Peer-Review 2: UML

Andrea Somaini, Agnese Taluzzi, Aaron Tognoli

Gruppo GC-04

Valutazione del diagramma UML della parte di rete del gruppo GC-14.

Lati positivi

All'arrivo del primo player vengono subito richieste le opzioni di gioco, così da poter accogliere un numero corretto di giocatori. Ottima la gestione asincrona dei messaggi, che segue una logica orientata agli eventi. Ottima la scelta di avere una classe Lobby, anche per l'eventuale futura implementazione di un server che può gestire più partite contemporaneamente (funzionalità aggiuntiva).

Lati negativi

Non viene controllato, lato server, la validità della modalità di gioco, del numero di giocatori, e validità sintattica dello username.

Confronto tra le architetture

Nel vostro caso la sollecitazione di un input parte dal server (per esempio modelRequest a cui corrisponde modelResponse) mentre nella nostra implementazione é sempre il client che invia dei messaggi al server, per poi ricevere un acknowledgement.
Quando viene giocata una carta character, nel caso in cui ci sia bisogno di un argomento, richiedete che venga inviato al server un secondo messaggio. Nel nostro caso, il messaggio di tipo "playCharacter" contiene già anche l'eventuale argomento, evitando così un ulteriore messaggio dedicato, stesso ragionamento anche per moveStudent: dalla descrizione sembra che inviate due messaggi quando ne basterebbe uno solo che contiene entrambi gli argomenti.