**Project Plan – Fake Invaders**

Tomasoni Francesco (1080980) – Maccari Luca (1081951)

Anno accademico 2023-2024

1. **Introduzione**

Il progetto prevede la realizzazione in Java di una versione “a modo mio” del noto gioco Space Invader. L'obiettivo principale è dimostrare di saper applicare i concetti, le metodologie e tecniche appreso durante il corso.

Il giocatore ha lo scopo di distruggere le navicelle aliene che lentamente si avvicinano alla sua navicella, se il giocatore distruggerà tutte le navicelle aliene passerà al livello successivo dove aumenterà la difficoltà.

Il punteggio finale è dato dal numero di alieni uccisi.

Se la navicella del giocatore viene distrutta, il gioco finisce.

Questa versione di gioco oltre ad avere tutte le funzionalità del gioco originale dovrà implementarne ulteriori, tra cui:

- possibilità del giocatore di cambiare tipo di arma

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

- possibilità di visualizzare una classifica dei migliori punteggi ottenuti dai giocatori

- la navicella può muoversi lungo due assi

Il team è composto da: Tomasoni Francesco e Maccari Luca.

1. **Modello di processo**

Il modello da noi utilizzato è l’Extreme Programming, ovvero un modello di tipo agile.  
Per sviluppare al meglio il nostro progetto abbiamo deciso di utilizzare questo modello in quanto rispettava questi requisiti:

|  |  |
| --- | --- |
| XP Practice | Descrizione |
| Pair programming | Tutto il codice viene scritto da due programmatori sulla stessa “macchina”. |
| Collective ownership | Ognuno può modificare il codice quando vuole. |
| Small release | Il codice deve essere funzionante a fine di ogni giornata, attraverso piccole modifiche. |
| Testing | Ogni modifica al codice viene testata all’istante. |

1. **Organizzazione del progetto**

Le persone coinvolte nella fase iniziale della progettazione del gioco sono i membri del team. Dopo la pubblicazione, i giocatori hanno l'opportunità di partecipare attivamente allo sviluppo proponendo nuove funzionalità, segnalando bug o, se lo desiderano, contribuendo direttamente attraverso il fork del repository.

Implementando le proprie modifiche, possono successivamente aprire una pull request. Affinché tali modifiche siano integrate nel gioco, è necessario che vengano attentamente revisionate e approvate dal team di sviluppo.

1. **Standards, linee guida e procedure**

Verrà utilizzato java come linguaggio di programmazione, utilizzando il framework JavaSwing per la realizzazione della GUI.

Gli IDE utilizzati sono VSC e Eclipse.

Per gestire le riunioni del team e assegnazione di compiti viene utilizzato Discord.

1. **Attività di gestione**

Si è deciso di organizzare lo sviluppo seguendo lo schema dell’*Extreme programming* in quanto permette di migliorare la qualità del codice e approcciarsi al cambiamento dei requisiti in modo più responsivo.

Si è deciso di accordare incontri settimanali in base al tempo a disposizione del team sono di breve durata, all'incirca 30 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.

1. **Rischi**

Il rischio principale consiste nell'eventualità di non completare il progetto entro la scadenza prefissata o di consegnare un risultato che non sia completamente funzionale.

Un ulteriore rischio scaturito dal numero ridotto dei membri del team riguarda eventuali contrattempi di una delle due parti che può compromettere le tempistiche prefissate in corso d’opera.

1. **Membri**

La metodologia di lavoro agile elimina una distinzione netta tra i ruoli all'interno del gruppo di lavoro. Seguendo le regole stabilite dalla metodologia XP, abbiamo evitato di suddividere il team in ruoli definiti, al fine di garantire pari importanza a ciascun membro.

Grazie alle diverse competenze individuali degli sviluppatori coinvolti, entrambi i membri del team hanno apportato un contributo attivo al progetto, sfruttando al massimo le proprie abilità.

1. **Metodi e tecniche**

Nel contesto della progettazione, abbiamo adottato StarUML come strumento principale, focalizzandoci in particolare su diverse prospettive, tra cui Use Case Diagram, Class Diagram, Sequence Diagram, Activity Diagram e State Machine Diagram.

Per quanto riguarda la coordinazione del lavoro e la gestione dei file, abbiamo fatto ampio uso di GitHub. Questa piattaforma ci ha fornito un ambiente collaborativo robusto, facilitando la condivisione e il monitoraggio delle modifiche apportate al codice e alla documentazione.

Per la fase di implementazione e sviluppo del codice Java, ci siamo affidati a Eclipse IDE.

In sintesi, l'integrazione di StarUML per la progettazione, GitHub per la gestione collaborativa e Eclipse IDE per lo sviluppo del codice Java ha consentito al nostro team di affrontare con successo le sfide del progetto in modo organizzato e efficiente.

1. **Garanzie di qualità**

L'obiettivo è sviluppare un software che rispetti i criteri di qualità delineati dal modello di McCall, concentrandoci su:

- Correttezza

- Affidabilità

- Robustezza

- Integrità

- Usabilità

Al fine di garantire la sicurezza degli utenti, abbiamo adottato librerie moderne e ampiamente diffuse, soggette a frequenti aggiornamenti e caratterizzate da una bassa propensione alle vulnerabilità.

Durante la fase di testing, abbiamo eseguito una verifica approfondita della corretta funzionalità del programma, esplorando ogni possibile scenario di gioco immaginabile. Inoltre, per monitorare la qualità complessiva, abbiamo condiviso il gioco con terze parti per valutarne la giocabilità e identificare eventuali bug nel software.

1. **Package di lavoro**

Alcuni dei sotto progetti (work package) che sono stati definiti a priori.  
In viste della natura agile del progetto, questi work-package saranno estesi e modificati o evoluti nelle diverse iterazioni del life-cycle:

• Fase di design di schemi UML  
• Imparare a usare java swing   
• Uso di un database  
• Fare il testing sul prodotto

1. **Risorse**

Abbiamo sfruttato le nostre competenze acquisite in precedenza e durante il corso come principali risorse. Dal punto di vista hardware e software, abbiamo impiegato computer con specifiche standard compatibili con l'utilizzo di Eclipse e server cloud.

Questi dispositivi presentavano diverse configurazioni di sistemi operativi e varie risoluzioni dello schermo, consentendo così di verificare l'adeguata risoluzione della GUI.

1. **Budget e pianificazione**

Il budget finanziario è stato di zero euro.

Possiamo considerare come budget il tempo speso, utilizzando il **PERT** (Programming Evolution and Review Techniques).  
Questa tecnica rappresenta le attività in una rete di nodi, nota come “rete di attività su nodo”. Ad ogni nodo viene assegnato un peso, il quale indica la durata dell’attività corrispondente, calcolato usando la formula: (tempo ottimistico + (4 volte il tempo più probabile) + tempo pessimista) / 6.  
I collegamenti con le frecce indicano che un’attività non può iniziare prima del completamente dell’attività precedente.

Immagine che contiene diagramma, Piano, linea, testo

Descrizione generata automaticamente

1. **Cambiamenti**

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

- Traduzione del software in altre lingue oltre all'inglese.

- Possibilità di personalizzazione della navicella e degli alieni.

- Supporto per i dispositivi mobili.

- Implementazione della modalità multigiocatore.

- Implementazione di meccanismi di difesa per il giocatore.

1. **Consegna**

Il project plan verrà consegnato entro il 19 Dicembre.

Il progetto entro la data del primo appello di esame (19 Gennaio).