Piano di progetto

1. Introduzione

Il progetto nasce per dimostrare di sapere applicare ad un caso di studio abbastanza corposo, i concetti, le metodologie e le tecniche studiate durante il corso di ingegneria del software.
Il nostro gruppo è composto da Gabriele Rota, Alessandro Lorenzi e Alessandro Rocco.
Per il progetto abbiamo deciso di ricreare il gioco “Monopoly” in maniera digitale per velocizzare e semplificare i passaggi meccanici e quindi ridurre i tempi morti di gioco, come la conta del denaro durante i pagamenti.
Il Monopoly è un gioco da tavolo nato agli inizi del 900’ in cui i giocatori cercano di guadagnare il massimo denaro e causare la bancarotta agli altri partecipanti raggiungendo il monopolio di mercato.
I partecipanti si muovono attraverso una tabellone rappresentante una città, acquistano terreni, costruiscono case e hotel, fanno pagare gli altri giocatori quando transitano sulla loro proprietà e subiscono imprevisti o probabilità, che possono essere positive o negative.

2. Modello di processo

Per sviluppare il nostro progetto software abbiamo scelto RUP (Rational Unified Process), adattato usando anche pratiche di RAD (Rapid Application Development). Abbiamo preso questa decisione dato che il requisito principale ed immutabile è ricreare il gioco monopoli, ma ci possono essere dei requisiti secondari che potremmo implementare in base al tempo rimanente rispetto alla scadenza.
La fase iniziale si concentra sul fissare obiettivi chiari: scopo del progetto, i suoi confini e i criteri di accettazione che verranno utilizzati dai clienti quando il sistema sarà consegnato. Qui vengono anche stimati i costi, le tempistiche e i rischi.
Nella fase di elaborazione si analizza il dominio del problema e si mira all’ottenimento di una solida base per eventuali implementazioni aggiuntive future. Al termine di questa fase verrà identificata la maggior parte dei casi d'uso.
La fase di costruzione è un processo di fabbricazione, costruzione. L'enfasi è sullo sviluppo di prodotti implementabili. I componenti completi sono sviluppati e accuratamente testati. I manuali dell'utente sono scritti. Al termine di questa fase è pronta la prima versione operativa del sistema, la versione beta.
Nella fase di transizione, il sistema completato viene rilasciato agli utenti e sottoposto a beta test da persone formate.

Gli step principali che determinano questo progetto sono la creazione del gioco mono-dispositivo con un’interfaccia base e successivamente l’integrazione delle funzionalità online per il supporto di più dispositivi.

Must have	**Should have**	**Could have**	**Won't have**
Rappresentazione accurata del gioco da tavolo Monopoli	Interfaccia fedele	Giocatori automatici (bot)	La versione mobile per telefono
Salvare e ripristinare una partita	Modalità multi dispositivo	Salvataggio automatico	Le modalità alternative
Interfaccia Base			Statistiche di gioco

3. Organizzazione del progetto

Abbiamo deciso di adottare un'organizzazione di tipo SWAT. Il nostro team infatti è piccolo, quindi i canali di comunicazione sono molto brevi e non ha la necessità di fare lunghe riunioni, ma brainstorming giornalieri seguiti da un processo decisionale di gruppo basato su negoziazione e consenso.
Ogni membro approfondisce un aspetto del progetto (es. interfaccia grafica, coding, documentazioni) per il quale assume la figura di leader comunque restando collaborare attivo del resto; ciò facilita il coordinamento che avviene per adeguamento reciproco. Le persone coinvolte nella progettazione del software devono sapere svolgere ogni compito (tester, programmatore), non deve esserci specializzazione.
Le conoscenze che ci mancano riguardano la connessione dei vari dispositivi dei giocatori, che dovranno essere connessi tutti alla stessa rete per poter accedere alla partita. Anche l'implementazione di un'interfaccia utente (schermata di gioco) intuitiva rappresenta una nuova conoscenza da approfondire.

4. Standard, linee guida, procedure

Seguiremo le convenzioni di Java. Il codice sarà gestito tramite GitHub con commenti precisi, senza ambiguità, e sfruttando le sue funzionalità per i rispettivi problemi. La documentazione verrà aggiornata periodicamente in base al cambiamento riscontrato durante lo sviluppo del progetto, indicando le versioni su GitHub e sul nostro drive informale. Per la modellazione useremo UML secondo quanto spiegato a lezione.

5. Attività di gestione

I requisiti verranno scomposti in sotto-obiettivi per la stesura del codice, con scadenze temporali brevi e possibilità di ribilanciamento in base al riscontro giornaliero che avverrà ogni due giorni.
Qui discuteremo i risultati del lavoro svolto, con confronto per la risoluzione di eventuali problemi e difficoltà. Questa coordinazione tra membri ci permetterà di rispettare la scadenza finale.
La coordinazione seguirà una modalità informale mediante una cartella drive contenente to-do list con scadenze temporali, Kanban, ecc... guidate dai nostri impegni e regolate in base all'adattamento reciproco.

6. Rischi

Nessuno di noi ha mai sviluppato un'interfaccia grafica e quindi il rischio è quello di investirci più tempo del previsto rispetto all'importanza che ha all'interno del contesto di ingegneria del software.
Un altro pericolo è rappresentato dall'implementazione della modalità multigiocatore su più dispositivi con l'eventualità che non venga implementata o che la nostra architettura non sia compatibile con essa, non avendo esperienza con le connessioni su Java.
Implementazioni possibili:

- Base: multigiocatore su un unico dispositivo (a turni).
- Intermedio: multigiocatore multi dispositivo (su rete locale).
- Avanzato: multigiocatore multi dispositivo (su rete remote).

7. Personale

I membri del team per la creazione del progetto sono:

- Rota Gabriele [1079894] leader interfaccia;
- Lorenzi Alessandro [1075244] leader documentazione;
- Rocco Alessandro [1081179] leader coding;

Saranno poi interpellati utenti esterni per testare il gioco e raccogliere informazioni per migliorare l'esperienza d'uso.

8. Metodi e tecniche

Nel nostro primo incontro svolgeremo una partita di monopoli per ricordarci le regole, prendere nota di alcuni passaggi, conoscerci meglio e divertirci, ciò influenzerà positivamente la motivazione e il morale per il progetto.
Svilupperemo poi un diagramma dei casi d'uso che verrà usato per aiutarci nell'ingegneria dei requisiti.
L'implementazione seguirà il class diagram, che ci aiuterà a identificare gli attributi e i metodi dalla scomposizione dei requisiti.
Per procedere in modo sicuro nello sviluppo del codice verranno eseguiti dei test per ogni aggiunta e/o modifica fatta. Sarà così possibile evidenziare le criticità man mano che si procede e ciò renderà anche possibile la scrittura di un codice coerente e ordinato.

9. Garanzie di qualità

Per il processo di sviluppo del nostro software seguiremo la norma ISO 9001, standard generico applicabile su qualsiasi prodotto; inoltre cercheremo di garantire che il nostro prodotto sia usabile, corretto ed affidabile.

10. Pacchetti di lavoro

Seguiremo le principali operazioni di ingegneria del software discusse durante il corso.
L'implementazione viene suddivisa in diverse macro aree, nelle quali ogni membro del team partecipa attivamente allo sviluppo.

1. Backend
2. Database partite

3. Frontend
4. Gestione multidispositivo

Non sono fasi distinte ma strettamente collegate.

11. Risorse

La nostra applicazione per funzionare a livello utente necessita soltanto di un computer con installato java e nel caso di modalità online di una connessione a internet.
Per lo sviluppo invece utilizzeremo vari Programmi, Tool e Linguaggi:
Eclipse: per la stesura del codice scritto in linguaggio Java;
SQLite: per la gestione del database;
StarUML: per la modellazione;
GitHub: per il controllo della gestione della configurazione;
Discord: per le riunioni;
PowerPoint: presentazione progetto;
Microsoft Word: per il project plan e la documentazione;
Whatsapp: gruppo per organizzare i meeting e risolvere eventuali dubbi;
Questa lista verrà quasi sicuramente aggiornata.

12. Budget

Il nostro budget non è di tipo monetario, ma riguarda la quantità di tempo utilizzato, quindi ci siamo imposti un minimo di 30 ore a testa. Data la nostra mancanza di esperienza con l'ingegneria del software, di sviluppo software collaborativo e dovendo formarci per particolari implementazioni sicuramente supereremo questo minimo. Ognuno di noi si sforzerà di tenere traccia del tempo dedicato a ciascuna attività in modo che a fine progetto potremo ricostruire le tempistiche.

13. Cambiamenti

Le modifiche prima di essere apportate dovranno essere discusse durante una sessione di brainstorming, e se dovessero essere approvate, verranno inserite all'interno delle sezioni issue, pull requests e branch di GitHub.

14. Consegna

Per la consegna il progetto verrà condiviso in un repository GitHub, che ci permette di caricare la documentazione e il codice relativo all'applicazione sviluppata.
Indicativamente la consegna è fissata per il 08/11/2024 con possibilità di manutenzione fino alla data dell'orale da concordare con il professore.