

Peer-Review 1: UML

Allanda Marco, Ammaturo Pietro, Bocci Giuseppe, Rossi Luca

Gruppo AM15

Valutazione del diagramma UML delle classi del gruppo AM24.

Lati positivi

- Classi autoesplicative: poche e fondamentali
- UML view e rete già fatti
- Usa una function per i goal attuando una specie di strategy pattern molto flessibile
- Check dei common goal chiaro (Game->CommonGoal)
- Model racchiude interamente la logica di gioco avendo come unica classe di interfaccia Game
- Uso eccezioni per gestire casi di errore

Lati negativi

- Le relazioni non sono per niente chiare: sono tutte sovrapposte e in molte non si capisce dove vanno
- Gli oggetti passati ai metodi sono anonimi, questo rende difficile la comprensione del metodo stesso a chi non ha scritto l'uml, ma deve implementare i metodi
- fasi del gioco/turno non separate. Come faccio a capire che le sto eseguendo nel giusto ordine?
- Le tuple aggiungono una complicazione non necessaria (era sufficiente una map)
- Nel caso in cui non si usi un setup file(non si capisce): bassa estensibilità
- Non si può facilmente modificare il funzionamento dei punteggi o aggiungere carte
- Non è chiara la gestione del personal goal: non si capisce come personal goal tiene traccia dei punti e dove sono salvati
- I personal goal e i common goal da dove sono caricati, da un file esterno o sono scritti da qualche parte nel codice?
- InvalidMoveException molto generale: ci sono più sottocasi(almeno due) per cui la mossa è invalida
- Utilizzo di attributi pubblici nel Player (erano meglio getter e setter)
- C'è il metodo updateChat() nella view ma nessuna classe che la definisce
- Manca il controller: è sbagliato connettere la parte di rete direttamente al Game. Anche se il Game è un'interfaccia tra Model e Controller, si dovrebbero creare delle classi nel controller(da affiancare alla Lobby) per gestire la fase di playing e la fase di end del gioco. Quest'ultimo più che un lato negativo è un errore.

Confronto tra le architetture

Le nostre architetture sono molto simili nella separazione dei moduli (separazione model e controller tramite Game) e nel check dei goal tramite funzione (noi carichiamo i pattern da un file json). Un vantaggio di questa architettura è la snellezza e la sobrietà nella gestione dell'informazione al contrario nostro. Infatti, cosa che sicuramente potremmo migliorare nella nostra architettura è la ridondanza dell'informazione (es. noi usiamo due classi per le tile: Tile e TileType, seppur utile nella pratica per "trasportare" informazioni, potremmo ridurre ad un unico tipo, aumentando la complessità nei metodi). In generale ci sono più nostre classi che potrebbero essere collassate in una, ciò semplificherebbe sia la comprensione dell'uml e l'implementazione (l'ultima solo in alcuni casi), ma abbiamo progettato questo design per renderlo il più elastico possibile a cambiamenti futuro, ma forse abbiamo esagerato(?).

Peer Review 2: Sequence Diagram

Allanda Marco, Ammaturo Pietro, Bocci Giuseppe, Rossi Luca

Gruppo AM15

Valutazione del diagramma UML delle classi del gruppo AM24.

Lati positivi

- Poche tipologie di messaggio, sintetiche e comprensibili
- Flow lineare delle transazioni
- Inizializzazione del gioco fatta bene

Lati negativi

- I messaggi non vengono descritti sufficientemente (cosa sono concretamente i payload?)
- Manca una specifica dei messaggi "move" (scegliere le tiles, inserire le tiles etc.) e risulta quindi difficile capire il flusso di gioco e come si sviluppa il protocollo, ogni cosa è "Move".
- Non ci sono notifiche per i cambiamenti di stato: cambio turno, cambiano i token sulle carte...
- La chat è confusa: si può inviare un messaggio anche a un solo giocatore? Tutti i client aggiornano la view per ogni messaggio anche se non per loro? Cos'è "response: sender" ?
- Che succede se un giocatore prova ad entrare durante il FirstLogin di un altro?

Confronto tra le architetture

- Le nostre architetture sono molto simili nella parte di login (togliendo la parte di persistenza da noi implementata), mentre no per il resto del game.
- Noi abbiamo gestito molti più casi particolari: che succede se 2 giocatori fanno login contemporaneamente? entrambi first player?
- Noi abbiamo utilizzato messaggi molto specifici e dettagliati, che può incrementare la complessità del protocollo ma fornisce più informazioni, probabilmente il giusto sta in una via di mezzo tra il loro "Move" per ogni cosa e i nostri messaggi ad-hoc.