

# Peer-Review 2: Network-Controller-Model

Lorenzo Veronese, Giada Silvestrini, William Stucchi

Gruppo GC49

Valutazione del diagramma UML delle classi del gruppo GC59.

## Lati positivi

Dall'analisi del modello presentato, sono stati riscontrati alcuni aspetti positivi, come l'utilizzo del pattern State per la gestione del controller, che dovrebbe permettere la gestione delle diverse fasi di gioco. Inoltre, abbiamo apprezzato l'implementazione di gerarchie logiche, attraverso l'uso di interfacce ed enumerazioni, così da favorire la leggibilità dei diagrammi e del codice.

In aggiunta riteniamo appropriata l'implementazione del protocollo di comunicazione attraverso oggetti messaggio (seppur la scelta del nome piuttosto opinabile).

## Lati negativi

Il problema che è stato principalmente riscontrato nell'analisi degli schemi UML ricevuti è la poca chiarezza e completezza degli stessi, nonché la loro scarsa interconnessione: infatti, è stato particolarmente complesso comprendere come fosse possibile compiere le azioni del gioco solo attraverso i metodi a disposizione (può essere che ciò sia dovuto ad una mancanza di implementazione della parte interessata), sia nel model che nel controller, dato che nel model ricevuto in precedenza non c'erano metodi di alto livello che permettessero il compimento di una mossa in maniera atomica (ci saremmo dunque aspettati di trovarli nel controller). Ugualmente non è chiaro il collegamento tra controller e connessione: avendo a disposizione solo i metodi compresi negli schemi ricevuti, come si scambiano i messaggi? Come sono strutturati il server e le view? Probabilmente lo schema è parziale, ma avendo solo questo a disposizione non sono chiare le interazioni tra i componenti, e neanche il sequence diagram è servito a chiarire a sufficienza la struttura del protocollo di comunicazione.

La scelta dei nomi dei messaggi di comunicazione non risulta sufficientemente esplicativa della direzione della comunicazione, inquanto messaggi "response" sia in invio che in ricezione non chiarificano il ruolo delle parti (chi risponde a chi? Dove sono le richieste di risposta?).

## Confronto tra le architetture

Analogamente ai nostri colleghi abbiamo sviluppato, seppur in maniera diversa, il pattern State per la gestione del controller; inoltre, anche nel nostro protocollo i messaggi sono singoli oggetti con vari attributi a seconda della loro funzione.

Purtroppo, la parte di network e comunicazione non è confrontabile, perché incompleta.