

Peer-Review 2: UML

Lorenzo Paracchini, Christian Piccoli, Lorenzo Pittelli, Nicola Siniscalchi

Gruppo 17

Valutazione del diagramma UML delle classi del gruppo 7

Lati positivi

- Già implementata sia la connessione tramite socket che tramite RMI
- Comunicazione asincrona gestita dal controller, per evitare comportamenti indesiderati da parte dei client al di fuori dei propri turni
- UML (e in generale suddivisioni delle classi e dei metodi) estremamente chiari per persone esterne

Lati negativi

- La suddivisione in così tante classi e metodi rende un po' difficile seguire il flusso, soprattutto a livello di UML
- La classe immutablePlayer (e annesse classi immutable) risultano ridondanti, avendo attributi praticamente identici. La scomodità maggiore è, secondo noi, che in caso di modifiche future alla classe Player, sarà necessario dover modificare anche immutablePlayer (e così per tutte le classi con equivalente immutable).

Confronto tra le architetture

- Già implementato sia RMI che socket (noi per ora solo socket, in futuro anche RMI)
- Utilizzati i listener a differenza nostra che abbiamo usato observer e observable
- Come nel nostro caso il client non ha una copia locale del gioco ma si affida al server con messaggi specifici di update