

# 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, focalizzandoci sulla replica del celebre gioco arcade Pac-Man adattandolo a una narrazione accademica.

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 da grandi fanatici dei videogame abbiamo deciso di cogliere l'opportunità, scegliendo di utilizzare le nostre conoscenze per svilupparne uno.

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 tipo incrementale in quanto ci concentreremo sull'obiettivo e non sulla tempistica.

In quanto neofiti preferiamo non avere una tempistica in quanto preferiamo metterci il tempo più corretto per sviluppare i singoli obiettivi, avendo anche delle competenze da acquisire da zero non sappiamo stimare effettivamente le criticità e i problemi.

## Organizzazione del progetto

Essendo tutti e tre neofiti in quest'ambito abbiamo deciso di non avere una distinzione così netta dei ruoli visto che non abbiamo conoscenze pregresse.

Questo permette di incrementare i tempi in quanto ci aiutiamo a vicenda delle diverse fasi senza oberare di lavoro una singola persona.

## Standard, linee guida, procedure

Come standard da utilizzare prevediamo di seguire Javadoc in tutte le sue formalità.

La documentazione verrà aggiornata periodicamente frequentemente in quanto la caricheremo fin dall'inizio 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 che sarà tutto memorizzato sulla piattaforma.

## Attività di gestione

Abbiamo deciso di eseguire incontri quotidiani mattutini di aggiornamento sul lavoro svolto nella giornata precedente e per delineare gli ordini del giorno.

## Rischi

Il rischio maggiore risiede nel capire il funzionamento e utilizzo della libreria JMonkey, inoltre potremmo incontrare 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;
- Branch per sezioni critiche o differenziazioni personali, il merge sarà eseguito trovata la versione migliore;
- Tag identificativo per identificare in modo specifico le parti di codice;
- Issue per porre agli altri componenti eventuali bug o chiarimenti.

Per il funzionamento grafico vogliamo utilizzare la libreria *JMonkey* e per l'implementazione della classifica utilizzeremo un database embedded creato con *SQLite*. Utilizzeremo *Maven* per le dipendenze su *Eclipse*.

Per i diagrammi UML utilizzeremo *StarUML* con estensione *Rebel* che ci può aiutare nello scheletro del codice. Consideriamo di rendere ogni nuova implementazione funzionante prima di proseguire con l'obiettivo successivo in modo da avere un prodotto sempre funzionante.

## 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. E terremo in considerazione anche gli attributi di revisione e transizione.

Al termine dello sviluppo eseguiremo test utente con persone a noi vicine, 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 di gioco: dinamiche di gioco e movimenti di base;
- Implementazione grafica personalizzata a tema universitario;
- Implementazione database per la classifica.

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

Le varie sotto attività che creeremo verranno suddivise tra i componenti del gruppo in base a disponibilità e competenze.

## 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.

## Budget e programma

Il budget non è a livello monetario in quanto non necessita l'acquisto di software/hardware, ma a livello temporale per apprendere le conoscenze.

## 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 dell'università di Bergamo.