

# UNIPAC·MAN

## PROJECT PLAN

**REALIZZATO DA**

**BELLOSI JACOPO ( m. 1081058 )**

**LONGHI LARA ( m. 1079261 )**

**POLONI LUCA ( m. 1078817 )**

## Introduzione

In questo progetto realizzeremo un'applicazione java per Windows che replicherà il celebre videogioco arcade Pac-Man degli anni '80 riadattandolo a una narrazione universitaria.

UniPac-Man è stato concepito con l'obiettivo di offrire un'esperienza ludica e formativa, coinvolgendo gli studenti in un ambiente virtuale ispirato alla vita di uno studente universitario.

La scelta è ricaduta su un gioco in quanto vogliamo cimentarci in qualcosa di nuovo per apprendere nuove conoscenze.

I responsabili del progetto sono Bellosi Jacopo, Longhi Lara, Poloni Luca.

## Modello di processo

Per lo sviluppo del processo abbiamo deciso di utilizzare l'approccio *agile*, in particolare utilizzeremo un approccio di *eXtreme Programming*. In quanto abbiamo previsto di non avere dei ruoli precisi all'interno del team e perciò in linea con la pratica *whole team* e proprietà collettiva dove tutto il team è coinvolto in tutto il lavoro e libero di modificare codice in qualunque momento. Essendo neofiti in questo ambito prevediamo di utilizzare anche il *pair programming* in quanto spesso scriveremo il codice su una sola macchina sotto il controllo e l'aiuto di un altro componente del gruppo.

Inoltre, vogliamo apportare cambiamenti di tipo incrementale in quanto ci concentreremo sulla definizione di obiettivi Step-by-Step, ponendo meno attenzione alle tempistiche, concentrandoci sull'oggi e lasciando a domani il lavoro per domani.

In quanto neofiti non possiamo stimare con precisione le tempistiche di sviluppo, perciò, preferiamo non porci deadline temporali per concentrarci maggiormente sui singoli obiettivi, in linea con la pratica XP di *abbracciare il cambiamento*.

## Organizzazione del progetto

Non prevediamo di avere una distinzione netta dei ruoli, abbracciando così le pratiche XP, tutto il gruppo lavorerà in simbiosi su tutte le fasi del progetto e in tutti gli ambiti avendo così la possibilità di verificare e controllare meglio il lavoro svolto. Questa scelta consente di poter lavorare in maniera più dinamica e collaborativa nella stessa fase senza oberare eccessivamente di lavoro una singola persona.

## Standard, linee guida, procedure

Come standard, durante la fase di programmazione, prevediamo di seguire le formalità di Javadoc.

La documentazione verrà aggiornata periodicamente in quanto verranno caricate fin dall'inizio tutte le versioni su GitHub, riuscendo così a tenere traccia di tutte le modifiche. La documentazione seguirà i punti chiave esposti a lezione. La stessa modalità la prevediamo di utilizzare per il codice.

## Attività di gestione

Abbiamo deciso di eseguire incontri quotidiani mattutini di aggiornamento sul lavoro svolto nella giornata precedente all'interno del quale definiamo gli ordini del giorno e gli step da svolgere per il giorno successivo.

## Rischi

Il rischio maggiore risiede nel capire il funzionamento e utilizzo della libreria *Java Swing* per l'implementazione grafica. Potremmo incontrare qualche difficoltà nell'implementazione di alcune funzionalità del gioco, quali la classifica e la coordinazione del movimento dei fantasmini.

## Personale

Il personale del progetto è composto da tre studenti che si divideranno in modo equo e collaborativo i compiti senza prelezioni durante tutto lo sviluppo dell'applicativo.

## Metodi e tecniche

Durante il progetto abbiamo deciso di utilizzare le seguenti funzionalità di *GitHub*:

- Commit: per avere sempre la versione più aggiornata di codice e documentazione;
- Pull request: per chiedere il consenso prima di effettuare variazioni importanti;
- Tag: identificativo per identificare in modo specifico le parti di codice;
- Issue: per porre agli altri componenti eventuali bug o chiarimenti;
- Branch: per apportare le sostanziali modifiche di refactoring e per implementare soluzioni diverse al codice principale.

Per il funzionamento grafico vogliamo utilizzare la libreria *Java Swing*, per l'implementazione della classifica utilizzeremo un database embedded creato con *SQLite*. *Maven* verrà utilizzato per le dipendenze *jar4* su *Eclipse*.

Per i diagrammi UML utilizzeremo *StarUML* con estensione *Rebel* che ci può aiutare nella stesura dello scheletro del codice.

Consideriamo di rendere ogni nuova implementazione funzionante per progredire al prossimo step con un prodotto sempre utilizzabile.

## Garanzia di qualità

Per garantire la qualità seguiremo gli attributi di qualità di McCall, garantendo correttezza, affidabilità, efficienza, integrità, usabilità a livello di funzionamento di prodotto. Terremo in considerazione anche gli attributi di revisione e transizione.

Al termine dello sviluppo eseguiremo test utente con alcuni studenti universitari, consegnandogli un eseguibile user-friendly, in modo da avere un riscontro da parte del consumatore finale.

## Pacchetti di lavoro (workpackages)

Le attività principali del progetto sono:

- Pianificazione;
- Modellazione della struttura del progetto attraverso diagrammi UML;
- Implementazioni delle classi in Eclipse partendo dallo scheletro datoci in *Rebel*;
- Creazione della struttura base: dinamiche di gioco e movimenti;
- Implementazione grafica personalizzata a tema universitario;

Nel corso dell'implementazione viene aggiunta e modificata costantemente la documentazione.

Le varie sotto attività che creeremo verranno suddivise tra i componenti del gruppo in base a disponibilità e competenze, rimanendo sempre un gruppo collaborativo e dinamico.

## Risorse

Per lo sviluppo dell'applicazione prevediamo di utilizzare:

- Computer personali dei membri della squadra;
- Eclipse per lo sviluppo del codice con le diverse librerie;
- *StarUML* per realizzare i diagrammi UML;
- *Structure101* per controllare la struttura del progetto;
- *UCDetector* per fare information hiding e pulizia dei metodi inutilizzati;
- *CodeMR* per l'analisi della struttura del software.

## **Budget e programma**

Il budget destinato a questo progetto non è monetario in quanto la realizzazione non necessita l'acquisto di software e/o hardware, bensì è un budget temporale impiegato prevalentemente per apprendere nuove conoscenze pratiche e teoriche.

## **Cambiamenti**

Le modifiche verranno continuamente tracciate con i GitHub pull ed eventualmente con la creazione di GitHub Branch. I cambiamenti durante lo sviluppo sono inevitabili.

## **Consegna**

La nostra applicazione verrà distribuita gratuitamente tramite un eseguibile destinato agli studenti iscritti presso l'Università di Bergamo.