POLITECNICO DI TORINO

Collegio di Ingegneria Informatica, del Cinema e Meccatronica

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

Tesi di Laurea Magistrale

Analisi del mondo videoludico - Redazione di un manuale di Game Design e Productions in collaborazione con IVProductions

Relatore
Prof. Andrea Giuseppe Bottino
Correlatori
Prof. Marco Mazzaglia
Ivan Venturi

Candidato
Giovanni Zabardi
s261591

Anno Accademico 2019/2020

Indice

1 Introduzione **4**

2 Panoramica sui Videogames **6**
2.1 Utenza 7
2.2 La realtà internazionale 18
2.3 La realtà italiana 20
2.3.1 Le principali software house indie italiane 22

3 Fare Videogiochi **29**
3.1 Ivan Venturi e IVProductions 30
3.1.1 Svilupparty 35
3.1.2 Italian Games Factory 37
3.2 Fare Videogiochi in Italia 38
3.3 Una Simulazione: da hobby a mestiere 42
3.3.1 Sviluppo I: Pre-Produzione 45
3.3.2 Sviluppo II: Produzione 49
3.3.3 Sviluppo III: Post-Produzione 53
3.3.4 Sviluppo IV: Post-lancio 57
3.4 I 4 Pilastri 62
3.4.1 Il Ruolo della Narrativa 62
3.4.2 La Scelta della Tecnologia 79
3.4.3 L'Importanza dell'Estetica 86
3.4.4 La Centralità delle Meccaniche 90
3.5 Costruire un team: le figure professionali 93
3.5.1 Game Designer 95
3.5.2 Game Programmer 111
3.5.3 2D - 3D Artist 120
3.5.4 Publisher 131
3.6 Design vs. Mercato 135
3.6.1 Le Piattaforme e la Distribuzione 143
3.7 Genere e Tipologie 147
3.7.1 Action 148
3.7.2 Adventure 160
3.7.3 Action-Adventure 164
3.7.4 Sportivo 166
3.7.5 Gioco di Ruolo 167
3.7.6 Strategico 174
3.7.7 Racing 176
3.7.8 Simulazione 177
3.7.9 Ritmico 179
3.7.10 Puzzle Game 179
3.7.11 Gli Pseudo-Generi 181

3.7.12 Progressione Vs. Emersione 182

4 Redazione del Manuale: le necessità e le finalità 187
4.1 Target di riferimento 187
4.2 Lo Stato dei Lavori 188
4.2.1 Idee di Progettazione 188
4.2.2 Forma Comunicativa e Tone of Voice 189

5 Conclusioni 191

Abstract

Lo scopo di questa tesi, redatta in collaborazione con **Ivan Venturi** e **IV-Productions**, è quello di esplorare ed analizzare il mondo dei videogames a partire dal mercato e presentare quali sono le caratteristiche principali del mondo dello sviluppo video-ludico: il documento analizza il settore a partire da una panoramica generale e successivamente sviscera quali sono le criticità e le fasi lavorative, inserendo un compendio dettagliato sulle figure professionali, i generi ed il loro rapporto con il mercato, le tecnologie utilizzabili e gli step produttivi. Oltre a questo primo obiettivo, la tesi ha la funzione di prototipo per la redazione di un manuale di game design e production, in sviluppo con IVProductions, e la presentazione dello stato dei lavori di quest'ultimo. Per la stesura della tesi sono stati utilizzati dati e sessioni divulgative prodotti dall'azienda stessa, al fine di avere un'ingente raccolta dati utile al trasporre "su carta" l'esperienza lavorativa dell'azienda di come realizzare videogames e raggiungere l'obiettivo preposto all'inizio della collaborazione; oltre a questi dati si sono sfruttate conoscenze personali e diverse fonti testuali di spicco all'interno del settore video-ludico.

1 Introduzione

Questo lavoro, sviluppato in collaborazione con Ivan Venturi ed IVProductions, vuole essere un prototipo per un manuale di game design e production utile ai futuri professionisti del settore o agli sviluppatori già presenti sul mercato. La tesi ha un carattere analitico ed informativo, seguendo un percorso di fruizione che possa iniziare anche i meno informati per quanto riguarda il mondo video-ludico: il documento è diviso in tre macro-sezioni, le prime con carattere introduttivo ed analitico, mentre la terza illustra il progetto di redazione del manuale al momento in sviluppo con l'azienda, mettendo in luce le necessità, lo stato dei lavori, il target di riferimento e le idee di progettazione futura.

La prima sezione effettua una panoramica generale del mondo dei videogames, analizzando sia la realtà internazionale e sia la realtà italiana da un punto di vista economico e produttivo; in questa sezione, viene fatto uno studio dell'utenza dividendola per fasce di età e si propone un'analisi delle preferenze e delle occasioni di gioco, oltre ad effettuare un approfondimento sul game design femminile: qui si spiega come alcune tendenze di game design stiano mutando in favore di nuove visioni legate ad un movimento che vuole opporre il *tend and befriend* al da sempre dominante *fight or flight*, tipico della tradizione video-ludica. Nella sezione riguardante il gaming italiano, l'analisi si sofferma sui molteplici punti di vista riguardanti il rapporto tra i videogame e la realtà italiana, mostrando i dati di crescita ed i limiti attuali che ovviamente dovranno essere superati per rimanere al passo con la realtà internazionale; al termine di questa sezione, è stata riportata una lista delle più influenti, in presente ed in passato, software house italiane: di ognuna si evidenziano la storia, i risultati ed il genere di competenza.

La seconda sezione, la più corposa della tesi, compie un excursus storico sull'attività di Ivan Venturi e sul lavoro di IVProductions: partendo dagli albori dello sviluppo italiano, con la fondazione di Simulmondo, il testo ripercorre le tappe e le scelte dell'epoca seguendo la linea del tempo. Infatti dal 1993 fino alla fondazione di IVProductions, il lavoro di Venturi come produttore e sviluppatore di videogiochi ha interessato diversi e numerosi progetti.
Terminato l'excursus, il documento analizza in poche pagine altre due realtà parallele/affiliate a quella di IVProductions, ovvero Italian Games Factory e Svilupparty: proprio quest'ultima è quella che risulta più significativa per il mondo dello sviluppo italiano; trattasi di un evento (chi scrive questa tesi ha partecipato all'edizione di quest'anno) al quale partecipano appassionati e professionisti del settore, creando un'occasione di incontro e confronto volto alla crescita del settore: si scambiano opinioni, si mostrano i progetti in cantiere e si assiste a più talk di vari ospiti provenienti dal mondo lavorativo

dei videogiochi (sviluppatori, giornalisti, influencer, ...). Successivamente la tesi mette a fuoco cosa significa sviluppare videogames in Italia, evidenziando quali siano le istituzioni di riferimento, quali sono i possibili costi standard e le prime mosse da fare per poter dare un'identità al proprio gioco; da questa analisi, si passa allo studio dettagliato di quali siano i passi e le considerazioni progettuali fa fare durante lo sviluppo di videogiochi, anche grazie ad una simulazione verosimile del mestiere video-ludico: si immagina di supervisionare lo sviluppo del primo videogioco di un team di sviluppo alla prima opera, dal primo incontro in cui si decide di intraprendere un progetto di questo tipo alla pubblicazione del titolo, spiegando passo passo quali siano le domande da porsi, gli obiettivi da perseguire e gli errori in cui è possibile incappare. Una volta conclusa la simulazione, il testo propone la generalizzazione di questi step, suddividendo lo sviluppo in quattro fasi fondamentali: pre-produzione, produzione, post-produzione e post-lancio. Per fornire una visione completa del lavoro in ambito video-ludico, la tesi si sofferma su quali siano le sezioni d'investimento utili per lo sviluppo di un videogioco, ovvero i quattro pilastri utili alla sua produzione; proprio per questo le scelte effettuate a livello economico e progettuale, determinano il tipo di team che lavorerà al progetto: vi è un'intera sotto sezione che esplora le differenti figure professionali collocandole nel processo lavorativo e definendone le mansioni principali e secondarie, dando maggiore attenzione alla figura del game designer. Il documento scende ancor più nei particolari approfondendo le insidie, i requisiti del mercato e di come questi siano intrinsecamente collegati alla tipologia e generi di gioco che uno sviluppatore decide di creare; come ogni prodotto, il videogioco ha una sua merceologia ed una sua tassonomia legata alla tipologia di gioco, quindi all suo genere: l'ultima parte della seconda sezione stila una lista di tutte le tipologie di gioco a partire dai proto-generi fino a quelli ibridi, analizzandone le specifiche e le caratteristiche.

La terza sezione della tesi esprime i risultati ottenuti durante il lavoro parallelo con IVProductions per la costruzione di un manuale di game design e production basato sullo sviluppo italiano, di cui la tesi stessa è sia prototipo che scheletro di base; al momento il progetto è si trova in piena preproduzione: si stanno raccogliendo dati ed elaborando idee di progettazione in modo da rendere il prodotto di livello professionale, ma accessibile.

In conclusione, si tenga presente che tutta l'analisi è accompagnata da immagini ed esempi che aiutano la fruizione del testo, in modo da rendere pratico e visibile il tipo di studio effettuato; inoltre parecchio del materiale grafico ed analitico è stato fornito dall'azienda stessa per l'arricchimento e lo sviluppo della tesi, con l'obiettivo di formare un piccolo compendio utile sia agli addetti ai lavori sia a chi non conosce il mondo ed il mercato dei videogiochi.

2 Panoramica sui Videogames

Il valore assunto dai videogames all'interno della nostra società è incredibilmente alto: chiunque nella propria routine quotidiana ha che fare con i giochi o fenomeni di *gamefication*; che si tratti di brevi sessioni di gioco su smartphone o di intense sessioni dei giocatori più appassionati, la nostra intera vita è costellata di momenti di gioco grazie al vasto utilizzo delle interfacce digitali. Negli ultimi vent'anni, il mondo del gaming ha visto una crescita esponenziale in tutto il pianeta, diventando uno dei settori più redditizi e permeanti fra tutte le attività umane. Questa crescita è anche dovuta ad una caratteristica intrinseca del settore stesso: tutti giocano, non c'è persona che non giochi o non abbia giocato nella propria vita; non esistono limiti di età, ma semplicemente tipi di gioco e momenti di gioco differente. Ovviamente oltre ad una crescita e ad una contaminazione culturale senza precedenti, risulta evidente anche il cambio economico e produttivo che è avvenuto: il mercato del gaming è cresciuto tantissimo ed ogni realtà produttiva è differente; nelle sezioni 2.2 e 2.3 verranno analizzate quali sono le situazioni e lo stato dell'arte a livello globale e poi, in particolare, per quanto riguarda la situazione italiana.

Quando si tratta del mondo dei videogames, bisogna pensare al fatto che si stia sorvolando su un vasto orizzonte di professioni: i videogames sono un'industria enorme, vastissima, che non riguarda solamente il videogioco in quanto tale. Basti pensare all'analisi dell'utenza, ai servizi che vengono forniti, allo *streaming* e relativo mercato degli *influencer*, per non citare i publisher, la stampa e la distribuzione.

Figura 1: Pow3erTv è il primo gaming influencer italiano secondo Buzzoole Rankings

Ed anche limitandosi solo ai videogiochi, esistono una miriade di generi, un sacco di modalità di realizzazione ed anche diverse motivazioni alla base dei vari progetti: una di queste è sicuramente il target, quindi l'utente finale a cui è destinato il prodotto.

2.1 Utenza

Per prima cosa, bisogna valutare e discernere l'utenza che gioca ai videogiochi. Ma esiste qualcuno che al giorno d'oggi non gioca ai videogame? Nel 2020, tutti giocano ai videogiochi: esiste una mera differenza di quanto si gioca.
Si può però evidenziare una differenza del tipo di giocatore a seconda dell'età.

- **3 years – Infants**: si tratta del primo segmento di pubblico, nato con l'avvento del tablet; l'input dato dalle possibilità del touch screen garantisce un output diretto, anche se ovviamente non decidono a cosa giocare, sono i genitori che gli propongono i giochi. Ecco, anche su quest'ultima parola bisogna ragionare: non sono veri e propri giochi, ma piuttosto *giocattoli* che provocano piacere sensoriale audiovisivo, non hanno regole (es: giochi della **Toca Boca**).

Figura 2: *Toca Kitchen 2* è un gioco per bambini del 2014, disponibile su IOS e Android, in cui i giovanissimi giocatori devono far mangiare i propri esperimenti culinari a tre personaggi.

- **5 years – Prescolare**: a quest'età iniziano ad emergere le prime capacità di comprensione delle regole e del mondo; si può dire che i giocattoli diventano giochi.

- **8 years – Primaria 1**: tramite la conoscenza acquisita nei primi anni di scuola (lettura, matematica,...), si maturano i propri gusti ed il fatto di vincere/perdere diventa chiaro.

Figura 3: *Crash Team Racing*, sviluppato da **Naughty Dog** per Playstation nel 1999, è un gioco competitivo basato sulle corse di go kart del franchise di *Crash Bandicoot*, sicuramente adatto a quest'età.

- **9-13 years – Primaria 2/Pre-teen**: in questa fase, si inizia a diventare veri e propri giocatori; si vanno a formare le passioni reali anche perché si ha uno sviluppo neurologico importante.

- **14-18 years – Teen**: qui si ha un vero e proprio split di genere poiché le ragazze sono più interessate alle dinamiche sociali, che ai videogames; stando ai numeri, risultano più interessate alle esperienze di vita, mentre i ragazzi amano di più la "distruzione" che spesso il gaming fornisce.

Figura 4: La serie *Doom*, sviluppata originariamente da **John Romero** e **ID Software** nel 1993 per MS-DOS, è l'emblema della mascolinità nel gaming, basato sulle avventure di un innominato space marine che massacra i peggiori demoni dell'Inferno.

- **19-24 years – Young adult**: iniziano le responsabilità della vita, il tempo libero si riduce e si diventa più selettivi.

- **25-35 years – Thirties**: il tempo da dedicare al gaming è davvero scarso, si inizia a lavorare e terminare gli studi; si cerca di mettere su famiglia...ma in mezzo a queste complicazioni, nascono gli hardcore gamer; chi, a quest'età, continua a giocare, ritagliandosi i tempo, viene definito hardcore gamer.

Figura 5: Esempio di un'elaborata postazione da hardcore gamer

- **36-50 years – Adults**: tutto viene pensato in maniera plurale e non individualistica, la famiglia, il lavoro vengono prima.
- **50+ years**: paradossalmente in questa fase, si riguadagna tempo anche perché i figli sono grandi; il problema risulta quello dell'invecchiare, per cui molti giochi diventano meno accessibili a causa del rallentamento dei riflessi e della coordinazione mano-occhio.

Va evidenziato come anche il genere dei giocatori, naturalmente osservando le statistiche, ha una valenza anche sul mondo dei videogiochi: le donne hanno un maggior interesse nei confronti del mondo reale e al far "crescere" le cose, solitamente giocano a puzzle verbali e giochi basati sui dialoghi. Preferiscono imparare con tutorial e guide, odiano perdere, quindi scelgono spesso giochi, ad esempio, del tipo *hidden object*, a cui giocano solitamente donne tra i 35 ed i 45 anni nordamericane.

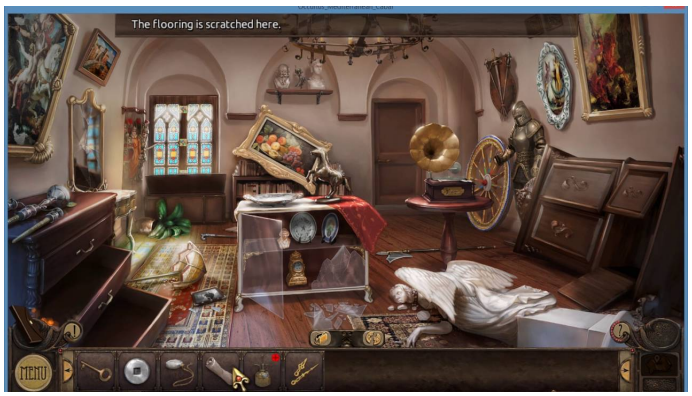


Figura 6: *Occultus: Mediterranean Cabal* è un hidden-object game sviluppato da **Sylphe Labs** e **IVProductions** per Windows e MacOS, uscito nel 2017.

I maschi, invece, hanno una maggiore tendenza a voler vincere, al distruggere, raggiungendo il proprio obiettivo senza leggere i manuali o seguire i tutorial, ma piuttosto seguendo un percorso di *trial and error*. Si tratta ovviamente di differenze di massima, infatti ogni giocatore ha una componente maschile e femminile nel proprio approccio con il gaming.

Figura 7: La serie *Dark Souls*, sviluppata da **Hidetaka Miyazaki** e **From Software**, è un RPG rivoluzionario che basa il suo gameplay sul trial and error e la capacità del giocatore nel affrontare situazioni differenti, adattando il proprio stile di gioco.

Femme Game Design Come in ogni ambito sociale/lavorativo umano, anche nel mondo dei videogames esiste un sistema di catalogazione e suddivisione legata al genere. In generale l'approccio maschile ha da sempre influito e imposto il proprio trend alla maggior parte dei videogiochi mainstream, invece la femminilità è stata spesso relegata a videogiochi di nicchia/casual; negli ultimi anni, gli stessi sviluppatori di videogames hanno messo in discussione questa realtà e gli schemi tradizionali, cercando di ampliare la base di giocatori: si tratta della nuova frontiera dell'inclusione di genere nei videogame, ovvero il **femme game design**. Iniziamo a fare qualche esempio: *Boyfriend Dungeon* (**Kitfox**) è un *dungeon crawler hack 'n' slash* che incontra un *dating sim* (si faccia riferimento a sezione 3.7), infatti sarà possibile per la protagonista uscire ed innamorarsi delle proprie spade durante degli appuntamenti; per giochi più mainstream e conosciuti, il DLC *The Last Of Us: Left Behind* (**Naughty Dog**) ha preso l'esperienza action-horror del gioco principale trasformandola in una sezione video-ludica più basata sull'esperienzialità che sul gioco hardcore. Questi sono solo due esempi importanti di game design femminile, ma la lista non si esaurisce ovviamente qui. Anche sul mercato ci sono stati risultati molto importanti:

Animal Crossing New Horizons, sebbene faccia parte di un franchise molto conosciuto, proprio recentemente ha raggiunto vette importanti: è stato il gioco più giocato del periodo, nonostante la concorrenza diretta di *Doom: Eternal* che incarna lo stile tipico ipermascolino dominante nel game design mainstream. Che si tratti dello stile o delle meccaniche implementate, questi titoli testimoniano la svolta verso una progettazione più femminile, o meglio, un allontanamento dal classico stereotipo di schema di gioco tradizionalmente maschile: attraverso l'unione di "generi" ed il rovesciamento degli schemi classici, gli sviluppatori ed i creativi del gaming stanno rompendo i dettami del binarismo e aprendo un nuovo mondo di esperienze ai giocatori. Se da un lato, va riconosciuto lo sforzo di inclusione di genere da parte del mondo dell'industria video-ludica, dall'altro bisogna continuare a lavorare sulla creazione di diversi tipi di esperienze che sfidino i principi su cui si basa la teoria del game design.

Figura 8: *Animal Crossing: New Horizons* è il nuovo capitolo del franchise sviluppato da **Nintendo** e conta milioni di fan in tutto il pianeta sia tra gli hardcore gamers che tra i casual gamers.

Cos'è il game design femminile?

È bene sottolineare come non si intenda distruggere il blocco di game design "maschile" per passare ad uno "femminile" o sostenere l'esistenza di poli opposti nell'approccio di progettazione, ma piuttosto spiegare che in ogni giocatore esiste una parte femminile ed una maschile, come fossero due energie che regolano diversi aspetti della nostra persona. Secondo **Brie Code**, capo programmatrice AI di *Assassin's Creed* e *Child of Light*, si è dato troppo spazio alla parte maschile e l'equilibrio di queste due energie si è perso: gioverebbe a tutti ritrovarlo.

> "Le donne e le persone di identità femminile sono spesso state ignorate dalla teoria del game design e dalla ricerca psicologica

che ne è alla base. Nello sviluppo del gioco favoriamo empatia e compassione più che competizione e dominio." ***Brie Code*** (1)

Code spiega che molto spesso le donne o le persone di identità femminile sono ignorate dalla teoria del game design e dalla ricerca psicologica che ne è alla base: infatti, essendo una convinta sostenitrice del femme game design, ha deciso di fondare **Tru Luv**, una società dedicata all'AI con funzioni di compagnia e alla tecnologia al servizio della persona, per trovare equilibrio nelle relazioni.

Figura 9: Logo della società fondata da Brie Code

Alla base di questa scelta, vi è la convinzione che l'industria video-ludica stesse rinunciando ad importanti elementi di innovazione, marginalizzando determinate identità di genere che avrebbero potuto contribuire molto in materia di progettazione. Ad esempio, preferire l'empatia e la compassione, piuttosto che il dominio o la competizione. In particolare, Code si oppone al modo in cui i designer concepiscono il **flow**, ovvero quel principio psicologico che descrive uno stato di totale coinvolgimento e concentrazione, o meglio una sensazione di estremo focus e piacere che si prova quando si gioca ad un videogame; ovviamente se ci riesce, significa che il game design è di ottimo livello, su questo non si discute, ma piuttosto è il dibattito sul come arrivarci a non essere molto sviluppato.
Spesso e volentieri il game design tradizionale sfrutta il concetto di **"fight or flight"** per creare esperienze di tensione sempre crescente e quindi mantenere il giocatore nel flow ed instillare in lui il desiderio di vincere: questa è una tipica progettazione prettamente maschile, presente in titoli come *Doom* o *Tetris*. Code evidenzia come la metà delle persone, però, non reagisce allo stesso modo alle situazioni di stress e tendono a preferire risposte come quelle di **"tend and befriend"**, attenzione e amicizia, ovvero un concetto che fa leva sul desiderio di prendersi cura, entrare in contatto e risolvere un problema in maniera comune. A Tru Luv hanno ricercato e sviluppato circa 150 modi differenti per generare il flow che non si basano su un modello di difficoltà crescente (stress), ma favoriscono la modalità "tend and befriend", tramite iterazioni focalizzate su evoluzioni dal caos all'ordine, dal conflitto

alla pace e così via.
Si punta a creare legame ed empatia durante la progressione di gioco piuttosto che lavorare su dominio e competizione, infatti l'AI favorisce la co-creazione, sviluppo psicologico e la collaborazione. Si tratta quindi una differenza alla base dell'emozione su cui si vuol lavorare senza metterle in ordine di qualità: un design che punta sull'istinto della paura del giocatore da un lato e uno che mira all'empatia dall'altro; non c'è migliore o peggiore, ma entrambi sono essenziali per l'esperienza umana.

Oltre: sfidare le convenzioni di genere.
Il design femminile non è assolutamente una novità o una realtà così alternativa: in Giappone, ad esempio, è un design che ha molto successo, infatti generi come dating sim, giochi romantici *otome* ed un marketing sempre molto inclusivo e meno ipermascolino, hanno instaurato un rapporto più sano tra il mondo del gaming e le giocatrici giapponesi.

Figura 10: *Persona5* è uno dei migliori JPRG degli ultimi anni, facente parte del franchise *Shin Megami Tensei* sviluppato da **Atlus**; la sua particolarità sta proprio nell'essere un sim/dating simulator oltre ad un gioco di ruolo: ha riscosso molto successo anche in Europa.

In America, la scissione di giochi "maschi contro femmine" risale agli anni '80 quando per risollevare il mercato dopo il crollo dell'industria venne fatto questo marketing mirato per fornire una prassi che etichettasse i giochi e audience: rigidità di questo tipo hanno creato un mancato equilibrio tra le energie di cui appunto si parlava sopra. In Giappone, le giocatrici hanno a disposizione diverse esperienze di gioco tra cui poter scegliere: gli sviluppatori del sol levante continuano ad essere fonte di ispirazione per questo motivo, a partire dal mondo Nintendo o alla serie *Persona (Shin Megami Tensei)*(**Atlus**)

o titoli come *Journey* (**Thatgamecompany**) o *Wattam*(**Funomena**).

Figura 11: *Journey* è un gioco basato sul concetto del "tend and befriend"; uscito il 13 marzo 2013, ha riscosso un successo tale da essere acclamato come capolavoro assoluto del medium.

Il marketing anni '90 ha esasperato la divisione tra generi, pompando gli stereotipi e sessualizzando molti elementi di caratterizzazione, interrompendo una filosofia diversa proveniente dagli anni del NES. **Nina Freeman**, designer di *Cibele* ha evidenziato il ruolo che hanno avuto i giochi rosa di Barbie (dress up per ragazze) nei nineties, ovvero quello di fare da capostipite alla meccanica di personalizzazione dei personaggi, key feature di moltissimi "dudebro games", come *Mass Effect* (**BioWare**) o *Fortnite* (**Epic Games**).

> "Nei paesi occidentali, il marketing degli Anni '90 ha esasperato ogni stereotipo e ha sessualizzato ogni cosa. Quindi, di colpo, gente come me e te, che giocavamo senza alcun problema sulla NES, si è ritrovata nell'era della PSP a chiedersi: "Aspetta un attimo, ma dove sono finiti i nostri giochi? Perché mi dite che questo gioco non mi deve piacere? Perché questa pubblicità mi esclude così prepotentemente?" ***Tanya Short***, Lead Designer di *Boyfriend Dungeon* (1)

I giochi per ragazze sono sempre esistiti, anche se non trovano spazio nel dibattito storico del game design e rimangono ai margini, le loro meccaniche vengono utilizzate nei giochi di combattimento: *Mass Effect* per l'appunto è all'apparenza estremamente mascolino, ma sotto sotto è un dating sim di dress-up. Anche altri generi, tipicamente giapponesi, come le visual novel, hanno influenzato designer occidentali come **Christine Love**, co-creatrice di *Get in the Car, Loser!* (gioco stile road trip lesbico), che nel suo ultimo gioco oltre alle scelte di design "femminili" ha aggiunto le meccaniche di combattimento a turno classiche degli RPG giapponesi, con la differenza che

la leader del gruppo non è la guerriera, ma la guaritrice. Proprio quest'ultima scelta ha rovesciato il classico cliché del personaggio femminile relegato a guaritore in secondo piano, concentrando l'attenzione sull'healer (archetipo di personaggio dedicato al curare e fare da supporto agli altri membri del party) piuttosto che sul personaggio con l'arma più rara.

Figura 12: Uno screen della battle phase di *Get In The Car, Loser!*

> "I personaggi femminili finiscono sempre per essere relegati al ruolo di guaritori. Fare di lei la protagonista rovescia questo cliché. È una presa di posizione netta. Perché l'attenzione, invece di concentrarsi sulla persona con la spada più grande, è riservata al personaggio che aiuta tutti gli altri. È lei quella su cui il gruppo fa maggiore affidamento. Se non fosse per la guaritrice, sarebbero tutti morti al primo combattimento e il gioco sarebbe finito lì." ***Christine Love***, Designer di *Get In The Car, Loser!* (1)

Ma tutte queste rigide etichette funzionano davvero? Non proprio.
Molte delle persone che hanno giocato alle visual novel di Love, non erano necessariamente il target di riferimento, ma semplicemente giocatori in cerca di qualcosa di nuovo. Eppure, secondo **Tanya Short** (**Kitfox Games**), editori e piattaforme alla Steam per intenderci, pongono un sacco di ostacoli per i giochi che non rientrano in una determinata etichetta riconoscibile dal pubblico, quindi lavorare su una nicchia di pubblico preciso. Anche quando si parla di game design "femminile" o "maschile" si ricade in una ambiguità grossissima: tutto ciò che non ricade nel segmento maschile viene interpretato come femminile...ma non è così.
Molti giochi che magari piacciono a chi li gioca, li fa comunque sentire esclusi:

è un po' il leitmotif di ogni game designer donna, ovvero hanno fatto il gioco che nessuno faceva per sentirsi incluse.

> "È un po' strano se si è convinti di aver creato qualcosa di più inclusivo rispetto ai tipici giochi di Steam sentirsi invece dire che si dovrebbe puntare ad un segmento di pubblico preciso. È anche per questo che è così difficile definire cosa sia il game design 'femminile'. Dal momento che al termine 'maschile' è stata data un'interpretazione molto restrittiva a causa di strategie di marketing e commerciali ripetitive e semplicistiche, praticamente qualsiasi cosa non ricada in questa definizione così angusta, automaticamente, viene per contrasto considerata 'femminile'."
> ***Tanya Short***, Lead Designer di *Boyfriend Dungeon* (1)

Figura 13: In *Boyfriend Dungeon* si affrontano svariati dungeon grazie all'utilizzo di armi ed oggetti; è possibile sbloccare nuove armi invitandole ad un appuntamento.

Oltre: dopo la logica binaria del game design

Ma la ipermascolinità non influisce solo sulle giocatrici e creatrici femme: è proprio la cultura in toto a perdere metà dell'esperienza umana. Perché un giocatore hardcore non potrebbe emozionarsi giocando ad un gioco tratto dalla vita di una ragazza? Ogni esperienza di questo tipo arricchisce e proprio qui sta il punto. Se la Code sembra ottimista per il futuro, ammette anche che ci è voluto davvero tanto per avviare il dibattito e iniziare una rivoluzione: la prima è stata **Sheri Graner Ray**, con il testo *Gender Inclusive Game Design* (2004) e la stessa Code ha iniziato a tenere conferenze sul tema circa 9 anni dopo. È un concetto che sta prendendo piede solo negli ultimi anni, pian piano, ma le cose per lo meno stanno cambiando.

Vi sono numerosi segnali: un esempio è *Sayonara Wild Hearts* (**Simogo**), un gioco in cui si corre all'infinito con una ragazza disperata per una delusione d'amore come protagonista, che è stato creato da uomini, quindi le idee del femme game design vengono utilizzate anche da creator non marginalizzati, ma appartenenti alla cultura dominante.

Figura 14: *Sayonara Wild Hearts* è un ottimo esempio di come anche chi appartiene al genere video-ludico dominante possa produrre videogames utilizzando al meglio i concetti del femme game design.

Rimane da chiedersi: sì il mercato sta andando in una buona direzione, ma questi giochi avranno mai la stessa legittimità dei giochi maschili dei tempi andati? Nel 2014 ha fatto notizia il cambiamento dell'audience, quando le donne hanno superato gli uomini nel mondo dei videogiochi: si trattava però di giochi mobile o ancor prima di giochi PC come *Myst* (**Cyan Worlds**) o *The Sims* (**Electronic Arts**), quindi non venivano viste come "vere gamer".
Secondo la Short, più il mercato si diversifica, più i suoi "guardiani" combatteranno. Non solo giocatori, ma anche produttori, designer, venditori o recensori, continuano ad applicare in modo inconscio il proprio gusto personale, quindi oltre ad una maggiore parità di genere, servono nuovi trendsetter per avere un vero e proprio cambiamento culturale.

2.2 La realtà internazionale

Il settore dell'entertainment comprende cinema, Tv, editoria, musica, ma i videogames rappresentano la parte più ampia del settore stesso: nonostante i medesimi caratteri distintivi e realizzativi dei settori sopracitati, il mercato del gaming è cresciuto esponenzialmente nell'ultima decade e nel 2020 ha raggiunto un valore complessivo di 159,3 miliardi di dollari, con un incre-

mento percentuale di 9,3 punti relativamente all'anno precedente. La vetta del mercato non è cambiata: a primeggiare rimane il mercato cinese, che ad oggi è il più grande del pianeta, con il 55% dei giocatori sul totale mondiale, circa 1.5 miliardi. Il mercato video-ludico è particolare: presenta mutamenti e tassi di crescita significativi, che vanno di pari passo con la crescita tecnologica ed i mutamenti che questa subisce di anno in anno. Infatti, ad oggi, non si gioca più solo sulle console classiche o sul PC, ma anche su piattaforme mobile come smartphone o tablet e persino sui social network. La nascita e la crescita parallela di queste nuove "sale giochi", oltre alla sempiterna presenza di prodotti retail, ha favorito una diffusione trasversale per sesso e fasce d'età del medium, spingendo lo sviluppo in direzioni mai pensate: sono usciti prodotti dedicati ad un pubblico sempre più eterogeneo, con nuove tipologie di gioco che si distaccano totalmente dal modo di pensare videogames di qualche anno fa. Sicuramente una maggior professionalizzazione e riflessione sul mondo del gaming ha aiutato in questa evoluzione: lo studio del comportamento dei giocatori, il continuo fondersi di contenuto e tecnologia, ha aiutato gli sviluppatori a creare dinamiche di gioco totalmente nuove, portando l'immersività e l'esperienza ludica su livelli totalmente diversi. L'utilizzo di AI sempre più sofisticate, tecnologie grafiche sempre più avanzate e modelli di narrazione rivoluzionari, hanno contribuito a rendere il videogioco come la forma di intrattenimento totalizzante.
Come è stato detto sopra, l'impatto sul mercato dei dispositivi mobile ha cambiato le carte in tavola: oltre ad avere un ricavo annuo di 77,2 miliardi, rappresentano circa il 48% del mercato globale dei videogames. E questo dato ci porta ad una riflessione su chi gioca, poiché del totale dei videogiocatori, il 46% è rappresentato da donne e con una età media generale che si aggira attorno ai 34 anni, avendo presenze in tutte le fasce di età, nonostante i giocatori dai 18 anni in su siano il 70% del totale. Quindi di quei 159,3 miliardi, il 48% deriva da giochi per smartphone e tablet, confermandosi al vertice di una classifica in cui seguono giochi per console (45,2 mld)ed infine il mercato PC (36,9 mld). L'evolversi continuo di questo medium, ha permesso negli ultimi anni il passaggio dalla distribuzione retail, a forme di distribuzione digitaletramite piattaforme PC e console quali Steam, Epic Games Store, PSN, X Box Live, Apple Store e Google Play. Risulta evidente come la tendenza di mercato sia orientata allo sviluppo di piattaforme digitali diversificate per la pubblicazione dei videogiochi, implementando sempre più piattaforme e modalità di fruizione specifiche: streaming, cloud gaming, distributori e publisher specifici per un determinato genere di gioco.
Tra queste nuove, modalità di fruizione, il Cloud gaming è quello più rilevante a livello di innovazione: definito anche gaming on demand, è un servizio che permette di accedere a contenuti video-ludici in streaming su qualsiasi dispositivo, senza possedere un hardware specifico né una copia fisica del videogioco. Questo espande e rivoluziona totalmente il concetto di "luogo" per giocare: i giocatori possono avere accesso ai videogiochi in ogni momento

e in qualsiasi luogo, permettendo aggiornamenti continui delle piattaforme, senza l'acquisto di hardware costosi. Solitamente il game streaming può essere offerto tramite un servizio in abbonamento o l'acquisto di giochi "on demand" da parte dei giocatori: tra i principali servizi disponibili c'è il Game Pass di Microsoft, Playstation Now di Sony, Arcade di Apple ed infine Stadia, il cloud gaming di Google.

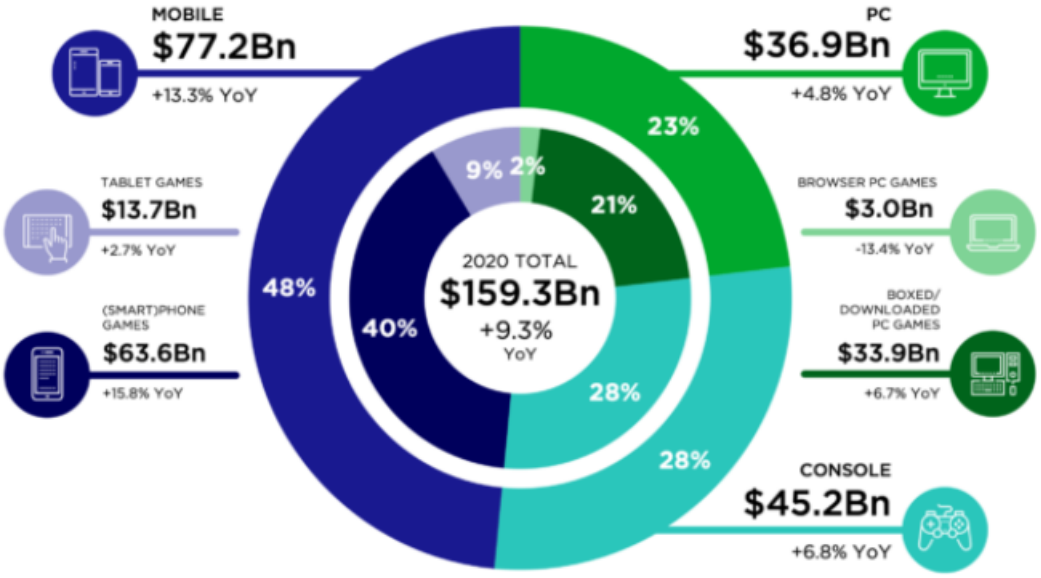


Figura 15: Global Games Report 2020 (5)

2.3 La realtà italiana

Cerchiamo ora di analizzare nello specifico quella che è la realtà italiana e quali possibili differenze ci sono con quella precedentemente descritta. Prenderemo come riferimento alcuni dati derivanti dall'anno 2019: in Italia, il mercato gaming si aggira attorno al valore di 2 mld. Questi dati sono raccolti e forniti da **IIDEA**, ovvero l'associazione che rappresenta l'industria dei videogiochi in Italia, che presenta annualmente lo stato dell'arte del mercato video-ludico: il settore è in crescita di circa 1,7% rispetto alla precedente rilevazione, delineando un giro di affari di **1 miliardo e 787 milioni di euro**. Nonostante sia un anno di transizione in attesa del lancio della nuova generazione console, l'andamento è stato positivo vedendo gli ottimi risultati del comparto software bilanciare le performance in leggero calo fisiologico del mercato hardware.

Infatti proprio il comparto software ha registrato un aumento del 7,3% anno su anno: il segmento, comprensivo di digital download su console, PC ed App per mobile, restituisce un aumento annuale del 13,5%.

> "I risultati che emergono dalla nostra analisi raccontano di un rapporto sempre più consolidato degli italiani con i videogiochi in un momento di una evoluzione significativa del mercato. La dinamicità è la principale caratteristica dell'industria dei videogiochi, che ogni anno è capace di reinventarsi grazie alla sua straordinaria capacità di confrontarsi con sfide tecnologiche sempre nuove e di valicare i confini della creatività. In attesa della nuova generazione di console, il videogioco, che sia un passatempo, un lavoro o una passione, continua a diventare più accessibile e a consolidare la propria posizione come forma di intrattenimento, coinvolgendo un numero di giocatori crescente e diversificato in Italia, come nel resto del mondo". ***Marco Saletta, Presidente IIDEA*** (2)

Il rapporto presentato va a fotografare le classifiche dei giochi più venduti nel nostro paese, sia per console che per PC, che sia formato digital o retail: a livello di classifica aggregata e top 10 videogiochi su console, *FIFA20* guida la vetta seguito dal sempiterno *Grand Theft Auto V* e *Call of Duty: Modern Warfare*; mentre su PC troviamo in prima posizione *Tom Clancy's Rainbow Six: Siege*, con *GTA V* secondo e *Borderlands: The Pre-Sequel* in terza posizione.

> "Il mercato italiano ha mostrato la sua forza e il suo stato di salute anche durante il 2019. Nonostante il rallentamento delle vendite di hardware legato all'annuncio della nuova generazione di console, le buone prestazioni del software (fisico e digitale) hanno confermato come l'attenzione verso i videogiochi sia ormai radicata nei consumatori italiani. È interessante anche vedere i risultati relativi all'acquisto di software digitale, con una crescita significativa durante l'anno passato, in linea con trend simili registrati in altri paesi europei" ***Antonio Rubio,International Sales Director B2Boost*** (4)

Action, *sports games* e *shooter* sono stati i generi più venduti nel 2019, delineando ancora di più che tipo di videogiocatori risiedano in Italia. Negli ultimi dodici mesi, si è stimato che sono circa 17 milioni le persone che hanno giocato con i videogiochi, quindi circa il 39% di tutta la popolazione italiana con un'età compresa tra i 6 ed i 64 anni: di questi il 53% sono uomini ed il 47% sono donne (+1% rispetto al 2018), registrando una maggior diffusione

tra le fasce di età più giovani, concentrandosi maggiormente tra gli 11 ed i 24 anni.

> "I videogiochi in Italia sono cresciuti ulteriormente nel 2019 e oggi raggiungono quasi 4 italiani su 10, che giocano su qualsiasi dispositivo. I circa 17 milioni di giocatori in Italia giocano in media per 7,4 ore a settimana (aumentando anche da 6,7 ore a settimana nel quarto trimestre 2018) a riprova del fatto che il gaming è un mezzo di intrattenimento molto coinvolgente.Questa crescita del numero di giocatori e del tempo dedicato ai videogiochi è stata alla base di un altro anno di successo per l'industria italiana nel 2019 - più giocatori investono più tempo e generano più entrate nell'economia. L'intero ecosistema di gioco è cresciuto a 1,8 miliardi di euro nel 2019, risultati impressionanti tenendo conto dell'attuale fase del ciclo della console, con nuovi dispositivi in arrivo nel prossimo anno" ***Eduardo Mena, Research Director di IPSOS*** (3)

Le piattaforme di gioco maggiormente utilizzate sono i dispositivi mobile come smartphone o tablet (10,5 milioni di giocatori, di cui 5,6 milioni di donne); seguono i PC con 7,8 milioni di persone e le console con 6,7 milioni. Nonostante l'ultimo posto, però sono proprio le console ad avere la media ore di gioco più alta, ovvero 5,4 ore medie a settimana (5,3 su mobile, 4 su PC).

2.3.1 Le principali software house indie italiane

In questa, sezione verranno riportati i principali studi italiani che hanno fatto sentire la propria voce nel mondo del gaming (all'elenco mancherà IVProductions, poiché trattata più ampiamente nel capitolo successivo).

Artematica Fondata nel 1996 a Chiavari, nasce da un'idea di Riccardo Cangini, designer ed autore di videogames con molti anni di esperienza nell'industria di settore ed uno dei precursori del fare videogiochi in Italia. Lo studio produce giochi per console(Playstation 2, Playstation 3 e PSP), PC e mobile device, ma soprattutto è pioniere nel *advergaming*, ovvero quei videogiochi creati per trasmettere messaggi pubblicitari.
Grazie alla partnership con Ubisoft, Warner Bros e Disney, i giochi di Artematica sono distribuiti in tutto il mondo; inoltre sono anche produttori di avventure grafiche, giochi tipici degli anni 90 tornati in auge nel nuovo millennio proprio grazie alle implementazioni tecniche sulle interfacce grafiche. Nel corso degli anni ha vinto numerosi premi grazie a titoli come *Druuna: Morbus Gravis* (premiato come miglior gioco multimedia al Lucca Comics

Award), o come *Martin Mystère - Operation Dorian Gray* (miglior gioco italiano al Alta Badia Award) ed infine *Diabolik: The Original Sin* (miglior videogame al Play Mont Festival International Multimedia Award). Nonostante questa brillante carriera, Artematica chiude i battenti per fallimento nel 2006.

Figura 16: Artematica Entertainment

Milestone Milestone viene fondata a Milano nel 1996 e ad oggi è la più grande compagnia produttrice di videogame per PC e console, grazie all'incredibile qualità dei suoi prodotti. La specialità della casa sono videogames basati sulle corse di auto, moto, sia su strada che fuori strada: questo deriva dalla grande passione del team per il mondo delle corse, arrivando a realizzare videogiochi davvero di alta qualità, anche grazie alla collaborazione strettissima tra gli sviluppatori e gli ingegneri delle più grosse case produttive di moto e auto, garantendo al giocatore un game feel davvero realistico.

Tra i migliori lavori realizzati, possiamo ricordare *MotoGP 13*, *MotoGP17*, *MXGP The Official Motocross Videogame* e *WRC 4 FIA World Rally Championship*; L'ultimo prodotto di successo è *Monster Energy Supercross - The Official Videogame*, disponibile su PS4, Xbox One e Windows PC/STEAM.

Figura 17: Milestone

Ovosonico Nata nel 2012 da un incontro tra Massimo Guarini, creative director ed esperto del settore video-ludico, e Gianni Ricciardi, audio director per sviluppatori di videogiochi, è una delle aziende italiane maggiormente coinvolte nella produzione di giochi su console. La filosofia dello studio è quella di raggiungere il proprio pubblico attraverso emozioni e sensazioni, attraverso l'utilizzo di immagini e la condivisione tra tutti i giocatori, creando un canale di connessione comune. Il reale successo dell'azienda arriva nel 2014 con il puzzle sviluppato per Playstation Vita, *Murasaki Baby*. Nell'estate del 2017 arriva il loro ultimo lavoro, ovvero *Last Day Of June* pubblicato da 505 Games su PS4 e PC: il titolo è stato molto apprezzato e ha vinto diversi premi, anche a livello internazionale. Ad oggi Ovosonico non esiste più in quanto tale, ma è stata acquisita dal gruppo **Digital Bros.** ed è stata ribattezzata **AvantGarden S.r.l.** .

Figura 18: Ovosonico

34BigThings Studio indipendente fondato nel 2013 che nonostante i pochi anni di vita, conta tra le sue fila diversi professionisti da qualsiasi ambito utile allo sviluppo dei videogiochi: game designers, 2D/3D artist, programmatori, concept artist, che condividono la volontà di creare prodotti qualitativamente elevatissimi. Non a caso, lo studio di Torino ha vinto il Drago D'Oro per il miglior gioco italiano nel 2017 con *Redout*.

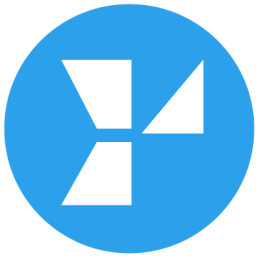


Figura 19: 34BigThings

Forge Reply Si è distinto in pochissimo tempo grazie all'utilizzo delle più moderne ed innovative tecnologie presenti sul mercato. Fondato nel 2011, Forge Reply ha acquisito diversi professionisti del settore calandoli in una realtà all'avanguardia e proponendo sviluppo di progetti anche non tradizionali: l'utilizzo della AR e VR dedicate al mondo dei videogames, o anche per applicazioni sociali, è sempre stato un punto chiave della loro produzione.
Gli ultimi lavori prodotti dallo studio sono *In Space We Brawl*, un action game competitivo sviluppato per PS3, PS4, Xbox One, PC e Mac; *Joe Dever's Lone Wolf* sviluppato per le stesse piattaforme e anche per Android, mentre l'ultimo lavoro è l'innovativo *Theseus*, un adventure game basato sul mito di Teseo ed il Minotauro, riadattato in chiave più dark, è l'emblema delle capacità professionali dello studio in ambito VR: il giocatore infatti verrà immerso totalmente nel mondo di gioco, calcando la mano sull'atmosfera.

Figura 20: Reply Forge

Storm in a Teacup Sull'onda dell'improvviso sviluppo delle software house italiane, lo studio nasce nel 2013 grazie a Carlo Ivo Alimo Bianchi, un grande CG artist con tantissimi anni di esperienza in grosse case di sviluppo come Ubisoft, IO Interactive e MetricMinds. La prima opera di Storm in a Teacup è *NERO*, un gioco con un particolare stile artistico, sviluppato su PS4, Xbox One e PC, che ha avuto un discreto successo.
Il team contiene un mix di differenti skill artistiche e competenze derivanti da culture e paesi differenti: questa particolarità rende unici i prodotti che

vengono rilasciati. Nel 2017, la Gamescom viene presentata con la demo del loro ultimo lavoro *Close to the Sun*, un horror game ambientato in un oscuro complesso navale creato da Nicola Tesla: quest'anno ha vinto il premio miglior gioco italiano 2020 all'Italian Games Award 2020 di Pisa.

Figura 21: Storm in a Teacup

MixedBag Un piccolo studio nato dal talento del graphic designer Andrea Gellato e dal programmatore Mauro Farinelli: fondato nel 2011 con l'idea di creare un prototipo di adventure game per il sistema IOS, fu il primo studio italiano a pubblicare un titolo su PS4. Questa realtà gioca un ruolo importante per quanto concerne lo sviluppo del gameplay e delle meccaniche VR, oltre che nello sviluppo di giochi innovativi su ogni piattaforma.
Le produzioni celebri includono *Secret Oops!*, lanciato su IOS Arcade, *Futuridium EP Deluxe* sviluppato inizialmente su PS4 e PS Vita, e poi lanciato anche su Steam OS, Windows, MAC, infine *Forma.8*, che regala un intero mondo da esplorare con uno stile visivo e grafico davvero particolare.

Figura 22: MixedBag

Tiny Bull Studios Studio indipendente fondato a Torino nel 2011 da Matteo Lana e Rocco Luigi Tartaglia, entrambi specialisti in VR, iniziarono a lavorare al loro primo progetto nel mondo gaming. Se inizialmente si proponevano di fare giochi solamente per console, hanno poi cambiato idea a favore della piattaforma IOS, quindi per mobile. Ma dopo qualche tempo, i Tiny Bull Studios si sono ritrovati a sviuppare per mobile, PC ed anche console: in pochi anni di attività, possono contare 28 dipendenti nel loro team.
Dopo diversi prodotti lanciati sul mercato, lo studio presenta *Blind*, un pro-

dotto interamente creato per la realtà virtuale, e *Game Omen Exitio: Plague*, un gioco che richiama lo stile dei libri gioco di fine 1800.

Figura 23: Tiny Bull Studios

3 Fare Videogiochi

Lavorare nel campo dei videogiochi è complicato: la molteplice natura dei ruoli e delle realtà in cui si va ad operare, spesso richiede una flessibilità sia tecnica, sia mentale. Come è stato già evidenziato nei precedenti paragrafi, a seconda della grandezza dello studio e dal budget medio di produzione, ci saranno diversi obiettivi e possibilità di progettazione: in Italia, dove il mercato produttivo è ancora acerbo, difficilmente si superano le produzioni da un milione di euro, mentre all'estero questo limite è ampiamente superato. Ad esempio, uno studio su tutti, **Rockstar Games** ha sedi in diverse città del mondo (Bogotà, New York, Londra e Parigi) ed è stata fondata nel 1998 da **Take Two Interactive**; è conosciuta in tutto il mondo per la serie *Grand Theft Auto* con la quale ha venduto da sola 220 milioni di copie (dato risalente al 2014, quindi risulta sicuramente minore di quanto in realtà sia oggi) e solamente con il quinto capitolo, dotato di un'immensa componente online, ha fatturato dal 2013, quasi 9 miliardi di dollari (6). Numeri impressionanti possibili grazie alla mole immensa dello studio e alla preparazione tecnica di chi ci lavora dentro.

Figura 24: Rockstar Games

Ecco tali numeri nel nostro paese sono difficilmente raggiungibili al momento, ma comunque esistono realtà che stanno spingendo per arrivare a competere con il mondo fuori dai nostri confini.
Ma cosa significa esattamente fare videogame in Italia? Viene da chiedersi quali sono i punti di partenza, gli step da seguire e se esiste una ricetta ottimale: ogni realtà ed ogni progetto ha una propria specificità, che si potrebbe dire unica, e gli sviluppatori, come gli alchimisti un tempo, ricercano ancora la propria tavola "mendeleeviana". Provando a rispondere a tale domanda, serve analizzare chi i videogiochi li fa da anni proprio in Italia: nel prossimo paragrafo, verrà presentata la IVProductions e quali sono i passi da compiere per lavorare nel mondo dei videogiochi.

3.1 Ivan Venturi e IVProductions

Parlando di case di sviluppo italiane del panorama indipendente, è impossibile non fare riferimento a **IVProductions** e al suo fondatore Ivan Venturi: nato a Bologna nel 1970, inizia a lavorare nel mondo dei videogiochi in giovane età, co-fondando la prima casa di sviluppo di videogiochi italiana, **Simulmondo** assieme a Francesco Carlà e lavorandoci fino al 1993.

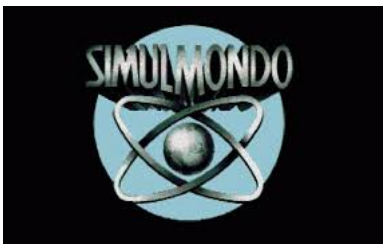


Figura 25: Simulmondo

Nel 1994 fonda **Colors** dove tenta di proporre una serie di videogiochi da edicola, in cui il binomio gioco-fumetto avrebbe dovuto formare un *transmedia storytelling* ante litteram, poiché il mondo del gioco e quello del fumetto erano gli stessi. Lo studio ha prodotto dozzine di giochi per PC fino al 1999.

Figura 26: Colors Arti Multimediali

Il lavoro di Venturi come produttore di videogames e sviluppo aziendale, riprende ufficialmente nel 2003 con la fondazione di **Ticonblu**, precedentemente chiamato **Koala Games**, parte del **Gruppo Armonia**, con il quale ha prodotto moltissimi serious games e giochi educativi fino alla pubblicazione con Microids di **Nicolas Eymerich, Inquisitor**; tramite questa fase lavorativa riesce a costruirsi una nicchia di mercato basata sul software educativo indirizzato verso la pubblica amministrazione, garantendosi quindi una domanda lavorativa continua ed un mercato capace di assorbire la produzione: alcuni esempi virtuosi sono *GuidaTu* (2010), una scuola guida virtuale, ed il simulatore di disabilità, che pone l'utente nei panni di un disabile e simula difficoltà legate al movimento o alla percezione sensoriale tramite la manipolazione dello schermo.

Un ulteriore prodotto di livello è *The Invisible Hand* (2012), sviluppato in collaborazione a 4 ONG, ovvero un adventure game che mette in luce le storture del sistema economico e lo sfruttamento nascosto dietro la produ-

Figura 27: TiconBlu

zione di merci di uso comune.
Nel 2013, Venturi fonda lo studio di produzione IVProductions: tra i suoi prodotti cardine spuntano *Albedo: Eyes from Outer Space*, un escape room 3D che recupera l'ambientazione e l'immaginario della fantascienza anni '60/'70, il gioco di ping pong per realtà virtuale *VR Ping Pong Pro* (PS4, Steam) o anche il simulatore di rivolte *Riot - Civil Unrest*, disponibile per Microsoft Windows.

Figura 28: IVProductions

Quest'ultimo nasce da un'idea di Leonard Menchiari, precedentemente editor e direttore alla fotografia per Valve, che ha il fine di far provare al giocatore diverse situazioni di scontro tra manifestanti e forze dell'ordine realmente accaduti negli ultimi anni; infatti il gioco propone diversi scenari, quali Spagna, Egitto, Grecia e Val di Susa, con l'intendo di produrre un newsgame in grado di far capire le dinamiche di tali avvenimenti, mettendo in luce le motivazioni che spingono le due parti in causa: scegliendo uno dei due schieramenti, il giocatore dovrà guidare la manifestazione o tentare di arginarla decidendo se assumere un atteggiamento pacifico o sfociare nello scontro fisico. Il gioco, in base alle azioni dell'utente, proporrà situazioni differenti nel livello successivo: ad esempio, un uso eccessivo della violenza porterà ad un inasprimento del conflitto tale che si potrà passare allo stato di guerra civile.

Un'altra produzione di rilievo della IVP è il serious game **Progetto Ustica**, un progetto che prevede l'utilizzo dei nuovi strumenti di realtà virtuale per la simulazione degli ultimi minuti del volo IH870, sviluppato collabo-

Figura 29: Uno screen in game di *Riot - Civil Unrest*.

rando con l'Associazione Parenti delle vittime della strage. Progetto Ustica pone il giocatore all'interno del DC9 poco dopo il decollo, proponendo una visuale in prima persona: bisogna esplorare la cabina ed interagire con persone ed ambiente, cercando oggetti per sbloccare nuove funzionalità. Gli oggetti in questione consentono di accedere a nuovi punti di vista intercambiabili dalla posizione degli aerei militari coinvolti, dallo schermo radar che va a rappresentare l'affollamento del cielo in quei momento o dal punto di vista di un pescatore che osserva il cielo dal basso. Nel gioco sono rintracciabili le trascrizioni delle comunicazioni radio di quella notte fra il DC9 e le torri di controllo contattate durante il volo; è possibile acquisire informazioni contestuali sull'attività militare nel Tirreno a cavallo del 1980, indizi sugli eventi di quella notte e su ciò che è avvenuto in seguito. Progetto Ustica non si propone né come una ricostruzione scientificamente accurata dei fatti, né come un gioco investigativo per ricercare la verità: è un gioco per la memoria, dove alcuni dettagli dello scenario sono veri, mentre altri verosimili, lasciando il compito ai giocatori di approfondire. Il gioco è stato pubblicato su MAC e Windows.

Figura 30: Capsule imagine di Steam di *Progetto Ustica*.

Una delle ultime produzioni della IVP, e di tutt'altro genere, è **Pride Run**: vincitore di due premi alla Game Connection di Parigi del 2018 (Most Original Game e Best Casual Game), il gioco celebra tutto ciò che fa parte del mondo Pride e LGBTQ+. Il giocatore dovrà guidare la parata in diverse città chiave in giro per il mondo, cercando di proporre divertimento, colore e quel senso di festa tipico del mondo Pride. La versione "Vanilla" del gioco è riconoscibile come un Rhythm Action game, facendoti tenere il ritmo con la colonna sonora composta da Hard Ton (dj vicino alle cause LGBTQ+): più si starà sul beat della musica, più gente si unirà alla parata, se invece si perderà il ritmo, la folla inizierà a disperdersi. Alla fine del porcorso della parata, partirà uno scontro con il boss finale del livello: solitamente interpretati da personalità che mettono in dubbio i diritti del mondo LGBTQ+, come ad esempio Trump o Putin. La modalità "Play Hard" aggiunge un tocco RTS unico al gioco, proponendo elementi più tattici: si possono personalizzare le parate scegliendo i propri leader ed usare ad hoc le loro abilità per amplificare il potere inclusivo della parata stessa o difendersi dagli attacchi nemici; inoltre è possibile muovere i gruppi in tempo reale per assicurarsi le capacità più efficaci nel momento e posto più corretto.
Entrambe le modalità di gioco possono essere giocate come giocatore singolo o in cooperativa locale, con tre livelli di difficoltà, ed ogni città avrà eventi e personaggi unici, specifici del luogo. Il gioco è stato pubblicato su Steam, sia in versione Windows che MacOS.

Figura 31: *Pride Run*

Per concludere, questa carrellata di produzioni della IVP, possiamo citare anche **Albedo: Eyes from Outer Space**, come già scritto in precedenza, richiama e miscela con sapienza il look da fantascienza low budget degli anni 60/70 e meccaniche old school, riuscendo a stupire il giocatore fin dalle prime fasi dell'avventura. Sicuramente nel panorama italiano è un gioco anomalo, un corpo "estraneo", modellato da Fabrizio Zagaglia, esperto programmatore del settore, attraverso l'uso della logica, cercando di puntare forte su atmosfera, divertimento e sfida; è definibile come anomalo proprio per la sua natura ermetica ed impegnativa, in un mercato che punta alla semplificazione ed alla facilità.

Figura 32: Cover di gioco di *Albedo: Eyes from Outer Space*.

Il protagonista del gioco è John T. Longy, guardiano notturno del centro di ricerca Jupiter del gruppo Olympus, che dopo un turno di lavoro viene sbalzato in cantina da una misteriosa esplosione: si risveglierà circondato da disgustosi bulbi oculari extraterresti e, affidandosi al suo ingegno, dovrà trovare una via di fuga dal centro di ricerca e far luce su quanto accaduto. Come il protagonista, anche il giocatore dovrà aguzzare l'ingegno e affrontare questa situazione, senza nessun sussidio esterno: è qui che emerge la forza di *Albedo: Eyes from Outer Space*, il mettere davvero in difficoltà il giocatore, lasciandolo solo con la propria intelligenza, come il protagonista del gioco.

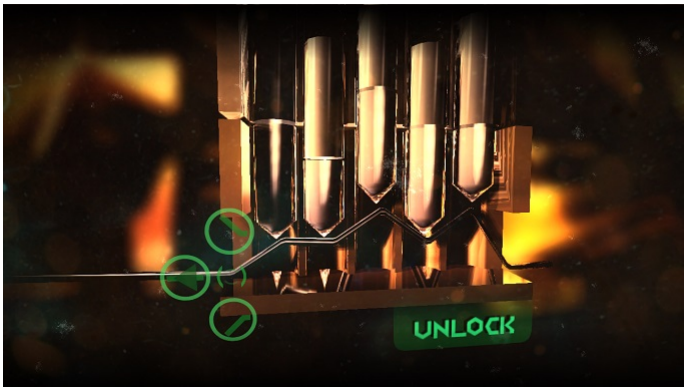


Figura 33: Esempio di enigma in game

3.1.1 Svilupparty

Oltre alla produzione di videogiochi, Venturi si occupa anche del sostenimento e della diffusione della cultura del videogioco in Italia, con particolare attenzione al supporto delle attività produttive del nostro Paese. Per raggiungere questo scopo, ha fondato **Svilupparty**, in collaborazione con l'Archivio Videoludico di Bologna: è un evento annuale che vuol riunire tutte le realtà produttive italiane, tentando di promuoverne il lavoro e creare una rete inter-collaborativa di conoscenze ed esperienze tra i vari studi e professionisti del settore.

Figura 34: Logo di Svilupparty

Nato ufficialmente nel 2010, sul solco tracciato da Simulmondo Party anni prima, è un evento promosso dall'**Associazione Italian Party of Indie Developers - IPID** che solitamente si divide in tre giornate in cui sia rappresentanti del settore che appassionati possono incontrarsi e parlare dei propri progetti o direttamente raccontarsi al pubblico; tutto questo è permesso grazie ad IPID, un'organizzazione apartitica e no-profit che raggruppa sviluppatori, professionisti ed amatori, operando a sostegno della valorizzazione delle produzioni delle realtà videoludiche del nostro paese, sia a livello di volumi che a livello di qualità.

Figura 35: Una foto dalla convention 2019

Per raggiungere tale scopo, l'Associazione ha costruito e mantiene una rete di contatti tra sviluppatori, in modo da condividere progetti, esperienze e conoscenze, proponendo programmi di formazione e pubblicazione/distribuzione dei prodotti dei propri associati e, naturalmente, garantisce la cadenza annuale dell'evento a Bologna. Attraverso il tesseramento, si sostengono le attività dell'Associazione e la realizzazione di Svilupparty. Non solo si contribuisce all'organizzazione concreta dell'evento, ma se ne condividono e promuovono gli obiettivi: valorizzare il lavoro degli sviluppatori italiani ed offrire loro visibilità, luogo d'incontro e confronto.

Figura 36: Italian Party of Indie Developers

Svilupparty 2020 L'edizione 2020 di Svilupparty, con il nome di *Svilupp4arTPO 2020*, come tanti altri eventi, ha subito modifiche nella sua struttura e svolgimento a causa della pandemia in corso: l'accesso era consentito solamente agli sviluppatori, ai collaboratori stretti e non al pubblico degli appassionati; ovviamente sono state prese tutte le misure di prevenzione utili per il periodo, quindi l'utilizzo di mascherine e di igienizzante ad ogni varco di accesso. Si è svolto al TPO di Bologna per una singola giornata, il 5 settembre scorso, proponendo una serie di talk dalle dieci del mattino fino all'ora di cena, oltre alla presenza fissa di stand popolati da diversi team di sviluppo intenti a presentare il proprio gioco e a farlo provare a chiunque volesse prestarsi ad una mini sessione di testing.
Da sottolineare la presentazione di un motore grafico innovativo, al cui sviluppo sta lavorando Frederic Naar, ovvero **Unigine**: molto più potente di Unity, ma più accessibile dell'Unreal Engine, si propone come un motore grafico molto versatile e completo sotto ogni punto di vista. Quello che colpisce di più è il fermento e l'atmosfera costruttiva dell'evento: è possibile creare contatti, avere consigli lavorativi e confronti sul mercato video-ludico italiano; si respira condivisione e la possibilità di formare legami lavorativi per la produzione di nuovi progetti è assolutamente reale.

3.1.2 Italian Games Factory

Un'ulteriore realtà da tenere presente per lo sviluppo video-ludico italiano è **Italian Games Factory**, nata dal connubio di due realtà differenti e complementari, ovvero IVProductions e *iMasterArt*, realtà leader nell'ambito della didattica rivolta all'intrattenimento con Master specialistici e unici nel genere, che vanno dal concept 2D fino alla produzione di cinematiche full CG, videogame e post produzione cinematografica.
Grazie a queste due realtà, i prodotti sono sviluppati secondo standard commerciali di elevata qualità, insieme a divertimento, concretizzazione nell'uso

e fascinazione sull'utente finale. La realizzazione di videogiochi è indirizzata a tutte le piattaforme (pc, console, mobile e web) e a tutte le categorie merceologiche (platform, serious game, adventure, action, simulatori, strategici, casual, casual,...), attraverso l'utilizzo di tecnologie web, visual FX, realizzazione ed uso di hardware ad hoc.
Parallelamente a questo settore di entertainment, è presente un particolare focus su tematiche scientifiche, sociali e culturali: questo dà modo di accrescere direttamente l'esperienza con scuole e pubblica amministrazione, basandosi sulla collaborazione con esperti in ambito psicologico e pedagogico. Il pubblico a cui IGF si rivolge è molto ampio, infatti le tematiche di interesse hanno portato nel corso degli anni a specializzarsi nello sviluppo di app e videogames accessibili per non vedenti, ipovedenti, dislessici, persone con disabilità motorie/cognitive, progettando ad hoc ogni particolarità grazie alle collaborazioni con esperti ed istruzioni. Significa quindi progettare una difficoltà ed un'interattività tarate per un pubblico non solamente di videogiocatori e videogiocatrici, bensì anche di utenti mobile e web.

Figura 37: Italian Games Factory

3.2 Fare Videogiochi in Italia

Molto spesso, a differenza delle esperienze riportate nella sezione sopra, si formano team principalmente per passione: si condivide l'amore per i videogiochi e si decide di imbastire un progetto. Purtroppo questi team non raramente "muoiono" dopo pochi mesi di lavoro. Senza fraintendimenti, la passione è importante ma non basta. Bisogna guadagnare. Fare soldi. "Portare a casa la pagnotta" si direbbe in gergo. Il concetto è chiaro: bisogna passare dalla passione al mestiere, poiché spesso si rischia di rimanere ingannati dall'entusiasmo e si tralasciano tutti quegli aspetti concreti del lavoro. Non ci si deve mai scollegare dalla realtà.
Soprattutto nella realtà italiana, si ha una componente **artigianale** anche nel creare i videogiochi, ma non bisogna lasciare offuscare quella più **lavorativa** e **manageriale**. Proprio in Italia sono attive associazioni e festival che aiutano queste piccole realtà a crescere. La prima, già analizzata, è Svilupparty, ovvero il festival di 3 giorni che si tiene solitamente a Bologna, che nasce dalla necessità di incontrarsi e fare gruppo per scambiarsi informazioni e collaborazioni.
La seconda è **IIDEA**: è un'associazione delle aziende che intercede direttamente con il governo italiano per discutere dell'aspetto istituzionale del

mercato gaming ed inoltre rappresentano una delegazione a cui aderire per presentarsi alle **GameConnections**.

Figura 38: Italian Interactive Digital Entertainment Association

Ogni progetto ha una sua merceologia specifica e dimensioni ben precise: esistono infatti dei costi standard minimi! Nel mercato odierno ci sono aspettative che vanno rispettate ed è per questo che risulta chiave l'analisi dei competitor: non posso presentarmi sul mercato con un prodotto scadente o con una qualità base non opportuna. C'è bisogno di soldi, persone e risorse per raggiungere quest'obiettivo; il business del gaming è complicatissimo: oltre al gioco vero e proprio ed alla sua produzione, vanno gestiti l'ambito amministrativo e quello commerciale. E no, non sono secondari se ve lo steste chiedendo: sono fondamentali anche in sede di design, perché poi il gioco va venduto, non bisogna dimenticarlo.

HIDDEN OBJECT X

DIRECT COSTS per SINGLE GAME	**Min Unit**	**(WD) per unit**	**WD x min**	**Single WD cost**	**Cost**	**Professional**
market research	1	1	2	200	400	PRODUCER
production/design document	1	5	5	200	1000	PRODUCER
Storyboard/dialogues/texts	1	22	22	100	1100	STORYBOARDER
Environments PhotoImage search/first work	40	0,5	20	100	2000	STORYBOARDER
Environment Graphics part 1	40	3	120	150	18000	2DARTIST
Minigames graphics part 1	20	2	40	150	6000	2DARTIST
Coding/Assembly minigames/environments	1	40	40	200	8000	JR PROGRAMMER
Cover Image/icon	1	1	1	250	250	2DARTIST
Site / FB page	2	1	2	200	400	WEB PROGRAMMER
English translation	2	5	10	500	5000	TRANSLATORS
Sounds	2	1	2	200	400	MUSICIAN
Musics	3	1	5	200	1000	MUSICIAN
ENG dubbing	2	1	2	200	2000	DUBBING STUDIO
Scrapbook graphics contents	20	0,25	5	100	500	2DARTIST
33% AMORTIZATION (max)					25000	
2D Graphics 2D GUI, menu	1	5	5	100	500	2DARTIST
Subversioning (SDKs ecc)					4000	SR PROGRAMMER
testing	4	1	4	75	300	TESTER
total					75850	

COSTS in AMORTIZATION	Min	Max	
company amortization	5000	5000	
IDEATION & SAGA GUIDELINES	4000	5000	
HH engine	15000	30000	SR PROGRAMMER
Minigames templates	2500	10000	
Saga Producer	5000	20000	
SAGA graphics	1500	5000	2D ARTIST
total	33000	75000	
TOTAL INVESTMENT (MIN HYPOT)	227550		
PRODUCER COST	38700		
EXTCOSTS per TITLE	42300	126900	

Figura 39: Esempio di costi standard per la produzione di un hidden object game.

Infatti non bisogna usare termini generici o poco rappresentativi perché il publisher (o l'utenza in genere) può disinteressarsi, anzi in questi casi lo fa sempre. Persino nelle pagine di presentazione Steam, ad esempio, basta guardare i font usati nelle capsule image per capire che tipo di gioco mi troverò davanti: un *rpg* (*role play game*) classico a turni fantasy-oriented avrà un tipo di font ed un'immagine ben specifici, un simulatore di bus ne avrà un altro. Tutto deve comunicare l'identità del gioco. Quasi fosse la sua carta di identità allargata. Se la personalità del mio gioco non è chiara, tanto meno potrà esserlo la pagina di presentazione su una piattaforma online.

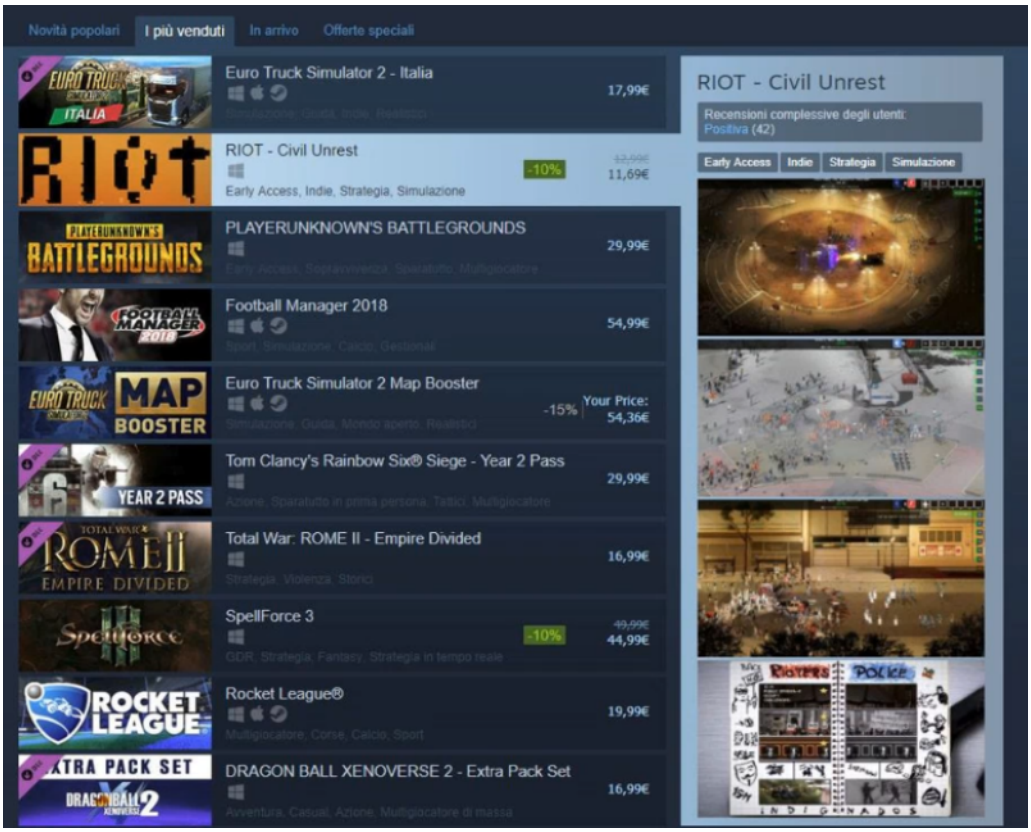


Figura 40: Ecco come si presenta una pagina Steam: ogni gioco deve comunicare la propria essenza tramite image capsule, font, prezzo e tags.

3.3 Una Simulazione: da hobby a mestiere

In questa sezione si cercherà di raccontare tramite una sorta di simulazione, i passi per arrivare ad essere degli sviluppatori veri e propri e non solo degli appassionati.
Immaginiamo di voler creare un videogioco: come fare a partire? Quali sono le domande da porsi? Innanzitutto si parte da una risorsa imprescindibile, ovvero la **determinazione**: è ciò che conta di più, ciò che risulta essenziale per lavorare; ovviamente non basta da sola, poiché c'è bisogno anche di avere le capacità e le skill adatte al lavoro che si vuol fare.

Chi siamo? Rispondendo a questa domanda, definiamo l'identità del team e di chi lavorerà al progetto: in questa simulazione, supponiamo di avere un gruppo di amici che seriamente decidono di produrre un videogioco e dopo una "pizzata" insieme si decide di produrre un videogame.
Ognuno dei componenti del gruppo ha background e skill differenti che ovviamente influenzeranno il tipo di progetto che verrà realizzato; più le abilità in gioco copriranno un campo conoscitivo ampio e più ci saranno possibilità di sperimentare e di risparmiare su costi diretti per lavori esterni al team. Solitamente un publisher non finanzia mai un team alle prime armi: spesso vengono cercati a caso, chiedendo stipendio pieno, appoggio...al 99% la risposta è no.
Bisogna quindi riflettere sulle risorse disponibili ad inizio del progetto e concentrarsi sulla questione publisher a tempo debito: anche perché la parte più complessa spesso risulta la coda della produzione.

Quanto tempo abbiamo? Il tempo è denaro. Ogni componente del team ha la sua vita, magari già con lavori avviati o studi ancora da completare, quindi va valutato bene quanto tempo si può dedicare al progetto. C'è disponibilità di cinque ore al giorno? Si può lavorare lavorare di più? Di meno? Sono tutte cose da valutare prima di partire con il progetto vero e proprio.

Cosa fare in termini concreti? Dopo aver stabilito i primi passi, almeno in teoria, ci si trova a fare i conti con la scelta delle tecnologie da utilizzare: su quale piattaforma uscirà il gioco? Che tecnologia verrà utilizzata per produrlo?
Allora si ponga il caso di fare un gioco sia per mobile che per PC, lavorando su Unity. Il videogioco avrà infatti un prezzo, un pubblico ed un canale di vendita: la scelta della piattaforma determina proprio questi elementi ed in questa situazione, si supponga di puntare ad un utente europeo/nord americano, con un certo grado culturale e che utilizza parecchio il mobile.

Dove si giocherà il gioco? Sicuramente la versione mobile, che è quella a cui verrà data la precedenza, sarà giocata in luoghi comuni e rilassanti: il divano, magari in bagno, a letto. Proprio per questo motivo nei giochi mobile l'attenzione per la UI è fondamentale: se per esempio, si sceglie un layout di tipo *portrait*, la parte utile per gli input sarà in basso, mentre quella a cui l'utente deve prestare attenzione sarà in alto. Il game design è un tutt'uno con utente e piattaforma: occorre effettuare delle prove di stile, meccaniche, grafica, insomma risulta necessario il *prototyping*.

Quanto dovrà costare? Il prezzo va stabilito. Essendo prodotto per il mondo mobile, va fatta un'analisi dei competitor e delle realtà circostanti presenti sul mercato. Fondamentalmente su mobile esistono due mondi: IOS e Android; sono realtà completamente opposte, una basata sul mercato *premium* (a pagamento) e l'altra più *free*, quindi con una maggiore presenza di titoli gratis. Oltre a questi due soluzioni, si può optare per una via di mezzo, ovvero quella *freemium*: il gioco è scaricabile gratis in versione demo e se si vuole giocarlo completamente, lo si può acquistare successivamente. In ogni caso il prezzo da stabilire va pensato dopo un'oculata analisi dello stato del mercato: che prezzi hanno gli altri giochi? Che modalità di acquisto/download hanno utilizzato gli altri team di sviluppo? Che qualità hanno? Non si può assolutamente pensare di lanciare sul mercato, un titolo che non rispetti gli standard qualitativi dei suoi diretti concorrenti.

Come si racconta il gioco? Bisogna avere un'idea chiara dell'identità del gioco per poterlo presentare: questa viene raccontata in un documento ben precisa, utile per far capire a grandi linee che tipo di gameplay avrà, di cosa parlerà, chi lo sviluppa, etc.
Tale documento è chiamato **trade info** e rappresenta una potente arma di comunicazione per il team di sviluppo, sia nei confronti del pubblico sia nei confronti di un possibile publisher o distributore: vanno curati con precisione i font, le immagini di gioco da mostrare, l'image capsule ed, ovviamente, la parte testuale. Risulta molto efficace gestire e produrre in maniera dettagliata e puntuale una scheda che racchiuda in maniera perfetta la personalità del gioco: sta alla base del game design, poiché senza idee chiare non si ha la possibilità di avere successo.

Si supponga di fare un salto temporale: dopo 3 o 4 mesi di lavoro, si arriva allo sviluppo di un prototipo funzionante e lo si vuole mostrare a qualcuno. Solitamente si comincia con amici e parenti: farlo testare ai propri legami può essere utile, ma anche fuorviante. Meglio organizzare sessioni di testing con utenti che non sono per nulla legati al team di sviluppo, poiché da lì si otterranno i migliori feedback per lo sviluppo del prodotto finale. Una volta terminata questa sessione, si passa alla produzione: i lavori entrano nel vivo e si cerca di ottenere un prodotto valido per la vendita. Viene fatto fare

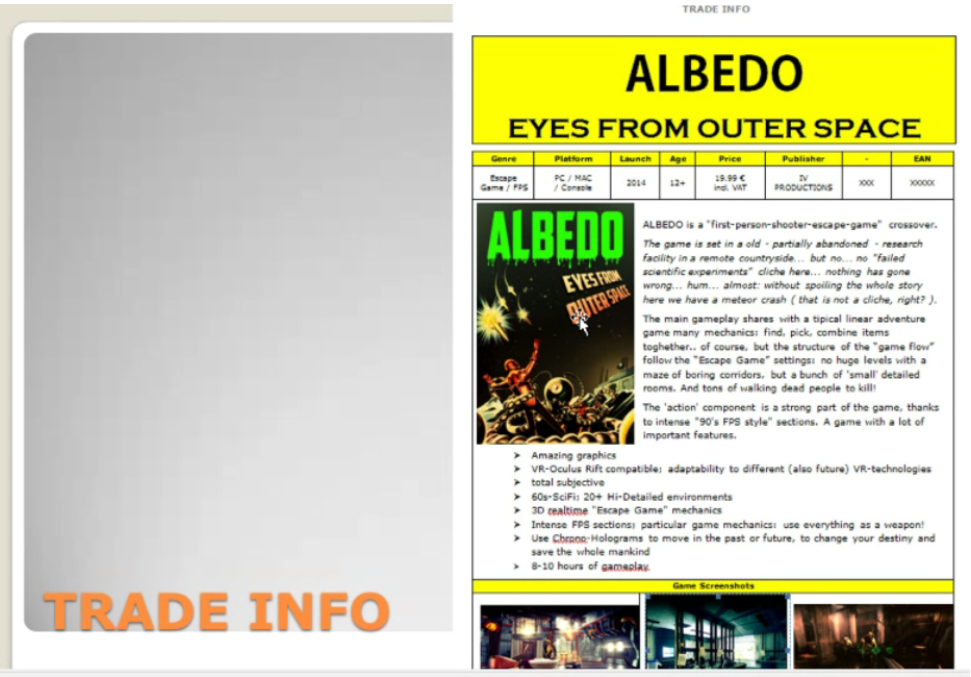


Figura 41: Trade info di *Albedo:Eyes from Outer Space*.

altro *testing*, ma stavolta da parte di altri sviluppatori che possono regalare consigli molto utili per cancellare errori ed attuare miglioramenti importanti. Dopo questi passaggi si supponga di avere una **beta** del gioco, ovvero una versione funzionante e quasi completa del gioco: fino a questo punto il team si è autofinanziato e si è basato sulle proprie forze, sia economiche, che lavorative, quindi ora bisogna far parlare del gioco tramite fiere, eventi legati al gaming o al mondo dei media.
Nel frattempo si comincia a lavorare anche alla versione PC del titolo, in modo da metterlo in vendita dopo aver lanciato la versione mobile.

Pubblicazione Arriva il momento di pubblicare la versione mobile su IOS; Apple decide di *featurarlo*, ovvero sponsorizzare il gioco nelle sue chart: fa successo ed si ottiene qualche buon risultato in termini economici. E su PC? La cosa si complica perché i costi sono molto alti: produrre una versione internazionale di un gioco costa circa 70.000 €. Serve un publisher: si decide mandare due elementi del team alla Game Connection di Parigi ad Ottobre (oppure a quella di San Francisco a Marzo). Lì si riesce a trovare un accordo con un publisher: propone di pubblicarlo anche su console a prezzo maggiorato e rilancia più soldi di quelli che il team ha chiesto. Con questo "boost" economico (più le entrate derivanti da lavori di porting esterni), il team amatoriale decide di costituire una società (s.r.l.s.) e affitta un ufficio. Dalla "pizzata" iniziale è passato un anno e mezzo.

Iniziano arrivare i soldi dalle vendite su mobile, su PC e dalle *royalties* del publisher. E ora? Si decide di cominciare il progetto del secondo gioco: ecco che il team entra di fatto nel mestiere dello sviluppo video-ludico. Dopo la pubblicazione del secondo titolo, che quasi sicuramente verrà meglio del primo, si è attivi nel mercato dei videogames: in 3 anni circa, i componenti del team sono diventati degli sviluppatori a tutti gli effetti.

Nelle successive sezioni, verranno spiegate in maniera più approfondita e particolareggiata, le varie fasi dello sviluppo di un videogioco.

3.3.1 Sviluppo I: Pre-Produzione

La creazione di un videogame è un processo assai complesso che coinvolge figure professionali differenti negli ambiti di design, programmazione e marketing; per ottenere un prodotto che sia allo stesso tempo valido dal punto di vista tecnologico e appetibile sotto il profilo strettamente commerciale, è fondamentale fare in modo che tutte queste personalità collaborino tra loro in maniera coesa e coordinata. In questo senso, la fase di **pre-produzione**, prevede la predisposizione di un piano di lavoro ben definito a partire dalla fase di progettazione che può essere suddivisa in diversi "momenti". Ovviamente non c'è una ricetta standard, ma esistono vari percorsi e soluzioni: le fasi scritte qui di seguito sono un possibile percorso standard.
La prima fase progettuale per la creazione di videogame consiste nella definizione dell'idea, ovvero del *concept* su cui si realizzerà il videogioco: è un momento fondamentale in cui si mettono sul tavolo diverse opzioni narrative, diverse storie da "raccontare" potenzialmente appetibili per il pubblico che si vuole intercettare. In tal senso, è importantissimo in questa fase l'apporto del **copywriter** ed esperti di marketing e comunicazione in grado di leggere le "pulsioni" del mercato e individuare la storia giusta da raccontare.
La seconda fase di progettazione è quella dello **storyboard**, ovvero la creazione di una sequenza di disegni che mostrano i livelli di gioco e le diverse scene con gli obiettivi da raggiungere per il giocatore. Ogni "tavola" dello storyboard deve essere accompagnata da un paragrafo o due in cui si descrive in modo non troppo dettagliato ciò che avviene ad ogni livello del gioco.

Una seconda opzione aggiuntiva potrebbe essere quella del **moodboard**, cioè la creazione di tavole di immagini/concept che aiutino a comprendere meglio il mood del gioco ed assimilare l'idea originaria in maniera più efficacie possibile. In questo punto, **illustratori** e **designer**, si aggiungono alla presenza del copywriter per lavorare passo passo per dare una rappresentazione coerente e sensata dell'idea di partenza.
Arriva quindi il momento in cui si scende nei particolari della "storia": così, dopo aver creato lo storyboard, si iniziano a descrivere tutti i dettagli sia a livello di design che di narrazione (i due aspetti a questo punto sono strettamente interconnessi tra loro). Il processo creativo dunque si complica perché

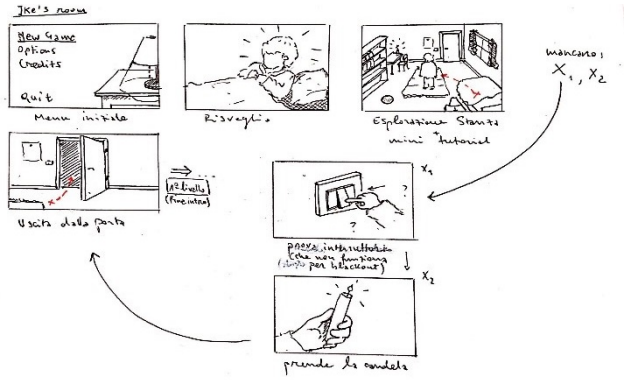


Figura 42: Esempio di storyboard per una sequenza iniziale dell'indie game *Ike - A low poly Nightmare*.

occorre pensare ad ogni possibile variabile ed opzione che possa intervenire nel gioco: deve emergere la qualità del prodotto finale.
La progettazione si conclude con un **documento finale** in cui si registrano e si elaborano tutte le idee prodotte in un formato che assomiglierà molto ad una sceneggiatura cinematografica. La creazione di questo script è un passo conclusivo fondamentale che offre tuttavia ai progettisti la possibilità di correggere alcune imperfezioni o talvolta anche di cambiare idea su alcune scelte prese in precedenza. In quest'ultimo caso si procederà ad una revisione complessiva del progetto.

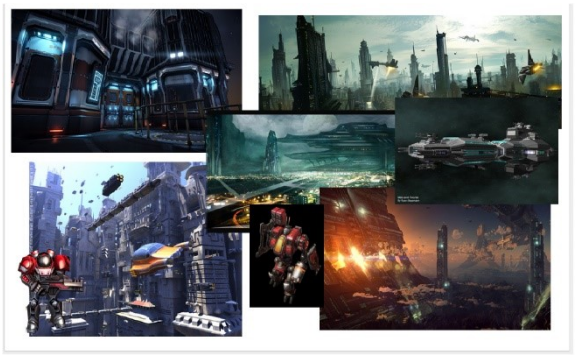


Figura 43: Esempio di moodboard a tema fantascientifico.

Sviluppare un videogame è costoso, mentre la stesura di un documento, seppur tecnico e molto preciso, è molto più economico: tale script è definito come **GDD**, o *Game Design Document*, fornisce il maggior numero di dettagli possibili riguardo le meccaniche di gioco, l'aspetto estetico, il gameplay *second-to-second*, *minute-to-minute* e molti altri dettagli. Ovviamente, sia moodboard che storyboard devono essere inseriti al suo interno. Tale livel-

lo di dettaglio nella progettazione aiuta il team ad identificare problemi ed incongruenze legate al concept, chiarire gli aspetti più confusi ("il combat system è come quello di The Witcher 3" non spiega esattamente come sarà all'interno del gioco in sviluppo: come saranno gli attacchi? E la difesa? Le armi sono distruttibili o no? Etc...), pianificare aspetti o sezioni a cui non si è dato peso in fase creativa, quindi essere sicuri che l'intero team sia a consapevole di ciò che va a realizzare. Una volta conclusasi la fase progettuale si passerà alla creazione del videogame vera e propria ed in questo caso entreranno in gioco figure più tecniche altamente specializzate come sviluppatori e programmatori sia per console, sia PC che per app mobile.

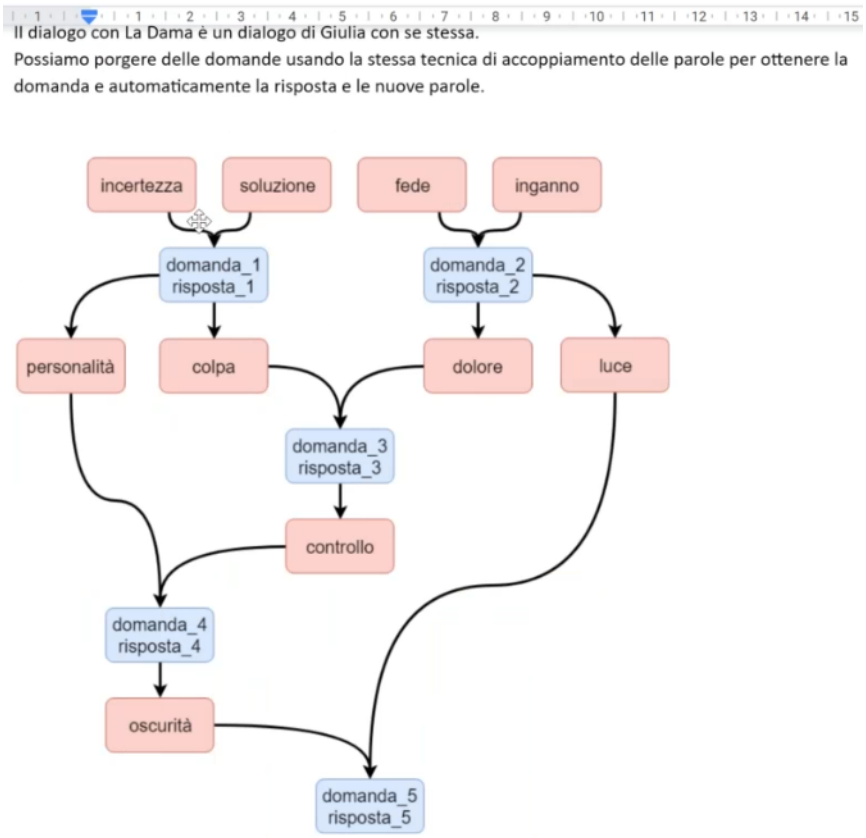


Figura 44: Esempio di documento di gameflow utilizzato durante lo sviluppo di *The Town of Light*. I documenti di game flow spiegano nel dettaglio cosa succede in ogni fase del gioco ed è consigliabile spezzare tale trattazione in più files; tali documenti devono essere aperti a commenti ed interventi da parte di tutto il team, dato che risulteranno fondamentali per il lavoro di tutti i componenti del team.

Figura 45: Cover di lancio di *The Town of Light*.

Esempio virtuoso I A: la pre-produzione di *The Town of Light* Uscito nel 2016 e sviluppato da **LKA** (studio fondato da **Luca Dalcò**), è un titolo cosciente di avere una mole immensa di informazioni sia per quanto riguarda il mercato nel quale sarebbe uscito, sia per il gioco in essere.
The Town Of Light tratta tematiche molto complesse legate alla malattia mentale e al mondo dei manicomi italiani di metà ventesimo secolo: forse, si può dirlo, è una scelta poco commerciale. Eppure è proprio quella la forza dirompente del titolo: riscrivere le regole del mercato, a volte può essere un'arma molto potente (la parte riguardante il mercato verrà approfondita in seguito). L'idea era quella di creare un videogioco che avesse una base culturale forte da poter legare ad un determinato territorio e veicolare ancora di più un messaggio ben specifico attraverso la storia che si vuole raccontare. Si ricordi però: lo sviluppo di un videogioco prettamente *narrativo*, non deve trasformare il gioco in un ludo-documentario, ma occorre mantenerne la parte giocosa. Per *The Town of Light*, si è partiti dalla visualizzazione di ambienti già esistenti per passare in seguito alla costruzione della storia al loro interno; per farlo è servita una mole immensa di ore per documentarsi sia sul tema della malattia mentale, che sulla storia di come questa malattia veniva trattata agli inizi del XX secolo: questo mostra che lavorare su *setting* reali o immaginari, presenta spesso la stessa mole di lavoro, seppur su fronti diversi. Anche la costruzione della protagonista è fondamentale: l'età, le problematiche e la sua caratterizzazione sono in funzione della drammatizzazione del prodotto.
La scelta della visuale in prima persona, serve per aumentare l'immersione del giocatore: la visione soggettiva garantisce la sensazione di vivere una vera e propria esperienza personale; inoltre è una scelta molto furba anche a livello produttivo: non c'è bisogno del modello del personaggio, di animazioni aggiuntive o di altro lavoro di pre-produzione. Esiste un rovescio della

medaglia, poiché questa scelta porta tutte le possibili imperfezioni in primo piano: la gestione delle textures deve essere molto precisa in modo da non rovinare l'immersività (uno step in questa direzione è stato il passaggio da *Unity* ad *Unreal Engine*). Un altro modo per ridurre i costi produttivi legati all'ambito 3D è l'utilizzo dell'illustrazione 2D per le sequenze narrative che spezzano le sezioni di gameplay tra i livelli: non si svilisce la storia e si risparmia moltissimo per quanto riguarda risorse e tempo produttivo.

Figura 46: Uno screen dal gameplay di *The Town of Light*: risulta evidente come la visuale in prima persona "costringa" il team ad un lavoro molto oneroso e puntuale sulle texture degli ambienti.

3.3.2 Sviluppo II: Produzione

Una volta che sono stati pianificati sufficienti elementi per permettere a programmatori ed artisti di mettersi all'opera, il processo di sviluppo vero e proprio può cominciare. In realtà questa fase comincia anche prima: se il concept del gioco si basa su un platform 2D a scorrimento laterale single player con meccaniche di scatto e salto, i programmatori possono dedicarvisi mentre il resto del progetto si arricchisce; stesso discorso vale per gli artwork e la realizzazione delle concept art.

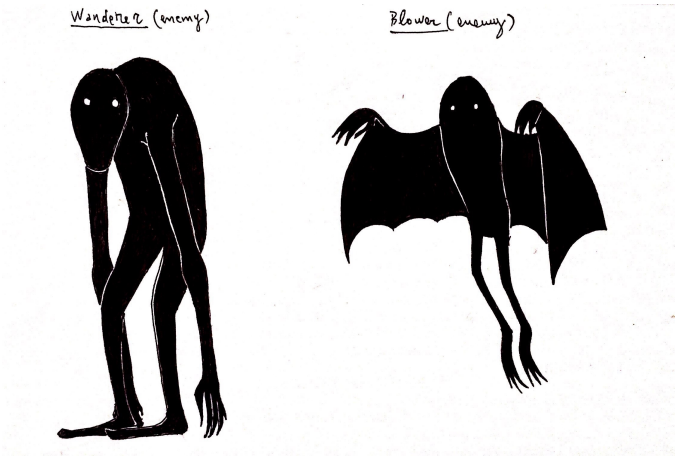


Figura 47: Concept art realizzate durante lo sviluppo dell'indie game *Ike - A low poly Nightmare*.

Se è prevista la pubblicazione anche su piattaforme diverse dal PC, solitamente è necessario richiedere un **devkit**, ovvero un kit di sviluppo dedicato. Si tratta di un dispositivo hardware sul quale gira lo stesso sistema operativo della console per cui si sta sviluppndo il gioco (es. Nintendo Switch, PS4, Xbox One), con funzioni aggiuntive di debug e testing.

Figura 48: Una foto del devkit utilizzato per lo sviluppo sulla nuova PS5.

Una volta realizzato un prototipo funzionante, avrò un titolo con un design di gioco formato ed arricchito rispetto al concept iniziale. Il passaggio successivo, solitamente, è quello di lavorare ad una **vertical slice**: una sezione giocabile del titolo che include la maggior parte delle meccaniche e un certo numero di contenuti per mostrare tutte le sue caratteristiche.

Figura 49: Uno screen della vertical slice di *Ike - A low poly Nightmare*; in questa sezione di gioco si possono testare le meccaniche principali di questo piccolo stealth/puzzle game.

L'utilizzo di una vertical slice è diffusissimo nel modello publisher/studio *work-for-hire*, dal momento che dimostra la fattibilità del gioco ai finanziatori. Può fungere anche da banco di prova per alcune funzionalità e si può rivelare un ottimo strumento di marketing. Tuttavia, si tratta di una pratica controversa, che non sempre premia nei confronti del pubblico.

Figura 50: Un altro screen dello sviluppo di *Ike - A low poly Nightmare*: è stato fatto un attento lavoro sugli shader da utilizzare in funzione dell'atmosfera "creepy" voluta dagli sviluppatori.

Una volta che le funzioni principali del gioco sono pronte, arriva il momento di creare il resto dei contenuti che si vogliono inserire nel gioco: costruire livelli, realizzare i restanti asset grafici, scrivere (o terminare di scrivere) la storia e fare tutto ciò che serve per trasformare una sezione di gameplay abbozzata in un'esperienza completa e fruibile dall'inizio alla fine.
La definizione di **alpha** non ha standard e varia da team di sviluppo a team di sviluppo, ma nel gergo comune, l'obiettivo è quello di avere una versione completa del gioco per quanto riguarda i contenuti: può esserci la presenza di placeholder al posto di asset grafici, funzioni da rifinire, una UI basilare e qualche errore di gioco ancora da sistemare, ma tutti i livelli giocabili devono esserci e tutto ciò che il GDD prevede deve essere presente.

Figura 51: La versione alpha di *Minecraft*.

3.3.3 Sviluppo III: Post-Produzione

Dopo la versione alpha, si ha un processo di rifinitura che conduce alla **beta** del gioco: i placeholder devono essere sostituiti dagli asset veri e propri, i crash e gli altri bug principali devono essere sistemati e tutti i problemi di bilanciamento più evidenti devono essere risolti, assieme alle incertezze del gameplay.
In questa fase, il controllo qualità entra in azione: il team del Q&A completa per intero il gioco più e più volte, provando a "romperlo" in maniera creativa, sottolineando quello che non è divertente e tutto quello che non funziona secondo i loro standard qualitativi; tutti i problemi rilevati sono registrati in un **bug tracker**, e vengono eliminati una volta risolti.

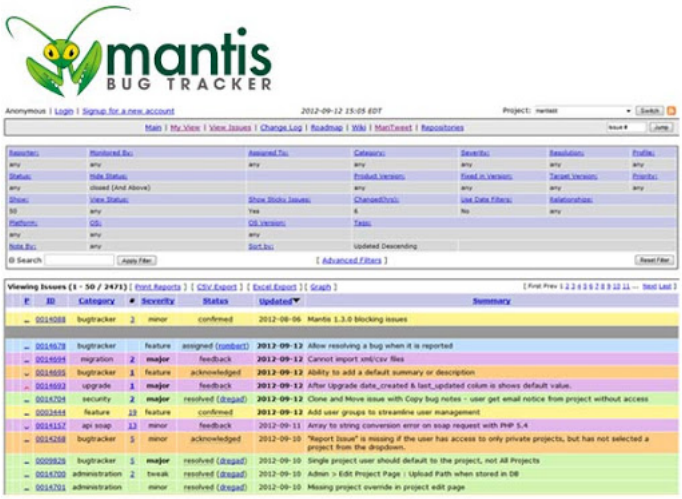


Figura 52: *Mantis Bug Tracker* è un popolare bug tracker open-source utile al controllo qualità: un sistema di tracciamento degli errori è un'applicazione che tiene conto dei bug software segnalati nei progetti di sviluppo software. Molti sistemi di tracciamento dei bug, come quelli utilizzati dalla maggior parte dei progetti software open source, consentono agli utenti finali di inserire direttamente le segnalazioni di bug. Altri sistemi vengono utilizzati solo internamente in un'azienda o organizzazione che si occupa di sviluppo software. In genere i sistemi di tracciamento dei bug sono integrati con altri software di gestione dei progetti. Un sistema di tracciamento dei bug è di solito un componente necessario di un'infrastruttura di sviluppo software professionale e l'uso coerente di un sistema di tracciamento di bug o problemi è considerato uno dei "tratti distintivi di un buon team di software".

Il **debugging** (o semplicemente debug), in informatica, indica l'attività del programmatore che consiste nell'individuare la parte di software affetta da errore (*bug*) attraverso l'utilizzo di un programma ausiliario (*debugger*). Si tratta di un'attività molto difficile data la complessità dei programmi odierni e la delicatezza dell'attività stessa, poiché il rischio di introdurre nuovi errori tentando di risolvere quelli già presenti è altissimo.
Ogni sessione di debug ha una storia a sé, ma esistono alcuni principi generici applicabili in ogni situazione. In particolare, si possono riconoscere cinque fasi:

1. identificazione del bug
2. individuazione del componente in cui è presente il bug

3. individuazione della causa del bug
4. progettazione di una correzione per il bug
5. implementazione e testing della suddetta correzione

L'errore può essere rilevato sia in fase di collaudo del programma, ovvero quando ancora ci si trova in fase di sviluppo, sia in fase di utilizzo da parte dell'utente finale. Alla rilevazione dell'errore segue la fase di debugging, ossia l'individuazione della parte software nella quale si annida l'errore. Questa attività è oggi supportata da programmi specifici (debugger), che mostrano allo sviluppatore l'esecuzione, istruzione per istruzione, del programma: in assenza di tali strumenti, si ricorre alla più semplice e meno efficace tecnica di loggare (stampare) a video o su file le istruzioni che il programma sta eseguendo, inserendo a tal scopo nel codice delle istruzioni di debug.

Promuovere il proprio videogioco al pubblico solitamente spetta al **publisher**, nel caso in cui se ne abbia uno. Le principali operazioni di marketing avvengono a ridosso del **lancio**, ma alcuni publisher sono noti per iniziare a muovere la macchina dell'hype per giochi che, in quel momento, non sono altro che qualche bel artwork ed una manciata di parole su carta. Altri, come Nintendo, evitano di annunciare i propri titoli finché lo sviluppo non è praticamente completato, con giochi che sono già in fase avanzata di sviluppo ed aiutare ad inviduare i bug.

Figura 53: *Death Stranding*, creato dal genio di **Hideo Kojima** e sviluppato da **Kojima Productions**, fu annunciato per la prima volta il 14 giugno 2016 (uscito poi circa 3 anni dopo) sul palco dell'E3 di Los Angeles; Kojima si presentò al pubblico accompagnato con la semplice (ma iconica) frase "I'm back" e da un breve teaser trailer del gioco: fu una delle standing ovation più indimanticabili della storia della manifestazione.

Una volta che tutti i bug vengono sistemati (o quasi, qualcuno può pur sempre sfuggire ed essere sistemato tramite patch successive), arriva la deadline: è il momento di pubblicare la build definitiva. Ovviamente prima della pubblicazione, i produttori di hardware devono verificare che il gioco rispetti gli standard da loro imposti: ogni console ha alcune linee guida standardizzate che includono aspetti come la terminologia hardware, la gestione degli errori, e come ci si interfaccia con i menù di sistema (es. procedura che mette correttamente in pausa il gioco per mostrare l'interfaccia di sistema dopo la pressione del tasto HOME).
Oltre a controllare questo aspetto, il controllo deve assicurarsi che il gioco funzioni dall'inizio alla fine, non sia soggetto a crash e che i contenuti non

violino le policy dei produttori; questo processo può durare settimane, poiché se dovessero esserci problemi, lo sviluppatore dovrebbe risolverli prima di effettuare una nuova **submission**.

3.3.4 Sviluppo IV: Post-lancio

Dopo che il gioco ha superato la submission, è pronto per essere pubblicato: se si può contare su un publisher di solito, questo si occuperà (solitamente) di tutti gli aspetti tecnici riguardanti il lancio, tipo stabilire la data di uscita con i titolari della piattaforma di destinazione, la distribuzione delle copie fisiche ai negozi (se previste) e via dicendo.

Figura 54: La Collector's Edition di *Hollow Knight*, uscito nel 2017 e sviluppato da **Team Cherry**, è un esempio di versione fisica esclusiva da acquistare: il publisher interverrà anche in questo campo e proporrà versioni standard, deluxe o limited, in modo da ampliare le possibilità di acquisto dell'utenza.

Il processo necessario per lo sviluppo di un videogioco è completo, quindi spetta allo sviluppatore e al publisher decidere se supportare il titolo dopo con *patch* e risoluzione di bug. Farlo rappresenta una grande opportunità per dimostrare buona volontà alla community e mantenere il proprio gioco in vista per un periodo più lungo di tempo, cosa che potrebbe tradursi in un maggior numero di copie vendute sia per il titolo in questione che per il successivo.
Il post-lancio è un momento critico e richiede ulteriori competenze che non sempre hanno a che vedere con quelle tecniche o creative. Quando poi si parla di tematiche delicate all'interno del gioco, l'operazione di lancio diventa ancora più complessa: rischio di essere poco commerciale.

Nel caso di *The Town of Light*,ad esempio, la stampa non condannò il progetto: nonostante l'atmosfera cupa ed esplicitamente cruda (nel titolo sono presenti abusi, torture psicologiche,...), venne premiato dalla stampa

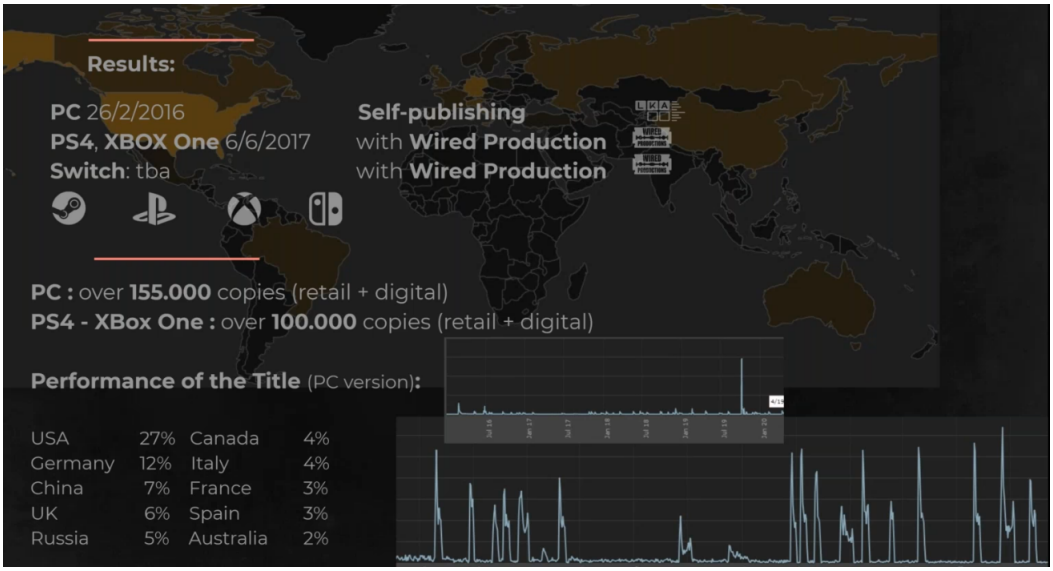


Figura 55: Nell'immagine sopra sono riportati i numeri emersi nel post lancio di *The Town of Light*.

nazionale ed internazionale, sia quella di settore che quella generalista. Fu un successo poiché si identificò il videogioco come vettore di dramma e quindi valore sociale, ma non si pensi a guadagni faraonici: il costo per la localizzazione, la pubblicità ed il supporto post lancio (fiere, festival) è altissimo (30/40 mila €), inoltre la media del prezzo di vendita rimane limitata (es: se ho un prezzo base di 10€, avrò una media vendita di 2/3 € effettivi). In questo caso, sono state fondamentali piattaforme come Steam o Humble Bundle. Una mossa intelligente a livello di marketing, quando il gioco è ambientato in spazi fisici realmente esistenti, sarebbe perfetto effettuare il lancio proprio nella location trasportata virtualmente nel gioco.

Figura 56: In occasione del lancio di *Dark Pictures: Man of Medan*, rilasciato da **Bandai Namco Entertainment** e sviluppato da **Supermassive Games**,ad influencer ed alla stampa di settore fu mandato un invito a partecipare all'evento lancio su una nave ancorata in un porto tedesco: qui, grazie all'uso di attori e costumi (oltre all'ambiente della nave), venne ricreata l'esperienza di gioco nel mondo reale, una mossa di marketing davvero azzeccata.

Esempio virtuoso II: Heartbit Interactive, Doom & Destiny ed il rapporto con la community. Per uno sviluppatore è importantissimo avere un rapporto ben strutturato con la **community**, tramite i vari punti di contatto disponibili ad oggi: fiere e festival sono ottimi strumenti per interfacciarsi con il pubblico, ma anche piattaforme social, come Facebook, che consentono un confronto diretto con il singolo giocatore. Così è stato per **Doom & Destiny**, RPG di stampo classico dalla forte scrittura narrativa, sviluppato da **Heartbit Interactive**, piccolo studio indipendente italiano; il forte legame creatosi con la community ha garantito al team una crescita costante della fan-base e la costruzione di un rapporto biunivoco con l'utenza.

Creare un sito web dedicato al proprio videogioco è importante, però al giorno d'oggi non è così privilegiante, anche perché molto costoso; le piattaforme social hanno quasi completamente sostituito il ruolo del sito web, poiché costano meno ed hanno generalmente maggior traffico ed engagement. Un'altra via è quella delle piattaforme di gaming che hanno aiutato alla pubblicazione di giochi indipendenti o poco famosi, come *IndieGala*, *IndieRoyale* o *HumbleBundle*.

Come scritto inizialmente, le fiere sono ottime occasioni per parlare con i propri giocatori, ma rischiano di essere dispersive: se si porta un gioco narrativo, difficilmente le persone possono giocarci in maniera approfondita

Figura 57: Uno screen della battle phase di *Doom & Destiny*, uscito il 25 agosto 2011 e sviluppato utilizzando **RPG-Maker**, è un gioco di ruolo con meccaniche classiche, caratterizzato da una narrativa brillante ed avvincente; conta un numero spropositato di quest secondarie, segreti e boss nascosti, elementi tipici del genere amatissimi dalla community. Proprio grazie al rapporto con quest'ultima, il team è riuscito a "patchare" ed aggiungere contenuti apprezzatissimi dai giocatori.

Humble Bundle

Figura 58: Logo di Humble Bundle

o rimanerne colpite. Ha più senso affiancare al titolo di lancio un gioco più "caciarone", magari multiplayer, magari semplicissimo, che faccia avvicinare la gente allo stand e da lì iniziare a tessere contatti, proporre il gioco principale, vendere qualcosa.
Da questo possibile primo contatto si può arrivare a costruire una community appunto sulle piattaforme social, come Facebook per l'appunto: si comunica

con la propria fanbase o si fanno post di contenuto (si parla del proprio prodotto) o di pubblicità, di prodotti di altri sviluppatori (*crosspromo*). Meglio non bombardare troppo l'utenza di Facebook con questo tipo di pubblicità: si è notato che i post legati al contenuto del gioco sono meglio accettati rispetto a quelli legati alla cross-promozione. Ma quanto tempo si deve dedicare alla community? E quanto è complicato bilanciare questa attività? Sicuramente avere uno spazio dedicato alla community aiuta nell'equilibrarne l'attività, come una fanpage (gruppo chiuso) dedicata al gioco: così si ha la possibilità di poter lavorare e gestire la community, senza perdere troppo tempo (quindi denaro) nel rispondere a messaggi singoli. Ovviamente sono tutte strategie utili nel caso in cui si abbiano contenuti da mostrare.
La gestione della community spesso ha bisogno di qualcuno che se ne occupi a tempo pieno: attenzione però che spesso molta gente che si propone come social media manager non è preparata. Un requisito base è quello di conoscere in maniera ottima la lingua inglese ed il prodotto di cui si sta gestendo la fan page; grazie ad un gestione sana, infatti, è possibile trovare molti lavori e collaborazioni all'interno della community: ad esempio, lavori di localizzazione fatti da fan del gioco per altri fan che magari non hanno il gioco tradotto nella loro lingua.

Un altro modo di interagire con la community è lavorare su un importante supporto post lancio: tale realtà è utilissima nei titoli emergenti, quando vengono fatti grossi update per completare un titolo magari uscito incompleto per necessità di pubblicazione.

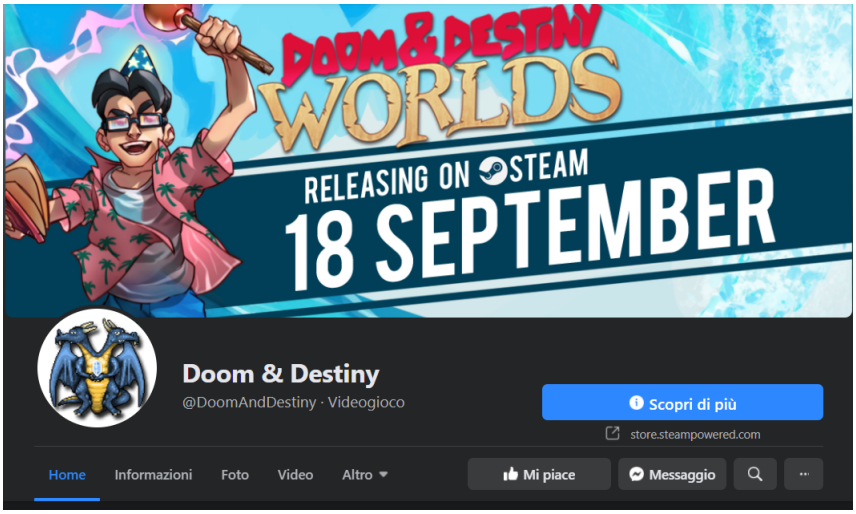


Figura 59: La pagina Facebook ufficiale di *Doom & Destiny*.

3.4 I 4 Pilastri

Nello sviluppo video-ludico occorre, come è già stato scritto, valutare attentamente quante e quali risorse impiegare nel progetto: nessun team o studio possiede risorse infinite, quindi analizzare la distribuzione delle risorse è un aspetto fondamentale. É importante distribuire la qualità nel gioco senza creare scompensi o problemi "ritmici" e proprio per tale motivo che si sono stabiliti quattro aspetti fondamentali per effettuare questa operazione: **narrativa**, **tecnologia**, **estetica** e **meccaniche**. Attorno a questi pilastri si dovrà costruire uno sviluppo omogeneo, cercando di dare maggior spazio ad un aspetto piuttosto che un altro, a seconda della tipologia di gioco che si va a sviluppare: se si vuole creare un gioco che punti di più sulla narrativa, ovviamente la percentuale di risorse impiegate per quel pilastro avrà un impatto rilevante sull'economia qualitativa del gioco in generale. Raramente si trovano giochi sviluppati con investimenti equipollenti in tutti e quattro i settori: è una questione di equilibrio. Nelle prossime sezioni verranno affrontate le particolarità di ogni settore e quale apporto possono singolarmente dare al progetto complessivo.

3.4.1 Il Ruolo della Narrativa

Da sempre nei forum video-ludici si è sempre discusso cosa conti di più in un videogioco: chi si allinea al pensiero "la grafica è la cosa più importante", chi si allinea a "no! il gameplay è l'unica cosa che conta", la guerra è tutt'ora aperta. Negli ultimi anni si è fatta sempre più sentire al voce di chi sosteneva "la cosa più importante è la storia": la narrazione negli ultimi dieci anni è diventata sempre più centrale. Numerosi titoli hanno fatto la loro fortuna grazie alla loro capacità di raccontare una storia, di trasportare il giocatore all'interno di un mondo immaginario, con una costruzione coerente e convincente; ogni gioco ha uno stile singolare di raccontarsi e narrare la propria storia: la narrativa silenziosa della serie *Dark Souls*, il mondo fantasy proposto dallo studio **CD Project Red** nel franchise di *The Witcher*, il game design *sottrattivo* di Ueda usato come veicolo narrativo nei suoi particolarissimi giochi o la fantapolitica della *Metal Gear Saga*.

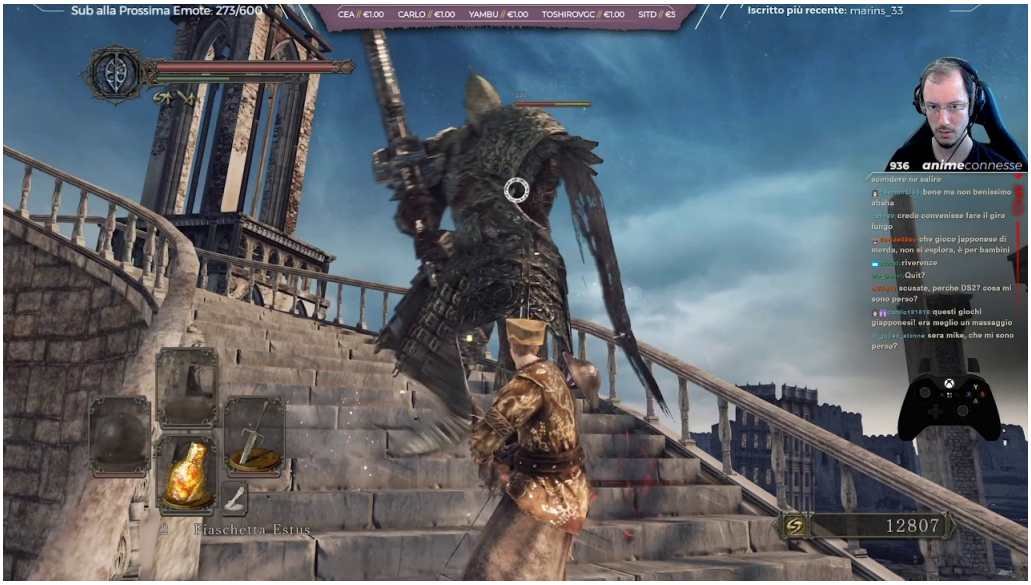


Figura 60: Per alcuni titoli la narrativa è talmente centrale da creare veri e propri canali di approfondimento su YouTube o Twitch: nel caso della serie *Dark Souls*, la narrazione criptica e fatta esclusivamente dal world building ha fatto sì che alcuni giocatori abbiano deciso di aprire canali su YouTube per raccontare e spiegare la narrativa tramite video di gameplay, come ad esempio VaatiVidya o **Sabaku No Maiku** (in figura durante una live Twitch su *Dark Souls II*), per quanto riguarda la scena italiana, arrivando a creare importantissime community di riferimento per gli appassionati del gioco.

La narrativa quindi è diventato strumento video-ludico a tutti gli effetti, andando a permearsi con tutti gli altri aspetti del gioco: il gameplay diventa coerente con la narrazione, la grafica e l'impatto visivo devono avere un valore narrativo e la tecnologia deve supportare tutto questo.

Esiste un metodo di costruzione dei giochi narrativi definibile come **non formalista**: "da dove si parte a creare un gioco?"; bisogna considerare molteplici prospettive e visioni sul progetto, oltre a tenere conto del BIAS di chi racconta il metodo di sviluppo. Si supponga di partire dalla storia e l'implementazione delle meccaniche avverrà *along the way*: l'inizio sarà cercare di capire quale **processo emozionale** si può suscitare nel giocatore. Con questo non si intende che le meccaniche non verranno curate, ma saranno adattate alla storia avendo una funzione narrativa; non servono per forza meccaniche originali, piuttosto serve avere un **game feel** ed un contesto efficaci: la storia deve precedere le meccaniche. Prima di continuare,

si definisca un concetto utile alla rapporto gameplay-narrativa, ovvero la **depth**: a livello generale, si intende lo spazio di combinazioni appartenente alla meccaniche di base, prestando attenzione al *balancing* dal punto di vista del giocatore. Una seconda definizione, è legata alla narrativa: un gioco avrà una depth più alta, se la ho un'alta densità di narrativa di background; l'esempio più calzante di tale nozione è la modalità di scrittura di Philip K. Dick, autore di *Ma gli Androidi Sognano Pecore Elettriche?*, il romanzo da cui hanno tratto *Blade Runner*. Fondamentale per arrivare a questo risultato è il mantenimento della coerenza e della **connettività diegetica**: tutte le dimensioni nella costruzione del gioco devono essere consistenti e coerenti con la narrativa di fondo e tra loro stesse (es: l'indossare o meno una camicia in *Disco Elysium*, sviluppato da **ZA/UM** nel 2019, ha una stretta funzione narrativa e caratterizzante per la storia).

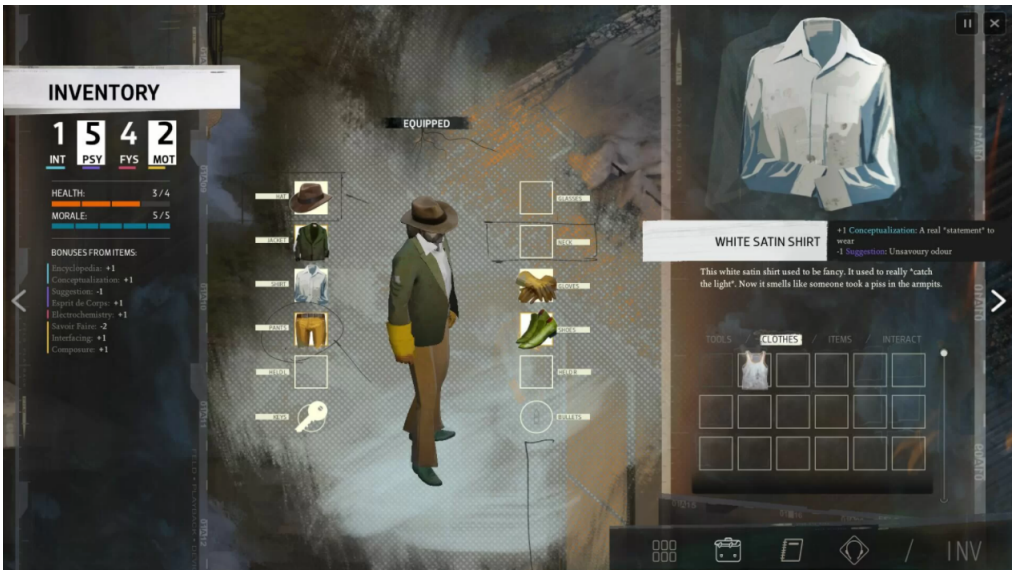
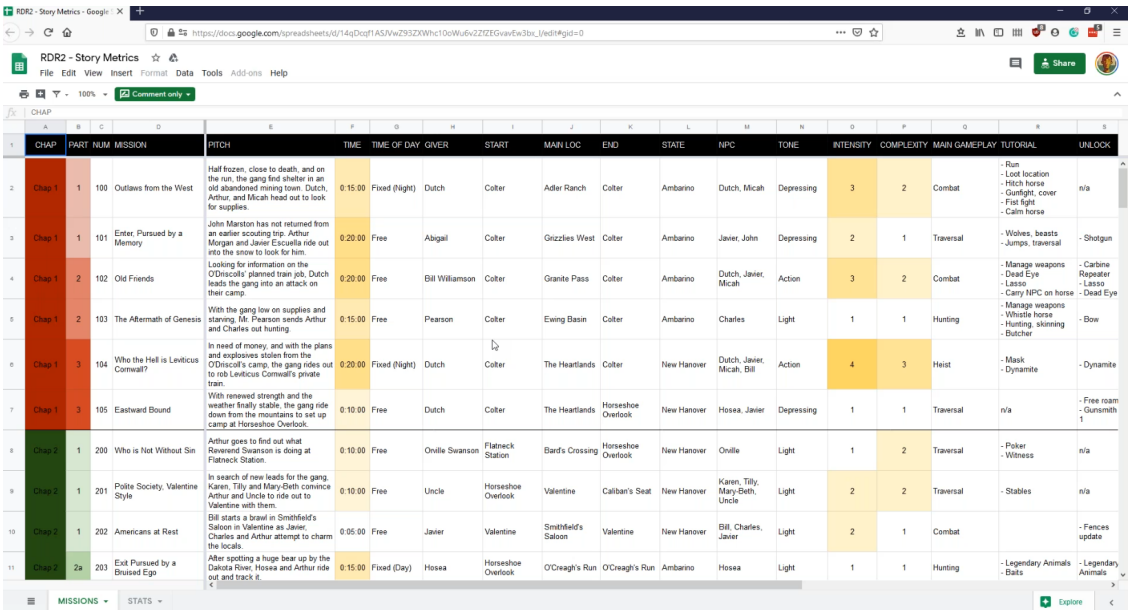


Figura 61: In *Disco Elysium* ogni scelta, anche dalla più banale, ha una forte influenza narrativa, integrata perfettamente con il gameplay.

Documentazione & Narrativa Lo sforzo narrativo autoriale dovrebbe essere diretto in modo che il gioco che si vuole sviluppare, sia il più lontano possibile del classico *walking simulator* (giochi in cui il gameplay è praticamente assente); servono trucchi ed escamotage che facciano sentire meno imbottigliato il giocatore:

- Spazi esplorabili **aperti**
- **Quest** assegnate che non ricorrono all'uso di un quest giver
- **Minimizzare** la quantità di documenti collezionabili utili al world building
- **Variazione del gameplay** durante la progressione del gioco

Più queste soluzioni sono semplici e meglio è: il team non è caricato troppo di lavoro da fare ed andando in questa direzione si sceglie di utilizzare un gran numero di documenti diversi tra loro; questo è utile per riuscire a far assimilare al team quello che va fatto. Ovviamente una volta terminata la documentazione non è possibile iniziare il gioco.
In ogni caso è utile avere sempre la gestione dei task e dei processi creativi a disposizione, anche se dopo una certa fase il **GDD** viene abbandonato e le modifiche si applicano direttamente sul gioco, facendolo diventare il documento stesso.



CHAP	PART	NUM	MISSION	PITCH	TIME	TIME OF DAY	GIVER	START	MAIN LOC	END	STATE	NPC	TONE	INTENSITY	COMPLEXITY	MAIN GAMEPLAY	TUTORIAL	UNLOCK
Chap 1	1	100	Outlaws from the West	Half frozen, close to death, and on the run, the gang find shelter in an old abandoned mining town. Dutch, Arthur, and Micah head out to look for supplies.	0:15:00	Fixed (Night)	Dutch	Colter	Adler Ranch	Colter	Ambarino	Dutch, Micah	Depressing	3	2	Combat	- Run - Loot location - Hitch horse - Gunfight, cover - Fist fight - Calm horse	n/a
Chap 1	1	101	Enter, Pursued by a Memory	John Marston has not returned from an earlier scouting trip. Arthur Morgan and Javier Escuella ride out into the snow to look for him.	0:20:00	Free	Abigail	Colter	Grizzlies West	Colter	Ambarino	Javier, John	Depressing	2	1	Traversal	- Wolves, beasts - Jumps, traversal	- Shotgun
Chap 1	2	102	Old Friends	Looking for information on the O'Driscolls' planned train job, Dutch leads the gang into an attack on their camp.	0:20:00	Free	Bill Williamson	Colter	Granite Pass	Colter	Ambarino	Dutch, Javier, Micah	Action	3	2	Combat	- Manage weapons - Dead Eye - Lasso - Carry NPC on horse	- Carbine Repeater - Lasso - Dead Eye
Chap 1	2	103	The Aftermath of Genesis	With the gang low on supplies and starving, Mr. Pearson sends Arthur and Charles out hunting.	0:15:00	Free	Pearson	Colter	Ewing Basin	Colter	Ambarino	Charles	Light	1	1	Hunting	- Manage weapons - Whistle horse - Hunting, skinning - Butcher	- Bow
Chap 1	3	104	Who the Hell is Leviticus Cornwall?	In need of money, and with the plans and explosives stolen from the O'Driscoll's camp, the gang rides out to rob Leviticus Cornwall's private train.	0:20:00	Fixed (Night)	Dutch	Colter	The Heartlands	Colter	New Hanover	Dutch, Javier, Micah, Bill	Action	4	3	Heist	- Mask - Dynamite	- Dynamite
Chap 1	3	105	Eastward Bound	With renewed strength and the weather finally stable, the gang ride down from the mountains to set up camp at Horseshoe Overlook.	0:10:00	Free	Dutch	Colter	The Heartlands	Horseshoe Overlook	New Hanover	Hosea, Javier	Depressing	1	1	Traversal	n/a	- Free roam - Gunsmith 1
Chap 2	1	200	Who is Not Without Sin	Arthur goes to find out what Reverend Swanson is doing at Flatneck Station.	0:10:00	Free	Onille Swanson	Flatneck Station	Bard's Crossing	Horseshoe Overlook	New Hanover	Onille	Light	1	2	Traversal	- Poker - Witness	n/a
Chap 2	1	201	Polite Society, Valentine Style	In search of new leads for the gang, Karen, Tilly and Mary-Beth convince Arthur and Uncle to ride out to Valentine with them.	0:10:00	Free	Uncle	Horseshoe Overlook	Valentine	Caliban's Seat	New Hanover	Karen, Tilly, Mary-Beth, Uncle	Light	2	2	Traversal	- Stables	n/a
Chap 2	1	202	Americans at Rest	Bill starts a brawl in Smithfield's Saloon in Valentine as Javier, Charles and Arthur attempt to charm the locals.	0:05:00	Free	Javier	Valentine	Smithfield's Saloon	Valentine	New Hanover	Bill, Charles, Javier	Light	2	1	Combat		- Fences update
Chap 2	2a	203	Exit Pursued by a Bruised Ego	After spotting a huge bear up by the Dakota River, Hosea and Arthur ride out and track it.	0:15:00	Fixed (Day)	Hosea	Horseshoe Overlook	O'Creagh's Run	O'Creagh's Run	Ambarino	Hosea	Light	1	1	Hunting	- Legendary Animals - Baits	- Legendary Animals

Figura 62: Quando si lavora ad un videogioco narrativo serve avere chiara la struttura e lo svolgimento del titolo, dalle prime fasi al finale: in figura è presente lo*story metrics* di *Red Dead Redemption 2* di **Rockstar Games** che presenta le fasi di gioco in base allo svolgimento della storia.

Quali sono i tipi di documenti spendibili per lo sviluppo di un gioco? Il *documento di sintesi* deve essere breve e conciso nello spiegare in cosa consiste il gioco (sintesi storia, sintesi gameplay,...): esso può essere ad uso del team o ad uso esterno, utile per far conoscere il gioco a publisher, audience e stampa. Il *documento di Q&A* è basato totalmente sulla meccanica di domanda e risposta, utilissimo per il mantenimento della coerenza: "dove si voleva arrivare?", "Qual era il fine della storia?", "Come si supponeva che il giocatore si comportasse in questo punto del livello?", sono domande che aiutano la comunicazione intra-team a non perdere la rotta durante lo sviluppo.
Viene fatta anche una lista di cose che è meglio non dire o fare, cioè tutte quelle idee o implementazioni che non funzionano o potrebbero danneggiare il progetto.

Il *documento delle idee* serve a raccogliere tutti gli spunti raccolti durante le sessioni di *brainstorming*: possono essere "buone" idee, possono essere "cattive" idee, ma servono tutte per avere un "serbatoio" a cui attingere, oltre a tenere traccia di quelle già implementate (o quelle da implementare), che non vanno cancellate; sul documento infatti è possibile trovare sia le idee implementate, sia quelle che ancora devono essere realizzate: per mantenere un ordine ed una logica, vengono distinte con sottolineature o colori diffe-

renti, a volte addirittura con font diversi. Altro documento fondamentale è quello di *gameflow* che spiega nel dettaglio cosa accade in ogni fase del gioco; tali documenti devono essere aperti a commenti ed interventi da parte del team, quindi utili alla segmentazione del gioco: durante la prototipazione, infatti, si scrivono dialoghi di massima con più opzioni, mappe, disegni, angoli di vedute, possibilità e altre soluzioni che servono a compilare il documento.
Alcuni team dotati di abili *writers*, utilizzano la scrittura creativa per scrivere il gioco come fosse un racconto: questa modalità facilita e rende più godibile una documentazione che altrimenti risulterebbe più asettica, mentre è sempre buona pratica far trasparire il mood del progetto in ogni sua componente di produzione.

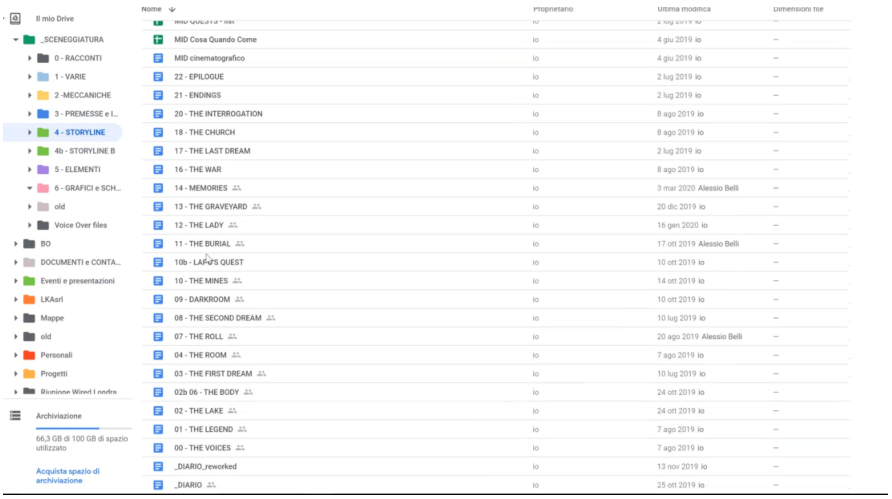


Figura 63: Il documento di gameflow deve essere ordinato in maniera logica e frammentato in modo da poter ricostruire il gioco nella sua interezza: la suite di **Google** offre numerose soluzioni per la costruzione di questi documenti.

Se il progetto presenta una storia molto intricata può risultare utile avere un *documento delle spiegazioni*: infatti alcune forme narrative diventano complicate e si snodano lentamente, evitando di rivelare tutti i propri punti cardine al giocatore. Come i libri gialli o mistery, dove trame ed azioni dei personaggi non hanno un senso logico sempre chiaro agli occhi dello spettatore: ecco questo documento serve ad organizzare la narrazione all'interno della progressione di gioco, segnando **quando**, **dove** e **come** vanno o non vanno rivelati determinati elementi della trama. Ogni genere ha una sua progressione narrativa, quindi anche il documento avrà una customizzazione ad hoc. Un documento più tecnico è quello delle *meccaniche*: solitamente isolato poiché tratta delle meccaniche generali, senza un preciso punto interno alla trama; quelle che sono pensate per un preciso momento narrativo,

vengono solitamente inserite nel documento di gameflow.
I documenti prettamente tecnici servono alla raccolta artwork, reference fotografiche o grafiche o testuali, utilissimi nella gestione dell'art direction. Spesso nella gestione di narrative *non lineari*, soprattutto nella distribuzione dei dialoghi, dei momenti narrativi e delle meccaniche, occorrono dei *documenti grafici*: servono ad avere la visione d'insieme, quindi analizzare il gioco nella sua totalità; ad esempio si sfruttano dei **flowchart** per la storia o per l'uso di meccaniche, gestendo in maniera efficace il ritmo della narrazione o la scoperta di errori, di criticità e varie problematiche.

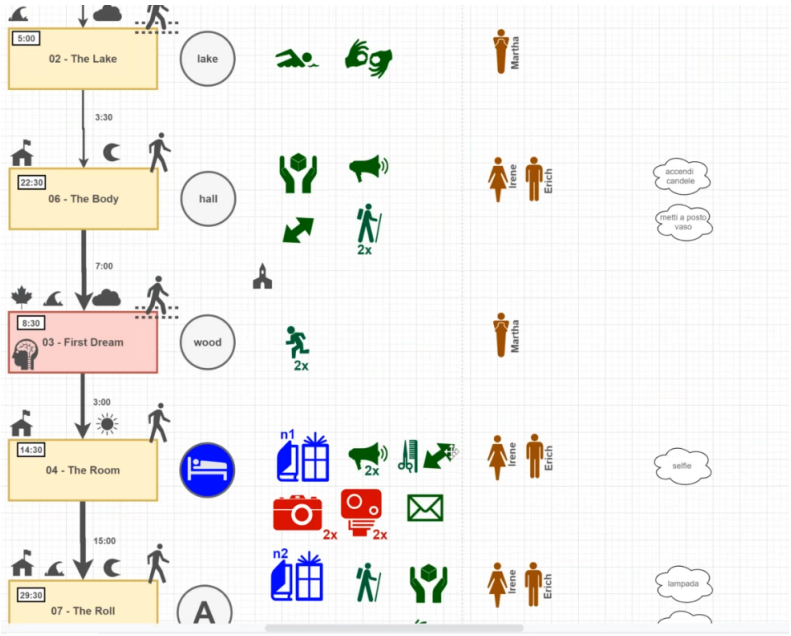


Figura 64: Utilissimi per la distribuzione delle meccaniche, i *documenti grafici* consentono di visualizzare in maniera ordinata sia le parti di gameplay sia lo sviluppo narrativo; questo metodo consente si evidenziare squilibri di qualità, errori di vario genere o scelte sbagliate nel rapporto tra meccaniche e momento narrativo.

Utilizzare differenti **rami narrativi** o **finali multipli** aumenta di molto la complessità ed il costo di un progetto; se una molteplicità di finali aumenta la rigiocabilità e la mole di contenuti di un titolo da un lato, svilisce il vigore narrativo dall'altro: secondo alcuni, la potenza dei vari finali viene meno, rispetto ad un finale predeterminato...in realtà dipende tutto dal tipo di prodotto. Una buona soluzione è utilizzare un finale pre-determinato e premiare i giocatori con sezioni di gioco facoltative: magari un giocatore completa il gioco, ma solamente al 40% e ha lasciato indietro collezionabili, aree inesplorate e altri contenuti di lore.

Trigger Narrativi Il prodotto può presentare essenzialmente due tipologie di narrazione: la narrazione a **filo di perle** (*The Last Of Us*, *Metal Gear Solid*, *Town Of Light*, ...) oppure quella definita **story machine** (*Minecraft*, *The Sims*, ...).

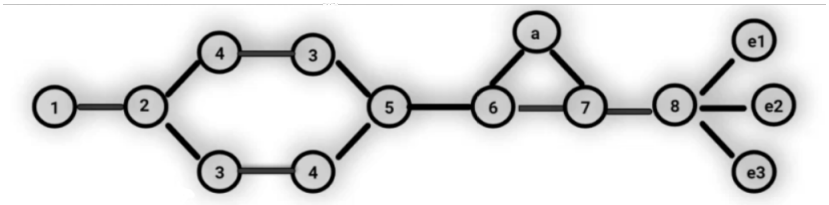


Figura 65: Schema della narrativa lineare a *filo di perle*: ogni perla è un nodo narrativo al quale si "entra" ed "esce" in maniera prestabilita.

Quest'ultima inserisce costantemente biforcazioni che mutano costantemente la storia, cosa che in termini letterali è impossibile: è il giocatore che crea la storia con le sue scelte, ma anche queste possono essere limitate e scriptate (*Detroit Become Human*(**Quantic Dreams**, 2018), ad esempio, ha tanti filoni narrativi, ma la struttura del gioco rimane essenzialmente quella). Non significa però che la struttura lineare conduca il giocatore in modo passivo nel progredire nella storia: anche in casi estremi come *Dear Esther* (*walking simulator* sviluppato da **The Chinese Room** nel 2012) il giocatore deve fare qualcosa di attivo, un qualcosa che fa avanzare la storia, portandoci da un momento narrativo ad un altro.

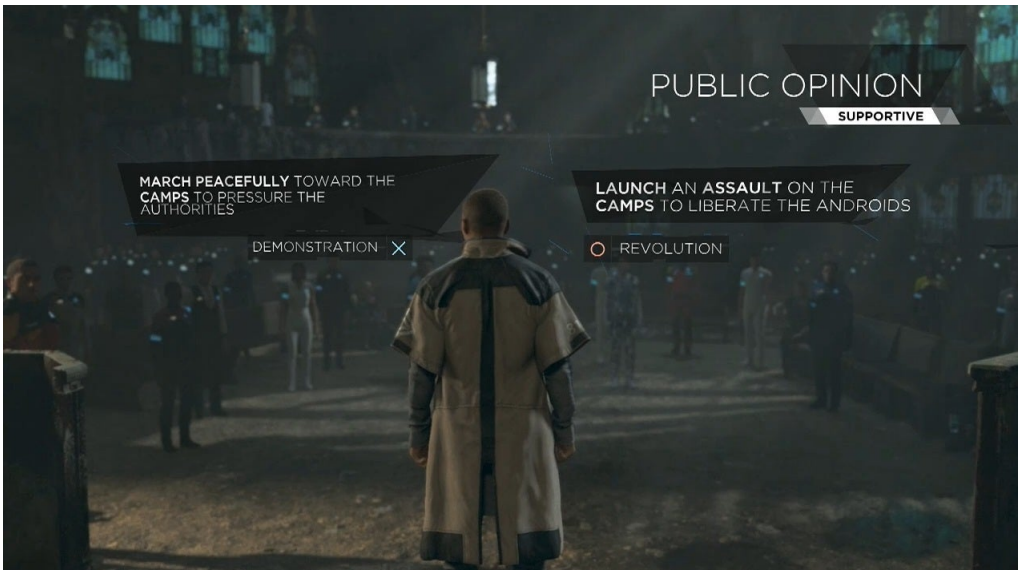


Figura 66: Un esempio di snodo narrativo in una scena di *Detroit Become Human*: la scelta del giocatore sblocca determinati rami narrativi che proporranno un'esperienza ludica di un certo tipo a seconda di quella scelta.

Questi espedienti narrativi sono identificati come eventi e sono composti da trigger, azione e conseguenza. Il **trigger** si può definire come la "motivazione a fare qualcosa"; la si definisce come la ragione per cui il giocatore dovrebbe fare qualcosa all'interno del gioco: il design è mirato a far capire cosa il giocatore è chiamato a fare o non fare, per esempio sopravvivere, combattere oppure si impone una scelta semi-obbligata (es: il giocatore non ha nient'altro da fare). L'**azione** è l'interazione del giocatore con ambiente ed oggetti inseriti nel gioco e la conseguenza è quello che serve a passare da una *perla all'altra*.
Ma come si relazionano queste tre componenti? Vi sono due "direzioni" di percorrenza generate dal contrasto tra **game flow** e **project flow**.

- TRIGGER -> ACTION -> CONSEQUENCE = GAME FLOW
- TRIGGER <- ACTION <- CONSEQUENCE = PROJECT FLOW

A livello progettuale si parte a lavorare sulle **conseguenze** e il trigger sia l'ultima cosa che va creata, anche perché risultano davvero complessi da inserire in un gioco, soprattutto se narrativo; i trigger sono la cosa più difficile da implementare poiché si rischia di banalizzare il gioco e/o ridicolizzare la narrativa. In un gioco prettamente narrativo, la sfida solitamente è bassa e trovare motivazioni per continuare a giocare è fondamentale. Il rischio di cadere nella dinamica da *walking simulator* è altissimo: è lì che bisogna lavorare, provando a rendere naturale il meccanismo di progressione; i giocatori, vogliono sentirsi parte della storia, rimanendo nei limiti del *flow state* che deve essere mantenuto da scelte oculate per quanto riguarda i trigger ed i **ban trigger**.

Figura 67: Il concetto di trigger si può anche applicare a meccaniche non strettamente narrative: in *Cuphead*, uscito nel 2017 e prodotto dal canadese **Studio MDHR**, essendo un *run & gun* si alternano livelli a scorrimento e battaglie con i boss; proprio nelle boss battle il decremento dei punti vita del nemico determina il susseguirsi di *fasi* in cui il boss può cambiare il suo *move set*, il suo aspetto ed i suoi attacchi: questo è un chiaro esempio di trigger non narrativo.

Si può vederli come "conseguenze inverse" dell'azione stessa: se il giocatore vede un raggio di luce che filtra da una finestra e si posa su un pulsante, avrà la pulsione a premerlo, senza sapere perché deve premerlo; vanno trovate modalità interessanti ed originali per creare questi meccanismi. Inoltre si devono eliminare tutte le azioni **indesiderate** nel mondo di gioco: alcune cose non possono essere fatte all'interno del mondo di gioco ed il vietare queste azioni sta al game design. Occorre creare il modo di eliminare una possibile azione da parte del giocatore senza però risultare incoerente o invadente: non usare *muri invisibili* o soluzioni che spezzano il realismo interno del gioco. Un esempio: se il personaggio è in grado di saltare due metri in verticale, usare una staccionata alta un metro come **ban trigger** di esplorazione spezzerà l'immersività e rovinerà l'esperienza di gioco. Il metodo di progettazione dei ban trigger sfrutta lo stesso ordine imposto dal game flow.

BAN TRIGGER -> ACTION -> CONSEQUENCE = GAME FLOW

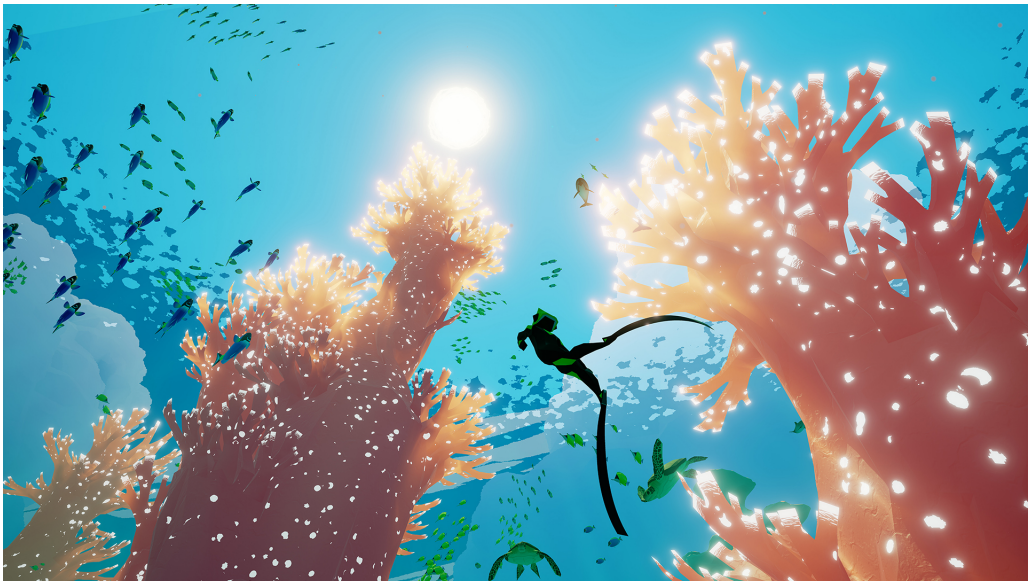


Figura 68: L'ottima avventura sottomarina *Abzù*, sviluppato da **Giant Squid Studios** nel 2016, presenta diverse superfici trasparenti ed invisibili che bloccano il passaggio del giocatore in quel punto della mappa, senza giustificazione; questo tipo di ban trigger è poco giustificata e non funzionale anche per il mercato indie.

Esempio Virtuoso I B: la narrativa di The Town of Light La narrazione usata in *Town Of Light* è composta da molteplici layer: se ci si pensa bene, molti giochi presentano narrative che hanno più punti di vista e più punti di lettura; in questo caso, c'è il punto di vista della protagonista, affetta da problemi psichici, e la narrazione attraverso la lettura delle cartelle cliniche posizionate in game. In pratica si utilizza lo scontro tra due realtà, una oggettiva ed una fittizia, irrazionale, generata dalle psicosi e problemi mentali. Oltre a questo, il nodo narrativo è centrato sulle sperimentazioni usate nella cura di queste malattie: manipolazione delle personalità, lobotomia, elettroshock, sono pratiche utilizzate nell'Italia del 1938 che non riconosceva la psicoanalisi come cura effettiva; ecco questo punto è fondamentale perché fa sì che ci sia coerenza a livello narrativo all'interno del gioco.
Ciò che nasce dal sopracitato scontro di realtà è una spinta per il giocatore: una spinta progettata appositamente per immergerlo in una situazione di costante ricerca della verità; una verità che però non esiste, se non nel fatto che la realtà è terribile ed orrifica; il tutto va reso **credibile**: non si pensi che il giocatore creda ad una storia se il set non è coerente o verosimile. Fondamentale per il progetto, sono stati lo studio degli ambienti, la composizione degli elementi in scena e la **regia**: lo sviluppatore deve lavorare in maniera

precisa perché poi deve fare i conti con la futura libertà del giocatore. La differenza con altri media è tutta qui: quando si gioca si è liberi, solitamente, di muoversi, sostare, guardare, analizzare l'ambiente di gioco; bisogna poter prevedere e progettare come questa libertà indotta possa essere sfruttata. Sempre per una questione di immersione, è fondamentale la **localizzazione**: serve avere una forma ispirata di scrittura e l'uso conscio della forza della lingua per mantenere coerente lo svolgimento narrativo e far leva sulla forza della poetica. Diventano di rilevante importanza le indicazioni per l'attore o l'attrice che deve fare il *voice over*: un suggerimento è quello di fornire indicazioni di massima, per non complicare o rendere meccanica la recitazione.

Videogames, Fumetto e Narrativa: costruire un personaggio Esistono molteplici affinità tra la produzione di una sceneggiatura per fumetti e la produzione di una narrativa per i videogiochi e tra queste vi sono sicuramente anche i problemi che affliggono entrambi i media; tra i tanti, il pericolo maggiore è sempre quello di incappare nello sviluppo di una miriade di contenuti e *sottotrame* che spesso portano a complicare il progetto. Dare l'impressione di non avere un'idea chiara del progetto è fatale: nessun publisher o editore lo prenderà in considerazione. Quando si presenta un soggetto narrativo bisogna conoscere perfettamente il plot, il mondo che si è costruito, i personaggi principali e quelli secondari. Questi vanno rispettati: i personaggi che siano già esistenti (es: Marvel) o che siano inventati e scritti per il soggetto in essere. Bisogna saper dimostrare di conoscere alla perfezione il personaggio e saperlo muovere coerentemente con la sua **codifica narrativa**; in caso il personaggio, sia già esistente, come Batman o Spider-Man, bisogna fare in modo che la storia si adatti al protagonista, ai comprimari o al mondo costruito attorno ad essi.

Figura 69: Nel recente *Marvel's Spider-Man*, sviluppato da **Insomniac Games** nel 2018, la figura di Peter Parker aka Spider-Man è ripresa nella sua totalità: la storia del personaggio e della trama principale non viene stravolta, ma rimane coerente con quanto costruito nei fumetti, introducendo sì nuove situazioni, ma senza snaturare i propri punti cardine.

Ciò che distrugge una sceneggiatura o una storia è l'**incoerenza**, quindi anche se il mondo in cui è ambientata è fittizio o finto ha delle regole imposte dall'autore e vanno rispettate. Come per il gaming, scrivere una sceneggiatura è una questione prettamente tecnica, differente dallo scrivere una storia (anche se a volte ho sovrapposizione), ma dall'altro lato anche artistica: la mente va allenata guardando serie, leggendo libri o giocando, appunto. Si devono analizzare i meccanismi alla base del prodotto rimanendo critici: questo lavoro di accumulo di informazioni, va a creare una sfera di suggestioni che può essere concretizzata in un prodotto. O anche no, perché alcune idee quando scritte perdono di valore e risultano funzionali: parevano più belle come ipotesi, eppure non si concretizzano in maniera rispettosa. Che fare? Vanno cercate quelle criticità e parti che non funzionano: a volte si riesce a correggere qualcosa, a volte no. Il lavoro verrà criticato o smontato: è un processo che serve assolutamente perché aiuta a far funzionare meglio la storia.
Molte storie, archetipi, personaggi, sono già stati scritti e attraverso la dissezione di una sceneggiatura è possibile raggiungere un certo livello di originalità e giocare coerentemente con la storia stessa.

Figura 70: Considerato probabilmente il miglior videogioco di sempre, *The Legend Of Zelda: Ocarina of Time*, sviluppato da **Nintendo EAD** nel 1998, rappresenta una delle iterazioni più importanti del franchise: da sempre la serie *The Legend Of Zelda* incarna perfettamente l'archetipo narrativo del viaggio dell'eroe, proponendo di volta in volta variazioni e modifiche sul tema.

Un personaggio va presentato in *azione*: cosa fa, cosa dice, come lo dice, cosa veste e come si rapporta al mondo circostante sono tutti elementi che aiutano il **character building** ed introducono il personaggio in maniera naturalmente fruibile. Oppure è possibile farlo descrivere in maniera contrastante da altri personaggi: "ha fatto questo", "non è vero! ha fatto quest'altro", etc.
Nel caso dei fumetti le informazioni girano su due canali diversi, quello visivo e quello scritto, mentre in quello dei videogames, si hanno ancora più canali d'informazione: saperli sfruttare al meglio, uno su tutti il canale del gameplay, innalza notevolmente la qualità del prodotto.
Stesso discorso vale per la storia: per avanzare, le storie, necessitano di **conflitti** e traguardi da superare, altrimenti non hanno uno svolgimento. La stesura narrativa deve prestare particolare attenzione alla distribuzione della qualità: tutto va costruito in funzione del finale, creando un'aspettativa (che ovviamente va rispettata col finale) nel giocatore, invogliandolo a continuare la propria avventura. Fondamentale per il finale, ma in generale per tutta la storia, è la costruzione dei personaggi che devono essere **realistici**; si devono creare le schede delle caratteristiche dei personaggi, dando loro tridimensionalità attraverso un nome, un abbigliamento, una storia, una psicologia, relazioni, etc.
Questi devono parlare come se stessi e non come parlerebbe l'autore, devono avere una loro voce, credibile anche nella crescita durante lo svolgimento

della storia: i personaggi meglio riusciti sono quelli che evolvono e crescono in maniera coerente con la natura di base del personaggio stesso; ai protagonisti delle storie possono succedere cose che influenzano la loro psicologia e comportamento in maniera positiva o negativa. Non si possono avere cambiamenti repentini e totali. Gli autori devono capire i personaggi che scrivono al punto da poterli prevedere: "come reagiranno?", "Cosa faranno in quella situazione?", sono domande utilissime al character building. Nella narrativa moderna non esistono più personaggi totalmente buoni o totalmente cattivi, ma piuttosto personaggi ambivalenti con zone d'ombra e zone luminose.

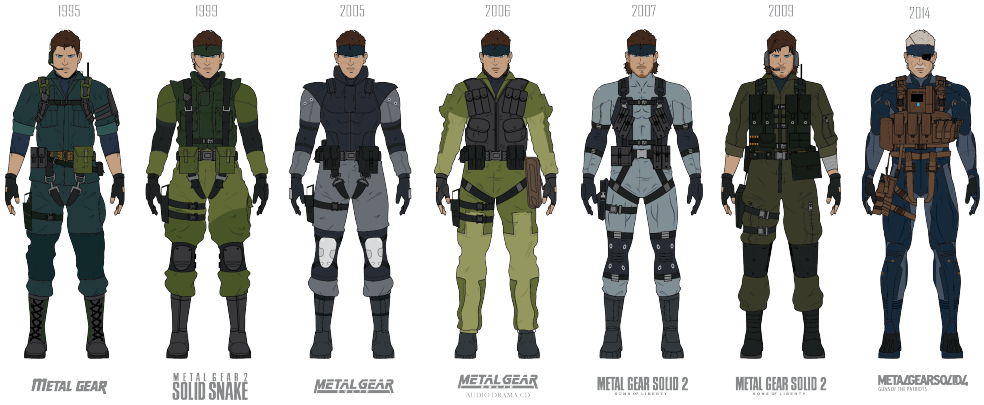


Figura 71: L'iconico protagonista della *Metal Gear Saga*, Solid Snake è amatissimo dai fan, proprio per la sua veridicità: si conosce il suo profilo psicologico, ha delle ritualità ed un background mostrati a schermo, un'evoluzione con l'avanzare dei capitoli ed un finale; è un personaggio ben costruito ed ultra caratterizzato proprio grazie all narrativa costruita appositamente per il gioco.

Creare una narrativa fruibile ed accattivante non è cosa semplice: più si rivela al lettore , più paletti metto alla storia, più aumentano le difficoltà di mantenimento della coerenza; si deve centellinare quello che si racconta e quello che si rivela sul personaggio o sulla storia. Lo spettatore, il giocatore, il lettore non deve essere trattato da stupido, ma progredire dentro la storia essendone partecipe. Lo spessore dei protagonisti è costruito attraverso la creazione di rituali e world building, assegnando caratteristiche credibili al personaggio: conflitti, approfondimenti, rituali, sono tutte meccaniche narrative che devono rimanere invisibili, ma fondamentali per creare empatia con i vari personaggi. L'introduzione di questi deve avvenire non con un *pop-up effect*, ma piuttosto con un meccanismo di *inception* basato su collegamenti narrativi: elementi messi apparentemente a caso nella mia storia, consentono di introdurre personaggi o vicende in maniera naturale ed armonica. Un'altra parte fondamentale per una storia è il ritmo, il quale consente al pubblico di rimanere ancorato ed attento a quello che sta fruendo. E non bisogna avere un ritmo forsennato o troppo lento, piuttosto un equilibrio che generi dei sali/scendi utili alla scorrevolezza della trama. Spesso si utilizzano due trame: una **orizzontale** ed una **verticale**. La prima si riferisce alla trama di base del prodotto, quella che vede il viaggio dei personaggi iniziare e concludersi; la seconda riguarda la trama di un singola quest o livello, che va ad arricchire la trama orizzontale. Questo meccanismo è fondamentale per la narrativa seriale e contribuisce alla ritmica della narrazione: flashback, cambi di personaggio, punti di vista narrativi.

Figura 72: Importantissime per la narrativa sono le *side quest*: in *The Witcher 3: Wild Hunt*, sviluppato da **CD Project Red** nel 2015, vi sono numerosissime *side quest* (missioni secondarie) che supportano la *main quest* nella costruzione della storia, sia del mondo in cui si gioca, sia di Geralt di Rivia, il protagonista.

Per concludere,a livello tecnico nella sceneggiatura video-ludica va indicato tutto: non solo i dialoghi, ma ogni elemento utile a referenziare la narrazione, come la posizione di oggetti, i personaggi, il luogo, il tempo, tutto ciò che serve per fornire uno schema chiaro della visione narrativa del prodotto. La scelta su cosa mostrare e cosa no, è determinata dalle risorse disponibili o dall'attenzione di chi costruisce la storia.

3.4.2 La Scelta della Tecnologia

Quando si inizia a creare un videogioco, solitamente gli sviluppatori in erba non si pongono troppe domande su quali tecnologie usare per creare il proprio progetto, ma è un errore madornale non farlo: la scelta tecnologica concorre direttamente al tipo di esperienza e game design che verrà sviluppato. La scelta tecnologica si basa fondamentalmente sul tipo di **game engine** utilizzare per la realizzazione del proprio prodotto. I game engine o, in italiano, *motori grafici* sono la parte principale software che sta dietro la creazione di un videogioco con grafica in tempo reale; la grafica 3D ha apportato una rivoluzione nel campo dello sviluppo tecnologico: hanno cambiato gli hardware, i software e sono cambiate anche le competenze richieste ai programmatori del settore.

Figura 73: Il termine "motore grafico" è nato intorno alla metà degli anni novanta. Allora ci si riferiva solo ai videogiochi 3D, mentre ora, si è molto diffuso anche su altri generi di giochi e non solo 3D. Divennero popolari dal 1993, quando uscirono due videogiochi: *Doom* (1993) e *Quake* (1996) (in figura), gli unici fino a quel tempo, ad utilizzare motori grafici. Vista la popolarità di quei due giochi, molte altre aziende invece di creare da zero i loro giochi preferirono comprare da chi creò Quake e Doom pezzi di codice e usarli come base per il loro gioco.

Molti dei programmatori grafici scrive un codice che gira sulla GPU e non sulla CPU standard: è parte integrante della strategia di business nel mondo gaming, la **riutilizzabilità** del codice. I game engine servono proprio a spingere in questa direzione: software che forniscono ambienti di sviluppo *preimpostati* in cui costruire le proprie produzioni, permettendo al programmatore di dedicarsi solo agli aspetti specifici del gioco, quindi sfruttare quello che è già presente all'interno del motore stesso. Questo dipende da che game engine si sta usando; in generale, ogni motore grafico che si rispetti deve avere alcune funzionalità di base: gestione degli spazi, collisioni, fisica, intelligenza artificiale, suono, animazioni, ecc. Ma ciò che rende diversi i vari motori grafici è il motore di **rendering**, essenziale in ogni game engine, elaborando informazioni in ingresso e le trasforma in rappresentazione grafica. I motori grafici si basano su dati forniti dai pacchetti integrati e dai dispositivi di input (tastiera, mouse, ecc), quindi fanno da "medium" tra hardware e

software. I motori grafici sono costruiti nella maggior parte dei casi su delle **API grafiche**; per consentire ai motori grafici l'accesso hardware vengono usate librerie di basso livello tipo DirectX. Prodotti come **Unity**, che punta tantissimo sulla democratizzazione dello sviluppo video-ludico, hanno consentito alla creazione di numerosi studi indipendenti, riducendo la barriera d'ingresso nel settore per molti programmatori e developer; altri come **Unreal Engine 4**, puntano ad offrire qualità più elevata, più vicina ai prodotti tripla A. Va sottolineato come grazie alla sempre maggiore specializzazione dei programmatori in ambito di game engine, durante lo sviluppo di numerosi giochi sono stati prodotti grazie all'utilizzo di engine prodotti ad hoc.

Figura 74: Per lo sviluppo di **Horizon Zero Down**, sviluppato per Ps4 da **Guerrilla Games** nel 2017, è stato utilizzato un motore di gioco customizzato chiamato **Decima**: concepito ed utilizzato la prima volta per un altro titolo prodotto dalla software house, ovvero *Killzone: Shadow Fall*, è un engine che possiede sistemi di AI, un *motore fisico* integrato ed è in grado di supportare la risoluzione 4K e l'uso dell'*high dynamic range imaging*.

La varietà e le possibilità offerte dai game engine, permettono agli sviluppatori di fare scelte ben specifiche in funzione del design che poi verrà implementato: se si ricerca il realismo, occorrerà optare per un determinato motore grafico, ma dovrò tenere conto dei costi produttivi, sia in termini tecnici che economici; se opto per l'utilizzo di un motore *open source*, potrò lavorare su un progetto con meno costi produttivi, ma con possibilità realizzative sicuramente inferiori. Ad esempio, *CryENGINE* è il motore di gioco utilizzato da **Crytek**, un'azienda fondata dai fratelli Cevat Yerli, Avni Yerli e Faruk Yerli: CryENGINE vanta una grafica tra le più potenti di tutti i motori di gioco attualmente sul mercato.

Figura 75: *Crysis*, sviluppato nel 2011 da **Crytek**, ad oggi risulta ancora uno dei giochi con la grafica più potente da supportare per le piattaforme da gioco in commercio.

D'altro canto *Frostbite Engine*, creato da **Digital Illusions CE** viene utilizzato per creare *sparatutto* (genere in cui il gameplay è focalizzato sull'uso di armi da fuoco) in prima persona ed ha svolto un ruolo importante in quasi tutti i giochi di **EA Games**, come ad esempio le ultime iterazioni di *FIFA*. *Anvil Engine* è stato creato da **Ubisoft** per essere utilizzato dall'azienda ed è molto apprezzato per i livelli elevati di intelligenza artificiale. Mentre *Source* è un motore di gioco 3D, sviluppato da **Valve Corporation** ed uscito nel 2004, molto flessibile; fra le sue caratteristiche si possono citare il sistema di render basato sugli *shader*, l'accurata sincronizzazione labiale, un sistema avanzato per realistiche espressioni facciali e un modello fisico molto accurato.

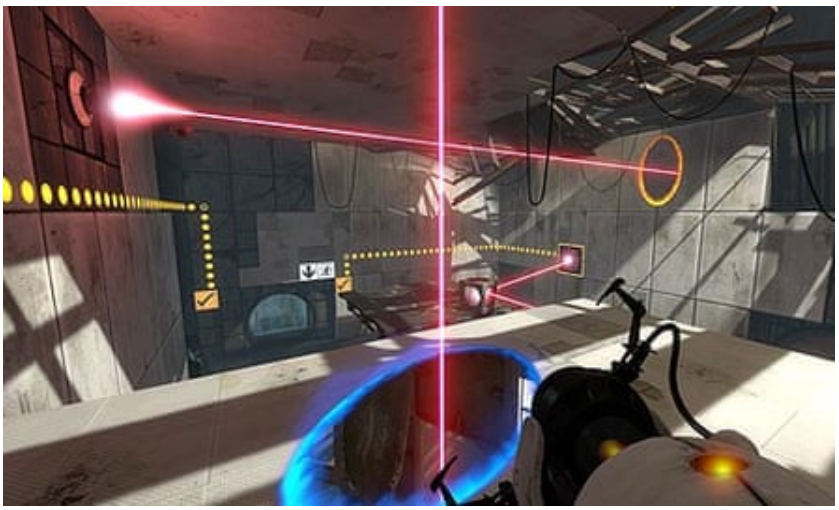


Figura 76: *Portal 2*, sviluppato da **Valve** nel 2011, è un videogioco "rompicapo" dove viene sfruttata pienamente la simulazione fisica interna del Source Engine.

I game engine non sono gli unici ingranaggi che determinano la scelta tecnologica, quindi la direzione artistica del progetto; infatti se i motori grafici riguardano il lavoro del programmatore o dei developer più tecnici, esistono altri software sviluppati appositamente per facilitare il lavoro dei 2D/3D artist: sono programmi dedicati alla realizzazione di modelli tridimensionali; i più noti sono **Maya** o **3D Studio Max** dato che sono i più utilizzati all'interno dell'*industry*, ma non meno potenti o funzionali sono **Blender**, **Modo** o **Houdini**. Applicazioni di questo tipo consentono di modellare personaggi, oggetti e ambienti partendo da un semplice solido geometrico composto da poligoni, agendo poi con appositi strumenti di trasformazione, sulle componenti di tali poligoni.

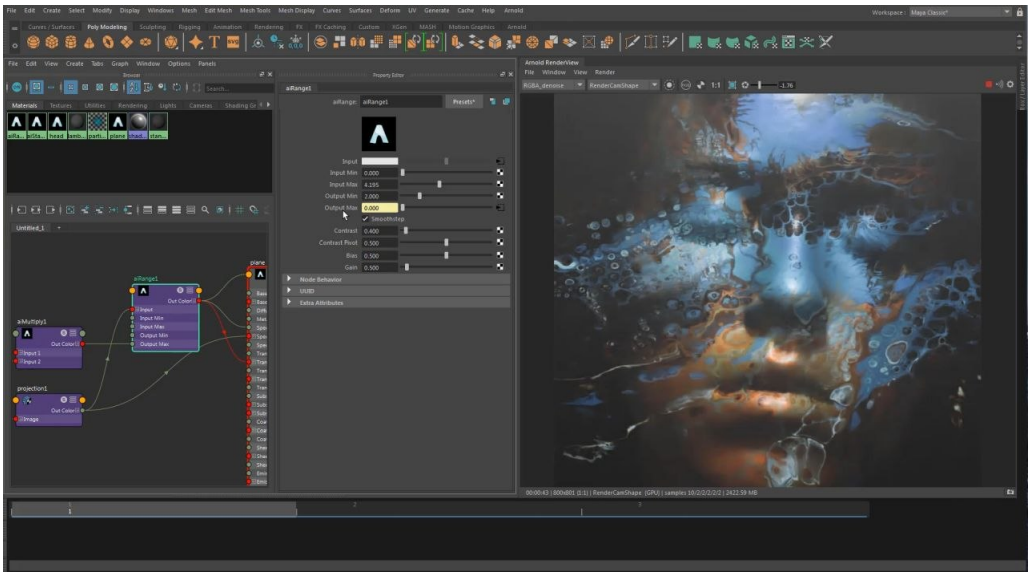


Figura 77: **Maya** è un potentissimo tool di modellazione utilizzato fortemente nel settore del gaming con una struttura logica basata sui nodi.

Altri software hanno invece un approccio molto più vicino alla scultura tradizionale e alla modellazione della creta e dell'argilla: **ZBrush** è il più famoso e permette di operare su un enorme numero di poligoni con strumenti di scultura molto simili a quelli analogici, garantendo un altissimo livello di dettaglio. Dal momento che i motori grafici non riescono a gestire in tempo reale un grandissimo numero di poligoni a schermo, la prassi lavorativa è quella di trasferire tale livello di dettaglio dal modello *high-poly*, composto da milioni di poligoni, a quello *low-poly*, composto da un numero ridotto di poligoni e che andrà inserito nel motore di gioco, tramite un processo *baking*. Il risultato di tale processo è la cosiddetta *normal map*, ovvero una **texture**, un'immagine bidimensionale; questa, una volta applicata al modello low-poly, darà l'illusione della presenza di una gran mole di dettagli che, pur non essendo materialmente presenti in forma di poligoni nel modello low-poly, reagiranno coerentemente alle variazioni di illuminazione.

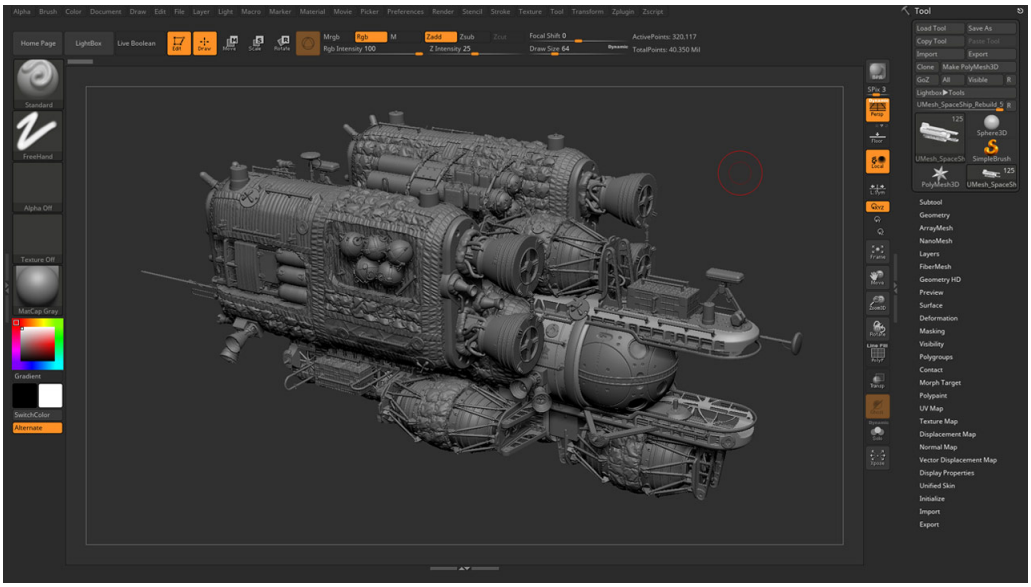


Figura 78: **ZBrush** è un software importantissimo per la modellazione più tradizionale: l'utente infatti agisce come uno scultore e deve avere conoscenze anatomiche per realizzare progetti di un certo tipo.

Le texture servono a caratterizzare personaggi, oggetti ed ambienti: la loro produzione è affidata ad artisti che lavorano con software specifici, o anche il più generico *Photoshop*, che si stanno diffondendo rapidamente anche altre applicazioni che consentono di "colorare" direttamente l'oggetto tridimensionale: i prodotti più diffusi sono quelli dell'*Allegorithmic* e della *Quixel*, ma sono usati anche *3DCoat* e *Mari*.
Per semplificare o velocizzare alcune fasi si inseriscono nella *pipeline* di produzione altri software sussidiari, come *Marvelous Designer*, utilissimo per realizzare vestiti e tessuti, *SpeedTree* per la vegetazione e *WorldMachine*, per terreni di grande estensione.

3.4.3 L'Importanza dell'Estetica

Direttamente conseguente alla tecnologia utilizzata, il comparto grafico e visivo sono centrali per tutto lo sviluppo di un videogioco; non si tratta solamente della qualità grafica o della potenza grafica proposta, piuttosto si tratta di quanto sia centrale l'impianto estetico di un'opera video-ludica, essendone lo scheletro e l'anima, sia per quanto riguarda la narrativa che la popolarità sul mercato.

Una precisazione appare necessaria: parlando di **estetica**, ci si riferisce alla componente stilistica di un videogioco e conferisce all'opera un'identità visuale; è sempre questa che verrà presa in esame dopo anni, quando la tecnologia sarà obsoleta. Proprio per questa motivazione, lo stile estetico può avere un peso addirittura pari se non superiore a quello del gameplay: così come delle pessime meccaniche di gioco sono da condannare anche nei titoli più splendidi visivamente, un'estetica insufficiente è sempre un difetto, persino nel gioco più divertente del mondo.

Nel panorama dei giochi più venduti la raffinatezza tecnologica della grafica ha raggiunto lo stato dell'arte, permettendo agli sviluppatori di creare ambientazioni foto-realistiche di altissima qualità. Senza scomodare i più celebri giochi *multi-player* (per i quali entrerebbero in gioco numerosi altri fattori rendendo il discorso sin troppo complesso), un *single player* di qualità deve farsi notare in qualche modo anche dal punto di vista stilistico.
Uno dei maggiori esempi di successo, in questo senso, è *Horizon Zero Dawn* che, seppure non innovativo nella formula, ha catturato milioni di giocatori grazie alla sua personalissima ambientazione preistorico-futuristica popolata da grandi bestie robotiche. Se le meccaniche di gioco non fossero state realizzate con criterio, comunque, un superficiale appeal estetico non sarebbe bastato: è il caso di un grande flop dell'ultima generazione di console, l'esclusiva Sony *The Order: 1886*. Il gioco vantava un'estetica steampunk ispiratissima, con riferimenti a *The League of Extraordinary Gentlemen* di

Alan Moore, ma alla prova dei fatti si trattava di un banale *third person shooter* ed il responso di critica e pubblico fu tiepido.

Figura 79: *The Order: 1886*, sviluppato in esclusiva per Ps4 da **Ready at Dawn**, fu uno dei titoli di lancio della console Sony, presentando un comparto estetico davvero di rilievo, a fronte di un gameplay banale ed una durata non superiore alle 7 ore di gioco.

Lo stile da solo, dunque, non basta, ma rimane un fattore molto importante: anche un titolo tutto sommato ben realizzato ludicamente, come *Days Gone* (**SIE Bend Studio**, 2019), frenato da una natura derivativa, anche nell'estetica che proponeva l'ennesimo gioco a tema "apocalisse zombie" senza particolari idee che lo differenziassero dalla massa. Infatti, sempre nell'inflazionato contesto delle epidemie globali, *The Last of Us* (**Noughty Dog**, 2013) si è guadagnato lo status di indiscusso capolavoro, abbinando una storia e personaggi di spessore ad un gameplay solido, il tutto accompagnato da un lavoro estetico efficace e personale, tanto nelle ambientazioni devastate, quanto nei vari tipi di infetti che popolavano il mondo di gioco.

Nel mercato indie, invece, il discorso cambia: per sviluppatori indipendenti dalla limitata disponibilità finanziaria, l'estetica è un eccellente biglietto da visita per farsi notare nell'enorme calderone dei più utilizzati store digitali. In un mondo dove la concorrenza è tanto numerosa, le buone idee sono fondamentali per farsi apprezzare, ma se non si cattura in qualche modo immediato l'attenzione dell'acquirente sarà difficile che qualcuno possa effettivamente provare il gioco e innescare così il meccanismo del passaparola. *Cathedral* (**Decemberborn Interactive**, 2019) è un ottimo *metroidvania* ma, in due mesi e mezzo dall'uscita, ha racimolato meno di sessanta recensioni su Steam: probabilmente perché l'estetica del gioco ricorda in maniera quasi spudorata il ben più famoso *Shovel Knight* (**Yacht Club Games**, 2014) e trasmette quindi un messaggio di scarsa personalità, il che sicuramente farà desistere la maggior parte degli acquirenti. Abbracciare un particolare stile visivo aiuta a ritagliarsi nicchie di pubblico: è il caso dei *retro-FPS* degli ultimi anni come *Dusk* (David Szymanski, 2018) e *Ion Fury* (**Voidpoint LLC**, 2018), rispettivamente con un'estetica retro ispirata a *Quake* e un motore grafico bidimensionale basato sullo storico **Build Engine**.

Figura 80: *Cathedral* (a destra) è un *metroidvania* di ottima fattura, ma proprio questo suo ispirarsi a titoli vecchio stile lo fa perdere nel vasto parco titoli del genere di cui fa parte, avendo una personalità non troppo spiccata o originale; infatti risulta troppo simile a *Shovel Knight* (a sinistra), titolo dello stesso genere che ebbe un ampio successo nel 2014.

La scelta estetica porta con sé delle implicazioni sullo stile di gioco che il titolo vuole presentare, riuscendo così a farsi notare in modo molto naturale dagli appassionati del genere. Più lo stile del gioco è unico, più possibilità si hanno di farsi notare, ma a quel punto sono il sistema di gioco, la trama e gli altri tipici fattori di valutazione a fare la differenza.

Ma l'estetica non ha solo un valore superficiale o promozionale: hanno una funzione molto più centrale una volta in *sinergia* con le altre componenti del titolo. La componente visiva è fondamentale per l'identità e l'immersività del gioco: ad esempio, senza l'inconfondibile stile grafico di Akira Toriyama, probabilmente il franchise di *Dragon Quest* non sarebbe diventato così iconico. La scelta di un'estetica lo-fi portata all'estremo ha imposto agli sviluppatori degli horror atipici *The Last Door* (**The Game Kitchen**, 2013) e *Faith: The Unholy Trinity* (**Airdorf Games**, 2019) di trovare dei modi creativi di inquietare il giocatore, diventando così dei piccoli cult nelle proprie nicchie di riferimento. Giochi come *Okami* (**Capcom**, 2006),il cui mondo di gioco realizzato come un dipinto giapponese è una tavolozza in cui il giocatore stesso può dipingere, creando e manipolando gli elementi a seconda dei propri desideri.

Figura 81: Lo splendido mondo di *Okami*, realizzato da **Capcom**, permette al giocatore di colorare e disegnare nel mondo oggetti o forme che gli consentano di avanzare nell'avventura, il tutto tramite l'uso del controller.

3.4.4 La Centralità delle Meccaniche

Nei precedenti paragrafi sono stati presi in esame elementi che caratterizzano fortemente il gioco, sia a livello identitario sia a livello immaginifico, manca l'analisi su quello che rappresenta il *core* di un gioco, ovvero le sue **meccaniche**. Con "meccanica", nella definizione più generale, si intende il processo con cui vengono regolate le azioni interne al gioco stesso o, per essere più precisi, sono le procedure con cui le risorse si relazionano con gli eventuali effetti: allargandone la definizione, l'insieme delle meccaniche di gioco forma la **dinamica** di gioco. Infatti se le meccaniche rappresentano gli strumenti, i mezzi e le logiche da utilizzare per raggiungere l'obiettivo posto dal gioco, le dinamiche rappresentano le azioni ed i comportamenti fondamentali che il gioco deve favorire nei giocatori affinché questi, tramite le meccaniche, siano sempre spinti a raggiungere l'obiettivo finale.
Chiaramente sono le meccaniche, quindi la dinamica di gioco, che rendono un gioco degno del suo nome e più sono originali, solide ed accattivanti, più il gioco assume identità, importanza e rafforza i pilastri analizzati precedentemente.

Figura 82: In *Prince of Persia: The Sands of Time*, sviluppato da **Ubisoft** nel 2003, il giocatore poteva sfruttare una particolare meccanica di gioco: nel pugnale del protagonista sono racchiuse le Sabbie del Tempo, che se rilasciate grazie all pressione di un tasto del controller, permettevano al giocatore di riavvolgere, fermare o rallentare il tempo.

In generale, aggiungendo nuove meccaniche, sistemi o modalità d'interazione si raggiunge una maggiore complessità delle dinamiche di gioco. Ad esempio, si confrontino *Scacchi* e *Dama*: Scacchi ha sei tipi diversi di pezzi (invece di due) e un numero maggiore di azioni che ogni pezzo può fare, quindi è più profondo strategicamente. Non necessariamente la complessità è un male: *Tetris* (**Sega**, 1984) è un gioco molto semplice ma di successo, mentre *Advanced Squad Leader* (**Don Greenwood**, 1985) è un board-game incredibilmente complicato, ma è considerato comunque un successo. Infatti meccaniche complesse non necessariamente portano a dinamiche di gioco complesse: esistono casi in cui meccaniche semplici creano una complessità estrema (vedi Scacchi). Ci sono altri casi in cui le meccaniche sono estremamente complicate, ma le dinamiche sono semplici.

Figura 83: Spesso i giochi prodotti da **Nintendo** ripropongono lo stesso franchise, focalizzando la nuova iterazione del gioco su una meccanica/dinamica specifica ed innovativa: in *The Legend of Zelda: Wind Waker*, sviluppato per GameCube nel 2003, venne inserita la possibilità di utilizzare una barca a vela per esplorare un vastissimo mare ricco di isole e punti di interesse, grazie all'evocazione del vento da parte del giocatore.
Su questa meccanica ruota l'intero gioco, proponendo un open-world innovativo ed originalissimo per l'epoca.

In alcuni casi alcune meccaniche/dinamiche di gioco vengono sfruttate per sopperire ad alcuni limiti tecnici, che altrimenti non consentirebbero l'avanzamento nello sviluppo; scelte di design accurate che supportano l'hardware e alleggeriscono l'intero prodotto in termini di costi produttivi.

Figura 84: In *Silent Hill*, classico della prima Playstation, sviluppato da **Konami** e **Team Silent** nel 1999, la nebbia veniva utilizzata per creare atmosfera e terrorizzare il giocatore, nascondendovi i mostri al suo interno, ma non solo: la nebbia serviva ad alleggerire lo sforzo computazionale dell'hardware Sony, nascondendo parti della vasta mappa al giocatore, che venivano caricate di volta in volta al momento utile.

Concludendo, si può affermare che le meccaniche rimangono centrali per quanto riguarda lo sviluppo di un videogioco: i giochi migliori sono quelli in cui la narrazione e l'estetica si integrano armonicamente al gameplay, quindi le meccaniche sono un tutt'uno con gli altri aspetti del progetto, raggiungendo uno stato di coerenza totale, a cui non è semplice arrivare.

3.5 Costruire un team: le figure professionali

Come è già stato scritto precedentemente, la creazione e la distribuzione di un videogioco è un processo lungo e complesso comprendente differenti "attori", ognuno con un compito ben specifico, necessari alla realizzazione di un videogame di successo.
In cima vi si trova il **producer**: un'azienda, un individuo, o un gruppo di investitori interessati al finanziamento della creazione di un videogioco, con delle caratteristiche specifiche, destinato ad uno o più mercati differenti (mobile, console, PC). Solitamente, il produttore possiede i fondi necessari a produrre un gioco. le conoscenze del mercato utili alla riuscita del progetto in termini pubblicitari, ma gli mancano ovviamente le conoscenze tecniche per svilupparlo: è qui che entra in gioco il **team** di sviluppo, un eterogeneo gruppo che possiede le competenze adatte allo sviluppo e produzione di un videogioco. Spesso può capitare il contrario, ovvero esiste già un team che lavora ad un videogame, ma non ha i fondi necessari per terminare o pub-

blicizzare il prodotto: e qui entrano in gioco tutte le questioni legate alla partnership ed alla presentazione del proprio gioco a differenti investitori.

Con il termine **sviluppatore** si intende l'azienda o il gruppo di persone con le competenze tecniche e creative per sviluppare e produrre un videogame. Per via della natura eclettica del mondo video-ludico, non si può pensare ad una formula standard per la composizione del team: ogni prodotto richiede una specifica composizione del team e figure tecnico/creative ben delineate, basti pensare che per un gioco in prevalenza 2D occorreranno molti più disegnatori anziché modellatori 3D e viceversa.

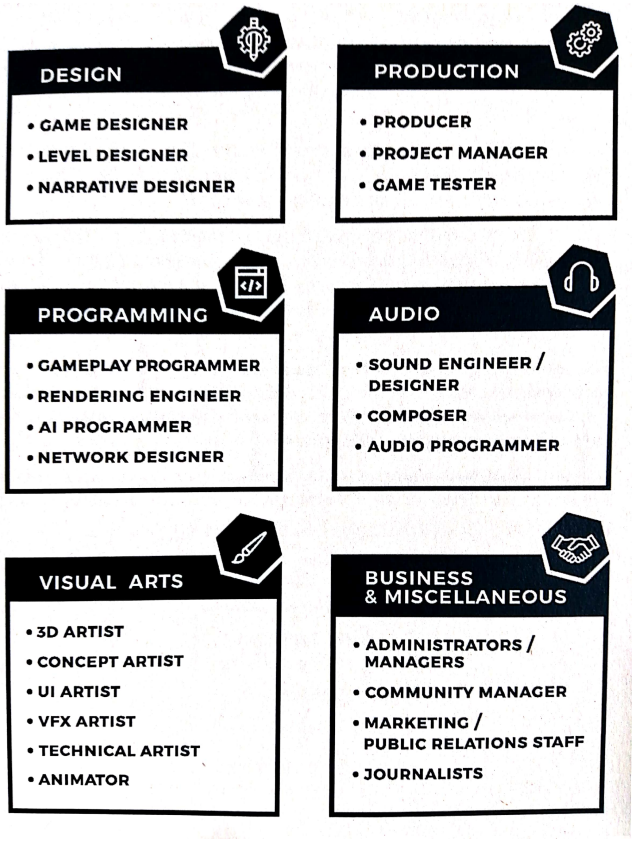


Figura 85: I settori coinvolti nella produzione di videogames sono tra i più disparati e coprono diversi ambiti dello scibile umano.

L'area legata al **design** si occupa di progettare il videogioco: l'idea, la storia ed ogni singola interazione e decisione che il giocatore prenderà all'interno del gioco. La **produzione** si occupa della gestione del team e delle scadenze previste dal progetto: è fondamentale rispettare i tempi per non ritrovarsi sovraccarichi nelle fasi dello sviluppo del gioco, anche perché il fenomeno del *crunch* non è per nulla positivo sull'andamento del progetto.
Il team di **programmazione** si occupa ovviamente del codice di gioco, necessario alla mera esistenza del prodotto; un altro compito del team riguarda l'ottimizzazione del videogioco, cercando di raggiungere le migliori prestazioni possibili. Il reparto **audio**, o del sound design, crea ed implementa ogni elemento audio/sonoro presente in game, dalla soundtrack fino ai più piccoli rumori.
L'area **visual arts** si occupa di ogni elemento visivo del gioco: concept art, modelli 3D, effetti visivi ed elementi di UI; è un comparto molto variabile a seconda del gioco che si va a produrre, poiché comprende anche tutta la parte legata alle animazioni. Ultimo, ma non meno importante, è il reparto **business** che si occupa della gestione economica del team e del progetto, del marketing interno o aziendale e della creazione e gestione economica del team e del progetto, del marketing interno o aziendale e della creazione e gestione della comunità che si può creare attorno ad un videogame o uno sviluppatore.
Tutti questi reparti collaboreranno alla creazione del videogame seguendo una linea di produzione (**pipeline**) ed una gerarchia ben precisa.

Il **lead designer** e l'**art director** lavorano insieme alla visione del gioco, creando le regole visive e meccaniche che il gioco ed il team devono seguire. Il lead designer divide la sua visione in elementi più piccoli, che vengono passati ai vari **designer** che si occuperanno di progettare quella parte del gioco.
L'art director segue lo stesso processo con i **concept artist**, dividendo gli elementi visivi del gioco tra i membri del proprio team. Quando la fase di progettazione è completata, i designer passano tutta la documentazione ai programmatori (ovvio questo è un'indicazione generale) che cominceranno l'implementazione e stessa cosa faranno i concept artist che passeranno i loro disegni ai modellatori. I vari team di **animazione** ed il team **audio** lavorano in modo più indipendente rispetto agli altri gruppi, seguendo le indicazioni del lead designer.
Nelle prossime sezioni verranno analizzate con più attenzione le figure del game designer, del programmatore, del 3D/2D artist e del publisher.

3.5.1 Game Designer

Sakaguchi, **Miyamoto**, **Shafer**, **Meier**, **Romero**, **Kojima**, sono solo alcuni "eroi" degli appassionati del mondo video-ludico: menti brillanti ed autori capaci di innalzare il termine di game designer ad un livello talmente

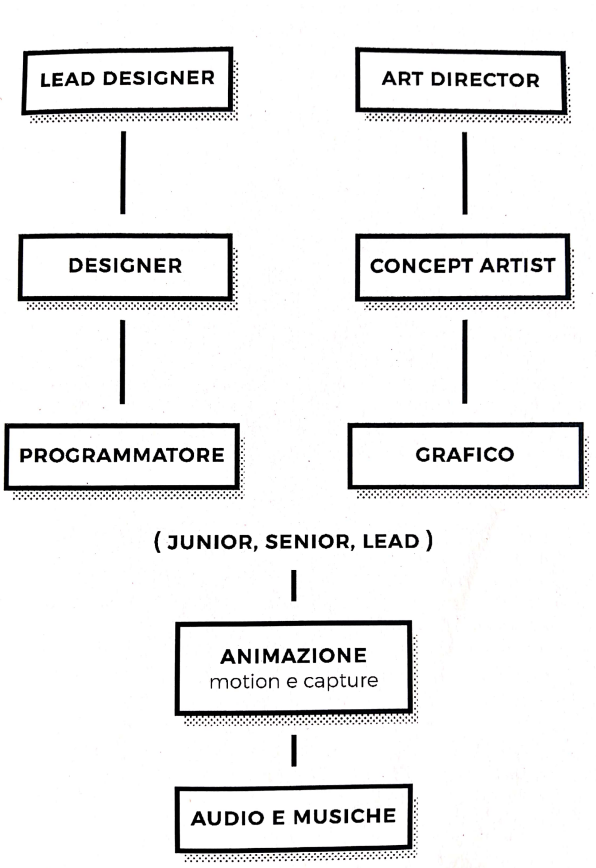


Figura 86: Gli sviluppatori di videogame possono sinteticamente riassumersi in questo schema.

alto, da far sfumare i confini della reale area di definizione del loro lavoro. Forse proprio per la natura intrinseca del ruolo, che il termine di game designer, e l'annesso campo professionale (game design), diventa poco chiaro e spesso soggetto a fraintendimenti. Molto spesso sui social o nei forum dedicati ai videogiochi, il termine viene confuso e si leggono spesso definizioni fantasiose su questa "nuova" figura professionale spesso associata a "colui che scrive le storie dei videogiochi" oppure "il creativo che fornisce le idee". Queste sono solamente due di tante affermazioni improprie che tentano di rispondere alla domanda "Che cosa fa il Game Designer? Cos'è il Game Design?": in questa sezione, si cerca di fare chiarezza attraverso esempi e definizioni in questo ambito, cercando di eliminare alcune convinzioni che spesso non rispecchiano la realtà delle cose.

Cos'è il Game Design? Chi è il Game Designer? Solitamente si tende ad utilizzare erroneamente il termine game design solamente in ambito

video-ludico: i giochi di carte o i giochi da tavolo in generale, sono attività ludiche e, come tali, possono presentare game design, seppur semplice o di facile concezione. Si pensi al seguente esempio: la meccanica classica ed inflazionata dello scontro-elementale (es: Acqua batte Fuoco, Fuoco batte Erba, Erba batte Acqua, ...) tipica di moltissimi giochi di ruolo, è del tutto assimilabile al popolare gioco della morra cinese, quindi risulta chiaro che il concetto di game design nasce molto prima del videogioco per come lo si intende ora.

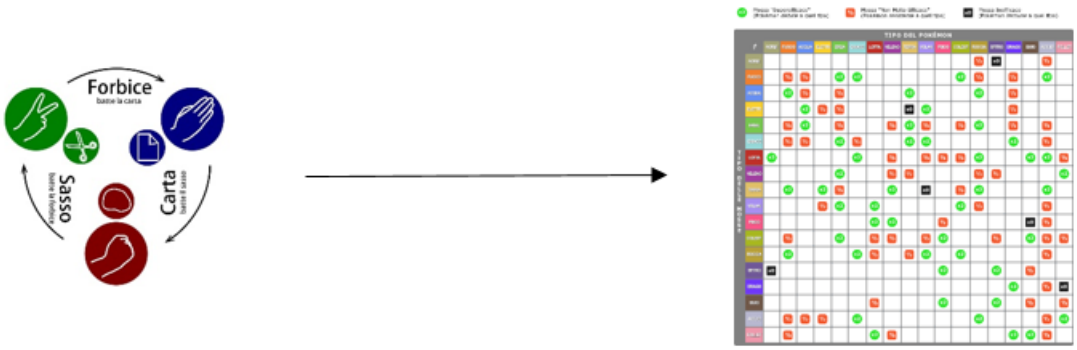


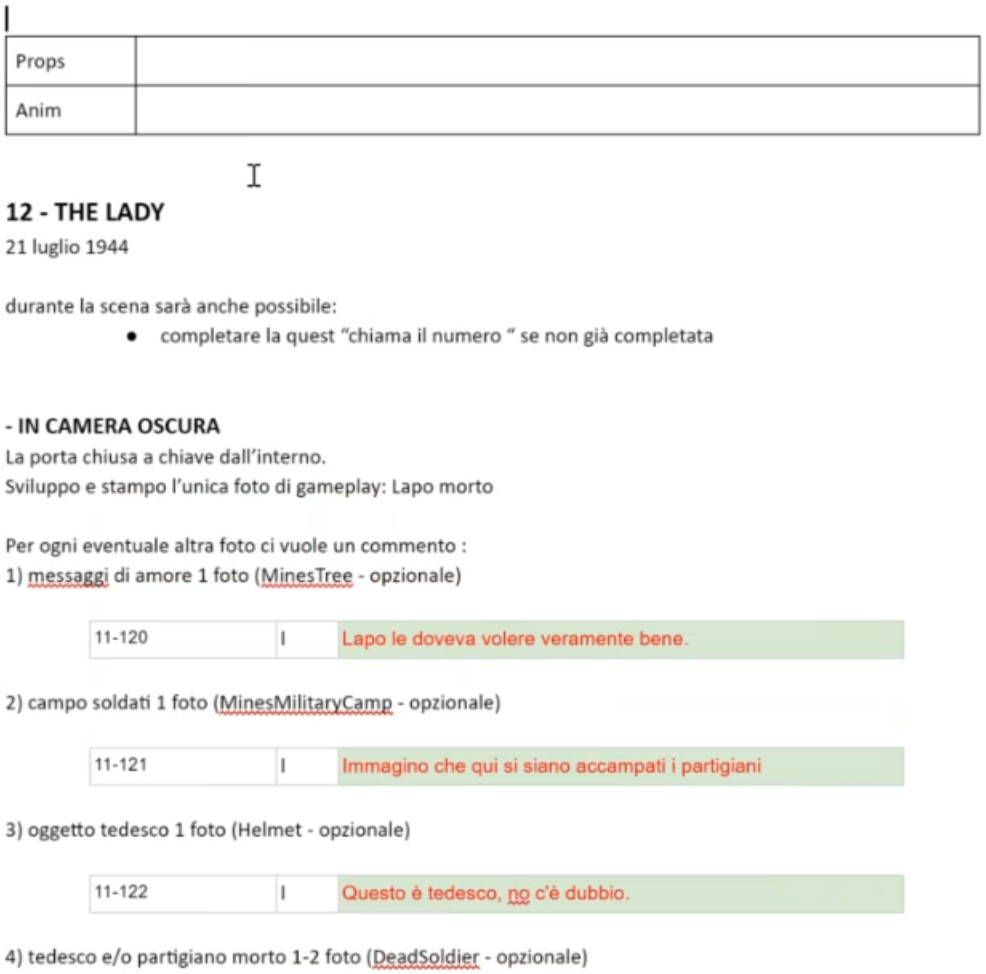
Figura 87: Il confronto tra lo schema della morra cinese e la sua evoluzione rivista nello scontro elementale del franchise *Pokémon*: lo schema logico di base è lo stesso, ha solamente subito un'evoluzione molto raffinata.

Il game design ha una storia antecedente al mondo dei videogiochi, ma è quella materia di studio e pratica che, in senso stretto, descrive il gioco in essere, le sue meccaniche e le regole per il suo svolgimento; da un punto di vista filosofico, si può asserire che è l'arte di creare/progettare delle esperienze, ma non l'esperienza in quanto tale: non sta al giocatore far sì che il gioco funzioni, ma è il design, che grazie ad una serie di sistemi visivo-cognitivi, forza "dolcemente" il giocatore verso il funzionamento corretto del gioco, quindi a seguirne le regole.
Proprio per questa natura in bilico tra arte e scienza, il game design abbraccia molteplici discipline che spaziano dalla psicologia all'antropologia, dal design all'architettura: secondo **Eric M. Lang**, il linguaggio essenziale della materia è formato da solide basi di Meccanica, Teoria dei Giochi e Psicologia Comportamentale, strumenti teorici che permettono la connessione tra le idee di chi progetta e di chi gioca. Risulta evidente che la figura del game designer è per natura trasversale e soventemente viene associata ad una sorta di "tuttofare" all'interno dei team di sviluppo di giochi.
Ma è realmente così? Si potrebbe rispondere con "dipende": in primis, le dimensioni di un team di sviluppo determinano fortemente il workflow del progetto, quindi le varie mansioni di un game designer; ad esempio, in team di sviluppo ristretti (es: 4-5 persone), il game designer può rivestire il ruolo di designer, ma anche di programmatore o di produttore, mentre in team composti da un numero di elementi molto superiore, le aree di lavoro diventano più settoriali e gerarchiche, dove vi sono interi sub-team dedicati a porzioni del gioco a cui si sta lavorando. Il sopracitato Kojima non è solamente un game designer, ma è anche un autore e director, a capo di altri game designer sotto di lui: si sta parlando di un caso virtuoso ed estremo del settore, ma aiuta a capire come la figura del game designer, possa assumere

differenti connotazioni e avere background culturali disparati. In secondo luogo, proprio questa ultima frase, porta ad osservare come numerosi professionisti del settore provengano da aree di studio completamente diverse fra loro: chi ha studiato nel mondo dell'architettura, chi proviene da settori più artistici, come la pittura e la fotografia, o più tecnici, come l'ingegneria e l'informatica. Va evidenziato che negli anni le cose sono cambiate (e cambieranno ancora), ma risulta evidente il trend della specializzazione: gli studi di sviluppo, infatti, ricercano sempre più figure altamente preparate, consci del fatto che un cattivo design può affondare un intero progetto, e, spesso, la reputazione della società stessa.

Figura 88: Nell'immagine è evidente come la dimensione di un team possa essere diversa: a sinistra, **Team Cherry**, creatori di *Hollow Knight*, e a destra la divisione **Ubisoft Toronto**, sviluppatori di giochi tripla A come *Far Cry 5*.

Questa rimane però una risposta vaga alla domanda iniziale: effettivamente, spogliando questa figura professionale di tutto il background possibile, che cosa rimane nella pratica? Il design, generalmente si occupa di realizzare il miglior connubio possibile tra forma e funzione, e questo si riflette anche nel mondo dei giochi e dei videogiochi. Preoccupandosi di facilitare la comprensione dell'interfaccia, rendendola intuitiva ed equilibrando il carico cognitivo, il game designer progetta e delinea ogni meccanica del gioco. Una delle principali capacità da affinare è proprio la *scrittura tecnica*, cioè la capacità di mettere su carta come il gioco si svolgerà, quali regole dovranno essere seguite e come le dinamiche di gioco verranno influenzate dall'azione dei giocatori: risulta chiaro che gran parte del tempo lavorativo del game designer si basa sulla stesura di questa tipologia di documentazione, indicando features e mantenendola aggiornata man mano che il progetto procede.



Props	
Anim	

12 - THE LADY

21 luglio 1944

durante la scena sarà anche possibile:

- completare la quest "chiama il numero " se non già completata

- IN CAMERA OSCURA

La porta chiusa a chiave dall'interno.
Sviluppo e stampo l'unica foto di gameplay: Lapo morto

Per ogni eventuale altra foto ci vuole un commento :
1) messaggi di amore 1 foto (MinesTree - opzionale)

11-120	I	Lapo le doveva volere veramente bene.

2) campo soldati 1 foto (MinesMilitaryCamp - opzionale)

11-121	I	Immagino che qui si siano accampati i partigiani

3) oggetto tedesco 1 foto (Helmet - opzionale)

11-122	I	Questo è tedesco, no c'è dubbio.

4) tedesco e/o partigiano morto 1-2 foto (DeadSoldier - opzionale)

Figura 89: Esempio di scrittura tecnica per la stesura di un documento di design, in particolare per la gestione del game flow e progressione nel gioco.

Il documento di design deve presentare, per quanto riguarda le meccaniche, tutte le azioni e reazioni tra le varie entità all'interno del gioco: ad esempio, come un'azione del gioco risponde ad un'altra azione del gioco, come funziona il mondo di gioco nel rapporto con il giocatore e, ovviamente, che dinamiche genera l'interazione del giocatore al suo interno. I compiti del game designer, oltre al coordinare la sezione tecnica e la sezione artistica, cercando di veicolare maniera chiara la propria idea, sono essenzialmente racchiusi in tre macro-aree principali: *creatività*, *progettazione*, *gestione/rifinitura*. Nel prossimo paragrafo si vedrà come, la combinazione di queste tre differenti macro-aree ed il possedere determinate **skill** consentano

di sviluppare un artefatto pronto per l'esperienza ludica, in particolare nel campo dei videogames.

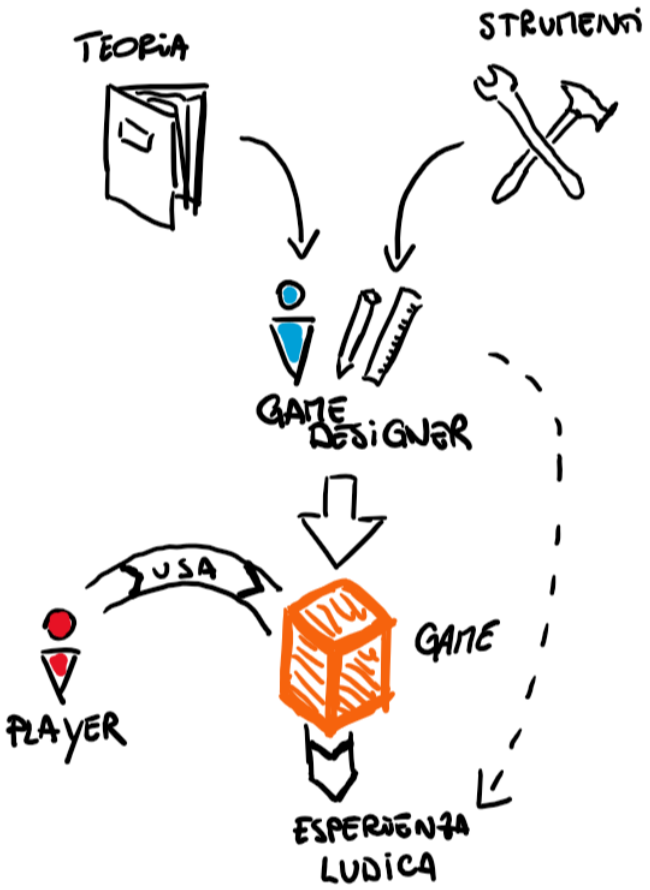


Figura 90: Schema del workflow di game design.

Precedentemente, si è data una definizione di game designer descrivendolo come un "architetto delle scelte": un gioco è tale perché l'interazione si realizza nella necessità di compiere delle scelte; il giocatore si trova di fronte una moltitudine di scelte, che però, a differenza della vita reale, non sono casuali, ma pensate, decise ed implementate dal game designer per suscitare emozioni e far vivere specifiche esperienze. Per creare tale prodotto, il game designer deve partire da un campo specifico, ovvero quello **creativo**. Occorre da subito fare una precisazione: il game designer non propone un'idea, ma piuttosto una **visione**; le idee possono averle tutti (ognuno di noi può avere un'idea per un libro, un film o un gioco), ma la visione è quella capacità di percepire, comunicare e provare ogni dettaglio inerente all'idea: ogni elemento, ogni features, ogni particolare vengono percepiti e collocati in uno schema mentale ben preciso. Si pensi ad uno dei ruoli più importanti

assumibili da un game designer è proprio quello di creative director, cioè colui che si occupa del mantenimento della coerenza del prodotto, durante il periodo di sviluppo: fa sì che la visione iniziale non venga snaturata o modificata radicalmente. Dalla fase creativa, si passa a quella di progettazione: la visione, che sia proposta dal game designer o da un cliente, va comunicata ed implementata in modo che tutte le figure del team di sviluppo riescano a capirla e assimilarla. Questo è un processo che richiede una mole di lavoro immensa: si comincia, solitamente, attraverso la target audience, ovvero lo studio di prodotti con visione simile o diametralmente opposta oppure lo studio di prodotti di moda, o che potenzialmente lo diventeranno; tutto ciò, al fine di carpire i "perché" alla base di ogni prodotto meritevole: ad esempio, perché ha successo? Perché determinate features funzionano? Perché altre non sono piaciute? Perché si usano questi colori? E così via.
Dopo questa ricerca e analisi, entrano in gioco le skill necessarie al game designer perché la propria visione possa diventare un progetto di successo: tutto viene cristallizzato nel game design document, ovvero un insieme di documenti che contiene tutto il lavoro di progettazione del game designer, scritto in modo che qualsiasi membro del team di sviluppo possa leggerlo e comprenderlo in ogni sfaccettatura. Infatti, il game designer scrive molti documenti a seconda del ruolo dei destinatari: se il documento serve ai programmatori avrà un certo tipo di linguaggio, mentre se deve essere letto dal reparto di concept art, avrà altre connotazioni linguistico-comunicative; è molto importante che la visione venga comunicata attraverso strumenti efficaci: non è raro l'utilizzo di mood board, software di schematizzazione grafica, l'uso di oggetti fisici per raccontare le dinamiche pensate, etc....

Figura 91: **Hideo Kojima** ed alcuni suoi collaboratori provano una scena utile per una cinematica durante lo sviluppo di *Death Stranding* all'interno degli studios della **Kojima Productions**. Un game designer deve sfruttare qualsiasi mezzo a sua disposizione per trasmettere la sua visione al team: in questo caso, anche la recitazione.

Una volta che la visione è assimilata da tutto il team e la documentazione risulta esaustiva, è possibile iniziare a creare il gioco vero e proprio; mentre le altre figure del team si occupano del prodotto in termini tecnico/artistici, il game designer deve occuparsi di due importanti questioni: la gestione del team e la rifinitura del prodotto, che in gergo viene chiamata fase di tuning e polishing. In particolare, il game designer deve organizzare la divisione del lavoro e la divisione dei compiti, gestire le relazioni interpersonali tra i membri del team, ottimizzare la ripartizione delle risorse lavorative e ragionare su eventuali compromessi da operare, tra ciò che si vuol fare e ciò che è possibile fare. Inoltre, è incaricato della gestione di elementi esterni al team, come la relazione con il cliente o con il publisher, la ricerca di fondi o la partecipazione a bandi; si deve occupare delle possibili interviste, presentazioni e partecipazioni ad eventi per la promozione del gioco: si potrebbe dire, che in questa fase il game designer si occupa del far passare la propria visione anche all'esterno del team, cercando di convincere il publisher o altri sponsor a supportare il progetto. Oltre alla sfera prettamente manageriale, esiste anche quella legata alla rifinitura: il game designer provvede a rifinire il proprio prodotto, che funziona a livello tecnico, ma non a livello qualitativo; si eseguono una serie di operazioni di test e tuning che richiedono

ore ed ore di lavoro: il gioco viene provato e riprovato al fine di ricercare bug e sperimentare tutti i parametri presenti all'interno del mondo di gioco, cercando di ottenere il feeling migliore.
Facendo un esempio concreto, si pensi alla velocità che deve avere un determinato personaggio in un free-roaming in terza persona: apparentemente può sembrare un compito facile, ma non è così poiché tale valore è dato da una molteplicità di fattori, come la curva di accelerazione o decelerazione, la velocità massima, la curva di pendenza, i moltiplicatori di clima o di terreno,.... È chiaro che ognuno di questi fattori deve essere impostato in maniera corretta o l'esperienza di gioco potrebbe risentirne: la ricerca di tali valori viene definita **tuning**.

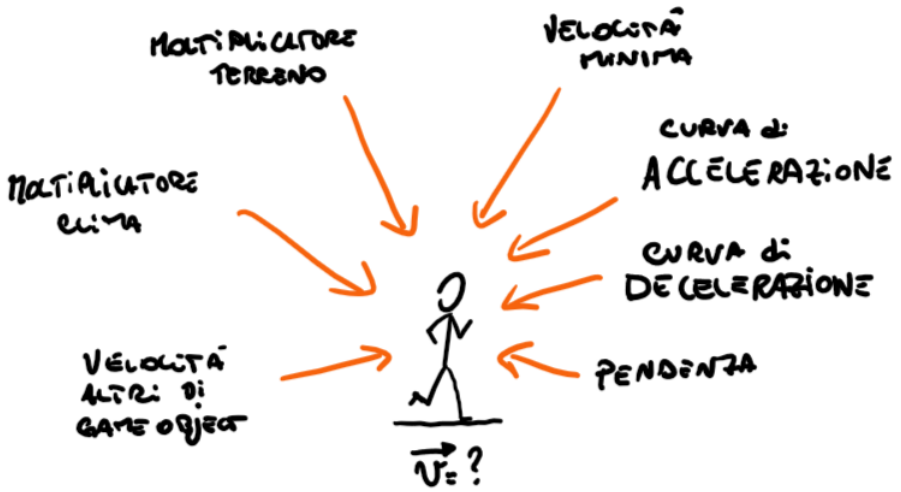


Figura 92: Possibili fattori che influenzano il valore della velocità di un personaggio all'interno di un videogame.

Per riassumere, si può asserire che il game designer è figura chiave durante tutta la pipeline di lavoro per lo sviluppo di un videogioco, ma i suoi compiti variano a seconda della fase di lavoro in cui si trova: nella prima fase (pre-produzione) fino allo sviluppo del prototipo, chiamato anche vertical slice, il game designer lavora sulla parte creativa e sulla progettazione del gioco, cercando di materializzare su "carta" la propria visione; una volta terminato il vertical slice, comincia il percorso del gioco verso la sua versione alpha: il prodotto presenta tutte le features desiderate, ma ancora non è rifinito, ed in questa fase il game designer lavora su tutte le tre macro-aree trattate; una volta superata la fase alpha, il game designer si occupa essenzialmente della gestione e della rifinitura del prodotto.

Tools & Skills Nonostante si sia cercato di smontare l'idea del game designer "tuttofare", rimane lampante la sua trasversalità, sia nel background che nei compiti professionali.
Nei grossi studi di sviluppo, dove c'è abbondanza di personale, la figura del game designer viene declinata a seconda di determinati settori di specializzazione: il **level designer**, ad esempio, è la figura che si occupa di implementare tutte le componenti realizzate dagli altri professionisti del team, all'interno di un engine grafico; semplificando si potrebbe paragonare il level designer ad un giocatore di LEGO, che allestisce la propria scena tramite mattoncini preparati da altri: grazie alle indicazioni presenti nel game design document, raccoglie tutti gli asset grafici (modelli poligonali, elementi 2D, 3D) realizzati dai 2D/3D artist, il codice sviluppato dai programmatori (necessario all'esecuzione delle azioni in game), sezioni cinematiche, i VFX e assembla una determinata componente giocabile dell'esperienza.

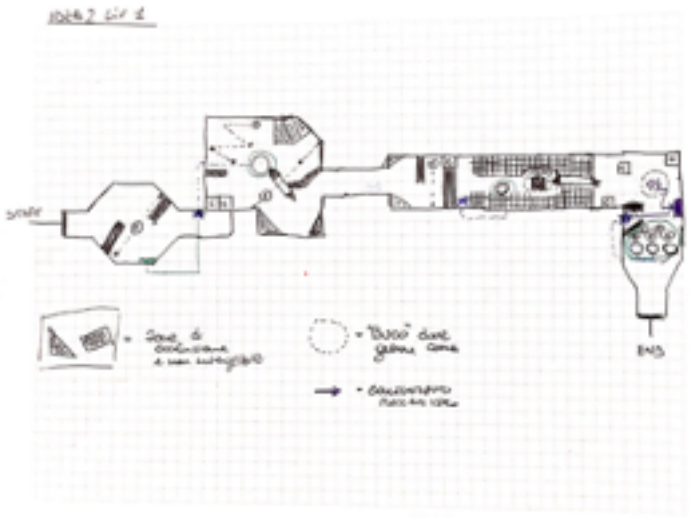


Figura 93: Esempio di schema o mappa utile per il lavoro di level design.

Altro esempio di specializzazione lo si ritrova nella figura del **narrative designer**: è un ruolo che ha cominciato ad affermarsi nell'industria video-ludica a partire dal secondo decennio degli anni Duemila, in seguito al successo di progetti come *Heavy Rain* (2010, **Quantic Dreams**) o *The Walking Dead* (2012, **Telltale Games**), anche se è possibile riscontrare elementi di narrative design simili già in periodi videoludici precedenti (si pensi alle avventure grafiche alla *Grim Fandango*, 1998, **Double Fine Productions**, **LucasArts**).

Figura 94: *Grim Fandango* (**Double Fine Productions**, **LucasArts**) è un avventura grafica dotata di una brillante storia ed è ritenuta molto significativa per l'intero sviluppo del medium videoludico, tanto che nel 2014 fu annunciata un edizione *remastered* uscita poi su console.

Il compito del narrative designer è quello di progettare elementi narrativi di un gioco e sostenere la struttura della storia durante le fasi di sviluppo: il videogioco, infatti, deve avere un significato e le azioni del giocatore devono essere guidate da una motivazione, cercando di instaurare un rapporto emotivo reale tra chi gioca ed i personaggi o le storie all'interno del mondo virtuale.

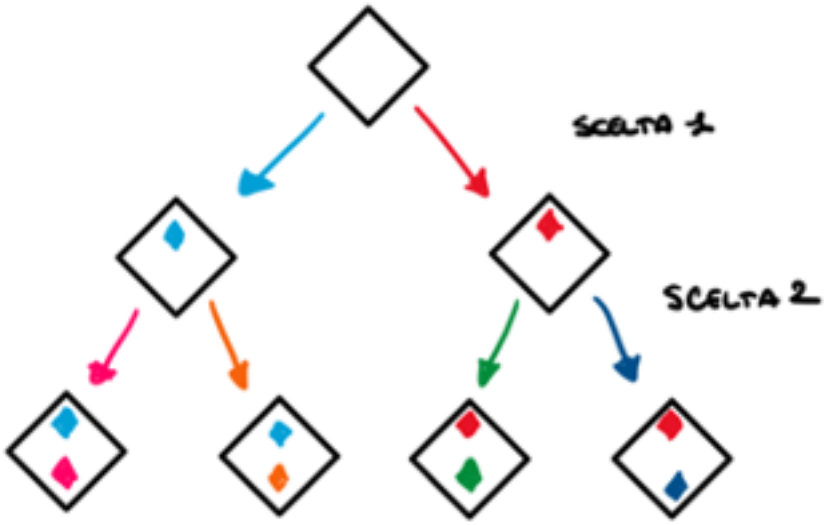


Figura 95: Esempio di *branching narrative*; ogni scelta porta ad uno sviluppo narrativo diverso, terminando con un finale proprio di quel percorso nodale.

Queste sono alcune delle possibili specializzazioni per un game designer, ma risulta evidente come rappresentino ambiti completamente diversi: sorge spontaneo chiedersi quali siano le skill che occorre sviluppare.
L'obiettivo di creare un gioco originale richiede davvero molteplici capacità e conoscenze a 360° a diversi ambiti: oltre alle sopracitate abilità manageriali e creative, un buon game designer deve avere nozioni fisiche, storiche, antropologiche, cinematografiche, matematiche e logiche; una tendenza al saper ascoltare, in modo che la gestione dei rapporti lavorativi non risulti un ostacolo alla buona riuscita del progetto, e al confrontarsi con il team, evitando di innamorarsi della prima idea trovata, ad esempio, attraverso sessioni di *brain storming* collettivo (questo consente di vagliare più idee ed esplorare più possibilità di sviluppo): esistono molti strumenti utili per schematizzare il gioco sia nelle sue componenti narrative che in quelle legate alle meccaniche (es: **Twine**, **Ren'Py**, **Construct**).
La comunicazione ricopre un ruolo essenziale, non solo con gli altri (*public speaking*), ma anche con se stessi: la capacità di ascoltarsi permette molto spesso di mettersi in discussione e non arenarsi su idee magari brillanti, ma non realizzabili per problemi legati al budget o a limiti tecnici. Riassumendo questo punto: avere capacità empatica e conoscitiva permette di risolvere problemi e creare occasioni di networking sia all'interno del team di lavoro che al di fuori.
È utile anche avere conoscenze in termini di hard skill: conoscenza in merito

ai codici di programmazione più usati, sapersi orientare all'interno di motori grafici (es: **Unity**, **Unreal Engine**), la padronanza di software di modellazione (es: **Blender**) o di software di prototipazione (es: **Machinations**), rendono la faretra del game designer sempre più capiente e adattabile.

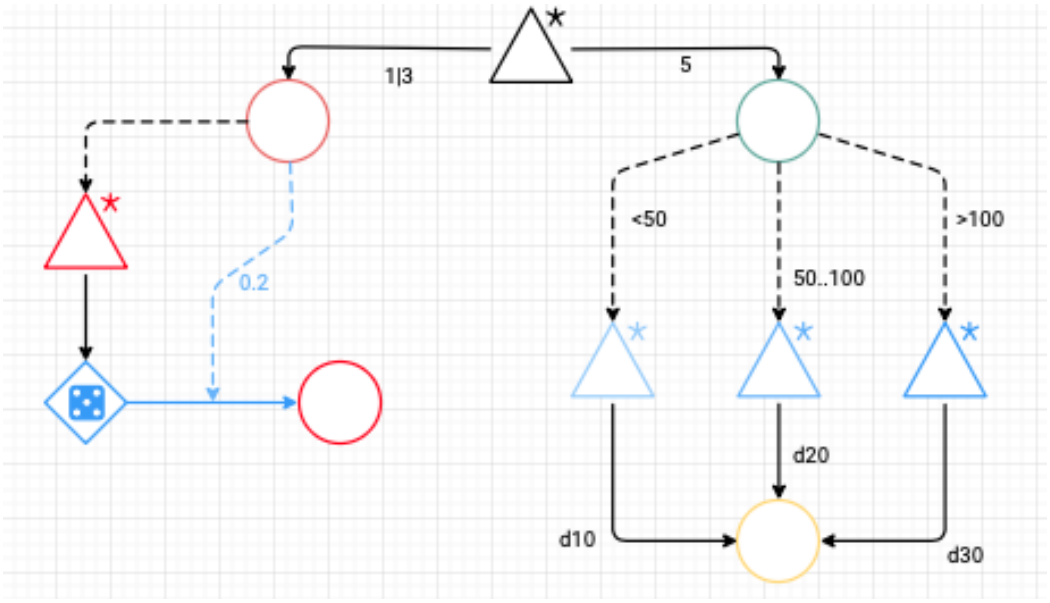


Figura 96: Esempio di schema logico e concettuale di un *endless runner* costruito con **Machinations**; l'utilizzo di questo programma consente ai designer di implementare con poco sforzo degli schemi concettuali utili per visualizzare la dinamica delle meccaniche progettate, man mano che il gioco avanza nel tempo. Questo sistema è utilissimo quando si esegue un'analisi della *distribuzione della qualità* nel gioco.

Ovviamente non possono mancare nozioni di design, visual art e psicologia cognitiva: il game designer deve fare in modo che il gioco sia bilanciato e mantenere il giocatore "incollato allo schermo", attraverso un meccanismo di **challenge** e **rewarding** ben implementato. Le conoscenze di psicologia, aiutano il game designer in questo senso: il riuscire a portare il giocatore al superamento della dualità fra mente e corpo, immergendolo in uno stato dove si perde la cognizione del tempo e delle proprie decisioni; all'uscita da questo stato, detto di *flow* (**M. Csíkszentmihályi**), che spesso corrisponde al termine della sessione di gioco, il giocatore si sentirà appagato e felice: questa che può sembrare pura elucubrazione teorica, risulta una cartina tornasole molto utile nella valutazione della qualità del gioco a cui si sta lavorando.

Figura 97: Grafico che presenta l'andamento dell'esperienza di gioco all'interno del *flow channel*; gli sviluppatori devono fare in modo di garantire la permanenza del giocatore all'interno del flow state, segnale di un'esperienza di gioco godibile ed appagante.

Tutte le skill e tools elencati sono tutte ottime armi che il game designer può utilizzare per la creazione del proprio prodotto video-ludico. Ognuna di queste è importante e difficilmente una vita sola basta per dominarle tutte alla perfezione, quindi un buon approccio è quello di non smettere mai di voler acquisire nuove nozioni ed imparare nuovi concetti.

3.5.2 Game Programmer

Al giorno d'oggi il videogioco è considerato una forma artistica, alla pari del cinema o della letteratura. In effetti molti titoli odierni hanno cambiato totalmente il modo di giocare ed intendere il videogiocare, sottraendo dinamica al gameplay e in favore della narrazione o della fruizione artistica.

Figura 98: *Gris*, uscito nel dicembre 2018 e prodotto da **Nomada Studio**, è un platform 2D con una spiccata personalità artistica volta a comunicare un profondo messaggio: la storia "muta" di Gris vuole portare il giocatore all'interno di un viaggio intimo, fatto di colori, forme e musiche che raccontano le cinque fasi del lutto, in maniera delicata ed assolutamente scorrevole per il giocatore.

Dal punto di vista del programmatore, tecnicamente parlando, il videogioco rimane un complesso software da codificare, un programma per computer con innumerevoli sfaccettature, ma pur sempre un programma per computer. Se si pensa all'inizio della storia del videogioco, erano proprio i programmatori che svolgevano la maggior parte del lavoro per realizzare un videogioco: erano singoli o piccoli team che lavoravano come pionieri per produrre e finire progetti video-ludici che sarebbero diventati piloni portanti dell'intera industria.

A differenza di un programmatore generico, un programmatore di videogiochi deve valutare ogni scelta poiché deve essere presa in funzione dell'hardware di riferimento sul quale sviluppare il gioco: questo si traduce in compromessi continui, tempi di sviluppo più lunghi e competenze specifiche sempre più richieste. Approcciarsi alla programmazione, ancora oggi scoraggia i più per la mole di lavoro che, nonostante i software di sviluppo sempre più evoluti e comodi, risulta tutt'ora titanica e di difficile comprensione. Ovvio, raramente si troveranno singoli programmatori al lavoro su un progetto, ma piuttosto un team di programmatori che si dividono porzioni di un progetto enorme: si pensi che negli ultimi anni, i progetti tripla A hanno raggiunto standard hollywoodiani, con budget e mole di lavoro impressionanti, impossibili per un numero ristretto di persone; proprio per questo, esistono differenti specializzazioni che interagiscono con discipline esterne, come la grafica, il networking o l'AI.

Figura 99: *Super Mario Bros*, sviluppato per Famicom nel 1985 e per NES nel 1986 dalla Nintendo, è stata la prima opera di **Shigeru Miyamoto** legata ad un franchise oggi leggendario; il numero di righe di codice di giochi dell'epoca ovviamente era irrisorio rispetto alla mole di codifica per i progetti odierni.

Gameplay Programmer Probabilmente si tratta della branca in cui tutti i programmatori cominciano ad esercitarsi: il *gameplay programmer* è la figura che più di tutte lavora gomito a gomito con il game designer occupandosi di fornire al computer le istruzioni che vanno a determinare le logiche e le meccaniche di gioco. Si analizzi l'esempio seguente che vuol spiegare il rapporto e che sincronia esiste tra regole di gioco e gestione dei controlli; un team vuol sviluppare un gioco spaziale legato al controllo di un'astronave.

Gestione dei controlli:

1. Leva del joystick verso l'alto -> vai al punto 10, altrimenti al passo successivo.
2. Leva del joystick verso il basso -> vai al punto 9, altrimenti al passo successivo.
3. Leva del joystick verso dx -> vai al punto 8, altrimenti al passo successivo.
4. Leva del joystick verso sx -> vai al punto 7, altrimenti al passo successivo.
5. Se viene premuto il pulsante *fuoco* -> vai al punto 11, altrimenti al passo successivo.
6. Torna al punto 1.
7. Decrementa la posizione dell'astronave sull'asse **x** -> vai al punto 5.
8. Incrementa la posizione dell'astronave sull'asse **x** -> vai al punto 4.
9. Decrementa la posizione dell'astronave sull'asse **y** -> vai al punto 3.
10. Incrementa la posizione dell'astronave sull'asse **y** -> vai al punto 2.
11. Genera un nuovo *bullet*.
12. Torna alla fase 1.

Risulta evidente che dal punto di vista della programmazione dei controlli, i comandi e le operazioni da tenere presente sono parecchie ed articolate anche per una semplice astronave che si muove sullo schermo e che spara alcuni proiettili per distruggere gli eventuali nemici. E per quanto riguarda le meccaniche?

Regole di gioco:

1. Se il nemico si scontra con un proiettile sparato dal giocatore -> vai al punto 4, altrimenti al passo successivo.
2. Se l'astronave del giocatore si scontra col nemico -> vai al punto 8, altrimenti al passo successivo.
3. Torna al punto 1.
4. Decrementa le vite del nemico.

5. Se le vite del nemico sono 0 o minori di 1 -> vai al punto 7, altrimenti vai al passo successivo.
6. Torna al punto 2.
7. Distruggi il nemico -> vai al punto 2.
8. Distruggi il giocatore -> **GAME OVER**.

Salta all'occhio la verbosità delle istruzioni ed un certo sforzo per non commettere errori di distrazione: si pensi ad un salto di fase sbagliato, che implicherebbe la nascita di un errore (bug) nella gestione dei controlli (es: joystick verso l'alto, ma l'astronave si muove a sinistra). Negli ultimi anni la programmazione è stata sempre più formalizzata, introducendo regole sia sintattiche che comportamentali: in questo modo si tenta di ridurre al minimo l'impatto umano sulla macchina. Gestione di stress, stanchezza ed interazioni intra-team, sono soggette ad un elenco di pratiche efficienti utili per raggiungere il successo; anche la banale nomina dei file deve essere fatta in maniera efficacie perché va semplificato e supportato il lavoro di tutti: il codice non viene scritto ovviamente da una sola persona, quindi tutti i programmatori devono potervi accedere. Progettare un software funzionante non basta più nel 2020: il codice deve essere leggibile e semplice secondo la logica del *Less is more*.

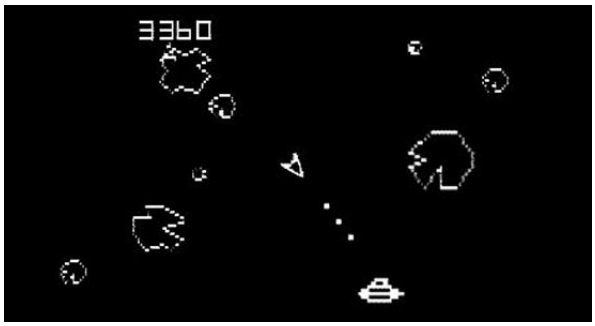


Figura 100: *Asteroids*, sviluppato e pubblicato da **Atari** nel 1979, è forse il più tipico esempio di gioco spaziale identificabile con le regole ed i controlli listati precedentemente in questo paragrafo.

FX & Graphic Programmer Una delle sfide più difficili per il programmatore di videogames è quella di dover rappresentare sullo schermo il maggior numero di dettagli nel minor tempo di caricamento possibile. Se si guarda indietro, nel periodo in cui il **NES** o il **Commodore 64** erano tra gli oggetti più desiderati dai giocatori di videogames, il programmatore aveva già qualche strumento di supporto per la programmazione grafica, ma possedevano capacità computazionali molto limitate: era l'hardware che

si adattava ai software e queste esigenze portavano verso nuovi limiti ed orizzonti. Grazie alla qualità ridotta delle immagini, legata ad un numero limitato del numero di pixel e di colori presenti a schermo, anche un hardware primitivo poteva riprodurre videogames di qualità.
Vi sono due elementi fondamentali da aggiungere a colori e risoluzione, ovvero gli **sprite** e le **tile**; va precisato che un tempo gli sprite erano reali specifiche hardware, mentre oggi vengono identificati come una qualsiasi immagine 2D all'interno di un gioco.

Scansionato con CamScanner

Figura 101: Un esempio di *sprite* utilizzato nei videogiochi: nel mercato odierno, molte software house indie, utilizzano il metodo della **pixel art** per infondere stile ed estetica ai propri titoli, oltre che risparmiare notevolmente sul comparto di modellazione.

Sia allora che oggi, gestire i singoli pixel risulta estremamente complesso: per questo motivo le immagini vengono costruite come fossero un puzzle virtuale, i cui pezzi venivano appunto definiti tile e poi assemblati per formare l'immagine complessiva. Si pensi che il NES (*Nintendo Entertainment System* gestiva una risoluzione massima di 256x240 pixel, decisamente bassa rispetto allo standard **Full HD** presente già negli anni 2000; ecco senza l'utilizzo delle tile, il NES avrebbe dovuto gestire 61.440 informazioni (risultato della moltiplicazione 256 x 240): esso infatti, gestiva tile da 8x8 pixel, arrivando a riempire una schermata pienamente grazie a 960 informazioni (32x30

pezzi del puzzle). Risulta chiaro il risparmio di operazioni/informazioni che comportava questo meccanismo: le tile erano fondamentali per gestire i fondali dei giochi, mentre per gli oggetti in movimento si utilizzavano gli sprite, disegnabili in ogni punto dello schermo a differenza delle tile. Ma non era possibile usarne più di poche decine su NES, come su Commodore 64 (si arrivò a circa 96, negli anni '90, su **Neo Geo**).

Figura 102: L'iconico *Duck Hunt* sviluppato da Nintendo per NES nel 1984: in quest'immagine è possibile vedere nettamente la divisione tra tile usate per lo sfondo e gli sprite delle anatre in movimento.

Con l'avvento della grafica 3D, il lavoro del programmatore è cambiato radicalmente: con questo salto tecnologico sono cambiati hardware e software in maniera più che significativa, ma anche le competenze richieste ai programmatori sono diverse. Si pensi che il codice scritto dai programmatori odierni non gira sul computer vero e proprio, ma piuttosto sarà eseguito sulla scheda grafica del sistema (GPU). Oltre al passaggio in 3D, un programmatore dovrà tenere conto anche della **fisica** presente all'interno del motore di gioco: la gestione delle collisioni è la base di molti giochi sia vecchi sia nuovi e con avvento del 3D, le cose si son complicate parecchio per i programmatori; solitamente l'approccio più diffuso è quello di associare una forma geometrica il più semplice possibile al *game object*: in questo modo il calcolo delle collisioni diventerà più leggero in termini computazionali. Più le forme sono complesse e più sarà dispendioso il calcolo delle collisioni: più efficace è la scelta della forma geometrica e migliore sarà l'esperienza di gioco finale. Dalla gestione delle collisioni si arriva alle vere e proprie simulazioni fisiche, ovvero alla gestione di corpi solidi soggetti a forze ed interazioni con altri corpi solidi. Servono competenze di altissimo livello per lavorare bene in questo settore della programmazione: scrivere un simulatore fisico è mol-

to complesso e per questo diverse software house si affidano a prodotti già collaudati, come il **PhysX** di **NVIDIA**.

Audio Programmer Se in passato ci si avvaleva dei sintetizzatori, dispositivi in grado di produrre forme d'onda semplici, non potendo immagazzinare forme d'onda più complesse, oggi il comparto audio è diventato fondamentale per l'esperienza video-ludica: non dovendosi più occupare anche della produzione delle soundtrack a livello creativo, l'audio programmer (o *sound programmer*) è la figura designata alla gestione del materiale prodotto dai musicisti e dai sound designer, integrandolo con il progetto di gioco.

Figura 103: In *Hellblade: Senua's Sacrifice*, prodotto da **Ninja Theory** nel 2017, il sound design ed il reparto audio in toto, sono fondamentali per l'esperienza ludica: alcune sezioni di gioco infatti vanno affrontate affidandosi solamente all'udito che ci consentirà di schivare gli attacchi nemici grazie ad input sonori.

AI Programmer Si intende il programmatore che si occupa della progettazione dei meccanismi di intelligenza artificiale all'interno del progetto: inizialmente si faceva largo uso di semplici algoritmi basati sul concetto della **FSM**, macchina a stati finiti, ovvero un eventuale entità terza all'interno del gioco potrà essere in un solo stato, in ogni singolo momento del gioco. Ad esempio, se si tratta di una guardia nemica potrà avere diversi stati disponibili come "pattuglia", "vai verso obiettivo", "ingaggia", e via dicendo. Il primo stato è appunto quello di pattuglia in cui la guardia si limiterà a cercare un obiettivo di rilievo lungo il proprio percorso; se dovesse individuarlo, la guardia passerà allo stato "vai verso obiettivo" fino a quando la distanza

non sarà sotto una certa soglia. A questo punto si hanno due opzioni: se l'obiettivo si allontana troppo andando *out of range*, la guardia tornerà sul proprio percorso di pattuglia, altrimenti passerà allo stato "ingaggia".
Questo esempio vuol evidenziare la potenziale complessità del compito del programmatore: se non venissero vagliate abbastanza possibilità situazionali e, quindi, stati da inserire, l'AI risulterebbe poco interessante e l'esperienza di gioco ne uscirebbe rovinata. Negli ultimi tempi, vengono usati metodi più sofisticati, basati sugli studi e gli sviluppi nel campo del **machine learning**, facendo in modo che sia l'AI ad imparare dalle scelte dei giocatori, aumentando il grado di sfida e la caratterizzazione dell'intelligenza artificiale.

Figura 104: L'AI dei nemici in *The Last of Us Part II*, sviluppato da **Naughty Dog** ed uscito a giugno 2020, è una delle migliori mai viste in campo video-ludico: i nemici umani si muovono in pattuglie ed accerchiano il giocatore con movimenti a tenaglia, reagendo in maniera realistica ad ogni input derivante dall'azione del giocatore. In foto, vengono evidenziati le entità amiche, quelle nemiche e gli oggetti utilizzabili per ingannare l'intelligenza artificiale.

Network & Multiplayer Programmer Un'ulteriore specializzazione è proprio quella legata al mondo delle reti: è un ramo che nasce dalla possibilità di giocare online con altre persone in tutto il mondo, grazie alla connessione Internet. Occorrono quindi figure specializzate che implementino e gestiscano il trasferimento dati, senza inficiare l'esperienza di gioco. Si pensi al fenomeno del **lag**, ovvero il ritardo del tempo che i dati del giocatore impiegano per raggiungere la destinazione e, in alcuni casi, per tornare indietro: non si può rimuovere, dato che è un limite fisico, quindi il compito dei programmatori è quello di implementare tutti i trucchi possibili per far credere al giocatore che il lag non esista proprio. In poche parole, il programmatore inganna i giocatori per garantirgli una buona esperienza di gioco.
Per fornire una spiegazione pratica di come il programmatore affronti il problema del lag, si segua tale ragionamento: se il personaggio del giocatore si sta muovendo a 5 metri al secondo, lo si farà muovere alla stessa velocità su tutte le console dei partecipanti alla partita; se il personaggio dovesse fermarsi, la simulazione sulle altre console sarebbe errata poiché il personaggio risulterebbe in movimento, quando sulla console principale non è così. I sistemi di gioco vengono di conseguenza notificati del cambiamento e "riavvolgono" la simulazione (una sorta di macchina del tempo) fino al punto in cui i valori erano corretti. Tutto avviene a velocità altissime, quindi il giocatore la maggior parte delle volte non si accorge di nulla, però *randomicamente* può capitare di vedere "saltare" i personaggi in game ad una posizione precedente: questo fenomeno viene definito **reconciliation**, in cui la simulazione era andata troppo avanti ed il software ha provveduto a rimettere in ordine la situazione.

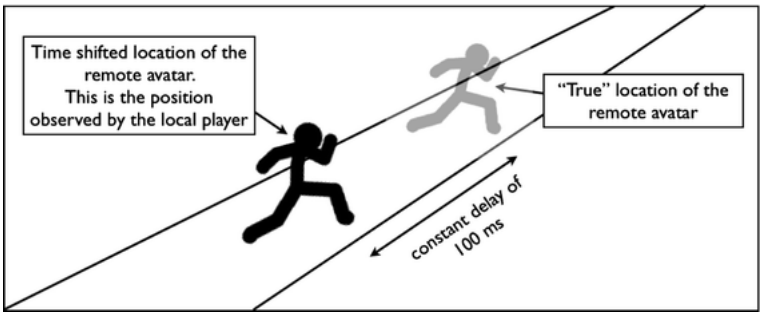


Figura 105: Una raffigurazione "estremizzata" del fenomeno del lag.

Il programmatore di reti deve inoltre occuparsi dell'integrazione e la gestione del rapporto tra i videogames ed i moderni social network; si pensi alla piattaforma **Twitch** che permette ai giocatori di trasmettere in diretta le proprie sessioni di gaming, creando un vero e proprio show dove poter interagire con il pubblico mentre si gioca. L'interazione con i servizi forniti dal web è una componente fondamentale per lo sviluppo video-ludico degli ultimi anni.

3.5.3 2D - 3D Artist

Se ancora oggi, ci si interroga sul fatto che i videogiochi siano forme d'arte o meno, sicuramente non si può negare che siano completamente permeati di arte: quasi tutte le forme artistiche conosciute concorrono alla creazione dei prodotti di intrattenimento video-ludici. La prima fase dello sviluppo di un videogioco, ovvero la pre-produzione, dove si vanno a stabilire tutti gli aspetti finali del gioco: tra questi ovviamente anche quelli visivi, utilissimi per gli autori e designer del gioco per trasmettere a tutto il team la loro visione. Proprio per questi il ruolo del **concept artist** è fondamentale, proprio per l'ideazione dell'aspetto di personaggi, environment e della loro rappresentazione visiva; possiede una formazione prettamente artistica, che è dotato di eccezionali capacità di disegno e pittura, e abile nell'uso di strumenti tradizionali o digitali per la realizzazione delle proprie opere. Molto spesso capita che questi professionisti lavorino anche in altri settori: non è raro che fumettisti di altissimo livello vengano ingaggiati per la produzione di videogames.

Figura 106: Esempio di concept art utilizzata per lo sviluppo di *Final Fantasy VII Remake*, sviluppato da **Square Enix** e pubblicato nel 2020; lo stile ed il design dei personaggi è a cura dell'itaiano **Roberto Ferrari**, esempio virtuoso per l'Italia nel mondo dei videogames.

Nella fase produttiva vera e propria le opere del concept artist arrivano nelle mani del 2D/3D artist (per semplicità definito "grafico" d'ora in poi) che creerà tramite appositi software gli asset del videogioco, rispettando con la massima fedeltà le opere del concept artist. Il grafico però non deve solo realizzare opere artisticamente valide, ma che siano anche funzionanti e funzionali.

Character artist Si tratta forse della specializzazione più ambita ed affascinante: è il grafico che si occupa di realizzare i personaggi del videogioco: protagonisti, creature, nemici, boss. Questo professionista si avvale di strumenti software estremamente versatili, ma anche di conoscenze artistiche ed anatomiche esaustive, che gli consentono di sviluppare forme anatomiche anche in forme non realistiche. Estremamente importanti risultano le skill di disegno e quelle riguardanti la gestione/applicazione dei colori, che benché guidate dai riferimenti del concept artist, devono essere supportate dalle nozioni teoriche sopracitate, tanto quanto la pratica nel disegno.
Risulta importante che il grafico conosca un serie di accorgimenti tecnici da adottare affinché il modello poligonale possa subire in maniera non distruttiva le eventuali deformazioni derivanti dall'animazione: ogni modello infatti viene realizzato con l'idea che venga successivamente animato da altri professionisti.

Figura 107: Esempio di studio per Aloy, la protagonista di *Horizon Zero Dawn*(2017) prodotto da **Guerrilla Games** per PS4.

Character Rigger Figura specializzata che ha il compito di creare lo "scheletro" dei personaggi presenti in gioco, in modo da renderli animabili; questa sezione di sviluppo, solitamente appannaggio di grafici ed animatori, è diventata di una complessità tale da rendere necessaria la presenza di una figura professionale dedicata.
Un personaggio costruito da poligoni di varia natura ha bisogno di un sistema scheletrico che gli consenta di muovere arti ed assumere determinate espressioni facciali, quando necessario. Quando si usa il termine **rigging** si sta indicando il processo di creazione dello scheletro ed altri strumenti utili affinché un personaggio, una creatura o una qualsivoglia entità possano avere animazioni in game, dalle semplici rotazioni o traslazioni alle più complesse presenti nei titoli ultimi dell'industria moderna; il rigger inserisce scheletro e "strumenti" ausiliari nel modello, associando ossa ed articolazioni con i vertici dei poligoni: questa operazione è definita come **skinning** o **binding** e permette la deformazione coerente del modello, che asseconderà le anima-

zioni delle articolazioni inserite. Il rigger deve implementare anche i controlli che consentono all'animatore di agire sul personaggio: servono conoscenze tecniche sia legate ai software da utilizzare sia in termini di programmazione, in modo da avere un risultato efficace.

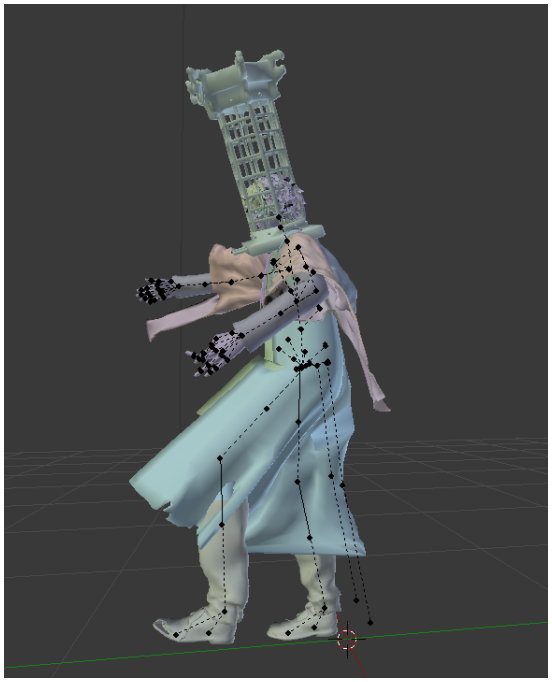


Figura 108: Fase di rigging di un character di *Bloodborne*, sviluppato da **From Software** nel 2015.

Animator Non è difficile intuire in cosa consista la professione dell'animatore: si occupa delle animazioni del character in questione, dalla *idle* position alla corsa, dalla posizione di guardia a quella di attacco, il salto, le schivate e così via. Si può affermare che storicamente l'animatore altro non è che un artista, un illustratore che realizza una serie di *frames* che, se riprodotte in sequenza ad una certa velocità, danno l'illusione del movimento, cosa ancora valida per videogames e cartoni 2D; nel mondo 3D, invece assume una valenza più tecnica soprattutto per l'utilizzo di specifici software e soprattutto perché non servono abilità pittoriche o di disegno. Il flusso di lavoro prevede che l'animatore crei delle animazioni sufficientemente piacevoli, coinvolgenti e coerenti a livello visivo, non solo anatomicamente, ma anche con il personaggio su cui si sta lavorando: un personaggio deve muoversi e "recitare" coerentemente con la sua natura e backstory, non deve esserci dissonanza (se non voluta). Le tecnologie di questo settore stanno avanzando e migliorando sempre più, richiedendo sempre maggior competenza a livello tecnico: si pensi alle tecniche di *motion capture* o di quelle *macchine a stati*

finiti che consentono di legare animazioni tra loro. Tutta questa mole di contenuto dovrà essere ripulita e gestita, in modo che le animazioni create anche con le tecniche più avanzate risultino armoniche e coerenti con l'ambiente circostante.

Figura 109: Sessione di motion capture realizzata per *Beyond Two Souls* della **Quantic Dreams**, uscito nel 2013: in questo caso sono stati assunti attori professionisti come Ellen Page e Willem Defoe, proprio un virtù delle abilità recitative ed espressive da regalare ai character del gioco.

Environment Artist Una delle parti più vaste della produzione videoludica riguarda la creazione di **ambienti** di gioco, solitamente piuttosto vasti e ricchi di dettagli, fondamentali per la loro caratterizzazione. L'environment artist è un grafico specializzato nella creazione degli asset (naturali o architettonici, reali o fantastici) che andranno a popolare l'ambiente: dalle rocce alle case, dagli alberi alle strutture tecnologiche o irrealistiche.

Figura 110: Esempio di savana costruita in maniera realistica per lo sviluppo di *Uncharted 4: A Thief's End*, sviluppato da **Naughty Dog** nel 2016: risulta evidente come i materiali e gli oggetti assumano una connotazione realistica, la luce e l'illuminazione sono state create tenendo conto delle leggi fisico-ottiche realistiche, vista anche la natura del gioco che vuole essere sì avventuroso, ma non irrealistico.

Gli strumenti di lavoro non cambiano rispetto a quelli presentati precedentemente, ma diventa essenziale la capacità di osservazione, di saper analizzare i dettagli dell'oggetto nella realtà e saperli riprodurre digitalmente nella maniera più fedele possibile. Oltre alla mera modellazione, ciò che rende credibile un oggetto è il lavoro compiuto sulle **texture** e sugli **shader**, ovvero sui materiali di cui sono composte le superfici e alla loro resa visiva. Ad esempio, se il gioco avrà uno stile più cartoon, verrà preferita la capacità di sintesi, mentre con uno stile grafico realistico o iperrealistico, serve conoscere perfettamente come si comportano i materiali e le superfici in base all'illuminazione, sia nel mondo reale sia nel motore di *rendering* usato. Esistono anche parecchie sotto-specializzazioni, come quella del grafico che si occupa della vegetazione o della formazione del terreno.

Figura 111: Esempio di environment con uno stile più cartoon derivante da *The Legend of Zelda: Breath of the Wild*, titolo **Nintendo** uscito nel 2017 per Nintendo Switch e Wii U; è evidente la differenza stilistica con l'immagine precedente: si punta molto meno al realismo e i dettagli sono meno marcati, ma la caratterizzazione artistica è comunque ben sviluppata.

Prop Artist Si tratta di un modellatore che si occupa della creazione di oggetti di scena di natura variabile, che concorrono all'arredamento scenico, sia ambientale che per la creazione di interni: attrezzi, detriti, armi, veicoli e simili, rientrano in una categoria di modelli definiti come **prop**. Il numero di prop per i progetti video-ludici è talmente alto che esistono alcune aziende che forniscono la produzione di prop in **outsourcing**, e non è raro trovare diverse sotto-specializzazioni, come quella del **weapon artist** o del **vehicle artist**; il prop artist è la variante meno complessa a livello tecnico, poiché i vincoli sono relativamente pochi e di semplice gestione, ma diventa fondamentale la velocità di esecuzione data la mole di oggetti da produrre.

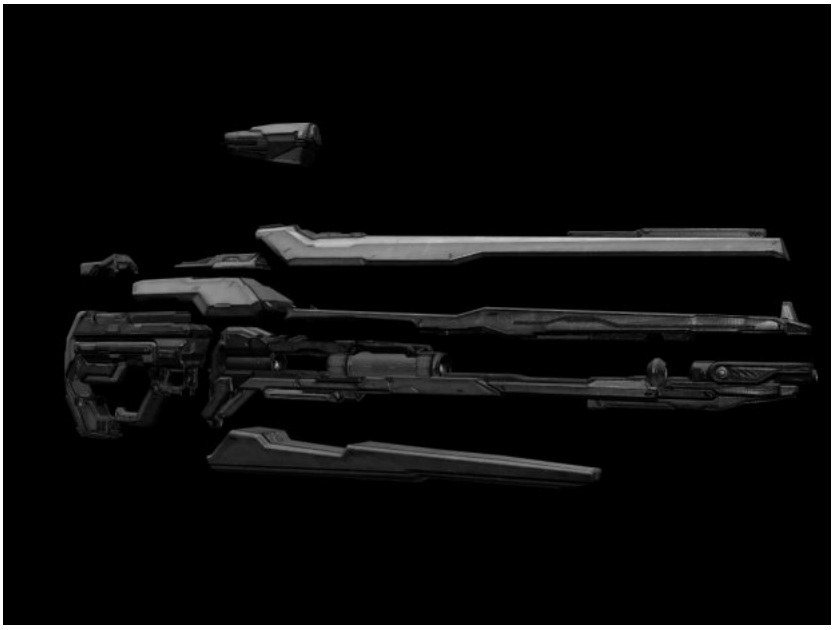


Figura 112: Modello di un fucile da cecchino per *Halo 4*, sviluppato da **343 Industries** e pubblicato da **Microsoft** nel 2012; è un esempio di come la qualità dei prop debba essere di altissimo livello, soprattutto per produzioni ad alto budget.

UI Artist Ogni dispositivo elettronico dotato di display necessita di un'interfaccia utente che consenta la navigazione all'interno del software: che si tratti del display di un ATM o del cruscotto di un automobile, il concetto della **User Interface** è essenziale. Si intende l'insieme degli indicatori visuali (icone grafiche, finestre, pulsanti, messaggi) che consentono all'utente di interagire con il dispositivo o il software.

Figura 113: Esempio di UI per videogames particolarmente ispirata: sviluppato da **EA Redwood Shores**, *Dead Space* è il primo titolo della serie uscito nel 2008 e presenta una UI solidissima ancora oggi; l'interfaccia utente viene rappresentata tramite la proiezione di un ologramma senza mettere in pausa il gioco, integrando visivamente le informazioni senza spezzare il flow si gioco (la salute ad esempio è rappresentata dalla linea spezzata al neon presente sulla tuta del protagonista).

Nel mondo del gaming, va sottolineata la coesistenza tra UI ed **HUD**, che apparentemente risultano identici come concetti, ma ci sono sottili differenze: si può affermare che l'HUD, *Head-Up Display* o *Status Bar*, faccia parte della UI, ma sia precisamente il metodo con il quale le informazioni vengono trasmesse visivamente al giocatore come parte dell'interfaccia utente di un gioco.
Risulta quindi evidente di cosa si occupi lo User Interface artist: è un grafico specializzato nella creazione di interfacce utente per videogiochi, dalla navigazione all'interno dell'interfaccia stessa ai vari menù, dalla barra della "salute" al sistema di selezione delle armi del personaggio. Si preoccupa della prototipazione e del testing dell'usabilità dell'interfaccia, della realizzazione dell'aspetto grafico e della sua implementazione all'interno del game engine. Queste interfacce possono essere molto invasive come nel caso dei MMORPG alla *World of Warcraft* (2004, **Blizzard Entertainment**) oppure minimali come nel caso di *Dead Space* (2008, **EA Redwood Shores**), horror sci-fi che ha fatto la storia per quanto riguarda la UI nel gaming. I software utilizzati sono gli stessi che normalmente vengono adottati in altri campi grafici (Photoshop, Illustrator, Flash e After Effects), ma esistono anche software dedicati, come Scaleform. Le competenze necessarie all'UI artist si possono acquisire anche tramite corsi non necessariamente inerenti ai videogiochi,

come quelli di graphic, industrial e web design.

Figura 114: Nella storia del gaming anche la UI e l'HUD si sono evoluti, a volte presentandosi nelle forme più estreme: una di queste è sicuramente quella proposta da *Shadow Of The Colossus*, sviluppato dal **Team Ico** e **Japan Studio** con la supervisione del designer **Fumito Ueda** nel 2005; Ueda propone un design sottrattivo, dove l'HUD è ridotto al minimo per favorire il senso di esplorazione ed immersione nel giocatore.

VFX Artist Figura molto richiesta nel settore video-ludico, si occupa della realizzazione degli "effetti speciali", che contribuiscono in maniera decisiva per rendere accattivante il videogioco; sebbene i suoi strumenti di lavoro siano relativamente pochi, il VFX artist deve avere ottime capacità sia artistiche (di ideazione, realizzazione grafica ed animazione) sia tecniche, dal momento che il risultato del suo lavoro ha un forte impatto sulle prestazioni del videogioco. L'acronimo VFX comprende i più disparati effetti speciali: esplosioni, fiamme, scintille, foglie al vento, incantesimi, portali magici, spari d'armi da fuoco, simulazioni fluide e così via. La principale differenza con la grafica 2D e 3D trattate precedentemente consiste nel fatto che il processo di creazione dei VFX non è lineare: ogni asset ha problematiche specifiche, e dunque richiede un diverso approccio per essere realizzato in modo ottimale. Quando si creano i VFX, oltre a puntare alla bellezza degli effetti ed ottimizzarli per l'hardware su cui verranno utilizzati, occorre tener conto della coerenza comunicativa e narrativa: devono comunicare una serie di informazioni sull'effetto pratico dell'azione che li scatena (area di effetto, tipo di

effetto, interazione o non interazione). Per concludere, si può affermare che le competenze del VFX artist siano davvero sfaccettate, cioè spaziano dall'arte dell'animazione, dalla conoscenza degli *shader* alle tecniche di simulazione della fisica, fino a toccare aspetti psicologici.

Figura 115: Come in tanti altri giochi, *League of Legends*(2009) di **Riot Games**, la cura dei VFX è fondamentale: essendo un MMORPG, l'utilizzo dei poteri è parte fondamentale del gameplay, quindi gli effetti visivi devono essere molto chiari nei confronti dei giocatori che istantaneamente devono comprenderne gli effetti e gli utilizzi.

Lighting Artist Figura professionale legata prettamente alla produzione di asset grafici: il lighting artist è quel professionista che ha l'importante compito di gestire l'illuminazione della scena, quindi del livello di gioco, non limitandosi a posizionare semplicemente le sorgenti luminose, ma consiste anche nel raccontare una storia, suggerire un mood, proporre un'atmosfera e supportare il gameplay, fornendo al giocatore dei punti di riferimento che lo guidino. Se artisticamente è assimilabile ad un direttore della fotografia, il lighting artist tecnicamente opera all'interno del motore di gioco, senza grande necessità di usare strumenti terzi: grazie agli strumenti che possiede deve poter apportare modifiche ai materiali (*shader*) affinché le superfici reagiscano coerentemente e correttamente all'illuminazione.
Luci e ombre hanno un fortissimo impatto sulle performance del videogioco e sulle necessità di calcolo; quindi come per il VFX artist, la conoscenza tecni-

ca e la capacità di problem solving del lighting artist risultano determinanti per ottimizzare gli ambienti in game in modo da ben illuminarli e renderli ben immersivi, senza pesare troppo sull'hardware.

Figura 116: Fondamentale per chi si occupa di illuminazione, è lavorare in maniera efficace sull'atmosfera che si vuol trasmettere; ad esempio, in *Control*, prodotto da **Remedy Entertainment** e pubblicato da **505 Games** nel 2019, presenta un'illuminazione particolarmente ricercata per favorire un'atmosfera pericolosa e misteriosa, invitando il giocatore ad avanzare con cautela e sviscerare i misteri che il gioco gli pone davanti.

Technical Artist Figura professionale relativamente recente, imputabile all'avanzamento della tecnologia, si colloca in una zona limbo non facilmente classificabile: è una zona in cui la capacità artistica, nella sua accezione base, lascia spazio alla parte tecnica che lo avvicinano alla figura del programmer. Il technical artist interviene in diversi ambiti: **shading**, **lighting**, **pipeline**, **script**, creazione procedurale, ottimizzazione,..., lo si può definire un programmatore con forti doti artistiche, o viceversa un grafico con forti nozioni tecniche o legate all'ambito della programmazione. Grazie a questa sfera eclettica di competenza, il technical artist tendenzialmente fa da ufficiale di collegamento tra il team dei grafici e dei level designer e quella dei programmatori, avendo la capacità di interagire con entrambi i settori. La sua funzione è quella di condividere in maniera efficace una grande quantità di informazione tra i reparti così da aiutarli al migliorare alcune *pipeline* produttive ed alcuni aspetti visivi, soprattutto nelle fasi di prototipazione e

nell'ottimizzazione. Se la prototipazione è l'insieme delle operazioni volte a testare nuovi metodi di creazione e a suggerire nuove funzionalità o caratteristiche per i sistemi già adottati, le quali, una volta approvate, verranno implementate dal reparto di programmazione, l'*ottimizzazione* rappresenta una serie di operazioni che hanno lo scopo sia di rendere più snello, rapido ed economico il flusso di lavoro, sia di poter distribuire il videogioco su diverse piattaforme, ognuna con caratteristiche tecniche specifiche. Ad esempio, per quanto riguarda la prototipazione, il technical artist deve dare supporto agli artisti mediante la creazione di strumenti utili ad automatizzare o rendere più veloci alcune procedure: spesso le conoscenze del technical artist si applicano nella creazione di *tools* per i software di grafica. Per quanto riguarda l'ottimizzazione, si tratta di fare determinate scelte intelligenti e veloci per implementare variazioni al motore di gioco per personalizzare e rendere fluido il titolo in sviluppo; si pensi alla riduzione del calcolo degli shader o della complessità delle scene, all'ottimizzazione delle allocazione in memoria o alla regolazione delle impostazioni della qualità grafica, tutto per raggiungere il miglior risultato possibile su ogni piattaforma.

3.5.4 Publisher

Un publisher è colui che pubblica videogiochi che sono stati sviluppati internamente dall'editore o esternamente da uno sviluppatore di videogiochi. Spesso finanziano lo sviluppo, a volte pagando uno sviluppatore di videogiochi esterno o talvolta pagando uno staff interno di sviluppatori. I grandi editori di videogiochi distribuiscono anche i giochi che pubblicano, mentre alcuni publisher più piccoli assumono invece società di distribuzione (o publisher più grandi) per distribuire i giochi che pubblicano. Altre funzioni normalmente svolte dall'editore includono l'acquisto di eventuali **licenze** utilizzate dal gioco o l'investimento per la **localizzazione** multi-lingua, impaginazione, stampa, eventuale scrittura del manuale utente e la creazione di elementi di design grafico come il *packaging*. Alcuni grandi editori con struttura verticale possiedono anche filiali editoriali, chiamate **etichette**.

Figura 117: **505 Games** è stata eletta da Metacritic come il miglior publisher del 2020: la società italiana di giochi, che pubblica titoli di terze parti per ogni piattaforma, ha avuto un ottimo 2019 grazie a un flusso costante di giochi recensiti positivamente e pochi fallimenti. Tra le sue produzioni ricordiamo *Control* (**Remedy**, l'omaggio al franchise *Castlevania* di **ArtPlay** con lo sviluppo di *Bloodstained: Ritual of the Night* ed il porting per PC di *Death Stranding*(**Kojima Producitons**).

I grandi editori possono anche tentare di aumentare l'efficienza di tutti i team di sviluppo interni ed esterni fornendo servizi come il comparto per il sound design ed i pacchetti di codice per le funzionalità comunemente necessarie.
Poiché l'editore spesso finanzia lo sviluppo, cerca di gestire il rischio di sviluppo con uno staff di produttori o project manager per monitorare i progressi dello sviluppatore, quindi monitorare l'andamento del proprio investimento: la maggior parte dei videogiochi creati da uno sviluppatore esterno viene pagata con anticipi periodici sulle **royalties**. Questi anticipi vengono pagati quando lo sviluppatore raggiunge determinate fasi di sviluppo, chiamate **milestone**.

Figura 118: Esempio di slide usata durante la presentazione al publisher di *Pride Run* (**IVProduction**):le milestone vanno esplicitate in maniera plausibile in modo che il publisher possa farvi affidamento e concordare il piano di marketing con il developer; inoltre possono essere fornite più soluzioni o opzioni di investimento, in modo che il publisher abbia possibiltà di scelta.

Va esplorato fortemente anche il rapporto che c'è fra lo sviluppatore ed il publisher; solitamente il primo titolo viene *autopubblicato* dal developer, ma supponendo di doversi interfacciare con un publisher, un buon esempio di interfaccia publisher-developer è la fiera dedicata al publishing, ovvero la **Game Connection**: i developer pagano una cospicua cifra in denaro (si aggira attorno ai 3.000 €) per ottenere un tavolo a questa fiera, tavolo che serve per incontrare i publisher presenti all'evento. I publisher fissano appuntamenti con i team presenti e li incontrano; quando si siedono sanno già chi hanno di fronte e che gioco gli verrà proposto: lo scopo finale di questi "meet-table" è quello di convincere i publisher ad investire sul proprio gioco. Si ha una mezz'ora di tempo ad incontro: in una giornata si fanno circa venti incontri ed il **deck** (documento di presentazione del gioco e della proposta economica), fatto dal team di sviluppo, deve essere convincente.

Figura 119: Esempio di slide usata durante la presentazione al publisher di *Pride Run* (**IVProduction**): immagini e notazioni esaustive ma sintetiche, rendono chiaramente identificabile il titolo.

Il developer non deve farsi cogliere impreparato, considerando anche gli interessi del publisher oltre ai propri. Servono un profilo aziendale, i giochi portati in fiera e le idee chiare su che cosa si sta cercando: publishing e funding? Distribuzione? Publishing ma senza funding? Ogni situazione è un caso particolare, ma generalmente la cosa più importante è la chiarezza sul lavoro economico che si intende intraprendere, indicando precisamente costi e guadagni possibili.

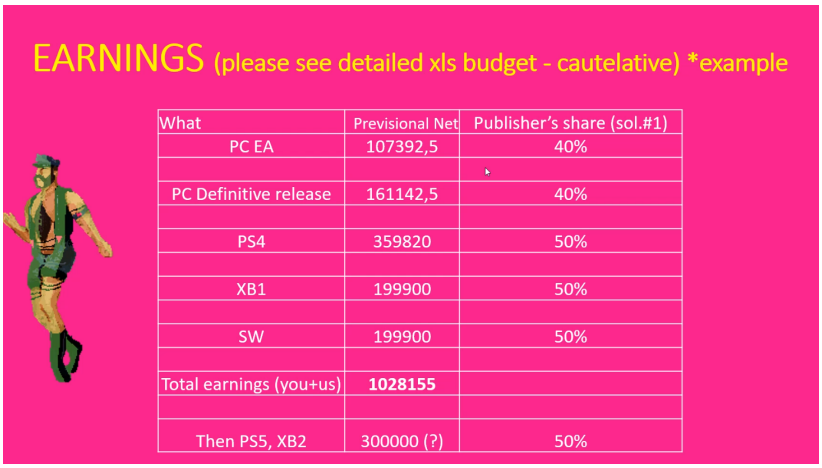


Figura 120: Esempio di slide usata durante la presentazione al publisher di *Pride Run* (**IVProduction**):qui si può notare una stima ben articolata delle possibili vendite e guadagni conseguenti.

3.6 Design vs. Mercato

Uno degli aspetti più sottovalutati dagli sviluppatori alle prime armi è l'ambito economico, legato a doppio filo al mercato di acquisto: è proprio in queste mancate scelte che solitamente avvengono i primi fatali errori. Il mercato e l'aspetto economico sono importanti demiurghi per il design di un videogame, al pari delle scelte autoriali dell'art director.

All'inizio dello sviluppo di un gioco, molto spesso capita di perdersi nell'entusiasmo del momento: l'idea di mettersi alla prova su un proprio titolo spesso porta gli sviluppatori in erba, ma non solo, a compiere errori di valutazione lasciandosi trasportare dall'enfasi del momento; occorre fare attenzione perché le cose rischiano di complicarsi in brevissimo tempo, quindi meglio non esagerare nell'ampliare il progetto. Mai partire a testa bassa senza un buon lavoro di pre-produzione e senza sperimentare le meccaniche pensate, perché inserire feature su feature nel gameplay rischia di portare in una fase analoga all'*uncanny valley*, ovvero il gioco appare molto buono inizialmente, mentre quando poi si inizia a giocare per un po' di tempo, si notano sempre più dettagli e particolari che ci fanno capire che il gioco non funziona, non è così bello come inizialmente appare. Le cose solitamente non funzionano mai bene al primo colpo, bisogna cercare di mantenere editabile il materiale che si produce man mano, in modo da poter intervenire senza problemi in ogni momento del progetto: serve ordine e logica. La prototipazione va fatta per verificare e testare le meccaniche su carta, così facendo si può già imbastire il piano di produzione a seconda del feedback che ci è arrivato dal test.
Prototipare necessita di **feedback** che consentano di percepire cosa funziona realmente e cosa no, e non solo nel gioco in sé, ma anche nel gioco inserito nel mercato di riferimento; esistono due possibilità di avere feedback continui durante la fase di prototipazione:

- **Prototipo -> grezzo ma funzionante -> efficiente capacità autocritica -> loop.**
  Questo metodo consente al developer di lavorare autonomamente al test delle meccaniche, ma richiede una capacità di astrarsi da sé stessi molto elevata: è il developer stesso ad essere il critico del suo gioco e se non ha una cultura video-ludica esaustiva, si rischia di avere un feedback non valido o poco funzionale, quindi di danneggiare il progetto.

- **Prototipo -> rifinirlo e renderlo bello visivamente -> assistere a sessioni di gioco di giocatori esterni al team -> loop.**
  I feedback in questo caso arrivano direttamente da possibili futuri consumatori, ovvero i giocatori: tutto quello che arriverà agli sviluppatori saranno feedback basati sul comparto visivo e quanto impatta sul gameplay, quanto può essere fuorviante o intuitivo; è un ottimo metodo di procedere.

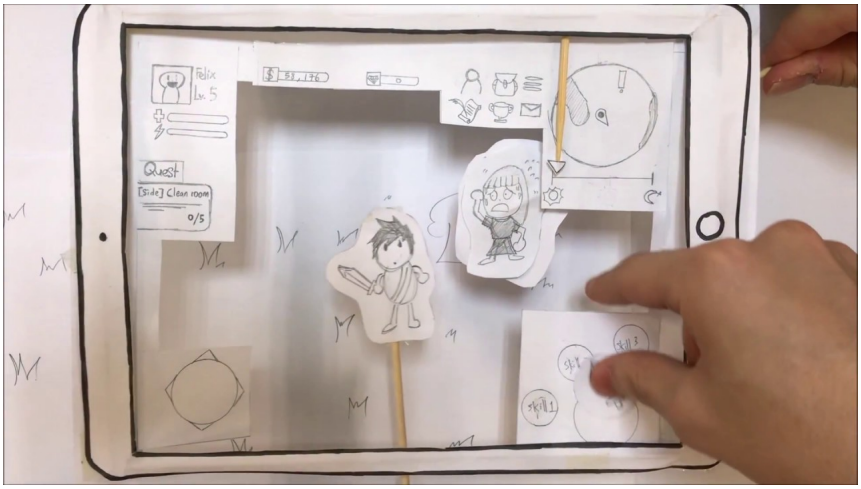


Figura 121: Il test delle meccaniche, o prototipazione, può avvenire su carta o grazie all'utilizzo di altri oggetti, come mattoncini LEGO, soldatini e così via: risulta ottimale per rendersi conto quanto ancora il gioco sia acerbo o difettoso.

Lo sviluppo del gioco può richiedere molto tempo e a volte non avere una scadenza rischia di far morire il progetto; quindi può essere una soluzione rilasciarlo in **early access** per sfruttare una possibile *onda verde* del mercato. Ma attenzione: anche in questo caso possono esserci dei rischi! Ad esempio, un gioco *story driven* rischia di avere una storia incompleta che può smontare il progetto facendo perdere la validità della storia stessa, o può aumentare l'*hype*, imponendo molto lavoro sul team a fronte di un'eventuale richiesta sempre più pressante di terminare il gioco.

Figura 122: La metodologia dell'*early access* ha dato anche buoni frutti: lavorando sul dialogo tra azienda e pubblico è possibile produrre eccellenze come *Darkest Dungeon*, titolo targato **Red Hook Studios** che grazie al supporto di Steam ha potuto godere di uno sviluppo serio e ben fatto tramite il sopracitato meccanismo.

Solitamente è consigliabile optare per un periodo di **betatesting**. Un'ulteriore variabile da tenere in considerazione è la possibilità di fare **crowdfunding** ed ovviamente anche questa modalità può avere le problematiche di fallimento o successo viste sopra. I soldi quindi risultano sempre "Il Problema", quindi per i primi progetti da lanciare sul mercato o per i progetti indipendenti in generale, spesso bisogna lasciare perdere le versioni fisiche: costano troppo e difficilmente si intercetta un buon numero di acquirenti.

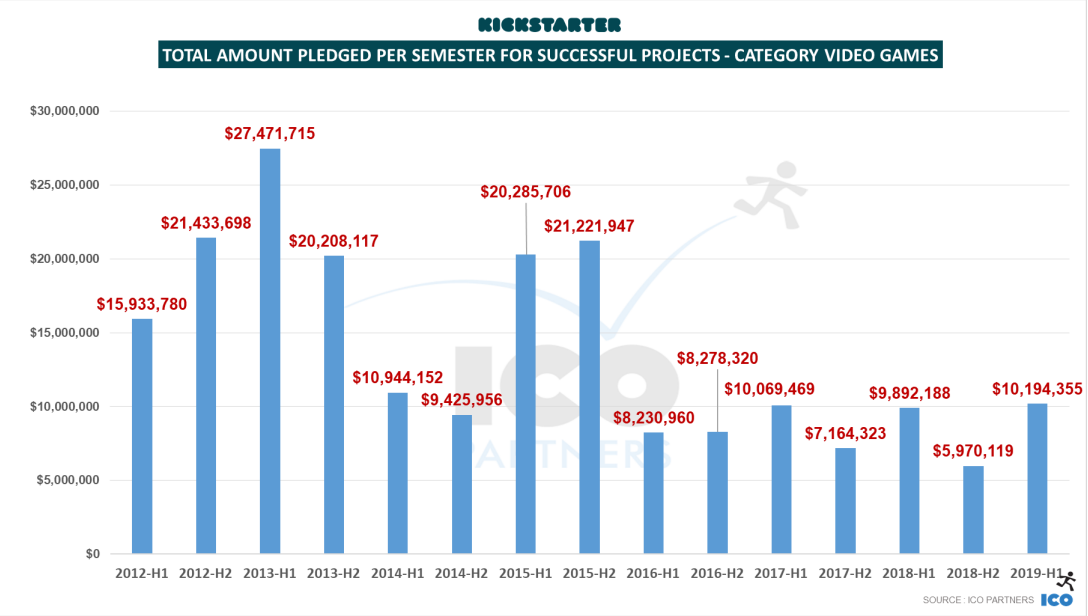


Figura 123: In figura sono riportati i cumulativi annuali in dollari per progetti video-ludici finanziati su **Kickstarter** forniti da **ICO**, una società esperta nel game publishing.

Ricapitolando quando si inizia un progetto di sviluppo è obbligatorio chiedersi quanta disponibilità di tempo e denaro si ha per lavorare ad un videogioco: valutare la propria determinazione, il capitale di partenza e quali tecnologie impiegare, sono step necessari per iniziare in maniera efficiente un progetto del genere. Se non si dispongono risposte affermative o positive a tali interrogativi, meglio abbandonare il progetto. Al contrario, un secondo interrogativo risponde al problema economico già citato in questo paragrafo: quale tipo di business si sceglie? Risulta un punto chiave: si può vendere il proprio tempo per lavori o collaborazioni a terzi, in modo da autofinanziare il progetto e "sopravvivere" mentre lo si completa, oppure si può optare per un modello **B2B** (*business to business*), cioè creare un videogioco su commissione, collaborando con un cliente (ad esempio, la software house **Melazeta** ha sviluppato un gioco per il comune di Bologna).

Figura 124: Uno screenshot da *WunderBO*, sviluppato da **Melazeta**, per Android e IOS, in collaborazione con il comune di Bologna; WunderBo è un videogioco gratuito che porta alla scoperta dei poli museali del capoluogo emiliano: i giocatori sono invitati a trovare opere d'arte e reperti d'epoca provenienti dalle collezioni del Museo Medievale e di Palazzo Poggi. L'obiettivo è collezionare tale serie di oggetti preziosi, conservandoli all'interno di una personalissima Wunderkammer digitale.

Un altro modello è quello **B2C** (*Business To Client*): lo sviluppatore produce autonomamente il proprio gioco e lo vende mantenendone la proprietà intellettuale (**IP**). È un modello opposto a quello del B2B: spesso si tentano degli ibridi tra questi due modelli, ma difficilmente coesistono. É importante scegliere il tipo di business model poiché trovare un publisher al primo gioco che si sviluppa è praticamente impossibile: il publisher ricerca esperienza e competenza, quindi è naturale che un team al suo primo progetto non ne abbia; esistono altri modi per trovare fondi: bandi, *business angel*, imprenditori di ventura e via dicendo.

Figura 125: Un esempio del modello B2C: *Mario + Rabbids: Kingdom Battle* è un videogioco sviluppato dalla divisione italiana **Ubisoft Milan** che utilizza la IP di Nintendo.

Una volta che si sono stabilite le modalità di finanziamento e guadagno, occorre pensare per quale piattaforma pubblicare, a che pubblico arrivare, al genere e alla tipologia, alle *key features,* ai *selling points*: il numero di decisioni e scelte da effettuare è effettivamente alto; operazioni del genere sono processi che richiedono molto lavoro e possono durare molto tempo, ma vanno effettuate all'inizio del progetto: analizzare competitor, scegliere l'engine giusto per il gioco e la piattaforma di riferimento (la prima su quale pubblicarlo) sono tutte cose di fondamentale importanza.
Ad esempio, se il team, nello specifico il game designer, conosce bene ed ama il genere del gioco al quale lavora, può avere un riscontro interno sul lavoro svolto, altrimenti bisogna informarsi e capire **cosa cercano** gli utenti in un determinato genere. Bisogna interrogarsi sul **prezzo**: vi sono numerosissimi parametri da considerare, come ad esempio la qualità grafica, che incide fortemente sulla scelta del giocatore (più sarà alta, più sarà disposto a spendere).

Prendendo come esempio *VR PingPong* e *VR PingPongPro*, sviluppati da IVProductions e pubblicati in anni differenti (2016 e 2019), si nota come siano essenzialmente lo stesso gioco, ma la qualità grafica incide a tal punto da metterli su fasce di mercato completamente diverse. *VR Ping Pong* ha una grafica più low-poly, con uno stile più cartoon ed assolutamente non realistico: la tecnologia VR è sfruttata a dovere e propone una fisica ultra responsiva, ma il comparto grafico non è sicuramente la key feature del progetto.

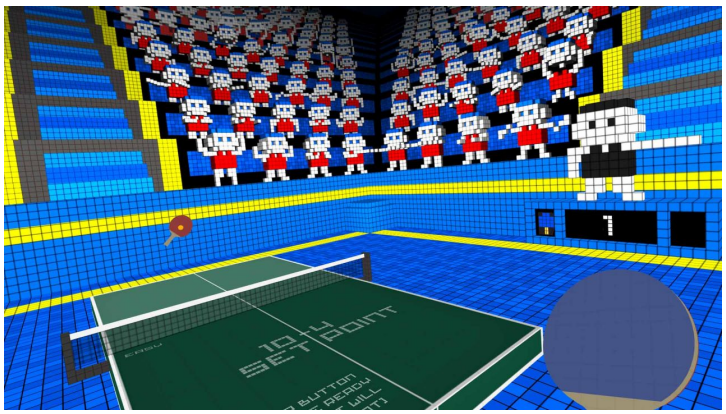


Figura 126: Uno screenshot da *VR Ping Pong*: fu il primo gioco di ping pong VR ad uscire su PS4.

VR Ping Pong Pro è il seguito di *VR Ping Pong* dove è stato fatto un grosso lavoro in ambito visivo: si ricerca il foto-realismo, sia di gioco, sia a livello di ambienti di gioco; risulta ovvio che il prodotto abbia richiesto più lavoro del precedente.

Figura 127: Risulta evidente come *VR Ping Pong Pro* abbia un comparto grafico di altro livello: ambienti 3D differenti, shader e texture di alto livello.

Il primo viene venduto a 9,99 €, mentre il secondo a 24,90 € su Steam: questo è un esempio di quanto una scelta tecnico-artistica possa incidere sui possibili guadagni e scelte di lancio; per arrivare a tali risultati bisogna fare **budget previsionali** corretti e precisi: consente di svolgere il lavoro in maniera serena e senza imprevisti economici (raramente fila tutto liscio).
Una volta chiariti il "cosa", il "come", il "dove", il "chi" ed il "quando" posso partire con lo sviluppo del game design, della relativa documentazione (**GDD dinamico**) e della stesura di un piano di produzione, di un piano commerciale (ip, corporate, ...) e di un prototipo. Tutta questa fase rientra nella già citata pre-produzione: da questa non si "esce" finché le cose non funzionano alla perfezione; l'utente, infatti, non prova affetto per il gioco in sviluppo, non ha empatia, quindi la comunicazione commerciale va gestita in maniera saggia. Nel caso di piccoli progetti, meglio non far parlare molto prima del gioco, ma pubblicizzarlo a ridosso del lancio.

3.6.1 Le Piattaforme e la Distribuzione

Nell'ultimo paragrafo sono stati definiti i possibili step e suggerimenti per ragionare sul design tenendo ben presente l'impatto sul mercato ed il torna conto economico; a proposito di quest'ultimo, è lecito chiedersi: "Dove si comprano i videogiochi?", "Dove vengono giocati? Console, PC o Smartphone?", "Quali sono le abitudini di gioco dei giocatori?". In fase di design, le risposte a tali domande possono influire moltissimo sul futuro del mio progetto: conferiscono un direzione e dei limiti utili a delimitare il progetto.
Ma tornando alla distribuzione dei videogiochi: in Italia, a differenza di stati come Inghilterra o USA, ci sono pochissimi publisher, anche se il mercato è florido in termini di acquirenti.
Proprio per questo motivo esistono interessanti zone *grigie* tra la distribuzione ed il settore publishing: molto spesso vi sono società intermedie che fanno sia da publisher che da distributore. Se i canali di distribuzione sono quelle piattaforme digitali o fisiche dove poter comprare i videogiochi, come i comportano si publisher nei loro confronti? Letteralmente sono quelli che *mettono* i giochi nei canali di distribuzione. Ne esistono di vari tipi e con particolari caratteristiche: ogni publisher ha dei focus su determinati distributori e relativi tipi di utenti.
Ad esempio, **Annapurna Interactive** pubblica giochi super ricercati, con una narrativa ed uno stile molto particolare, oltre ad essere molto differenti tra loro.

Figura 128: **Annapurna Interactive** propone giochi con una forte componente narrativa e dotati di stili particolarissimi e diversi fra loro: uno di questi è *Kentucky Route Zero* è un'avventura punta e clicca su un'autostrada segreta che attraversa le caverne sotto il Kentucky e le persone misteriose che la percorrono; salta all'occhio il particolare stile grafico che cattura immediatamente l'attenzione del giocatore.

Altri come **Astragon**, un publisher tedesco, ha tutto un altro target: infatti si occupa della pubblicazione di simulatori, in particolare di *Bus Simulator* ed ha il focus solo su quel tipo di giochi.

Figura 129: **Astragon** propone giochi basati sulla *simulazione*, ovvero la replicazione millimetrica di un'attività umana o meno: tra questi sicuramente si può menzionare *Bus Simulator 2020*, che sfida il giocatore nell'operare esattamente come farebbe un autista di bus; sono giochi che hanno saputo ritagliarsi una community molto attiva, tanto che è prevista l'edizione *Bus Simulator 21*, in uscita nel 2021.

Alla luce di tali considerazioni, la proposta ad un publisher o ad un distributore di un videogioco va ponderata: un certo tipo di videogame sarà compatibile con un determinato publisher piuttosto che con un altro. Facendo un esempio: non è efficace proporre un gioco basato sulla narrativa fantasy ad un publisher che ha sempre lavorato con simulatori di volo.
Sarebbe un contro senso economico-societario per il publisher in questione, oltre che un fallimento per i developer. Quindi anche in questa fase, il lavoro degli sviluppatori in fase di design deve tenere conto di queste meccaniche commerciali, delle preferenze dell'utenza e del genere di gioco da proporre: grazie a questi "reminder" è possibile districarsi nel mondo della distribuzione.

Figura 130: In figura sono rappresentate alcune tra le più importanti piattforme di distribuzioni digtali presenti sul mercato: **Epic Games Store**, **UPlay**, **Steam**, **GOG.com**, **Origin** e **Battle.net** di **Blizzard Entertainment**.

In Italia è raro occuparsi solamente di game design e trovare un mercato florido di publisher; le aziende che fanno videogiochi sono poche ed un terzo di esse fatturano meno di 150.000 € annui, evidenziando una realtà ancora acerba. Il game designer o chi lavora in questo campo, deve conoscere questo sottobosco fatto di competenze tecniche ed anche manageriali: è necessaria una mentalità imprenditoriale.

Figura 131: Nonostante lo stra potere delle piattaforme digitali, esistono ancora, anche in Italia, negozi specializzati, non appartenenti a catene (vedi Gamestop)internazionali, gestite da appassionati in funzione dei giocatori: spesso è possibile trovarvi pezzi da collezione, edizioni speciali o giochi ormai fuori mercato, tornei organizzati ed accessori difficilmente reperibili (in figura, **Game Over - Video & Games** di Ostellato, Ferrara).

3.7 Genere e Tipologie

La definizione di videogioco è cambiata parecchio nel corso del tempo, a seconda della sua natura spesso variabile: se alle sue origini, il videogioco era definito come un dispositivo elettronico in grado di simulare quasi esclusivamente attività sportive, era dovuto al fatto che i primi successi commerciali erano per lo più videogiochi sportivi.

Figura 132: La console *Magnavox Odissey*, ad esempio, aveva un parco giochi di 27 titoli, 10 dei quali erano sportivi.

Per lunghi periodi, infatti, il videogioco era caratterizzato da pochi generi, ben definiti e molto riconoscibili: sportivi, automobilistici (variante degli sportivi), spara-tutto, casinò (simulazione dei giochi d'azzardo), di ruolo, strategici, rompicapo ed educativi. Questi vengono definiti come *proto-generi*, utili alla tassonomia contemporanea, ma ad oggi talmente ibridati (ad eccezione degli sportivi/automobilistici) da mettere in discussione l'applicabilità stessa del concetto di genere ad un medium complesso come quello del videogame. Malgrado questo, la classificazione in generi è fondamentale per il pubblico, la critica e l'intero mercato di acquisto e sviluppo: esiste una tendenza da parte di tutti questi attori a sviluppare e creare una tassonomia sempre nuova e strutturata per identificare i giochi, tra generi e sottogeneri.

3.7.1 Action

Si tratta di quei videogames nei quali la coordinazione occhio-mano rappresenta il core dell'esperienza di gioco: questa componente è presente in una buona parte dei videogiochi odierni, quindi molti titoli ricadono all'interno di questa macro area; se si considera la grande volatilità dei generi video-ludici, uno standard dei giochi d'azione è costituito dalla presenza di un personaggio controllabile dal giocatore che ha l'abilità di interagire con l'ambiente di gioco, la cui struttura è solitamente organizzata in livelli, più o meno riconoscibili. La sfida offerta è solitamente legata ad ostacoli o a nemici (standard o boss di livello); il personaggio controllato dal giocatore è vulnerabile, quindi può subire danni che possono compromettere i suoi punti vita, provocandone il **game over**; se il giocatore riesce a superare tutti gli ostacoli, i nemici e quindi a completare il livello, riceverà un **reward**, traducibile in *game score*

o in una ricompensa utile per l'avanzamento del gioco.
Se queste sono le caratteristiche standard tipiche dell'era *arcade*, ad oggi la situazione è molto cambiata: elementi come il punteggio o la presenza delle vite sono obsoleti e sempre più rari, come la struttura a livelli che gradualmente ha perso terreno in favore dell'*open world*, in cui il giocatore ha a disposizione un vasto mondo di gioco in cui muoversi con un certo grado di libertà. Quest'ultimo passaggio è dovuto soprattutto ad una evoluzione del linguaggio video-ludico, ma anche alle innovazioni tecnologiche: i primi videogiochi non avevano una durata elevata proprio per la limitata capacità delle risorse utilizzate, quindi un giocatore abile poteva completare il gioco in un tempo molto ridotto; oltre a questo motivo tecnico, vi era anche una motivazione economica: trattandosi di macchine *coin-op* si volevano favorire sessioni di gioco corte e un ricambio di giocatori frequente, in modo da ottenere più monete in meno tempo. Proprio per questa ragione la meccanica delle vite e del punteggio erano così apprezzate, ma con l'avvento delle console casalinghe le cose sono cambiate anche all'interno degli stessi franchise: in *Super Mario*, primo gioco della serie, il giocatore aveva a disposizione un certo numero di vite, uno score da totalizzare giocando e un certo numero di livelli, mentre nella sua ultima iterazione, *Super Mario Odissey*, il giocatore si ritrova 15 location *open world* la cui esplorazione può richiedere ore ed un sistema di gioco che non fa uso di vite.

Figura 133: *Super Mario Odissey*, sviluppato da **Nintendo** per Nintendo Switch nel 2017, presente un gameplay totalmente diverso dallo standard action dell'era arcade di cui faceva parte il primo titolo della saga, ovvero *Super Mario*.

Platform Allacciandosi proprio al franchise di *Super Mario*, uno dei capostipiti di questo sottogenere, si può definire il *platform* quel genere di giochi in cui il personaggio controllato deve saltare di piattaforma in piattaforma per superare i vari ostacoli e raggiungere un determinato obiettivo. Ad aiutare il giocatore in questo compito, intervengono spesso i **power up**, ossia strumenti che consentono al personaggio di diventare più forte, di ottenere un potere speciale o un'arma. Il primo gioco *platform* può essere identificato in *Donkey Kong*, diretto da **Shigeru Miyamoto**: l'avatar del protagonista deve evitare dei barili saltando verticalmente di piattaforma in piattaforma per raggiungere la principessa alla sommità del livello e salvarla dalle grinfie di un gorilla gigante, il tutto in un'unica schermata.
Da quella prima espressione, il genere ha cambiato veste molte volte, diventando uno dei generi più apprezzati degli anni Ottanta e la seconda metà degli Novanta; nei primi 2000 entra in una profonda crisi, risollevandosi soltanto nel 2007 con *Super Mario Galaxy* ed un successivo revival dei giochi in 2D. Ad oggi nel mercato mobile, il *platform* ha trovato una sua dimensione ben specifica grazie agli *endless runner*: il personaggio controllato si muove indipendentemente, mentre il giocatore è chiamato a toccare lo schermo per farlo saltare o cambiare direzione della corsa.

Figura 134: Primo lavoro di **Jonathan Blow**, *Braid* è uno dei platform più apprezzati e riusciti degli ultimi anni: racconta la storia di un uomo alla ricerca della sua amata in un mondo in cui le leggi del tempo possono essere piegate a seconda della necessità. Rewind, rallentamenti e meccaniche simili rappresentano il cuore dell'esperienza, assieme alle lunghe e talora ardue sessioni puzzle del gioco; Musica e art style fanno da padroni, dando all'esperienza un incantevole piglio fiabesco che è impossibile dimenticare anche a distanza di anni.

Picchiaduro Rappresenta il genere in cui la coordinazione occhio-mano viene spinta ai massimi livelli, il *picchiaduro* porta il giocatore ad affrontare uno o più nemici, controllati dal computer o altri giocatori, facendo uso delle arti marziali o di poteri speciali; il personaggio, come l'avversario, ha a disposizione una barra dell'energia che si riduce ad ogni colpo subito ed una volta svuotata completamente, il personaggio perderà l'incontro. Il genere si suddivide in due grandi aree: incontro singolo o struttura a scorrimento.
In quest'ultimo, il giocatore sfida più nemici controllati contemporaneamente dall'intelligenza artificiale in un ambiente aperto a scorrimento laterale o in 3D. Spesso può raccogliere dei *power-up* con durata limitata utili per sconfiggere i nemici, suddivisi in comuni, potenziati o boss: questi hanno spesso comportamenti prevedibili, detti **pattern**, che il giocatore deve apprendere per anticiparne le mosse e affrontarli nel miglior modo possibile. Questa tipologia di *picchiaduro* è apparsa per la prima volta nel 1984 con *Kung-Fu Master* (**Takashi Nishiyama**) sui cabinati arcade e su NES; nonostante un rapido declino a partire dagli anni 90, ebbero un successo clamoroso per poi essere sostituiti dai più moderni *fighting game*, ovvero i *picchiaduro* a scontro singolo.

Figura 135: Nato da una branca del *picchiaduro* a scorrimento, il genere *musou* è amatissimo in Giappone: si tratta un gioco in cui l'eroe controllato dal giocatore deve affrontare orde di nemici da solo combattendo all'arma bianca fino alla comparsa del boss di livello, la cui sconfitta determinerà il superamento del livello. In figura, *Warriors Orochi 3* di **Koei Tecmo Games** sviluppato nel 2011.

Il *picchiaduro* standard è basato su una struttura di gioco a scontri singoli, nel quale il giocatore affronta, di solito, un nemico per volta, in un'area ben delimitata e in una serie di round: in pratica, le meccaniche di gioco ricalcano pedissequamente le regole proposte da un incontro sportivo di arti marziali, in cui l'abilità del giocatore e la sua capacità di affrontare la sfida vengono messe in primo piano.
Con l'arrivo di *Street Fighter II*, sviluppato da **Capcom** nel 1991, il genere *picchiaduro* introduce una meccanica basata sull'esecuzione di mosse concatenate, dette *combo*: queste, al fine di essere realizzate, richiedono al giocatore una grandissima abilità, mettendole al centro del gameplay. Giochi come *Street Fighter 2* uniti alle possibilità fornite dal multiplayer, hanno dato il via ad una serie di tornei competitivi e leghe che hanno spianato la strada alla nascita del professionismo tra i giocatori di videogiochi, e sono tutt'oggi al centro di alcune fra le più importanti competizioni mondiali.
Nonostante l'avvento del 3D, il genere *picchiaduro* rimane uno dei baluardi della bidimensionalità, anche ai giorni nostri.

Figura 136: Delle molteplici iterazioni della serie *Street Fighter*, *Street Fighter II* rimane tutt'ora il più amato della serie: ancora oggi, schiere di giocatori si riversano online per assistere o partecipare a tornei del gioco.

Con l'avvento del 3D, il genere ha cominciato ad ibridarsi e a mutare, generando importanti varianti e combinazioni interessanti di gameplay: è opportuno citare il sottogenere **hack 'n' slash**, una forma di *picchiaduro* in cui si fa uso di armi bianche.

Sparatutto É il più longevo dei sottogeneri *action*, risalente ai tempi di *Spacewar!* (**Steve Russell**, 1962) considerato uno dei primi videgiochi di sempre, nello *sparatutto* o *shooter* i nemici si affrontano con le armi da fuoco; è un genere vastissimo, nel quale rientrano diverse tipologie di gioco che variano a seconda del punto di vista del giocatore o, nel caso di grafica 2D, del tipo di scorrimento.
I primi esempi di successo di questo genere risalgono al termine degli anni '70, con *Space Invaders* e *Asteoids* (**Atari**, 1979): entrambi i giochi si svolgono all'interno di un'unica schermata, il primo si basa sull'utilizzo di una torretta semovente che spara proiettili verticalmente contro una fila di nemici, mentre il secondo permette di muovere una navicella e sparare contro gli asteroidi fluttuanti. Queste sono le prime iterazioni del genere shooter a tema spaziale e negli anni titoli come *Xevious* (**Namco**, 1982) e *Scramble* (**Konami**, 1981), hanno apportato sempre più modifiche e varianti al genere, introducendo lo scorrimento verticale o laterale.

Figura 137: *Spacewar!* è un videogioco *sparatutto* per computer PDP-1, realizzato principalmente da **Steve Russell**tra il 1961 e il 1962. *Spacewar!* è stato il primo a presentare un mondo dotato di regole fisiche, con situazioni variabili e completamente in tempo reale; è stato dichiarato uno dei dieci videogiochi più importanti di sempre da Henry Lowood della Stanford University, nel marzo del 2007.

Meccaniche simili vennero implementate in giochi in cui il protagonista è un essere umano: *Metal Slug* (**SNK**, 1996) e *Contra* (**Konami**, 1987) furono capostipiti di shooter ibridi, capaci di miscelare le meccaniche classiche degli *sparatutto* a quelle dei *platform*; vengono ricordati come i migliori interpreti del genere dell'era pre-3D e rappresentano un sottogenere indipendente definito **shoot 'em up** o in alcuni casi, quando la "pioggia" di proiettili diventa molto intensa, si identificano come **bullet hell**.

Figura 138: Ricordato ancora oggi per la sua difficoltà, *Contra* è uno degli *shoot 'em up* più amati di sempre e non è un caso che vi sia ancora un vasto mercato basato sul retrogaming che lo pone tra le prime posizioni dei titoli più richiesti.

Con l'avvento dell'era 3D, gli *shooter* cambiano radicalmente: si passa alla visuale in prima persona (**FPS** o *first person shooter*), grazie allo sviluppo della telecamera virtuale (gioco in soggettiva), creando un nuovo filone tassonomico tanto solido, quanto variegato. Se i primi FPS, come *Doom* o *Quake*, avevano una trama scarna e puramente di cornice al gameplay, non si può dire lo stesso delle nuove reincarnazioni del genere, che proponevano componenti narrative sempre più importanti e profonde.

Figura 139: *Half Life*, sviluppato da **Valve** nel 1998, rappresenta un punto di svolta per il gaming moderno: l'introduzione di una trama fantascentifica profonda ed avvincente, unita ad un gameplay innovativo che miscela componenti *sparatutto* e *puzzle*, lo rendono ancora oggi uno degli FPS più importanti del medium.

La diffusione delle reti Internet ha permesso al genere di evolvere nuovamente, portando la componente multi-player all'interno del gioco, facendo passare in secondo piano la componente narrativa in favore di quella competitiva: titoli come *Call Of Duty* (**Activision Blizzard**, 2003) o il più recente *Overwatch* (**Activision Blizzard**, 2016) sono il terreno di scontro fra team di professionisti in tornei ufficiali.

Figura 140: Gli *sparatutto* in prima persona sono il genere più di successo nel mercato contemporaneo dei videogiochi e franchise come *Call Of Duty*, hanno superato la quota di 250 milioni di copie vendute; la sua prima iterazione rappresenta la prima volta in cui uno *shooter* presenta una componente multi-player.

Esistono anche altri tipi di *shooter* da analizzare: basta una variazione nella soggettiva di gioco e si passa dagli FPS ai **TPS** (*third person shooter*), giochi in cui la componente narrativa ha ancora un grosso peso, mentre quella competitiva è meno sviluppata, anche se alcuni esempi recenti (*Splatoon*, **Nintendo**, 2015) hanno portato svolte curiose nel genere; nei TPS la telecamera è spostata in terza persona, ovvero dietro le spalle del protagonista, generando possibilità di gameplay totalmente differenti.
Sicuramente da sottolineare altri due sottogeneri, ovvero i **tactical shooter**, *sparatutto* in cui è possibile pianificare l'azione attraverso una mappa di gioco per poi eseguirla in tempo reale sul campo di battaglia (es. *Tom Clancy's Rainbow Six* di **Red Storm Entertainment** uscito nel 1998), e gli *sparatutto* su **binari**, in cui il giocatore non ha libertà di movimento se non nell'uso del mirino e del "grilletto" (es. la serie *Star Fox* sviluppata per le console Nintendo).

Stealth Se il gioco spinge il giocatore ad evitare gli scontri con i nemici in favore di un approccio silenzioso, concentrando il gameplay sul nascondersi e sul rimanere nell'ombra, si parla di *stealth* game. Nasce nel 1979 con *Manbiki Shounen*, il genere *stealth* incontra il grande pubblico con la serie *Metal Gear*, creata nel 1987 da **Hideo Kojima** all'epoca ancora un giovane sviluppatore; nel gioco si interpreta un soldato scelto per operazioni di infiltrazione oltre le linee nemiche: questo personaggio incarna l'archeti-

po di tutti i giochi *stealth*. I giochi di questo tipo offrono al giocatore la possibilità di utilizzare l'ambiente a proprio vantaggio, come ad esempio l'utilizzo dell'erba alta per mimetizzarsi o sfruttare l'illuminazione ambientale per muoversi tra le zone d'ombra.

Figura 141: *Metal Gear Solid*, sviluppato da **Konami** nel 1998, è considerato uno dei massimi capolavori del medium: con un gameplay rivoluzionario, una narrativa fantapolitica degna di un film hollywoodiano e un comparto visivo strabiliante per l'epoca, fu un successo senza precedenti, consacrando per sempre il suo creatore Hideo Kojima, che distanza di qualche anno avrebbe lanciato sul mercato la seconda iterazione del franchise, ovvero *Metal Gear Solid 2: Sons of Liberty*, che la reso una delle sette persone più influenti al mondo di quegli anni.

Giochi di questo tipo necessitano spesso di una buona intelligenza artificiale, capace di mettersi sulle tracce del giocatore se questi risulta troppo rumoroso o poco attento.

Survival Questo particolare genere pone il giocatore in un ambiente ostile con lo scopo principale di sopravvivere tramite un'ampia libertà d'azione. Solitamente all'inizio della partita si è privi di risorse e occorre iniziare a recuperarne il più possibile: acqua, cibo ed oggetti utili alla sopravvivenza si accumulano nell'inventario; tali oggetti possono essere combinati tra loro attraverso la meccanica di *crafting* per ottenerne di nuovi e di migliore qualità.
Meccaniche di questo tipo sono alla base di *Minecraft*, sviluppato da **Marcus Persson** (aka **Notch**) e **Mojang** nel 2011, pietra miliare dell'ultima decade videoludica: *Minecraft*, inizialmente distribuito solamente via Internet, diventa un successo planetario grazie alla diffusione online supportata dalla community e dai forum nati attorno al gioco, tanto che nel 2014 viene acquistato da **Microsoft** per la cifra di 2,5 miliardi di dollari. Il gioco diventa un fenomeno di massa, oltre ad essere il gioco più acquistato della storia superando l'iconico *Tetris*, tanto da catalizzare la creazione di tantissimi altri giochi indipendenti dopo di lui; oltre alla modalità single player, i survival gamer cercano maggior sfida online scontrandosi con altri giocatori: *PlayerUnknown's Battlegrounds* (**PUBG Corporation**, 2017) e *Fortnite* (**Epic Games**, 2017) sono giochi molto simili, ma con target differenti, che danno la possibilità a 100 giocatori di sfidarsi in una mappa aperta in continuo restringimento, dove vince l'ultimo giocatore che resta in vita; questo filone di giochi viene identificato come **battle royale**.

Figura 142: Giocando sul fattore *thalassophobia*, *Subnautica*, sviluppato da **Unknown Worlds Entertainment** nel 2014, è un survival horror game di pregevole fattura: il giocatore sarà chiamato ad esplorare i vasti mari e le profondità oscure di 4546B, un pianeta sul quale si precipita ad inizio gioco; il compito non sarà per nulla semplice dato che le acque sono infestate di mostri marini e le difficoltà legate alla sopravvivenza sono all'ordine del giorno.

3.7.2 Adventure

Nei videogiochi di *avventura*, la narrativa è il fulcro dell'esperienza di gioco, senza intaccare l'importanza dell'interattività basata sull'esplorazione, sulla risoluzione di enigmi o dalla presenza di dialoghi a scelta multipla; tuttavia, in moltissimi titoli appartenenti a questo genere, la componente interattiva è blanda o totalmente assente. Un'altra caratteristica tipica di questi giochi è la presenza di finali multipli, determinati dalle scelte prese durante il corso della partita: indipendentemente che la telecamera virtuale sia posizionata in prima o in terza persona, i giochi di avventura risultano ad oggi tra i più amati del mercato.

Avventura Testuale É un sottogenere che risale agli albori del videogioco ed è la forma primordiale di ogni gioco di avventura; il primo videogame a definirsi tale è *Colossal Cave Adventure*, sviluppato da **William Crowther** e **Don Woods** nel 1976 per computer mainframe DEC PDP-10 in cui il giocatore risponde con la tastiera a delle linee di testo che compaiono sullo schermo. Questa tipologia di gioco é caratterizzata solamente

da linee testuali senza immagini e rappresenta una versione informatica del libro-game, una forma narrativa a bivi popolarissima negli anni Settanta e Ottanta. Le avventure testuali trovano terreno fertile su personal computer poiché richiedono la tastiera per poter essere giocate. L'evoluzione della grafica nei videogiochi ha sostanzialmente determinato la scomparsa delle avventure testuali, che si sono evolute nelle avventure grafiche e nelle **visual novel**.

Avventura Grafica La forma più popolare e diffusa dei giochi di avventura ricade sotto il termine di *avventura grafica*: questo genere pone il giocatore all'interno di ambienti interattivi con i quali è possibile interagire, solitamente tramite la risoluzione di rompicapi o enigmi. Solitamente questa interattività è ottenuta tramite hotspot presenti in gioco: toccando uno di questi il giocatore potrà avanzare nella storia; tale tipologia di gioco è definita **punta e clicca**, molto famosi negli anni '80 e '90: la casa di produzione **LucasArts** ha realizzato alcuni tra i migliori giochi del genere, tra cui *Maniac Mansion* (1987), *Il Segreto di Monkey Island* (1990) e *Grim Fandango* (1998). Solitamente questi giochi venivano giocati su PC proprio per la presenza del mouse, mentre su console il sistema di attivazione degli hotspot è basato sulla vicinanza del giocatore ad essi, chiamandolo a premere un pulsante invece che portarlo alla frenetica ricerca di punti di interesse.
Molte *avventure grafiche* moderne hanno investito maggiormente sugli aspetti narrativi, sui dialoghi o su brevi intermezzi d'azione definiti **quick time events**, in cui il giocatore deve premere tasti a tempo; tra questi si possono citare *The Walking Dead* (**Telltale Games**, 2012), *Life Is Strange* (**Dontnod Entertainment**, 2015) o *Until Down* (**Supermassive Games**, 2015).

Figura 143: *The Secret Of Monkey Island*, primo titolo del franchise sviluppato da **LucasArts** nel 1990, pone il giocatore nei panni di Guybrush Threepwood, un giovane uomo che sogna di diventare un pirata; tra gli sviluppatori del titolo troviamo uno dei guru del game design moderno, ovvero **Tim Schafer**, fondatore di **Double Fine Productions**.

Walking Simulator Sono giochi in cui la sfida è totalmente assente visto che l'interattività ambientale è annullata in favore dell'esplorazione; l'avventura si trasforma in una lunga "camminata" in un ambiente virtuale con forte connotazione narrativa, infatti più si esplora e più la storia avanza. Sono giochi creati, solitamente, da studi indipendenti: *Firewatch* (**Campo Santo**, 2016), *The Stanley Parable* (**Galactic Cafe**, 2013) o *The Vanishing Of Ethan Carter* (**The Astronauts**, 2014), apprezzati da critica e pubblico, sono titoli che hanno dimostrato quanto un genere apparentemente "noioso" può invece essere amato ed essere apprezzato anche da chi non è avvezzo al video-giocare.

Figura 144: In *The Vanishing Of Ethan Carter* il giocatore impersona l'investigatore del paranormale Paul Prospero che riceve una lettera d'aiuto dal dodicenne Ethan Carter, invitandolo ad indagare sui misteri della Red Creek Valley.

Puzzle Adventure L'elemento principale dell'avventura è dato dalla risoluzione di rompicapi; giochi di questo tipo sono caratterizzati dalla presenza di enigmi interconnessi tra loro da sequenze in cui si esplora o da intermezzi narrativi con dialoghi a scelta multipla.
Nato sui Macintosh con *Myst* (**Cyan Inc.**, 1993), il genere ha sempre goduto di una buona schiera di appassionati. Prodotti di successo come *The Talos Principle* (**Croteam**, 2014) e *The Witness* (**Jonathan Blow**, 2016) dimostrano quanto il genere sia ancora vivo e abbia creato una nicchia di giocatori affezionati.

Visual Novel É un genere poco diffuso in Occidente, questo genere è straordinariamente apprezzato in Giappone, dove nasce agli inizi degli anni Novanta: è un'avventura testuale a bivi, con lunghissimi dialoghi, accompagnata da immagini statiche o leggermente animate disegnate in stile manga ed un'interattività davvero ridotta. Nonostante la nicchia di mercato sia prevalentemente locata in Oriente, alcuni successi come *Ace Attorney* (**Capcom**, 2001) per la console portatile Nintendo DS sono arrivati sul mercato occidentale formando una schiera di giocatori appassionati; per completezza di informazione, va citato il sottogenere **dating simulator**, ovvero giochi basati sul dialogo al fine di conquistare un partner, talvolta con epiloghi erotici.

Figura 145: *Doki Doki Literature Club!*, sviluppato dal **Team Salvato** nel 2017, è incentrato sulla storia di un ragazzo che entra nel club di letteratura del proprio liceo e interagisce con le quattro ragazze che lo compongono. La trama della storia è piuttosto lineare, con la possibilità però di sbloccare delle scene aggiuntive e un finale alternativo a seconda delle scelte fatte durante il gioco. Sebbene sulle prime appaia un semplice simulatore di appuntamenti, si tratta in realtà di una visual novel di carattere horror psicologico, con frequenti scene in cui si rompe la quarta parete.

Film Interattivo Si definisce *film interattivo* quando il gioco è basato su sequenze cinematiche non lineari; ha la sua origine nei FMV elettromeccanici come *Wild Gunman* (**Nintendo**, 1984), ovvero all'epoca dei *laserdisc* (1978). Il sistema di controllo di questi giochi è basato su dialoghi a scelta multipla o *quick-time-event*: a seconda dell'azione del giocatore si avrà una determinata scena animata.
Forse il massimo esponente del genere è *Dragon's Lair* (**Cinematronics**, 1983), gioco esclusivamente basato sui *quick-time-event* in cui un cavaliere affronta una serie di pericoli per salvare una principessa. Il genere è decaduto nella seconda metà degli anni Novanta proprio per l'avvento della grafica 3D.

3.7.3 Action-Adventure

Primo genere ibrido comparso nel mondo dei videogames, i giochi di questo genere sono così diffusi e così codificati da poter essere classificati tassonomicamente come macrogenere indipendente. Tipicamente negli *Action-Adventure* si ritrovano le componenti tipiche degli *action* e dei giochi di *avventura*: si basa su meccaniche di esplorazione e combattimento, e dunque alterna sequenze concitate a momenti più rilassati e di respiro ampio: spesso

fa uso di mappa *open world* o di livelli molto grandi, esplorabili liberamente. Esempi di questo genere sono la serie *The Legend Of Zelda* (**Nintendo**, 1985) o la serie *Tomb Rider* (**Eidos Interactive**, 1996).

Survival Horror Dagli *action-adventure* è nato un intero filone di giochi definiti *survival horror*, comparsi negli anni Ottanta e divenuti popolarissimi con *Alone in the Dark* (**Infogrames**, 1992) e *Resident Evil* (**Capcom**, 1996), in cui il giocatore ha a disposizione una quantità di risorse limitata e deve sopravvivere all'assalto di mostri o assassini, spesso usando l'ambiente come strumento di difesa.
Per questa tipologia di gioco è necessaria la componente narrativa (ovviamente horror), tanto che sono sfruttate le tecniche tipiche del cinema "hitchcockiano" come la suspense o il *jump scare*, il tutto per inquietare o spaventare il giocatore.

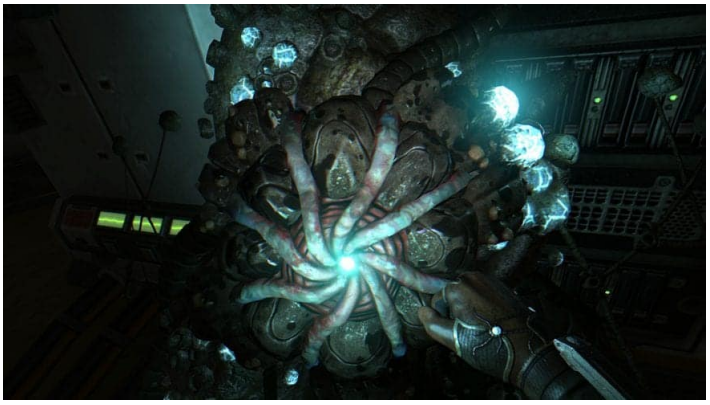


Figura 146: *SOMA*, gioco di **Frictional Games**, studio specializzato nella produzione di *survival horror*, è un titolo terrificante che porta il giocatore in uno stato continuo di ansia durante l'esplorazione di una stazione di ricerca; il gioco mette in primo piano la relazione tra corpo e mente agendo più come horror psicologico che come horror di stampo più classico.

Metroidvania Si deve il nome alla crasi di di due titoli importantissimi del medium, risalenti al 1986, ovvero *Metroid* (**Nintendo**) e *Castlevania* (**Konami**); sono giochi caratterizzati dalla grafica 2D, in cui il giocatore si muove in una mappa di gioco labirintica affrontando numerosi nemici e, talvolta, saltando di piattaforma in piattaforma in uno stile che ricorda il genere *platform*.
L'esplorazione dell'ambiente è obbligatoria, poiché per esplorare determinate aree di gioco servono particolari oggetti, detti *perks*, che conferiscono potenziamenti al protagonista (es. doppio salto, arma che distrugge determinati muri invalicabili altrimenti,...) consentendogli di avanzare nell'esplorazione di ambienti prima inaccessibili. Spesso questi oggetti si trovano all'estremità opposta del livello e, proprio per questo motivo, nei *metroidvania* è frequente attraversare aree di gioco già visitate in precedenza, in sessioni di gioco definite di *backtracking*.

3.7.4 Sportivo

Il videogioco *sportivo* è uno dei *proto-generi* del medium, poiché esistono titoli come *Tennis For Two* (**William Higinbotham, Robert Dvorak**, 1958) che sono esistiti molto prima della nascita dell'industria video-ludica; ad oggi il videogioco sportivo include franchise di altissimo livello e presentano licenze multimilionarie ottenute dalle più importanti leghe sportive mondiali.
I videogiochi sportivi si dividono in *simulativi* o *arcade*: i primi seguono in maniera più fedele possibile alla realtà, cercando contemporaneamente di simulare un modello di leggi fisiche simile a quello reale, mentre i secondi possono rappresentare sport esistenti o inventarne di nuovi, modificando le normali leggi fisiche che governano l'universo in favore di una maggiore spettacolarità. Appartenenti alla prima tipologia di videogame sportivi, sono titoli come *FIFA* (**EA Sports**, franchise nato nel 1993), *NBA 2K* (**Sega Sports**,**2K**, franchise nato nel 1999) o *Nagano Winter Olympics '98* (**Konami**, 1998), mentre sono titoli *sportivi arcade Windjammers* (**Data East**, 1994), dove i giocatori si sfidavano a battaglie di frisbee, o **Rocket League** (**Corey Davis**, 2015), un gioco che miscela il demolition derby con le regole del calcio.

Figura 147: *Pro Evolution Soccer 3*, conosciuto anche come *World Soccer: Winning Eleven 7* in Giappone, è un simulatore calcistico sviluppato da **Konami** nel 2003; è considerato ancora oggi uno dei migliori giochi di calcio di sempre: *PES* è da sempre l'eterno rivale di*FIFA* che invece presenta ritmi più arcade e molto meno realistici.

3.7.5 Gioco di Ruolo

Spesso abbreviato come **GDR** o **RPG** (*Role Playing Game*, questi giochi vedono al centro dell'esperienza l'evoluzione di uno o più personaggi in un mondo ben definito e seguono regole spesso mutuate da giochi di ruolo cartacei alla *Dungeons & Dragons*. Solitamente sono giochi ambientati in universi *high fantasy* (ispirati alla cosmogonia di J.R.R.Tolkien), ma non sono rare le ambientazioni fantascientifiche o post-apocalittiche.
Negli *RPG* il giocatore è chiamato a completare una lunga missione chiamata **main quest**, collante dell'intera avventura, e, nel frattempo, può incontrare personaggi secondari che assegnano missioni secondarie, dette **side quest**, che prolungano l'esperienza ludica e consentono al giocatore di accumulare punti esperienza, utili per *salire di livello*. Grazie al completamento delle *quest* il **pg**, gergo dell'*RPG* per definire il personaggio controllato, o il **party**, gruppo di personaggi controllati, diventa più forte, quindi in grado di affrontare territori più ostili e nemici più temibili, quindi incontrare nuovi personaggi ed affrontare altre quest. Inoltre, durante i suoi viaggi, il giocatore solitamente può effettuare il **looting**, ossia esplorare **dungeon** (sotterranei o aree simili) o saccheggiare i nemici abbattuti, per ottenere un equipaggiamento migliore con cui vestire ed armare il proprio pg o il proprio party; non è insolito trovare meccaniche di *crafting* per le materie raccolte e la fabbricazione di oggetti utili da gestire nell'inventario.

Il giocatore ha ampia libertà nel plasmare il carattere del proprio personaggio ed il suo atteggiamento nei confronti del mondo che lo circonda, in modo da interpretare un personaggio positivo o vestire i panni dell'antieroe, avendo ripercussioni forti sull'andamento della storia. L'evidente complessità del genere ha permesso la creazione di svariati sottogeneri da esso derivati; inoltre non è raro trovare meccaniche *RPG* in giochi di genere completamente diverso: persino in alcuni giochi sportivi sono presenti meccaniche di *progressione* derivate dalla storia dei giochi di ruolo. In conclusione si può affermare che gli *RPG* risentano più di tutti delle influenze culturali dei paesi in cui vengono sviluppati, non è un caso che i giochi di ruolo orientali siano totalmente differenti da quelli occidentali.

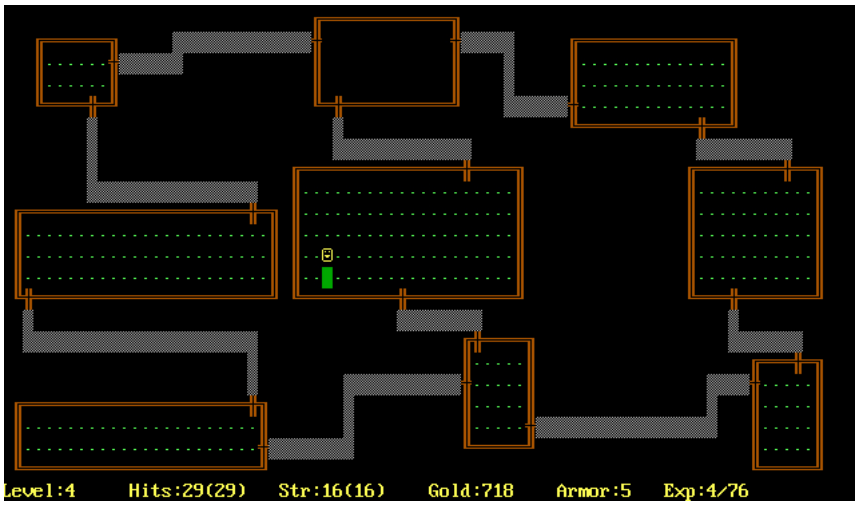


Figura 148: *Rogue: Exploring The Dungepons of Doom* è un videogioco apparso nel 1980 sui mainframe Unix e sviluppato da **Michael Toy**, **Glenn Wichman** e **Ken Arnold**, poi riedito in versione definitiva nel 1983 e votato da PC World Magazine come il sesto miglior gioco per PC di tutti i tempi; nonostante abbia la grafica composta unicamente da caratteri di testo e codice ASCII, ha ispirato una classe di giochi derivati noti collettivamente come **roguelike** ovvero "stile Rogue", oltre ad essere considerato come il primo *RPG* della storia dei videogames.

Action RPG Si tratta di una delle forme di *RPG* più popolari in assoluto, che mescolano le componenti del gioco di ruolo con quelle del gioco *action*; il giocatore è chiamato a gestire l'evoluzione di un personaggio, ma non solo: deve avere una buona coordinazione occhio-mano nelle fasi di combattimento, che si svolgono sempre in tempo reale (mai a turni). Il primo titolo che ha saputo rendere popolare questo genere è senza ombra di dubbio *Diablo* (**Blizzard Entertainment**, 1996), mentre serie come *The Elder*

Scrolls (**Bethesda Softworks**, 1992) e *The Witcher* (**CD Project Red**, 2007) sono tra i franchise più popolari al mondo.

Figura 149: *Baldur's Gate* è un franchise amatissimo che riprende a piene mani le meccaniche e le regole proposte dal cartaceo *Dungeons & Dragons*, miscelandolo con il genere action; in figura uno screenshot del terzo capitolo rilasciato in early access il 6 ottobre 2020.

JRPG I giochi di ruolo giapponesi, detti appunto **JRPG** (*Japanese Role Playing Game*), differiscono dagli *RPG* occidentali per svariati motivi; l'ambientazione, ad esempio, è sicuramente fantastica, ma raramente del genere *high fantasy*, oppure il carattere dei personaggi non è determinato dalla trama del gioco e non dalle azioni del giocatore, ma stabilito a priori. La differenza principale però sta nelle meccaniche di gioco: nei *JRPG* esiste un particolare sistema di alternanza tra esplorazione e combattimento. Durante l'esplorazione, il giocatore ha il pieno controllo del personaggio, mobile in ogni zona esplorabile come in un normale *action RPG*; quando si intercetta un nemico, la situazione cambia interrompendo, de facto, il gioco: si viene trasportati in un'arena dove si svolge un combattimento a turni con statistiche numeriche legate ai personaggi controllati e ai nemici, oltre ad una simbolistica utile a favorire la comprensione di uno specifico *combat system* (solitamente il giocatore deve selezionare le proprie azioni da un elenco).
Serie come *Final Fantasy* (prima **Square Soft**, poi *Square Enix*) o *Dragon Quest* (prima **Enix**, poi **Square Enix**) hanno reso famosissimo questo genere a partire dagli anni '80, vendendo milioni di copie nel mondo, anche se da alcuni anni i tradizionali *JRPG* sembrano orientarsi verso il modello degli *action RPG* occidentali, proponendo combattimenti in tempo reale, definiti appunto **action JRPG** (come ad esempio il franchise diretto da **Tetsuya Nomura**, *Kingdom Hearts* che ha avuto la prima iterazione nel 2002). Tuttavia, i *JRPG* contemporanei rimangono inconfondibili per stile e gameplay, oltre che molto apprezzati dal grande pubblico. Si pensi alla serie *Pokémon* (**Nintendo**, 1996) che ha generato veri e propri fenomeni di costume globali che resistono da oltre vent'anni.

Figura 150: La serie *Final Fantasy* ha cambiato il mondo dei videogames diverse volte con storie e videogiochi splendidi che hanno fatto innamorare schiere numerosissime di giocatori; in figura *Final Fantasy VI*, diretto da **Hironobu Sakaguchi** e **Yoshinori Kitase**, l'atmosfera cambia radicalmente dai suoi precedenti introducendo un'ambientazione più cupa ed uno dei villain meglio realizzati di sempre, Kefka Palazzo.

GDR Tattici Quando l'intero gioco è basato su combattimenti a turni che si svolgono su di una scacchiera, pur mantenendo un sistema di evoluzione dei personaggi ed una storia, si parla di *gioco di ruolo tattico*: il giocatore muove i propri personaggi come se fossero delle pedine, attivando le mosse corrette al fine di sconfiggere gli avversari.
Questa tipologia di giochi ha avuto molta fortuna in Giappone con serie di *Fire Emblem* (**Nintendo**, 1990) e *Disgaea* (**Nippon Ichi**, 2003) che hanno ottenuto successo anche nei paese occidentali.

Figura 151: *The Banner Saga* è un ottimo esempio di *GDR tattico* sviluppato da **Stoic Studio** e **Versus Evil** nel 2014, che si svolge in un mondo fittizio basato sulla mitologia norrena.

Roguelike Il già citato gioco del 1980, *Rogue: Exploring The Dungeons of Doom*, ha dato il nome ad un filone di giochi ben definito in cui il giocatore è posto in un sotterraneo generato in maniera procedurale, ossia la mappa di gioco, nemici e tesori si generano casualmente. Solitamente i *roguelike* sono caratterizzati dalla **permadeath** del personaggio: il giocatore, se perde la vita in game, non può continuare la partita, ma deve ricominciarla da capo con un nuovo eroe, in un nuovo labirinto.
Da questo genere deriva il sottogenere **dungeon crawler**, un gioco solitamente in prima persona in cui ci si muove in una mappa a scacchiera generata proceduralmente e dove, generalmente, non è presente il meccanismo della *permadeath*.

Soulslike Questi giochi derivano da una serie creata da **FromSoftware** ed è iniziata nel 2009 con *Demon's Souls* per proseguire nel 2011 con *Dark Souls*, divenuto un successo planetario. Il contesto storico in cui nasce questo genere è fondamentale per comprenderne il successo: se nel primo decennio dei Duemila, la popolarità dei videogiochi ha spinto gli sviluppatori ad abbassare la difficoltà media della sfida di gioco e proprio per questo molti titoli apparivano meno appaganti ai giocatori più esperti, affamati di challenge sempre maggiori. *Demon's Souls* esce nei negozi nel 2009 e sorprende il mondo videoludico con un grado di sfida elevatissimo per gli standard dell'epoca: il gioco è "brutale" e basta davvero poco per morire durante un

partita. Proprio la "morte" diventa parte integrante del gameplay: trasformando per sempre il concetto di game over, infatti morire in gioco diventa un meccanica intelligente e rivoluzionaria, integrante dell'esperienza di gioco. Il suo successore spirituale, *Dark Souls*, recupera questi elementi chiave e li inserisce all'interno di un'ambientazione di gioco con un'affascinante mitologia. Nonostante sia un genere video-ludico difficilmente codificabile e, pertanto, da quel momento in poi tutti gli altri prodotti sono stati definiti appunto come "cloni dei Souls".

Figura 152: L'amore per i giochi del genere *soulslike* dal 2011 in poi è cresciuto a dismisura: alle porte della nuova generazione di console, il lancio della PS5 sarà accompagnato dall'uscita di *Demon's Souls Remake*, sviluppato da **Bluepoint Games** come titolo di lancio.

3.7.6 Strategico

Uno dei primi esperimenti video-ludici della storia, *Nimrod* (basato sul gioco matematico *Nim*) è il primo gioco strategico della storia e da questo tipo di gioco derivano moltissime meccaniche adottate in giochi appartenenti a generi odierni, considerati totalmente alieni alla strategia.
Nel corso del tempo il genere *strategico* è stato ampiamente codificato, oltre che essere all'origine di alcuni importantissimi generi video-ludici, è giocato da milioni di appassionati da oltre quarant'anni.

Strategico a turni I primi giochi strategici si basano sulle regole proposte dal gioco da tavolo in stile *Risiko*: il giocatore ha a disposizione un turno per effettuare le sue mosse, che quindi vengono eseguite ed il turno passa all'avversario. Tale meccanismo venne adottato sin dai tempi di *Empire* (**Interstel**, 1977), poi impreziosito da un crescente numero di azioni che il giocatore può compiere durante il proprio turno, tanto che ad oggi i giochi strategici a turni sono davvero complessi; ad esempio, *Civilization*, creato da **Sid Meier** nel 1991, è la serie che più di ogni altra ha reso popolare questa tipologia di giochi.

Figura 153: *Sid Meier's Civilization 6* è l'ultima iterazione del franchise lanciato nel 1991.

RTS Definiti *Real Time Strategy*, i giochi strategici in tempo reale consentono al giocatore di effettuare le proprie mosse in tempo reale, assieme ai propri avversari: solitamente sono caratterizzati da una visuale *isometrica* grazie alla quale è possibile selezionare più truppe per direzionarle, dato che gli eserciti avversari vengono dislocati in luoghi distanti della mappa. É molto importante saper utilizzare le truppe e le risorse efficacemente per avere la meglio sugli eserciti avversari: le meccaniche si basano spesso su quelle del gioco sasso-carta-forbice, poiché un tipo di truppa è letale contro un secondo tipo di truppa, ma debole per una terza.
In molti casi, i giochi *RTS* si accompagnano ad una meccanica che consente al giocatore di creare e migliorare le proprie truppe attraverso l'**albero delle tecnologie** o le strutture per lo sviluppo delle truppe. Questa tipologia di *RTS* è la più comune e all'origine di alcune serie di grande successo come *Age of Empires* (**Microsoft**, 1997) e *StarCraft* (**Blizzard Entertainment**, 1998). Una curiosità: nonostante la maggior parte di questi titoli abbia trovato terreno fertile su PC, uno degli *RTS* più importanti per il genere è **Herzog Zwei**, titolo del 1989 sviliuppato da **Technosoft** per console, ovvero SEGA Mega Drive.

Tower Defense Se un gioco di strategia prevede la mera costruzione di una base capace di resistere alle ondate degli attacchi nemici, viene classificato sotto questa etichetta: sono giochi in cui si assiste agli attacchi di un nemico mentre si tenta di respingerli costruendo torri di difesa e marchingegni mirati a rallentare l'avanzata avversaria. Questo genere di giochi, caduto nel dimenticatoio negli anni Ottanta, è poi tornato in voga con l'avvento degli smartphone, dove giochi come *Plants Vs. Zombies* (**PopCap**, 2009) sono stati scaricati da decine di milioni di persone.

MOBA Acronimo di *Multiplayer Online Battle Arena*, è un genere affine alla strategia in tempo reale, in cui due squadre di giocatori si scontrano su mappe perfettamente simmetriche, dotate di un numero ristretto di corridoi di passaggi (*lane*), nel tentativo di espugnare la base nemica. É un gioco esclusivamente multiplayer, in cui più giocatori cooperano per sconfiggere un'altra squadra; solitamente ogni giocatore controlla un eroe con un ruolo preciso, determinato dalle skill possedute dall'eroe stesso, con il quale può distruggere sia gli eroi avversari che i *minions*, deboli servitori utili al farming e al level up dei personaggi stessi o all'avanzata nelle lane.
I *MOBA* vedono in *League of Legends*, sviluppato da **Riot Games** nel 2009, il titolo più celebre: questi giochi sono di un'importanza incredibile per la diffusione degli eSport, infatti *League of Legends* vanta cento milioni di giocatori mensili ed una lega professionistica con montepremi milionari.

3.7.7 Racing

Questo macrogenere è stato discusso ampiamente poiché i videogiochi di guida potrebbero ricadere sotto la definizione di giochi sportivi, poiché volti a virtualizzare un gara automobilistica; i *racing* game possono presentare un gameplay simulativo (**sim racer**) e giochi con un gameplay totalmente opposto, ovvero senza alcuna pretesa di simulazione quali gli **arcade racer** o i **kart game** (come ad esempio la serie *Mario Kart*). Anche se molti prodotti di grande successo come *Gran Turismo* (**Poliphony Digital**, 1997) o *Forza Horizon* (**Turn 10 Studios**, 2005) si autodefiniscono simulazioni di guida, sono ben lontani dai più complessi *racing sim* usati dai piloti per allenarsi.

Figura 154: Tra i *racing game* più amati sicuramente rientra anche la serie *Need For Speed*, sviluppata da **Electronic Arts**; nella sua iterazione presentata in figura, *Need For Speed Underground* rilasciato nel 2003, il giocatore viene catapultato in un mondo fatto di corse clandestine ed automobili modificate: la personalizzazione di queste ultime è per l'appunto una delle feature più apprezzate dai giocatori.

3.7.8 Simulazione

L'etichetta *simulazione* può essere applicata ad un'enorme varietà di giochi e risulta difficile tracciare un linea che permetta di chiarire nettamente quando un gioco è simulativo o quando un gioco appartiene ad un genere x con pretese simulative; non è sbagliato affermare che esistano simulatori di volo, simulatori di guida, simulatori di vita, simulatori calcistici e via dicendo. Esiste però una tassonomia ben precisa per alcuni sottogeneri di questa macro area che verranno spiegati nei prossimi paragrafi.

Gestionale Sono giochi dove il giocatore è chiamato a costruire e/o mantenere un ecosistema, sia esso di natura economica, naturale, politica o gerarchica, richiedendo una buona abilità manageriale da parte dei giocatori. Un serie di grande successo è *SimCity*, sviluppata per la prima volta da **Maxis** nel 1989, dove il giocatore è chiamato a far proliferare una città, mentre un altro esempio è *Football Manager* di **SEGA** (2004), dove l'obiettivo del gioco è la gestione efficace di una squadra di calcio.

Life Simulator Il nome è molto esplicativo della tipologia di gioco e dell'esperienza che il giocatore andrà a provare: i *life simulator* mirano a simulare le attività biologiche e sociali di una forma di vita umana, animle o fantastica; se la simulazione riguarda un animale si tratta di *pet simulator*, genere con un ampio riscontro di pubblico negli anni Novanta, con il gioco tascabile *Tamagotchi* in cui il giocatore si prendeva cura di un pulcino. Di grande impatto, per questo genere di giochi fu *The Sims*, prodotto sempre da **Maxis** nel 2000, in cui il giocatore deve gestire la vita biologica, lavorativa e sociale di un proprio alter ego virtuale.

Simulatori di volo I giochi simulativi comparsi per primi sono i *simulatori di volo*, sia militari che civili; la serie *Microsoft Flight Simulator* (**Microsoft**, 1982) fu fortemente voluta dall'azienda di DOS e Windows per dimostrare le potenzialità dei personal trainer a 16 bit rispetto ai computer a 8 bit, realizzati dalla rivale Apple di Steve Jobs e Steve Wozniak: proprio da questa prova di forza, è nato un genere di giochi che ha raggiunto livelli di simulazione molto elevati e community molto attive anche nella creazione di nuovi contenuti.

Job Simulator Tra le più curiose tipologie di giochi simulativi rientrano i *job simulator*, ovvero quei videogames che puntano a simulare un'attività lavorativa, come i simulatori di chirurgia, di cucina, dell'attività di un agricoltore o di un pescatore (ne esistono davvero di ogni tipo). La serie annuale di **Giant Software**, *Farming Simulator* (la cui prima iterazione risale al 2007), in cui si eseguono le attività di un contadino alle prese con i macchinari e le azioni utili alla gestione di un'azienda agricola, è diventata una

vera e propria saga di culto per migliaia di appassionati, vendendo circa un milione di copie ad ogni nuova uscita.

Figura 155: Il realismo raggiunto nei titoli di *Farming Simulator* è davvero strabiliante: il giocatore dovrà imparare ad utilizzare un numero davvero consistente di mezzi agricoli, tra cui irroratrici, mietitrebbie o seminatrici.

God game Si tratta di una simulazione davvero particolare, ovvero quella delle azioni di una divinità: il giocatore ha a disposizione poteri divini limitati con i quali supervisiona una comunità formata da individui dotati di libero arbitrio; guadagnandosi il favore dei fedeli seguendo la strada del dio misericordioso o del dio vendicativo, l'utente dovrà affrontare una serie di task e guidare la propria comunità verso la prosperità. Il titolo di maggior appartenente a questo genere è *Black & White* sviluppato da **Lionhead Studios** nel 2001: il titolo presentava una straordinaria intelligenza artificiale, capace di apprendere dalle proprie azioni.

3.7.9 Ritmico

Grazie alla diffusione di processori audio sempre più performanti, i videogame hanno iniziato ad investire sempre di più sul comparto audio e nel campo della ritmicità; il frutto di questa ricerca ed investimento è stata la creazione di un genere definito *rhythm game*, in cui il giocatore deve compiere delle azioni a tempo di musica, agendo su controller o su PC oppure su accessori particolari creati ad hoc per questo tipo di giochi. Giochi come *Guitar Hero* (**Harmonix Music Systems**, 2005) includevano nelle confezioni di gioco un chitarra in plastica con i pulsanti di controllo del gioco; un altro esempio celebre è *Dance Dance Revolution* (**Konami**, 1998) faceva uso di un *dance mat*, un tappetino sensibile alla pressione su cui il giocatore deve ballare seguendo le frecce che appaiono sullo schermo.

3.7.10 Puzzle Game

Essendo i rompicapi i giochi più diffusi nel mondo, le componenti *puzzle* sono quelle che con maggior frequenza compaiono in giochi appartenenti ad altri generi (come gli *action-adventure*). Nato alla fine degli anni Settanta, inizialmente come trasposizione virtuale di giochi popolari, tra i quali si ricordano *Hangman* (**Alan Miller**, 1978) e *Atari Video Cube* (**Atari**, 1982), il genere ha preso successivamente le distanze da giochi da tavolo e rompicapi di stampo classico, creando concept inediti che sarebbero diventati successi planetari nell'immaginario collettivo. Si pensi ad esempio a *Tetris*, creato nel 1984 presso l'Accademia Sovietica delle Scienze di San Pietroburgo da un giovane esperto nel campo dell'intelligenza artificiale, **Aleksej Leonidovič Pažitnov**: è probabilmente uno dei titoli più conosciuti a livello mondiale, distribuito su oltre 65 piattaforme e ha venduto più di 170 milioni di copie, dato che lo rendeva, prima dell'arrivo di *Minecraft*, il videogioco più venduto di sempre.
I *puzzle game* hanno curve di apprendimento docili, favorendo l'approccio da parte di un pubblico occasionale , e proprio con l'avvento degli smartphone hanno trovato nuovo slancio nel mercato: titoli come *Candy Crush Saga* (**King**, 2012) hanno invaso gli store, grazie ad un modello economico che

consentiva la loro distribuzione in maniera gratuita, sono stati scaricati 2,5 miliardi di volte.

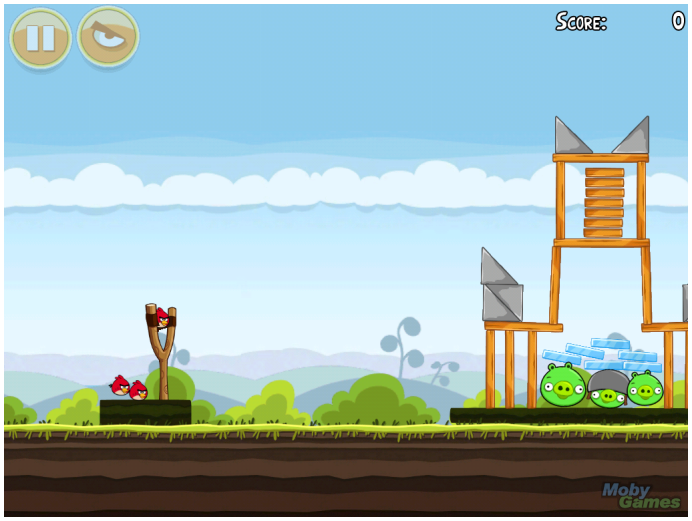


Figura 156: Sviluppato da **Rovio Entertainment** nel 2009, *Angry Birds*, originariamente venduto a 99 centesimi di dollaro, ha superato le 3 miliardi di copie in tutte le sue versioni.

3.7.11 Gli Pseudo-Generi

Oltre ai macro-generi appena visti, vi sono etichette applicate ai videogiochi che si possono identificare come dei "falsi generi": si tratta di elementi che determinano alcune features di gioco ma che contemporaneamente necessitano di essere abbinati ad un altro genere per risultare sensati.

MMO I *Massively Multiplayer Online* sono videogame caratterizzati dalla presenza di un grande numero di giocatori connessi fra loro tramite internet: proprio per questa feature, gli *MMO* sono dotati di universi persistenti, community enormi e sfide su vasta scala.
Il termine si applica solitamente ai giochi di ruolo online caratterizzati da milioni di giocatori, detti **MMORPG**: sottogenere di particolare fortuna che vede come al suo apice *World Of Warcraft* (**Activision Blizzard**, 2004) che ha raggiunto i 10 milioni di giocatori nel 2010.
Non mancano gli *MMO* legati ad altri generi come gli **MMORTS** (*RTS*), **MMOFPS** (*FPS*) o altri appartenenti al genere *racing*, *action* e così via.

Figura 157: Gli *MMORPG* sono giochi in cui i giocatori possono creare un alter ego virtuale con cui intraprendere numerose avventure all'interno del mondo di gioco: non è raro ascoltare storie di giocatori che si sono conosciuti su questi giochi e poi hanno intrapreso relazioni di amicizia o d'amore nella vita reale (con tanto di cerimonia nuziale fatta nel gioco: in figura un matrimonio celebrato su *World Of Warcraft*).

Casual Si tratta di giochi pensati per giocatori occasionali (definiti **casual gamers**, contrapposti ai **core gamers**), quindi caratterizzati da meccaniche molto semplici che richiedono uno scarso impegno da parte del giocatore, quasi fossero dei passatempi. Anche se il termine venne coniato nel 2007 con l'avvento dei giochi per mobile, è stato applicato ante-litteram a grandi titoli del passato come *Pac-Man* (**Namco**, 1980), appartenente al genere *maze*.

Arcade Termine coniato per indicare tutti i videogiochi presenti nelle sale giochi al fine di distinguerli dai giochi per console casalinghe, ad oggi è utilizzato per descrivere quella tipologia di gioco caratterizzato da brevi livelli, sistemi di controllo semplici, presenza di vite ed elevata difficoltà. Oltre questa definizione, l'etichetta *arcade* viene usata spesso in contrapposizione a giochi con ambizioni simulative: tutto ciò che si basa su un'attività reale (uno sport, un mestiere, ...), ma che viene reinterpretato per incentivare la giocosità ed il divertimento.

Sandbox I *sandbox* sono giochi in cui il giocatore non ha delle vere e proprie regole o uno scopo ben preciso: il nome rimanda volutamente alla sabbiera per bambini proprio per sottolineare come il giocatore può sfruttare il mondo di gioco a proprio piacimento, agendo in piena libertà sulle risorse disponibili nell'ambiente di gioco. Molti giochi *sandbox* appartengono al genere *survival*, ma l'etichetta è soventemente applicata a giochi come *Grand Theft Auto* o come *Elite: Dangerous* (**Frontier Developments**, 2014).

3.7.12 Progressione Vs. Emersione

Dopo la lunga analisi fatta sui generi video-ludici, occorre definire alcuni concetti all'interno dell'industria, utili alla rappresentazione delle dinamiche interne ad un gioco, in modo da sapersi sintonizzare sulla stessa idea e visione; infatti esistono due dinamiche di svolgimento diverse all'interno del mondo dei videogames: si individuano i concetti (quasi filosofici) di **progressione** ed **emersione**.
Il concetto di progressione, molto più semplice di quello di emersione, individua quelle dinamiche di gioco tipiche dei videogame con un impatto narrativo rilevante: il giocatore per avanzare nel gameplay, deve compiere una determinata azione che gli consenta di superare un ostacolo, rappresentato da un nemico, un puzzle o una qualsiasi altra sfida presentata in game. Per semplificare si faccia riferimento a quest'esempio: all'interno di un *RPG*, molto spesso, ci si imbatte in quest che richiedono al giocatore di entrare in un dungeon segreto, il quale è soventemente bloccato da un sigillo meccanico o magico; il giocatore infatti dovrà trovare una "chiave" per poter entrare ed affrontare il dungeon, al fine di completare la quest: questa è un tipico gameplay di progressione. Di più difficile comprensione è l'altro concetto

che verrà spiegato qui di seguito: nella nomenclatura di settore, quando ci si riferisce a gameplay di **emersione** si intende quel tipo di gioco dove ho una "singola" schermata che fa da testimone a tutto lo svolgimento ludico del gioco. Il gameplay, infatti, *emerge* gradualmente, inserendo meccaniche e possibilità di gioco man mano che il tempo passa. Di che giochi si parla? *Strategici*, *RPG*, *Simulatori* sono sicuramente generi che propongono un gameplay di emersione: titoli come *Sid Meier's Civilization VI*, *Adventure Capitalist* (**Hyper Hippo Productions**, 2014), *SimCity*, *Clash Of Clans* (**Supercell**, 2012), *Forza Horizon 3*, *Overcooked* (**Team17**, 2016) propongono come *key feature* la gestione di un'economia interna, basata su svariati aspetti a seconda del tema di gioco, con meccaniche precise che consentono di raggiungere determinati task o risolvere determinati problemi.

Figura 158: Nella serie *Overcooked*, il giocatore sarà chiamato a gestire una cucina: ordini, preparazione dei piatti e consegna avvengono tutti nella stessa schermata, nella quale i giocatori devono muoversi in maniera studiata e coordinata per eseguire più comande nel tempo stabilito dal livello; si tratta a tutti gli effetti di un gioco in cui risulta fondamentale gestire accuratamente le risorse di tempo e spazio.

La narrazione non è prestabilita poiché viene gestita tramite l'emersione del gameplay: in *The Sims*, ad esempio, le "storie" dei personaggi sono "scritte" dai giocatori e variano di partita in partita.
Oltre alla narrativa, i videogiochi di emersione hanno altri punti chiave legati irrimediabilmente alle dinamiche di gioco: trattandosi di giochi prettamente manageriali o gestionali, la presenza di una **mappa** all'interno della UI è fondamentale; la gestione di oggetti, pedine o personaggi singoli, quindi di **unità**, è una meccanica complessa ed avere una mappa è utile all'orientamento del giocatore. Infatti l'interfaccia deve essere particolarmente dettagliata, ben realizzata ed intuitiva: nei giochi di progressione, questa può non esserci, poiché non si ha necessariamente bisogno di inviare input ad entità terze, cosa "obbligatoria" nei giochi di emersione.
Uno dei nemici più ostici all'interno dei gameplay di emersione è il tempo: è il counter utile per determinare le azioni del giocatore e si fa demiurgo del susseguirsi degli eventi all'interno della partita; ad esempio, dopo circa un'ora all'interno di *SimCity* si ha una determinata probabilità che si verifichi un'eruzione vulcanica o un altro disastro progettato dagli sviluppatori. **Imprevisti** e **casualità** contribuiscono alla varietà del gameplay, quindi alla sua emersione costante, mentre l'attività prevalente del giocatore rimane legata alla gestione dell'economia interna del gioco.

Figura 159: Nel gameplay di emersione risultano fondamentali imprevisti e casualità per aggiungere dinamica al gameplay, evitando il possibile *plateau* dopo svariate ore di gioco: l'utilizzo delle calamità all'interno della serie *SimCity* è un esempio lampante di questo concetto.

Riassumendo quali elementi servono per sviluppare un videogioco di emersione, prettamente di tipo manageriale/gestionale, si può stilare una lista di questo tipo:

- Tempo
- Economia – Risorse, Denaro, ...
- Unità (intese come entità terze a cui inviare input)
- User Interface
- Eventi casuali ed imprevisti
- Meccaniche ben definite
- World building narrative

Oltre a questi elementi di base, si deve valutare come distribuire la qualità generale nel gioco senza creare scompensi, calcolando su quanto investire su ognuno dei *4 pilastri* (vedi 3.4) dello sviluppo video-ludico.
Si supponga di avere un budget per lo sviluppo di un *gestionale* a tema colonizzazione spaziale: come dividere l'investimento di risorse in maniera efficiente per la produzione per questo videogioco di emersione? Per un *gestionale/manageriale*, le meccaniche hanno sempre una grande rilevanza nella costruzione del game object, quindi una grossa fetta del budget verrà utilizzata in quel settore; parti consistenti verranno investite nel reparto estetico e tecnologico, mentre la fetta più piccola presumibilmente toccherà alla storia: in effetti non essendoci una narrazione prestabilita il lavoro di scrittura diventa marginale rispetto a titoli di progressione. Lavorando con gli elementi definiti prima: il giocatore dovrà espandere e gestire la colonia spaziale, facendo fronte alle sfide che che gli si porranno davanti (imprevisti, raccolta e distribuzione risorse, ...). La mappa sarà composta da celle esagonali (ispirandosi a *Civilization*) che favoriscono il movimento delle unità (pedoni, veicoli, ...) all'interno dell'ambiente di gioco, mentre l'economia sarà basata sulle risorse e non sull'utilizzo di una valuta in game, basando il gameplay completamente sull'esplorazione e la colonizzazione.
Le possibili meccaniche potrebbero essere: estrazione risorse, eventi casuali, esplorazione, il rapporto con altre civiltà, l'uso della diplomazia, il commercio, il punteggio, un sistema di turni, AI concorrente, la possibilità di analizzare lo "stato-celle" della mappa, eventi "alieni", ...; stilando liste del genere, si rischia di avere troppi elementi e diventa impossibile avere un game design ben bilanciato. Occorre fare una scelta e mantenere gli elementi che risultano più funzionali. In questo caso, l'estrazione delle risorse da accumulare in base per utilizzarle durante la partita, poiché utili di fronte ad imprevisti o task che potrebbero aumentare il livello della sfida e la produzione di veicoli/unità da usare per varie attività. Dopo aver stabilito

un certo numero di elementi fissi entro i quali è possibile arricchire il gameplay,si cerca di sviluppare parametri che possano cambiare il corso della partita, facendo emergere pian piano situazioni di gioco nuove. In che modo si può arricchire un gameplay di questo tipo? Si possono implementare aree esplorabili solamente con determinati veicoli, quindi serviranno vari tipi veicolo da sbloccare grazie ad un albero di tecnologie (una meccanica analoga potrebbe essere impiegata per gli edifici); si dovranno gestire le risorse utili per la demografia e lo sviluppo della popolazione, attraverso parametri specifici disponibili nell'interfaccia del giocatore ed anche il trasporto delle risorse diventerà fondamentale: si rischia di incrociare specie pericolose, affrontare imprevisti e dover effettuare la manutenzione di veicoli ed edifici. Si potrebbe andare avanti e aggiungere ulteriori elementi, cercando di costruire una narrativa di emersione: risulta evidente di come i vincoli che progettati all'inizio, servono per dare identità al gioco in questione, effettuando un focus sulle concrete possibilità di sviluppo. Se ci si sofferma su una singola meccanica, ad esempio, il veicolo si potrebbero creare ulteriori dinamiche di gioco: ogni veicolo appartiene ad un specifica classe che ne influenzerà caratteristiche e funzioni; si può immaginare che i veicoli di trasporto avranno determinate *caratteristiche di movimento* dettate dalle velocità su un determinato tipo di terreno, la *capacità di estrazione* delle risorse o di *tempo di costruzione*, e via dicendo. Lo stesso ragionamento è fattibile per le altre tipologie di veicoli creando sempre più varietà: lavorare anche su un singolo elemento consente al gioco di prendere forma ed espandere. Come si svolgerà la partita? Ad esempio, il giocatore potrà interagire con gli edifici: cliccando sull'edificio si potranno verificare i lavoratori presenti, quelli necessari, il consumo energetico ed altre opzioni; oppure cliccando su una casella vuota della mappa, il giocatore vedrà a schermo che edifici si possono costruire con le risorse disponibili, i lavoratori attualmente disponibili e le tecnologie attualmente disponibili; una volta che selezionato l'edificio da costruire, comincia l'animazione di costruzione (es: animazione cantiere). Quello appena analizzato è uno schema di come potrebbe essere lo sviluppo di alcune meccaniche di gioco per lo sviluppo di un videogioco di emersione.

4 Redazione del Manuale: le necessità e le finalità

Il fulcro di questa tesi, oltre alla propedeuticità dell'analisi fatta nel capitoli 2 e 3, è fungere da resoconto del progetto sviluppato in collaborazione con Ivan Venturi ed IVProductions, ovvero quello della redazione e stesura di un manuale di game design e production, completamente pensato per lo sviluppo italiano. Il manuale, infatti, rappresenta in potenza un futuro tomo di supporto, utile agli addetti ai lavori più esperti o ai novizi del settore.
Prerogativa di IVProductions, durante la collaborazione ancora in corso, è sempre stata quella di sottolineare come non esista un prodotto del genere sul mercato per quanto riguarda la realtà italiana; infatti dopo aver visionato i numerosi manuali diffusi nell'ambiente dello sviluppo video-ludico italiano, si è notato che la maggioranza dei volumi di rilievo sono in inglese e basati su realtà industriali completamente differenti da quella italiana. Sebbene si tratti di manuali utilissimi o riferimenti letterari di rilievo per quanto riguarda l'analisi del mondo dei videogames da diversi fronti, risultano carenti quando si tratta di entrare nel reale merito delle cose: obiettivo di IVProductions, attiva anche nella crescita dell'intero settore, è quello di creare un testo di riferimento che spieghi ed analizzi *step by step* la progettazione e la produzione di un videogioco in Italia, non solo cogliendo gli aspetti fondamentali del game development, ma anche quelli commerciali ed imprenditoriali. Una critica ad alcuni testi analizzati sembra proprio essere la mancanza di coerenza tra il *workflow* teorico e quello reale: questo cozzare tra teoria e pratica, legato soprattutto alla realtà italiana, ha spinto IVP a puntare su questo progetto, con la supervisione del proprio founder, Ivan Venturi.

4.1 Target di riferimento

Al centro del progetto ci sono gli sviluppatori: i circa trent'anni di esperienza maturata nel campo dei videogames da Ivan Venturi vogliono essere cristallizzati all'interno di questo manuale in modo da risultare utile per chi vorrà affrontare questa carriera o chi la sta già svolgendo e sente il bisogno di un testo di riferimento per il proprio lavoro. Proprio in occasione degli *Svilupparty* organizzati nel corso degli anni, si è notato come molti team di sviluppo capaci, abbiano riscontrato difficoltà nella produzione del loro gioco, non tanto a livello tecnico, quanto a livello imprenditoriale ed economico: spesso sviluppatori con ottimi giochi, non sono poi riusciti a concretizzare il progetto di sviluppo, proprio a causa dell'inesperienza in ambito imprenditoriale ed economico. Basti pensare che alcuni team con il prodotto pronto al lancio sul mercato, non hanno potuto procedere per una mancanza organizzativa interna in termini economici: ad esempio, è capitato che un team formato da più di una decina di persone avesse sviluppato un gioco molto interessante, ma non avesse ancora deciso il modello di business da applicare e nemmeno a come suddividere le quote del progetto.

Questo genere di errori è tipico degli sviluppatori inesperti: quello che manca, molto spesso, non è la passione, ma la capacità imprenditoriale e di produzione; il manuale è pensato proprio per questo tipo di persone.

4.2 Lo Stato dei Lavori

I lavori al progetto sono cominciati nell'estate 2020 proprio con lo sviluppo di questa tesi, utile a livello propedeutico per il progetto definitivo: l'idea era quella di creare un possibile indice di massima, che potesse fungere da base di lavoro per il manuale vero e proprio.
Ad oggi, il progetto è ancora in fase di **pre-produzione**: è in corso una notevole raccolta dati e schematizzazione della struttura del manuale.
Proprio il lavoro sugli schemi della struttura del libro risultano fondamentali per la scelta dei contenuti e per il percorso del lettore: ciò su cui ci si sta focalizzando è il filo conduttore del manuale, come impostare i contenuti selezionati e quali effettivamente selezionare; il prossimo passo sarà quello di terminare i lavori di pre-produzione entro la fine del 2020 e passare alla produzione, quindi scrittura del manuale nei primi mesi del 2021, entrando nella fase di scrittura a metà gennaio.

4.2.1 Idee di Progettazione

La redazione del manuale, nella sua fase embrionale, ha richiesto l'analisi del mercato dei libri di testo già disponibili al pubblico: il testo di riferimento per eccellenza è *The Art of Game Design: A Book of Lenses* di Jesse Schell, un libro fondamentale per ogni aspirante game designer; risulta essere un tomo gradevole da leggere, ma la sua struttura risulta molto più teorica con uno stampo didattico e poco pratico, nel senso più concreto del termine; è sicuramente uno dei manuali più completi sul game design, cercando di coprire una grossissima mole di argomenti legati all'ambito del game design, rimanendo legato ad una posizione generalista: il punto forte del libro è legato alla sua struttura a *lenti* che cerca di sintetizzare dei concetti in aree ristrette e divise per tematiche. Il testo è totalmente in lingua inglese ed ovviamente è stato scritto in un contesto industriale/produttivo molto più maturo rispetto a quello italiano: sicuramente è un punto chiave che mette in luce una lacuna in ottica del mercato italiano. Un altro esempio è *Game Mechanics: Advanced Game Design* di Ernest Adams e Joris Dormans, un libro più tecnico che entra nel merito della gestione delle meccaniche e di come utilizzare Machinations, ovvero un software che permette la schematizzazione e la simulazione delle meccaniche di gioco. É un testo più tecnico del precedente e con prerogative completamente differenti, coprendo un campo sicuramente più ristretto: non vengono coperti gli aspetti della produzione video-ludica, ma viene esplorato solamente l'ambito del *mechanics design* ed ovviamente non esiste una traduzione italiana del libro, rendendolo me-

no accessibile. Esplorando l'offerta italiana invece, spicca *Game Design - Gioco e giocare tra teoria e progetto* di Maresa Bertolo e Ilaria Mariani: un libro prettamente didattico diviso in due parti, la prima legata all'analisi dei presupposti esclusivamente teorici e filosofici del gioco, mentre la seconda esplora l'ambito progettuale sempre da un punto di vista teorico; è un buon libro, ma rimane scollato dalla realtà e di sicuro non entra nel merito della produzione o nel campo imprenditoriale di settore.
Proprio per quanto proposto in alcuni di questi esempi, la necessità di creare un manuale che proponga una struttura contenutistica maggiormente legata all realtà produttiva risulta potenzialmente un unicum nel mercato; l'idea è quella di inserire sì elementi generali della teoria del game design, ma anche strutturare un percorso letterario simulativo del **mestiere**: quello a cui si punta di più, è proprio trovare i contenuti più importanti per creare esempi o simulazioni di progetti video-ludici reali o verosimili, in modo da creare vari standard di produzione. Il sapere come, cosa, dove, quando e con chi farlo all'interno del settore dei videogiochi è molto diverso dall'avere una sequenza di nozioni sul game design o una formazione in ambito della programmazione, proprio per questo il manuale vuole puntare maggiormente su una struttura a doppio binario per quanto riguarda il percorso del lettore. Il progetto in itinere propone de facto due "binari", uno legato ad una contenutistica più generalista sul game design e sulla produzione video-ludica (ovviamente maggiormente incentrato sulla realtà industriale italiana), mentre un secondo binario vuole concentrarsi sulla simulazione e sull'esempio: come è stato proposto in questa tesi vi sono simulazioni (si veda sezioni **3.3** o **3.7.12**) o esempi tratti da esperienze professionali reali (gli esempi virtuosi presenti nel capitolo **3** , sono testimonianze reali raccolte grazie alla collaborazione con IVProductions), che raccontano percorsi e metodi di lavoro funzionali. Con questa idea di progettazione, il manuale vuole tracciare un solco per gli sviluppatori/imprenditori video-ludici del futuro, creando un contatto tra teoria e pratica reale/verosimile.
Anche l'impaginazione avrà un ruolo fondamentale nella creazione di una meta-simulazione formata dal manuale stesso: l'idea è quella di scrivere il manuale come fosse un percorso step-by-step, che consenta al lettore di crescere a livello esperienziale, in maniera analoga alla simulazione dello sviluppo di un videogames.

4.2.2 Forma Comunicativa e Tone of Voice

Le modalità di scrittura del manuale vuole essere il meno didattica possibile, cercando di avvicinarsi ad un linguaggio accessibile e quasi dialogico con il lettore; si tratta comunque di un linguaggio tecnico e dedicato a chi è appassionato del mondo dei videogames o a chi già è integrato nel mercato del lavoro, il tutto senza renderlo inaccessibile anche ai profani, ma immancabilmente è una modalità di comunicazione dedicata ad un target più elevato

del semplice "lettore curioso". In particolare, si vuole stabilire un *tone of voice* quasi "amicale" con il lettore, creando una forma di comunicazione più leggera, anche se precisa e puntuale: la scrittura strutturata *Q&A* e la scrittura in prima persona, quasi il libro impersonasse un game designer e producer esperto, consentono al libro di essere molto più fruibile ed al tempo stesso, al netto dei contenuti, molto attendibile e preciso. Il progetto ad oggi ha fatto questa scelta, imponendola come un punto fisso di progettazione, anche se potrebbe essere presa un'altra posizione sul tipo di comunicazione nei restanti mesi di pre-produzione.

5 Conclusioni

Al fine di chiudere questo documento, si possono trarre alcune conclusioni riguardo al lavoro fatto: se la prima parte vuole essere un'analisi ed un'esplorazione del mondo dei videogames a tutto tondo, incarnando sia una base di lavoro per la redazione del manuale, sia un mini manuale generalista sul mondo dei videogiochi, la seconda presenta essenzialmente lo status del lavoro con IVProductions ed Ivan Venturi, che vede in questa tesi un punto di partenza, e introduce idee e concetti cardine per il proseguimento del progetto. Al netto dei propositi posti ad inizio tesi, si può affermare che l'obiettivo di questa tesi sia stato raggiunto, dato che ha soddisfatto i requisiti stabiliti all'inizio dei lavori assieme ad IVProductions: i materiali forniti e le sessioni private proposte dall'azienda, hanno garantito un flusso di dati ed informazioni utilissimo alla stesura di questo documento.

Bibliografia & Sitografia

- *La maggior parte del materiale è stata fornita da Ivan Venturi e IVProductions attraverso la concessione di conferenze, documentazione e lezioni accessibili in via esclusiva all'autore della tesi.*
- *I mestieri del videogioco: Le figure professionali di un mercato del lavoro in espansione*, AIV - Accademia Italiana Videogiochi, Dino Audino editore, 2019
- *Il videogioco - Storie, forme, linguaggi, generi: Per conoscere il nuovo medium che ha conquistato il mondo*, Lorenzo Mosna, Dino Audino editore, 2018
- *Psicologia dei videogiochi: Come i mondi virtuali influenzano mente e comportamento*, Stefano Triberti, Luca Argenton, Apogeo education, Maggioli Editore, 2019
- *Game Design: Gioco e giocare tra teoria e progetto*, Maresa Bertolo, Ilaria Mariani, Pearson, 2019
- *The Art Of Game Design: A book of lenses*, Jesse Schell,3rd Edition, CRC Press, 2019
- *Game Mechanics: Advanced Game Design*, Ernest Adams, Joris Dormans, New Riders, 2012
- it.mashable.com, *Perché c'è sempre più bisogno del game design femminile*, Jess Joho, 2020 (1)
- www.ilvideogioco.com, *IIDEA presenta il nuovo rapporto annuale sul mercato dei videogiochi in Italia*, Eduardo Ullo, 2020 (2)

- www.eurogamer.it,*Videogiochi in Italia: un mercato che nel 2019 vale 1,8 miliardi €, ha 17 milioni di videogiocatori e continua a crescere*, Alessandro Baravalle, 2020 (3)
- thenexttech.startupitalia.eu ,*Videogiochi, il mercato italiano cresce ancora un po': vale 1,8 miliardi*, Carlo Terzano, 2020 (4)
- www.digitalbros.com, *The Videogame Market A growing market*, 2020 (5)
- www.nutscomputergraphics.com, *The 9 largest independent videogames production companies in Italy*
- www.wired.it, *Grand Theft Auto V batte ogni record: ha guadagnato più di Star Wars e Avatar*, Lorenzo Fantoni, 2018 (6)

Ringraziamenti