# Progetto di ingegneria del software (rev. 4.2)

### Anno accademico 2023-2024

Membri del team

Tomasoni Francesco (1080980)

Maccari Luca (1080980)

Indice

Il progetto\_\_

* Project plan\_\_

1.1 Introduzione

1.2 Modello di processo

1.3 Organizzazione del progetto

1.4 Standards, linee guida e procedure

1.5 Attività di gestione

1.6 Rischi

1.7 Membri

1.8 Metodi e tecniche

1.9 Garanzie di qualità

1.10 Package di lavoro

1.11 Risorse

1.12 Budget e pianificazione

1.13 Cambiamenti

1.14 Consegna

**Il progetto**

Realizzazione in Java di una versione “a modo mio” ispirandosi al noto gioco Space Invader

Il giocatore ha lo scopo di difendere la terra da un’invasione aliena, distruggendo le navicelle aliene che lentamente si avvicinano

Guida rapida al gioco

**3. Tipi di Invader**

* **Alieni Standard:** Si muovono lateralmente e scendono gradualmente verso il basso.
* **Alieni Speciali:** Alcuni invasori sono più veloci o cambiano direzione più rapidamente.
* **Capo Alieno (Boss):** In alcuni livelli, potresti affrontare un capo alieno più grande e difficile da sconfiggere.

**4. Barriera Protettiva**

Tra te e gli invasori ci sono delle barriere protettive. Esse si deteriorano con il tempo e vengono colpite dai proiettili, quindi stai attento al loro stato.

**5. Strategie di Gioco**

* **Tattica dell'Avanzamento:** Sposta la tua navicella da lato a lato per evitare i proiettili nemici e cerca di avanzare gradualmente verso gli invasori.
* **Priorità di Attacco:** Fai attenzione agli invasori più vicini al basso, poiché raggiungeranno il terreno più rapidamente.
* **Pausa e Osserva:** Pratica la pausa e osserva il modello di movimento degli invasori per pianificare i tuoi spari.

**6. Punteggio**

Ogni invasore abbattuto ti guadagna punti, e alcuni potrebbero lasciare bonus speciali. Cerca di ottenere il punteggio più alto possibile!

**7. Livelli Successivi**

Man mano che avanzi nei livelli, la difficoltà aumenta con la velocità degli invasori e i loro schemi di movimento.

**8. Modalità Multiplayer**

Alcune versioni del gioco includono modalità multiplayer. Divertiti a competere contro i tuoi amici per vedere chi può ottenere il punteggio più alto.

Project plan

Si sviluppa il gioco pensando ad una community generica di videogiocatori. Nella fase iniziale del lavoro i requisiti sono decisi all'interno del team di sviluppo, mentre nel momento in cui l'applicazione sarà resa pubblica, gli utenti avranno una parte attiva nel suggerire modifiche e miglioramenti nonché di segnalare eventuali bug attraverso le piattaforme social e/o issue di GitHub raggiungibili anche dall'interno del gioco.

L'azienda che si occupa della produzione è una società di sviluppo software costituita da Luca Maccari e Tomasoni Francesco ed è stata costituita con il fine di progettare e sviluppare il software.

L'obiettivo del lavoro è dimostrare di saper applicare ad un caso di studio, i concetti, le metodologie e le tecniche appreso durante il corso. Il progetto deve contenere tutti gli argomenti trattati all’interno dei libri di testo.

1.1 Introduzione

Dopo un'attenta analisi dei requisiti ideati dal team ponendosi nell'ottica di un possibile giocatore, sono stati stabiliti i passaggi da seguire al fine di realizzare una prima versione del sistema richiesto.

A tutte le fasi fi progettazione del gioco prenderanno parte entrambi i membri; infatti, non vi saranno suddivisioni in ruoli, entrambi parteciperemo in egual modo alle diverse fasi del progetto.

Durante la prima riunione operativa si è deciso di sviluppare un'applicazione desktop che implementi le seguenti funzionalità:

//TODO PENSARE ALLE FUNZIONALITA DEL SISTEMA

- Sviluppo di un gioco multigiocatore online

- Il gioco dovrà possedere una grafica piacevole, avere una chat testuale e possibilmente vocale

- Il gioco dovrà supportare più lingue

- Il gioco dovrà essere disponibile per Windows, Mac e Linux

All'interno della riunione si è stabilito anche che per tutto il periodo di progettazione e sviluppo si sarebbero tenuti degli incontri settimanali in modo da favorire la comunicazione e collaborazione tre i membri del team, nonché per discutere sugli obiettivi e avere un feedback dell'intero gruppo di lavoro riguardo alle modifiche apportate.

Si è inoltre posta come deadline per il primo prototipo il 30/12/2023 mentre per la versione alpha il 10/02/2024.

In seguito alla pubblicazione del gioco in versione alpha sarà data la possibilità agli utenti di giocare gratuitamente per un periodo di tempo limitato e di mandare feedback riguardo al gameplay ed eventuali implementazioni e segnalarci eventuali bug.

1.2 Modello di processo

Per il processo di sviluppo del gioco si è deciso di sfruttare un approccio di tipo \*agile\*, infatti, si è ritenuto che questo fosse più adatto per un progetto di questo tipo in cui la documentazione non è essenziale per l'utente finale, ossia i giocatori.

Il processo di sviluppo sarà strutturato nel seguente modo:

* Fase di sviluppo della versione alpha

Durante questa fase verrà sviluppata una prima versione del gioco con tutte le funzionalità di base necessarie al gameplay.

* Fase di manutenzione e aggiornamento

Durante questa seconda fase ci si occuperà di correggere eventuali bug e implementare nuove funzionalità del gioco.

Gli aggiornamenti potranno essere di due tipi: aggiornamenti minori che correggono bug e/o implementano nuove funzionalità minori, e aggiornamenti maggiori che implementano nuove funzionalità principali.

Come piattaforma di Version Control System si è scelto di sfruttare GitHub, in quanto è la piattaforma più utilizzata per lo sviluppo software ed è nota anche dagli utenti, inoltre permette di sfruttare le funzionalità di issue tracking, branch e wiki.

1.3 Organizzazione del progetto

Le persone coinvolte nella progettazione del gioco inizialmente sono quelle che compongono il team. In seguito alla pubblicazione, i giocatori potranno prendere parte allo sviluppo suggerendo nuove funzionalità o segnalando bug, o, se lo desiderano, potranno contribuire direttamente allo sviluppo facendo un \*fork\* del repository, implementando le loro modifiche e aprendo una \*pull request\*. Affinché le modifiche entrino a far parte del gioco dovranno essere revisionate e approvate dal team di sviluppo.

1.4 Standards, linee guida e procedure

//TODO

Il gioco è composto da due applicativi separati:

- Server: Gestisce l'andamento di una o più partite, non possiede un'interfaccia grafica ed è compatibile con i più diffusi sistemi operativi (anche ARM)

- Client: Gestisce l'interfaccia grafica e l'interazione con l'utente

Sia server che client sono scritti in Java e per la comunicazione utilizzano la libreria open source [MCLib-Network](https://github.com/MultiCoreNetwork/MCLib) sviluppata da Lorenzo Magni e basata a sua volta sulla libreria [Netty](https://netty.io) per la comunicazione in rete tramite pacchetti TCP.<

Per la gestione dei file di configurazione e delle traduzioni si sfruttano file json mediante la libreria [Gson](https://github.com/google/gson) sviluppata da Google.<br>

Per la grafica si è deciso di utilizzare la libreria [JCEF](https://github.com/chromiumembedded/java-cef) che permette di integrare all'interno di una finestra Swing un browser \*Chromium\*. Si è deciso di utilizzare questo tipo di approccio in modo da semplificare lo sviluppo dell'interfaccia grafica mediante l'uso dei linguaggi web quali HTML, CSS e JavaScript e permettere una facile modifica (anche da parte degli utenti) delle interfacce del gioco. La scelta dei linguaggi web permette inoltre di avere un rescaling automatico delle interfacce in base alle dimensioni della finestra.<br>

Sempre per la grafica sono state utilizzate le librerie CSS/JS [Bootstrap](https://getbootstrap.com) e [jQuery](https://jquery.com) e [Font Awesome](https://fontawesome.com/) gestite localmente.

### 1.5 Attività di gestione

//TODO

Si è deciso di organizzare lo sviluppo secondo la filosofia dell'\*extreme programming\* in quanto permette di migliorare la qualità del codice e approcciarsi al cambiamento dei requisiti in modo più responsivo.

Sulla base di questo si è deciso di dedicare del tempo alla lettura del codice scritto dagli altri membri del team come metodo di validazione, inoltre si è deciso di applicare l'approccio del \*pair programming\* per la scrittura del codice, con il quale si hanno coppie di sviluppatori che lavorano insieme su un singolo frammento di codice e si dividono il lavoro di scrittura e revisione in tempo reale.

Gli incontri settimanali sono di breve durata, all'incirca 60 minuti, durante i quali possono essere effettuate delle sessioni di \*brain storming\* per trovare nuove idee o cercare soluzioni a problemi riscontrati

Al termine del meeting, ci si accorda sui compiti da portare avanti durante le settimane successive.

// Utilizzando extreme programming non abbiamo organizzato un calendario di incontri preventivo, ma ci siamo organizzati settimanalmente in base al tempo a disposizione del team.

1.6 Rischi

I principali rischi che si possono incontrare durante lo sviluppo di questo gioco con le modalità sopra descritte sono principalmente 2:

1. Il team potrebbe non essere in grado di raggiungere gli obiettivi prefissati entro la deadline. Il prolungamento della fase di programmazione corrisponde ad un aumento dei costi di sviluppo che nella fase iniziale non sono coperti.

2. Il gioco potrebbe non riscontrare l'interesse sperato da parte degli utenti o di potenziali investitori. In tal caso il tempo speso per lo sviluppo non potrà essere pagato.

1.7 Membri

La modalità di lavoro \*agile\* fa si che non ci sia una netta distinzione tra i ruoli nel gruppo di lavoro.

Viste le regole date dall’XP abbiamo evitato la suddivisione in ruoli del team (manager…), in modo da avere la stessa importanza.

Grazie ai differenti skill personali degli sviluppatori coinvolti entrambi i membri del team hanno dato un contributo attivo al progetto sfruttando al meglio le proprie competenze.

1.8 Metodi e tecniche

//TODO

/\*

* **StarUML** per quanto riguarda il design, nello specifico: Use Case Diagram, Class Diagram, Sequence Diagram e Activity Diagram e State Machine Diagram.
* Per la coordinazione del lavoro e dei file abbiamo utilizzato **GitHub**
* **Eclipse IDE** per la realizzazione e sviluppo del codice Java.

\*/

//loro hanno messo diagramma use case

1.9 Garanzie di qualità

Si punta a sviluppare un software che rispetti i parametri di qualità indicati dal modello di McCall:

- Correttezza

- Affidabilità

- Robustezza

- Integrità

- Usabilità

Per garantire la sicurezza degli utenti sono state usate librerie moderne e molto diffuse, aggiornate frequentemente e poco inclini ad avere vulnerabilità.

Per verificare la corretta funzione del programma, nella fase di testing abbiamo testato ogni possibile scenario di gioco immaginabile. Inoltre per controllare la qualità abbiamo condiviso il gioco a terzi per verificare la giocabilità e i possibili errori (bug) del software.

1.10 Package di lavoro

//TODO

Il progetto del software è basato su maven e diviso in 3 moduli: \*client\*, \*server\* e \*common\*. Questo permette di avere una struttura ben organizzata, di poter avere un codice comune su client e server necessario per la comunicazione e di non avere codice inutilizzato sui lati (client o server).<br>

Il client contiene tutta la parte di codice necessaria per la visualizzazione del modello (view) e l'input/output da e verso l'utente, il server invece ha la funzione di controller e gestisce il modello (controller/model).

1.11 Risorse

Le risorse designate allo sviluppo del gioco sono un project manager e due sviluppatori. Il team ha deciso di utilizzare l'IDE [IntelliJ Idea](https://www.jetbrains.com/idea/) per lo sviluppo del software, la suite Office e gli strumenti di GitHub per la scrittura della documentazione e della presentazione e altri strumenti di photoediting per la creazione e modifica delle immagini del gioco e di grafici e diagrammi.

Ogni membro del team ha a disposizione un computer personale, una connessione ad internet e un luogo di ritrovo fisico dove potersi incontrare. Per l'organizzazione del lavoro a distanza e dei meeting è stato anche sfruttato [Discord](https://discord.com/)

1.12 Budget e pianificazione

Come precedentemente specificato, il lavoro non è attualmente retribuito ma i possibili ricavi futuri verranno redistribuiti tra i membri del team in base al lavoro svolto.

Le ore di lavoro sono così suddivise:

1.13 Cambiamenti

Oltre alle features base incluse nella versione alpha, il team ha fissato come obiettivo l'aggiunta di funzionalità aggiuntive e nuove modalità di gioco per fornire un gioco più completo e variegato.

//TODO

- Traduzione del software in altre lingue oltre all'inglese e all'italiano

1.14 Consegna

Una volta pronta la versione alpha, prima di essere pubblicato, verrà inviata una segnalazione tramite issue a tutti i membri del team in modo tale da poter effettuare un'ultima revisione prima della pubblicazione.