Peer Review n.2

Francesco Scroccarello, Nicola Zarbo, Luca Tombesi (GC62)

1 Lati negativi

L'UML si presenta molto incompleto e scarno di metodi e attributi, oltre che della parte di model e di controller, e ciò rende la comprensione del lavoro svolto davvero difficile. Inoltre manca la descrizione del protocollo di comunicazione, e non è possibile comprendere a pieno come avvenga lo scambio di messaggi tra client e server. Una delle cose che si evince subito è che sono impiegati due sistemi di comunicazione tra client e server diversi, rispettivamente Socket e RMI nonostante il requisito di progetto fosse di usare il Socket. Remote view ha un'istanza di controller, e il controller ha al suo interno un'istanza di model invece di usare il pattern observer, il che dimostra come non sia possibile seguire, secondo questa implementazione, il pattern MVC. In generale ci sono tanti doppi assegnamenti tra oggetti.

1.1 Possibili refusi

Il server RMI ha una sola istanza di client al suo interno, viene da pensare che sia una svista di generazione del diagramma, o questo implicherebbe che per ogni client venga istanziato un nuovo server.

2 Lati positivi

Poiché l'UML è fortemente incompleto non possiamo fare una valutazione completa del lavoro svolto.

3 Differenze

Noi usiamo una sola libreria di comunicazione, Socket, e la comunicazione avviene tramite il passaggio di oggetti messaggio, in formato JSON (String), per ogni azione atomica. Inoltre gestiamo le connessioni tra client e server tramite delle Lobby le quali hanno poi una singola remote view.