



Laboratorio di Videogiochi

Dispensa 4
Lezione del 4/12/2014

A cura di
Leonora Cappellini
Tommaso Mongelli

<http://hmr.di.unipi.it/LabVG>



Fare a pezzi un gioco

Nel corso della prima lezione abbiamo affermato, come ricorderete, che *il game design è l'atto di decidere come funziona un gioco*.

Ma come funziona un gioco? Iniziamo a porci il problema di costruire un buon gioco partendo da zero.

Per farlo, dobbiamo prima di tutto “farlo a pezzi”, individuarne tutti i componenti che vanno a costituire quello che noi, comunemente, chiamiamo *gameplay*:

- i **giocatori**: elemento fondamentale senza il quale il gioco, ovviamente, non può funzionare. È necessario definire non solo se vogliamo che il gioco sia in *single* o in *multiplayer*, ma anche i rapporti che intercorrano tra i giocatori in questo caso (possono essere alleati, antagonisti, possono organizzarsi in squadre o agire singolarmente). Nei casi in cui poi sia previsto un ordine in cui i giocatori agiscono (come negli strategici a turni ad esempio), è necessario anche stabilire la **sequenza di gioco**;
- gli **NPC**: in un videogioco, spesso e volentieri, troviamo anche un buon numero di personaggi e *mob* controllati dal computer. Anche in questo caso è necessario definire i rapporti dei personaggi controllati da giocatori con gli NPC (*non-playing characters*), giacché questi ultimi possono essere alleati, ostili, ed in che misura debbano essere “intelligenti”;
- lo **scopo del gioco**: abbiamo già visto che non tutti i giochi hanno un obiettivo finale da raggiungere (ricordate i God Games, le sandbox...). Nei casi in cui questo ci sia, è necessario definirlo fin dall’inizio, così come è opportuno fissare eventuali obiettivi intermedi;
- le **regole**: niente di più, niente di meno, le regole definiscono come il gioco vada giocato, come gli elementi interagiscano tra di loro, come e quando il gioco finisca, le condizioni per la vittoria o la sconfitta;
- le **risorse**: si intendono gli oggetti che il giocatore può eventualmente raccogliere nel corso della sua azione per raggiungere l’obiettivo (sia che sia il suo personaggio in prima persona a farlo o che siano risorse raccolte da altre unità controllate dal giocatore);
- gli **stati di gioco**: immaginiamo di effettuare un salvataggio per interrompere una partita, ciò che viene salvata è una “fotografia” del gioco in un dato momento (giocatori in gioco, risorse disponibili, obiettivi raggiunti e da raggiungere...). Tutto ciò compone uno stato di gioco. E’ buona norma cercare di prevedere più stati di gioco possibile per allocare correttamente le risorse e definire obiettivi realisticamente raggiungibili;
- le **informazioni di gioco**: insieme agli stati di gioco, è opportuno definire le informazioni di cui il giocatore ha bisogno per avanzare da uno stato all’altro.

Meccaniche, dinamiche, estetica

Le parti di un gioco che abbiamo appena descritto interagiscono tra di loro, creando una combinazione unica che cambia se cambia anche una sola di esse. Il prodotto della loro interazione crea l'esperienza del giocatore, il *gameplay* appunto.

Nello scomporre un gioco per analizzarne le parti, ci avvarremo del **framework MDA (Mechanics, Dynamics, Aesthetics)**, descritto nel 2004 nel paper *MDA: A Formal Approach to Game Design and Game Research* da Robin Hunicke, Marc LeBlanc e Robert Zubek¹ e considerato un caposaldo nella ricerca legata ai videogiochi.

Meccaniche

Proviamo ad immaginare che un videogioco sia come un orologio, un insieme di ingranaggi meccanici che insieme danno vita ad un sistema complesso che non può funzionare se viene meno una delle sue parti. Gli ingranaggi altro non sono che le **meccaniche**, lo strato degli elementi atomici costitutivi del gioco: numero di giocatori, regole, metodi di controllo e così via.

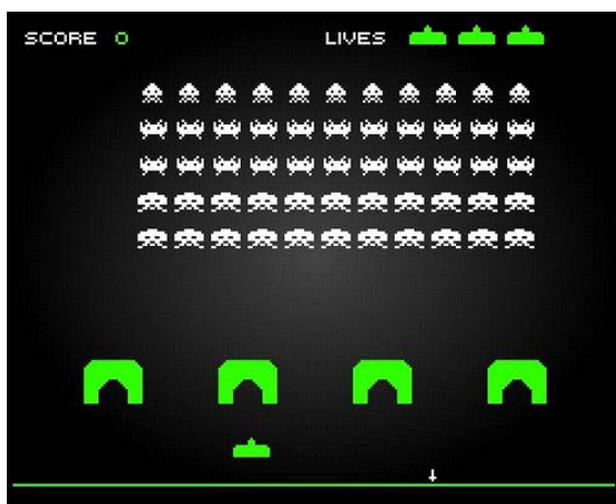
Normalmente è compito del programmatore codificare a basso livello le meccaniche in algoritmi che regolino il flusso di gioco e la risposta che esso da al giocatore in base alle sue azioni, ma il compito di definirle spetta normalmente al game designer.

Dinamiche

Quando le meccaniche vengono messe in movimento, danno vita alle cosiddette **dinamiche**. Le dinamiche descrivono *come si gioca*, il sistema di gioco, e spesso anche come il giocatore (ed eventuali avversari) si comporteranno per ottenere risultati ottimali.

Più il sistema di gioco è complesso ed ha regole, più presenterà dinamiche. Se il sistema è troppo complesso, le dinamiche di gioco potrebbero essere difficili da prevedere e comprendere per il giocatore.

Facciamo qualche esempio per comprendere meglio la relazione tra meccaniche e dinamiche.



Space Invaders

(per rivedere il gameplay su YouTube http://youtu.be/437Ld_rKM2s)

Il fatto che gli alieni si muovano tutti insieme e che possano essere uccisi, la possibilità di sparare, i ripari che si consumano quando colpiti dagli alieni, la velocità degli alieni che aumenta gradualmente, la regola che se il giocatore viene colpito muore sono tutte meccaniche di gioco.

Tipicamente, quando si gioca a Space Invaders, si cerca di colpire prima gli alieni che si trovano più in

¹ Per il paper originale si veda <http://www.cs.northwestern.edu/~hunicke/MDA.pdf>

basso, poiché il gioco termina quando nemico tocca il fondo dello schermo. Questa regola non sta scritta da nessuna parte, non c'è niente che ci costringa a colpire un alieno piuttosto che un altro: si tratta della risultante di alcune regole di base (movimento degli alieni e condizioni di sconfitta), quindi è una dinamica che si crea a partire da alcune meccaniche del gioco.

Pac-Man

(per rivedere il gameplay su YouTube http://youtu.be/i_OjztdQ8iw)

Anche qua abbiamo una serie di meccaniche che regolano il gioco: la condizione di vittoria (finire tutte le pillole dello schema), il fatto che i fantasmi inseguano il giocatore quando lo vedono, la possibilità di poter mangiare i fantasmi per un tempo limitato quando si prende una pillola grande (e che questi diventino anche più lenti), la condizione di sconfitta quando si viene presi da un fantasma.

La tipica strategia vincente in Pac-Man prevede che il giocatore cerchi prima di mangiare più pillole piccole possibile senza andare su quelle grandi fintanto che non si trovi in condizioni di pericolo (o che ci siano molti fantasmi vicini da mangiare per aumentare il punteggio): mangiando subito tutte le pillole grandi infatti si precluderebbe la possibilità di salvarsi quando realmente necessario. Anche in questo caso non esiste alcuna regola scritta che possa impedire al giocatore di mangiare subito tutte le pillole grandi, si tratta sempre di una dinamica originata dalla combinazione di alcune meccaniche.



Super Smash Bros. Melee

Esistono casi in cui certe dinamiche non previste dal designer del gioco vengono scoperte dal giocatore, che impara ad usarle piegando il sistema di gioco a proprio vantaggio. Questo è ciò che comunemente si chiama *exploit*.

Un caso di *exploit* piuttosto famoso è quello del *wavedash* in *Super Smash Bros. Melee* (Nintendo, 2001). Si tratta di un picchiaduro in cui i personaggi più famosi di proprietà della Nintendo combattono in vari scenari provenienti dalle rispettive serie di videogiochi (Mario, Zelda, Pokemon, Star Fox e così via). Il *wavedash* è un *exploit* che permette al giocatore di sfruttare la fisica del gioco per trasformare una schivata aerea in una scivolata (in avanti o in indietro) con la quale può allontanarsi rapidamente dall'avversario oppure attaccarlo senza permettergli di pararsi. Per un esempio pratico potete guardare http://youtu.be/N_eZoX8nAdg . Non trattandosi di un bug, ma di un una tecnica effettivamente permessa dalla fisica del gioco (anche se non prevista dagli sviluppatori), il *wavedash* è legale ed ampiamente utilizzato nel corso dei tornei ufficiali Nintendo.

Estetica

Per poter parlare di estetica, dobbiamo necessariamente parlare di divertimento, e prenderci la briga di capire cosa intendiamo quando ne parliamo.

È chiaro prima di tutto che il divertimento assoluto non esiste: diversi giochi offrono diverse esperienze che andranno a fare presa su diverse abilità di diverse tipologie di giocatori (se consideriamo poi il videogioco come opera d'intelletto, così come il cinema, la letteratura, la musica e via dicendo, è abbastanza facile poter affermare che niente è bello in modo assoluto). Se chiediamo a diversi tipi di giocatori cosa intendano per divertimento, possiamo ottenere risposte variegata, come *mi piacciono i giochi che mi tengono in tensione* o l'esatto contrario *io gioco per rilassarmi*.

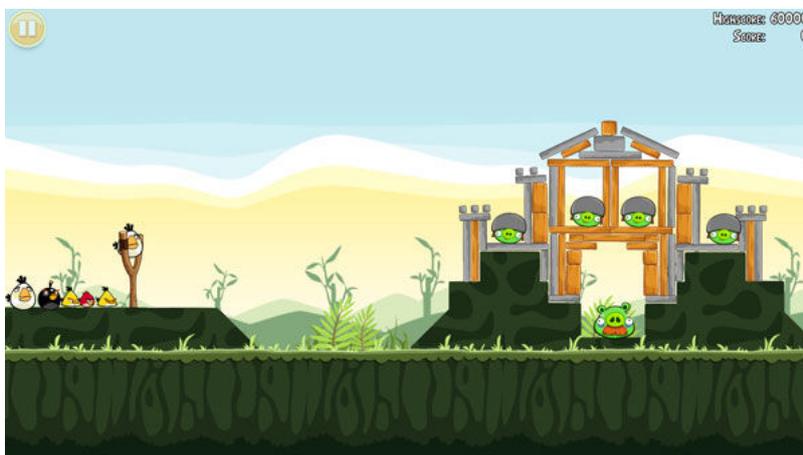
I giochi, di fatto, scatenano emozioni, ognuna legata ad un diverso modo di divertirsi. L'estetica di un gioco è l'insieme di emozioni che il designer intende suscitare attraverso di esso, e così come le dinamiche rappresentano le meccaniche in movimento, così *le estetiche vengono definite dalle dinamiche di gioco*.

Ogni gioco ha un *target*, una categoria di giocatori che mira a soddisfare, e che deve essere definito dal designer al momento del *concept*.

Alcuni esempi di estetiche:

- sfida: il videogiocatore avverte l'ebbrezza della competizione, come nei picchiaduro e giochi sportivi;
- narrativa: si gioca per il piacere di seguire una storia interessante, come nei giochi adventure (ed in tutti quelli che abbiano una storia da seguire);
- fantasia: il giocatore vive l'emozione di immergersi in un mondo fantastico, è un tratto tipico dei J-RPG;
- passatempo: il gioco è visto come fuga, generalmente breve, dalla realtà.

Potremmo fare tanti esempi quante sono le emozioni umane. Ovviamente è scontato che il design di un videogioco coinvolga più di una sola estetica e, talvolta, anche estetiche diametralmente opposte.



Un esempio che probabilmente è familiare a tutti voi è *Angry Birds* (Rovio), ci avete giocato? Si tratta di un gioco semplice a livello di meccaniche/dinamiche, ma ben articolato a livello di estetiche: certamente è un passatempo, i personaggi e gli ambienti carini e colorati richiamano un'estetica di fantasia, e non si può certo negare che non stimoli il senso della sfida!

Dallo sviluppatore al giocatore

I tre livelli che compongono il framework MDA possono essere considerati gli strati che si frappongono tra il lavoro dello sviluppatore e la fruizione del gioco da parte del giocatore. Immaginiamo uno schema di questo tipo²:



Se lo traduciamo utilizzando i termini che ci fornisce il modello otteniamo che



Risulta evidente che per il designer sia più immediato lavorare sulle meccaniche piuttosto che sul resto: non ha controllo diretto sulle dinamiche (come abbiamo visto nel caso degli exploit), ma può fare il possibile per pianificarle creando meccaniche adeguate ai suoi obiettivi. L'unico modo che il designer ha per verificare se il prodotto del suo lavoro rispecchi la sua idea è quello di far testare il gioco da giocatori del target opportuno.

Dal lato opposto, il giocatore è esposto immediatamente alle estetiche, sono la prima cosa che percepisce: solo giocando molto ad un gioco inizierà ad identificarne le dinamiche (ricordate quando, alla prima lezione, vi abbiamo chiesto di giocare o rigiocare ai giochi mostrati durante la lezione cercando di concentrarvi sulle meccaniche?).

Giocatore e designer vivono lo stesso videogioco da due punti di vista completamente diversi. Il designer è portato a guardare il gioco da una prospettiva *feature-driven* (concentrandosi cioè sulle caratteristiche del gioco stesso), rischiando di perdere quella *experience-driven* (vista cioè dal lato dell'esperienza del giocatore) e di creare un gioco divertente per il suo gusto personale, ma non per il target di giocatori che si era prefissato.

È di vitale importanza quindi, quando ci accingiamo a creare un gioco, calarsi sempre nei panni del giocatore e farsi le giuste domande, come *che tipo di esperienza voglio offrire? quali emozioni voglio che susciti il mio gioco?* Di conseguenza, ci chiederemo quali siano le dinamiche attraverso le quali possiamo veicolare tali esperienze ed emozioni, e quali regole e meccaniche sia necessario utilizzare.

² Le immagini, così come parte di questa dispensa fanno riferimento alla guida al Game Design di HTML.it <http://www.html.it/pag/46231/il-modello-mda-mechanics-dynamics-aesthetics>

Giochi basati sull'estetica: la Quantic Dream

Una azienda di videogiochi che basa il design dei propri prodotti su un approccio *experience-driven* è certamente la francese Quantic Dream, per la quale non a caso lavora una dei tre teorici del modello MDA.

La Quantic Dream ha prodotto solo quattro giochi a partire dal 1999: *The Nomad Soul* (per PC e Sega Dreamcast), *Fahrenheit* (per PC, Playstation 2 e X-Box), *Heavy Rain* (per Playstation 3) e *Beyond* (sempre per Playstation 3), tutti concepiti a partire dall'estetica.



Fahrenheit/Indigo Prophecy

Pubblicato da Atari per PC e PS2 nel 2005, Fahrenheit non riscosse al tempo il successo di vendite auspicato da David Cage (fondatore di Quantic Dream), che avrebbe avuto il suo riscatto, 5 anni dopo, con Heavy Rain.

Pur inquadrandosi pienamente nella categoria dei videogiochi d'avventura, Fahrenheit è un vero proprio film interattivo, come gli stessi sviluppatori della Quantic lo hanno definito, rifiutando qualsiasi altra etichetta di genere per il loro prodotto. Il giocatore controlla tre dei protagonisti della storia ed in base alle sue

scelte sono possibili tre finali differenti. Un fattore che deve essere costantemente tenuto in considerazione durante il gioco è lo stato d'ansia dei personaggi, che può migliorare o peggiorare in base alle scelte del giocatore e che può portare alla fine prematura del gioco. Il giocatore può fare scelte libere mirate a migliorare la salute psicologica dei personaggi, come fare una doccia per rilassarsi, ascoltare della musica, come in una sorta di simulatore di vita. L'ambientazione è profondamente inquietante, densa di elementi soprannaturali e ricca di colpi di scena. Il design fu concepito con l'obiettivo di portare il giocatore ad immedesimarsi nei personaggi ed a concentrarsi sull'evolvere della storia più che sulla loro sopravvivenza (cosa che lo distingue, ad esempio, da un altro caposaldo della narrativa videoludica horror quale è *Silent Hill*). Rispetto ad altre avventure grafiche, in Fahrenheit i personaggi non dispongono di un inventario, gli oggetti trovati nel corso della storia hanno un'utilità immediata e non ne è previsto l'uso col senno di poi. Il sistema di controllo è estremamente semplice ed intuitivo per non distrarre il giocatore dal gioco. Lo studio delle inquadrature è molto curato ed estremamente cinematografico.