Peer-Review 2: UML

Vaccarino, Vacis, Viganò, Zacchi

Gruppo GC05

Valutazione del diagramma UML delle classi del gruppo GC52.

Lati positivi

Remote Interfaces

- Corretto disaccoppiamento tra VirtualMainController e VirtualController, per definire metodi per creazione/gestione partita, e metodi per poter effettivamente giocare

RMI

- Rendere RMI asincrono (come Socket), permettendo l'unificazione dei due protocolli di rete e fornendo una migliore user experience
- Far comunicare il client direttamente con la sua partita, dandogli il riferimento ad essa quando si connette
- Facendo riferimento al precedente punto, un'altra buona scelta è quella di non fare implementare VirtualController direttamente al GameController

Controller

- Avere il GameSelector agnostico del fatto che i client si siano connessi via socket/RMI

UI

- L'utilizzo della classe astratta UserInterface offre una solida base comune per la CLI e la GUI, consentendo un riuso del codice efficace, una maggiore coerenza nell'esperienza utente.

Message

- Buona rappresentazione e distinzione dei vari tipi di messaggi

In generale

- Uso corretto e sensato del pattern Observer-Observable, definendo bene chi osserva e chi è osservato.

- Per tutta la parte di rete c'è spesso stata l'idea di avere delle interfacce o classi astratte comuni per Socket e RMI, cercando di unificare il più possibile il loro comportamento.

Lati negativi

- Invece di fare gestire al controller le connessioni, utilizzare un ipotetico "RmiServer" come intermediario tra i client e il MainController, distinguendo meglio la parte di rete dal controller. Ciò porterebbe a una migliore gestione delle connessioni e scalabilità del sistema.

- Non è presente una particolare gestione per gli errori di rete

Confronto tra le architetture

Guardando il lavoro svolto del gruppo GC52, ci siamo resi conto di:

- Prendere in considerazione l'idea di far comunicare il client con un altro RemoteObject(nel caso del gruppo GC52 GameControllerWrapper) nel caso di metodi relativi alla effettiva partita, evitando in quel caso di avere come intermediario il MainController
- Dover fare una più attenta distinzione tra interfacce per giocare e interfacce per connettersi e gestione partite
- Poter sfruttare il design pattern Observer-Observable per definire meglio gli aggiornamenti
- Dover curare meglio la parte di definizione dei messaggi, come fatto dal gruppo revisionato