POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria del Cinema e dei Mezzi di Comunicazione

Tesi di Laurea Magistrale

Dicembre 2025

Evoluzione del genere gestionale attraverso meccaniche action: un framework per l'integrazione di sistemi ibridi nell'ambito del game design

Relatore

MARCO MAZZAGLIA

Candidato

MATTIA SCAGLIOLA

Abstract

La ricerca seguente nasce dall'osservazione di come il panorama videoludico stia evolvendo verso forme sempre più ibride, dove i confini tradizionali tra i generi sfumano per creare esperienze di gioco uniche e innovative. In particolare, l'integrazione di meccaniche action all'interno di contesti gestionali (e viceversa) rappresenta una pratica molto diffusa all'interno delle opere videoludiche, ma ancora poco esplorata dal punto di vista metodologico.
La tesi mira ad esplorare l'integrazione dei due generi, proponendo un framework metodologico che supporti gli sviluppatori nella progettazione, bilanciamento e validazione empirica dei propri design.
Attraverso un'analisi dello stato dell'arte, che include lo studio dell'evoluzione storica dei generi e di case studies specifici, si identificheranno pattern ricorrenti e principi chiave del genere. Il framework proposto fornirà quindi linee guida pratiche, principi chiave, pattern e tecniche di design per lo sviluppo di giochi ibridi action-gestionali. Oltre a ciò, verranno forniti gli strumenti matematici volti a supportare lo sviluppo, il bilanciamento e la valutazione del proprio design.
Infine, la validazione del framework verrà effettuata su analisi dati specifiche e in particolare sulla raccolta dati di un progetto videoludico sviluppato ad hoc.
Questa ricerca mira a contribuire sia alla letteratura accademica sul game design sia alla pratica dello sviluppo di videogiochi, fornendo strumenti utili per sviluppatori, ricercatori e appassionati del settore.

Indice

1. Introduzione

1.1 Contesto, motivazioni e obiettivi della ricerca

Dall'ascesa del mondo videoludico negli anni '70 agli anni '20 del nuovo millennio si è osservata la nascita di numerosi generi videoludici nati dalla combinazione di generi originariamente separati. Tra questi, i generi action e gestionali hanno visto una progressiva ibridazione che ha portato alla creazione di esperienze sempre più complesse e articolate, spesso a partire dagli esperimenti di singoli sviluppatori. Considerata la rilevanza culturale ed economica del genere, nonché l'interesse personale dell'autore, questa ricerca si propone di elaborare un framework metodologico che:

- Permetta di definire meglio le caratteristiche dei generi action, gestionale e le loro combinazioni;

- Definisca un workflow che permetta agli sviluppatori di anticipare meglio l'impatto delle scelte prese in fase di design;

- Definisca metriche specifiche per il genere, utili a valutare in modo quantitativo le scelte di design e analizzare le previsioni formulate in fase progettuale;

- Evidenzi criticità e pattern ricorrenti, consentendo di intervenirvi preventivamente e di ottimizzare le risorse impiegate nello sviluppo.

1.2 Modello di analisi

In questa sezione verranno introdotti i metodi di analisi utilizzati sia per l'analisi propedeutica dei generi analizzati, sia come base per la creazione del framework.
Si è scelto di utilizzare il modello MDA (Mechanics–Dynamics–Aesthetics) [1] perché particolarmente consolidato all'interno dell'ambito videoludico, in combinazione con la scomposizione di Fullerton in elementi formali e drammatici [2] che sottolinea in maniera concreta quali siano gli elementi che compongono un'opera videoludica. Verrà

anche citato il sistema creato da Walter Nuccio relativo ai design pattern all'interno dei giochi da tavolo [3] che, pur nascendo in ambito non digitale offre numerosi spunti per la creazione del framework e l'analisi delle criticità.

1.2.1 Il modello MDA (Mechanics, Dynamics, Aesthetics)

Il framework MDA, proposto da Hunicke, LeBlanc e Zubek nei primi anni 2000, è un approccio formale per descrivere e analizzare i giochi [1]. Esso nasce con l'obiettivo di colmare il divario tra le diverse dimensioni del game design: progettazione, sviluppo, critica e ricerca tecnica, fornendo un linguaggio comune per collegare la struttura di un gioco con l'esperienza del giocatore [1]. In MDA ogni gioco è scomposto in tre livelli interconnessi: meccaniche, dinamiche e estetiche.

Dal punto di vista del game designer, i tre livelli sono legati da una relazione causale: le meccaniche implementate generano specifiche dinamiche di gioco, le quali a loro volta producono particolari estetiche (esperienze) nel giocatore. Tuttavia, il giocatore percepisce il processo in senso inverso: egli sperimenta prima l'estetica (le sensazioni e il divertimento offerti dal gioco), che deriva da determinate dinamiche osservabili durante la partita, e, infine, dalle meccaniche sottostanti [1].

Meccaniche, procedure e regole

Le meccaniche rappresentano i componenti fondamentali del gioco: dati, regole e algoritmi che definiscono come il gioco opera a livello base.

In stretta correlazione con il concetto di meccaniche vi è il concetto di procedure e regole, come approfondito da Tracy Fullerton in Game Design Workshop [2].

Le procedure sono infatti i metodi di gioco, ovvero l'insieme delle azioni possibili che i giocatori possono intraprendere, e in quali condizioni, per raggiungere gli obiettivi. Le procedure rappresentano l'implementazione concreta delle meccaniche di gioco.

Le regole definiscono il mondo di gioco e delimitano ciò che è permesso o proibito ai giocatori. Insieme alle procedure, le regole determinano lo spazio delle possibilità entro cui i giocatori possono agire.

Esempi di meccaniche, procedure e regole all'interno di un gioco sono:

- Regole di movimento;
- Azioni disponibili;
- Condizioni di punteggio.

Dinamiche

Le dinamiche rappresentano il comportamento del gioco in esecuzione, che emerge quando le meccaniche vengono messe in moto dal giocatore e si influenzano reciprocamente nel tempo [1]. Esse includono, ad esempio, i pattern di gioco che si sviluppano (e.g. strategie, interazione tra giocatori, equilibrio tra cooperazione e competizione, adattamento alla difficoltà).

Estetiche

Le estetiche indicano le esperienze emotive e gli obiettivi ludici che il giocatore prova durante il gioco [1]. Si tratta delle reazioni desiderate (e.g. divertimento, tensione, sfida, scoperta, solidarietà) che il designer intende suscitare. L'estetica riguarda quindi il motivo per cui il gioco è interessante per il giocatore, ossia quale tipo di "divertimento" o soddisfazione offre.

Questa visione si ricollega strettamente al concetto di player experience goals definito da Fullerton [2], ovvero gli obiettivi che il game designer si pone per il tipo di esperienza che vuole che il giocatore sperimenti mentre gioca. Alcuni esempi di estetiche sono:

- "Il giocatore sperimenterà un senso forte di tensione mentre esplora le caverne in cui il gioco è ambientato";
- "I giocatori saranno incentivati a pensare in maniera creativa per superare i livelli del gioco";

Fondamentale nel framework MDA è l'idea che i videogiochi "sono più simili ad artefatti che a media" [1]. Ciò significa che il contenuto di un gioco non risiede semplicemente in elementi audiovisivi o narrativi lineari, bensì nel comportamento interattivo che emerge dalle sue regole durante il gameplay. In altri termini, "Le meccaniche sono il messaggio" [4]: le interazioni presenti all'interno di un gioco veicolano direttamente significati, valori ed emozioni.

1.2.2 Elementi formali

Parallelamente al modello MDA, Tracy Fullerton, identifica una serie di elementi che costituiscono la struttura di ogni gioco [2]. Senza questi elementi, un'attività non può essere considerata un vero "gioco". Di seguito vengono riportati in maniera riassuntiva i cinque elementi che sono stati considerati utili nell'ambito di questa tesi:

Giocatori

I partecipanti che, volontariamente, prendono parte al gioco, entrando in una condizione artificiale con regole e limiti definiti. Nel design occorre considerare quanti giocatori possono partecipare, quali ruoli possono assumere e la natura delle interazioni tra di essi.

Obiettivi

I traguardi che i giocatori devono raggiungere all'interno del gioco e definiscono quindi lo scopo del gioco, cioè ciò che i giocatori intendono ottenere. È fondamentale che gli obiettivi garantiscano una sfida adeguata, pur essendo raggiungibili, poiché determinano il tono dell'esperienza e influenzano anche gli elementi narrativi o tematici.

Risorse

Gli elementi limitati che i giocatori possono acquisire, utilizzare o gestire nel corso del gioco.

Le risorse hanno due aspetti che le contraddistinguono:

- Utilità: servono a qualcosa nell'economia del gioco;

- Scarsità: non sono infinite o facilmente accessibili.

I videogiochi prevedono meccanismi per ottenere, spendere, scambiare o convertire le risorse, e il loro equilibrio influisce notevolmente sull'esperienza di gioco.

In seguito, si riprenderà più volte il concetto di risorse nella definizione del Framework poiché risulterà fondamentale.

Conflitto

Il conflitto emerge dai confini e le regole del gioco che impediscono ai giocatori di raggiungere immediatamente i loro obiettivi. Si osserva che il conflitto è solitamente progettato in modo da non permettere ai giocatori di raggiungere i loro obiettivi in maniera lineare o banale, ma li costringe a confrontarsi con sfide che richiedono abilità e strategia. Il conflitto è di fatto ciò che genera la sfida.

Esito

Il risultato finale del gioco, che può concretizzarsi in termini di vittoria, sconfitta, punteggio o stato finale raggiunto. Affinché un gioco mantenga alta l'attenzione dei partecipanti, l'esito dovrebbe rimanere incerto fino alla fine. Se il risultato è prevedibile o deciso troppo presto, infatti, la motivazione del giocatore ne risente, poiché risulta superfluo proseguire la partita. Questo aspetto è importante da tenere a mente perché importante nel raggiungimento dello stato di flow.

Si nota come gli elementi sopra descritti si colleghino ai livelli del modello MDA discusso in precedenza:

- Le meccaniche si fondano su regole, procedure e gestione delle risorse;

- Queste, interagendo durante il gioco, generano dinamiche di gioco (sfide, competizioni, pattern di gioco) e determinano l'andamento del gioco fino a un

certo esito finale;

- Infine, le dinamiche producono le estetiche, che definiscono l'esperienza del giocatore.

1.2.3 Lo stato di flow

Di seguito si evidenzia aspetto fondamentale dell'esperienza videoludica che rappresenta un obiettivo cardine del game designer durante la creazione di un videogioco nei confronti del giocatore: lo stato di flow. Il concetto di flow, introdotto dallo psicologo Mihály Csíkszentmihályi, è la condizione psicologica di totale coinvolgimento in un'attività che viene sperimentata da un soggetto, che risulta completamente immerso e concentrato in essa in uno stato di performance ottimale e al contempo di profondo appagamento [5].

Di seguito vengono presentati in maniera riassuntiva gli otto aspetti principali che caratterizzano questa particolare condizione:

Equilibrio tra sfida e abilità

L'attività deve risultare sufficientemente stimolante ma compatibile con le abilità del giocatore.

Fusione tra azione e consapevolezza

Le azioni del soggetto avvengono in modo spontaneo, quasi automatico.

Obiettivi chiari e feedback immediati

Il soggetto ha chiaro nella sua mente cosa vuole ottenere con le sue azioni e riceve feedback costanti sull'andamento dell'attività.

Focalizzazione totale sul compito attuale

Il soggetto è consapevole solo dell'attività che sta effettuando e pienamente concentrato sul momento presente.

Perdita di autocoscienza

La persona sperimenta un indebolimento o annullamento del senso del sé, che ritorna più forte una volta conclusa l'esperienza.

Il paradosso del controllo

Il soggetto sperimenta un senso di controllo sulla situazione, ma questo avviene paradossalmente proprio perché l'esito dell'attività è incerto, altrimenti non ci sarebbe nulla su cui esercitare il controllo.

Distorsione del tempo

Il tempo sembra passare in maniera distorta durante l'esperienza: più velocemente o, più raramente, in maniera più lenta.

Esperienza autotelica

L'esperienza diventa fine a sé stessa e il soggetto non ha altro motivo di provare appagamento se non dalla sperimentazione dell'attività in quanto tale.

Questa particolare condizione psicologica, sperimentabile in qualsiasi attività e non solo in ambito videoludico, risulta fondamentale poiché gli otto elementi sopra presentati permettono di capire e analizzare meglio molte delle scelte di game design che verranno presentate in seguito.

1.2.4 Elementi drammatici

Oltre agli elementi formali, rivestono grande importanza gli elementi drammatici, ossia quei fattori che coinvolgono il giocatore sul piano narrativo, emotivo e di immedesimazione [2].

Sfida

La sfida rappresenta il cuore del coinvolgimento ludico: è generata dai conflitti e dagli ostacoli che il giocatore deve affrontare per raggiungere gli obiettivi del gioco.

Il suo ruolo è fondamentale perché necessario al raggiungimento dello stato di flow da parte del giocatore,

Affinché la sfida risulti efficace e motivante, essa deve essere equilibrata: sufficientemente impegnativa da evitare la noia, ma al contempo raggiungibile per non sfociare in frustrazione.

Sfide eccessivamente semplici producono noia, mentre compiti troppo ardui generano ansia e frustrazione e portano all'abbandono.

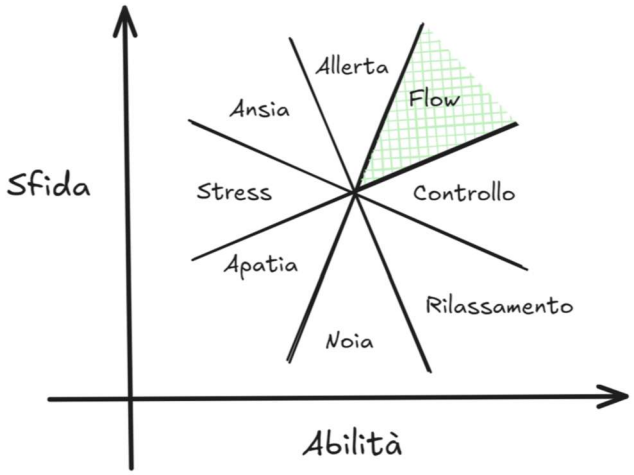


Figura 1: Diagramma dello stato di flow al variare di abilità del giocatore e sfida del gioco [5].

Play

Con il termine play, ci si riferisce alla libertà di azione e di espressione che un gioco consente all'interno della sua struttura formale. Un modo utile per descrivere il play è considerarlo come il "movimento libero all'interno di una struttura rigida" [2] una definizione che coglie il paradosso per cui il giocatore è vincolato da regole fisse, ma al contempo deve poter sperimentare, esplorare strategie e compiere scelte significative.

Dal punto di vista del flow, il play è fondamentale all'instaurazione dello stato: la libertà di azione e la possibilità di incidere attivamente sul mondo di gioco conferiscono al giocatore un forte senso di controllo, che è come precedentemente menzionato è uno

dei fattori chiave. Inoltre, la varietà di approcci offerta dal play aiuta a mantenere l'esperienza rigiocabile e stimolante.

Elementi narrativi

Fullerton introduce tra gli elementi drammatici anche i seguenti:

- Premessa;
- Storia;
- Personaggi;
- World building.

Premessa e storia vanno a introdurre una cornice narrativa alle attività di gioco, fornendo contesto e significato alle meccaniche di gameplay. Senza questi elementi molti giochi risulterebbero esperienze troppo astratte e poco coinvolgenti sul piano emotivo.

Il world building, che si riferisce alla costruzione del mondo di gioco dal punto di vista narrativo (regole fisiche, sociali, storia e atmosfera genera dell'ambiente) va invece a creare ambienti ricchi di profondità e significato.

Questi elementi, assieme alla creazione di personaggi ricchi di carattere e con motivazioni specifiche, aumentano l'immedesimazione all'interno del gioco, andando a fornire un contesto e motivazioni ben definite alle azioni del giocatore, ai conflitti e alle sfide che deve superare. Questi aspetti contribuiscono all'investimento emotivo del giocatore e a mantenere alta la curiosità e la concentrazione, oltre che a fornire feedback narrativi per le sue azioni (per esempio storia ed eventi che cambiano in base ai suoi successi e fallimenti), andando a contribuire e a sostenere lo stato di flow.

1.2.5 Dimensioni di gioco

Le dimensioni di gioco introdotte da Walter Nuccio [3] rappresentano diverse aree di analisi applicabili ai giochi da tavolo. Sebbene l'analisi nasca in ambito non digitale, verrà analizzato in seguito come queste possono essere applicate a prodotti digitali.

Nell'ambito di questa tesi si è deciso di selezionare quattro delle dimensioni di giochi da approfondire maggiormente poiché significative nell'ambito dei giochi action-gestionali.

Memoria

Viene definita "Memoria" l'insieme dei fattori che legano ogni istante di gioco al successivo. Grazie ad essa le scelte fatte continuano ad avere un'influenza sul resto della partita [3].

Un esempio di gioco con forte memoria sono gli Scacchi: ciascuna azione influenza il resto della partita dall'inizio alla fine in maniera diretta e irreversibile.

Controllabilità

Viene definita "Controllabilità" la misura in cui il giocatore può influenzare l'andamento e l'esito della partita attraverso le proprie scelte.

Un esempio di gioco con scarso fattore di controllabilità è il Gioco dell'Oca: in esso il giocatore avanza sul tabellone in base al valore dei dadi lanciati, quindi in maniera totalmente casuale [3].

Tensione

La tensione è la sensazione che nasce dalla percezione di una difficoltà, una competizione o un rischio [3].

Il concetto di Tensione è direttamente collegato al concetto di sfida e quindi all'instaurarsi dello stato di flow.

Stabilità

Rappresenta la tendenza al mantenimento di un equilibrio tra i giocatori che hanno conquistato un vantaggio (i leader della partita) e quelli che invece sono rimasti indietro (i loser della partita); in modo che questi ultimi possano recuperare sui primi [3].

Il nome, in questo modo, potrebbe essere fuorviante: un gioco risulta stabile se a risultare “stabile” è la possibilità dei giocatori di passare da leader a loser e viceversa.

In un gioco instabile si possono avere problemi di Runaway Leader e Difficoltà di Rimonta, che verranno descritti meglio nel framework.

2. Evoluzione del genere action - gestionale

2.1 Introduzione ai generi fondamentali

2.1.1 Il genere action: caratteristiche distintive

Il genere *action* (azione) nei videogiochi è tipicamente definito come quel tipo di gioco che enfatizza le sfide fisiche e la prontezza di riflessi da parte del giocatore [6].

In pratica i giochi appartenenti a questo genere richiedono una forte coordinazione mano-occhio, tempi di reazione rapidi e abilità motorie per superare sfide di varia natura in tempo reale [6].

Rientrano all'interno di questo genere una gamma vasta di sottogeneri, tra cui principalmente:

- Picchiaduro: giochi di lotta tra due giocatori, uno contro l'altro. Raramente a più giocatori;
- Beat'em up: giochi di combattimento contro orde di nemici;
- Sparatutto: giochi di combattimento con armi da fuoco;
- Platform: giochi di agilità con ostacoli fisici e ambientali;
- Rhythm game: giochi musicali in cui è necessario riprodurre o seguire il ritmo di un brano.

Ciò che accomuna tutti questi sottogeneri è il ritmo generalmente veloce e l'interazione in tempo reale: il giocatore ha il controllo diretto di un personaggio o di un'entità e deve eseguire azioni specifiche, in un ambiente o in un contesto dove gli errori di tempismo

possono risultare in una sconfitta. Questa enfasi sull'azione tempo-dipendente rende il genere action uno dei più immediati e adrenalinici.

È interessante notare come il genere action, a differenza di altri, sia trasversale a tutte le piattaforme. Dopo essere nato sui coin-op arcade e sulle prime console casalinghe, si è evoluto su PC e console moderne incluse quelle portatili, trovando spazio in particolare nel mondo mobile dove i controlli touch permettono l'implementazione di comandi semplici e intuitivi.

Quel che resta invariato tra le diverse piattaforme è il fato che un titolo action punta a offrire un'esperienza di gioco dinamica in cui l'abilità del giocatore nel prendere decisioni rapide e nell'eseguire comandi con precisione è messa costantemente alla prova [7].

Popolarità e mercato del genere action

Il genere action occupa storicamente una posizione di primo piano nel mercato videoludico mondiale, sia in termini di vendite sia di numero di giocatori.
Recenti statistiche affermano che i giochi d'azione e i loro sottogeneri sono tra i più giocati in assoluto. Ad esempio, in un sondaggio globale del 2022, i giochi sparatutto risultavano i più diffusi con il 57% dei giocatori che ne aveva giocato almeno uno nell'ultimo anno [8].
Subito a seguire vi erano i giochi di azione/avventura (caratterizzati da una forte componente narrativa) con il 54% dei giocatori coinvolti nell'ultimo anno [8].

Ciò significa che oltre la metà dei giocatori mondiali ha giocato recentemente titoli riconducibili al genere action. Anche limitando l'analisi a mercati specifici, come quello nordamericano, la tendenza rimane simile: secondo il rapporto dell'ESA (Entertainment Software Association) nel 2023 circa il 41% dei videogiocatori negli USA giocava regolarmente a titoli di genere action, classificandolo tra i cinque genere più popolari assieme a puzzle, arcade e giochi d'azzardo [9].

Dal punto di vista economico, il peso di mercato del genere è a sua volta significativo. Basti pensare che il segmento dei giochi mobile di azione a livello mondiale ha generato

ricavi stimati attorno ai 19 miliardi di dollari nel 2022 [10], a testimonianza della grande domanda di esperienze action. Il segmento ha generato nel 2024 18.7 miliardi di dollari, confermando comunque una domanda elevata dopo il picco pandemico [11].

L'action non è quindi solo un genere storicamente rilevante nella storia dei videogiochi, ma rimane attualmente uno dei generi più diffusi dell'industria, con un mercato che abbraccia diverse fasce d'età e aree geografiche.

2.1.2 Analisi del genere action puro

Come evidenziato nel capitolo precedente, una delle caratteristiche principali dei giochi action è l'enfasi su sfide di natura fisica in tempo reale. Di seguito viene fornita un'analisi degli elementi cardine del genere.

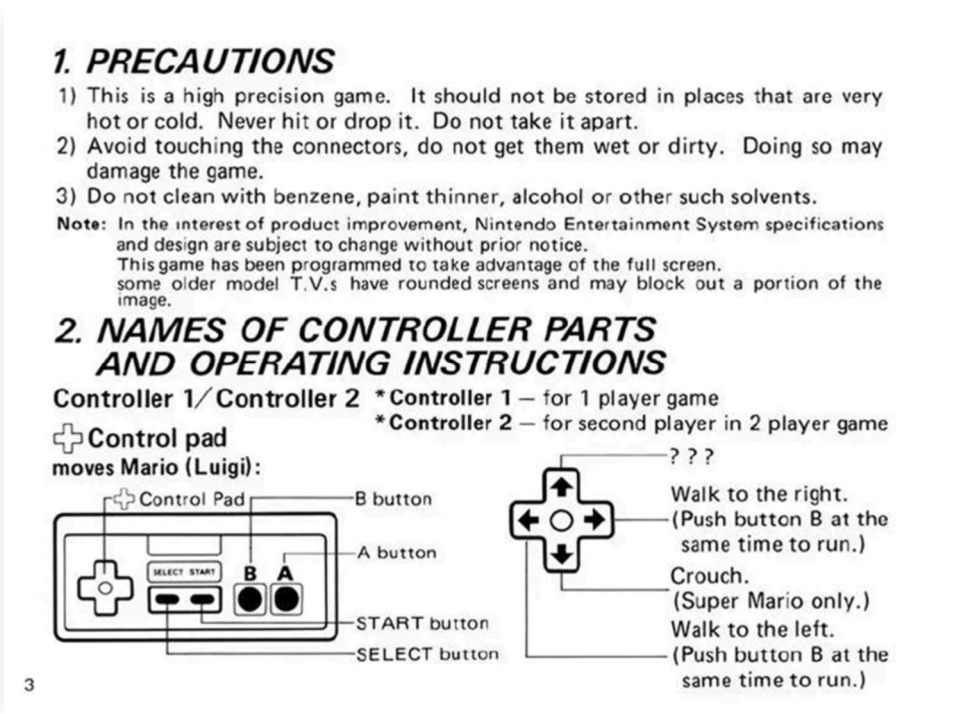
Movimento e controllo del personaggio

Nella maggior parte dei videogiochi action, l'esperienza di gioco si basa sul controllo diretto di un avatar o personaggio. È quindi naturale che molte delle meccaniche, procedure e regole di gioco vadano a definire in maniera precisa:

- Azioni eseguibili (e.g. salto, corsa, schivata, attacchi);
- Quando queste azioni possono essere compiute (e.g. se il giocatore deve essere in uno stato particolare o se è necessaria una risorsa);
- Come queste azioni possono essere compiute (attraverso le procedure che definiscono i comandi);
- Il comportamento effettivo di queste azioni (come la fisica, la traiettoria e le interazioni con il resto dell'ambiente di gioco);

Definire in maniera esatta questi aspetti risulta importante perché oltre ad influenzare l'esperienza, va a determinare il design e il bilanciamento dei livelli di gioco.

Nel platform 2D Super Mario Bros [12], per esempio, il salto è la meccanica fondante. Premendo il pulsante A del controller, Mario salta, raggiungendo 4 blocchi di altezza massima. Il giocatore può inoltre correre premendo il pulsante B del controller, Mario corre, accelerando gradualmente fino alla velocità massima e permettendo al giocatore di saltare fino a 5 blocchi di altezza massima. Se nel corso dello sviluppo si decidesse di cambiare questi valori, o rimuovere per esempio l'azione di corsa, si dovrebbero rivedere tutti i livelli precedentemente costruiti, ed è quindi fondamentale che questi siano definiti agli inizi dello sviluppo.



1. PRECAUTIONS

1) This is a high precision game. It should not be stored in places that are very hot or cold. Never hit or drop it. Do not take it apart.
2) Avoid touching the connectors, do not get them wet or dirty. Doing so may damage the game.
3) Do not clean with benzene, paint thinner, alcohol or other such solvents.

Note: In the interest of product improvement, Nintendo Entertainment System specifications and design are subject to change without prior notice.
This game has been programmed to take advantage of the full screen.
some older model T.V.s have rounded screens and may block out a portion of the image.

2. NAMES OF CONTROLLER PARTS AND OPERATING INSTRUCTIONS

Controller 1/Controller 2 *Controller 1 — for 1 player game
*Controller 2 — for second player in 2 player game

Control pad
moves Mario (Luigi):

3

Figura 2: schema controlli di Super Mario Bros [12] estratto dal manuale di istruzioni originale.

Risorse

Ogni gioco action include al suo interno specifiche risorse. Tuttavia, molte delle risorse presenti sono riconducibili a queste quattro:

- Numero di vite: determina il numero di sconfitte che il giocatore può subire prima di perdere definitivamente;

- Salute: quantifica il numero di errori che il giocatore può compiere prima di perdere una vita;

- Stamina: limita le azioni del giocatore quando è esaurita;

- Tempo: induce pressione temporale al giocatore.

In *Super Mario Bros [12]* il giocatore inizia con tre vite. Alla perdita di una vita, la ripartenza avviene dall'inizio dell'area oppure (se si è superata circa la metà) da un punto intermedio non segnalato. In caso di Game Over (quando si esauriscono le vite a disposizione), il gioco torna alla schermata iniziale; utilizzando il codice A + Start è possibile continuare dall'inizio dell'ultimo mondo.

Tipologia e comportamento degli ostacoli

Nella quasi totalità dei giochi action, conflitto e sfida sono creati dalla presenza di ostacoli che si oppongono attivamente al giocatore. Anche in questo caso vengono solitamente definiti, per ciascuno ostacolo:

- Tipologia (e.g. nemici e creature o elementi ambientali);

- Comportamento (e.g. se ha un'intelligenza artificiale o un pattern specifico di movimento);

- Come il giocatore deve affrontarlo (e.g. se è costretto a evitarlo o devono essere usate alcune azioni per superarli);

- Cosa succede se il giocatore lo affronta correttamente (e.g. se riceve una ricompensa);

- Cosa succede se il giocatore non lo affronta correttamente (e.g. se gli viene sottratta qualche risorsa);

- Come gli ostacoli vengono generati (e.g. se sono fissi oppure vengono generati a runtime secondo una logica specifica).

Ad esempio, in Contra [13], sono presenti diverse tipologie di soldati che ostacolano il giocatore. I soldati più semplici appaiono in un punto specifico dello schermo e corrono in una certa direzione finché non escono dallo schermo. Il giocatore può affrontarli semplicemente usando l'azione di salto ed evitandoli; tuttavia, può anche sparare con la propria arma e un solo colpo è sufficiente a eliminarli. Altri nemici invece stanno fermi, sparando nella direzione del giocatore, e il giocatore è costretto ad usare l'azione di sparo per affrontarli. In ogni caso, se il giocatore tocca un proiettile o un nemico, viene eliminato, perde una vita e ricomincia dal lato sinistro dello schermo.

In molti sparatutto (come Call of Duty [14]) il giocatore deve fermarsi a ricaricare se finisce le munizioni dell'arma (che possono essere considerate una forma di stamina assieme al tempo massimo di corsa del giocatore prima di fermarsi).

Anche in rhythm game come Guitar Hero [15] il numero di note che il giocatore può sbagliare prima di dover ricominciare può essere considerata una forma di salute.

Pressione temporale e frenesia

Una delle dinamiche che emerge maggiormente all'interno dei giochi action sono la pressione temporale e la frenesia [1], il cui obiettivo è quello di creare un'esperienza di gioco adrenalinica, rapida e che metta alla prova i riflessi e le abilità fisiche del giocatore.

Queste sono solitamente generate o da un qualche tipo di risorsa presente nel gioco all'esaurirsi della quale il giocatore perde (come nel caso dello scadere del tempo), così come dalle meccaniche relative alla generazione degli ostacoli e dal loro comportamento. All'interno di molti giochi, infatti, gli ostacoli vengono creati con un ritmo serrato e impediscono al giocatore di rimanere fermo senza venire sconfitto (e.g. in Contra [13]).

Interazioni tra giocatori

I giochi action hanno da sempre integrato al loro interno una varietà di pattern di interazioni tra giocatori estremamente ampia. In origine questo avveniva in locale e con un numero di giocatori limitato (per esempio sui coin-op arcade attraverso la presenza di due giocatori contemporanei), ma l'evoluzione tecnologica ha fatto sì che i giochi di questo genere si evolvessero in modo da supportare un gran numero di giocatori connessi in remoto e l'inclusione di pattern di interazioni multiple.

Questo fa sì che lo stesso gioco, a parità di meccaniche di controllo e risorse, produca dinamiche ed estetiche notevolmente diverse in base alla modalità di gioco scelta, incentivando ad esempio la cooperazione e lo scambio di informazioni e risorse tra giocatori o viceversa, rendendo il gioco maggiormente competitivo.

In Call of Duty [14], per esempio, è presente la modalità multiplayer, che integra diverse opzioni di gioco, tra cui la variante "Deathmatch a Squadre" e "Tutti contro Tutti". Secondo lo schema proposto da Fullerton queste due modalità appartengono rispettivamente ai pattern di interazione "Competizione a Squadre" e "Competizione Multilaterale" [2]. L'esperienza di gioco cambia drasticamente tra le due, poiché nel primo caso è incentivata la cooperazione tra giocatori, mentre nella seconda il giocatore è da solo ed è costretto a rimanere costantemente attento all'arrivo di giocatori da qualunque direzione.

Esito e Obiettivi

All'interno dei giochi action l'esito è solitamente binario: si vince (nel caso venga raggiunto il proprio obiettivo) o perde in base all'accumularsi degli errori (solitamente definito, come precedentemente descritto, dall'esaurirsi di qualche risorsa come, per esempio, la salute e le vite). Tuttavia, fin dagli albori del genere viene solitamente combinato all'obiettivo principale (che può essere per esempio raggiungere una determinata meta o sconfiggere un determinato numero di nemici) un obiettivo prestazionale (per esempio un punteggio che si accumula) che a parità di obiettivo raggiunto misura le abilità del giocatore rispetto ai tentativi precedenti di sé e degli altri. In questo modo il giocatore è incentivato a usare sempre il massimo delle proprie

abilità, favorendo la condizione di flow anche quando le abilità del giocatore superano la difficoltà dell'obiettivo principale.

In Armored Core 6: Fires of Rubicon [16], ad esempio, il giocatore deve completare i livelli della storia principale completando in ognuno uno o più obiettivi come il sopravvivere per un tempo sufficiente o sconfiggere un particolare avversario. Tuttavia, dopo aver completato un livello, il giocatore può rigiocarlo, ottenendo in ciascuno una valutazione da un minimo di D a un massimo di S sulla base di indicatori di performance (tra cui danni subiti e numero di munizioni usate a seconda della missione). Questo aumenta la longevità del gioco e spinge il giocatore a trovare le combinazioni e strategie più adatte all'interno di ogni livello.

Elementi narrativi

Nonostante il genere action si concentri principalmente su un ritmo frenetico e sulle abilità fisiche, gli elementi narrativi risultano comunque fondamentali a dare significato e contesto alle azioni del giocatore e alle meccaniche di gioco. Se le meccaniche di gioco e gli ostacoli creano l'azione in cui il giocatore è immerso, la premessa, la storia e il world building vanno a creare perché l'azione avviene e danno motivo al giocatore di prenderne parte. In alcuni giochi è sufficiente una breve premessa, altri giochi sviluppano maggiormente la storia vera e propria. In ogni caso, ciò che è comune tra i giochi di questo genere, è che la narrazione di solito non va a spezzare lo svolgersi dell'azione. Per questo è solitamente narrata attraverso filmati (cutscene) posizionati in punti strategici o dialoghi tra i personaggi, che potendo essere ascoltati attraverso l'udito non richiedono al giocatore di interrompere le proprie azioni per leggere.

Sempre in Armored Core 6: Fires of Rubicon [16], per esempio, la narrazione viene effettuata attraverso "briefing" introduttivi a ciascuna missione, prima che il giocatore si ritrovi immerso nell'azione vera e propria, dove vengono spiegati al giocatore gli obiettivi e lo scopo di quello che sta andando a compiere. Attraverso essi, agli obiettivi effettivi che il giocatore dovrà portare a termine viene anche narrata la storia. Tuttavia, durante la partita sono anche presenti dialoghi tra i personaggi che attraverso l'ascolto

forniscono al giocatore ulteriori dettagli sul mondo di gioco, senza tuttavia interrompere il ritmo dell'azione.

Figura 3: estratto del briefing della missione "Attack the Dam Complex" di Armored Core 6: Fires of Rubicon [16].

2.1.3 Il genere gestionale: caratteristiche distintive

"Un gioco è una serie di scelte significative" [17]. Non è un caso che questa definizione di "gioco" provenga dall'autore di una delle più emblematiche saghe appartenenti al genere gestionale, ovvero Civilization [18]. In effetti sebbene questa definizione possa essere in linea teorica estesa a tutti i videogiochi, è particolarmente adatta a descrivere questo genere.

Esso comprende infatti quei titoli il cui fulcro dell'esperienza è la pianificazione strategica e la gestione di risorse e sistemi complessi, piuttosto che i riflessi o le abilità motorie del giocatore [7]. In queste opere il giocatore assume tipicamente un ruolo di amministratore o pianificatore, controllando un sistema, (e.g. una città, un'azienda, una comunità) e prendendo decisioni per raggiungere specifici obiettivi all'interno della simulazione [19]. A differenza dei giochi d'azione, i gestionali non chiedono di sconfiggere un nemico ma di costruire, far crescere e ottimizzare qualcosa in un processo continuo.

Il macro-genere gestionale abbraccia numerose categorie di giochi. Di seguito viene presentata una panoramica dei principali sottogeneri appartenenti a questa categoria:

- City Builder: giochi focalizzati sulla progettazione, costruzione e amministrazione di città e insediamenti urbani;

- Tycoon: giochi in cui il giocatore è a capo di un'impresa o di un progetto economico, con l'obiettivo di far prosperare un'attività commerciale;

- Life Simulation: giochi il cui focus è la gestione della vita quotidiana di individui o comunità;

- Sport Management: giochi in cui il giocatore è chiamato a gestire una società sportiva, curandone gli aspetti manageriali piuttosto che l'azione sul campo;

- God Game: genere di giochi in cui il giocatore assume simbolicamente il ruolo di una divinità o di un'entità onnipotente che controllo in modo indiretto una popolazione o ecosistema;

- Strategici 4X: giochi dove il giocatore guida un impero attraverso la sua evoluzione, dall'esplorazione all'espansione fino all'eliminazione o sopraffazione degli avversari.

Popolarità e mercato del genere gestionale

Dal punto di visto commerciale e del pubblico, il genere gestionale rappresenta oggi una nicchia non trascurabile dall'industria. Anche se i gestionali puri raramente compaiono ai vertici delle classifiche di vendita, e siano in generale meno diffusi dei giochi d'azione, essi godono di una comunità dedicata e fidelizzata. Molti giocatori arrivano a toccare le centinaia (o addirittura migliaia) di ore giocando ai loro titoli gestionali preferiti, proprio grazie alla loro elevata rigiocabilità [20].

È' inoltre importante sottolineare che, sebbene il genere gestionale si sia sviluppato originariamente in maniera più forte per PC, con serie di punta come Football Manager che arrivano a decine di migliaia di utenti in contemporanea [20], il genere gestionale sta diventando sempre più cross-platform e accessibile. Sul mobile, i titoli di

simulazione e tycoon generano redditi settimanali che possono superare il milione di dollari (e.g. Isekai: Slow Life ha raggiunto circa 760 000 USD nell'ultima settimana del 2023 negli USA) [21].

2.1.4 Analisi del genere gestionale puro

Meccaniche di simulazione e evoluzione del sistema

Come anticipato, i giochi appartenenti al genere gestionale si basano sull'ottimizzazione e accrescimento di un sistema. Gran parte degli sforzi di design vengono quindi diretti a definire le procedure di sistema e regole secondo le quali il sistema (e.g. città, impero) evolve.

Queste includono solitamente la creazione di più sottosistemi, solitamente ispirati a logiche realistiche (e.g. economiche, fisiche, sociali) talvolta anche semplici che però interagiscono tra loro durante la partita dando origine a dinamiche di sistema estremamente complesse e difficili da prevedere in fase di design, che emergono solitamente quando il gioco è in azione.

Alcuni sottosistemi tipici rintracciabili sono quelli:

- Economico: definisce regole e procedure che simulano l'economia all'interno del gioco, che include i capitali monetari e i beni di scambio. Definisce le meccaniche di entrata (come il giocatore ottiene il proprio capitale) e di aggiustamento dei prezzi attraverso leggi di domanda e offerta;

- Demografico: modello che simula la popolazione e le sue dinamiche (come la natalità, la mortalità e la produttività) in funzione di diversi fattori (e.g. felicità, accesso ai servizi);

- Infrastrutturale: modello che definisce l'architettura topologica del sistema. Solitamente include elementi come strade, edifici e reti energetiche;

In Civilization VI [22], ad esempio, il mondo è composto da tile esagonali sulla quale il giocatore può edificare le proprie città, i propri distretti e costruzioni fornite dalle unità di lavoratori (sottosistema infrastrutturale). Questi producono risorse utili come l'oro, che può essere a sua volta speso per comprare altri edifici, unità e a mantenere ciò che è stato precedentemente costruito (sottosistema economico). La popolazione delle città cresce con il passare dei turni sulla base di fattori come la quantità di cibo disponibile e l'attrattiva, entrambe risorse fornite dalle infrastrutture. La popolazione a sua volta permette di sfruttare un maggior numero di caselle e edifici presenti nei confini della città, andando a fornire ulteriore cibo e risorse utili.

Meccaniche di gestione del sistema

Una volta definito il funzionamento e l'evoluzione del sistema, è necessario definire come il giocatore può agire su di esso e influenzarne l'andamento. Vengono quindi definite le procedure e le regole che definiscono le azioni che il giocatore può intraprendere per influenzare in maniera più o meno diretta ciascun sottosistema. Anche in questo caso il loro funzionamento è spesso ispirato a regole del mondo reale.

Nel caso del sottosistema economico, viene definito come il giocatore può sfruttare il capitale monetario o i beni acquisiti (e.g. comprando altre risorse utili o effettuare scambi con altri giocatori all'interno della partita).

Sul sottosistema demografico è più comune che il giocatore possa agire in maniera indiretta attraverso gli altri sottosistemi. La crescita della popolazione e le sue azioni sono solitamente spontanee, e il giocatore può semplicemente impostare preferenze su come viene impiegata la manodopera.

Il sottosistema infrastrutturale è invece quello su cui il giocatore ha solitamente maggiore controllo (e.g. attraverso la costruzione / rimozione degli edifici che alimentano ed influenzano gli altri sottosistemi).

Essendo la divisione in sistemi un punto cruciale del framework, questo aspetto verrà approfondito successivamente al suo interno.

Meccanismi di feedback e teoria del caos

Una caratteristica peculiare dei giochi gestionali è la presenza di circuiti di feedback interni multipli. Questi vanno creare dinamiche che emergono solamente in fase di gioco e che spesso sono difficili da prevedere in fase di design dalle singole procedure e regole. I feedback loop in questo caso vanno ad ampliare l'effetto delle singole azioni del giocatore.

Essi sono tipicamente di due tipi [2]:

- Feedback loop positivo: un incremento in un parametro del sistema genera a sua volta ulteriore aumento. In questo caso un successo iniziale può alimentare i successivi miglioramenti del giocatore;

- Feedback loop negativo: un aumento in un parametro provoca forze che ne rallentano la crescita ulteriore o la invertono;

La presenza di feedback loop multipli influenza non solo l'evoluzione dei singoli sottosistemi, ma anche l'evoluzione globale della partita. Possiamo vedere questo come una sorta di "Effetto farfalla" applicato all'ambito videoludico, dove singole azioni apparentemente insignificanti e piccoli variazioni nelle condizioni iniziali del giocatore possono cambiare potenzialmente l'andamento della partita e il bilanciamento.

L'applicazione dei concetti appartenenti alla dinamica dei sistemi risulta quindi particolarmente utili nel bilanciamento di questi sistemi e per capire meglio come meccaniche semplici possano generare comportamenti complessi e non lineari, che, se bilanciati in maniera inadeguata rischiano di penalizzare troppo il giocatore o impedirne la rimonta per il resto della partita (inficiando quindi la dimensione di "stabilità" del gioco).

Nel caso di Humankind [23], recensioni professionali hanno ad criticato il gioco a causa del forte effetto "palla di neve" che si crea all'interno dell'economia di gioco, permettendo al giocatore di accumulare facilmente numerose risorse sfruttando alcune categorie di edifici [24].

Elementi di conflitto e sfida

All'interno del genere gestionale la sfida non è quindi indotta dalla presenza di ostacoli come nemici che il giocatore deve combattere o affrontare, ma alla complessità di tenere insieme un sistema in crescita. La sfida è indotta dal problem solving continuo atto a mantenere l'equilibrio di fronte a difficoltà crescenti, e il conflitto è quindi generato dal sistema stesso.

Per mantenere la difficoltà ai livelli delle abilità del giocatore, il designer solitamente introduce un'escalation nella complessità del sistema. Questa parte da una dimensione ridotta per poi crescere sempre di più negli stadi avanzati del gioco, in modo che il giocatore non si trovi fin da subito a dover controllare una condizione di partenza eccessivamente complicata.

In molti giochi 4X come, ad esempio, nel già citato Civilization 6 [22], il giocatore non può costruire fin da subito tutti gli edifici del gioco, ma ricercando nuove tecnologie la scelta a sua disposizione si amplia.

Figura 4: estratto dell'albero di ricerca di Civilization VI [22].

Interfaccia Utente

In un genere che si fonda sulla gestione di sistemi complessi, fondamentale è il design dell'Interfaccia Utente e delle procedure che definiscono il modo con cui il giocatore può andare a compiere le proprie azioni e visualizzare lo stato attuale del sistema.

Come evidenziato precedentemente infatti, uno dei requisiti fondamentali per l'instaurarsi dello stato di flow è la presenza di feedback chiari che permetta al giocatore di comprendere il suo andamento rispetto agli obiettivi del gioco. In questo caso, se il giocatore fatica a comprendere lo stato del sistema, o ancora peggio le azioni che può effettuare (e.g. non sono facilmente accessibili o visibili sullo schermo) l'esperienza risulta frustrante.

La recente uscita di Civilization VII [25] è stata seguita da recensioni al di sotto della media rispetto ai precedenti capitoli (sulla piattaforma di Steam il numero di recensioni positive non raggiunge il 50% al momento della scrittura). Uno degli aspetti maggiormente criticati all'interno delle recensioni è proprio l'interfaccia utente, che risulta essere troppo confusa e densa di informazioni, rendendo l'esperienza di gioco poco scorrevole [26].

2.2 Panoramica del genere action-gestionale

Dalla combinazione di questi due generi nasce quindi il genere action-gestionale, che fonde il gameplay frenetico e coinvolgente dei giochi d'azione con la profondità strategica e l'impianto sistemico tipico dei gestionali. Per capire meglio come il genere si sia evoluto nel panorama storico, possiamo identificare i seguenti sottogeneri, nati dalla combinazione di azione in tempo reale con una sostanziale gestione di risorse, produzione o sistemi complessi.

Giochi di strategia in tempo reale (RTS)

I giochi di strategia in tempo reale sono quei titoli che combinano pianificazione strategia con azione in tempo reale. Il giocatore si ritrova spesso a comandare eserciti sottoforma di osservatore onnisciente che si scontrano in tempo reale e a raccogliere risorse, costruire edifici sul campo di battaglia e sviluppare alberi tecnologici.

Giochi di tattica di squadra in tempo reale (RTT)

I giochi appartenenti a questo sottogenere mettono il giocatore al comando di piccole squadre. Come gli RTS sono solitamente focalizzati su scontri sul campo di battaglia,

ma in questo caso il focus è minore sulla gestione di risorse e costruzione di edifici e maggiore sulla micro-gestione delle unità a disposizione.

Avatar-RTS

Genere di giochi di strategia in tempo reale dove il giocatore non è un osservatore onnisciente, ma pilota un personaggio che partecipa direttamente alla battaglia.

Tower defense

Sottogenere di giochi in cui il giocatore pianifica labirinti di difese (solitamente sottoforma di torrette di varia natura) cercando di impedire ai nemici di raggiungere un obiettivo strategico.

Survival-crafting

Questo genere di giochi unisce l'esplorazione di un mondo ostile con la necessità di raccogliere risorse grezze e la creazione (crafting) di oggetti per sopravvivere a nemici, fame, freddo o altri pericoli. Gli scenari sono solitamente aperti e il giocatore è costretto a bilanciare la raccolta di materiali con la costruzione di rifugi e combattimenti.

Factory-sim

Sottogenere in cui il giocatore si concentra sulla progettazione, ottimizzazione e automazione di linee complesse di produzione e reti logistiche. Queste prendono solitamente la forma di impianti industriali che trasformano materie prime in prodotti finiti, bilanciando produzione, spazio e costi.

God-Game

Il giocatore impersona una divinità che guida la crescita di una popolazione, ma può controllare singoli individui "possedendoli" o scatenare miracoli direttamente sul campo.

Squad-command action

Il giocatore controlla un avatar-comandante che impartisce ordini rapidi a un insieme di creature o minion che agiscono semi-autonomamente.

4X in tempo reale

In questo sottogenere i giochi 4X vengono riproposti ma senza turni: l'espansione dell'impero, le colonie e la diplomazia avvengono in tempo reale.

Grand-strategy RTS

Titoli di strategia dove oltre dover gestire aspetti legati a tasse, politica e dinastie, vengono sviluppati moduli separati per le battaglie comandate in tempo reale.

Management sportivo con partite "live"

Simulatori di gestione di squadre sportive che incorporano un motore di partita giocabile o visibile in tempo reale.

Giochi di battaglie in arena multiplayer (MOBA)

Giochi combinano elementi di controllo di un personaggio dove due squadre di eroi competono per distruggere la base avversaria in mappe simmetriche.

Giochi di ruolo d'azione (ARPG)

Sottogenere che unisce combattimenti in tempo reale e progressione del personaggio con un forte focus su esplorazione dei dungeon e gestione del loot. L'aspetto "gestionale" sta nel gestire equipaggiamento, statistiche e risorse per ottimizzare l'efficacia nelle fasi d'azione.

Battle Royale

Genere competitivo a larga scala in cui molti giocatori entrano in una mappa che si restringe progressivamente; vince l'ultimo sopravvissuto. La componente "gestionale"

riguarda prioritizzazione del bottino, gestione del rischio e delle risorse (munizioni, cure, posizionamento) a supporto dell'azione.

2.3 Evoluzione storica del genere action-gestionale

2.3.1 Anni 1981 – 1999: le origini

I primi esperimenti relativi al genere action-gestionale risalgono ai primi anni dell'industria videoludica. Uno dei primi tentativi di mescolamento dei due generi è Utopia (1981, Mattel Intellivision, Don Daglow) [27], strategico in tempo reale in cui i due giocatori in competizione amministrano la propria isola costruendo strutture e raccogliendo risorse. All'interno del gioco non è presente alcun tipo di intelligenza artificiale, limite imposto dalla RAM di 1kB della console [28], ed è infatti possibile giocarlo con un altro giocatore in locale. Sebbene presentasse evidenti limiti tecnici e di accessibilità, è di fatto possibile ritenerlo come il più antico antenato degli RTS [29].

Figura 5: estratto di Utopia [27], schermata di punteggio e di gioco.

Successivamente con il diffondersi dei computer domestici dotati di interfacce grafiche e mouse, il genere conobbe un'ulteriore evoluzione. The Ancient Art of War (Brøderbund, 1984) [30], gioco di strategia militare ispirato a Sun Tzu, è riconosciuto come uno dei primi RTS su PC (e in realtà predecessore del genere RTT). In questo gioco, che mostrava le potenzialità dei gameplay incentrati sulla gestione bellica in

tempo reale, sono tuttavia ancora mancanti componenti economiche e di costruzione, che verranno sperimentate in seguito in giochi di questo tipo [31].

Verso la fine degli anni 80' venne introdotto anche il concetto di God Game, con la pubblicazione di Populous (1989, Bullfrog Productions) [32]. Il giocatore veniva messo nei panni di una divinità con l'obiettivo di alterare il territorio per guidare il proprio popolo contro tribù nemiche. Il gameplay non permetteva di impartire ordini diretti alle singole unità, ma di influenzare le condizioni del mondo per arrivare alla vittoria. Populous ebbe grande successo (raggiungendo le 4 milioni di copie vendute [33], un numero elevato per gli standard dell'epoca) mostrando il potenziale di un gameplay d'azione meno incentrato sul controllo diretto di entità e maggiormente focalizzato sulla gestione dall'alto.

Parallelamente veniva pubblicato Herzog Zwei (1989, Sega Mega Drive) [34], che fondeva azione arcade e strategia in tempo reale. All'interno del gioco i giocatori andavano a controllare in prima persona un'unità motorizzata con cui combattere e trasportare truppe, mentre ne costruivano di aggiuntive e tentavano di conquistare basi sulla mappa che andavano a generare ulteriori ricavi. Il gioco, le cui vendite risultarono modeste, ispirò però numerosi importanti titoli successivi, tanto che gli sviluppatori di giochi Warcraft: Orcs & Humans [35] e StarCraft [36] lo citeranno come fonte di ispirazione diretta [37].

Fu però Dune II: The Building of a Dynasty (Westwood Studios, 1992) [38] a definire quelli che sono gli elementi cardini del genere RTS moderno. Fu il primo titolo a integrare in maniera completa meccaniche divenute poi classiche nel genere, come la raccolta di risorse da parte di un'entità con ruolo specifico, la costruzione di una base, l'albero tecnologico progressivo, l'addestramento di unità differenziate e l'uso intensivo del mouse per la loro gestione [39]. Inoltre, metteva a disposizione un'interfaccia sidebar verticale, gestibile da mouse e scorciatoie da tastiera ed era di fatto il primo gioco a implementare il path-finding locale in tempo reale per decine di unità simultanee. Dune II ispirò numerosi titoli successivi come Command & Conquer (1995, Westwood Studios) [40] e Warcraft: Orcs & Humans (1995, Blizzard) [35].

Quest'ultimo introdusse come novità due fazioni "asimmetriche", ovvero con unità e strutture differenti, supportando partite tra due giocatori sfruttando un modem [41].

Figura 6: estratto della schermata di gioco di Dune II [38].

Blizzard nel 1998, dopo il lancio di Battle.net nel 1996, pubblicherà Starcraft, che rappresenterà di fatto l'apice della formula RTS classica. Il gioco introduceva tre fazioni fondamentalmente diverse, ma bilanciate tra loro al punto da diventare estremamente popolare a livello competitivo e online, specialmente in Corea del Sud. Solo nell'anno di lancio il gioco arrivò a vendere circa 1.5 milioni di copie, per poi arrivare a 9.5 milioni entro il 2007 (di cui 4.5 milioni in Corea del Sud) [42].

Proprio dall'esplosione del genere RTS nacque poi il genere MOBA. Herzog Zwei aveva già introdotto il controllo di un singolo "eroe" e ondate di nemici controllate dall'intelligenza artificiale, ma nel 1998 un appassionato di Starcraft chiamato Aeon64 pubblicò una mappa personalizzata per il gioco chiamata Aeon Strife [43] nella quale tre giocatori dovevano controllare una singola unità eroe avanzando lungo tre corse predefinite (le cosiddette lanes) difese da torri, introducendo numerosi degli elementi chiave tipici del genere MOBA. Sebbene la mappa avesse un pubblico di nicchia, ispirò direttamente Warcraft III [44] che renderà popolare il genere.

Figura 7: estratto della schermata di gioco di Starcraft [36], mappa Aeon Strife.

Nella seconda metà degli anni ‘90 oltre ai titoli incentrati sul gameplay “bellico” sono nati anche esperimenti particolari come Dungeon Keeper (Bullfrog Productions, 1997) [45], che combinava il concetto di Dungeon Crawler con quello dei God Game mettendo il giocatore nei panni di un malvagio guardiano del dungeon che deve costruire stanze, raccogliere risorse e difendere la propria tana. Questo va ad evidenziare come gli elementi action e gestionali, attraverso l’evoluzione delle piattaforme, dell’esperienza degli sviluppatori e degli utenti stessi, iniziassero a integrarsi e complementarsi a vicenda.

2.3.2 Anni 1999 – 2009: il periodo d’oro degli RTS e la nascita dei MOBA

Alla soglia dell’uscita della Playstation 2 (anno 2000), il genere era dunque ampiamente diffuso. In questo periodo gli RTS raggiungono la piena maturità, e accanto a titoli basati sulle classiche meccaniche di costruzione ed economia, vengono sviluppati titoli che riducono o eliminano completamente la macro-gestione. Con Ground Control (2000) [46] e Sudden Strike (2000) [47] si diffondono i Tattici in Tempo Reale, dove il giocatore dispone solo di forze preassegnate e deve impiegarle al meglio sul campo di battaglia. Questi titoli si concentrano maggiormente sulla micro-gestione delle unità e spostano il focus delle meccaniche sullo scontro più che sull’economia, con meccaniche

(come i bonus da copertura, fuoco di soppressione, direzionalità delle truppe) che diventeranno poi standard di titoli successivi [48].

Cruciale per l'evoluzione degli RTS classici è invece l'adozione di nuove tecnologie 3D e lo sviluppo di motori di gioco proprietari. Già nel 1999 Homeworld [49] introduceva per primo ambientazioni 3D [50] complete nello spazio [51]. Command & Conquer: Generals (2003) [52] proponeva ambienti tridimensionali avanzati grazie al nuovo motore SAGE (Strategic Action Game Engine), sviluppato da EA per competere con Blizzard ed Ensemble Studios [53]. Il motore consentiva l'uso di luci e ombre dinamiche che permettevano effetti cinematografici e suggestivi. Anche Relic Enterteinment sviluppo un motore proprietario, l'Essence Engine, che integrava il middleware Havok permettendo effetti fisici di distruzione realistici [54].

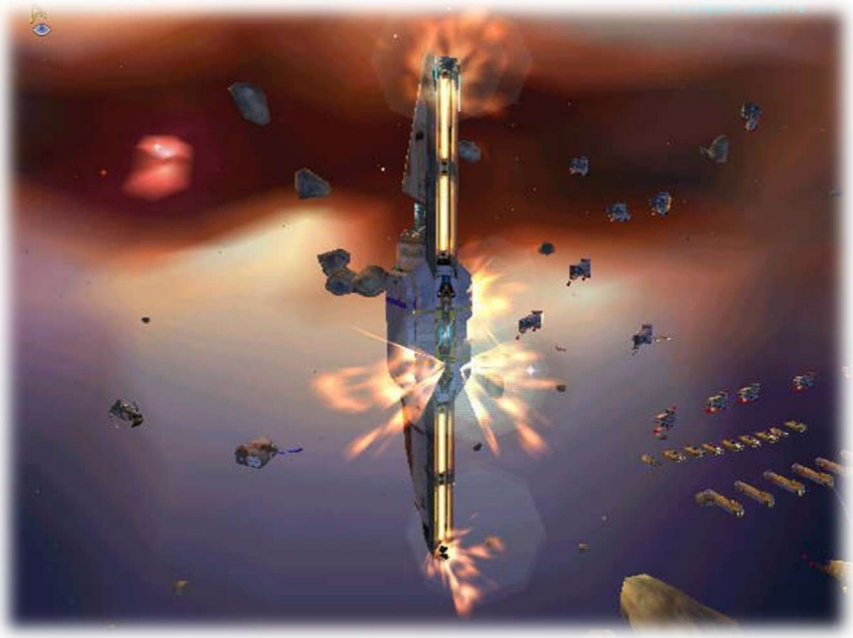


Figura 8: estratto della schermata di gioco Homeworld [49].

I miglioramenti tecnologici permettevano da un lato l'evoluzione dei titoli dal punto di vista immersivo e cinematico, consentendo quindi di creare ambienti 3D dettagliati, animazioni realistiche e effetti speciali evoluti, e dall'altro permettevano l'integrazione di meccaniche nuove basate sui cambiamenti che la mappa subiva durante il corso della partita. Dawn of War 1(Relic, 2004) [55] andava a sfruttare questi miglioramenti per offrire un gameplay immersivo e aggressivo (grazie anche all'uso di "Punti Strategici"

sparpagliati per la mappa anziché concentrati nella zona iniziale del giocatore) con un gran numero di entità, effetti su schermo e animazioni dettagliate. Company of Heroes (Relic, 2006) introdurrà invece alcune delle meccaniche meglio riuscite all'interno del genere, permettendo al giocatore di modificare (tramite bombardamenti e truppe specifiche) di modificare le condizioni della mappa, aumentando la profondità strategica e bilanciando la macro-gestione economica con la micro-gestione delle truppe, che ora potevano sfruttare ripari e accerchiare i nemici in uno scenario in continua evoluzione.

Figura 9: estratto della schermata di gioco di Dawn Of War 1 [55].

Dal graduale spostamento da un gameplay basato sulla macro-gestione a uno sulla micro-gestione, nacque quello che è di fatto il primo prototipo effettivo del genere MOBA. Defense of the Ancients (DotA), una mod per Warcraft III direttamente ispirata alla già citata Aeon Strife di Starcraft, andava a definire quelli che rimarranno gli elementi chiave nei giochi di questo genere [56]: gli elementi tipici degli RTS venivano quasi del tutto eliminati e ogni giocatore controllava un singolo personaggio dotato di abilità uniche, con l'obiettivo di eliminare la base avversaria. Viene rimossa quasi completamente la parte di gestione economica, relegata alla scelta su come spendere l'oro guadagnato sconfiggendo le unità "minori" controllate dall'IA. Il risultato è un gameplay che presenta sì degli elementi gestionali, ma è molto più concentrato sull'azione e uno stile "arcade". Questa formula divenne così diffusa a livello competitivo (grazie anche alla crescente diffusione della banda larga e dei servizi di

gioco multiplayer) che soppianterà nel periodo successivo i classici strategici in tempo reale e diverrà fondamentale nella diffusione degli eSports.

In questo periodo si tenta anche di trasporre il genere su console, operazione inizialmente difficile a causa dell'ampio uso di mouse e tastiera da parte dei titoli. Alcuni sviluppatori sperimentarono questo territorio attraverso modifiche ad-hoc di interfacce e gameplay. Kessen (Koei, 2000) [57], titolo di lancio per PS2, presentava battaglie risolte automaticamente e il giocatore si limitava a impartire ordini strategici alle truppe. Pikmin (2001, GameCube) [58] e Odama [59] combinavano invece elementi di strategia e puzzle. Nel primo il giocatore controlla un personaggio (Captain Olimar) che esplora i livelli, raduna buffe creature chiamate Pikmin e le dirige per raccogliere risorse o affrontare nemici. Nel complesso Pikmin, anche con l'uso di una limitazione temporale per completare la missione, riuscì a portare l'essenza del genere action-gestionale moderno su console. Odama invece permetteva di impartire ordini attraverso un microfono contenuto all'interno del gioco.

Figura 10: estratto dalla schermata di gioco di Odama [59].

Oltre ai sottogeneri già visti, in questi anni inizia a svilupparsi anche una nicchia di appassionati dei Tower Defense. Già in Warcraft III [44] la comunità iniziava a creare mappe personalizzate incentrate quasi interamente sul posizionamento difensivo di edifici, ma queste idee presero slancio poi attraverso giochi web sviluppati in Flash. Nel

2007 per esempio Desktop Tower Defense [60] divenne virale, attirando milioni di giocatori attraverso la sua formula immediata [61].

2.3.3 Anni 2009-2016: la nascita dei survival-crafting

Negli anni successivi, in particolare nel periodo appartenenti agli anni dal 2009 al 2016, il genere action-gestionale vide una profonda evoluzione guidata dai numerosi esperimenti degli sviluppatori.

I Tower Defense diventano estremamente popolari, non solo su PC ma anche in ambito mobile, poiché gli smartphone ora permettevano di integrare comandi e interfacce intuitive, e questo sottogenere si prestava particolarmente bene al tipo di input fornito dai nuovi dispositivi e ad un pubblico casual. Plants vs Zombies (PopCap, 2009) [62] fu un caso emblematico poiché reinterpretava la formula con cura e umorismo e divenne presto un best-seller sia su PC che su Mobile, vendendo oltre 300.000 copie in nove giorni su iPhone [63]. Kingdom Rush (Ironhide, 2011 [64]), nato come gioco Flash e poi portato su mobile, raggiunse 17 milioni di download combinati su App Store e Play Store [65].

Figura 11: estratto del gameplay di Plants Vs Zombies [62].

Da questo genere nacquero poi esperimenti degli sviluppatori di grande successo, come Dungeon Defenders (Trendy, 2010) [66] che univa gli elementi tipici dei Tower

Defense con un gameplay in terza persona action-rpg. Il gioco vendette oltre 250.000 copie nelle prime due settimane [67] e oltre 600.000 entro la fine dell'anno 2011 [68].

Il genere degli RTS viveva invece in questo periodo il suo canto del cigno. Starcraft II: Wings of Liberty (Blizzard, 2010) [69] rappresentava di fatto l'apice e l'ultimo grande successo commerciale di questo sottogenere. Il gioco includeva tutti gli elementi introdotti dai predecessori del suo genere, ma con un'interfaccia e un'esperienza modernizzate che mantenevano la profondità strategica e l'alta competitività tipici della serie, ma con numerosi miglioramenti di accessibilità, come le code di produzione più gestibili e gruppi di controllo più ampi, rendendo il gioco più adatto al pubblico moderno. Il gioco vendette oltre 1.5 milioni di copie nelle prime 48 ore [70], segnando un record assoluto all'interno del sottogenere e diventando in Corea del Sud quasi uno sport nazionale e in occidente uno dei giochi di punta dei primi anni degli eSports. Tuttavia, già intorno al 2014-2015 il genere mostrava un calo significativo. Gli ascolti dei tornei calavano e Blizzard stessa iniziò a spostare l'attenzione da Starcraft verso altri titoli. Legacy of the Void, espansione di Starcraft 2 uscita nel 2015, faticava a coinvolgere nuove generazioni e i principali publisher investivano in altri generi divenuti più popolari. Di fatto nessun nuovo RTS, anche se di ottima qualità, riuscì ad invertire la tendenza in modo significativo e la maggior parte del pubblico competitivo e casual si spostava altrove, in particolare verso i MOBA [71].

Figura 12: estratto del Gameplay di Starcraft 2[69].

Nel 2009 questo genere, nato come precedentemente visto da mod e mappe non ufficiali di Starcraft e Warcraft 3, divenne titolo standalone con la pubblicazione di League of Legends [72]. Il titolo, sviluppato da dei veterani di DotA [73] che fondarono Riot Games, divenne presto estremamente popolare, tanto che nel 2016 arrivava a contare circa 100 milioni di giocatori attivi mensili [74]. Nel 2011 Valve organizzò il primo torneo The International dedicato a Dota 2 [75], con un montepremi di 1.6 milioni di dollari [76], il più grande fino ad allora, e che arriverà a 25 milioni di dollari con The International 7 [77]. Questo segnò una svolta nel campo degli eSports e l'affermazione dei MOBA come discipline professionistiche, tanto che entro la metà del 2010 circa il 40% dei premi relativi agli eSports proveniva dai tornei di questa categoria [78]. Il successo di questo sottogenere era da attribuirsi in gran parte alla maggiore accessibilità: League of Legends e Dota 2 erano free-to-play e maggiormente apprezzabili da uno spettatore rispetto agli RTS, e di conseguenza il gameplay proposto si prestava maggiormente alle nuove piattaforme social e di streaming in via di sviluppo come YouTube e Twitch.

In questo periodo, esplose anche un nuovo genere che diverrà un caso unico nella scena dei videogiochi indipendenti: i giochi sandbox e survival-crafting. Il caso più emblematico è sicuramente Minecraft [79], sviluppato da Mojang, apparso in versione

alpha nel 2009. Il gioco permetteva al giocatore di raccogliere risorse e costruire oggetti e edifici in un mondo costituito interamente da blocchi generato proceduralmente, popolato da animali e mostri, in cui il giocatore poteva dosare liberamente creatività e capacità di sopravvivenza. Nel giro di poco tempo Minecraft diventerà uno dei giochi più venduti di sempre, raggiungendo nel 2016 le 100 milioni di copie vendute e un vero fenomeno di massa destinato a sopravvivere nel tempo [80]. Per capire l'entità di questo fenomeno, basti pensare che nel novembre 2014 i video di Minecraft su YouTube totalizzano 47 miliardi di visualizzazione, passando in prima posizione tra i brand videoludici [81].

A Minecraft seguirà un'intera ondata di titoli incentrati sulla sopravvivenza con elementi di crafting e/o sandbox, con enfasi diverse sulla parte di gestione delle risorse e creatività. La maggior parte di questi titoli non era sviluppata da case di sviluppo AAA, ma da piccoli team indipendenti che spesso rilasciavano il proprio gioco in accesso anticipato su Steam nella speranza di proseguire lo sviluppo ricevendo finanziamenti e supporto da parte della community. Rust (Facepunch, 2013) [82] raggiungse oltre 5 milioni di copie vendute nel 2017 nonostante la lunga permanenza nello stato di early-access [83].

Nacque di fatto oltre che un nuovo genere, un nuovo paradigma di sviluppo: quello dei giochi Early Access (anche detto "alpha founding"), che la piattaforma di Steam (divenuta ormai una delle più popolari piattaforme di distribuzione digitale videoludica) supportava ampiamente, e che permetteva agli sviluppatori di vendere il proprio gioco anche se incompleto sulla base della fiducia degli utenti, ottenendo quindi fondi per lo sviluppo e feedback sullo sviluppo. Molti di questi titoli non raggiunsero mai la release definitiva: solo il 25 percento dei titoli Early Access dal lancio del programma vennero in seguito rilasciati come giochi completi [84].

2.3.4 Anni 2017-oggi: la diffusione dei battle royale e la nascita delle olimpiadi eSports

Nel 2017, con la pubblicazione di PlayerUnknown's Batttlegrounds [85] in Early Access (PUBG) da parte di PUBG Corporation, nasce e diventa popolare il genere dei

Battle Royale (termine coniato dal romanzo omonimo di Koushun Takami del 1999). Il titolo trasformava in release standalone l'idea che Brendan "PlayerUnknown" Greene aveva precedentemente sviluppato in una mod di Dayz, Dayz : BattleRoyale e che a sua volta si ispirava ai server di Minecraft che implementavano meccaniche stile "Last man standing" tratte dal film Hunger Games [86].

Il genere andava ad unire l'azione adrenalinica e competitiva degli sparatutto con la tensione costante della raccolta e gestione di oggetti ed equipaggiamento ritrovabili all'interno della mappa popolata da altri giocatori ostili, e si prestava bene sia ad un pubblico casual/spettatore, che competitivo. PUBG arriva a vendere oltre 50 milioni di copie entro il 2018 [87], e verrà superato presto da Fortnite [88] di Epic Games, che cambia radicalmente il proprio gameplay per adattarlo alla formula Battle Royale. Questo genere diventa popolare anche nell'ambito degli eSports, tanto che nel 2019 il vincitore del mondiale di Fortnite arriva ad aggiudicarsi 3 milioni di dollari [89]. Di fatto i battle royale arrivano a dominare il mercato e le piattaforme di streaming, diventando un vero e proprio fenomeno culturale globale che coinvolgono spesso personaggi cinematografici, collaborazioni con brand di terze parti e concerti live, creando un vero e proprio metaverso ludico. PUBG arriverà a vendere oltre 70 milioni di copie entro il 2020 [90] e Fortnite nel 2023 arriva a contare più di mezzo miliardi di giocatori registrati [91].

I MOBA, nel frattempo, continuano a rimanere estremamente popolari e diventano il baluardo degli eSports mondiali. La finale del campionato mondiale di League of Legends del 2018 viene seguita da circa 99 milioni di persone in streaming [92], avvicinando il pubblico degli sport tradizionali e dei tornei videoludici. La pandemia di COVID-19 ha intensificato questo trend, accrescendo sia il pubblico di giocatori che di spettatori del genere [93], tanto che i Mondiali di League of Legends del 2025 hanno un montepremi previso di 5 milioni di dollari, oltre il doppio rispetto al 2024 [94].

Sebbene il genere degli RTS tradizionali sia invece passato ad essere maggiormente riservato ad un pubblico di nicchia, molti titoli riprendono il concetto di strategia in tempo reale in chiave differente rispetto ai classici. Tra i Grand-RTS, la serie Total War Warhammer [95] per esempio ottiene grande attenzione combinando l'iconico universo

fantasy di Warhammer con la strategia e gestionali delle grandi dinastie e battaglie tipiche del brand Total War. Total War Warhammer III [96] pubblicato nel 2022 da Creative Assembly, ottiene ottime vendite e recensioni, vendendo quasi 1 milione di copie nel primo mese e generando circa 40 milioni di dollari su Steam [97].

Nel frattempo, nuovi titoli che vanno a combinare in maniera sempre più particolare e sfumata componenti di gioco action e gestionale diventano popolari. Mount & Blade II: Bannerlord (TaleWorlds, 2020) [98] metteva il giocatore nei panni di un guerriero in prima linea implementando meccaniche in stile action RPG, e al contempo di un signore feudale che deve amministrare eserciti e città su larga scala. Valheim (Iron Gate Studio, 2021) [99] combina invece un gameplay survival e cooperativo a tema vichinghi in un mondo generato proceduralmente e ostile. La combinazione tra sopravvivenza, cooperazione e azione, suscita grande interesse da parte del pubblico, tanto che dopo essere uscito in accesso anticipato nel 2021 e senza una forte operazione di marketing, Valheim riesce comunque a vendere oltre 5 milioni di copie in un solo mese, e rientrando nella top 5 dei titoli più giocati di sempre sulla piattaforma di Steam [100], che ancora ad oggi risulta la principale piattaforma di distribuzione videoludica digitale.

2.3.5 Conclusioni dell'analisi storica del genere action-gestionale

Analizzando l'evoluzione videoludica dei giochi appartenenti a giochi action-gestionali, si può notare oltre a una progressiva evoluzione delle meccaniche e dinamiche di gioco e delle piattaforme videoludiche, una progressiva "simbiosi" dei generi action e gestionali, inizialmente nati come due formule distinte e apparentemente antitetiche, tanto che nei titoli moderni le due componenti spesso si fondono tra loro in maniera estremamente innovativa e difficile da incasellare in categorie pre-esistenti. D'altronde, il pubblico e gli utenti stessi hanno subito a loro volta una forte trasformazione, tanto che numerosi degli esperimenti che hanno poi portato a opere di successo sono nati da comunità di piccoli sviluppatori indipendenti o addirittura da mod di giochi sviluppate da altri giocatori. Lo sviluppo e diffusione di piattaforme social e streaming, ha infine contribuito enormemente allo sviluppo e diffusione del genere anche nei confronti del pubblico casual o addirittura non giocante, arrivando a dare vita agli eSports, fenomeno divenuto negli ultimi anni impattante non solo a livello economico ma anche sociale.

4. Framework

4.1 Caratteristiche del genere

4.1.1 Riassunto delle analisi

Ci si pone ora il quesito di formulare una definizione oggettiva di cosa sia un titolo action-gestionale sulla base dell'analisi storica del genere, l'analisi dei generi action e gestionale e i titoli precedentemente presentati. In questo modo è possibile approfondire cosa contraddistingue questo genere, qual è il suo valore ludico, e costruire quindi un framework che permetta al designer di costruire titoli compiendo scelte di design ponderate all'interno del workflow.

Analizzando il genere action, in particolare si sono osservati le seguenti caratteristiche:

- Lo svolgimento in tempo reale degli eventi di gioco;
- La presenza di ostacoli continui che si oppongono al giocatore e alimentano la sfida;
- L'enfasi sull'abilità manuale e riflessi del giocatore.

Per quanto riguarda il genere gestionale si è invece osservato che:

- Esso sia principalmente incentrato sul controllo, gestione e ottimizzazione di un sistema più o meno complesso, caratterizzato dalla presenza di risorse e meccaniche di gioco con memoria all'interno della partita;
- La forte enfasi sul concetto di "scelta significativa" all'interno del sistema di gioco e i suoi sottosistemi;

- Un ritmo generalmente più rilassato, non per forza in tempo reale, e che pone al giocatore un tipo di sfida maggiormente analitico e strategico, più che manuale.

4.1.2 Punti chiave del genere

Similmente a quanto fatto per i generi action e gestionale, si possono quindi delineare alcune caratteristiche ricorrenti all'interno del genere e dei suoi sottogeneri elencati nel capitolo 2.2.

Pressione temporale e prioritizzazione delle azioni

Nei titoli action-gestionali la complessità del sistema e dei suoi sottosistemi, unita all'evoluzione in tempo reale, introduce vincoli temporali che trasformano ogni azione in un problema di prioritizzazione: il giocatore deve decidere rapidamente cosa fare e in quale ordine, andando generare una forte dinamica di "urgenza" [1].

La qualità dell'esperienza e della sfida dipendono quindi fortemente dal bilanciamento tra complessità della gestione del sistema e il ritmo temporale imposto dagli ostacoli del gioco. Se questi sono sbilanciati rispetto alle abilità del giocatore, lo stato di flow e la qualità dell'esperienza ne risentono notevolmente.

In Overcooked [101], per esempio, i timer degli ordini e il punteggio legato alla rapidità impongono una costante prioritizzazione delle azioni (e.g. taglio, cottura, impiattamento, consegna). Il fallimento di un ordine comporta perdita di punti e opportunità in maniera cumulativa. La descrizione ufficiale del gioco enfatizza la necessità di "preparare, cucinare e servire prima che il tempo scada" [114].

Figura 13: estratto dalla schermata di gioco di Overcooked[101].

Tempo come meccanica, risorsa e ostacolo

La conseguenza dell'urgenza che si viene a generare è che il tempo diventa non solo una cornice, ma una meccanica (o un insieme di meccaniche) che modula direttamente la difficoltà e la strategia: più si è lenti, più le probabilità di raggiungere l'obiettivo diminuiscono; il giocatore può accelerare, ma questo a costo di risorse raccolte e qualità delle scelte effettuate se non è abbastanza abile. Il tempo diventa così contemporaneamente risorsa da investire nelle scelte e vincolo che penalizza l'attesa.

In Risk of Rain 2 [102], ad esempio, la difficoltà cresce in modo continuo con il passare del tempo ed è visualizzata da un indicatore che avanza durante la partita. Progredire in fretta permette di affrontare aree successive ad una difficoltà inferiore, ma riduce il numero di oggetti ritrovati. Il legame tra tempo e difficoltà all'interno del gioco è uno dei punti chiave che gli sviluppatori hanno scelto di implementare [115].

Multitasking e carico cognitivo

La compresenza di azioni fisiche veloci, il monitoraggio di più stati e la presenza di pianificazione a breve / medio / lungo termine, vanno a creare una forte componente di multitasking. Come evidenziato da studi moderni sulla Cogntive Load Theory, un

eccesso di stimoli eterogenei causa un calo delle performance [116]. Un carico eccessivo causa frustrazione e stress, andando a spostare in negativo l'instaurarsi dello stato di flow. Di conseguenza il bilanciamento tra intensità e cadenza degli eventi di gioco diventa un punto chiave da tenere in considerazione all'interno del design di gioco.

In They are Billions [104], è presente una funzione di "pausa in tempo reale" che permette al giocatore di fermare il tempo per impartire ordini e di analizzare con più calma lo stato del gioco. Si vedrà in seguito perché questa funzione non va a compromettere le caratteristiche chiave del genere e come modifica l'esperienza del giocatore.

Importanza della UI

L'interfaccia utente, che ricopre un ruolo importante in qualunque titolo, assume quindi un ruolo particolarmente rilevante in un genere ad alta densità d'informazione e urgenza. Essa rende visibile, leggibile e controllabile lo stato del sistema, supportando percezione, comprensione e proiezione [105]. Una UI sub-ottimale aumenta la difficoltà di selezione / azione e va a rompere lo stato di flow (che richiede una lettura coerente del sistema di gioco e degli obiettivi) anche con un ottimo design delle meccaniche.

Imperator: Rome (Paradox, 2019), per esempio, ha avuto un lancio problematico con critiche legate alla densità e leggibilità dell'interfaccia. È stata poi introdotta un'ampia revisione della UI/UX con la major patch 2.0 "Marius" proprio per ridurre l'impatto delle criticità legata all'accessibilità del gioco [106].

Caratteristiche motivazionali del genere

Il sottogenere action-gestionale si concentra quindi soprattutto attorno a quattro profili motivazionali ricorrenti:

- Achiever / Optimizer-Builder: coloro che cercano padronanza misurabile e ottimizzazione di pipeline all'interno di un sistema;

- Explorer / Tinkerer: spinti dalla scoperta sistemica e dall'esperimento su regole emergenti;

- Socializer / Coordinator: motivati dalla gratificazione nella coordinazione sotto vincoli temporali condivisi;

- Competitor / Speedrunner: coloro che ricercano la sfida agonistica.

La tassonomia precedente si rifà al filone classico di Richard Bartle [107] e ai cluster motivazionali di Nick Yee [108].

Tali profili, nel modello MDA, si concretizzano attraverso:

- Le dinamiche e estetiche ricorrenti precedentemente analizzate;

- La combinazione di varietà nelle scelte all'interno dell'economia di gioco e lo svolgimento di azione in tempo reale;

- Lettura / coordinazione facilitata da un'interfaccia ben sviluppata.

4.1.3 Definizione di gioco action-gestionale

Si può quindi arrivare ad una definizione precisa e coerente con l'analisi svolta fin ora di quali siano effettivamente le caratteristiche di un gioco appartenente a un genere così vasto.

Si definisce gioco "action-gestionale" un gioco che presenti al suo interno le seguenti componenti:

i) - Componente action

L'esito del gioco e il successo degli obiettivi del giocatore dipendono da abilità esecutive in tempo reale.

ii) – Componente gestionale

Il giocatore deve progettare e controllare un sistema affetto da memoria e sotto pressione temporale a causa della componente action.

iii) – Retroazione forte tra azione i e ii

Le due componenti action e gestionale sono legate tra loro da un feedback di retroazione bidirezionale che influenza direttamente l'esito di ciascuna delle due parti.

I punti sopra citati intendono formalizzare in maniera precisa e completa gli elementi chiave che caratterizzano il genere e i sottogeneri precedentemente visti.

4.2 Conseguenze e riflessioni sulla definizione

Di seguito si valutano quindi, sulla base della definizione precedentemente mostrata, quali siano le conseguenze effettive sui punti chiave che contraddistinguono il genere.

4.2.1 Importanza del concetto di memoria

La memoria del sistema rende l'economia e gli stati di gioco dipendenti dal percorso (la catena delle azioni del giocatore): decisioni pregresse vanno a influenzare il ventaglio delle opzioni future e introducono costi di inversione.

Questo effetto è desiderabile perché incrementa la significatività delle azioni ed eleva l'importanza della pianificazione, favorendo l'abilità gestionale.

Il punto ii) si ricollega quindi profondamente alle dimensioni di "Memoria" e "Stabilità" di gioco introdotta da Walter Nuccio [2] e ai design pattern da lui presentati. Senza di essa la parte gestionale del gioco perderebbe di rilevanza all'interno del gameplay e risulterebbe trascurabile, poiché non è necessario pianificare alcun tipo di azione se queste non hanno conseguenze permanenti sullo stato di gioco e sono reversibili senza costo.

È importante notare che il punto ii) non implica irreversibilità assoluta. La reversibilità parziale attraverso un costo è spesso salutare, e va ad impattare direttamente la dimensione della Stabilità. Un'eccessiva irreversibilità produce di fatto dei lock punitivi (impossibilità di rimonta) che riduce la percezione di controllo e il conseguente stato di flow.

In Starcraft II [69], la perdita di un esercito (unità precedentemente prodotte) non è reversibile: per ripristinare forze sul campo occorrono risorse (minerali e gas vespene), tempo di produzione e strutture adeguate; in questa situazione il giocatore è esposto tatticamente e strategicamente. Inoltre, l'importanza della catena delle azioni è così rilevante che per ciascuna fazione è definita la catena ottimale di azioni da eseguire nei primi minuti di gioco per ottimizzare i vantaggi futuri a livello competitivo.

Figura 14: Starcraft 2 [69], estratto della coda di costruzione iniziale consigliata per i Terran.

4.2.2 Simultaneità temporale parziale

La definizione non richiede presenza contemporanea continua di i) e ii): il genere ammette la presenza di fasi disgiunte, purché esistano finestre in cui sono combinate e sia soddisfatto il vincolo iii).

La simultaneità parziale consente ritmi di gioco più leggibili e flessibili e l'uso di pattern come la pausa tattica senza snaturare il genere, poiché il giocatore per poter

portare a termine il proprio obiettivo deve comunque passare attraverso fasi in tempo reale e seguire l'evoluzione del sistema.

FTL: Faster Than Light [109] alterna momenti di pura gestione (e.g. potenziamento della nave, posizionamento dell'equipaggio) e combattimenti in tempo reale con la possibilità di mettere in pausa premendo barra spaziatrice, durante i quali si impartiscono ordini e si controllano i sistemi della nave.

4.2.3 Concetto di retroazione forte e contributo di entrambe le componenti alla sfida del gioco

Per rispettare iii), gli aspetti action devono influenzare lo spazio decisionale della componente gestionale, e quest'ultima deve condizionare la prestazione esecutiva.

Perché la combinazione regga, l'obiettivo primario deve quindi richiedere un successo minimo in entrambi i domini e a impedire che l'utilità del progresso sia semplicemente additiva.

In termini di MDA: le meccaniche dei due domini devono generare dinamiche interdipendenti che sostengano un'unica estetica di sfida. Ciò implica progettare una bidirezionalità causale osservabile (e.g. posizionamento delle truppe che influenza la loro efficacia, e che di conseguenza riduce le perdite e quindi le risorse usate).

Company of Heroes [110] lega in modo stretto la tattica e l'economia territoriale: catturare punti territorio aumenta l'acquisizione di manodopera, munizioni e carburante, che a loro volta permettono di costruire unità, upgrade e edifici; al contrario, cattive scelte tattiche di posizionamento dell'esercito (e.g. mancato uso di coperture, direzionalità inefficace) riducono la capacità di tenere punti e dunque la produzione futura.

Figura 15: Esempio di truppa con direzionalità (artiglieria) di Company of Heroes 2

4.3 Workflow del framework

Il framework proposto all'interno del lavoro prende come punto di partenza il modello play-centric descritto da Tracy Fullerton [2], basato sull'alternanza iterativa di design → playtest → analisi. La struttura originaria viene estesa e rivista sulle caratteristiche dei giochi action-gestionali e il workflow che ne risulta può essere rappresentato attraverso il seguente diagramma a blocchi.

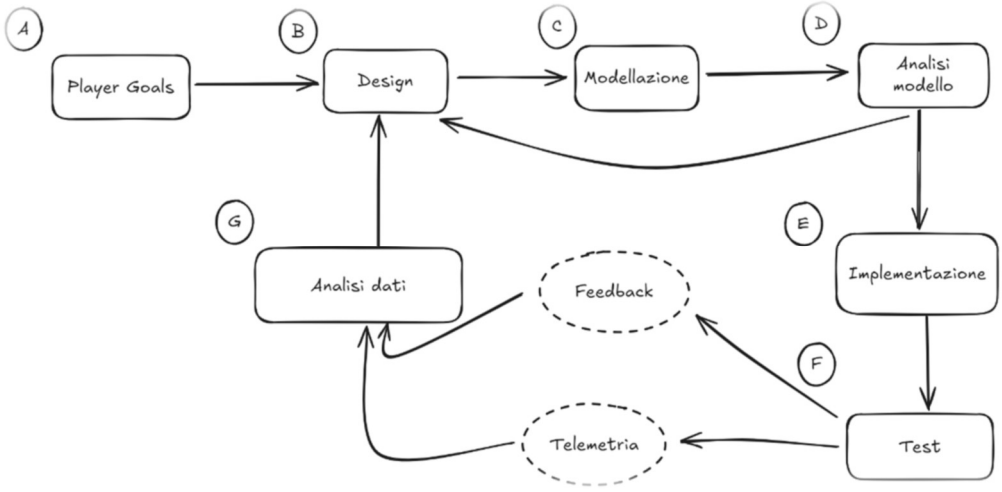


Figura 16: diagramma di workflow del framework.

A. Definizione dei Player Goals

Il passo iniziale mira a definire gli obiettivi ludici che il designer si pone nei confronti dei giocatori, come precedente descritto. Sebbene i Player Goals siano specifici per ciascun titolo e identificati da ciascun designer, è importante specificare questo passaggio perché è da esso che vengono prese successivamente le decisioni di design e la creazione del concept.

B. Design

Partendo dai Player Goals e dal concept iniziale, si prosegue creando il design del gioco, definendo le meccaniche che vanno a costituire il core loop e i vari sistemi di gioco. Verranno quindi identificati gli elementi chiave del core loop action-gestionale, quali sono i sistemi di gioco tipici, le loro caratteristiche e i loro collegamenti con le dimensioni di gioco precedentemente viste.

C. Modellazione

Le meccaniche vengono quindi modellate attraverso un apposito linguaggio e metriche di design, che permettano di visualizzare in maniera più precisa il flusso del gameplay e del core loop precedentemente individuato, oltre che le interazioni tra i diversi sistemi. In questa fase verrà definito un modello ad-hoc ispirato a Machinations [111], ma adattato nel contesto del genere action-gestionale e che permetta di coniugare formalizzazione rigorosa e sketch visivi.

D. Analisi

In seguito alla realizzazione del modello di design, si effettua l'analisi del gioco in modo da ottenere una visione preliminare della possibile esperienza utente a partire dalle scelte effettuate in fase di design.

D. Implementazione

L'implementazione del prototipo per test di telemetria. È importante notare che questo non comprende test preliminari legati alle funzionalità di gioco, che come suggerito da

Fullerton andrebbero effettuati in maniera costante a partire dai primi concept di gioco [2].

F. Test telemetrici

Si procede quindi con l'esecuzione di test utente che permettano di raccogliere dati empirici quantitativi e qualitativi attraverso l'uso della telemetria.

G. Analisi dati

In seguito all'esecuzione dei test si effettua una valutazione delle scelte di design, individuando criticità dell'esperienza di gioco sia in termini di struttura che di bilanciamento. Verranno discussi all'interno del framework alcuni punti critici e strategie risolutive tipiche.

A partire dal punto G si possono quindi effettuare modifiche al concept e design originale fino ad arrivare alla versione definitiva.

4.4 Modellazione del ciclo di gioco action-gestionale

Per l'analisi del design di gioco (compreso nel punto D del framework), è possibile formalizzare l'architettura sistemica e il ciclo di gioco di un titolo action-gestionale.

Il modello, realizzato a partire dalle analisi precedenti effettuate e ispirato ai modelli decisionali OODA [112] utilizzati in ambito militare e ai diagrammi di flusso logici, riassume il ciclo di gioco di un titolo action-gestionale, mettendo in evidenza come ciascun elemento di gioco interagisce all'interno del loop e quali siano i principali sistemi e sottosistemi di gioco che sono coinvolti.

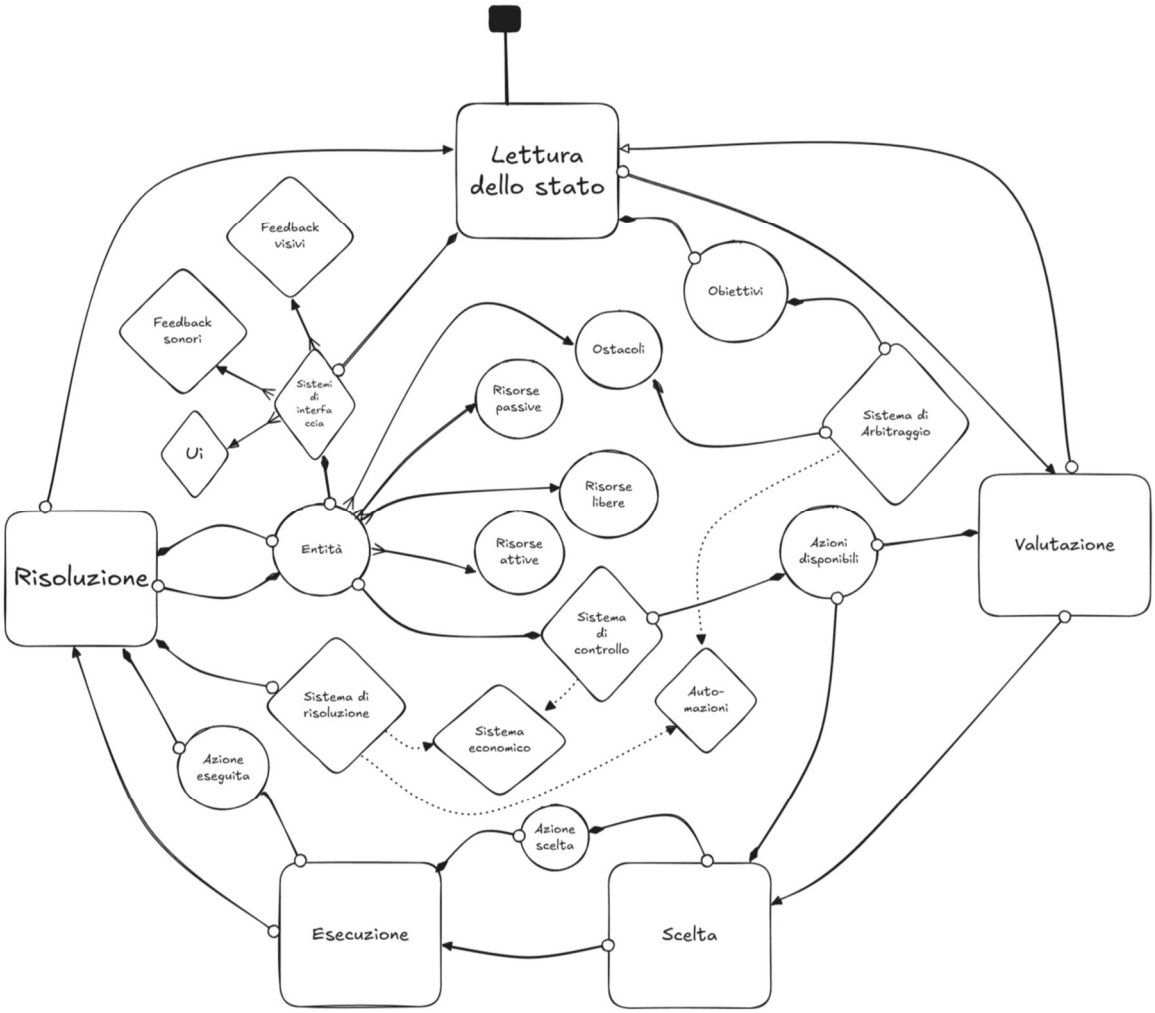


Figura 17: diagramma di modellazione del ciclo di gioco action gestionale.

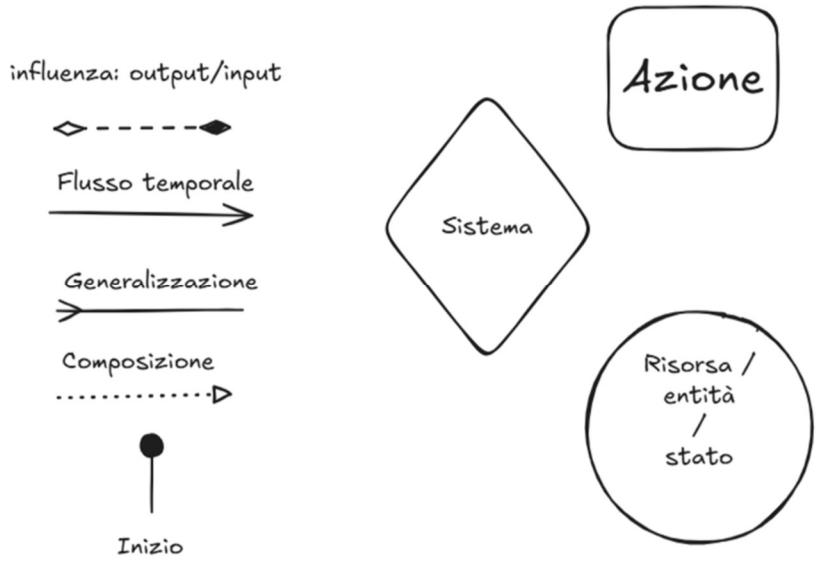


Figura 18: legenda del diagramma (figura 17).

4.4.1 Descrizione della legenda

Azione

Nodo che rappresenta un momento chiave dell'interazione del/dei giocatore/i all'interno del ciclo di gioco (input operativo dell'utente).

Entità

Astrazione di un oggetto rilevante del mondo di gioco (diegetico o sistemico). Non esiste al di fuori del gioco, ma influenza l'andamento della partita ed è utilizzata/trasformata dalle meccaniche.

Sistema

Insieme coordinato di meccaniche, procedure e regole che trasformano input in output all'interno della struttura di gioco.

Linea di flusso

Arco orientato che collega Azioni in ordine temporale/causale, indicando la sequenza di gioco e del loop.

Linea di influenza

Arco che collega Entità e Sistemi ad altri nodi per indicare un rapporto di influenza/controllo. Il cerchio vuoto sul lato sorgente segnala l'output (informazione in uscita); il diamante pieno sul lato destinatario segnala l'input (informazione in ingresso).

Generalizzazione

Indica che l'Entità / sistema può scomporsi in più categorie.

Composizione

Indica che l'entità/sistema va in parte a comporre il destinatario.

4.4.2 Osservazioni sulla classificazione dei sistemi

Sistemi di Interfaccia

Il Sistema di Interfaccia comprende l'insieme delle meccaniche e delle procedure che consentono al giocatore di percepire e interagire con il mondo di gioco. Esso costituisce il punto di contatto tra ciò che avviene a livello sistemico e l'esperienza diretta dell'utente, traducendo lo stato interno del gioco in segnali percepibili e azioni accessibili.

In questo modello viene ulteriormente articolato in tre sottosistemi:

- UI (User Interface): comprende tutti gli elementi e le scelte progettuali legate all'interfaccia utente e alle informazioni visualizzate su schermo. Tra le decisioni che appartengono a questo livello rientrano la tipologia di visuale (e.g. prima persona, isometrica, a volo d'aquila), la presenza di elementi extradiegetici (e.g. indicatori di risorse, pannelli di selezione, barre di stato) e la

gestione del movimento della visuale;

- Feedback visivi: riguardano la rappresentazione visiva di eventi specifici di gioco. A differenza della UI, che presenta informazioni in modo stabile e diretto, i feedback visivi mettono in evidenza criticità. La loro funzione è quella di "far risaltare" informazioni che sono già rappresentati dalla UI;

- Feedback uditivi: hanno la stessa funzione dei feedback visivi ma si basano sul canale sonoro.

Tali componenti risultano particolarmente rilevanti nei giochi appartenenti al genere action-gestionale, poiché supportano la rapidità di percezione e decisione del giocatore e riducendo la latenza cognitiva tra evento, riconoscimento e azione.

Sistema di Controllo

Il Sistema di Controllo comprende l'insieme delle meccaniche, procedure e regole che definiscono quali azioni il giocatore può compiere e in quali condizioni essere risultano disponibili. Questo sistema definisce le possibilità d'interazione ad ogni istante, senza però occuparsi degli effetti specifici che ne derivano.

Esempi tipici includono la possibilità di correre solo se la risorsa stamina del giocatore è sufficiente, o di sparare con un'arma solo in presenza di munizioni. Tuttavia, aspetti come la velocità del salto o il comportamento del proiettile una volta impattato non rientrano in questo ambito.

In termini funzionali, il Sistema di Controllo definisce quindi il ventaglio di opzioni di gioco disponibile in un dato istante e quindi le scelte a disposizione del giocatore.

Sistema di Risoluzione

Il Sistema di Risoluzione comprende tutte le regole e meccaniche che governano il comportamento del gioco "dietro le quinte", determinando come le azioni definite dal Sistema di Controllo ed eseguite poi dal giocatore vengano effettivamente eseguite e risolte all'interno del sistema.

Al suo interno sono incluse parte delle procedure di sistema definite da Fullerton [2], ovvero quei processi che traducono un input del giocatore in un risultato misurabile secondo le leggi interne del gioco.

Ad esempio, se il Sistema di Controllo stabilisce che il giocatore può compiere un salto, il Sistema di Risoluzione definisce a quale velocità il salto avviene e come la forza di gravità ne influenza la traiettoria.

Dunque, se il Sistema di Controllo determina "cosa" il giocatore può fare, il Sistema di Risoluzione determina "come" l'azione viene effettuata e le conseguenze all'interno del mondo di gioco.

Sistema Economico

Il Sistema economico definisce le meccaniche e procedure che regolano i processi di produzione, trasformazione e consumo delle risorse all'interno del gioco, costituendo il nucleo della componente gestionale.

Questo sistema stabilisce come le risorse all'interno del mondo si trasformano e secondo quali regole tali trasformazioni avvengono.

Ad esempio, in Minecraft [79] il Sistema Economico definisce che, combinando due bastoncini e tre blocchi di pietra in un certo ordine, il giocatore ottiene un piccone di pietra, il quale a sua volta abilita l'estrazione di materiali più resistenti come il ferro.

All'interno del modello, il sistema economico va in parte a comporre il Sistema di Controllo e il Sistema di Risoluzione, poiché esso stesso compone parte delle azioni che il giocatore può effettuare durante la partita (e.g. costruzioni, acquisti) e come queste vengono risolte (e.g. quante monete e materiali servono), ma con un maggiore focus sulla parte economica e trasformativa.

Sistema di Arbitraggio

Il Sistema di Arbitraggio raccoglie l'insieme delle regole che definiscono le condizioni di inizio, vittoria e sconfitta dei giocatori, costituendo l'infrastruttura che stabilisce gli obiettivi formali del giocatore. Inoltre, va a definire il comportamento degli Ostacoli,

attraverso le cosiddette Automazioni (meccaniche e procedure che determinano il comportamento autonomo delle Entità di gioco).

In Starcraft 2: Wings of Liberty [69] elementi di questo sistema sono ritrovabili nelle condizioni iniziali di gioco (i giocatori iniziano con dodici costruttori), e nel comportamento a ondate dei nemici presenti nelle missioni della modalità campagna.

La distinzione del Sistema di Arbitraggio dagli altri sottosistemi è cruciale, poiché molte criticità di bilanciamento e di chiarezza progettuale emergono proprio in questa componente: condizioni di vittoria asimmetriche, obiettivi calibrati male o condizioni iniziali sbilanciate, possono compromettere radicalmente l'esperienza e la percezione di equità tra i giocatori.

Entità e Risorse

Le Entità che compongono lo stato del sistema si dividono principalmente in due gruppi:

- Ostacoli: determinati dal Sistema di Arbitraggio, vanno a determinare la sfida del gioco;
- Risorse: entità funzionali ai giocatori per raggiungere il loro obiettivo.

Le risorse vengono a loro volta classificate in base a due dimensioni ortogonali:

- Titolarità (libere vs non libere): le risorse libere non appartengono a nessun giocatore, per essere sfruttate, devono essere acquisite secondo le regole del Sistema Economico. Le risorse non libere sono invece nella disponibilità di uno o più giocatori e possono essere impiegate da chi le possiede;
- Attività (passive vs attive): le risorse passive non hanno alcun tipo di autonomia: la loro funzione è principalmente essere convertite in altre risorse o sbloccare possibilità. Le risorse attive prendono invece parte al Sistema di Controllo o essere controllate da Automazioni: eseguono azioni, richiedono gestione e

influenzano direttamente l'interazione attraverso le regole definite dalle automazioni.

In Dawn of War 1 [55], i punti di controllo non generano requisizione finché non vengono conquistati (e risultano quindi una Risorsa Libera); una volta catturati diventano una Risorsa Non Libera e producono requisizione per il possessore.

In Starcraft 2 [69], i minerali sono una Risorsa Passiva spendibile per costruzioni e unità; le truppe sono invece Risorse Attive, perché possono essere selezionate, mosse, ingaggiate in combattimento e influenzano il Sistema di Controllo.

4.4.3 Analisi del diagramma

Il diagramma evidenzia, ad alto livello, il ciclo decisionale del giocatore e, di conseguenza, fornisce una base solida per l'analisi del core loop in fase di design. In secondo luogo, rende visibile come ciascun gruppo di meccaniche incida sull'esperienza di gioco. Data la natura gestionale del genere, il loop mostrato presenta evidenti parallelismi funzionali con i metodi OODA [112] e PDCA [113] impiegati in contesti manageriali e militari.

Lettura dello stato

Nella fase iniziale il giocatore interpreta le informazioni fornite dall'interfaccia per comprendere lo stato corrente del gioco. Poiché, come si è visto precedentemente, l'accessibilità e la leggibilità delle informazioni sono fattori critici, le scelte di design in questa fase sono volte a garantire una lettura il più possibile immediata e coerente, così da sostenere la continuità dello stato di flow.

Questa fase del diagramma è quindi strettamente collegata ai Sistemi di Interfaccia precedentemente descritti, e un design inadeguato rischia di generare frustrazione poiché non permette all'utente di creare un modello mentale sufficientemente chiaro nel corso della partita.

Valutazione e Scelta

A valle della lettura dello stato, il giocatore valuta cosa fare: pondera obiettivi, contesto e azioni disponibili, e ne sceglie una del ventaglio di opzioni. In questo modello si è scelto di non rappresentare esplicitamente il ritorno alla fase di lettura (la "non-scelta"), poiché in un sistema ad evoluzione continua nel tempo l'inerzia costituisce comunque una scelta implicita sempre disponibile (e.g. attendere, procrastinare, osservare) e alla base del senso di urgenza.

Valutazione e Scelta sono legate principalmente quindi al Sistema di Arbitraggio e al Sistema di Controllo poiché essi determinano le condizioni di vittoria e il ventaglio di opzioni disponibili.

Esecuzione (Controllo) e Risoluzione

L'azione effettuata viene eseguita operativamente attraverso gli input e risolta sulla base del Sistema di Risoluzione, che, come si è visto, ne determina le conseguenze.

Dettaglio cruciale all'interno del diagramma, è che questo passo è influenzato direttamente da:

- Azione scelta dal giocatore;
- Meccaniche di Risoluzione, tra cui quelle Economiche;
- Stato corrente del sistema (Entità e il loro stato).

Compatibilità con la definizione

Se analizziamo più da vicino il diagramma possiamo quindi notare meglio le caratteristiche i), ii) e iii).

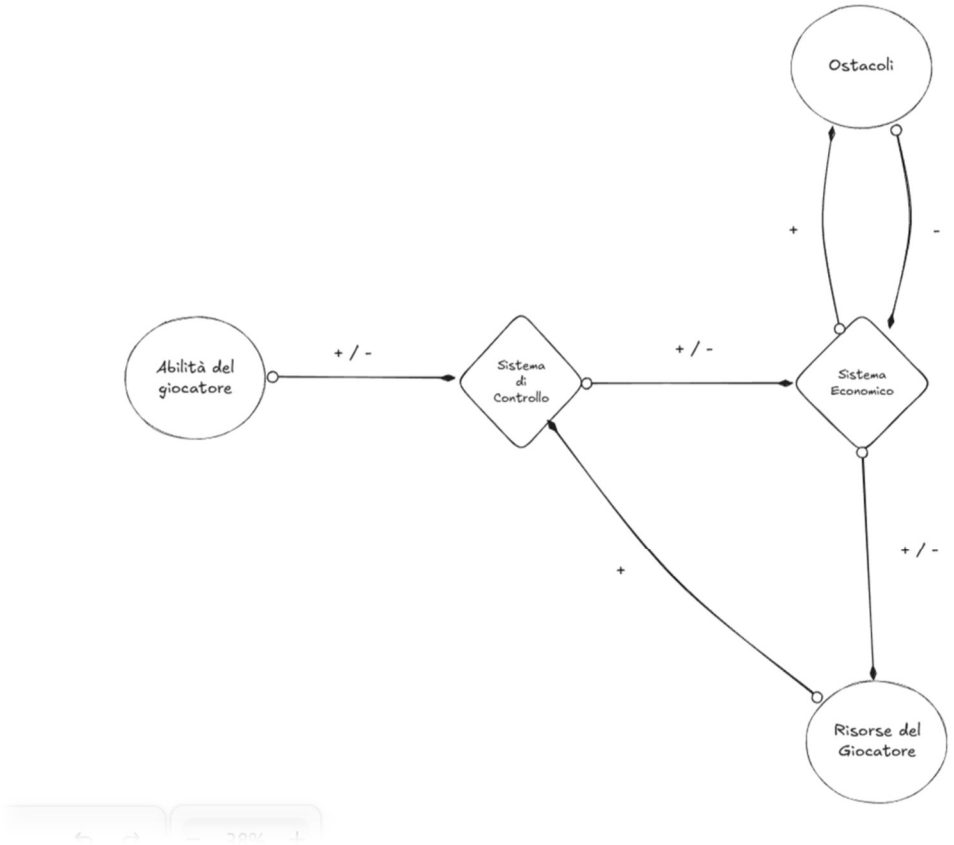


Figura 19: diagramma della retroazione.

All'interno del diagramma sono quindi riconoscibili i punti chiave presenti nella definizione:

- In primis la presenza della componente action, rappresentata dal Sistema di Controllo che, sulla base delle abilità del giocatore in tempo reale, esercita influenza positiva sul Sistema Economico di gioco (e quindi su risorse e ostacoli);

- La presenza della componente gestionale, rappresentata dal controllo del giocatore sul Sistema Economico, e quindi sulle scelte di tipo sistemico;

- La presenza del ciclo di retroazione tra le due componenti, caratterizzata dall'impatto delle abilità del giocatore sul Sistema di Controllo e sul Sistema

Economico, che attraverso le Risorse del giocatore influenzano in maniera retroattiva il primo sistema.

È importante notare che le linee di influenza sono in questo caso da interpretare come positive / negative nei confronti degli obiettivi di gioco del giocatore, e non in termini numerico / quantitativo.

4.5 Modellazione IME e metriche di design

Sebbene il modello introdotto in precedenza evidenzi in maniera efficace la struttura tipica del ciclo di gioco action-gestionale, esso non consente di comprendere con sufficiente precisione in che modo le singole meccaniche influenzino i sistemi e i sottosistemi precedentemente descritti. Inoltre, non permette di collegare l'aspetto qualitativo di vantaggio / svantaggio del giocatore nei confronti degli obiettivi di gioco a parametri quantitativi.

Per superare questi limiti si è deciso di sviluppare una formalizzazione più articolata, ispirata principalmente a due approcci: il già citato framework Machinations [111], ampiamente utilizzato nell'analisi per la sua accuratezza formale e per la capacità di supportare simulazioni precise; e i diagrammi causa-effetto, utili a mettere in evidenza le dinamiche di interdipendenza all'interno di sistemi complessi.

L'impiego diretto di Machinations [111] non è stato ritenuto appropriato in questa fase di lavoro, poiché sebbene rimanga uno strumento di riferimento per la simulazione quantitativa dei sistemi di gioco, la sua struttura avrebbe reso meno agevole la modellazione di un design ancora in fase di evoluzione. Si è pertanto optato per un approccio più snello e flessibile, complementare e non sostitutivo, alle analisi realizzabili tramite Machinations.

4.5.1 Costrutti di IME

Il modello sviluppato, denominato IME (Inputs, Mechanics, Entities), è costruito a partire da tre dei macro-concetti precedentemente introdotti.

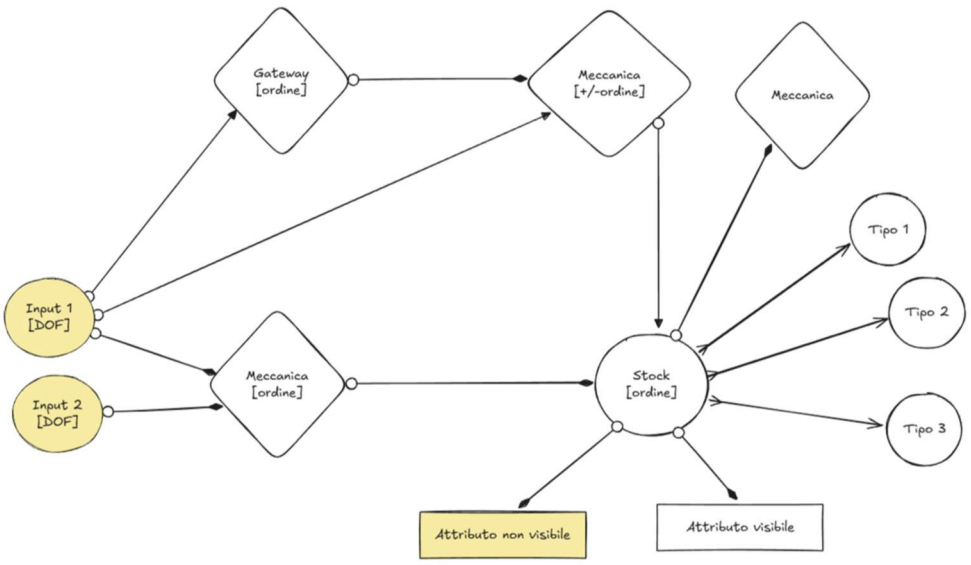


Figura 20: costrutti del modello IME.

Link

I Link collegano i nodi all'interno del diagramma. Anche in questo caso sono caratterizzati da un verso che ne determina il rapporto di influenza. Sono definiti tre tipi di Link.

Figura 21: tipologie di Link del modello IME.

Risorse e Stock

Tutte le Entità presenti nel mondo di gioco sono rappresentate tramite Stock, ossia contenitori che descrivono la quantità disponibile di una determinata risorsa. Questa

scelta deriva dal fatto che, all'interno dello spazio di gioco, possono esistere più istanze della medesima risorsa (e.g. monete raccoglibili, unità, materiali).

Nel diagramma, lo Stock non rappresenta necessariamente il numero esatto di istanze presenti, bensì un ordine di grandezza che descrive l'intervallo entro cui il valore della risorsa può variare. In questo modo il modello rimane leggibile ed è possibile focalizzarsi sulla visione d'insieme anziché su conteggi esatti, che in fase di prototipazione risultano spesso variabili.

I valori di grandezza assegnabili sono:

- Ordine 1: valori compresi tra 0 e 5;
- Ordine 1.5: valori compresi tra 5 e 10;
- Ordine 2: ordine delle decine;
- Ordine 3: ordine delle centinaia;
- Ordine 4: ordine delle migliaia.

I valori numerici degli Stock non sono da interpretare in maniera rigida. È possibile, per esempio, assegnare a uno Stock un Ordine di 3.8 qualora il Designer ritenga che il valore della Risorsa si avvicini più alle migliaia che alle centinaia.

A ciascuno Stock sono inoltre associabili gli Attributi, che ne descrivono le proprietà rilevanti per il sistema (e.g. posizione, valori di stato). Come si vedrà in seguito è importante la distinzione tra Attributi Visibili e Nascosti.

Nel modello, gli Stock possono essere organizzati gerarchicamente attraverso un Link di Generalizzazione, che instaura una relazione di ereditarietà tra una risorsa padre e una o più risorse figlie. In questa configurazione, le risorse figlie ereditano dallo Stock padre sia l'Ordine di Grandezza sia gli Attributi fondamentali.

Se presenti, è possibile introdurre Entità senza Stock ignorando l'Ordine di Grandezza. Questo può essere utile al designer in fase di modellazione per rappresentare elementi che influenzano il sistema, senza però andare a impattare le metriche di design che verranno introdotte.

Input

Rappresentano tutte le procedure fisiche legate all'interazione dell'utente con il mondo di gioco (e.g. click del pulsante sinistro del mouse, rotazione della levetta destra del joypad).

A ciascun Input è associato un grado di libertà (DOF) che ne influenza la complessità di interazione:

- 1-DOF: input binari, dotati di due soli stati mutuamente esclusivi (attivo/inattivo). Rientrano in questa classe i pulsanti digitali, come quelli presenti su tastiere e mouse;

- 2-DOF: input monodimensionali, rappresentabili lungo un singolo asse continuo di stati intermedi (e.g. slider o trigger analogici);

- 3-DOF: input bidimensionali, modellabili come un punto all'interno di un piano. Ne sono esempi le levette analogiche dei controller tradizionali;

- 4-DOF: input tridimensionali, utilizzati in contesti immersivi come la realtà virtuale, in cui la posizione nello spazio e il movimento generano un flusso di dati su tre dimensioni.

Esattamente come per gli Stock è possibile associare a ciascun Input determinati Attributi. Inoltre, qualora un Input sia in realtà caratterizzato dalla combinazione di più DOF (e.g. se al click sinistro del mouse è associata una posizione su schermo 2D), questo è rappresentabile andando semplicemente a sommare i DOF.

Meccaniche e Gateway

Ogni aspetto funzionale del gioco, in base alle analisi precedenti, è associabile a una Meccanica. All'interno di IME, ogni meccanica è rappresentata come un nodo logico, che applica una funzione di trasformazione ai propri ingressi e produce uno o più effetti in uscita.

Una meccanica può ricevere in ingresso attraverso i Link tre tipologie di elementi:

- Input fisici del giocatore provenienti da un elemento Input;

- Stock e suoi attributi: quando uno Stock è utilizzato come Input, la meccanica riceve non solo il valore corrente della risorsa, ma anche gli attributi a essa associati. Sarà la funzione logica interna alla meccanica a determinare quali attributi influenzano il comportamento e in che modo;

- Output di un'altra meccanica: in questo caso, la meccanica che fornisce l'output agisce da Gateway, ovvero seleziona e filtra le informazioni dirette verso un'altra meccanica. Un Gateway stabilisce quindi una relazione funzionale tra due meccaniche introducendo un vincolo.

In uscita, una Meccanica può essere collegata a:

- Stock: la meccanica modifica la quantità della risorsa contenuta nello Stock. In questo caso sul Link viene anche specificata la Granularità con cui la Meccanica agisce, che può essere positiva o negativa in senso numerico;

- Attributi di uno Stock: la Meccanica non interviene sul numero di istanze della risorsa, ma ne modifica uno o più Attributi andando così a influenzare il comportamento futuro di una o più istanze. In questo caso la Granularità indica l'ordine quantitativo di istanze dello Stock su cui avviene il cambiamento;

- Meccanica: nel caso dei Gateway. In questo caso non viene indicata la Granularità sul Link che collega le due Meccaniche.

In questo modo sono distinguibili all'interno del modello le tre tipologie di Meccaniche definite nella tassonomia dei paragrafi precedenti:

- Meccaniche di Controllo: Meccaniche foglia su cui il giocatore può agire direttamente attraverso gli Input o indirettamente attraverso un Gateway;

- Meccaniche Economiche: Meccaniche foglia che vanno ad agire sul valore di uno Stock attraverso una Granularità positiva o negativa;

- Automazioni: Meccaniche che agiscono sull'attributo di uno Stock ma che non sono raggiungibili direttamente dagli Input o un Gateway, e non sono quindi controllabili dal giocatore.

4.5.2 Metriche di design

Sulla base del modello IME è possibile quindi definire un insieme di metriche di design finalizzate a stimare, in fase progettuale, alcune proprietà del sistema di gioco.

Le tre metriche definite di seguito non hanno la pretesa di essere esaustive, ma servono a introdurre una prima quantificazione dei diversi aspetti individuati all'interno del genere.

Fattore di Multitasking (fattore MLTT)

Questa prima metrica si ricollega direttamente al punto i) della definizione del genere, ovvero alla componente action e al modo in cui l'interazione fisica con gli Input si traduce in "carico cognitivo". Studi sull'esecuzione di compiti simultanei evidenziano infatti come i costi prestazionali aumentino con l'incremento del carico [117].

Per rappresentare attraverso i costrutti di IME la complessità e la concorrenza di azioni che il giocatore è chiamato a gestire all'interno del gioco, si introduce quindi il Fattore di Multitasking.

Viene definito Fattore di Multitasking (fattore MLTT) associato a un determinato Input e a un determinato Stock Target il valore:

$$MLTT_{mech} = (DOF_{input} \cdot D_{mech} \cdot N_{attr})^{\frac{ord}{gran}}$$

Dove:

- DOF_{input} rappresenta il grado di libertà associato all'Input che controlla la Meccanica di Controllo associato allo Stock Target;

- D_{mech} rappresenta il numero di Meccaniche (inclusa quella foglia) prima di arrivare allo Stock Target. Stock intermedi, se presenti, vengono esclusi da questo valore;

- N_{attr} indica il numero di Attributi Visibili dello Stock Target;

- ord è l'Ordine associato allo Stock Target;

- $gran$ è la granularità della Meccanica di Controllo che agisce sullo Stock.

Il fattore MLTT si lega quindi a:

- Dimensionalità dell'Input: una Meccanica di Controllo caratterizzata da un numero minore di Gradi di Libertà è per definizione più semplice da controllare di una Meccanica a più dimensioni. Questo aspetto si ricollega al limite delle risorse attentive e motorie nella gestione simultanea di più canali di controllo, come discusso nei modelli di Multiple Resource Theory e negli studi sul multitasking [118];

- Numero di Attributi Visibili della Risorsa: maggiore sono i parametri di stato che il giocatore deve considerare per agire su una Risorsa, maggiore è il livello di complessità della Meccanica;

- Lunghezza della catena di Meccanica: se una Meccanica di Controllo è caratterizzata da numerosi Gateway (come numerosi sottomenu), il giocatore deve compiere più passaggi per poter compiere l'Azione e navigare diverse

sezioni;

- Rapporto tra Ordine dello Stock e Granularità della Meccanica: questo è il fattore che pesa di più all'interno della metrica. Infatti, se il giocatore si ritrova a dover controllare le singole istanze di uno Stock, dovrà ripetere molteplici volte lo stesso passaggio per controllarle tutte, e di conseguenza la Meccanica di Controllo peserà di più sulle abilità motorie del giocatore.

Presenza di Feedback Loop

La seconda metrica si riferisce alle catene di retroazioni presenti all'interno dell'economia di gioco. Nel contesto della dinamica dei sistemi, gli Stock sono comunemente interpretati come la "memoria" del sistema, mentre i Feedback Loop all'interno di esso determinano la dinamica temporale complessiva [119]. È quindi ragionevole assumere che la presenza e analisi di Feedback Loop all'interno del sistema di gioco permetta di coglierne informazioni sull'evoluzione e permanenza temporale.

I Feedback Loop vengono definiti in questo caso dalle catene di Meccaniche e Stock che, a partire da una Meccanica di Controllo, tornano alla Meccanica di partenza.

In questa metrica:

- Un loop è definito come una sequenza chiusa di nodi (Meccaniche e Stock) connessi dai Link orientati coerenti con il flusso causale;

- Vengono escluse le sequenze interamente contenute all'interno di altre sottosequenze per evitare ridondanze;

- Vengono anche scartate le sequenze che comprendono soli due nodi (collegamento diretto tra Stock e Meccanica).

Forza della retroazione

A partire dall'insieme dei Feedback Loop identificati, è possibile definire una terza metrica che mette in relazione l'aspetto iii) della definizione con il diagramma IME: il numero di Feedback Loop in cui un determinato Stock Target è coinvolto.

Poiché gli Stock Target sono per definizione quelli direttamente legati alle azioni del giocatore, mentre i Feedback Loop e gli Stock rappresentano principalmente l'aspetto sistemico ed economico, l'analisi di quali Risorse siano maggiormente coinvolte all'interno dei cicli permette di comprendere come e con quale intensità la retroazione tra le due componenti avviene.

Normalizzazioni

Per garantire la coerenza delle metriche in seguito alla creazione del diagramma, all'interno di IME vanno applicate alcune normalizzazioni:

- Tutte le meccaniche tranne i Gateway devono agire su almeno uno Stock o un suo Attributo. In questo modo si evitano Meccaniche "Spurie", ovvero che non hanno fondamentalmente impatto all'interno del gioco;

- Nel caso dei Gateway, questi devono essere raggiungibili da almeno un Input;

- Solo gli Attributi influenzati da una Meccanica vengono rappresentati, in modo da evitare un aumento esponenziale degli Attributi di gioco rappresentati e quindi un aumento eccessivo del fattore MLTT.

Osservazioni su IME

È importante notare che il modello sopra definito non ha come obiettivo quello di fornire un'analisi numerica esatta del modello di gioco, bensì di offrire al game designer una guida flessibile per ottenere una visione complessiva sulle scelte di design e del loro impatto sui tre aspetti principali del genere.

Le metriche derivate vanno quindi interpretate come valori di supporto nelle iterazioni in fase di design e nell'analisi dei dati di test, più che come un'analisi rigorosa (già messa a disposizione da framework come Machinations [111]).

Diventa quindi responsabilità del game designer selezionare quali aspetti di gioco modellare (soprattutto per titoli comprendenti numerose Meccaniche e Stock), facendo leva sulle tecniche di normalizzazione per far sì che il modello sia coerente.

In questo modo, grazie alla sua esperienza, il game designer può decidere di approfondire la modellazione degli elementi che ritiene critici per i player goals definiti, individuando ad esempio aree del gameplay in cui una delle tre metriche presenti valori marcatamente sbilanciati.

4.5.3 Metriche di telemetria

Questo tipo di metriche riguarda specificamente i punti f) e g) del framework. Vanno infatti a costituire il mezzo con cui è possibile raccogliere dati empirici sull'esperienza di gioco dell'utente.

Azioni Per Minuto (APM)

La peculiarità di questo tipo di metriche è che, pur essendo specifiche per ciascun titolo, all'interno di molto sottogeneri (e.g. RTS, MOBA) se ne può individuare una ricorrente: gli APM (Azioni Per Minuto) del giocatore.

Nel caso degli RTS e dei MOBA, gli APM sono diventati una metrica informale ma centrale per valutare il livello di abilità: come osserva Rauch nella sua analisi storica del genere, i migliori giocatori di Starcraft sono in grado di sostenere per lunghi periodi tra le 300 e 400 azioni per minuto, e la "velocità di esecuzione" è stata di fatto definita come uno dei principali marcatori di abilità, per cui i MOBA stessi ereditano gran parte della loro enfasi sulla rapidità e il multitasking [120].

In questa prospettiva, è ragionevole pensare che gli APM possano essere letti come una misura sintetica dell'intensità del gioco e, se correlata con l'esito della performance, delle abilità esecutive del giocatore.

Complessità Lempel-Ziv (LZC) e Complessità Comportamentale

La Lempel-Ziv Complexity (LZC) fornisce, per una data sequenza composta da verbi elementari, un indice della varietà e imprevedibilità delle sottosequenze presenti nella catena principale.

In questo contesto, data la catena di azioni del giocatore, l'assunto è che a una maggiore diversificazione dei verbi impiegati (cioè delle singole azioni), e una minore ripetitività delle loro combinazioni, corrisponda a un comportamento più articolato dell'utente.

La Complessità LZ è calcolata, nella sua forma normalizzata, come:

$$C_{LZ}(S) = \frac{C(n)log_k(n)}{n}$$

Dove:

- S è la sequenza dei verbi (catena delle azioni) prodotti dal giocatore;
- n è la lunghezza della sequenza, ossia il numero totale di azioni osservate;
- k indica la dimensione dell'alfabeto dei verbi, cioè il numero di azioni distinte utilizzate dal giocatore in S;
- $c(n)$ è il numero di "frasi" o pattern distinti individuati, che rappresentano i punti in cui la sequenza introduce informazione nuova all'interno della catena.

L'obiettivo di questa metrica è quello di fornire, assieme agli APM del giocatore, informazioni empiriche sul suo comportamento all'interno del gioco.

Metriche di successo

Le metriche di successo mirano a stimare, in ogni istante della partita, la probabilità che il giocatore raggiunga il proprio obiettivo. Poiché quest'ultimo varia da titolo a titolo (e.g. vittoria di uno scontro, controllo territoriale, sopravvivenza, punteggio), non è possibile definire una metrica di successo unica e generica per l'intero genere.

Ciò che risulta utile, tuttavia, è l'analisi della variazione nel tempo di queste metriche in relazione alle azioni del giocatore. L'osservazione combinata di andamento della metrica e scelte effettuate, può fornire indicazioni rilevanti sul design di gioco, per esempio:

- Individuare situazioni in cui il giocatore si ritrova a lungo in marcato svantaggio (criticità nota come "Difficoltà di Rimonta" [2]) oppure, al contrario, scenari in cui mantiene un vantaggio crescente e difficilmente colmabile dagli avversari ("Runaway Leader" [2]);

- Identificare azioni o strategie che producono variazioni della metrica di successo in misura nettamente superiore alle alternative disponibili, configurando di fatto delle "Scelta Ovvie" [2], o viceversa delle "Alternative Inferiori" [2];

- Calibrare dinamicamente ostacoli, sfide e risorse, in funzione del valore corrente della metrica di successo, mantenendo la partita in un range di sfida desiderato.

5. Applicazione del Framework a *Wolves and Sheeps*

L'obiettivo di questo capitolo è applicare il workflow proposto dal framework al prototipo di *Wolves and Sheeps* (di seguito WS), titolo action-gestionale sviluppato ad hoc come caso di studio. In particolare, WS viene scomposto attraverso le varie fasi del framework, incluse la modellazione IME e alcune prime osservazioni sulle metriche di design.

5.1 Costruzione del Design

5.1.1 Concept di gioco

WS è un gioco 2D top – down strutturato in livelli discreti, denominati Scenari, in cui il giocatore impersona un pastore chiamato a far pascolare il proprio gregge in aree dedicate (i Pascoli), accumulando Punti Pascolo (di seguito PP) e proteggendo al contempo da Lupi e minacce ambientali.

I singoli Scenari sono divisi in Zone più piccole, tali per cui per passare a quella successiva è necessario accumulare un certo ammontare di PP. Al raggiungimento della soglia, il giocatore, può accedere alla Zona successiva dello Scenario, fino all'ultima.

A suo supporto dispone di diversi Strumenti / Oggetti, tra cui il Fucile per difendere le Pecore dai Lupi, e il Bastone per non lasciar disperdere il gregge.

Ad ostacolarlo interverranno invece diversi fattori:

- Le Pecore, anziché pascolare, possono iniziare a vagare;

- Minacce esterne come i Lupi possono intervenire aggredendo e disperdendo il gregge;

- Ostacoli ambientali come Dirupi e Pozze minacciano le Pecore che si sono disperse;

Alla fine di ogni Scenario, i PP accumulati vengono convertiti in Monete, che possono essere spese nel negozio per acquistare nuovi Oggetti e Pecore aggiuntive. La sconfitta avviene quanto il giocatore perde tutte le Pecore prima di aver completato la sequenza di Scenari.

5.1.2 Analisi del concept

L'impianto definito nel concept di gioco introduce immediatamente:

- Una risorsa controllabile, le Pecore, che nell'economia di gioco permette di estrarre la risorsa chiave, i PP;

- Una risorsa intermedia, i PP, che oltre ad essere convertibili in Monete sono necessari per completare lo Scenario;

- Una valuta persistente, le Monete, che crea le condizioni per una componente gestionale con memoria.

Inoltre, emergono alcuni Player Goals primari, tra cui:

- Massimizzare i PP: Il giocatore trae soddisfazione nella raccolta dei punti e nel completare Scenari e Zone;

- Mantenere il controllo del gregge: il giocatore deve monitorare costantemente la posizione e lo stato delle Pecore, intervenendo quando si disperdono o vengono attaccate;

- Ottimizzare gli investimenti nel Negozio: tra uno Scenario e l'altro, il giocatore deve scegliere come investire le Monete, influenzando in modo irreversibile le partite successive.

Questi obiettivi si sovrappongono alle caratteristiche chiave del genere action-gestionale identificate nel capitolo 4: pressione temporale, multitasking, prioritizzazione continua delle azioni e memoria del sistema.

Inoltre, il concept è modellabile, ad alto livello, attraverso il seguente diagramma di retroazione:

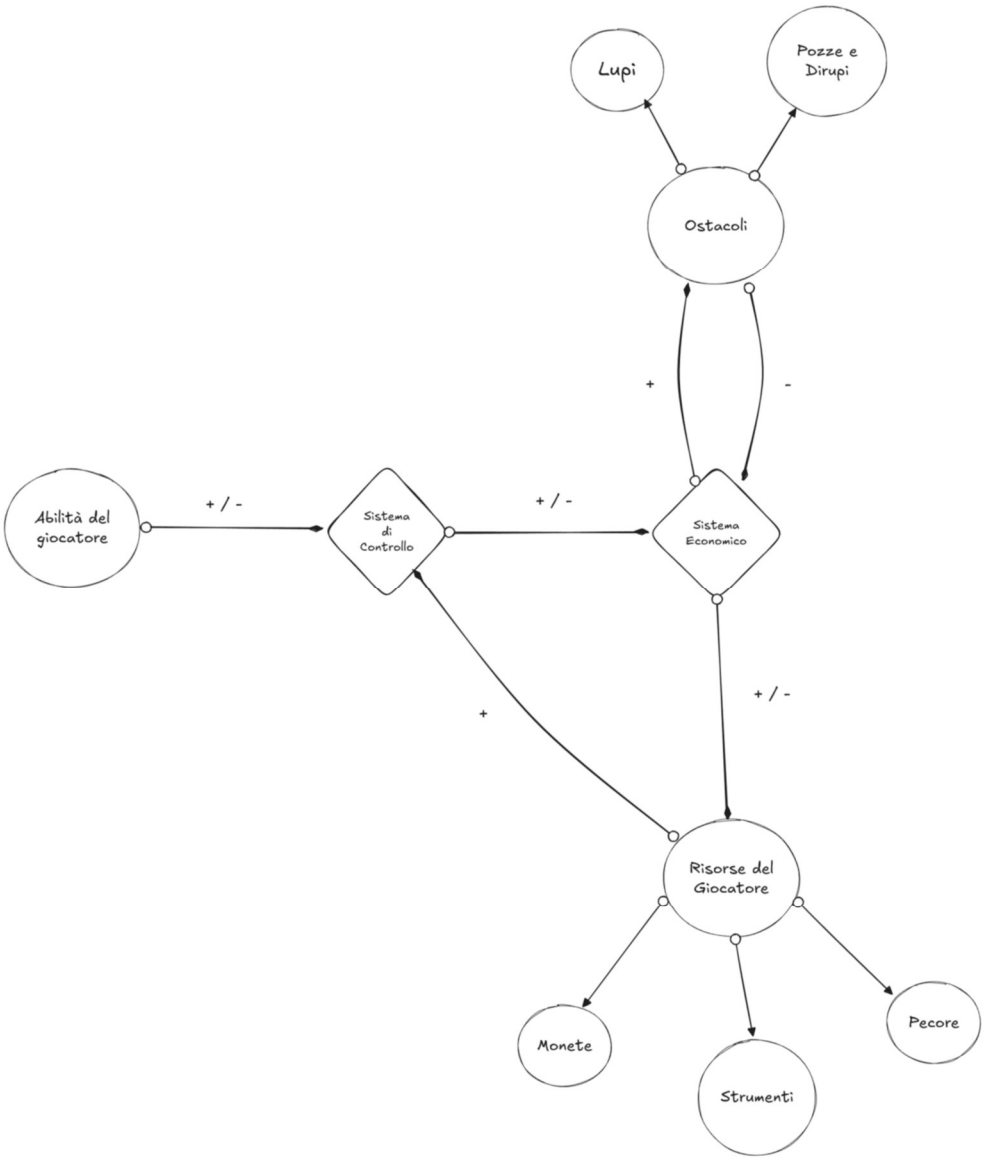


Figura 21: diagramma della retroazione di WS.

Che evidenzia le componenti i), ii) e iii) della definizione introdotta nel paragrafo 4.5:

- La componente action i) rappresentata dal controllo in tempo reale del Pastore, dallo spostamento nel mondo di gioco e dall'uso degli Strumenti e Oggetti per reagire a Lupi e dispersioni delle Pecore;

- La componente gestionale ii), che si realizza attraverso le scelte su come muovere il proprio gregge, su come spendere le Monete a fine Scenario, e su quando cambiare Zona;

- La retroazione forte tra le due componenti, che emerge dal fatto che le prestazioni lato action (e.g. Pecore salvate, Lupi sconfitti) determinano la quantità di risorse disponibili per le decisioni gestionali, che a loro volta impattano la difficoltà dell'azione (e.g. troppe poche Pecore per avanzare, Strumenti vecchi).

In sintesi, il gioco soddisfa la definizione di gioco action-gestionale ed è possibile applicare il workflow proposto dal framework.

5.1.3 Definizione del core loop di gioco

Per maggiore chiarezza, nell'analisi di WS si scompone il loop come definito da Fullerton [1] in tre livelli:

- Micro – loop: gestione e difesa del gregge all'interno di una singola Zona e conseguente guadagno di PP;

- Macro – loop: composto dal completamento sequenziale delle Zone che compongono uno Scenario singolo;

- Meta – loop: dato dall'alternanza tra Scenari e Negozio, con l'accumulo e spesa di Monete.

Questa suddivisione è coerente con il ciclo decisionale “Lettura dello stato → Valutazione → Scelta → Esecuzione → Risoluzione” introdotto nel capitolo 4, esteso su orizzonti temporali e granularità differenti.

Micro loop – fase principale

All’interno del micro – loop di WS il giocatore guida e difende il proprio gregge nella Zona in cui si trova, cercando di massimizzare il numero di PP ottenuti tramite la sosta delle Pecore nei Pascoli e cercando di minimizzare il numero di quelle perse a causa dei Lupi e dei pericoli ambientali.

In questa fase, in pratica:

- Lettura dello stato – Il giocatore osserva la posizione e lo stato delle Pecore e dei Lupi tramite la UI e i feedback);

- Valutazione – Valuta quali minacce siano più urgenti (e.g. Lupi in arrivo, Pecore disperse vicino a un Dirupo);

- Scelta – Sceglie di muoversi, cambiare Strumento, usare il Bastone o altri Oggetti a sua disposizione. Nel caso in cui non intervenga in alcun modo, l’evoluzione del gioco procede secondo le Automazioni e Meccaniche di sistema del gioco;

- Esecuzione – Esegue l’azione attraverso gli Input fisici (e.g. WASD, mouse, tasti numerici / rotellina);

- Risoluzione – L’azione viene risulta sulla base delle Meccaniche di gioco e lo stato corrente (e.g. uso del Bastone su una Pecora Dispersa, Proiettile che ferisce un Lupo).

Questo ciclo è continuo e scandito da eventi generati dalle Automazioni (e.g. generazione dei Lupi, dispersione delle Pecore), ponendo quindi il micro – loop come fulcro della componente action, del multitasking e della pressione temporale.

Macro loop

Il macro – loop organizza sequenze di micro – loop all'interno di uno Scenario.

Una volta guadagnati PP sufficienti in una Zona, il giocatore può:

- Permanere nella Zona corrente per sfruttare ulteriormente i Pascoli rimanenti e guadagnare PP bonus, rischiando di perdere ulteriori Pecore in quella Zona;

- Avanzare alla Zona successiva, chiudendo il macro – loop e accedendo al Negozio.

Il macro – loop di gioco, quindi, pone al giocatore di fronte a scelte di gestione del rischio e di ritmo.

Meta loop

Nel meta – loop, in seguito alla conversione dei PP in Monete, il giocatore accede al Negozio e può:

- Spendere le proprie Monete per acquistare nuove Pecore;

- Comprare nuovi Oggetti / Strumenti;

- Decidere quali di questi portare nello Scenario successivo.

Le scelte di acquisto determinano le condizioni iniziali del macro – loop successivo (e.g. dimensione del gregge, dotazione di oggetti, Monete residue), costituendo fattori importanti di Memoria all'interno del design di gioco.

5.2 Scomposizione in Sistemi e Meccaniche

Applicando la tassonomia introdotta nel capitolo 4, il design di WS può essere scomposto nei seguenti sottosistemi e Entità: Sistema di Interfaccia, Sistema di Controllo, Sistema di Risoluzione, Sistema Economico, Sistema di Arbitraggio, Risorse,

Ostacoli e Automazioni. Di seguito vengono analizzati i principali elementi di gioco in base a tale struttura.

5.2.1 Sistema di Arbitraggio

In WS, il Sistema di Arbitraggio definisce in particolare:

- Obiettivi e condizioni iniziali all'interno del meta – loop di gioco, ovvero:
    - Numero e sequenza degli scenari da completare;
    - Condizioni di vittoria e di sconfitta (perdita di tutte le Pecore);
    - Dotazione iniziale del giocatore: numero di Pecore, Monete inziali, Strumenti disponibili.
- I parametri che regolano il macro – Loop di gioco, in particolare:
    - Numero di Zone presenti nei singoli Scenari;
    - Soglia di PP per completare ciascuna Zona dello Scenario;
    - Posizionamento iniziale delle Pecore.
- Il comportamento delle Pecore e dei Lupi, corrispondenti in particolare alle Automazioni presenti all'interno del Micro – Loop di gioco. Il giocatore e le Pecore non possono inoltre superare i limiti imposti dai bordi della mappa e dalla Zona corrente.

Queste regole, in combinazione con le Automazioni descritte più avanti, determinano il grado di sfida e le condizioni in cui si svolge la partita.

5.2.2 Sistemi di UI e feedback del micro – loop di gioco

Il micro – loop è quello in cui avviene principalmente la combinazione di componente action e gestionale. Di conseguenza, il Sistema di Interfaccia assume un ruolo centrale per sostenere la Lettura dello Stato e sostenere lo stato di flow, evitando quindi sforzi superflui al giocatore.

Visuale

WS adotta una visuale top – down ortogonale con camera centrata sul pastore, ad inseguimento fluido e limitata ai confini della Zona corrente.

Barra dei Punti Pascolamento

In alto a destra è presente una barra di avanzamento che mostra due valori:

- I PP accumulati nello Scenario;
- La soglia di PP necessaria per sbloccare la Zona successiva.

Per permettere al giocatore di tenere traccia di eventuali PP residui nello Scenario, si è scelto di mantenere comunque il grado di riempimento dell'indicatore relativo ai PP complessivi dello Scenario, e non della singola Zona.

Cintura

Una serie di riquadri mostra gli Strumenti disponibili, con un cursore che evidenzia quello attualmente selezionato. Questo elemento rende esplicito il ventaglio delle azioni accessibili dal Sistema di Controllo, riducendo gli sforza di memoria di lavoro per il giocatore.

Indicatori di "Fuori Schermo"

Questi indicatori compaiono nel momento in cui una Pecora o un Lupo escono dal campo visivo, permettendo al giocatore di tenere sempre traccia della loro direzione.

Senza di essi il giocatore si troverebbe a fronteggiare minacce impreviste (soprattutto Lupi che minacciano Pecore ai bordi della telecamera, e quindi solamente visibili all'ultimo), e a dover ripercorrere costantemente lo Scenario in cerca di Pecore smarrite. Dinamiche come queste non sono state ritenute funzionali alla realizzazione dei player goals.

Icona e colori di Stato delle Pecore e Lupi

Allo stato delle Pecore corrisponde ad ogni istante un colore e un'icona posizionati sopra all'animale. L'assenza di questi indicatori renderebbe praticamente impossibile da parte del giocatore capire se le Pecore si stanno muovendo nel modo desiderato oppure se si stanno disperdendo, compromettendo la fase di Lettura dello stato da parte del giocatore.

Allo stesso modo, ai lupi viene assegnato un colore in base a se sono stati colpiti o se stanno attaccando.

Icona e suono di ottenimento dei PP

Ogni volta che una Pecora pascola su un Pascolo attivo, un indicatore numerico rosso segnala i PP / secondo guadagnati, accompagnato da un suono distintivo e ripetuto da ogni Pecora. L'accumulo di questi feedback fornisce al giocatore un'indicazione indiretta sul numero di Pecore attualmente produttive e crea una soddisfazione sensoriale proporzionale alla dimensione del gregge.

Versi degli animali

Cambiamenti di stato critici (morte di una pecora, attacco di un lupo, "richiamo" del Bastone) sono associati a suoni di feedback specifici. Questo parallelo consente di individuare rapidamente eventi fuori campo visivo, contribuendo a ridurre il tempo tra evento e reazione nel ciclo decisionale.

Nel complesso, quindi, si è cercato di rendere coerenti gli elementi visivo – uditivi di WS coerenti con quelli individuati per il genere e puntando a massimizzare la trasparenza dello stato del sistema.

5.2.3 Entità e Risorse di WS

Le Entità che compongono lo stato del gioco, in IME vengono modellate come Stock. Di seguito vengono individuate le principali presenti in WS.

Pastore

Il pastore è l'avatar controllabile del giocatore è caratterizzato da Attributi Visibili, tra cui la propria Transform. Essendo direttamente controllato dal giocatore, rappresenta uno Stock Target degli input relativi alla Meccanica Movimento, con Ordine 1.

Oggetti / Strumenti

Sono il principale tipo di risorsa con cui il giocatore può controllare e difendere il gregge, agendo sul sistema di gioco.

Ad ogni istante, il giocatore può avere equipaggiato un solo Strumento, modellato in IME tramite lo Stock "Oggetti a Disposizione" e il relativo Attributo booleano "Equipaggiato". Essendo disponibili diversi tipi di Strumenti e categorie, da questo Stock discendono:

- Armi da Fuoco: utilizzabili dal giocatore per difendere il gregge dai lupi;
- Bastoni: utilizzabili per impartire ordini alle Pecore adiacenti al Pastore;
- Oggetti Speciali (specificati in seguito): utilizzabili come gli altri, ma caratterizzati da un comportamento particolare e unico.

Lo Stock "Oggetti a Disposizione" risulta essere Target per gli input relativi alla Meccanica di Cambio Oggetto.

Pecore

Le Pecore sono il principale collegamento tra azione e gestione all'interno di WS. In IME sono rappresentate da uno Stock con ordine 2 (decine di unità) e attributi visibili di

Transform e Stato. Risultano essere Risorse Attive e Non Libere, infatti appartengono al giocatore, anche se il loro comportamento è governato in larga parte da Automazioni, e richiedono interventi di controllo diretti tramite Bastone e posizionamento del Pastore. Per questo motivo, le Pecore risultano anche risorsa Target degli Input legati al Bastone.

Punti Pascolo (PP)

I PP rappresentano, come già evidenziato, una risorsa che quantifica un Obiettivo intermedio per il giocatore (progresso all'interno dello Scenario), sia una "pseudo-valuta" che viene sistematicamente convertita in Monete.

Dal punto di vista del framework, si tratta di una Risorsa Passiva Non Libera: il giocatore non può agire direttamente sulla quantità di PP, ma solo influenzarne la produzione indirettamente (ovvero il controllo delle Pecore). Il loro Stock è caratterizzato da un Ordine compatibile con i valori richiesti per completare gli Scenari, tipicamente 3 (migliaia), coerente con il fatto che l'Ordine delle Pecore è 2 e che ogni Pecora produce ogni secondo una quantità di PP di Ordine 2. Il risultato è che quindi il completamento dello Scenario avviene entro alcuni minuti massimo.

Monete

Le Monete sono la valuta principale per gli acquisti nel Negozio.

Sono accumulate al termine di ciascuno Scenario convertendo i PP ottenuti. A differenza dei PP, le Monete persistono tra gli Scenari, costituendo come già visto il principale fattore di memoria economica assieme al numero di Pecore e agli Strumenti acquistati.

A rigore, nel modello proposto dal framework, le Monete vanno considerate Risorse Passive Non Libere: non hanno comportamento autonomo, non partecipano direttamente al Sistema di Controllo, ma sono convertibili in Pecore e Oggetti tramite Meccaniche Economiche. Definirle "attive" sarebbe fuorviante rispetto alla tassonomia adottata.

Pascoli

I Pascoli sono Entità stazionarie che fungono da sorgenti di PP quando occupate da Pecore brucanti.

Nel modello IME possono essere rappresentati come Risorse Libere Passive: non appartengono intrinsecamente al giocatore, ma possono essere "sfruttati" interagendovi attraverso le Pecore e la loro Automazione.

5.2.4 Ostacoli e Minacce

In linea con il framework, gli Ostacoli sono entità determinate dal Sistema di Arbitraggio che definiscono la sfida del gioco. All'interno del gioco ne sono definiti due.

Lupi

I Lupi sono generati dallo Scenario in base a fattori come Tempo Passato e Sezione attuale e, attraverso la loro Automazione, minacciano direttamente il gregge, attaccando e disperdendo le Pecore. Rappresentano quindi l'Entità – Ostacolo principale e sono responsabili della maggior parte della pressione temporale all'interno del micro – loop: senza di essi, il giocatore potrebbe prendersi un tempo virtualmente illimitato per ottimizzare il posizionamento del gregge.

Essendo i Lupi generati singolarmente, hanno Ordine 1, e risultano Risorsa Target per l'uso delle Armi da Fuoco.

Pozze e Dirupi

Terreni come dirupi e pozze rappresentano pericoli ambientali che causano la morte immediata delle Pecore disperse che vi transitano.

Questo tipo di ostacolo è reso particolarmente rilevante dal comportamento casuale delle Pecore, che possono disperdersi e, in fuga dai Lupi, andare incontro a catene di eventi potenzialmente letali.

Lupi e terreni pericolosi costituiscono quindi Ostacoli interdipendenti: l'azione dei Lupi aumenta la probabilità che le Pecore interagiscano con i pericoli ambientali, amplificandone gli effetti.

5.3 Meccaniche, Automazioni ed Economia

Questa sezione collega Entità e Risorse alle Meccaniche del gioco e alle Automazioni, sulla base di IME.

5.3.1 Input e Meccaniche di Controllo

Il Sistema di Controllo di WS si appoggia principalmente su tre Input con cui risultano controllabili tutte le Meccaniche e Stock Target:

- I tasti WASD per la direzionalità del giocatore. Presenta 2 – DOF perché legato a due assi dimensionali;

- Il tasto sinistro del mouse, associato all'uso dello Strumento equipaggiato e alla posizione del cursore sul piano dello schermo. Si è deciso di assegnare 2.5 - DOF, poiché il click del mouse pesa all'interno dell'input ma in maniera ridotta rispetto al posizionamento su schermo;

- I tasti numerici / rotellina del mouse per il cambio dello Strumento, entrambi corrispondenti a 1 – DOF.

Movimento

Il giocatore può muoversi in due direzioni sulla base dell'input direzionale. In particolare, i comandi WASD muovono il giocatore rispettivamente in alto, a sinistra, in basso e a destra. Quando incontra un animale o un qualunque ostacolo fisico viene bloccato. Questo pattern sia allinea con la maggior parte dei giochi basati su controllo diretto di un avatar.

Si è deciso di scartare un approccio “Punta e clicca” basato sul click del mouse e la relativa posizione del cursore, poiché sarebbe andato in conflitto con l’uso dello Strumento Equipaggiato.

Questa Meccanica agisce sullo Stock Target del Pastore, con una Granularità 1.

Cambio di Oggetto

Il giocatore può cambiare Strumento Equipaggiato premendo i tasti “1, 2, 3 e 4” sulla tastiera, oppure utilizzando la rotella del mouse. Anche in questo caso ha Granularità 1.

Bastone

Una volta equipaggiato, il Bastone viene attivato cliccando con il tasto sinistro su una Pecora. L’effetto è quello di “ordinare” alle pecore di ricominciare a muoversi in maniera ordinata verso i Pascoli e riprendere a brucare. Il Bastone influenza una singola istanza delle Pecore per volta, per cui ha Granularità 1.

Arma da Fuoco

Equipaggiato il Fucile, il tasto sinistro spara nella direzione del cursore. La direzione è quindi controllata dalla posizione relativa del mouse rispetto al Pastore. In questo caso lo Stock Target sono i Lupi, attraverso la Meccanica di Impatto.

Quando il giocatore spara, nella direzione indicata, viene creato un Proiettile che viaggia a velocità fissa fino a che non incontra un elemento dell’ambiente.

L’arma non può sparare per un tempo fisso di “Cooldown”, Attributo che ne regola la cadenza. Inoltre, la precisione dell’arma diminuisce e re-incrementa gradualmente fino a raggiungere di nuovo il massimo dopo una certa quantità di tempo dall’ultimo Proiettile sparato.

Impatto

Quando il giocatore spara, viene creato un Proiettile che si muove a velocità costante nella direzione definita. Esso:

- Si distrugge al contatto con qualsiasi Entità;
- Infligge danno a qualsiasi animale (Pecora o lupo) colpito.

5.3.2 Automazioni

Nel contesto di WS, sono rintracciabili due Automazioni rilevanti.

Automazione delle Pecore

Le pecore possono trovarsi in tre stati:

- Brucanti: si muovono verso il Pascolo più vicino tramite un algoritmo di pathfinding che evita Pozze e Dirupi. Una volta raggiunto il Pascolo, generano PP a intervalli regolari, dell'Ordine di decine / secondo come evidenziato precedentemente;

- Disperse: a intervalli parzialmente stocastici, una pecora può passare allo stato Disperso, muovendosi in maniera casuale. In questo stato non produce PP, non si dirige verso nessun Pascolo e rischia di venire eliminata per mezzo di Pozze e Dirupi;

- Fuga: con l'arrivo di un Lupo, le Pecore si muovono in direzione opposta ignorando il loro Stato precedente, diventando non responsive e vulnerabili a Pozze e Dirupi. Eliminare la minaccia ripristina lo Stato precedente.

In termini IME, queste automazioni agiscono sugli Attributi di Stato e posizione delle pecore, senza essere controllabili direttamente dagli Input.

Comportamento dei Lupi

I Lupi, una volta generati, si dirigono verso la pecora più vicina. La logica cerca di evitare che troppi lupi convergano sulla stessa unità (tramite il tracciamento delle Pecore già bersagliate). Questa scelta, dal punto di vista dell'estetica di gioco, fa percepire al giocatore i Lupi come "più intelligenti" e "pericolosi" rispetto alle Pecore.

5.3.3 Sistema Economico

Il Sistema Economico di WS connette **P**ecore, PP e Monete in una rete di trasformazioni e feedback**.**

Pecore → PP → Monete

In questa catena:

- Pecore Brucanti su Pascoli producono PP nel tempo;
- Alla fine dello Scenario, i PP vengono convertiti in Monete secondo una Meccanica di Conversione regolata dal Sistema di Arbitraggio;
- Le Monete diventano spendibili nel Negozio per ottenere nuove Pecore e Strumenti.

Monete → Pecore / Strumenti

Le Monete alimentano le Meccaniche economiche di acquisto:

- Monete → Pecore (incremento dello Stock del gregge);
- Monete → Strumenti (ampliamento del ventaglio di azioni disponibili).

Lupi → riduzione Pecore

I Lupi agiscono riducendo le Pecore a disposizione, introducendo quindi un elemento di smorzamento nei confronti dello Stock delle Pecore, con Ordine 1.

Si osservano quindi:

- Un feedback loop positivo (auto-rinforzante): più Pecore → più PP → più Monete → possibilità di comprare ulteriori Pecore e Strumenti;

- Un feedback negativo (stabilizzante / punitivo): più lupi o peggiore prestazione action → meno pecore → meno PP e Monete → minore capacità di ripresa.

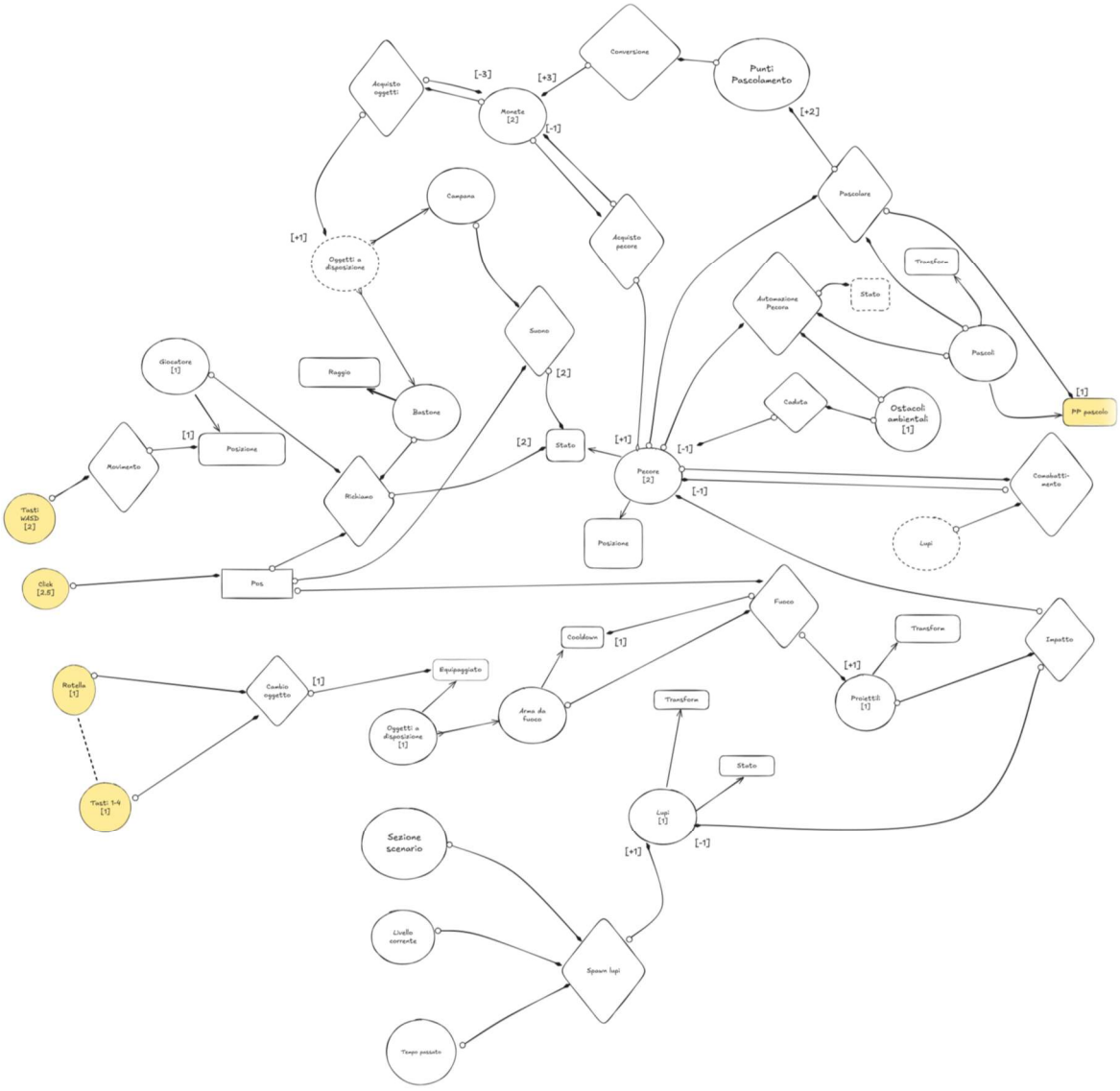


Figura 22: diagramma IME di WS.

6. Svolgimento dei test e calcolo delle metriche

Questa sezione descrive il processo di test empirici relativi al framework applicato a WS. L'obiettivo è quello di trarre conclusioni sulla base teorica definita nei capitoli precedenti mettendola in relazione con l'esperienza soggettiva dei giocatori.

6.1 Metodologia dei test

Per il test del prototipo e la raccolta dei dati è stata realizzata una demo composta da quattro livelli di gioco, seguiti da un breve questionario in-game. Le sessioni sono state eseguite in remoto, in forma completamente anonima, e i dati di gioco sono stati raccolti via codice tramite l'API messa a disposizione da Mixpanel.

Questo setup permette di:

- Acquisire i dati empirici di telemetria necessari al calcolo di APM, Complessità LZ e Metrica di Successo;

- Correlare tali misure con l'esperienza soggettiva dichiarata dal giocatore nel questionario finale.

Relativamente al framework, questa fase corrisponde ai punti f) e g) del workflow definito all'interno del capitolo 4.

6.1.1 Design della demo

All'interno del prototipo sono stati implementati quattro Scenari di difficoltà crescente, accompagnati da dialoghi di tutorial nelle fasi in cui vengono introdotte nuove Meccaniche o Strumenti.

All'avvio della demo, subito dopo il menu iniziale, è mostrata una schermata di istruzioni che introduce il test in modo neutrale:

- Scopo del test (raccolta di dati ai fini di ricerca);

- Durata conservativa prevista (circa 20 minuti), per ridurre il rischio di abbandono per motivi logistici;

- Presenza del questionario finale, la cui compilazione è necessaria perché la sessione venga considerata valida.

Per questa demo, si è deciso di regolare i parametri del Sistema di Arbitraggio in modo che il giocatore inizi con:

- 0 Monete;

- 0 Punti Pascolamento;

- 8 Pecore;

- Un Oggetto Disponibile: il Bastone.

Al termine di ogni Scenario, coerentemente con il meta – loop descritto nel capitolo precedente, il giocatore ottiene Monete in funzione dei Punti Pascolo accumulati nello Scenario, e può spenderle nel Negozio per acquistare Pecore e Strumenti.

Sequenza degli Scenari

Scenario 1

Il primo Scenario ha una funzione prevalentemente didattica, dato che presenta le regole di base: controllo del pastore, funzionamento dei Pascoli, comportamento di Pecore e Lupi.

Al termine di questo Scenario:

- Le Monete ottenute sono fisse a 250, indipendentemente dalla performance del giocatore;

- Il Negozio offre una sola opzione di acquisto oltre alle Pecore: il Fucile, il cui acquisto è obbligatorio;

Questa scelta fa parte dell'"arco di apprendimento", in modo che il giocatore arrivi a conoscere tutte le meccaniche di gioco indipendentemente dall'esperienza pregressa.

Scenari 2 e 3

I Livelli 2 e 3 hanno lo scopo di offrire una sfida bilanciata al giocatore. Al termine di essi il resto delle opzioni di acquisto è disponibile.

L'opzione di acquisto delle Pecore è sempre disponibile, e il numero di Monete ottenute è calcolato sulla base dei Punti Pascolo (come da design), ma bilanciato manualmente sulla base della difficoltà dello Scenario successivo.

Scenario 4

Il quarto Scenario è volutamente molto difficile. Un box di dialogo chiarisce che l'obiettivo non è necessariamente vincere ma "resistere il più possibile". In questo modo:

- Viene forzata la conclusione del test in un tempo ragionevole;

- Si consente al giocatore di esprimere al meglio le abilità acquisite, osservando il comportamento in condizioni di difficoltà maggiore.

Oggetti aggiuntivi

Oltre agli Oggetti descritti nel design di WS, si sono introdotti tre oggetti aggiuntivi all'interno del gioco tra le opzioni di acquisto:

- L'Oggetto "Tommy": un'Arma da Fuoco che rispetto al Fucile è più rapida e meno precisa;

- L’Oggetto “Pozione di Teletrasporto”: Oggetto che permette al giocatore di cliccare su schermo per muoversi molto velocemente;

- L’Oggetto “Campanella”: Strumento che istantaneamente riporta le Pecore Disperse nella condizione di Brucante.

La Campanella, in particolare, va a compiere la stessa funzione del Bastone ignorando però la necessità di muoversi e agendo sullo Stock delle Pecore con Granularità 2 (anziché 1) e costituendo potenzialmente un’opzione sbilanciata rispetto alle altre. Questo nell’ottica di comprendere meglio come le metriche si riflettono su questi aspetti.

Struttura del questionario

Il questionario in-game, somministrato al termine del quarto livello, è composto da quattro domande a risposta chiusa. Sebbene molto sintetico, permette di raccogliere le informazioni chiave sull’esperienza soggettiva del giocatore, in modo da poterle confrontare con i dati telemetrici.

Le domande incluse sono:

- Domanda 1 (presenza di problemi tecnici all’interno del gioco): per capire se si sono presentati bug o errori all’interno della sessione;

- Domanda 2 (gravità dei problemi tecnici): questa domanda assieme alla domanda 1 permette di capire quanto i problemi tecnici abbiano impattato sull’esperienza del giocatore. Una risposta positiva a entrambe le domande comprometterebbe l’integrità dei test;

- Domanda 3 (difficoltà percepita): questa domanda si ricollega all’instaurarsi dello stato di flow nel gioco e alla qualità dell’esperienza di gioco. Risposte come “Troppo Facile” o “Troppo Difficile” riassumerebbero un’esperienza di gioco noiosa o frustrante rispettivamente, e vengono interpretate come negative;

- Domanda 4 (possibilità di continuare): questa domanda chiede al giocatore se avrebbe continuato a giocare se ce ne fosse stata l'opportunità. Questa, assieme alla domanda precedente, aiuta a determinare in poche parole se la demo "è piaciuta". Risposte come "Troppo facile" / "Troppo difficile" alla domanda precedente combinata a "Sì" a questa domanda sarebbero indici di una forte robustezza del design di gioco, poiché nonostante lo sbilanciamento di difficoltà il giocatore nel complesso avrebbe continuato a giocare.

6.1.2 Definizione della Metrica di Successo

Per collegare il comportamento del giocatore alla probabilità di successo dell'obiettivo di gioco, è stata scelta una Metrica di Successo come definita nel Capitolo 4.

In questo caso si è scelto di implementare una forma di ETA, basata su due elementi principali:

- Il numero di Pecore Brucanti nello Scenario in un dato istante (e quindi produttive);

- La distanza residua in PP rispetto alla soglia necessaria per completare lo Scenario.

Concettualmente questo tipo di ETA permette di riassumere il "vantaggio" del giocatore rispetto all'obiettivo attuale, poiché:

- Considera il numero di Pecore, che, come visto dai loop economici del Capitolo precedente, impatta sul numero di PP e sul numero di Monete accumulate e si riflette direttamente negli Scenari successivi;

- Un minor numero di Pecore corrisponde a un tempo necessario per completare lo Scenario maggiore che, a sua volta, per come vengono generati i Lupi vuol dire che il giocatore dovrà sopravvivere per più tempo e affrontare più minacce.

Nel tracciamento telemetrico, il valore corrente di ETA viene registrato assieme ad ogni evento di gioco, così da poterlo correlare alle azioni specifiche compiute dal giocatore.

6.1.3 Dettagli implementativi del prototipo e tracciamento degli eventi

Il prototipo è stato realizzato con la versione 2023.3.50.f1 del motore di gioco Unity.

La raccolta dei dati telemetrici è stata implementata mediante uno script apposito (*TelemetryManager.cs*), che invia a Mixpanel gli eventi di interesse tramite chiamate alla *funzione Mixpanel.Track(value)* dell'API, aggregandoli in batch. Le azioni tracciate sono:

- Inizio Sessione (SESSION_START): tracciato all'inizializzazione dello script *TelemetryManager.cs* per tenere traccia dell'inizio effettivo della sessione;

- Transizione di Scena (SCENE_TRANSITION): tracciata quando una nuova scena del livello viene caricata. Permette di rintracciare a quale livello si trova il giocatore e quanti ne ha giocati;

- Movimento (Movement): tracciato quando il giocatore preme input di movimento in qualunque direzione (WASD);

- Stop Movimento (MovementStop): tracciato quando il giocatore termina un input di movimento;

- Selezione Bastone (StickSelect): tracciato quando il giocatore seleziona l'oggetto Bastone all'interno della sua Cintura;

- Selezione Fucile (RifleSelect); tracciato quando il giocatore seleziona l'oggetto Fucile all'interno della sua Cintura;

- Selezione Tommy (TommySelect): tracciato quando il giocatore seleziona l'oggetto Tommy all'interno della sua Cintura;

- Selezione Campana (BellSelect): tracciato quando il giocatore seleziona l'oggetto Campana all'interno della sua Cintura;

- Selezione Bastone (StickSelect): tracciato quando il giocatore seleziona l'oggetto Bastone;

- Fuoco Fucile (RifleFire): tracciato quando un proiettile viene sparato dal Fucile;

- Fuoco Tommy (TommyFire): tracciato quando un proiettile viene sparato dal Tommy;

- Lupo Colpito (WolfShot): tracciato quando un Lupo viene colpito da un proiettile. Usato per discriminare l'esito del tentativo del giocatore di colpire un Lupo;

- Pecora Colpita (SheepShot): tracciata quando una Pecore viene colpita da un proiettile del giocatore. Usato per tracciare quando il giocatore compie un errore colpendo una Pecora;

- Apertura Menu (MenuOpen): tracciato quando un menu viene aperto e il gioco viene messo in pausa, importante per scartare il tempo di pausa e ricreare l'effettiva sessione di gioco;

- Chiusura Menu (MenuClose): tracciato quando un menù viene messo in pausa;

- Risposta al Quiz (QuizAnswer): tracciata quando l'utente clicca su una delle risposte al questionario;

- Fine Sessione (SESSION_END): tracciata dopo la risposta finale al quiz, quando il giocatore viene portato al menu di ringraziamento;

Per ogni azione vengono inoltre registrate:

- L'ETA: la metrica successa precedentemente definita, in modo che sia possibile tracciarne i cambiamenti di rispetto alle azioni;

- Il timestamp: ottenuto tramite *Time.realtimeFromStartup*. Per sapere quando l'azione è avvenuta all'interno della sessione di gioco lato client.

6.2 Processing dei dati e calcolo delle metriche

6.2.1 Calcolo delle metriche di design

Le metriche di design sono state ottenute andando a implementare IME attraverso una struttura dati in python che ne riproduce la struttura composta da Entità, Link e Attributi (script *sheep_wolf_diagram.py*). Attraverso la DFS, gli script *mltt.py* e *feedback_loops.py* vanno a raccogliere i valori ottenuti nel file *design_metrics.json.*

6.2.2 Calcolo delle metriche di telemetria

Gli eventi raccolti tramite Mixpanel vengono elaborati attraverso una pipeline di pre-processing implementata in Python, in modo da produrre un dataset coerente con le definizioni delle metriche di telemetria (APM, complessità LZ e ETA) e che raccolga la catena delle azioni del giocatore.

Le fasi della pipeline sono le seguenti:

1. Filtraggio per data e rilevanza (*preprocess_telemetry_useful.py*): gli eventi precedenti all'inizio ufficiale dei test vengono scartati, assieme ai campi forniti come overhead da Mixpanel che non sono utili, mantenendo solo quelli tracciati dalla sessione di Unity;

2. Raggruppamento per utente (*preprocess_telemetry_useful_grouped.py*): Mixpanel genera ad ogni utente un ID anonimo, per cui è possibile raggruppare i dati per ID del giocatore, necessario poiché gli eventi di sessione contemporanei possono risultare rimescolati;

3. Ordinamento degli eventi (*preprocess_telemetry_useful_grouped_ordered.py*): all'interno di ciascuna sessione l'ordine degli eventi lato client non è garantito, per cui vanno riordinati per l'informazione di timestamp memorizzata all'interno dell'evento;

4. Filtraggio (*preprocess_telemetry_grouped_ordered_filtered_merged.py*): i dati delle sessioni incomplete non sono utili; pertanto, vengono scartati gli eventi generati dalle sessioni i cui eventi ordinati non iniziano con SESSION_START e SESSION_END. Al fine del calcolo delle metriche solamente l'ultima sessione portata a termine con successo viene considerata;

5. Merge (*preprocess_telemetry_grouped_ordered_filtered_merged.py*): alcune azioni (e.g. Movement / TommyFire) risultano numericamente sbilanciate in termini di quantità rispetto ad altre, ma non per effettiva scelta del giocatore. Movimenti multipli del giocatore in rapida successione o proiettili di mitragliatrice sparati a raffica fanno in realtà parte dello stesso "evento", per cui è stata introdotta una soglia di 0.2s tale per cui eventi in successione il cui Δt è inferiore a 0.2 vengono uniti in una singola azione, a cui viene aggiunto il campo "duration";

6. Pulizia (*preprocess_telemetry_grouped_ordered_filtered_merged_cleanded.py*): il timestamp registrato per ciascuna azione in gioco è in realtà distorto dal fatto che il giocatore può aver messo in pausa la sessione o dalla presenza dei box di dialogo. Pertanto, viene misurato il tempo passato tra ogni MenuOpen e MenuClose per traslare gli eventi successivi in maniera coerente;

La segmentazione del processo in script e file json intermedi ha permesso di comprendere e debuggare in maniera più fluida il pre-processing.

A partire dai dati pre-processati in questo modo, gli script *metrics_mltt.py* e *metrics_lempel_ziv* calcolano le metriche andando a creare un file distinto per ciascun utente nella cartella *metrics_per_player*, che raccolgono quindi i valori individuali.

Infine, lo script *metrics_summary.py* va a raccogliere i dati finali calcolando media e varianza tra tutti i dati e raggruppandoli per risposte ai questionari (in particolare sulla base della risposta alle domande 3 e 4 relativa al gradimento dell'esperienza), con l'obiettivo di individuare pattern ricorrenti e differenze tra i gruppi dei giocatori.

7. Analisi dei dati e conclusioni

7.1 Lettura delle sessioni

In questo capitolo vengono analizzati i datti raccolti tramite L'API di Mixpanel nel periodo 15-21 novembre 2025 e delle metriche IME.

Dal calcolo delle metriche di design risultano i seguenti valori di MLTT:

- Movimento del giocatore: 2;
- Uso del Bastone: 25;
- Uso della Campana: 5;
- Arma da Fuoco: 10;
- Cambio Oggetto: 1.

Mentre dall'analisi dei feedback loop di gioco:

- Sono stati rilevati 14 loop;
- Lo Stock delle Pecore risulta coinvolto in ognuno di essi, risultando lo Stock che compare più volte.

Per i dati di telemetria, il dataset complessivo supera i 27.000 eventi, che dopo le fasi di pre-processing descritte nel capitolo precedente corrispondono alle sessioni di 13 giocatori.

Sul totale dei partecipanti, 11 giocatori (84.6% del totale) hanno dichiarato che avrebbero continuato a giocare se fosse stato possibile, mentre 2 hanno espresso la preferenza opposta. Per comodità, ci si riferirà ai due gruppi rispettivamente con le sigle CN ("Continue") e WCN ("Won't Continue").

Nonostante il 54% dei partecipanti abbia segnalato la presenza di problemi tecnici minori, la maggioranza ha quindi valutato l'esperienza di gioco globalmente in positivo.

Un dato interessante, è che tutti i giocatori, indipendentemente dal gruppo di appartenenza, hanno giudicato la difficoltà come "Ottimale". Tali risposte sono coerenti con la media del livello massimo raggiunto, che è pari a 3.8 su 4, indicando che quasi tutti i partecipanti hanno raggiunto l'ultimo scenario senza tuttavia percepire l'esperienza come eccessivamente semplice.

Considerando l'intero campione, l'APM medio si attesta attorno a 105, con una deviazione standard di circa 20. In analisi sulla scena competitiva degli RTS, valori a 50 APM sono in genere associati a giocatori principianti, mentre i professionisti possono arrivare a sostenere 400 APM o più. Il valore osservato nelle raccolte dati suggerisce quindi per il prototipo un ritmo di interazione complessivamente sostenuto (circa 1.75 azioni al secondo) per un campione non professionistico, coerente con la natura action del gioco.

La complessità LZ normalizzata raggiunge un valore medio pari a 0.85. Tale valore suggerisce che:

- Le azioni presentano struttura riconoscibile, con pattern che si ripetono;

- Al tempo stesso, è presente una varietà sufficiente a evitare che il comportamento risulti puramente ripetitivo o triviale.

Questo comportamento è compatibile con la natura gestionale del gioco: il giocatore esegue spesso combinazioni ricorrenti (movimento + controllo del gregge + uso di armi), ma deve adattarle a situazioni nuove.

La lunghezza media delle frasi LZ, pari a circa 3.86 azioni, conferma la presenza di sequenze relativamente brevi ma variate. Questo quadro è coerente con il core loop descritto nel capitolo 5, in particolare con il micro – loop di gioco.

Allo stesso tempo, i dati raccolti mettono in evidenza differenza tra i gruppi CN e WCN. In particolare:

- APM
    - Gruppo CN: 108.5 APM medi;
    - Gruppo WCN: 89.8 APM medi;
- Complessità comportamentale (complessità LZ normalizzata)
    - CN: 0.75;
    - WCN: 0.71;

Dal punto di vista interpretativo, questo dato è coerente con l'idea che i giocatori che hanno percepito il prototipo in maniera positiva abbiano:

- Sfruttato in modo più intenso il ventaglio delle azioni offerte dal Sistema di Controllo;
- Mantenuto un livello di esplorazione e adattamento più alto rispetto ai giocatori WCN, pur all'interno delle stesse regole di Arbitraggio e degli stessi Scenari.

Sebbene le differenze in complessità LZ siano relativamente contenute, il gruppo CN mostra in media un ritmo di azione più elevato e una varietà comportamentale maggiore.

È importante sottolineare che il risultato non va letto come una relazione direttamente proporzionale tra LZC e indice di gradimento, ma piuttosto una correlazione specifica all'interno di WS.

Più interessante è invece l'evoluzione del valore di LZC all'interno della sessione di gioco. Andando ad analizzarne il valore per ciascuno Scenario si rileva infatti un pattern:

- Nel primo Scenario, il valore medio delle sessioni tende ad essere drasticamente più alto (intorno a 1.00 per il primo livello);

- Negli Scenari 2 e 3 la complessità si riduce sensibilmente;

- Nel livello 4, la complessità torna a crescere, ma mantenendosi a livelli intermedi;

È vero che la complessità calcolata a livello di singolo Scenario tende a essere più alta di quella globale (poiché il calcolo su intervalli più brevi tende a enfatizzare la novità per come è definita la LZC), ma l'andamento relativo resta significativo.

Combinando questi valori con l'andamento di APM risulta:

- Nei primi Scenari un numero di APM intermedio compatibili con i range visti precedentemente;

- Nello Scenario finale un loro aumento, fino a un valore medio di 175 APM.

Si possono trarre alcune considerazioni compatibili con la struttura del design di gioco:

- Nel primo Scenario il giocatore si trova in fase principalmente esplorativa, poiché deve apprendere le Meccaniche di base (e.g. controlli, comportamento delle Pecore, uso del Bastone) e senza alcun tipo di pressione temporale, per cui gli APM sono naturalmente più bassi e la complessità LZC è più alta;

- Negli Scenari 2 e 3 le meccaniche fondamentali sono apprese, e di conseguenza il comportamento risulta più stabile e consolidato, la LZC è più bassa e con numero di APM medio superiore;

- Nell'ultimo Scenario, l'aumento deliberato della difficoltà e della pressione temporale induce il giocatore a un nuovo picco di esplorazione e reattività, in virtù del fatto che i pattern finora appresi non sono più sufficienti a far fronte alla sfida. In questo Scenario gli APM medi risultano significativamente più alti.

Questo è compatibile con una complessità LZC solo leggermente più alta rispetto ai livelli 2 e 3, ma con azioni più veloci, compatibile con un'azione che sfrutta le abilità acquisite, ma sotto pressione intensa.

Passando all'analisi aggregata sui singoli Strumenti, i dati telemetrici confermano alcune ipotesi di design e ne mettono in discussione altre:

- Un numero frequente di utilizzo dell'oggetto "Bastone" è presente nelle finestre con ΔETA negativo. Essendo questa metrica calcolata direttamente sul numero di pecore che pascolano, questo risultato era effettivamente previsto;

- Nell'ultimo Scenario, si è osservato un aumento del numero di finestre temporali in cui compare con alta frequenza l'azione *bell_select.* Questo indica che nelle sessioni di gioco in cui è stato acquistato l'oggetto Campana, i giocatori sono riusciti a mantenere il controllo più a lungo rispetto agli altri. Questo risultato è particolarmente importante, perché la previsione sull'oggetto Campana basata sul valore MLTT è risultata fondata;

- 7 giocatori su 13 hanno usato l'oggetto Tommy. Le finestre con elevati numeri di utilizzi si associano quasi sempre a valori elevati di ΔETA, sia in positivo che negativo. Tuttavia, tale comportamento è imputabile principalmente al fatto che l'oggetto viene utilizzato ampiamente nell'ultimo livello, dove il ΔETA previsto porta a una progressiva riduzione del numero di pecore. Di conseguenza non è possibile trarre indicazioni robuste riguardo al suo impatto sul bilanciamento generale.

7.2 Osservazioni e conclusioni

La prima conclusione riguarda il contributo concettuale del lavoro svolto. La definizione proposta di gioco action-gestionale risulta soddisfatta: WS soddisfa i requisiti individuati dalle componenti i), e ii) e iii) del Capitolo 4, e può pertanto essere considerato una concretizzazione del framework teorico.

Inoltre, la modellazione attraverso il ciclo decisionale e IME (in particolare dei feedback loop) ha permesso di:

- Evidenziare i principali elementi di memoria del sistema (Pecore, Lupi, Oggetti Disponibili), che sono effettivamente risultati centrali nella dinamica di gioco;

- Supportare la definizione del core loop e di gioco, successivamente riscontrabili nei pattern osservati nelle sessioni di gioco tramite telemetria.

In questo senso, il framework non si limita a fornire una tassonomia astratta del genere action-gestionale, ma si dimostra effettivamente utilizzabile come strumento di design e analisi.

Sul piano dell'esperienza soggettiva, l'84.6% dei partecipanti (11 su 13) ha dichiarato che avrebbe continuato a giocare se possibile. Data l'analisi precedente, due considerazioni appaiono centrali:

- Il tasso di completamento, almeno nel contesto di una demo breve e guidata, non è un buon indice del gradimento complessivo: anche i giocatori del gruppo WCN hanno completato la totalità dei contenuti disponibili;

- Allo stesso modo, una valutazione soggettiva positiva della difficoltà non implica automaticamente un alto livello di coinvolgimento o di desiderio di proseguire l'esperienza.

Ne consegue che, in fase di test su prototipi brevi, fare affidamento esclusivo su indicatori quali "completamento" e "difficoltà percepita" rischia di essere fuorviante dal punto di vista della qualità dell'esperienza.

L'uso delle metriche di Telemetria ha invece permesso di trovare collegamenti diretti con l'andamento della sessione di gioco e i comportamenti del giocatore. In particolare, la combinazione di APM e della LZC sembra avere correlazione diretta con il grado di esperienza, novità e difficoltà all'interno del gioco. Dai dati risulta che queste due metriche siano direttamente collegate alla curva di difficoltà e apprendimento

determinate dalla demo e in fase di design, permettendo di creare una correlazione tra il loro andamento e le sezioni di gioco (in particolare al tutorial, al medio-gioco e al livello finale di difficoltà crescente). Seppur in maniera debole dal punto di vista statistico, questi due valori sembrano anche collegati all'indice di gradimento soggettivo del gioco.

Inoltre, l'analisi congiunta di frequenza delle azioni e finestre temporali del giocatore (e quindi indirettamente dell'andamento della metrica di successo del giocatore), ha permesso di individuare una criticità prevista in fase di design grazie al valore di MLTT: cioè l'uso della Campana come strategia vincente.

In sintesi, l'analisi dimostra che:

- Il framework e il workflow per i giochi action-gestionali sono applicabili in modo operativo, consentendo di guidare la progettazione di un prototipo coerente e ha portato alla realizzazione di un'esperienza di gioco nel complesso gradevole e bilanciata;

- Esistono segnali coerenti di relazione tra pattern d'azione, strategie di gioco, grado di coinvolgimento e le metriche definite, sebbene tali risultati debbano essere confermati su scala maggiore;

- L'integrazione della modellazione proposta, metriche di design e telemetria definite all'interno del lavoro è collegata in maniera profonda al comportamento dei giocatori.

7.3 Limiti e sviluppi futuri

I risultati ottenuti vanno letti alla luce di alcune limitazioni. La prima e più evidente è la dimensione del campione statistico preso in considerazione. Le sessioni hanno coinvolto 13 utenti, un numero poco indicativo per generalizzare i risultati. Una popolazione più ampia permetterebbe non solo di rafforzare le correlazioni osservate, ma anche di

dividere il campione secondo variabili più ampie (e.g. esperienza pregressa, preferenza di genere).

Un secondo limite riguarda la natura prototipale di WS e il numero ridotto di iterazioni design–testing condotte. L'applicazione del framework a un singolo titolo, pur utile in chiave esplorativa, non consente di valutare fino a che punto i risultati ottenuti siano rappresentativi su più iterazioni. Per la validazione del framework sarebbe richiesta una sperimentazione su un set più ampio di prototipi e più iterazioni, idealmente con variazioni controllate su complessità delle Meccaniche, struttura dei feedback loop, intensità dell'azione e parametri di gioco.

Nonostante ciò, i risultati raccolti suggeriscono una serie di possibilità di sviluppo rilevanti sia sul piano della ricerca sia su quello applicativo. In primo luogo, la correlazione osservata tra metriche di telemetria, metriche di design ed esperienza soggettiva costituisce un punto di partenza per l'integrazione di sistemi adattivi runtime. L'idea è quella di utilizzare valori come APM, complessità delle sequenze e indicatori di successo per calibrare automaticamente elementi di gioco quali densità degli ostacoli, aggressività dei nemici, frequenza degli eventi o disponibilità degli Oggetti. Un tale sistema costituirebbe una forma di difficoltà adattiva in grado di mantenere il giocatore entro una "zona ottimale" di sfida, sulla base del comportamento osservato piuttosto che su parametri statici.

Su questa base, l'addestramento di modelli predittivi basati su deep learning e addestrati su un campione statisticamente significativo di dati telemetrici, risulta un orizzonte particolarmente interessante di sviluppo per poter inferire in tempo reale parametri di gioco numerici (e.g. Monete raccolte, numero di PP generati, tasso di generazione dei Lupi) che vanno a modificare l'esperienza del giocatore e adattandola a ciascun individuo.

Bibliografia

[1] Hunicke, R., LeBlanc, M., & Zubek, R. (2004). MDA: A formal approach to game design and game research. In *Proceedings of the AAAI Workshop on Challenges in Game AI*.

[2] Fullerton, T. (2018). *Game design workshop: A playcentric approach to creating innovative games* (4th ed.). A K Peters/CRC Press.

[3] Nuccio, W. (2016). *La progettazione dei giochi da tavolo. Strumenti, tecniche e design pattern*. Ugo Mursia Editore.

[4] Romero, B. (2008). *The mechanic is the message*.

[5] Csíkszentmihályi, M. (1990). *Flow: The psychology of optimal experience*. Harper & Row.

[7] Crawford, C. (1984). *The art of computer game design*. McGraw-Hill/Osborne.

[19] Rollings, A., & Adams, E. (2003). *Andrew Rollings and Ernest Adams on game design*. New Riders.

[108] Yee, N. (2006). The demographics, motivations, and derived experiences of users of massively multi-user online graphical environments. *Presence: Teleoperators and Virtual Environments*

[112] Boyd, J. R. (2018). *A discourse on winning and losing* (G. T. Hammond, Ed.). Air University Press.

[113] Shewhart, W. A. (1939). *Statistical method from the viewpoint of quality control*. U.S. Department of Agriculture.

[116] Balconi, M. (Ed.). (2017). *New frontiers in neuromanagement*. LED Edizioni Universitarie.

[117] Livoti, V., Del Popolo Cristaldi, F., Contemori, G., Saccani, M. S., & Bonato, M. (2025). Web-based assessment of dual-task costs at different ages: An analysis across cognitive domains. *Frontiers in Psychology*

[118] Wickens, C. D. (2002). Multiple resources and performance prediction. *Theoretical Issues in Ergonomics Science*

[119] Tram, N. T. B. (2022). Simulation modeling – An effective method in doing research for business and management. *Ho Chi Minh City Open University Journal of Science – Economics and Business Administration*

Sitografia

[6] DBpedia. (n.d.). Action game. DBpedia. https://dbpedia.org/page/Action_game

[8] GWI. (n.d.). Most popular gaming genres. GWI Blog. https://www.gwi.com/blog/most-popular-gaming-genre

[9] Entertainment Software Association. (2023). Essential facts about the U.S. video game industry 2023. https://www.theesa.com/resources/essential-facts-about-the-us-video-game-industry/2023-2/

[10] Statista. (n.d.). Action games – App market outlook, worldwide. Statista. https://www.statista.com/outlook/amo/app/games/action-games/worldwide

[11] Statista. (2024). App market outlook – Action games, worldwide, 2024. Statista.

[17] Game Developer. (n.d.). Video: Sid Meier explores “interesting decisions” in gameplay. Game Developer. https://www.gamedeveloper.com/design/video-sid-meier-explores-interesting-decisions-in-gameplay

[20] SteamDB. (n.d.). Steam database. https://steamdb.info

[21] Sensor Tower. (2023). Q4 2023 recap: Top city building games in the US on unified platforms.

[24] PC Gamer. (n.d.). Humankind review. PC Gamer. https://www.pcgamer.com/humankind-reviedi

[26] Polygon. (n.d.). Civilization 7 developer responds to mixed reviews. Polygon. https://www.polygon.com/news/520793/civilization-7-firaxis-responds-mixed-reviews

[28] GameFAQs. (n.d.). Utopia (Intellivision) – FAQs and guides. GameFAQs. https://gamefaqs.gamespot.com/intellivision/916427-intellivision/faqs/80325

[29] Ars Technica. (2017). Build, gather, brawl, repeat: The history of real-time strategy games. Ars Technica. https://arstechnica.com/gaming/2017/09/build-gather-brawl-repeat-the-history-of-real-time-strategy-games/

[31] Ars Technica. (2017). Build, gather, brawl, repeat: The history of real-time strategy games. Ars Technica. https://arstechnica.com/gaming/2017/09/build-gather-brawl-repeat-the-history-of-real-time-strategy-games

[33] GameSpot. (2012). Game Masters: Peter Molyneux. GameSpot. http://gamespot.com/articles/game-masters-peter-molyneux/1100-6384787/

[37] Nintendo Life. (2020). Herzog Zwei is the next SEGA Ages classic to come to Switch. Nintendo Life. https://www.nintendolife.com/news/2020/06/herzog_zwei_is_the_next_sega_ages_classic_to_come_to_switch

[39] The Digital Antiquarian. (2018). Controlling the spice, part 3: Westwood's Dune. https://www.filfre.net/2018/12/controlling-the-spice-part-3-westwoods-dune/

[41] Blizzard Entertainment. (n.d.). Warcraft: Orcs & Humans. Battle.net. https://us.shop.battle.net/it-it/product/warcraft-orcs-and-humans

[42] Korea JoongAng Daily. (2010). Gamers judge StarCraft II against classic that spawned an industry. https://koreajoongangdaily.joins.com/2010/07/29/industry/Gamers-judge-StarCraft-II-against-classic-that-spawned-an-industry/2923913.html

[43] MMOS.com. (n.d.). The first MOBA: Aeon of Strife. https://mmos.com/editorials/the-first-moba-aeon-of-strife

[48] Massive Entertainment. (2000). Ground Control anthology: Field commander's handbook. https://shared.akamai.steamstatic.com/store_item_assets/steam/apps/254820/manuals/Ground_Control_Anthology_Manual.pdf

[50] ModDB. (n.d.). SAGE – Strategy Action Game Engine. ModDB. https://www.moddb.com/engines/sage-strategy-action-game-engine

[51] Ars Technica. (2020). War stories: How Homeworld brought the third dimension to real-time strategy. Ars Technica. https://www.arstechnica.com/gaming/2020/04/war-stories-how-homeworld-brought-the-third-dimension-to-real-time-strategy/

[53] GameSpot. (2002). Command & Conquer: Generals – Preview. GameSpot. https://www.gamespot.com/articles/command-and-conquer-generals-preview/1100-2854173/

[54] GameSpot. (2005). Relic champions Company of Heroes. GameSpot. https://www.gamespot.com/articles/relic-champions-company-of-heroes/1100-6122929/

[56] Game Developer. (2011). Postmortem: Defense of the Ancients. Game Developer. https://www.gamedeveloper.com/design/postmortem-i-defense-of-the-ancients-i

[61] Ingram, M. (2007, May 28). Desktop Tower Defense totally rulez. https://mathewingram.com/work/2007/05/28/desktop-tower-defense-totally-rulez/

[63] Pocket Gamer. (2010). Plants vs Zombies becomes fastest-selling iPhone game to date. Pocket Gamer. https://www.pocketgamer.com/news/plants-vs-zombies-becomes-fastest-selling-iphone-game-to-date

[65] bdnews24.com. (n.d.). Articolo su Kingdom Rush. https://bdnews24.com/stripe/pw0wz3i78g

[67] Engadget. (2011). Dungeon Defenders sales hit 250K, majority on PC. Engadget. https://www.engadget.com/2011-11-02-dungeon-defenders-sales-hit-250k-majority-on-pc.html

[68] Engadget. (2011). Dungeon Defenders picks up gold from 600K sales. Engadget. https://www.engadget.com/2011-12-22-dungeon-defenders-picks-up-gold-from-600k-sales.html

[70] Shacknews. (2010). StarCraft 2 sells 1 million. Shacknews. https://www.shacknews.com/article/64982/starcraft-2-sells-1-million

[71] Wired. (2018). The fall and rise of real-time strategy games. Wired. https://www.wired.com/story/fall-and-rise-real-time-strategy-games/

[74] Polygon. (2016). League of Legends has 100 million monthly players. Polygon. https://www.polygon.com/2016/9/13/12865314/monthly-lol-players-2016-active-worldwide

[76] The Verge. (2017). Dota 2's The International 2017 prize pool explained. The Verge. https://www.theverge.com/2017/8/16/16155504/dota2-the-international-valve-2017-esports-economics

[77] GosuGamers. (2017). Throw-back TI: A glance at The Internationals from 2011 to 2016. GosuGamers. https://www.gosugamers.net/dota2/features/44878-throw-back-ti-a-glance-at-the-internationals-from-2011-to-2016

[78] Konvoy. (2018). Esports prize pools – 155.9M 2018. Medium. https://medium.com/konvoy/esports-prize-pools-155-9m-2018-ce1ff44bd41

[80] Time. (2016). Minecraft is now the best-selling game of all time. Time. https://time.com/4354135/minecraft-bestelling/

[81] The Guardian. (2014). Minecraft: The unstoppable rise of the game that has conquered YouTube. The Guardian. https://www.theguardian.com/technology/2014/nov/24/minecraft-youtube-videos-mojang-views

[83] Facepunch Studios. (2018). Rust – Leaving Early Access. Rust Blog. https://rust.facepunch.com/news/leaving-early-access

[84] Polygon. (2014). Steam Early Access is a risky bet for players and developers. Polygon. https://www.polygon.com/2014/11/14/7221557/Steam-early-access-bad-bet

[86] Vice. (2018). The creator of PUBG on where battle royale started and where it's going. Vice. https://www.vice.com/en/article/the-creator-of-pubg-on-where-battle-royale-started-and-where-its-going/

[87] TechPowerUp. (2018). PUBG has sold 50 million copies, 400 million players total. TechPowerUp. https://www.techpowerup.com/245313/pubg-has-sold-50-million-copies-400-million-players-total

[89] Wired. (2019). Fortnite World Cup viewership. Wired. https://www.wired.com/story/fortnite-world-cup-viewership

[90] Game Developer. (2020). PUBG has sold over 70 million copies worldwide. Game Developer. https://www.gamedeveloper.com/business/-i-pubg-i-has-sold-over-70-million-copies-worldwide/

[91] Business of Apps. (n.d.). Fortnite statistics. Business of Apps. https://www.businessofapps.com/data/fortnite-statistics/

[92] Nevion. (2020). Riot Games' LoL finals use JPEG XS. Nevion. https://nevion.com/news/press-releases/riot-games-lol-finals-jpeg-xs/

[93] Time. (2020). Esports and the coronavirus. Time. https://time.com/5812633/esports-coronavirus/

[94] Esports.gg. (2025). LoL Worlds 2025 will have a $5 million prize pool. Esports.gg. https://esports.gg/news/league-of-legends/lol-worlds-2025-will-have-a-5-million-prize-pool-more-than-double-last-years-amount-according-to-chris-greeley

[97] GameSensor. (n.d.). Total War: Warhammer – Steam performance. GameSensor. https://gamesensor.info/news/total_war_warhammer_

[100] The Verge. (2021). Valheim has sold five million copies. The Verge. https://www.theverge.com/2021/3/3/22311338/valheim-five-million-copies-sold-valve-steam-early-access-success/

[105] Nielsen, J. (1994/2010). 10 usability heuristics for user interface design. Nielsen Norman Group. https://www.nngroup.com/articles/ten-usability-heuristics/

[106] PC Gamer. (2021). Imperator: Rome's 2.0 overhaul makes big changes to the UI and warfare. PC Gamer. https://www.pcgamer.com/imperator-romes-20-overhaul-makes-big-changes-to-the-ui-and-warfare/

[107] Bartle, R. (n.d.). Richard Bartle's MUD and player types resources. https://mud.co.uk/richard/

[111] Machinations. (n.d.). Machinations – Game economy and system design platform. https://machinations.io/

[114] Valve Corporation. (n.d.). Overcooked on Steam. Steam Store. https://store.steampowered.com/app/448510/Overcooked/

[115] Game Developer. (2019). How moving from 2D to 3D shaped the design of Risk of Rain 2. Game Developer. https://www.gamedeveloper.com/design/how-moving-from-2d-to-3d-shaped-the-design-of-i-risk-of-rain-2-i-?

[120] Video Game Tourism. (2015). Demystifying MOBAs – A history of speed. Video Game Tourism. https://videogametourism.at/content/demystifying-mobas-history-speed

Ludografia

[12] *Super Mario Bros* [Videogioco]. (1985).

[13] *Super Contra* [Videogioco]. (1987).

[14] *Call of Duty* [Serie di videogiochi]. (2003–presente).

[15] *Guitar Hero* [Serie di videogiochi]. (2005–2015).

[16] *Armored Core VI: Fires of Rubicon* [Videogioco]. (2023).

[18] *Sid Meier's Civilization* [Serie di videogiochi]. (1991–presente).

[22] *Sid Meier's Civilization VI* [Videogioco]. (2016).

[23] *Humankind* [Videogioco]. (2021).

[25] *Sid Meier's Civilization VII* [Videogioco]. (annunciato 2024).

[27] *Utopia* [Videogioco]. (1981).

[30] *The Ancient Art of War* [Videogioco]. (1984).

[32] *Populous* [Videogioco]. (1989).

[34] *Herzog Zwei* [Videogioco]. (1989).

[35] *Warcraft: Orcs & Humans* [Videogioco]. (1994).

[36] *StarCraft* [Videogioco]. (1998).

[38] *Dune II: The Building of a Dynasty* [Videogioco]. (1992).

[40] *Command & Conquer* [Videogioco]. (1995).

[44] *Warcraft III: Reign of Chaos* [Videogioco]. (2002).

[45] *Dungeon Keeper* [Videogioco]. (1997).

[46] *Ground Control* [Videogioco]. (2000).

[47] *Sudden Strike* [Videogioco]. (2000).

[49] *Homeworld* [Videogioco]. (1999).

[52] *Command & Conquer: Generals* [Videogioco]. (2003).

[55] *Warhammer 40,000: Dawn of War* [Videogioco]. (2004).

[57] *Kessen* [Videogioco]. (2000).

[58] *Pikmin* [Videogioco]. (2001).

[59] *Odama* [Videogioco]. (2006).

[60] *Desktop Tower Defense* [Videogioco]. (2007).

[62] *Plants vs. Zombies* [Videogioco]. (2009).

[64] *Kingdom Rush* [Videogioco]. (2011).

[66] *Dungeon Defenders* [Videogioco]. (2011).

[69] *StarCraft II: Wings of Liberty* [Videogioco]. (2010).

[72] *League of Legends* [Videogioco]. (2009).

[73] *Defense of the Ancients (DotA)* [Mod videoludica]. (ca. 2003).

[75] *Dota 2* [Videogioco]. (2013).

[79] *Minecraft* [Videogioco]. (2011).

[82] *Rust* [Videogioco]. (2018).

[85] *PlayerUnknown's Battlegrounds (PUBG)* [Videogioco]. (2017).

[88] *Fortnite Battle Royale* [Videogioco]. (2017).

[95] *Total War: Warhammer* [Videogioco]. (2016).

[96] *Total War: Warhammer III* [Videogioco]. (2022).

[98] *Mount & Blade II: Bannerlord* [Videogioco]. (2022).

[99] *Valheim* [Videogioco]. (2021).

[101] *Overcooked* [Videogioco]. (2016).

[102] *Risk of Rain 2* [Videogioco]. (2020).

[104] *They Are Billions* [Videogioco]. (2019).

[109] *FTL: Faster Than Light* [Videogioco]. (2012).

[110] *Company of Heroes* [Videogioco]. (2006).

Note e link

Link alla demo di WS: https://dramaticteapot.itch.io/wolvesandsheeps

Note sull'uso di IA generativa: i codici degli script python contenuti nell'analisi di dati di telemetria ottenuti da Mixpanel, e i codici degli script *ShopUIManager.cs*, *ShopManager.cs*, *InventoryManager.cs* e *InventoryUIManager.cs,* sono stati scritti con l'assistenza di Cursor AI.

Note sulla licenza dei materiali: tutti contenuti inclusi nel presente elaborato sono utilizzati nel rispetto delle relative licenze d'uso e dei diritti dei rispettivi proprietari.

Si ringraziano gli sviluppatori The Crazy Hyper-Dungeon Chronicles e Atomicrops per aver contribuito alla realizzazione del lavoro attraverso il confronto diretto e il loro materiale di analisi.