

Indice

1	Introduzione	5
1.1	Piattaforme	6
1.2	Processo di sviluppo	7
1.3	Multiplayer	8
2	Design	15
3	Direzione artistica	16
4	Sviluppo	17

Elenco delle figure

1	Overcooked - Party game a visuale condivisa	9
2	Heroes of Might & Magic - Multiplayer locale a turni	10
3	Mario Karts - Multiplayer in split-screen	10
4	Battlerite - Multiplayer online battle arena.	11
5	EVE Online - Massive multiplayer online game	12
6	PlayerUnknown's Battleground - Battle royale	12
7	Battlefield 4 - Multiplayer con interoperabilità pc\mobile	14

Ringraziamenti

Sommario

In questo elaborato di tesi si descrive il processo di sviluppo di Orbtail, un gioco multiplayer multiplatforma che coniuga elementi *racing* ed *arena* e diverse modalità di gioco in un gameplay rapido e competitivo. Il prodotto, nato nel 2013 come progetto per il corso di Videogame Design and Programming, é stato rielaborato ed esteso, andando ad aggiornare l'intero comparto tecnico, il design originale e la direzione artistica. Il pregio principale di Orbtail risiede nella sua capacità di poter essere giocato da piú giocatori contemporaneamente, online o offline, in una qualsiasi combinazione di piattaforme e/o numero di giocatori locali. Questo documento ne descrive il processo produttivo, a partire dal *concept* fino alle varie scelte tecniche, facendo numerosi paralleli con la versione originale al fine di individuare carenze e spunti per migliorie.

Questa tesi é strutturata in capitoli.

Nel primo capitolo viene dato un contesto al progetto, fornendo una panoramica generale ed andando ad analizzare i requisiti dello sviluppo multiplatforma e di varie modalità multigiocatore.

Nel secondo capitolo vengono delineate le meccaniche del gioco e definiti i requisiti di *level design*. Vengono inoltre analizzati alcuni titoli concorrenti al fine di individuare meccaniche simili ed analizzarne i vari aspetti e criticità.

Nel capitolo successivo verrà inquadrata la direzione artistica del progetto usando diversi titoli esistenti come riferimento. Viene inoltre analizzata la produzione dei vari *asset* utilizzati all'interno del gioco considerando i requisiti tecnici e di design.

Il quarto capitolo verte sulle scelte tecnologiche, a partire dalla scelta dell'engine, delle piattaforme e dell'architettura del gioco. Ogni scelta verrà contestualizzata rispetto a varie alternative e opportunamente giustificata, considerando necessariamente i requisiti di design.

Il capitolo finale é dedicato alle conclusioni nonché ad eventuali sviluppi futuri.

1 Introduzione

I videogame rappresentano una delle forme di intrattenimento moderne più affermate e diffuse. A differenza di quanto avviene con i mezzi classici quali *film* e *libri*, l'utente ha pieno controllo dell'esperienza: alcuni giochi richiedono concentrazione, altri una buona dose di coordinazione e reattività, altri ancora pazienza e pianificazione. Sebbene vi siano videogiochi che si concentrano principalmente sulla componente *single-player*, i giochi, intesi anche nel senso classico del termine, nascono come mezzo di *condivisione* dell'esperienza con altri partecipanti.

In questa tesi si descrive il processo di sviluppo di *Orbtail*, un videogioco *multiplayer* con elementi *racing* disponibile per dispositivi mobile e desktop quali Android, iOS, Windows e OSX. Il gioco prevede tre modalità di gioco differenti e consente fino a quattro giocatori di sfidarsi online in modalità *cross-platform play*¹ su una delle tre arene disponibili. Vi è inoltre la possibilità di condividere uno stesso dispositivo desktop tra due o più giocatori locali mediante un'apposita modalità *splitscreen*².

La necessità di supportare più piattaforme, impedendo che ciascuna di esse potesse risultare avvantaggiata rispetto alle altre durante le sessioni *cross-platform play*, ha portato allo sviluppo di un *gameplay*³ essenziale ed immediato. La durata limitata delle sessioni, unita ad una curva d'apprendimento semplice, rende il prodotto particolarmente adatto ad un pubblico di *casual gamer*⁴.

¹Il termine *cross-platform play* identifica una modalità di gioco online in cui giocatori possono giocare tra loro indipendentemente dalla piattaforma utilizzata.

²Il termine *splitscreen* identifica una modalità di gioco in cui lo schermo viene suddiviso in più quadranti in modo da consentire a più giocatori di usare uno stesso dispositivo contemporaneamente.

³Il *gameplay* (in italiano traducibile come “*esperienza utente*”) comprende gli elementi di gioco quali storia, regole, obiettivi, progressioni, interazione utente, ecc.

⁴Si definisce *casual gamer* un giocatore saltuario, non particolarmente interessato alla cultura dei videogiochi e, solitamente, alla tecnologia in generale.

1.1 Piattaforme

L'esperienza videoludica ha il grande vantaggio di poter essere studiata per soddisfare utenze dai gusti profondamente diversi su un vasto numero di piattaforme. Ad oggi esistono decine di tipologie di dispositivi di intrattenimento differenziati per costo, performance e modalità d'interazione.

Le piattaforme *mobile*, pur essendo nate solo di recente, si sono subito affermate come una delle principali piattaforme da gaming. Il successo senza precedenti di *smartphone* e *tablet* ha consentito all'industria mobile di generare un volume d'affari pari a quello di *pc* e *console* combinati [1].

Pur non godendo delle stesse performance e manifestando un ciclo di vita più breve rispetto alle controparti classiche, i dispositivi mobile godono di una più vasta diffusione e si rivolgono ad un pubblico decisamente più ampio.

L'interazione primaria affidata all'*input touch* è affiancata da un grande numero di sensori accessori quali *GPS*, *accelerometri*, *giroscopi*, *bussole* e *videocamere*. Recenti sviluppi tecnologici quali *augmented reality*⁵ e *virtual reality*⁶, unito all'uso sapiente di queste nuove modalità d'interazione, ha permesso la nascita di nuovi paradigmi di *gameplay* rimasti finora inesplorati.

I *pc* e le *console* sono le piattaforme storiche su cui sono nati e diffusi i videogame. Accomunate da paradigmi d'interazione e performance paragonabili, queste due famiglie di dispositivi sono nate per scopi diversi. Laddove le *console* sono pensate per essere un dispositivo d'intrattenimento *dedicato*, che fa della *facilità d'uso* il suo cavallo di battaglia, i *pc* hanno una natura più generica e garantiscono performance e flessibilità maggiori al costo di una richiesta più elevata di competenza da parte dell'utenza.

Il ciclo di vita e il costo delle due piattaforme è inoltre molto diverso. Le *console* sono dei *sistemi embedded* precostruiti e limitatamente aggiornabili; il loro ciclo di vita è generalmente superiore al quinquennio e sono caratterizzati da un costo relativamente basso. I *pc*, d'altro canto, possono essere assemblati scegliendo i componenti che meglio si adattano alle esigenze dell'utente ed aggiornati quando ritenuto necessario. Il costo di un *pc* è generalmente molto più alto rispetto a quello di una console, anche a parità di specifiche tecniche.

⁵La *augmented reality* (in italiano «realtà aumentata»), consiste nell'arricchire il mondo circostante attraverso contenuti di tipo visivo, aptico o uditivo generati da un elaboratore.

⁶La *virtual reality* (in italiano «realtà virtuale») consiste nel simulare un ambiente tridimensionale tramite un elaboratore e lasciare che l'utente vi interagisca attraverso periferiche di input specializzate quali visori, cuffie e controller.

L'ampio parco di periferiche di input supportate, dalle piú comuni quali tastiere, mouse e gamepad a quelle piú specializzate quali volanti, pedaliera e *HOTAS* ⁷, rendono queste piattaforme adatte a qualsiasi tipologia di gioco ed interazione utente.

1.2 Processo di sviluppo

Il processo di sviluppo di un videogioco cambia radicalmente in funzione della piattaforma e dell'utenza a cui viene destinato il prodotto. Un design di successo deve essere in grado di sfruttare le peculiarit  di ciascun dispositivo e considerarne le limitazioni tecniche, senza pregiudicare l'esperienza utente.

Lo sviluppo risulta solitamente tanto pi  avvantaggiato quanto minori sono le limitazioni delle piattaforme o variet  di specifiche tra dispositivi. La presenza di dispositivi *identici* tra *console* appartenenti ad uno stesso ecosistema garantisce un'elevata consistenza dell'esperienza utente e consente interventi di ottimizzazione ad una granularit  molto fine. Laddove questo processo   solitamente molto *efficiente* su console, lo stesso non pu  essere detto per piattaforme *mobile* e *desktop* per via dell'elevata variet  di specifiche tecniche o combinazioni di componenti. I *pc* risultano avvantaggiati per via delle elevate performance e assenza di grosse limitazioni e ci  consente loro di mantenere un'elevata fedelt  e fruibilit  del contenuto. Lo sviluppo su dispositivi quali *smartphone* e *tablet*, d'altro canto, deve scontrarsi con la presenza di dispositivi con capacit  profondamente diverse e comparti tecnici non sempre in equilibrio tra loro (non   raro assistere a dispositivi che associano elevate risoluzioni a performance mediocri). In questo caso   richiesto uno sforzo maggiore affin  l'applicazione scali in funzione del dispositivo per garantire una buona esperienza utente.

Sebbene esistono titoli sviluppati in *esclusiva* per alcune piattaforme, la necessit  di aumentare il bacino d'utenza, e di conseguenza i ricavi, solitamente richiede che un prodotto venga distribuito su pi  ecosistemi. Un tempo processo lungo ed oneroso per via delle marcate differenze tra le architetture e carenza di *tool*, ad oggi lo sviluppo *cross-platform* risulta molto avvantaggiato. Il rilascio di *engine third-party*, unito alla convergenza delle piattaforme verso architetture simili, permette agli sviluppatori di lavorare

⁷L'*HOTAS* (acronimo di «*hands on throttle and stick*»), usato solitamente per giocare a simulatori di volo, consiste di un joystick a 4 o pi  assi e una leva d'accelerazione.

ad un livello di astrazione piú elevato evitando di creare supporti di basso livello specifici per dispositivo.

Laddove sviluppare un videogioco per piattaforme simili quali potrebbero essere *pc* e *console* oppure *smartphone* e *tablet*, costituisce piú un problema implementativo che non a livello di *gameplay*, sviluppare per piattaforme profondamente diverse richiede importanti considerazioni a livello di *design* e produzione *asset*. In primo luogo il differente grado di apprezzamento dell'utenza delle varie piattaforme può precludere il successo a certe tipologie di gioco di nicchia (quali potrebbero essere simulatori o strategici) e favorire i design che si rivolgono un pubblico piú *mainstream*. Le differenti modalità d'interazione potrebbero inoltre richiedere la rivisitazione dell'interfaccia grafica e l'eventuale eliminazione degli input disponibili solo su certe tipologie di dispositivi. Il comparto tecnico deve inoltre consentire al prodotto di poter scalare in funzione delle performance a disposizione. Possibili interventi consistono nella riduzione del dettaglio delle *texture* o della complessità poligonale, riduzione o rimozione dell'effettistica, limitazione degli oggetti a schermo. Ove ciò non fosse possibile o insufficiente potrebbe essere necessario rivisitare il *gameplay* attraverso la riduzione di giocatori o avversari, semplificazione della *IA* o rimozione di feature particolarmente onerose.

1.3 Multiplayer

L'interazione sociale come mezzo per aumentare il coinvolgimento videoludico é il motivo principale per cui i videogiochi multiplayer hanno da sempre riscosso un grande successo. Sebbene esistono infinite variazioni sul tema, le modalità di gioco multiplayer possono essere classificate in due famiglie: *cooperative* e *competitive*. Alla prima categoria appartengono quelle modalità in cui due o piú giocatori *collaborano* tra loro al fine di raggiungere un obiettivo comune. In queste modalità l'elemento di sfida é rappresentato dal gioco stesso e governato da *intelligenze artificiali* piú o meno sofisticate. Nelle modalità *competitive* l'elemento di sfida é invece rappresentato dai giocatori stessi: gli utenti sono portati a confrontarsi gli uni con gli altri al fine di raggiungere un obiettivo impedendo al contempo che gli altri giocatori possano fare altrettanto. Il livello di sfida offerto da alcuni titoli é tale per cui, pur di eccellere, alcuni giocatori hanno deciso di farne una carriera, sottoponendosi costantemente a veri e propri allenamenti. Questo fenomeno ha di recente portato alla nascita di competizioni a livello agonistico, organizzate e regolate da entità terze, in cui i partecipanti non sono semplici giocatori ma



Figura 1: Overcooked - Party game a visuale condivisa

veri e propri atleti. Queste competizioni prendono il nome di *e-sports* (sport elettronici).

Le modalità multiplayer hanno accompagnato lo sviluppo dei videogiochi fin dagli albori, in un'epoca in cui la diffusione di Internet era molto limitata. Le realizzazioni più semplici consistono nello sfruttare uno stesso dispositivo col quale tutti i giocatori possono interagirvi. La presenza fisica di tutti i partecipanti coinvolti favorisce un'interazione sociale più immediata e rende queste modalità di gioco particolarmente adatte ai *party-game*⁸. Le implementazioni più semplici consistono nell'utilizzare una visuale condivisa tra tutti i partecipanti (Fig.1) oppure, laddove l'interazione in contemporanea non fosse necessaria, sfruttare la meccanica dei *turni* affinché ciascun giocatore goda di un punto di vista unico sul mondo di gioco (Fig.2). Un'implementazione più sofisticata, nota col termine *split-screen*, consiste nel suddividere lo schermo in quadranti, solitamente da due a quattro, e mostrare in ciascuno di essi il punto di vista di uno dei giocatori in maniera indipendente (Fig.3). L'elevato impatto sulle performance, tale da rendere necessarie ottimizzazioni particolarmente aggressive, unita al ridotto bacino d'utenza cui queste modalità si rivolgono, ha di recente portato ad un calo di titoli che offrono questo tipo di esperienza.

⁸I *party-game* sono l'equivalente videoludico dei giochi di società.



Figura 2: Heroes of Might & Magic - Multiplayer locale a turni



Figura 3: Mario Karts - Multiplayer in split-screen



Figura 4: Battlerite - Multiplayer online battle arena.

La rapida diffusione di Internet, unita ai suoi sviluppi degli ultimi decenni, ha consentito la nascita di nuove modalità multigiocatore *online*. Rispetto all'approccio classico, un multiplayer online permette agli utenti di usare il proprio dispositivo per accedere a sessioni di gioco con un numero di partecipanti che varia tra la decina e il migliaio, indipendentemente dalla distanza fisica che li separa. Il rinnovato successo dei giochi multiplayer ha consentito la nascita di nuovi generi, dai *MOBA* in cui due team dal numero ristretto di giocatori si sfidano all'interno di un'arena (Fig.4) , agli *MMO* i cui mondi persistenti vantano migliaia di giocatori attivi contemporaneamente (Fig.5) , passando per i più recenti *Battle Royale* caratterizzati da centinaia di giocatori che lottano per la sopravvivenza (Fig.6).

Le implementazioni più comuni sono riducibili a due diverse *architetture*, in funzione del ruolo che i diversi dispositivi hanno all'interno della sessione di gioco. Nell'architettura *client-server* i dispositivi dei giocatori, detti *client*, sono connessi ad un dispositivo centrale, detto *server*, il cui ruolo consiste nel coordinare tutti i partecipanti, gestire l'evoluzione della partita, propagare lo stato condiviso di gioco ed eventualmente validare le azioni dei singoli giocatori per evitare *cheat* ⁹. I *client* comunicano esclusi-

⁹Il *cheat*, in italiano «imbrogliare», rappresenta una qualsiasi tecnica atta a sovvertire le regole di gioco affinché un giocatore ne ottenga un beneficio immeritato.



Figura 5: EVE Online - Massive multiplayer online game



Figura 6: PlayerUnknown's Battleground - Battle royale

vamente col *server* limitandosi a dichiarare le azioni effettuate dall'utente. Il ruolo di *server* può essere ricoperto sia da una macchina esterna che non prende parte alla partita detta *server dedicato*, oppure direttamente da uno dei dispositivi dei giocatori, detto *host*. L'approccio con *server dedicato* permette di sfruttare macchine remote caratterizzate dalla elevate performance al fine di gestire efficientemente un numero anche elevato di giocatori. Il secondo di questi, invece, consente di evitare i costi associati alla gestione o noleggio del server al costo di una minore scalabilità. L'esperienza multigiocatore con questo tipo di approccio è inoltre soggetta a fattori difficilmente prevedibili quali prestazioni del dispositivo che fa da *host*, qualità della connessione verso i *client* e la possibilità di disconnessione del *server* durante una partita che potrebbe richiedere il *trasferimento* del ruolo di server ad uno degli altri dispositivi (processo noto col termine *host recovery*).

Nelle architetture *peer-to-peer*, a differenza di quanto avviene con l'approccio *client-server*, lo stato di gioco è condiviso su tutti i dispositivi e gestito in maniera distribuita. Sebbene questa architettura impedisce che il carico computazionale si concentri su un unico dispositivo, evitando inoltre i costi associati a macchine server esterne, la condivisione dell'*autorità* sullo stato di gioco favorisce il proliferare del fenomeno del *cheating*, oltre che a complicare la sincronizzazione tra i vari dispositivi e la risoluzione di eventuali discrepanze nello stato di gioco.

Per far fronte al numero crescente di giocatori all'interno di ciascuna sessione di gioco, le implementazioni più moderne si basano su un approccio *misto client-server* in cui parte della computazione viene delegata ai vari *client* che, pertanto, possiedono un certo grado di autorità sulla partita.

Sebbene esistano numerosi paradigmi consolidati per la gestione dei giochi multiplayer, la quasi totalità delle implementazioni *segmentano* il bacino d'utenza in funzione della piattaforma d'appartenenza. Questo approccio semplifica enormemente la gestione dei servizi online accessori, specifici per piattaforma e molto diversi tra loro, evitando i costi associati alla gestione dell'*interoperabilità* tra di essi, tuttavia impedisce che giocatori su piattaforme diverse possano giocare gli uni contro gli altri. Esistono alcuni esempi di giochi multipiattaforma che consentono a giocatori su *smartphone* e *tablet* di interagire ed influenzare partite in corso su altre piattaforme (quali *pc* e *console*), tuttavia essi rappresentano una nicchia ristretta che non ha mai riscosso un vero e proprio successo per via dei ruoli particolarmente asimmetrici e gameplay profondamente diversi tra i vari dispositivi (Fig.7).



(a) PC - First person shooter. Il *soldati* prendono controllo di punti strategici facendo guadagnare risorse al *comandante*.



(b) Mobile - Strategico. Il *comandante* gestisce la conquista dispiegando soldati e risorse, lanciando attacchi missilistici e proteggendo i propri soldati dagli avversari.

Figura 7: Battlefield 4 - Multiplayer con interoperabilità pc\mobile

2 Design

3 Direzione artistica

4 Sviluppo

Conclusioni

Riferimenti bibliografici

- [1] http://www.theesa.com/wp-content/uploads/2018/05/EF2018_FI-NAL.pdf