

Peer-Review 2: Protocol design

Emma Zaroli, Marco Rocchi, Riccardo Dell'Oro
Gruppo GC10

7 maggio 2022

Valutazione del diagramma UML delle classi del gruppo GC20.

1 Lati positivi

- La gestione delle possibili richieste dell'utente in classi che estendono PlayerRequest rende facilmente scalabile il protocollo. Lo stesso per le risposte.

2 Lati negativi

- L'implementazione di messaggi tramite stringhe rende meno estendibile il protocollo (oltre che a complicare la gestione della GUI)
- Secondo il pattern MVC, quando arriva un messaggio dal client su SocketClientConnection, non deve essere quest'ultima a notificare il controller, ma dovrebbe passare per la Remote View (sembra così dal sequence diagram).
- Non ci sono metodi per notificare al client un eventuale errore
- La fine di una partita non dovrebbe essere gestita tramite eccezioni (EndGameException)

3 Confronto tra le architetture

La nostra scelta di implementare le funzionalità avanzate di multithreading e di resilienza alle disconnessioni, oltre che a gestire in modo differenziato le code di giocatori a seconda della modalità di gioco scelta, ha portato alla progettazione di un protocollo molto diverso da quello del gruppo da revisionare e quindi difficilmente comparabile.