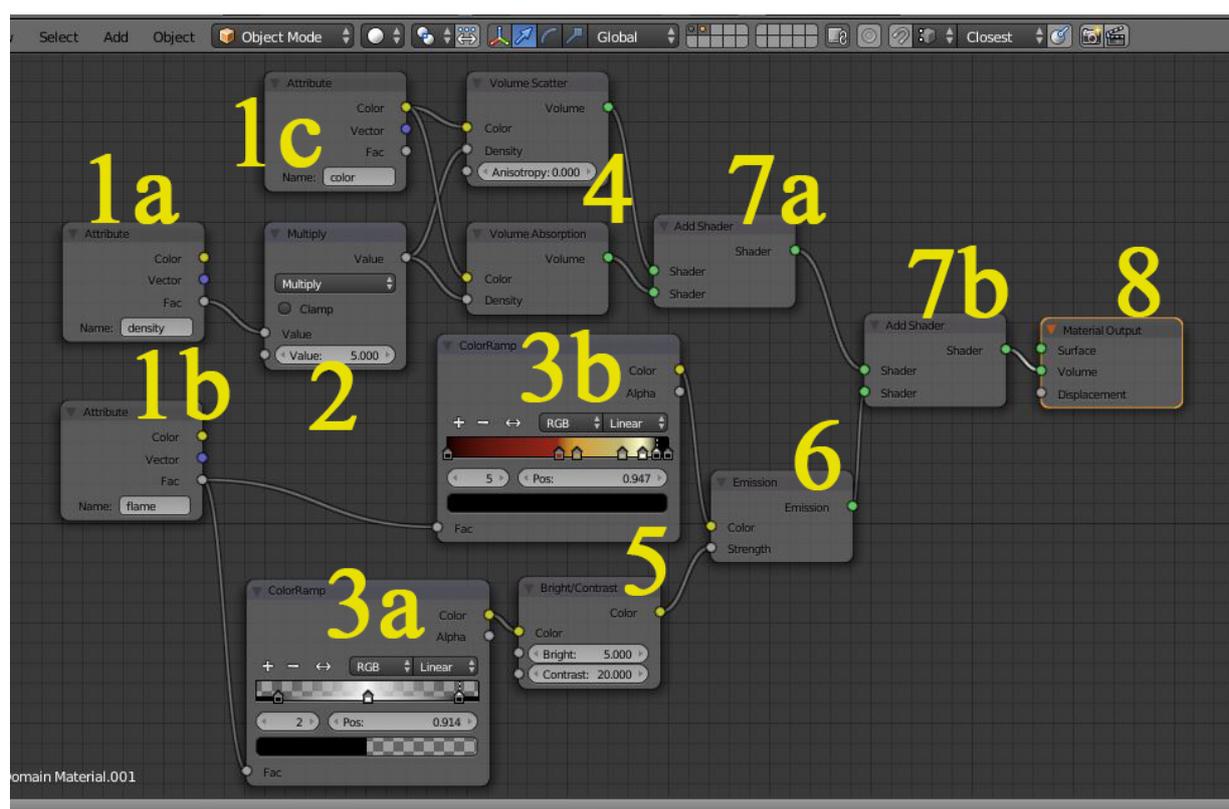


Figura 7.9: L'esempio del complesso materiale per il fuoco visto nella Figura 7.7, ottenuto con l'editor a nodi.

### 7.3. I SOFTWARE UTILIZZATI

---

Per capire meglio come funziona l'editor a nodi per i materiali di *Blender*, vediamo nel dettaglio i singoli nodi della Figura 7.9:

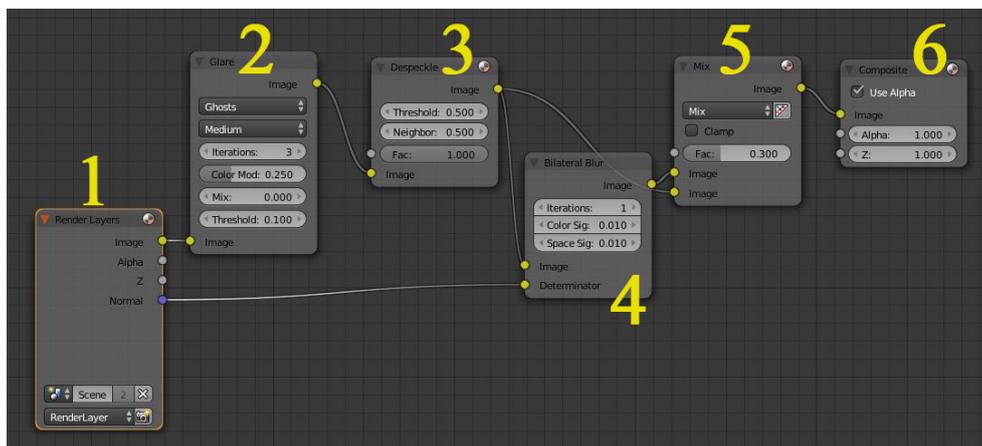
- 1 a, b e c sono attributi relativi alla simulazione fisica del fuoco;
- 2 *Multiply*, crea una maschera mirata a coprire determinati difetti;
- 3 *Color Ramp* (a e b), sono nodi usati per gestire le trasparenze e le diverse gradiazioni di colore;
- 4 *Volume Scatter* e *Absorption*, servono per calcolare gli effetti di luce più complessi, come la rifrazione;
- 5 *Bright/Contrast*, gestisce luminosità e contrasto;
- 6 *Emission*, gestisce l'emissione della luce;
- 7 *Add Shader* (a e b), creano una combinazione degli shader precedenti;
- 8 crea il materiale.

#### Post produzione

Con lo stesso sistema a nodi presente per la creazione dei materiali, *Blender* offre la possibilità di introdurre filtri grafici e modifiche di vario tipo alla scena una volta *renderrizzata* (Figura 7.10).

Partendo da un nodo di input contenente l'immagine vi si possono applicare: **distorsioni** che vanno da semplici rotazioni o scalamenti all'immagine fino a più complessi *displacement*; **matte** (o maschere) per nascondere alcune parti scelte sia dando la forma alla maschera, oppure identificando un parametro di selezione come ad esempio la luminanza o il colore; **color correction** con possibilità di modificare le curve RGB, saturazione, intensità e gamma con l'opzione di usare vari filtri matematici per combinare tra loro più effetti con intensità diverse; **filtri particolari** come il *blur*, suddiviso in varie tipologie, il *glare*, anche qui disponibile in più tipologie, l'effetto *sun beams* per simulare raggi solari, il *defocus* e il *pixelate*.

### 7.3. I SOFTWARE UTILIZZATI



(a) *Semplice utilizzo dell'editor a nodi per la post produzione di un rendering: 1 corrisponde al layer su cui applicare gli effetti, 2 applica un effetto di glow, 3 diminuisce il rumore nelle zone in cui è più visibile, 4 applica un effetto di blur, 5 applica un mixing pesato tra due immagini e 6 crea l'immagine.*



(b) *Particolare effetto luminoso ottenuto grazie alla post produzione.*

Figura 7.10: Un editor a nodi, (a), e il corrispondente risultato finale del rendering, (b).

## 7.3. I SOFTWARE UTILIZZATI

Anche in questo caso non scendo nel dettaglio poiché la post produzione inserita in *The Villa*, limitata comunque allo scuire leggermente alcuni colori, non è stata curata da me in prima persona.

### *Game Engine*

*Blender* comprende anche un proprio *Game Engine* attraverso cui si possono creare applicazioni e videogame tramite una semplice editor *Drag & Drop* chiamato **Logic Editor**, che funziona in tre step: si seleziona l'oggetto a cui va applicata la logica e si sceglie un **Sensor**, elemento che definisce quando deve avvenire l'azione; un **Controller**, che utilizza i comandi logici di And, Or, Xor... per controllare altri eventuali **Sensor**; e infine si dichiara un **Actuator**, che identifica l'azione da fare quando la condizione dei **Sensor** è vera. Questo semplice sistema permette di realizzare la logica che gestisce movimenti, collisioni, calcoli punteggi, modifiche alla UI, passaggi tra diverse schermate e attivazione di animazioni.

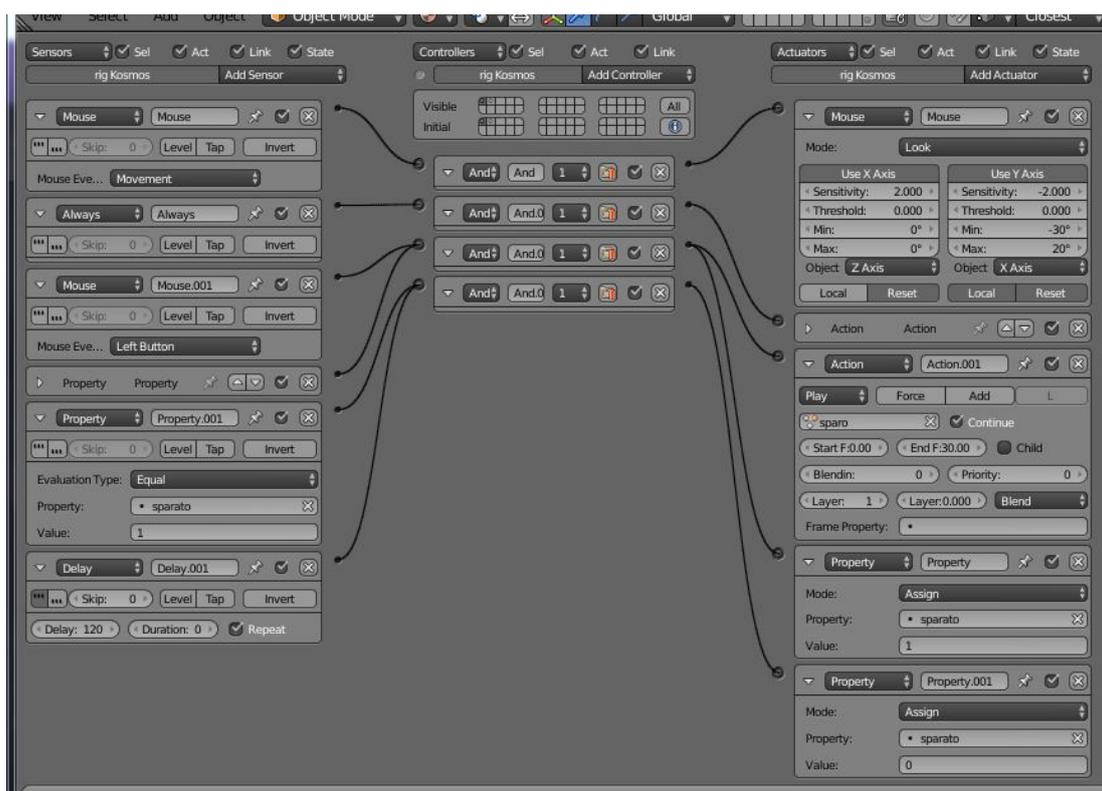


Figura 7.11: Esempio di programmazione a nodi del *Game Engine*, *Blender Game*, per gestire i controlli di un personaggio in una semplice applicazione.

### 7.3. I SOFTWARE UTILIZZATI

---

Pur se molto meno completo e profondo rispetto ad altri strumenti per la creazione di un videogame, è comunque possibile arrivare alla creazione di un prodotto finito ed eseguibile, ma vista la sua scarsa malleabilità è stato preferito l'utilizzo del più noto *Unity 3D* come *Game Engine* per il nostro progetto.

#### *Texturing*

Oltre a permettere di *texturizzare* un oggetto con immagini preparate con programmi di editing più complessi come *Adobe Photoshop*, *Blender* permette di disegnare le nostre texture direttamente all'interno del programma tramite l'apposita schermata *texture paint*. Ogni texture può essere letta come diffuse, per il colore, come alpha, per le trasparenze, come *bump map* per rendere le profondità oppure come *reflection map* per la resa particolareggiata dei riflessi. Si può ovviamente usare le texture come maschere per ottenere effetti particolari per applicare un dato *shader* solo su una data parte della texture.



(a)



(b)

Figura 7.12: La sottile, ma importante differenza tra una semplice texture applicata per il colore, (a), e l'utilizzo di texture per colore, *bump map* e *reflection map*, (b).

### 7.3. I SOFTWARE UTILIZZATI

---

Le texture possono essere mappate sulle superfici selezionate sfruttando vari metodi per avere una mappatura omogenea senza dover mappare singolarmente ogni poligono dell'oggetto, ma per mesh complesse è sempre ideale andare ad agire manualmente sulla mappatura.

#### *Rendering*

*Blender* dispone di due diversi motori di rendering. Il primo, *Blender render*, è un motore interno molto versatile e più semplice nelle impostazioni, utile per creare soggetti non fotorealistici specialmente per le ombre molto nette che crea; il secondo, disponibile dalla versione 2.61 di *Blender* è chiamato *Cycles*, più pesante e complesso del precedente, ma migliore per cercare di raggiungere risultati complessi che puntino al fotorealismo. La maggiore realistica di *Cycles* è data dal suo essere *unbiased*, ovvero non affetto dall'errore, ovvero privo di errore sistematico, permettendogli di creare effetti di luce ed ombra e materiali volumetrici molto complessi per una migliore resa visiva.



(a)



(b)

Figura 7.13: Stessa identica scena renderizzata con *Cycles Render*, (a), e con *Blender Render*, (b).

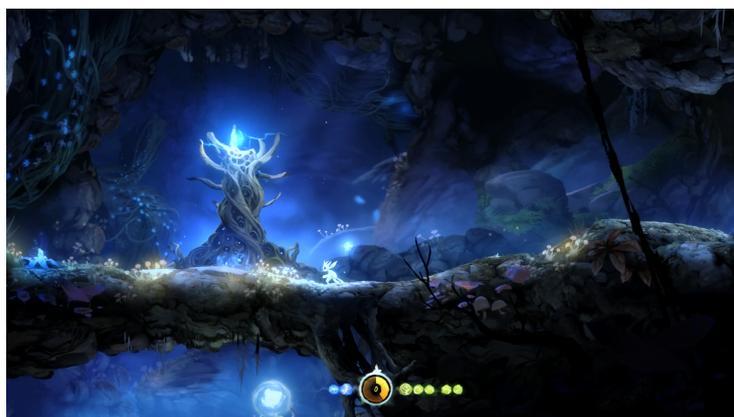
### 7.3.2 *Unity 3D*: le caratteristiche che ne hanno decretato la scelta

Come *Game Engine* si è optato per l'utilizzo di *Unity*, molto usato per lo sviluppo dei videogames 3D o 2D e, nelle sue più recenti versioni, supporta lo sviluppo di applicazioni in realtà aumentata e realtà virtuale, proponendo pacchetti di contenuti dedicati.

Questo motore è stato visto per la prima volta nel 2005 come esclusiva per i sistemi Mac, ma poi diffuso su altre piattaforme fino a raggiungerne più di 25 diventando uno dei *game engine* più usati nell'industria, anche per titoli di fama mondiale come *The forest* o *Ori and the Blind Forest*.



(a) Schermata di gioco di *The Forest*, gioco per PC.



(b) Schermata di gioco di *Ori and the Blind Forest*, gioco per XboxOne.

Figura 7.14: Esempi di videogames sviluppati con *Unity*.

Segue una breve panoramica sulle principali caratteristiche di *Unity* che ci hanno spinto a sceglierlo per la fase di sviluppo della nostra applicazione VR.

### Tipologia di gioco

*Unity* supporta sia lo sviluppo di giochi 2D che 3D proponendo differenze sostanziali nelle impostazioni dei due tipi di progetto facilitandone lo sviluppo.

Inoltre, come già detto, nelle sue più recenti versioni supporta lo sviluppo di giochi per la VR con una serie di *assets* dedicati ed i modelli ed animazioni creati in *Blender* sono facilmente esportabili ed utilizzabili all'interno di *Unity*.

Per la nostra applicazione si è ovviamente scelto un modello di gioco in 3D, sia per permettere di mettere in atto le meccaniche di gioco pensate, sia per sfruttare a fondo la sensazione di immersione ottenibile con i visori VR.

### Scripting

*Unity* supporta i linguaggi **C#** e **Javascript**, precedentemente supportava anche Boo e UnityScripting ormai entrambi deprecati.

Per la scrittura del codice, *Unity* ha integrato un ambiente di sviluppo (IDE) open source chiamato *MonoDevelop*. I file scritti possono essere applicati agli oggetti in *Unity* mediante il semplice paradigma del *Drag & Drop*.

Per il nostro gioco si è scelto il **C#** come linguaggio, semplicemente perché quello meglio conosciuto nel team di sviluppo.

### Texture mapping

*Unity* permette diversi tipi di compressione texture e *setting* di risoluzione a seconda del dispositivo su cui si andrà ad usare l'applicazione creata.

Supporta le texture per *bump mapping*, *reflection mapping*, *screen space ambient occlusion* e *shadow map* per le ombre dinamiche.

Le texture usate in *The Villa* sono *Atlas* creati in *Photoshop* e mappati accuratamente in *Blender*, poiché mette a disposizione un editor più *user friendly* e completo per l'operazione di mappatura; le mappature UV create vengono esportate in *Unity* insieme all'oggetto in automatico.

### Supporto alle piattaforme

Come anticipato *Unity* è supportato da molte piattaforme per permettere il più ampio pool di scelta agli sviluppatori: *Android*, *Android TV*, *Facebook Gameroom*, *Fire OS*, *Gear VR*, *Google Cardboard*, *Google Daydream*, *HTC Vive*, *iOS*, *Linux*, *macOS*, *Microsoft HoloLens*, *Nintendo 3DS family*, *Nintendo Switch*, *Oculus Rift*, *PlayStation 4*, *PlayStation*

### 7.3. I SOFTWARE UTILIZZATI

---

*Vita, PlayStation VR, Samsung Smart TV, Tizen, tvOS, WebGL, Wii U, Windows, Windows Phone, Windows Store, and Xbox One.*

Oltre che per il supporto a *Gear VR*, si è scelto *Unity* anche pensando a possibili conversioni future per una o più delle altre piattaforme supportate.

#### **Personalizzazione della *GUI***

Dalla versione di *Unity 4.3*, pubblicata nel 12 novembre 2013, è stata aggiunto un *tool-kit* decisamente profondo per la personalizzazione della *GUI* in modo da non dover usare un programma esterno dedicato per le modifiche all'interfaccia; c'è la possibilità di inserire testi, campi per scrivere, *slider*, *scrollbar*, immagini, icone e vari tipi di contatori. Ognuno di questi elementi può essere facilmente raggiunto via codice per modificarlo in seguito ad un'azione del giocatore.

Per il nostro gioco, proprio per mantenere il massimo dell'immersione si è deciso di non proporre alcun indicatore a schermo se non per il menu principale che funge anche da tutorial.

#### **Possibilità di *debugging***

*Unity* permette di provare l'applicazione durante il suo sviluppo per vedere attraverso la visuale dell'ipotetico giocatore.

Questa possibilità è estremamente utile per rilevare eventuali difetti; è anche possibile mettere in pausa l'applicazione e navigare attraverso il mondo di gioco precedentemente stoppato per analizzare la scena comodamente.

#### **Gestione audio**

È possibile inserire sorgenti audio e piazzarle nella scena 3D; di ogni sorgente si può scegliere la priorità sulle altre, l'eventuale riproduzione in loop un file audio e se mutarla ad un dato evento.

Inoltre sono regolabili tutti i parametri che conferiscono l'identità al suono di una sorgente audio come il volume e il pitch. Tutti questi parametri sono poi modificabili in game via codice ponendo condizioni di modifica attivabili dall'utente.

Essendo il suono una delle componenti più utili per generare ansia nel giocatore, il posizionamento delle sorgenti audio è stato calcolato con grande cura.

### Gestione delle luci

*Unity* supporta l'utilizzo di vari tipi di sorgente di luce: puntiforme, direzionale, ad area ed a cono.

Ognuna di queste illumina gli oggetti nella scena in un spazio differente: la prima illumina a 360° intorno a se, la seconda illumina l'intera scena lungo una direzione selezionata, la terza illumina perpendicolarmente rispetto ad un area di dimensione variabile e l'ultima ha un effetto simile ad un faretto (appunto *spotlight*).

Tutte loro hanno diversi parametri modificabili come colore, intensità, intensità delle ombre e dei rimbalzi e distanza di *falloff*.

È fondamentale ricordare che si può scegliere se precalcolare l'effetto delle sorgenti di luce sulla scena, oppure lasciare il calcolo in real time per avere ombre dinamiche e maggior realismo, ma anche una maggior possibilità di generare errori visibili.

### Gestione collisioni

*Unity* può gestire le collisioni tra i vari oggetti di gioco assegnando ad ognuno di loro un ***box collider*** che va a definire lo spazio che occupano, questo si muove insieme alla mesh vera e propria e rileva quando un altro oggetto lo incontra; a quel punto si crea un "evento" che può portare all'effetto voluto via codice.

Per la gestione della fisica nelle collisioni si usa il ***rigidbody***, ovvero un componente applicabile all'oggetto che lo dota di una massa, lo fa reagire ai campi di forza come la gravità, calcola una reazione agli scontri con altri ***rigidbody*** e, più in generale, ne determina i comportamenti legati alle simulazioni fisiche.

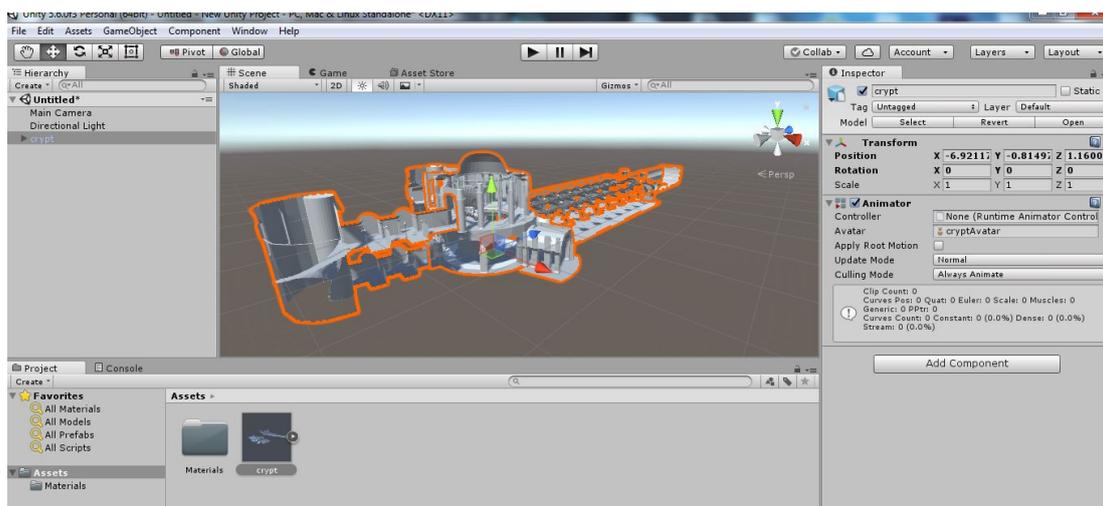


Figura 7.15: Una delle zone di *The Cathedral: Allison's Diary* all'interno dell'ambiente di lavoro *Unity*.

### 7.3. I SOFTWARE UTILIZZATI

---

Oltre a queste caratteristiche specifiche anche *Unity* permette la modellazione di mesh partendo da una serie di primitive geometriche come cubi, sfere o cilindri, ma si è preferito delegare ad un programma esterno questo compito, nel nostro caso *Blender*, perché, per quanto funzionale, la modellazione in *Unity* risulta meno intuitiva e profonda per i progetti 3D, specialmente per quanto riguarda le animazioni.

## Capitolo 8

# Lo sviluppo di *The Villa: Allison's diary* presso KR Games

Per elaborare la mia tesi ho lavorato per un periodo di sei mesi, dal primo giugno al trenta novembre 2017, presso la casa di sviluppo indipendente KR Games che, appunto, si occupa di sviluppare esperienze per realtà virtuale, come *The Villa: Allison's diary* che andremo ad analizzare nel dettaglio, e videogame per il mercato mobile, come il recente *Pinj*, disponibile da dicembre 2017 su dispositivi Android.

Come modellatore e animatore 3D, è stato mio compito creare passo passo i personaggi, gli ambienti, gli oggetti utilizzabili dal giocatore e le relative animazioni seguendo le direttive decise in fase di creazione dei concept di gioco proprio per *The Villa*: un'esperienza horror pubblicata per *Gear VR*.

### 8.1 La trama e l'atmosfera

*The Villa* è un videogioco che può essere ricondotto al recente genere dei *walking simulator*, ovvero avventure che puntano molto sull'immersione dell'utente nel mondo di gioco e nella sua linea narrativa, ma prive di combattimenti o altre meccaniche di gameplay più complesse, a tinte horror classico.

La scelta del tema horror risiede proprio nella già discussa peculiarità dello spettatore che risulta più portato a provare alcune sensazioni rispetto ad altre, come appunto, la paura e i dispositivi VR, con l'alto tasso di immersione che permettono, risultano particolarmente adatti a questo genere di esperienze; non a caso oltre alle molte esperienze horror offerte da software house indipendenti, anche le grandi case di produzione che stanno iniziando a sperimentare con la VR puntano molto sul questo genere con i loro prodotti di punta.

## 8.1. LA TRAMA E L'ATMOSFERA

---

Come primo passo del mio percorso lavorativo mi è stata spiegata la trama di questa esperienza VR in modo da poter meglio capire l'atmosfera che si stava cercando di creare e che doveva essere trasmessa anche attraverso i contenuti 3D da me creati: la saga di **Allison's Diary** è in realtà una trilogia iniziata con *The Hospital* (Figura 8.1) nel 2015, proseguita con *The Cathedral* (Figura 8.2) nel 2016 e terminata a novembre del 2017 con lo stesso *The Villa* (Figura 8.3).

Segue una rapida sinopsi delle tre esperienze tratte direttamente dal sito ufficiale di *Oculus Rift*, [Oculus.com](http://Oculus.com), su cui sono state pubblicate.

### ***The Hospital***

"Nel 1956, una bambina di nove anni muore nell'ospedale psichiatrico Santa Benedetta Martire. Il suo nome era Allison. Prima di morire, Allison scrisse un diario personale, che si dice contenga la verità sul suo passato e la vita raccapricciante in ospedale. Il diario non fu mai trovato. Pochi anni più tardi, l'ospedale fu chiuso, e diventò un luogo terrificante dimenticato da Dio. Tu sei Guglielmo Carter, un impavido reporter interessato a trovare il diario di Allison e rivelare i suoi segreti nascosti".

Tratto da fatti reali, "The Hospital: Allison's Diary" è una nuova esperienza horror di realtà virtuale, che pone il giocatore da solo in un ospedale abbandonato alla ricerca dell'ignoto, con il solo aiuto di una torcia.

Vivrai la paura vera in un modo che mai avevi pensato possibile. L'intero ambiente è libero di essere visitato, mentre jump scares ed effetti cinematografici ti faranno letteralmente dimenticare il mondo reale che ti circonda.



Figura 8.1: Una schermata di gioco di *The Hospital: Allison's Diary*.

## 8.1. LA TRAMA E L'ATMOSFERA

---

### *The Cathedral*

"Nel 1956, Allison morì nell'ospedale psichiatrico di Santa Benedetta Martire. Il diario che scrisse conteneva la terribile verità riguardo a come essa uccise i suoi genitori. Guglielmo Carter, un impavido reporter, sparì subito dopo aver trovato il diario".

Dove sono tenuti i cadaveri dei genitori di Allison?  
Cosa successe a Guglielmo quella notte?

Due anni dopo i fatti di "The Hospital", sei interessato a visitare il posto dove Rachel ed Alfredo dovrebbero essere stati sepolti.

"The Cathedral" è il secondo capitolo della saga horror Allison's Diary. Nella sacra cattedrale di Sant'Andrea c'è qualcosa di spaventoso che aspetta di essere scoperto. I funerali non sono la fine: sono solo l'inizio.

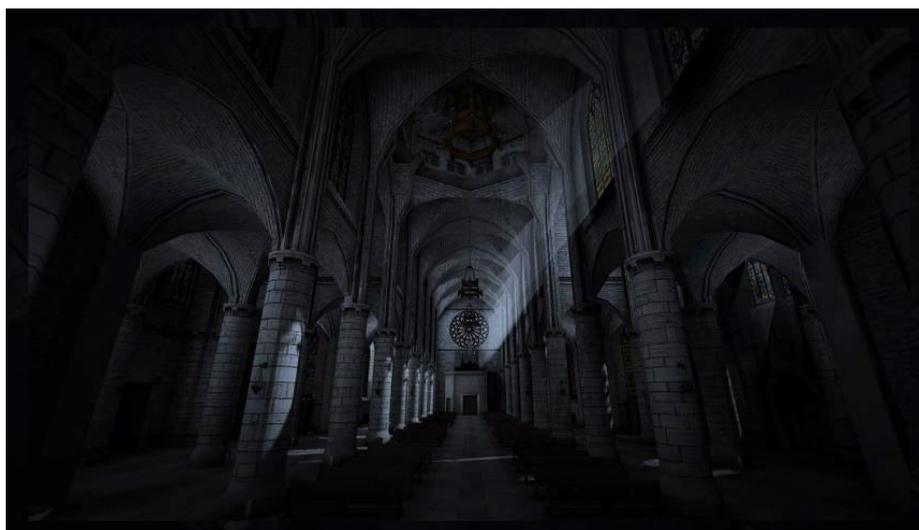


Figura 8.2: Una schermata di gioco di *The Cathedral: Allison's Diary*.

## 8.1. LA TRAMA E L'ATMOSFERA

---

### *The Villa*

Una villa, una routine quotidiana ed una famiglia apparentemente felice. Ecco dove tutto cominciò.

Dopo "The Hospital" e "The Cathedral", "The Villa" è l'ultimo capitolo della trilogia horror Allison's Diary, dove rivivrai, attraverso flashback ed effetti cinematografici immersivi, la spaventosa notte in cui Allison uccise i suoi genitori nel 1956.

Jump scares terrificanti ed un ambiente super realistico rendono "The Villa" una delle migliori esperienze horror per Gear VR.



Figura 8.3: Una schermata di gioco di *The Villa: Allison's Diary*.

Come analizzato in precedenza, ciò che rende unica un'esperienza in VR è l'immersione totale del giocatore nel mondo virtuale, e, poiché l'esplorazione della villa doveva essere il fulcro dell'esperienza, è stato dedicato parecchio tempo alla ricerca di reference dettagliate da usare come spunto per creare un'ambientazione inquietante ed al tempo stesso realistica, in linea con il periodo storico in cui la villa venne idealmente costruita, ovvero agli inizi del novecento.

Per la planimetria della villa si è infatti preso spunto da un edificio realmente esistente, ovvero Villa de Vecchi, a Cortenova, in provincia di Lecco (Figura 8.4), a cui sono state applicate in seguito varie modifiche per adattare la disposizione delle stanze e degli infissi al gameplay.

## 8.1. LA TRAMA E L'ATMOSFERA

---



(a) *Reference per gli esterni.*



(b) *Reference per gli interni.*

Figura 8.4: Due delle immagini di reference usate per la creazione della Villa, ambientazione del videogioco.

### 8.2 Creazione degli ambienti

Passiamo ad analizzare la creazione del suddetto ambiente di gioco di gioco prendendo in esame le varie fasi che hanno visto la mia collaborazione.

**Fase 1** - Come prima cosa sono stati stabiliti dei **vincoli sia spaziali che di numero di poligoni**.

Per i primi si è pensato di creare stanze di dimensioni diverse, a seconda della funzione a cui erano idealmente adibite, pur nel rispetto delle reali dimensioni dei locali di quel genere di edifici; partendo dal presupposto che il modello dell'intero interno della villa non avrebbe dovuto sfiorare i diecimila poligoni, arredamento escluso, abbiamo stanze più piccole e semplici come il bagno e più grandi e con decori maggiormente complessi, come l'atrio.

Per il conteggio dei poligoni si è deciso di limitarsi al migliaio per ogni stanza, intesa come l'insieme di struttura e infissi, ma esclusi mobilio ed arredo, cercando di distribuire il maggior numero di vertici, e quindi dando miglior qualità visiva, nei punti dove si sapeva si sarebbe soffermato lo sguardo del giocatore, come una finestra che dà sull'esterno, la zona in prossimità di un oggetto da raccogliere o i dettagli delle porte che il giocatore deve aprire.

**Fase 2** - Per ottimizzare i tempi in fase di modellazione si è deciso di **modularizzare le stanze**, ovvero crearle usando una *pool* di possibili elementi fissi modellati e texturizzati in precedenza.

Ad esempio sono stati creati 4 diversi muri con una singola apertura per una finestra, due diversi muri con aperture per due finestre, 4 diversi tipi di soffitto, 4 diversi pavimenti e così via.

Ogni piano della villa aveva a disposizione una diversa *pool* di oggetti in quanto le stanze dei diversi piani risultavano adibite a compiti diversi; ad esempio il pian terreno che comprendeva stanze di ritrovo come l'atrio o il salotto ha decori più vistosi su infissi e soffitto, mentre ai piani superiori, dove abbiamo le camere private, la *pool* di oggetti comprende elementi più spartani nel tipo di decorazione.

Ogni singolo oggetto della *pool* poteva poi essere leggermente modificato nel modello stesso o nella texture usata a seconda delle necessità, ma ciò ha consentito comunque una netta diminuzione della mole di lavoro senza rinunciare alla qualità.



Figura 8.5: Modello di una stanza interna della Villa come appare in *Blender* con le sole texture diffuse applicate.

**Fase 3** - Il passo seguente è anche il vero e proprio inizio della fase di modellazione: si passa a **creare i singoli oggetti delle varie *pool*** iniziando dal pian terreno, primo ambiente visitabile del gioco, e proseguendo verso l'alto.

Ogni singolo oggetto è stato creato da zero con il software *Blender 2.76* e, essendo muri e infissi formati da geometrie semplici, si è sfruttato il metodo di modellazione "a spostamento di elementi"<sup>1</sup> per creare un modello a basso dettaglio, poi, utilizzando la tecnica a "definizione progressiva"<sup>2</sup> si va ad aumentare il numero di poligoni e quindi di dettagli visibili, fino ad avere il livello di qualità richiesto.

Realizzato il modello del singolo oggetto lo si mappa su apposite *texture atlas* (Figura 8.6 (a) e (b)), una per ogni stanza, con particolare attenzione alle proporzioni e agli angoli, di modo che, qualora, ad esempio, si fosse deciso di inserire crepe o dettagli sulla texture di un muro, questi avrebbero seguito fedelmente la geometria del modello.

Per questo passaggio si è usata la tecnica "*unwrap from view*" partendo dalle viste ortogonali in modo da avere sempre corrispondenza fra proporzioni dell'oggetto e della sua mappatura. Questa scelta di delegare alle texture la rappresentazione dei dettagli più minuti è stata presa per poter rispettare il vincolo di conteggio dei poligoni analizzato in precedenza.

---

<sup>1</sup>Capitolo 5.1.1

<sup>2</sup>Capitolo 5.1.4

## 8.2. CREAZIONE DEGLI AMBIENTI

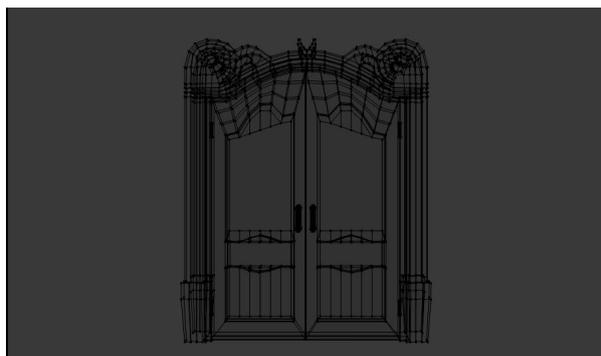
Ogni oggetto creato deve rispettare rigorosamente la nomenclatura in *lower camel case* del tipo "elemento+Stanza+Numerazione", ottenendo ad esempio "wallWindowC01.001".



(a) Oggetti vari in wireframe.



(b) Gli oggetti della figura (a) con l'applicazione delle texture atlas dedicate.



(c) Particolare: porta per le stanze modellata in Blender (wireframe).



(d) Particolare: porta per le stanze modellata in Blender (wireframe).

Figura 8.6: Alcuni degli oggetti usati come componenti per creare le varie stanze della Villa.

**Fase 4** - Per avere oggetti facilmente utilizzabili vengono loro applicate le trasformazioni eseguite in modo da avere come risultato *location* (0,0,0), *rotation* (0,0,0) e *scale* (1,1,1) con l'origine dell'oggetto posizionato in concomitanza al centro della geometria.

L'unica eccezione a questo principio sono le porte (Figura 8.7), che, essendo animate tramite uno script che le ruota attorno l'asse z nel verso che va da x verso y, è stato necessario di volta in volta ruotare l'oggetto di modo che le sue normali permettessero il corretto funzionamento dell'animazione coerentemente alla posizione dei cardini della porta.

Per un maggiore tasso di realismo, infatti, ogni porta è stata creata in modo da aprirsi in un solo verso, ruotando lungo i cardini su cui giace il centro dell'oggetto.

## 8.2. CREAZIONE DEGLI AMBIENTI

---

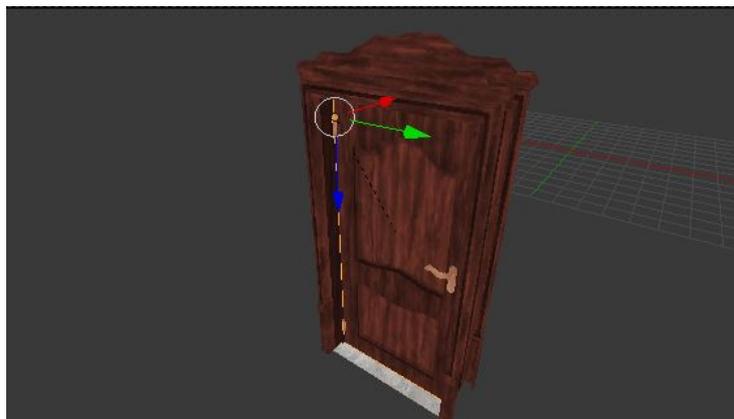


Figura 8.7: Porta con centro del modello situato in corrispondenza dei cardini e avente le normali orientate in modo da potersi aprire ruotando sull'asse z passante per i cardini stessi.

**Fase 5** - Inizia qui la **creazione delle singole stanze**, fatte assemblando tra loro i blocchi modellati in precedenza, facendo attenzione a seguire le misure spaziali definite nella planimetria sia per il posizionamento di eventuali finestre o porte, che per le dimensioni della stanza stessa.

Per concludere la stanza, ogni suo "blocco" viene inserito in una gerarchia (Figura 8.8) formata da: un oggetto padre *empty* chiamato "villaFloorA", "villaFloorB" o "villaFloorC", uno per ogni piano dell'edificio; a questi corrispondono tanti oggetti *empty* quante sono le stanze del dato piano, chiamati "room+nome del piano+numerazione crescente con partenza da 01" iniziando a nominarle partendo dalla stanza d'angolo frontale sinistro della struttura; a loro volta questi oggetti avranno come figli tutti gli elementi della stanza indicata ad eccezione degli infissi, che, per comodità, risultano imparentati con un oggetto separato per ogni piano chiamato "doorFrames+nome del piano".

Questo passaggio oltre ad essere molto utile per una corretta gestione della pulizia del file, risulta fondamentale in fase di scripting, poiché molte righe di codice fanno riferimento ad eventuali oggetti figli che devono quindi essere imparentati correttamente.

Una particolare menzione meritano le porte, create in modo che l'infisso, la porta stessa e la maniglia fossero tre elementi separati, ma gerarchicamente legati in modo da poter animare correttamente l'apertura della porta e la rotazione della maniglia.

## 8.2. CREAZIONE DEGLI AMBIENTI



(a) Gerarchia degli elementi di un singolo piano della Villa.

(b) Gerarchia degli elementi che costituiscono una porta.

Figura 8.8: Gerarchie degli elementi in *Blender*.

- Fase 6** - Alla creazione di ogni stanza parte una **fase di testing su *Unity*** volta a confermare che il player possa effettivamente muoversi nello spazio creato, interagire con gli oggetti scelti ed è stata utile anche per trovare eventuali criticità nella modellazione di modo da poterla correggere immediatamente, risolvendo il problema per l'assemblaggio delle stanze seguenti.
- Fase 7** - Oltre alle stanze "standard" ci sono state zone particolari che hanno richiesto **modellazione specifica** come le scale interne (Figura 8.9) per cui sono state calcolate accuratamente le alzate dei singoli gradini per garantire sia realistica al modello che un facile utilizzo del giocatore oppure il grande atrio che ha richiesto uno studio più ricercato per le decorazioni in modo da ricreare quella che doveva essere la stanza più sfarzosa dell'intero edificio.



Figura 8.9: Le particolari scale interne della Villa.

**Fase 8** - Creati e texturizzati tutti gli ambienti interni si è passati alla **modellazione dell'esterno della villa** (Figura 8.10).

Qui il processo è stato simile a "glassare una torta", ovvero si è dovuto coprire tutto ciò che è stato modellato e posizionato con uno strato esterno che andasse a simulare la struttura unitaria dell'edificio.

Anche in questo caso si è proseguito seguendo le numerose reference per i dettagli estetici dei decori facendo molta attenzione a sistemare porte e finestre in concomitanza con le loro controparti interne.

Per ragioni di ottimizzazione di tempo e risorse, zone non visibili come il tetto della villa, o il retro della stessa, non sono stati accuratamente modellati per poter impiegare altrove i poligoni così risparmiati.

Per il terreno esterno si è pensato di simulare una sorta di pendio utilizzando un piano a cui è stato applicato una successiva suddivisione in modo da avere un buon numero di facce per rappresentare i dettagli di pendenza che sono stati creati tramite un semplice lavoro di sculpting, sfruttando i vari tipi di pennelli messi a disposizione dal software per scendere nel particolare, soprattutto per le zone in cui il terreno incontra l'abitazione.

Il tutto è stato cinto da un muro con ringhiera che funge da limite per il mondo virtuale esplorabile, facendo attenzione a creare una struttura che sembri realmente non superabile per non rompere la sospensione all'incredulità del giocatore, che, vedendo ad esempio una zona libera da ostacoli sarebbe portato dall'esperienza della vita reale a pensare di poterci passare, ma se questo non avviene a causa di un muro invisibile non giustificato, verrebbe meno l'immersione, concetto che sta alla base di tutta l'esperienza di gioco.



Figura 8.10: Visuale dell'esterno della Villa e del giardino creato per *The Villa*.

- Fase 9** - Completata la creazione di tutti gli ambienti si passa ad **aggiungere tutti i singoli oggetti** che arredano le stanze: letti, armadi, sedie e altri mobili, ovviamente disposti in modo logico e sensato sempre per mantenere coerenza e realismo. Questi oggetti a scopo puramente estetico hanno una *texture atlas* dedicata ed un oggetto padre empty che li controlla tutti per poterli selezionare con facilità.
- Fase 10** - Vengono infine aggiunti tutti quegli **elementi che servono a guidare l'esplorazione** del giocatore come macerie che bloccano un passaggio, un pezzo di scala crollato che obbliga a cercare un altro accesso per i piani superiori oppure assi inchiodate ad una porta che ne prevengono l'apertura. Il posizionamento di questi è stato già deciso a grandi linee durante le precedenti fasi, ma solo vedendo e navigando attraverso l'ambiente finale si è potuto decidere della bontà (o meno) della posizione scelta per determinati oggetti.
- Fase 11** - Per i controlli finali si è provveduto ad **analizzare uno per uno ogni ambiente** ritoccando eventuali piccole problematiche riscontrate, facendo particolare attenzione alla direzione delle normali che, se sbagliate, oltre a creare problemi di illuminazione non rende visibili le facce su *Unity*.
- Fase 12** - Importando su *Unity* i modelli creati, questi mantengono la corretta mappatura eseguita su *Blender* in precedenza e vengono creati automaticamente dei materiali standard aventi il nome della texture su cui tali oggetti sono stati mappati.

L'importazione su *Unity* è stata comunque eseguita più volte a progetto in corso poiché, il passaggio sul game engine, è utile a svelare eventuali errori che non compromettono la resa visiva su *Blender*, ma sono facilmente notabili in questo ambiente di sviluppo.

**Fase 13** - Sempre su *Unity* viene **aggiunta l'illuminazione definitiva dell'ambiente**, creata con diverse fonti di luce, sia diegetiche, come la torcia in dotazione al giocatore, che extradiegetiche per evitare ambienti totalmente bui e lo *skybox*, elemento che va a simulare il cielo e alcuni semplici eventi atmosferici per le scene in esterno, in particolare alcuni lampi.

È stata posta particolare attenzione alla creazione della luce della torcia elettrica in possesso al giocatore che illumina a diversa intensità a seconda di una texture che simula il vetro rovinato della torcia.

Nonostante questo non è stato utilizzato il calcolo real time delle ombre per evitare di appesantire l'applicazione soprattutto in presenza di piccoli oggetti.

## 8.3 Creazione dei personaggi

Oltre alla struttura della villa e del mondo di gioco ho contribuito a creare alcuni personaggi che vediamo nel videogame, in particolare Allison e i suoi genitori.

Nei paragrafi che seguono descriverò il processo di modellazione, rigging e animazione della protagonista, il quale è stato lo stesso seguito per la creazione dei suoi genitori.

### 8.3.1 Modellazione

Allison è una ragazzina di nove anni che, nel corso della storia, scopriamo essere, pur sembrando apparentemente normale per aspetto e comportamento, capace di atti estremamente crudeli, essendo posseduta da un'entità malevola.

Nel dettaglio è una bambina caucasica di 1.52 metri d'altezza, capelli neri lunghi fino alle spalle, occhi del medesimo colore, indossa sempre il suo preferito vestitino celeste e presenta una lieve deformità alla mandibola.

Per Allison si è preso ispirazione dalla classica "bambina inquietante" dei film horror, questo perché avendo a disposizione una singola e breve esperienza in VR è bene fare in modo che i personaggi che appaiano possano essere ricondotti dal giocatore a figure che già conoscono e che fanno parte dell'immaginario collettivo, in modo che si necessiti di meno tempo per far capire il ruolo del personaggio all'interno della narrazione.

Si è quindi cercato di utilizzare un archetipo classico del cinema per massimizzare l'effetto narrativo voluto in poco tempo.

### 8.3. CREAZIONE DEI PERSONAGGI

---

Una versione grezza del modello ancora priva di scheletro mi è stata fornita direttamente da KR Games; partendo da questa ho provveduto ad eseguire varie modifiche volte a migliorare la qualità visiva del personaggio, migliorarne le geometrie per renderlo più comodamente animabile e aumentare la densità poligonale nelle zone che, in movimento, si sarebbero dovute piegare o ruotare, richiedendo quindi un maggior numero di facce per essere visualizzate correttamente (gomiti, collo, spalle, la zona in prossimità della bocca, ecc...).

In particolare il modello, pur essendo di buona fattura, presentava un'eccessiva quantità di dettagli per un'applicazione VR real time.

Come prima cosa ho quindi compiuto un'analisi per decidere **quanti e quali dettagli mantenere**, eliminando il superfluo: i capelli, la definizione delle dita e di alcune pieghe del vestito sono stati semplificati perché poco visibili dal giocatore nell'ambiente di gioco che, puntando ad un'atmosfera horror, sarà ovviamente buio.

Mentre per gli abiti e le dita è bastato semplicemente ricorrere a spostamenti ed eliminazioni di vertici e facce in *edit mode*, per i capelli si è reso necessario ricrearli da zero, utilizzando al posto di una complessa mesh ad alta definizione, un insieme di singole facce 2D disposte "a ciuffi" e poi mappate con la texture voluta, delegando a quest'ultima il compito di simulare i dettagli della capigliatura.

Per quanto concerne i problemi legati all'animazione in corrispondenza dei punti critici elencati prima, mi sono limitato ad aggiungere **edge loops**, ovvero anelli di vertici, in quelle determinate zone mentre ho quasi totalmente eliminato facce non visibili facenti parte delle zone del corpo coperte dal vestito onde evitare possibili problemi di compenetrazione qualora la gestione dei pesi delle ossa dell'armatura si fosse rivelata complessa.

Nel suo stadio finale il modello è composto da 10 876 facce o 21 532 triangoli un buon compromesso per mantenere discreta la qualità di una figura umanoide senza appesantire troppo l'applicazione real time e risulta mappato su un'unica *texture atlas* per il canale diffuse, sempre per questioni di ottimizzazione.

Per la mappatura dei personaggi si è optato per un approccio congiunto sfruttando laddove possibile lo **smart UV mapping**, tecnica che tiene unite nella mappatura le zone effettivamente unite nel modello, procedendo poi a mano utilizzando la *texture atlas* come riferimento.

È importante ricordarsi di mappare ogni faccia, anche quelle non visibili del modello poiché, all'importazione su *Unity*, vengono a crearsi materiali extra per le facce non mappate, minando la pulizia del progetto che deve essere facilmente leggibile e utilizzabile anche da altri.

### 8.3. CREAZIONE DEI PERSONAGGI

---

Discuto a parte i materiali, in quanto non sono stati aggiunti da me in prima persona durante il lavoro al progetto: al completamento di ogni modello, questo veniva testato in game inserendolo nell'ambiente di gioco su *Unity* ed applicandovi alcuni script di prova e visionandolo in realtà virtuale, poiché la resa visiva differisce molto da quella che si ha su di un software per la modellazione 3D.

Verificata la bontà del modello e della disposizione delle texture vengono applicati semplici materiali diffuse con aggiunta di *bump mapping* per far meglio risaltare l'effetto 3D di, ad esempio, pieghe sui vestiti o crepe sui muri, per la maggior parte delle superfici aggiungendo un effetto *glossy* per tutti quegli oggetti vagamente lucidi come possono essere parti in metallo di arredo o finestre.

La semplicità nei materiali è voluta per permettere all'applicazione di non scendere al di sotto dei 60 frame al secondo, fondamentali per una buona esperienza in VR.



Figura 8.11: Versione finale del modello di Allison in posa.

### 8.3.2 Applicazione delle armature

A modello concluso si è dovuto creare un'armatura per rendere possibili le varie animazioni.

Essendo il modello di forma umana si è preferito partire da **Human (Meta-rig)**, un'armatura di default messa a disposizione da *Blender* tramite l'add-on *Rigging: Rigify*, adatta appunto a personaggi umanoidi, piuttosto che creare da zero un'armatura customizzata.

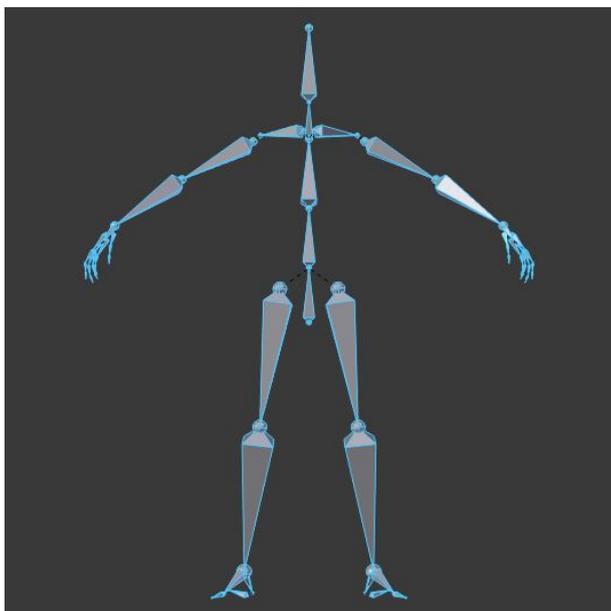


Figura 8.12: Come appare l'armatura *Meta Rig* creata con l'add-on *Rigify*.

L'armatura così creata va poi adattata alle dimensioni del personaggio muovendo le singole ossa in *edit mode* ricordando alcune regole per il corretto funzionamento dopo l'applicazione dell'armatura:

**Regola 1** - Gli arti devono risultare il più diritti possibile in *front view*, ovvero guardando frontalmente il personaggio le ossa devono seguire una linea retta dalla spalla (o anca) fino al polso (o caviglia), e ci deve essere una leggera curva all'altezza di gomiti e ginocchia per indicare in che verso devono poter ruotare.

Stesso ragionamento può essere applicato per l'ossatura delle dita della mano.

**Regola 2** - Le ossa che formano la spina dorsale devono anch'esse essere il più allineate possibile in *front view*, con l'ultimo osso in verticale che deve raggiungere la sommità del capo, essendo delegato ai movimenti della testa.

### 8.3. CREAZIONE DEI PERSONAGGI

---

Va detto che, per semplicità, il meta-rig utilizzato per Allison è stato modificato eliminando le ossa per il controllo facciale, dato che, non dovendo animare scene parlate o primi piani del viso, è possibile ottenere un buon risultato con poche ossa aggiunte manualmente all'armatura: 2 per il controllo degli occhi e 2 per il controllo della bocca, sfruttando poi gli shape keys per i dettagli più minuti.

Posizionate correttamente tutte le ossa, si deve tornare in *object mode* dove, nel pannello relativo all'armatura, si può generare il rig (Figura 8.13), ovvero l'insieme di controllori che andremo ad utilizzare per muovere il personaggio, semplicemente premendo l'apposito comando *Generate*.

L'oggetto **Rig** ottenuto sarà formato da cinque "controllori" principali: "**Torso**", che controlla l'omonima parte del corpo; "**hand.Ik.R**", che controlla la mano destra e, a catena, sfruttando la cinematica inversa analizzata in precedenza, il resto del braccio; "**hand.Ik.L**", che svolge la medesima funzione per il braccio sinistro e infine "**foot.Ik.R**" e "**foot.Ik.L**", che, come per le braccia, controllano le gambe, partendo dal controllo diretto del piede, mediante la cinematica inversa.

Ovviamente il Rig comprende numerosi altri controllori per ogni parte del corpo come testa, collo e dita. Inoltre è possibile andare ad intervenire singolarmente su parti degli arti mossi tramite cinematica inversa qualora l'approssimazione ottenuta muovendo il solo controllore finale dell'arto non dia risultati soddisfacenti, ad esempio sistemando manualmente gomiti o ginocchia tramite cinematica diretta.

A ciascun controllore possono essere modificati vari parametri: vincoli di posizione, rotazione o scalamento, abilitazione della "cinematica inversa" e stretching della mesh, per ottenere sempre risultati il più possibile fedeli a quelli cercati.

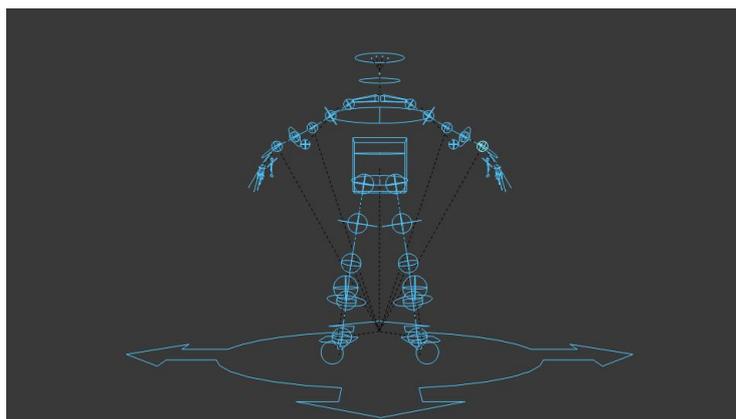


Figura 8.13: L'insieme dei controllori generati tramite l'armatura *Meta Rig*.

### 8.3. CREAZIONE DEI PERSONAGGI

---

Ora basta selezionare il personaggio e l'oggetto Rig appena creato e selezionare dal menu a tendina richiamabile con CTRL+P l'opzione *set parent with automatic weights*.

Così facendo il Rig si applicherà al personaggio con i pesi delle relative ossa calcolato automaticamente a seconda della loro posizione rispetto alla mesh e, se il passaggio precedente di sistemazione delle ossa è stato fatto con precisione, il suddetto calcolo dei pesi risulterà molto preciso, in caso contrario è possibile annullare la creazione del rig e tornare a modificare le posizioni delle ossa dell'armatura iniziale.

Nel particolare caso di Allison, al momento di applicare i pesi, si è verificata una criticità all'altezza delle ginocchia dove, per movimenti ampi delle gambe, il ginocchio risultava compenetrare con il vestito.

In questi casi si rende necessario agire manualmente selezionando il gruppo di vertici relativi all'osso che deve avere più influenza sulla zona del personaggio che presenta la criticità dalla relativa tabella e gli si aggiungono i vertici desiderati selezionandoli in *edit mode* dalla mesh.

Ovviamente può servire anche il processo opposto di rimuovere da un determinato gruppo di vertici alcuni che vengono mossi quando non dovrebbero. I vertici selezionati possono essere aggiunti alla zona di controllo relativa ad un dato osso selezionando anche quanto l'osso va ad influire sulla loro posizione.

Il personaggio così ottenuto è pronto per essere animato a piacere.

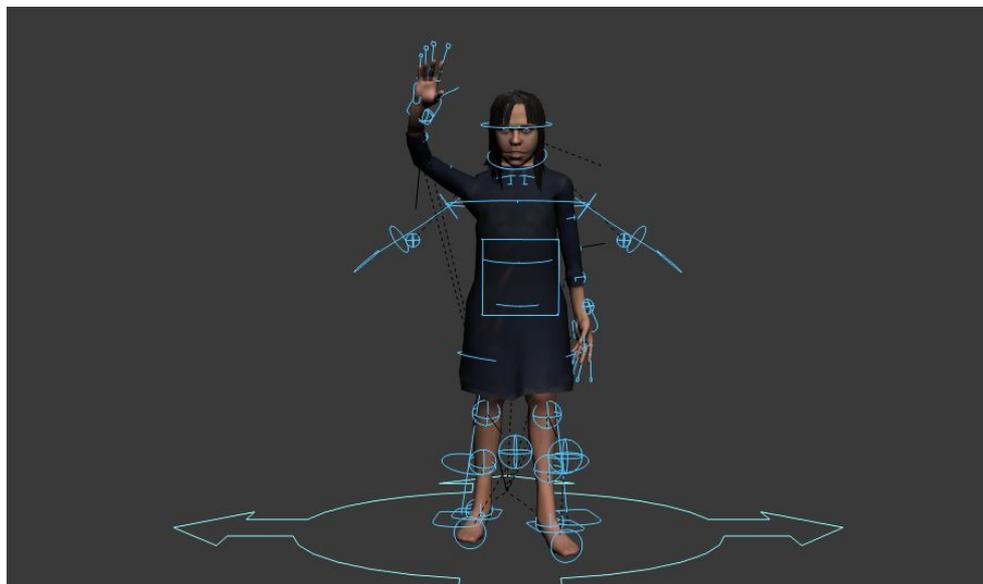


Figura 8.14: Il modello 3D di Allison con i controllori usati per le animazioni.

### 8.3.3 Animazione

Per ognuno dei personaggi di gioco sono state create apposite animazioni che vengono attivate tramite codice quando il giocatore compie una determinata azione, come aprire una determinata porta, voltarsi in una certa direzione o avanzare in un'area.

Queste animazioni sono state create sempre con *Blender* con tecniche leggermente diverse a seconda dei movimenti richiesti, ma sempre utilizzando il rigging creato come illustrato in precedenza e seguendo un pattern di lavoro fisso.

- Fase 1** - Si crea una **nuova azione** nella schermata *Action Editor* del *Dope Sheet* nominandola come mi viene indicato, sempre rispettando la convenzione *Lower Camel Case*, ottenendo nomi come ad esempio "slowWalkAllison".
- Fase 2** - Mi viene comunicato in secondi il tempo che deve prendere la singola animazione, quindi preparo la **posa iniziale**, in corrispondenza del frame 0, e **finale**, il cui frame può essere facilmente ricavato sapendo che a 24 frame corrisponde un secondo di animazione; per settare queste pose del personaggio basta muovere, ruotare o scalare le componenti dell'oggetto Rig in *Pose Mode* e inserire un keyframe rispetto alle trasformazioni che abbiamo eseguito nel frame selezionato.
- Fase 3** - Vengono aggiunti i **keyframe principali** lungo la durata dell'animazione, partendo da quelli più importanti per la buona riuscita della stessa; nel caso di una camminata, ad esempio, verranno inseriti prima i keyframe relativi all'istante in cui il piede tocca il suolo e quello in cui si stacca da terra.
- Fase 4** - Si inseriscono altri keyframe, oltre a quelli già posizionati, per aggiungere dettaglio e realistica all'azione: sempre nel caso di una camminata saranno gestiti i movimenti di braccia, testa e spalle oppure saranno inseriti frame per controllare meglio i tempi dei movimenti principali.  
Questi keyframe secondari sono detti *inbetweener*.
- Fase 5** - Si scende nel dettaglio delle tempistiche andando eventualmente a modificare il metodo di **interpolazione tra i keyframe** operato da *Blender* per ottenere effetti diversi: ad esempio un'azione in partenza da fermo necessita di avere un'accelerazione iniziale prima di raggiungere una velocità costante, questo si ottiene applicando un effetto di *ease in* ai primi due frame, ovvero la loro interpolazione non darà origine ad una linea retta, ma ad una sinusoide che può essere ulteriormente modificata a mano nell'apposita schermata chiamata *Graph Editor*.  
In questa schermata vediamo tramite grafico le trasformazioni sugli assi del componente selezionato, rispetto al tempo, quindi, modificando direttamente il grafico, si possono modificare le tempistiche del movimento.

### 8.3. CREAZIONE DEI PERSONAGGI

---

**Fase 6** - È importante fare un **controllo di qualità generale** andando a vedere quelle che sono le comuni criticità delle animazioni 3D.

Piccole compenetrazioni, sia al terreno, sia tra parti del modello del personaggio, specialmente se questi indossa vestiti lunghi o presenta dettagli come sciarpe, trecce, ecc. . .

Coerenza con l'ambiente in cui avviene l'animazione, ovvero, va controllato che il personaggio segua correttamente l'ambiente in cui dovrà essere eseguita l'animazione, facendo attenzione che questo non vada a collidere con eventuali ostacoli della scena, ma anzi, che vengano sfruttati per dare credibilità al suo movimento, magari facendolo sfiorare un muro con la mano per appoggiarsi nel caso di un'azione di camminata.

È utile anche un controllo generale delle tempistiche dei movimenti, e nel caso non sia soddisfacente, si può semplicemente muovere i keyframe lungo la timeline fino ad ottenere il risultato voluto.

**Fase 7** - Si passa a **pulire l'animazione** eliminando eventuali keyframe creati per componenti che non subiscono trasformazioni selezionando tutti gli elementi del Rig in pose mode e utilizzando l'apposito comando *Clean Channels* in quanto anche keyframe inutili influiscono sul peso dell'animazione.

**Fase 8** - L'animazione terminata viene **salvata nella sezione *NLA Editor*** con il nome assegnatole. Si può quindi tornare al *Dope Sheet* e selezionare "New" per iniziare a creare una nuova animazione sullo stesso personaggio.

Ovviamente ogni animazione così salvata può essere successivamente modificata se la cosa si rende necessaria.

**Fase 9** - Qualunque azione si cerchi di rappresentare è importante ricordare che il pubblico è estremamente critico verso le animazioni di personaggi umani in quanto, essendo cose che si vedono ogni giorno, basta anche il minimo errore per renderle meno credibili allo spettatore e romperne così parte dell'immersione, a maggior ragione nel mondo della VR dove, come già detto, tutto si basa sul fattore immersione.

L'idea è quindi di avere **massima fluidità**, per farlo è importante cercare in fase di ideazione il modo più semplice per rappresentare ogni azione evitando di cadere nella trappola di voler ad ogni costo rappresentare movimenti estremamente complessi per abbellire l'animazione, in quanto i movimenti naturali molto raramente sono spettacolarizzati.

### 8.4 Implementazione dei modelli

Avendo a disposizione tutti i modelli con relative animazioni, questi vengono inseriti in *Unity*, il motore di gioco, dove vengono sistemati coerentemente con la funzione che dovranno svolgere nel gameplay facendo attenzione ad associare agli oggetti i relativi script creati dai programmatori del team; questi servono a gestire l'interazione dei vari oggetti con l'ambiente e con il giocatore, oltre che a gestire il loro ciclo vitale e le loro animazioni.

Viene creata una versione **Alpha** del gioco, ovvero una versione non definitiva che può essere oggetto di modifiche e cambiamenti, a scopo di testing volti a scoprire eventuali situazioni problematiche per l'utente finale.

Segue una successiva fase **Beta** in cui viene fatto provare il prodotto a persone esterne allo sviluppo dello stesso.

Da alcuni feedback si è vista la necessità di modificare il percorso del player che risultava eccessivamente tortuoso e poco comprensibile per alcuni utenti; questa modifica ha occupato poco tempo dato che è bastato spostare alcuni oggetti che coprivano alla vista la porta verso cui si sarebbe dovuto proseguire, ma l'importanza della questione non è da sottovalutare dato che, se non risolta, sarebbe potuta essere un'eventuale causa di critiche dagli utenti.

### 8.5 La distribuzione

Da qui è partita la fase di pubblicazione del gioco sugli store designati, in particolare *The Villa: Allison's Diary* è stato pubblicato sul **Oculus Store** per **Gear VR** ed in futuro sarà reso disponibile per la piattaforma *Oculus Rift*, visore più performante che permetterà l'integrazione delle ombre dinamiche e una maggior definizione delle texture per il nostro gioco.

*The Villa: Allison's diary* è uscito ufficialmente sul mercato a fine novembre 2017 e, come predetto nello svolgimento di questo elaborato, il gioco, pur offrendo una semplice esperienza da *walking simulator* senza innovare per estetica o meccaniche di gioco, ha prodotto discrete vendite complice il basso numero di app disponibili per *Gear VR* rispetto altre piattaforme per videogiochi.

Le vendite complessive di questo titolo e dei precedenti capitoli della saga sono sufficienti a permettere alla piccola azienda di KR Games di rimanere in attivo e di poter quindi continuare a maturare esperienza in questo settore in attesa di proporre prodotti più complessi in futuro. Il nostro prodotto è riuscito a raccogliere su Oculus.com il 58% di valutazioni a cinque stelle e le relative recensioni lodano proprio la tipologia di esperienza proposta, mentre le comunque presenti recensioni negative, indicano come causa del voto la breve durata dell'esperienza e la pochezza del gameplay.

## 8.5. LA DISTRIBUZIONE

---

La cosa è in linea con le aspettative in questo tipo di esperienze non vuole rivolgersi ad un pubblico che cerca un videogame tradizionale, complesso nelle meccaniche e longevo nella fruizione.

## Conclusioni

Questo studio ha voluto analizzare le modalità di sviluppo di un'applicazione per realtà virtuale ponendo attenzione alle differenze col videogioco tradizionale cercando di capire le potenzialità narrative del mezzo.

Per questo è stata svolta una fase di ricerca sulla storia e sul presente del mondo videoludico analizzando alcuni casi specifici e le particolarità del mercato italiano; inoltre la partecipazione personale alla creazione di un videogioco destinato al mercato VR presso l'azienda KR Games ha permesso di ampliare la mia visione su questo campo.

L'esperienza vissuta e le ricerche mostrano come sia l'utenza che l'industria abbiano ancora diffidenza verso la VR anche se questa può potenzialmente avere nelle sue opere la stessa profondità narrativa delle opere videoludiche classiche.

Il risultato conferma le aspettative espresse nell'introduzione dell'elaborato secondo cui le potenzialità delle tecnologie VR sono ancora in larga parte inesprese nell'ambito del videogioco seppur questa tecnologia generi un maggior fattore di immersione per l'utente, sia questo un utente *casual* o *hardcore*.

Le spiegazioni di questi risultati possono essere trovate nelle parole dei fondatori di KR Games, nell'intervista a me concessa (Capitolo 4.3.1), e nelle analisi sul mercato videoludico odierno che offrono una dettagliata panoramica sul rapporto tra pubblico e VR.

Il risultato ottenuto rispecchia la situazione odierna in cui i maggiori produttori di videogame guardano marginalmente al mercato VR, spesso ignorato dall'utenza, nonostante le grandi potenzialità di cui dispone, mentre piccole imprese possono provare a sperimentare con questa nuova tecnologia senza incontrare la concorrenza dei grandi nomi del settore.

Questo studio, oltre ad integrare ricerche su documentazione già esistente, presenta un'analisi interna sullo sviluppo di questo nuovo tipo di esperienze che sfruttano la VR come mezzo per trasmettere sensazioni, analizzando le diverse tecniche di modellazione ed animazione usate e del perché ne vengono preferite alcune al posto di altre.

## CONCLUSIONI

---

Per ulteriori ricerche future si potrebbe analizzare nello specifico il neonato *Playstation VR* per capire come un'azienda leader del settore come Sony punti a far evolvere questa recente tecnologia ed integrarla con i suoi titoli caratterizzati dalla forte componente narrativa in stile cinematografico spesso mista ad un gameplay frenetico.

## Ringraziamenti

Con questa tesi si conclude il mio percorso di studi iniziato fin troppi anni fa; a volte mi sembra di star studiando da tutta la vita e che per tutta la vita abbia inseguito un'unica idea per il mio futuro: lavorare nel mondo dei videogiochi. Altre volte mi sembra di star videogiocando da tutta la vita e che per tutta la vita abbia inseguito un'unica idea per il mio futuro: lavorare nel mondo dei videogiochi.

A volte divertimento e lavoro vanno a braccetto.

Crescendo in un piccolo paese di montagna, con poche persone e ancor meno accessi al mondo “nerd”, che fossero negozi o eventi, ho faticato a trovare chi potesse capire le mie passioni e i miei interessi, ma l'università, in particolare gli anni a Torino, sono stati un toccasana per me: non ero più l'unico “strano”.

Continuerò sempre a non sentirmi completamente normale, ma grazie al corso di Ingegneria del Cinema ho incontrato persone con cui non ho avuto problemi a condividere le mie stranezze, cosa che fino a soli pochi anni fa non avrei creduto possibile: ora tutti sanno che seguo il Wrestling con passione (unico VERO sport), che leggo manga romantici destinati ad un pubblico femminile e che vorrei trovare il tempo di scrivere un monologo comico. . . Sono una persona tanto solitaria, non sono bravo a socializzare e non mi interessa diventarlo, ma devo ringraziare l'ambiente del Politecnico per avermi fatto trovare persone che posso considerare amici e che ricorderò per tanto tanto tempo.

Tanto i ringraziamenti non li legge nessuno, quindi posso scrivere tranquillo le peggio cose imbarazzanti.

Nonostante i vari dubbi durante il mio percorso di studi e di vita, alla fine ce l'ho fatta: sono riuscito a inserirmi in KR Games, un'azienda che produce piccole opere videoludiche, piccole opere d'arte, piccole opere che mettono su video un'esperienza plasmabile dallo spettatore, uno spettatore che può interpretare, analizzare capire o meno l'idea che mettiamo nel nostro lavoro. È incredibile pensare che qualcuno stia giocando qualcosa che ho contribuito a creare, davvero una sensazione incredibile e voglio che questa cosa continui per tutto il tempo possibile.

## RINGRAZIAMENTI

---

Mi illudo di essere già un grande artista. . . ma so benissimo che non c'è nulla di più lontano dalla verità, non ho grandi ambizioni, non pretendo di diventare l'Hideo Kojima italiano, mi basta vivere una vita tranquilla e felice, con la consapevolezza di aver raggiunto il mio obiettivo; obiettivo che pochi hanno capito fino in fondo.

Ma se sono arrivato a questo risultato non è solo merito mio, ci sono molte persone, o molti NPC, senza di cui non ci sarebbe questa tesi e senza di cui non ci sarebbe il mio io attuale.

Ringrazio innanzitutto il mio relatore, il professor Riccardo Antonino, che mi ha permesso di scrivere in libertà una tesi su un argomento a me molto caro, e KR Games, in particolare Roberto Comella e Kabir Ferro, che hanno dato fiducia alle mie capacità e hanno sopportato i miei errori, spero davvero che il rapporto di lavoro/amicizia continui a lungo.

Entrando in ambito ancora più personale non posso ringraziare abbastanza i miei genitori, Enrica e Sandro, per avermi sostenuto in ogni mia scelta, per l'incredibile supporto morale e per il grande aiuto economico che spero di poter ricambiare in futuro; se sono la persona che sono oggi è solo merito loro.

Ringrazio le mie stupende nonne, Carla e Illeana, per essere sempre pronte ad aiutarmi a modo loro, con tutte le cortesie e le gentilezze che solo le nonne possono avere.

Ma anche fuori dalla famiglia ho avuto molti amici che mi hanno sostenuto nonostante il mio carattere: ringrazio caramente tutti i colleghi universitari di triennale e magistrale, in particolare Alessia, Riccardo, Giuseppe, Melissa C., Michele B., Giorgia e Valentina con cui ho passato tanti bei momenti in allegria.

Un grazie particolare a Michele C. per spronarmi a continuare a migliorare sul lato tecnico delle mie brutte opere e per i tantissimi consigli, e un grazie tutto speciale per Margherita, la persona più dolce che conosco, che mi ha dimostrato che posso fare amicizia anche con persone con cui non ho neanche mezzo interesse in comune.

Ovviamente i ringraziamenti tra colleghi vanno estesi a tutti quegli amici che non vedo da un po', ma che mi hanno tenuto compagnia e sopportato, in particolare ringrazio Corrado, Matteo V., Guglielmo, Matteo F., Andrea e Gerry.

Poi ci sono gli amici di sempre: Federico e Melissa. Cosa scrivo a fare su di voi, tanto vi vedo sempre.

## RINGRAZIAMENTI

---

Per i ringraziamenti inaspettati aggiungo Emanuele e Giulia; so che sono secoli che non vi sento e probabilmente non leggerete mai queste righe, ma senza di voi non so se sarei sopravvissuto spiritualmente al liceo, davvero.

In ultimo ringrazio mio nonno Ermanno per essere stato il mio migliore amico quando amici non ne avevo.

I nomi sono inseriti in ordine puramente casuale perché a me l'organizzazione non piace; se ho dimenticato qualcuno, lo sapete, ho una memoria terribile, specialmente per i nomi, ma se anche solo mi avete dato una mano, o mi avete fatto compagnia sui lunghi viaggi in treno, sappiate che vi sto ringraziando spiritualmente. Mica posso fare 5 pagine di ringraziamenti, no?

## RINGRAZIAMENTI

---

## Bibliografia

- [1] Beane, Andy *3D Animation Essentials* (2012), Sybex, Alameda
- [2] Carlà, Francesco *Space Invaders. La vera storia dei videogames* (1996), Castelvecchi, Roma
- [3] Felinto, Dalai *Game Development With Blender* (2013), Cengage Learning Ptr
- [4] Kerlow, Isaac *The Art of 3D Computer Animation and Effects* (4<sup>th</sup> edition, 2009), Wiley, New Jersey
- [5] Lavalle, Steven M. *Virtual Reality* (2017), Cambridge University Press
- [6] Norman, Donald Arthur *La caffettiera del masochista* (1990), Giunti, Milano
- [7] Vaughan, William *Digital Modeling* (2012), New Riders, Thousand Oaks

### **Testi, articoli e video informativi**

- [8] Bittanti, Matteo *I videogiochi? Pezzi da museo* (2013), Wired Italia
- [9] Contin, Alessandra *Il mercato dei videogiochi in Italia vale un miliardo* (2017), La Stampa
- [10] Cutini, Paola *Vigamus, il passato e il futuro dei videogiochi. Dai reperti di E.t. the Fall all'Oculus Rift* (2014), RAI NEWS 24
- [11] Ferrari, Filippo *Intervista a Hideo Kojima, Mister Metal Gear Solid* (2014), Wired Italia
- [12] Poggi, Michele *FRAMERATE, cos'è? La diatriba "Cinematografica" - L'Effigie del Mercato* (2015), Sabaku no Maiku
- [13] Poggi, Michele *Un Primo Sguardo alla "Playstation VR"* (2016), Sabaku no Maiku
- [14] Ruffilli, Bruno *Adesso i videogiochi parlano anche italiano* (2017), La Stampa
- [15] Zorloni, Luca *L'alleanza tra Mario e Rabbids nata in quel piccolo studio di Milano* (2017), Wired Italia

**Sitografia**

- [16] [AESVI: associazione editori sviluppatori videogiochi italiani](#)
- [17] [Blender Artists](#)
- [18] [Blender Guru](#)
- [19] [Blender reference manual](#)
- [20] [Gallo, Giovanni \*Materiale Didattico - Corso di Computer Graphics\*,  
Dipartimento di Matematica e Informatica, Università degli Studi di Catania](#)
- [21] [Oculus sezione developer](#)
- [22] [Unity reference manual](#)
- [23] [Virtual Reality, New World Encyclopedia, \(2016\)](#)