

Peer-Review 1: UML

Matteo Sala, Filippo Raimondi, Luca Trifaro, Mattia Federico Volpi

Gruppo GC03

Valutazione del diagramma UML delle classi del gruppo GC50.

Lati positivi

Buona idea quella di usare un'interfaccia ClientInterface implementata in ClientRMI e ClientHandler ed entrambe che estendono la classe Observer.

Buona idea quella di usare una classe Server per gestire la lobby.

Buona divisione tra le classi del client e del server, sia socket che RMI.

Lati negativi

Nell'UML mancano i collegamenti e le relazioni tra le varie classi, il che rende più difficile la lettura. Non risulta completamente chiara l'utilità di Match, ovvero non risulta chiaro perché non vengano fatte le chiamate direttamente al controller dalla view, e come mai abbia dei metodi di update, secondo il pattern MVC, dovrebbe essere il model che, quando viene modificato, notifica la view, non il controller o un meccanismo di mezzo.

Non è chiaro il controller in quanto sembra essere un'implementazione di un'altra classe ma non appare in nessuno dei documenti, in caso, se si volessero gestire partite multiple potrebbe mancare un "main controller" che gestisca più ad alto livello le varie partite.

Per quanto riguarda il socket, potrebbe essere utile usare uno strategy pattern per gestire i vari messaggi.

Confronto tra le architetture

La nostra implementazione è più volta all'utilizzo di RMI, rendendo l'implementazione del socket più complessa e "distaccata", potrebbe essere comodo invece, come è stato fatto da loro, rendere i client implementazioni di una stessa classe Client.

Potrebbe essere utile usare una classe Server come lobby.