Documentation for Software engineering project

Andrea Brembilla - Matricola 1053908

1 Project Plan

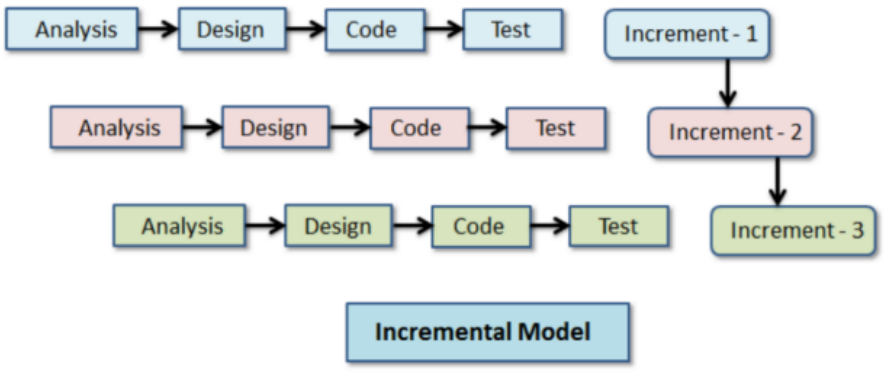
1.1 Introduzione

Questo progetto è stato sviluppato interamente da me, per il corso di Ingegneria software.
Il progetto prevede lo sviluppo di un'applicazione Java per il gioco Brick Breaker: uno strato di mattoncini posizionato nella parte alta dello schermo e l'obiettivo è distruggerli tutti facendo rimbalzare ripetutamente una pallina su un paddle. Tale applicazione è dotata di un'interfaccia grafica principale contenente due interfacce, una per il gioco ed una per le informazioni di gioco, inoltre presenta una funzionalità di registrazione del risultato per una classifica locale.

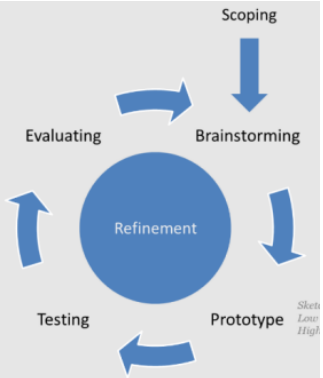
1.2 Modello di Processo

È stato deciso di utilizzare un modello agile dato che il progetto non è di dimensioni elevate. Inoltre, non si vuole essere limitati dal documento iniziale senza avere una specifica dei requisiti troppo estesa, ma si vuole che questa cambiasse con le considerazioni che si faranno con il costruirsi del progetto e senza escludere miglioramenti che potrei trovare lungo il percorso.

Precisamente, dati i requisiti del progetto, già definiti in anticipo, ma che si potrebbero modificare piano piano, è stato scelto il modello di sviluppo incrementale. Le funzionalità del sistema sono prodotte e consegnate al cliente in piccoli incrementi e miglioramenti. Partendo dalla situazione iniziale procediamo in piccoli passi. In ciascuno di questi passi si applica un approccio a fasi (simile a quello del waterfall model). In questo modo, viene messa maggior attenzione sulle feature essenziali: non avendo mai svolto un progetto in java di questo tipo ho preferito innanzitutto capire come fare le feature principali richieste dal cliente (io, in questo caso), e quindi la fisica di gioco e la gestione di interfacce, e poi feature secondarie, che potrebbero anche nascere con lo sviluppo incrementale progetto.



In teoria questo modello è simile al prototyping: si potrebbe infatti fornire un prototipo al cliente ogni qual volta un incremento ne consenta la creazione. Questo prototipo può essere utilizzato come punto di partenza per esplicitare le richieste del cliente mentre il progetto progredisce (questo punto è fondamentale per tutti i modelli agili, in cui collaborare con il cliente è meglio che negoziare all'inizio il contratto), per verificare di star rispettando i requisiti di questa fase e non aver commesso errori.



Abbiamo anche diviso le funzionalità in base a cosa ritengo sia necessario avere per un'ambiente di gioco completo e altre che sarebbe carine fossero aggiunte ma non sono necessarie. Di seguito vi è un elenco, ispirandoci al modello MoscoW:

Funzionalità necessarie	Funzionalità aggiuntive
Interfaccia grafica	Nuovi Powerup
Interfaccia pulsanti fisici	Gestione punteggi
Lettura e scrittura file	Modifica impostazioni di gioco
Fisica di gioco	

1.3 Organizzazione del progetto

Il progetto non ha un vero cliente, dato che sono io l'ideatore e sono l'unico che risponde del lavoro. Per la natura del progetto, esso si interfaccerà con una tipologia di utente, il giocatore che andrà ad utilizzare ogni interfaccia progettata tramite mouse e tastiera. Per portare a termine il progetto ci sono delle lacune da colmare: la creazione di un'interfaccia grafica, l'integrazione di comandi hardware nel software e la fisica del gioco.

Il progetto, come già detto, si basa sul porsi degli obiettivi da raggiungere ogni tot tempo e una volta raggiunti ci si è posto ulteriori obiettivi.

Come milestone sono state scelte le seguenti funzionalità:
- comprensione della Fisica di gioco: approfondimento della fisica di gioco, design, testing ed implementazione di essa
- UI: approfondimento di Java Swing e Java AWT, design della UI
- integrazione: testing ed implementazione della UI

Le tempistiche di ogni milestone sono di durata settimanale, in modo da non avere una mole eccessiva di studio delle funzionalità.

1.4 Standard, Linee guida, Procedure

Siccome il progetto richiedeva l'utilizzo di un linguaggio di programmazione ad oggetti, è stato scelto Java. Questo è stato esteso con i moduli Swing e Awt per creare l'interfaccia grafica. Per quanto riguarda il codice, esso sarà caricato settimanalmente nella piattaforma Github.

Lo standard seguito è quello basato sulle Java Code Conventions di Oracle.

1.5 Gestione Attività

Ogni fine settimana viene stilato un report informale sui progressi in corso eseguiti per avere un'idea sull'avanzamento del progetto. Tale report rappresenta la principale strategia per valutare lo status del progetto e, di conseguenza sono utili a fare course-correction. Difatti è proprio in questo modo che è previsto bilanciare l'equilibrio tra requirments e l'impegno necessario per soddisfarli. Si ricorda inoltre che ci sono prima le varie funzionalità necessarie e poi le funzionalità aggiuntive.

1.6 Rischi

Il rischio principale è di non consegnare nei tempi prestabiliti il progetto oppure consegnarlo, ma non completamente funzionante.
Nella realizzazione di questo sistema c'è la possibilità di entrare in contatto con alcuni rischi che potrebbero creare dei problemi durante l'utilizzo del software. Questi rischi possono essere causati da diversi fattori, da quelli amministrativi a quelli tecnologici. Il rischio principale in cui potremo incombere riguarda la specifica dei requisiti; il fatto di essere io stesso il "cliente" che ha commissionato il lavoro ha obbligato a pensare ai vari requisiti da poter descrivere, l'unico problema è che questi potrebbero risultare troppo limitanti per possibili integrazioni future. Anche durante la fase di design o implementazione la mancanza di un utente potrebbe causare lo sviluppo di alcuni rischi: uno tra tanti potrebbe essere la mancata approvazione del design o di una funzionalità, realizzati per il software. Per quanto riguarda la tecnologia i rischi sono rappresentati dai malfunzionamenti della parte software o della parte hardware. I rischi generati dai software possono riguardare la possibilità che un modulo venga aggiornato o corrotto e quindi "rompa" funzionalità delle classi java.

1.7 Personale

Il progetto è interamente gestito dallo staff di sviluppo, cioè me. Non ci saranno cambiamenti.

1.8 Metodi e tecniche

I software e tool che ci aiuteranno a sviluppare il codice dell'applicazione saranno:

- Java come linguaggio di programmazione: un linguaggio ad alto livello orientato agli oggetti e a tipizzazione statica.
- Swing per l'UI del software: framework per java orientato allo sviluppo di interfacce grafiche
- AWT per l'UI del software: libreria Java contenente le classi e le interfacce fondamentali per il render grafico
- JTest per il test del software: prodotto di analisi statica e tet software Java automatizzato
- Github Repository: servizio di hosting per progetti software

- StarUML per il design del software: strumento per la modellazione di sistemi che utilizza il linguaggio di modellazione unificato.
- Eclipse: ambiente di sviluppo multi-linguaggio e multipiattaforma
- Windows, Mac OS, Distro Linux per l'esecuzione del software: sistemi operativi

1.9 Controllo qualità

Il software si basa essenzialmente su tre qualità fondamentali: portabilità, efficienza e fruibilità. Queste verranno analizzate passo dopo passo, ogni qual volta che il design, i requisiti o l'implementazione verranno modificati. L'obiettivo finale è quello di implementare un software in grado di essere utilizzato su diverse piattaforme, da diversi utenti, al meglio delle prestazioni. E stato deciso di improntare il lavoro su queste tre caratteristiche per il semplice fatto che il cliente deve essere in grado di utilizzarlo in maniera adeguata ogni volta che è disponibile.

1.10 Work Package

In viste della natura agile del progetto, questi work-package definiti a priori, saranno estesi e modificati o evoluti nelle diverse iterazioni del life-cycle:

-comprensione della fisica di gioco: approfondimento della fisica di gioco, design, testing ed implementazione di essa

- UI: approfondimento di Java Swing e Java AWT, design della UI

-integrazione: testing ed implementazione della UI

Anche se il risultato non è totalmente completo, si riconsidera comunque lo sviluppo nella sua interezza, ciò permette di potersi adattare in tempi brevi a cambiamenti di richieste o a nuove proposte, senza occuparsi troppo a lungo su parti che potrebbero dover essere radicalmente cambiate in futuro.

1.11 Risorse

Per la realizzazione del progetto sarà messo a disposizione un computer e diversi software reperiti gratuitamente online.

1.12 Budget e schedulazione

Il budget a disposizione E nullo, però le varie risorse saranno valorizzate, le quali saranno divise in base alle esigenze che si incontreranno durante lo sviluppo. La schedulazione non sarà lineare: siccome per questo progetto è un process model di tipo agile non si avranno mai dei requisiti troppo specifici sin dall'inizio, ma questi saranno modificati con il progredire del progetto.

Di seguito una schematizzazione approssimativa delle ore spese per la creazione di tale progetto, con un totale di circa 39 ore (senza contare le ore di studio per l'utilizzo delle varie tecnologie non conosciute):

1.13 Cambiamenti

Durante lo sviluppo del software, ma anche in seguito, potrà essere richiesta la modifica del codice sorgente o della struttura generale del progetto. Tali modifiche saranno correttive nel caso di errori nell'esecuzione o adattive a nuove richieste nate durante il progetto, come l'aggiunta di nuove funzionalità secondarie. Anche se il cliente sono io, il codice sarà ben organizzato in diverse classi , per favorire l'aggiunta di nuove caratteristiche, ad esempio nuovi pulsanti o, non comporteranno problemi. L'organizzazione del codice è infatti stata pensata in maniera preventiva di eventuali cambiamenti, dove ogni classe rappresenta una delle tante schermate che si presentano al cliente. Nel caso si voglia apportare una modifica basterà lavorare su quella specifica parte senza preoccuparsi troppo del resto del codice.

Qualora si verificasse un cambiamento, essi verranno approvati settimanalmente e riportati nel repository GitHub del progetto.

1.14 Consegna del progetto

La consegna prevista è per il 18/07/2022