Monopoly

Davide Rossi, Alessio Minniti, Lucas Prati, Martina Ravaioli

5 aprile 2025

Indice

Capitolo 1

Analisi

1.1 Descrizione e requisiti

Il software mira alla costruzione di una versione virtuale del famoso gioco da tavolo "Monopoly". Quest'ultimo è un gioco di società nel quale più giocatori muovono a turno le proprie pedine su un tabellone composto da caselle; la maggior parte di queste sono proprietà: ovvero terreni acquistabili dai giocatori. Lo scopo del gioco è di creare un monopolio, aumentando i beni o il denaro posseduto. Durante la partita i giocatori possono acquistare le proprietà capitandovi sopra e pagando una somma alla banca. Una volta acquistate il giocatore potrà anche decidere se eventualmente migliorare le proprietà. I giocatori si scambiano denaro a vicenda sotto forma di pagamenti. Quando un giocatore capita su una proprietà di un altro giocatore è costretto a pagare a questo una somma di denaro. L'obiettivo è mandare gli altri giocatori in bancarotta, vince l'ultimo giocatore rimasto. Il software permetterà di creare e giocare un partita con un determinato numero di giocatori dallo stesso terminale.

Requisiti funzionali

- Il gioco potrà essere giocato su uno stesso dispositivo da un minimo di 2 giocatori ed un massimo di 6.
- Il giocatore potrà tirare i dadi per far muovere la propria pedina sul tabellone.
- Durante il proprio turno ciascun giocatore avrà la possibilità di acquistare la proprietà sulla quale si trova la sua pedina, se la proprietà non appartiene a nessuno, altrimenti potrà pagare l'affitto al proprietario.

- Sul tabellone saranno presenti delle caselle speciali con effetti specifici sulla partita
- Nel corso della partita un giocatore, quando capita sulle sue proprietà, le potrà migliorare spendendo del denaro per posizionare delle case che elevano il costo dell'affitto.
- Durante il proprio turno, se lo desidera, il giocatore potrà vendere alla banca le proprietà o eventuali case presenti su esse.

Requisiti non funzionali

- Monopoly dovrà permettere agli utenti di aggiungere e modificare gli effetti speciali del gioco in facilità e senza toccare il codice
- NON LO INSERIREI salvare su file lo storico dei risultati delle partite precedenti
- Monopoly potrà essere giocato attraverso un'interfaccia grafica che dovrà essere in grado di adattarsi alle dimensioni di vari schermi
- Monopoly dovrà permettere agli utenti di alterare in facilità la configurazione del gioco (impostazioni, struttura del tabellone e caselle che lo compongono)
- NON LO INSERIREI rendere l'applicazione portabile su più SO

1.2 Modello del dominio

Il sistema (TurnationManager) deve gestire l'avvicendarsi dei vari turni dei giocatori. A ogni giocatore (Player) viene concesso periodicamente il proprio turno d'azione sulla base di una rotazione ciclica. Durante il turno il giocatore tira i dadi e muove sul tabellone (Board) la propria pedina (Pawn) finendo sempre su una casella (Tile). Questa casella può essere una proprietà (Property) oppure una casella speciale (Special), e sulla base di ciò cambia radicalmente l'interazione dell'utente. Se la pedina capita su una proprietà che non è ancora posseduta da alcun giocatore allora il giocatore può chiedere alla banca (Bank) di acquistarla pagando dei soldi. Se invece la proprietà era già stata comprata in precedenza da un altro giocatore allora si deve pagare l'affitto al proprietario. Se la pedina del giocatore capita su una sua proprietà può decidere se migliorarla aggiungendo case e pagando dei soldi alla banca. Oppure se desidera può vendere la proprietà, in tal caso riceverà dei soldi

dalla banca e la proprietà ritornerà disponibile all'acquisto. Se capita su una casella speciale invece si attiverà un effetto caratteristico della casella (Effect) che avrà ripercussioni sullo svolgimento del gioco (guadagno/perdita denaro, saltare un determinato numero di turni, affrontare sfide). Una volta che il giocatore avrà svolto tutte le azioni obbligatorie potrà decidere di terminare il proprio turno. Fatto questo se il suo saldo è in negativo perde la partita e tutte le sue proprietà tornano disponibili per l'acquisto.

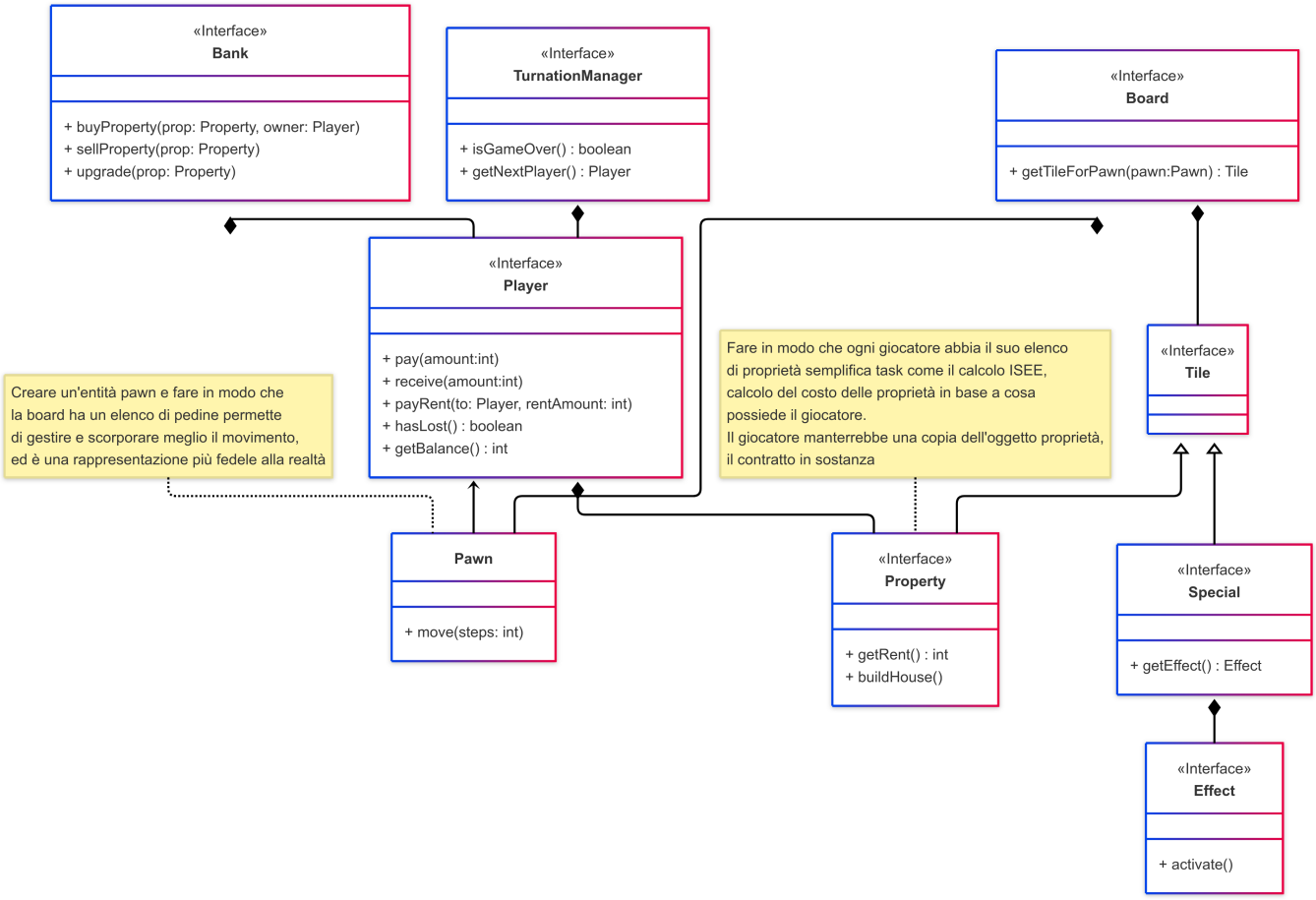

Figura 1.1: Schema UML dell'analisi del problema, con rappresentate le entità principali ed i rapporti fra loro

Capitolo 2

Design

2.1 Architettura

L'architettura di MPoly è basata sul pattern architetturale MVC. Si è scelto di implementare l'architettura nella sua forma classica in quanto il gioco ha una modalità di interazione prettamente statica: l'utente interagisce con la view, facendo scaturire un evento; a seguito di questo evento si eseguono una serie di operazioni specifiche per la sua gestione con conseguente aggiornamento della view, che mostra all'utente che cosa è successo. È stato predisposto un controller principale denominato GameManager. Questo implementa il pattern observer e si mette in ascolto degli eventi che vengono catturati dalla view. Nel momento in cui viene notificato di un evento il controller interroga il model. La componente model dell'applicazione ha dei determinati punti d'entrata. Nello specifico abbiamo Board, che si occupa di gestire la struttura e lo stato del tabellone, TurnationManager, che si occupa di gestire l'avanzamento della partita, l'avvicendarsi dei turni dei giocatori e il tiro dei dadi, e Bank, per gestire lo scambio di denaro e la compravendita delle proprietà. Questi punti d' entrata del model offrono delle primitive che incapsulano la logica di funzionamento delle principali azioni che si possono compiere durante il gioco. Queste azioni sono caratteristiche del gioco stesso Monopoly. Con questa architettura il model è perfettamente scorporabile e utilizzabile per costruire un software diverso che risponda allo stesso dominio. Una volta finita l'interrogazione del model il GameManager si occuperà di chiamare delle funzioni sulla view che aggiornano il contenuto di quest'ultima. Il controller agisce su model e view mediante delle interfacce che sono completamente indipendenti dall'implementazione di quest'ultimi. Questo, in particolare, fa sì che la modalità di implementazione della view non determini cambiamenti sul controller o sul model in alcun caso. Il software

prevede anche un menù iniziale di configurazione della partita. Questo menù è a sua volta costituito da una sua architettura MVC più ridotta, modellata tenendo in mente gli stessi principi descritti sovrastante. L'entità GameBuilder è colei che si occupa di instanziare poi le classi dell'MVC principale del software e avviare effettivamente il gioco, costruendo tutti gli oggetti.

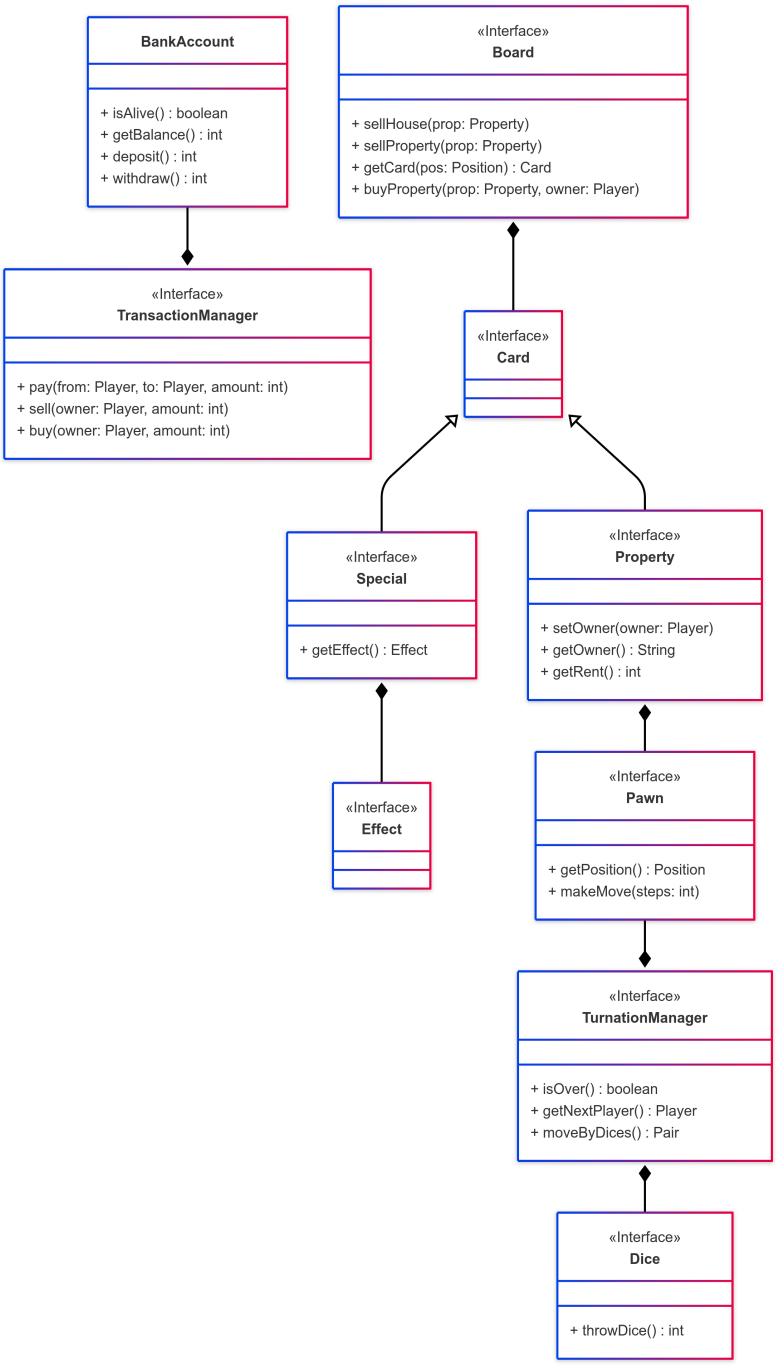

Figura 2.1: SCHEMA SBAGLIATO