

a.a. 2023-2024



**UNIVERSITÀ
DEGLI STUDI
DI BERGAMO**

Project plan “Monopoly”

Ingegneria del software

Rota Gabriele [1079894]

Lorenzi Alessandro [1075244]

Rocco Alessandro [1081179]

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 un 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 è stato scelto il metodo RAD; questo procedimento di tipo Agile prevede di stabilire una serie di requisiti che successivamente saranno soddisfatti in un certo lasso di tempo (Time Box). Se non si dovessero rispettare le scadenze temporali, alcune funzionalità dovranno essere sacrificate. Per stabilire le priorità dei requisiti utilizziamo il paradigma MoSCoW.

Le 4 fasi del RAD sono:

1. Requirements planning
2. User design
3. Construction
4. Cutover

Gli step principali che determinano questo progetto sono la creazione del gioco monodispositivo con un’interfaccia già definita e successivamente integrare le funzionalità online per il supporto di più dispositivi.

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, database, 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

Utilizzeremo Eclipse per la stesura del codice sorgente, che sarà scritto in Java con l'aiuto del tool Rebel, StarUML per la creazione dei diagrammi (stato, classi, sequenza, casi d'uso, attività), il linguaggio SQL per la gestione dei dati nel database.

Architettura:

1. Server: Gestisce l'andamento della partita, non possiede un'interfaccia grafica ed è compatibile con i più diffusi sistemi operativi. Ospita il database locale di una specifica partita.
2. Client: Gestisce l'interfaccia grafica e l'interazione con l'utente. Connesso al server per comunicare con il database.

5. Attività di gestione

I requisiti del RAD 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.

6. Rischi

Il rischio maggiore è rappresentato dall'implementazione della modalità multiplayer al variare del numero di reti, con il rischio di non essere in grado di raggiungere gli obiettivi prefissati entro la deadline. Implementazioni:

- 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]
- Lorenzi Alessandro [1075244]
- Rocco Alessandro [1081179]

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

8. Metodi e tecniche

La modellazione delle classi verrà implementata grazie al class diagram, che ci aiuterà a identificare gli attributi e metodi da implementare nelle classi.

Il diagramma dei casi d'uso sarà usato per l'ingegneria dei requisiti.

Per procedere in modo sicuro nell'implementazione del codice verranno eseguiti dei test per ogni aggiunta e/o modifica fatta. Sarà così possibile evidenziare 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

Il progetto viene suddiviso in diverse macro aree, nelle quali ogni membro del team partecipa attivamente allo sviluppo:

1. Backend
2. Frontend
3. Database partite
4. Gestione multidispositivo

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.

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 sullo sviluppo del progetto e sulla creazione della documentazione.

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" 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 16/02/2024 con possibilità di manutenzione fino alla data dell'orale da concordare con il professore.