The Starving Game : un gioco roguelike multipiattaforma open-source - Human Computer Interaction-

Marioemanuele Ghianni
E-mail address
marioemanuele.ghianni@stud.unifi.it

Abstract

Questo progetto finale è stato realizzato per l'esame del corso di Human Computer Interaction tenuto dal Professore Andrew D. Bagdanov; l'obiettivo è di realizzare un semplice prodotto videoludico usufruibile su varie piattaforme (PC e dispositivi mobile) con un focus particolare sull'esperienza di gioco, sui feedback visivi e sonori resi all'utente e sull'integrazione dei comandi per dispositivi mobile.
In questo paper verranno descritte le varie fasi del processo creativo a partire dal needfinding, l'implementazione vera e propria e i test di usabilità.

1. Introduzione

1.1. Concetto di roguelike

[5]"Roguelike" è un termine che indica un particolare tipo di videogiochi di ruolo, caratterizzato da alcuni elementi comuni tra cui: mappe sviluppate su livelli e generate in maniera casuale o semicasuale, un gameplay basato su turni, un sistema di movimento su griglia e la morte permanente del personaggio principale.
Il nome del genere, traducibile dall'inglese come "simile a Rogue" o "stile Rogue", si riferisce a Rogue, videogioco del 1980 basato sul gioco di ruolo Dungeons Dragons.
Da allora, sono nati una serie di giochi di impronta roguelike che però hanno iniziato a introdurre e mischiare caratteristiche di altri generi

Figure 1. icona dell'applicazione.

fino a un vero e proprio boom commerciale del genere con titoli come The Binding of Isaac, Spelunky e, il più recente, Hades.

1.2. The Starving Game come roguelike/roguelite

Entrando nello specifico di questo progetto, il gioco sviluppato condivide un impianto classico a turni, un sistema di movimento su griglia e livelli generati in maniera pseudocasuale ma si è scelto in fase di progettazione di introdurre una meccanica di tipo "survival" e renderla centrale nel sistema di giocoa cercando di offrire una variante al gameplay standard dei classici roguelike.
Nelle sezioni successive verranno poi spiegate nel dettaglio le meccaniche e il loop di gioco.

1.3. Unity

Per la realizzazione di questo progetto è stato scelto di utilizzare Unity per i motivi che verranno di seguito esposti.
[4]Unity è un motore grafico multipiattaforma rilasciato nel 2005 e che si è reso sempre più popolare nel tempo grazie alla sua ampia acces-

sibilità e la sua capacità di spaziare e sviluppare prodotti videoludici e simulazioni grafiche sia in ambito 2D, 3D, VR e realtà aumentata.
Essendo principalmente gratuito è diventato lo strumento principale degli sviluppatori indipendenti di videogames e molti tra i più recenti giochi di successo sono stati sviluppati tramite Unity; tra questi rientrano titoli come Hollow Knight, Ori e Fall Guys.
Il motore grafico è scritto principalmente in C++ ed espone agli sviluppatori un' API di scripting in C#, un'interfaccia grafica sotto forma di editor e un sistema di gestione degli eventi.
Lo sviluppo di un gioco su Unity, in termini di programmazione, consiste dunque sulla scrittura di una serie di script C# [2] che definiscono le interazioni tra gli oggetti di scena andando a definire una programmazione ad oggetti ma con al centro gli eventi e la gestione di essi.
Essendo un motore grafico multipiattaforma, l'applicazione è stata sviluppata sia per PC (Windows, Mac e Linux) sia per dispositivi mobile Android e infine resa anche giocabile da browser(build WebGL).
Si è partiti dunque con lo sviluppo di una build per PC per poi integrare e programmare i comandi mobile adattando l'interfaccia alle varie risoluzioni.

2. Needfinding

La prima fase dello sviluppo dell'applicazione riguarda il Needfinding: esso si configura come un processo in cui si osservano e intervistano dei candidati per scoprire i loro bisogni, obiettivi e valori riguardante un particolare prodotto.
A partire da questi si costruiscono delle Personas, dei requisiti e degli scenari da cui estrarre poi una prima idea del software.

2.1. Esigenza di un roguelike open-source

Una delle prime necessità che ha dato vita al progetto è stata, come da titolo, la carenza di giochi roguelike open-source; ci sono infatti moltissimi giochi roguelike in commercio ma sono veramente rari quelli open source e quindi adatti a successive modifiche del codice per essere integrati in altri progetti.
La necessità, in particolare, è nata in ambito di ricerca universitaria dove poteva essere utile, per motivi di varietà nel testing, un gioco roguelike con meccaniche semplici che esponesse una competizione tra il giocatore e l'AI nemica.
Quest'ultima, in particolare, doveva poi successivamente essere addestrata con algoritmi di Reinforcement Learning per valutare l'efficacia degli agenti.

2.2. Sondaggi

A seguito di questa prima esigenza, che poi diverrà secondaria e temporaneamente accantonata nel corso dello sviluppo del gioco, è stato aperto un text-channel su un server Discord di una community di videogiocatori.
Discord è una piattaforma statunitense di VoIP, messaggistica istantanea e distribuzione digitale progettata per la comunicazione ed è attualmente la piattaforma più popolare tra le comunità di videogiocatori principalmente PC.
Nel text-channel è stato chiesto alla community le proprie esigenze in termini di roguelike, meccaniche di gioco, caratteristiche tecniche che dovrebbe avere l'applicazione, piattaforme da supportare e anche consigli su che video e tutorial visionare per prendere dimestichezza con lo sviluppo di videogames.
Gli utenti del canale Discord sono principalmente ragazzi universitari o comunque giovani lavoratori e dunque in un range dai 18/19 ai 26/27 anni.
In sostanza, da questi sondaggi è emerso, nell'ambito delle meccaniche di gioco, una preferenza verso un roguelike con un sistema di combattimento che si discostasse dai classici e non troppo lento e old-style se implementato a turni.
Il riscontro inoltre è stato molto positivo alla proposta di un gameplay contaminato e incentrato su meccaniche survival invece che incentrato sull'azione e il combattimento.

Per quanto riguarda le piattaforme , i dispositivi mobile sono risultati i preferiti con l'esigenza di avere controlli touch pratici e responsivi.

2.3. Personas

A seguito dei sondaggi sono dunque emerse due particolari Personas:

- **Riccardo, 23-28 anni**, è un ragazzo che lavora in un' azienda nella città vicina a quella in cui abita; lavora dalle 9.00 del mattino alle 18.00 durante la settimana mentre nel weekend solitamente gioca a calcio con gli amici.
  E' un assiduo videogiocatore; fin da bambino ha giocato vari tipi di videogiochi compresi vari roguelike, dai più classici ai più moderni, su varie piattaforme.
  Lavorando durante il giorno, Riccardo riserva un paio d'ore dopo cena a videogiocare e la sua piattaforma preferita è il PC.
- **Edoardo, 18-25 anni** , è uno studente che frequenta l'università lontano da casa e quindi passa molto tempo tra autobus e treni.
  Nell'adolescenza usava videogiocare raramente e adesso, con i vari impegni tra università e sport, non gioca più al PC e si limita a qualche gioco di breve durata per cellulare nei tempi morti e nei mezzi pubblici.
  Anche se non ha mai giocato a un roguelike e non conosce né i classici né le più moderne iterazioni del genere, è tuttavia curioso e aperto a nuovi giochi, purchè non troppo impegnativi.

2.4. Scenari e requisiti

Sono stati dunque definiti dei possibili scenari in cui si possono trovare le personas sopra definite:

- Edoardo ha una breve pausa tra le lezioni e in questo tempo morto vorrebbe fare una partita veloce; ha un cellulare vecchio e poco spazio in memoria quindi cerca un'applicazione molto leggera, poco impegnativa e dunque dal gameplay immediato per passare qualche minuto di svago.

- Riccardo vuole fare una partita dopocena con calma al PC ragionando meglio sulle mosse e prendendosi del tempo per elaborare strategie e migliorare la run fatta in pausa pranzo in quanto il computer è la piattaforma dove preferisce giocare, quando possibile.

- sempre Edoardo, dopo le lezione universitarie, necessita di un pò di svago sul treno di ritorno verso casa.
  Cerca un gioco immediato, con partite brevi e con controlli mobile comodi ed efficaci anche quando si trova in spazi ridotti del treno per l'affollamento.

Da questi scenari sono stati dedotti una serie di requisiti tra cui la natura **multipiattaforma** del titolo, con particolare attenzione alla versione **mobile** e l'**immediatezza** e **chiarezza** del prodotto in questione.
In particolare, il gioco deve poter essere chiaro anche su piccoli schermi e non deve richiedere particolari sforzi visivi o controlli touch troppo complessi dovendo adattarsi a sessioni di gioco su mezzi pubblici.
Inoltre, lato gameplay, deve portare un pò di varianza rispetto ai classici per avere un certo appeal nei confronti di un videogiocatore esperto che ha consumato vari roguelike.

3. Paper prototyping e ispirazioni

In questa fase sono stati inizialmente disegnati alcuni sketch dell'interfaccia principale del gioco (Fig. 2).

Si è partiti dunque da un modello piuttosto semplice e funzionale per poi visionare un pò su internet l'interfaccia dei più moderni roguelike.
In particolare, è risultata molto d'ispirazione l'interfaccia iniziale del famoso "The Binding

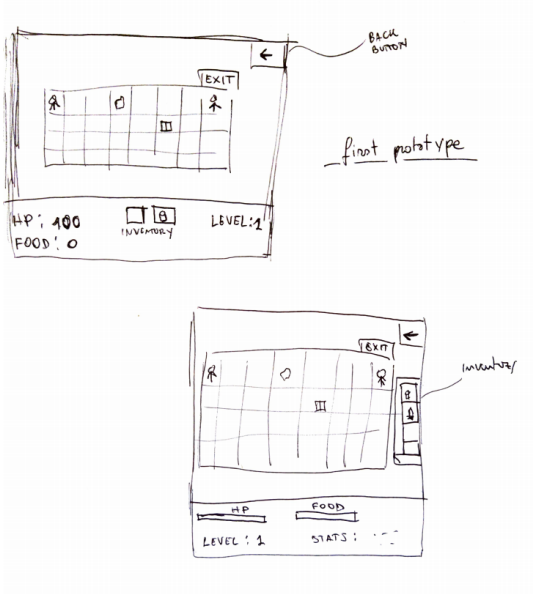

Figure 2. I primi sketch dell'interfaccia principale di gioco.

of Isaac" in quanto ha portato a riflettere, anche in ottica mobile, sulla collocazione dell' HUD e dell'inventario (Figura 3).
Infatti, prevedendo di orientare il gioco in modalità "Landscape" e di usare controlli touch diretti sulla board di gioco, è stato più funzionale e più intuitivo una collocazione degli elementi dell'interfaccia in alto, discostandosi dai primi sketch.

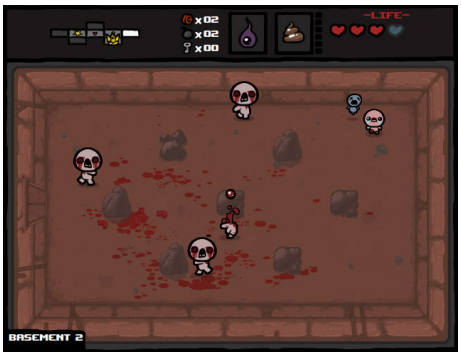

Figure 3. Una delle prime interfacce di gioco di "The Binding of Isaac".

4. Implementazione

L'implementazione, come precedentemente accennato, è costituita fondamentalmente da una serie di scene e oggetti istanziati tramite l'editor Unity e una serie di script C# che si interfacciano con l'editor e con l'engine e permettono le varie interazioni del gioco.
La programmazione su Unity è comunque una programmazione fortemente orientata ad oggetti ma che incentiva la frammentazione del codice in svariati script con conseguente confusione e difficoltà di organizzazione.
In questo contesto, non risulta facile applicare i concetti cardine del paradigma **MVC (Model-view-controller)** visti nel corso.
In particolare, si è ottenuta un'ottima **modularità** del codice ma risulta complesso testare singolarmente e indipendentemente ogni componente in un sistema come Unity.
In ogni caso, come si può vedere in figura 4, il codice è stato organizzato attorno a una classe *GameManager* che rispetta il design pattern **singleton** in modo da fornire un unica istanza accessibile globalmente e, nel contesto di Unity, anche tra scene differenti.
Questa istanza dunque funziona da **controllore** processando, rispondendo ai vari eventi e invocando cambiamenti in quello che si può considerare concettualmente il model e la view.
Il **model**, in questo caso è rappresentato concettualmente da una serie di script che corrispondono agli oggetti di scena che possono avere varie interazioni tra cui il giocatore stesso, i nemici, i forzieri, le bombe, le trappole a muro e a terra ecc..
La **view** in questo contesto particolare si può identificare con lo script *boardScript* che si occupa di istanziare gli oggetti sulla board e, in generale, con l'editor e l'engine di Unity che espongono, appunto, una view che, attraverso la pressione di bottoni o elementi dell'interfaccia utente, notifica l'evento al controllore che poi si occupa della sua gestione e di eventuali cambiamenti nel modello.

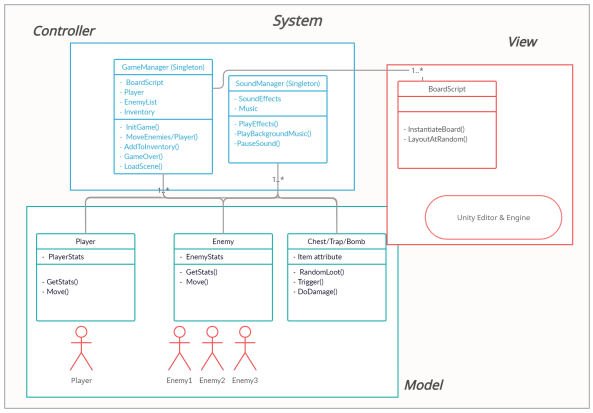

Figure 4. Schema UML riassuntivo e semplificato del progetto.

Figure 5. Menu iniziale e delle opzioni.

5. Il gioco

5.1. Menù iniziale e opzioni

Per quanto riguarda il menù iniziale e il menù dell'opzioni, questi sono piuttosto minimali e contengono solo bottoni e opzioni essenziali in rilievo su uno sfondo animato raffigurante una versione sfocata dell'ambiente di gioco.
Ogni bottone rende sia un feedback visivo che sonoro una volta premuto e i settaggi inseriti nelle opzioni vengono salvati in locale in dei registri di sistema.
Questo fa si che l'utente debba scegliere le proprie impostazioni preferite una sola volta; queste infatti verranno settate automaticamente nelle future sessioni di gioco.

5.2. Interfaccia di gioco principale

Lo sviluppo dell'interfaccia, come anticipato nella sezione dedicata al paper prototyping, è andato avanti ciclicamente per varie iterazioni e modifiche fino ad arrivare al prototipo in alto in figura 6 di cui comunque riconosciamo molte caratteristiche simili allo sketch iniziale ma al contempo anche nuove features e cambiamenti stilistici.
Tuttavia, questa prima interfaccia era stata sviluppata su PC senza considerare un eventuale resa mobile e infatti, nel momento in cui si è passati allo sviluppo mobile sono venute fuori non poche problematiche di chiarezza e di praticità.
Nello specifico, i caratteri dell'interfaccia e i dialoghi che apparivano risultavano troppi piccoli e necessitavano di uno sforzo visivo su schermi piccoli mentre, per quanto riguarda il menù, era addirittura complicata l'interazione stessa.
A seguito di queste problematiche è stato fatto un ulteriore restyling e redesign fino ad arrivare all'interfaccia finale molto più mobile-friendly, pur rimanendo adatta al gaming su PC.
In sostanza, nella versione finale, abbiamo le statistiche del personaggio, una barra della vita e una barra dei food points in alto a sinistra mentre nel lato destro si trovano gli slot inventario, un pulsante per disabilitare i suoni e un pulsante di "help" per i nuovi giocatori che mostra un pannello con "le regole del gioco".
Questo è stato inserito successivamente per migliorare l'accessibilità del gioco; alcuni giocatori hanno interesse nello scoprire via via le meccaniche mentre alcuni possono provare un pò di smarrimento e l'introduzione di questo pulsante si rivolge principalmente a quest'ultimi.
Per concludere, gli slot dell'inventario sono interagibili e con un click o un touch viene utilizzato l'oggetto presente.

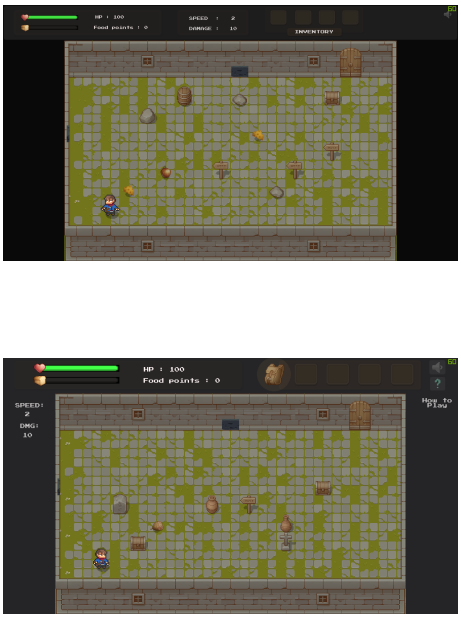

Figure 6. In alto una delle prime configurazioni dell'HUD, in basso quella definitiva.

Figure 7. Schermata di help che illustra le regole del gioco.

5.3. Meccaniche di base

Alla luce delle interviste e di una serie di ricerche su video, tutorial e documentazione varia per lo sviluppo su Unity, è stato scelto, come accennato, di optare per un roguelike a turni, con un sistema di movimento su griglia e con al centro una meccanica survival.
Molto utile come base di partenza è stato il corso-tutorial ufficiale fornito da Unity stesso per la creazione di un roguelike.
L'idea di base del gioco è stata di incentrare il gameplay invece che sull'azione, sulla velocità e su un combat system elaborato, su un gameplay più tattico e basato sul cercare di accumulare più risorse rispetto all'avversario per sopravvivere più di lui in una sorta di competizione.
In particolare, il giocatore si troverà davanti, per ogni livello, uno o più nemici e dovrà cercare di raccogliere più cibo di loro per evitare che i propri HPs scendano a zero andando incontro al game over.
Per maggiore dinamicità del gameplay, è possibile "attaccare" l'avversario se non si ha cibo (a differenza sua) per rubarlo e ovviamente vale anche il viceversa.
Ciò comporta un approccio diverso al gioco in base alla situazione; nel caso il giocatore riesca ad ottenere più cibo degli avversari dovrà cercare di fuggire da loro fino alla loro morte per mancanza di cibo, mentre, se si trova lui stesso in mancanza di cibo, dovrà cercare di aggredire l'avversario per appropriarsi delle sue scorte.
Il tutto si traduce in due modalità in cui possono trovarsi giocatore e nemici: *la modalità "sazio"* che si ha quando si è raccolto del cibo e quindi i "food points" sono maggiori di zero; in questa modalità, ogni turno si consumano 2 food points, si ripristina 1 HP e non si può attaccare i nemici.
la modalità affamato invece si attiva automaticamente quando i food points sono a 0 e prevede il raddoppio di velocità, la perdita di 2HPs per turno e la prossibilità di attaccare per rubare cibo.
Ovviamente a queste regole di base sono stati inseriti elementi quali casse che contengono equipaggiamento (e, dunque, potenziamenti e pozioni), trappole a muro, trappole a terra, bombe, tagliole e altro che permettono, se sfruttate a dovere, al giocatore di scappare più agilmente o di prendere il cibo e le altre risorse più in fretta degli avversari.
Il "loop di gioco" poi consiste in una serie di stage con nemici di statistiche variabili e dalla numerosità crescente logaritmica ma comprende anche una stanza che contiene un altare che fornisce power-up e due boss principali con il secondo dotato di una meccanica unica.

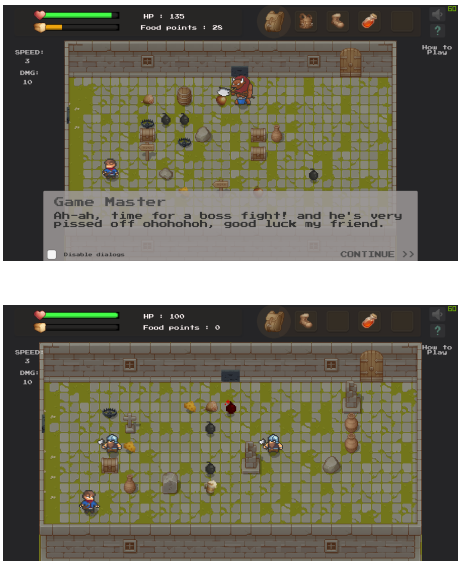

Figure 8. Alcuni screenshot di fasi di gioco avanzate.

5.4. Eventi e interazioni

Il gioco è stato programmato e strutturato per essere, nei limiti delle meccaniche, il più interattivo possibile con l'utente e capace di rendere un feedback distinguibile per ogni possibile interazione.
Il tutto è realizzato tramite l'*event system* di Unity che, se definite, invoca in automatico delle funzioni specifiche per dei particolari tipi di eventi; nel caso delle collisioni, ad esempio, viene utilizzato un componente collider che quando interseca un altro collider invoca la funzione *"OnTriggerEnter2D"* utile magari per dare il via a un'animazione o triggerare a catena un altro evento.
Stesso discorso vale per gli input utente quali la pressione di pulsanti e l'uso dei comandi di gioco che verranno poi approfonditi in seguito.
Attraverso gli eventi vengono, come detto, gestite le varie animazioni e, soprattuto, i suoni; il gioco comprende effetti sonori distinti per ogni tipo di interazione a partire dai menù fino al suono dei passi nello spostamento dei personaggi sulla board.
Grazie al feedback sonoro l'utente comprende appieno cosa sta succedendo e riesce ad assimilare più in fretta alcune meccaniche di gioco.
A tal proposito sono stati usati anche particolari effetti visivi; nella modalità "affamato" infatti il personaggio avrà dei "blink" rossi ogni turno e avrà un blink verde al raccoglimento di cibo.
Questo permette di comunicare visivamente al giocatore la necessità urgente di porre rimedio alla situazione e allo stesso tempo istruire i più "frettolosi" che non leggono le regole o i dialoghi alla meccanica principale del gioco.
Inoltre il cibo presente sulla board è stato animato con un effetto fluttuante in modo da essere subito evidente all'occhio umano, focalizzando l'attenzione e comunicando immediatamente che è possibile raccogliere quel tipo di oggetti e quindi istruendo alla meccanica fondamentale del gioco.

5.5. Controlli

Per quanto riguarda i controlli della versione PC, semplicemente vengono usate le frecce direzionali o i pulsanti WASD per il movimento e il mouse per interagire con l'inventario e con i menù di gioco.
L'aspetto più interessante è forse la versione mobile del gioco che sostituisce semplicemente la navigazione dei menù tramite mouse con il touch ma su cui è stato fatto un pò di lavoro per quanto riguarda i controlli di movimento.
Inizialmente era stato implementato il movimento tramite la gesture di swipe; uno swipe, ad esempio, verso destra spostava il personaggio di una casella a destra.
In fase di testing questi controlli sono risultati poco adeguati in quanto è risultato frustrante fare una dozzina di swipe per attraversare la board e, in generale, comportavano un ritmo di gioco molto lento.
A questo punto è stato pensato di utilizzare un joystick virtuale in modo da avere un input continuo; per attraversare la board verso destra, in questo caso, basta tenere il joystick orientato verso destra rispetto all'origine invece di fare una dozzina di swipe a destra.

Inizialmente questo è stato implementato fisso ma non è stato possibile trovare una collocazione che non desse fastidio all'utente oscurando caselle della board o elementi dell'interfaccia.
Dunque, successivamente, tramite la scrittura di alcuni script, è stato reso il joystick virtuale dinamico e quindi "a comparsa" nella porzione di schermo dove avviene il touch in modo che l'utente possa avere un controllo completo e a tutto schermo che non ostruisse la visuale.
In fase di usability test verrà poi approfondita questa parentesi sui controlli.

6. Deployment

Terminato lo sviluppo, è stata inizialmente creata una build standalone per PC (Windows, Linux Mac) per poi generare un' apk Android.
Verificato il corretto funzionamento di quest'ultime, è stato valutato dove pubblicare l'applicazione ed è stato scelto il sito web **itch.io**.
Itch.io è un sito web che consente agli utenti di ospitare, vendere e scaricare giochi indie.
Creato nel marzo 2013 da Leaf Corcoran, il servizio rende disponibili quasi 100 000 giochi e articoli a partire da febbraio 2018.
Itch.io è specializzato inoltre nell'ospitare dei game jam, ovvero eventi in cui i partecipanti hanno un tempo limitato (di solito 1-3 giorni) per creare un gioco.
Particolarità di questo sito è che permette l'hosting gratuito dei videogiochi senza un pagamento una tantum o annuale.
E' stata dunque sviluppata una build WebGL5 e quindi compatibile per essere giocata da browser ed è stata creata una pagina sul sito con una descrizione base e una sezione commenti.
In particolare è stato modificato il javascript per rilevare se si sta visitando la pagina da mobile e attivare i controlli touch di conseguenza.
Riassumendo, il gioco è dunque giocabile da browser da PC e mobile ma anche scaricabile sulla pagina:
`https://ema-tko.itch.io/thestarvinggame`.

7. Usability Test

Seppur lo sviluppo è stato principalmente ciclico, con varie iterazioni di test seguite da fasi di re-design e di modifica di meccaniche o elementi di interfaccia, l'ultima fase concettualmente è quella dove si va a valutare il lavoro svolto tramite un'analisi dell'usabilità.
Nelle prime fasi di sviluppo, e dunque con i primi prototipi dell'applicazione, sono stati svolti una sorta di " *pilot test*" su un campione molto ridotto di utenti facili da contattare per far emergere alcuni problemi strutturali che lo sviluppatore, senza un punto di vista esterno, spesso non nota.
I test si sono svolti, anche a causa delle restrizioni per la pandemia, tramite canale Discord [1] e dunque da remoto.
Sono stati dunque contattati 5 utenti che rientrano nella categoria dei videogiocatori ed è stato condiviso l'eseguibile dei primi prototipi.
E' stato poi chiesto a questi utenti di condividere lo schermo durante l'uso dell'applicazione, fare apertamente domande al sottoscritto e evidenziare problematiche e miglioramenti come nella situazione descritta in figura 9.
Da questi test sono emerse gran parte delle features e dei miglioramenti descritti in precedenza; inizialmente sono stati evidenziati alcuni **bug nella navigazione dei menù** e l'assenza di alcuni pulsanti, poi sono emerse varie **problematiche con i controlli mobile** (risolte con l'implementazione finale del controller virtuale dinamico descritta in precedenza) e infine varie migliorie da apportare al gameplay vero e proprio per renderlo più divertente e appagante.

In particolare è emersa una certa frustrazione dovuta alla troppa **randomicità del gioco** e dei pochi strumenti a disposizione del giocatore per far valere le proprie scelte e ricevere appagamento da esse.
A seguito di queste osservazioni sono stati poi implementati elementi di gameplay quali le

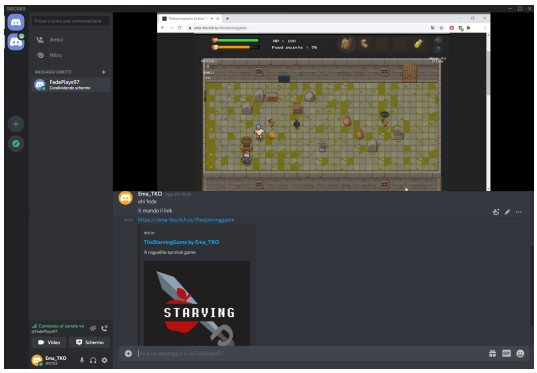

Figure 9. Configurazione di test remota su Discord.

trappole, le bombe e alcuni potenziamenti che permettessero al giocatore di effettuare determinate giocate tattiche per ridurre la randomicità.
Arrivato il momento della prima release, questo test è stato esteso a 12 videogiocatori a cui è stato chiesto di fare almeno 3 partite esplorando le varie funzionalità del gioco per poi venire sottoposti a un sondaggio di tipo SEQ tramite un form di Google (come in figura 10; la tabella riassuntiva del test è invece presente a fine relazione).
Anche in questa fase, sono emersi molti commenti interessanti durante le sessioni di gioco a schermo condiviso, tra cui:

- Molti utenti hanno apprezzato lo stile retro del gioco e delle musiche così come il tono ironico e autoironico del progetto.

- C'è stato un generale apprezzamento e stupore per le meccaniche survival e per le dinamiche di interazione con l'AI nemica seppur alcuni utenti hanno trovato alcune fasi del gameplay dal ritmo lento e poco avvincente.

- A seguito della prova su dispositivi mobile c'è stato un consenso praticamente unanime nel definire i controlli adeguati e intuitivi.
  Alcuni utenti hanno anche affermato di preferire la versione mobile grazie ai controlli e al layout mobile-friendly.

- Dopo alcune partite sono emerse alcune critiche ai comportamenti dell'AI nemica che, essendo semplice e scriptata, compie azioni spesso illogiche o prevedibili in alcune situazioni.
  Sempre a seguito di varie partite, alcuni soggetti hanno evidenziato la mancanza di varietà di equipaggiamento, nemici e stanze.

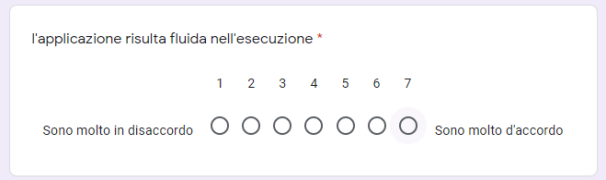

Figure 10. Form di Google utilizzato per il sondaggio.

8. Analytics

Successivamente ai test di usabilità è stato svolto anche un lavoro di analitica post-release.
L'analitica post-release è servita principalmente per valutare alcuni elementi di gameplay in un contesto di post-release e quindi con un bacino d'utenza e temporale molto maggiore rispetto a quello dei test precedenti.
Su itch.io è presente una sezione analitica molto basilare che però permette comunque di vedere l'andamento delle visite e dei giocatori giornalieri come in figura 11
l'apporto più interessante invece, in termini di analitica, è dato da **Unity Analytics** [3] che permette, attraverso alcune funzioni particolari da inserire negli script di gioco, di inviare a un server di analitica degli eventi customizzati.
Si possono dunque inviare in automatico alcune statistiche sulla partita in corso e queste vengono esposte comodamente tramite una dashboard che permette svariate visualizzazioni grafiche.

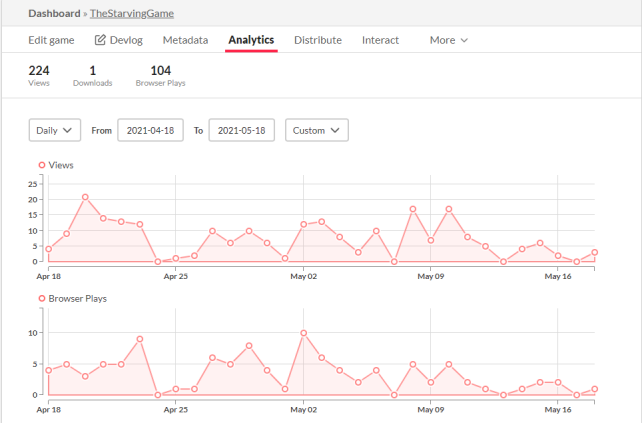

Figure 11. Analitica delle visite e delle partite su itch.io .

Tramite questo tool esterno di analitica, è stato possibile valutare l'uso di determinati elementi di gameplay ed è stato possibile effettuare un primitivo bilanciamento degli oggetti e delle trappole.
Interessante è stato anche utilizzare questo strumento per valutare l'andamento della progressione dei livelli del gioco ed eventualmente bilanciarlo.
Come visibile in figura 12, si è dapprima valutata la progressione temporale, vedendo appunto come, col passare dei giorni, il livello medio raggiunto dai giocatori è diventato via via sempre più grande all'aumentare della loro conoscenza delle meccaniche.

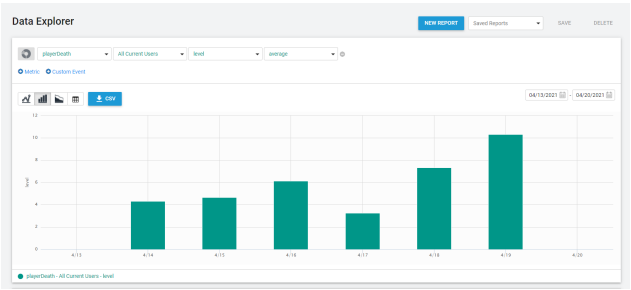

Figure 12. Grafico che mostra il livello medio raggiunto da parte dei giocatori col passare dei giorni e quindi con la comprensione delle meccaniche.

Successivamente, è stata analizzata, livello per livello, la percentuale di giocatori che riuscivano a superarlo in modo da identificare livelli critici e gestire la curva di difficoltà.
Questi dati sono poi espressi in un grafico "funnel" come in figura 13 .
Da questo grafico è particolare notare come inizialmente, il boss al livello 5 causava la maggior parte dei gameover; infatti, molto meno di 1/3 dei giocatori che vi arrivavano riuscivano a superarlo.
Si è dunque provveduto a implementare un piccolo "nerf" alle statistiche del boss in modo da rendere la curva di progressione un pò più smooth e quindi aumentare le possibilità del giocatore di superare il livello e trovarsi a livelli successivi.
Come è visibile dal grafico, adesso circa la metà dei giocatori riesce con successo a superare il livello e questo è solo un esempio dell'utilità dell'analitica post-release.

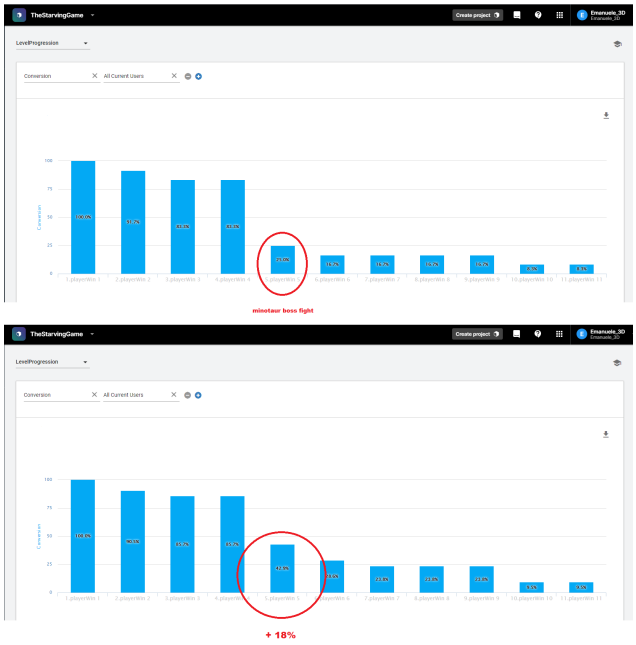

Figure 13. Grafico che mostra il prima e il dopo un intervento di "nerf" su un boss presente al livello 5 che provocava la maggior parte dei gameover.

9. Conclusioni e sviluppi futuri

In questo progetto è stata dunque sviluppata un'applicazione videoludica multipiattaforma con particolare attenzione alla Human Computer Interaction attraverso un processo ciclico di needfinding, implementazione e test di usabilità.
In conclusione è stato dunque sviluppato un eseguibile per pc, un'applicazione android ed è stato pubblicato il gioco su itch.io rendendolo giocabile da browser sia su pc che dispositivi mobili.
Il gioco è stato accolto positivamente dai giocatori che lo hanno testato e ha registrato oltre 100 partite in un mese su itch.io senza essere stato pubblicizzato e senza aver condiviso su vari social o siti il link.
Per quanto riguarda bug o glitch, sono stati fatti vari update ed il gioco ne è praticamente esente anche se, ovviamente, ci sono vari aspetti da migliorare per rendere più accattivante e varia l'esperienza videoludica.
Infatti, alcune considerazioni e critiche ricevute dagli utenti sono state focalizzate principalmente su:

- maggiore varietà di ambienti e stanze di gioco per livello, maggiore varietà di nemici e oggetti inventario.
- l'AI ha comportamenti che, alla lunga, diventano molto prevedibili e quindi in parte aggirabili.

Entrambi questi difetti sono frutto di tempi di sviluppo ridotti e, in particolare, non è stata aggiunta quantitativamente molta varietà al gioco perché non era il focus principale del progetto.
Allo stesso modo, l'AI è composta da semplici script in quanto inizialmente l'idea era di portare un gioco open source dove fosse possibile integrare tecniche di RL per migliorare proprio quest'ultima.
In conclusione, questi sono i due campi principali per rendere ancora più competitivo il prodotto anche se, questo progetto comunque risulta essere una buona base per un roguelike open source, capace di intrattenere e di funzionare bene su varie piattaforme come emerso dai test di usabilità e dalla parte relativa all'analitica.

n.	Domande	media	σ
1	l'applicazione ha un aspetto grafico distinguibile e coerente con il "mood" e le meccaniche di gioco	5.67	1.88
2	l'applicazione risulta fluida nell'esecuzione	6.73	0.61
3	i menù e la loro navigazione sono poco intuitivi / la collocazione di alcuni pulsanti crea confusione	1.92	1.26
4	la musica di background e i suoni sono fastidiosi e non si adattano bene alla tipologia di prodotto	2.73	1.48
5	risulta semplice regolare i suoni dal menù delle impostazioni	7.00	0.00
6	il feedback audio e video per ogni azione è soddisfacente	5.82	1.34
7	si imparano molto facilmente e velocemente le meccaniche di gioco principali	5.89	1.45
8	la GUI in-game risulta chiara e funzionale	6.18	0.94
9	la difficoltà del gioco risulta frustrante	2.25	1.69
10	ho trovato il confronto con l'AI poco appagante e, a volte, tedioso	3.89	1.91
13	ho compreso la centralità della meccanica del cibo e ho apprezzato questa variante di gameplay	5.82	1.53
14	avrei preferito un gameplay più action e un combattimento più classico	3.50	1.65
15	il gameplay ha varietà e si presta a multiple sessioni di gioco	2.56	1.34
16	i controlli mobile sono funzionali e intuitivi	6.75	0.62
17	In conclusione, la mia esperienza di gioco è stata positiva	6.11	0.99

Table 1. Tabella illustrativa del sondaggio di usabilità di tipo SEQ con scala da 1 a 7.

References

[1] Discord. *Discord — Your Place to Talk and Hang Out*. URL: https://discord.com/brand-new.

[2] Microsoft. *Guida per programmatori C*. 2017. URL: https://docs.microsoft.com/it-it/dotnet/csharp/programming-guide/.

[3] Unity. *Leverage your game usage data with Unity Analytics*. 2015. URL: https://unity.com/features/analytics.

[4] Unity. *The leading platform for creating interactive, real-time content*. URL: https://unity.com/.

[5] Wikipedia. *Roguelike*. URL: https://en.wikipedia.org/wiki/Roguelike.