

Peer Review 2, Gruppo AM40

Di seguito i Sequence diagram per la comunicazione fra server e client. Abbiamo optato per riportare i JSON scambiati come fatto in un esempio delle slide viste a lezione con il Professor Conti.

Come richiesto dal Professor Margara, abbiamo implementato una logica reattiva (i comandi possono essere inviati in qualsiasi ordine e in qualsiasi momento, anche non nel proprio turno. La gestione dei comandi avviene completamente lato Server).

Per la conversione da stringa a messaggio sotto forma di JSON (contenente anche altre informazioni, fra cui il mittente del pacchetto) abbiamo creato la classe MessageAdapter.

Quando parliamo di elementi del Server ci riferiamo a vari oggetti di diverse classi, il più importante è ClientHandler, che:

- È in ascolto dei comandi dell'utente
- Invia le risposte ai comandi (stringhe per l'utente)
- Esegue i comandi digitati dall'utente (per la gestione degli errori guardare l'ultima pagina)

Nelle prossime pagine troverete la comunicazione fra lato server e lato client. In ciascuna immagine specifichiamo gli oggetti JSON passati durante la comunicazione. In caso di errore, abbiamo dei messaggi di default (l'ultima immagine del documento spiega questa dinamica).

DISCLAIMER: nel testo vedrete usate le parole "oggetto" e "informazione" come sinonimi. Questo per semplificare la lettura del testo, in quanto non tutti gli oggetti passati tramite JSON diventeranno oggetti java, ma alcuni di essi servono solo da contenitori di informazioni.

(Fine disclaimer, buona lettura!)

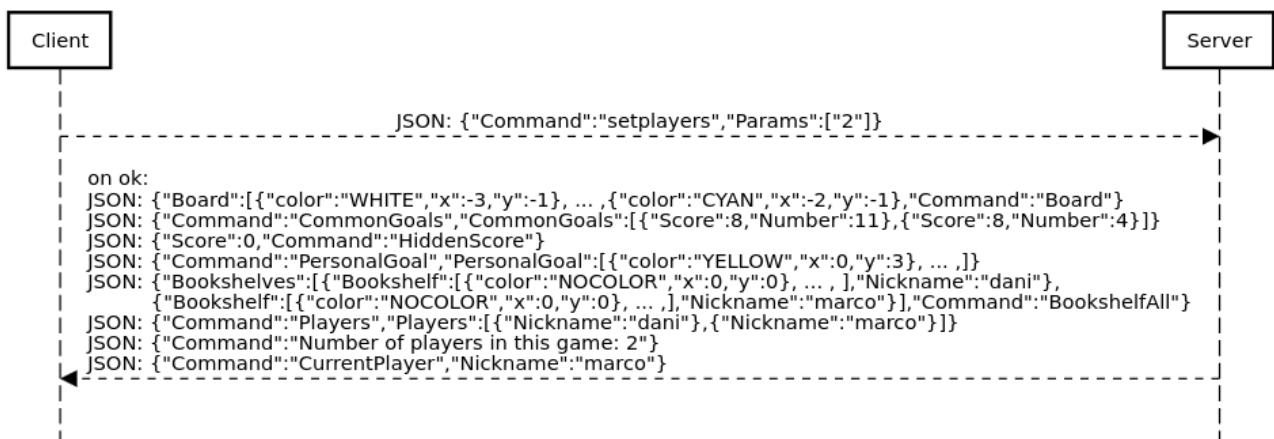
Comando Login: comando di inizio partita per partecipare al gioco, richiede l'inserimento del nickname del giocatore

Login



Comando SetPlayers: comando di inizio partita, nel momento in cui tutti i giocatori sono pronti, verranno tolti dalla lobby e verranno istanziati gli oggetti necessari all'inizio della partita. Al client vengono inviati gli oggetti e informazioni per inizializzare il Model (Board, Personal Goal, Common Goal, BookShelf, ...)

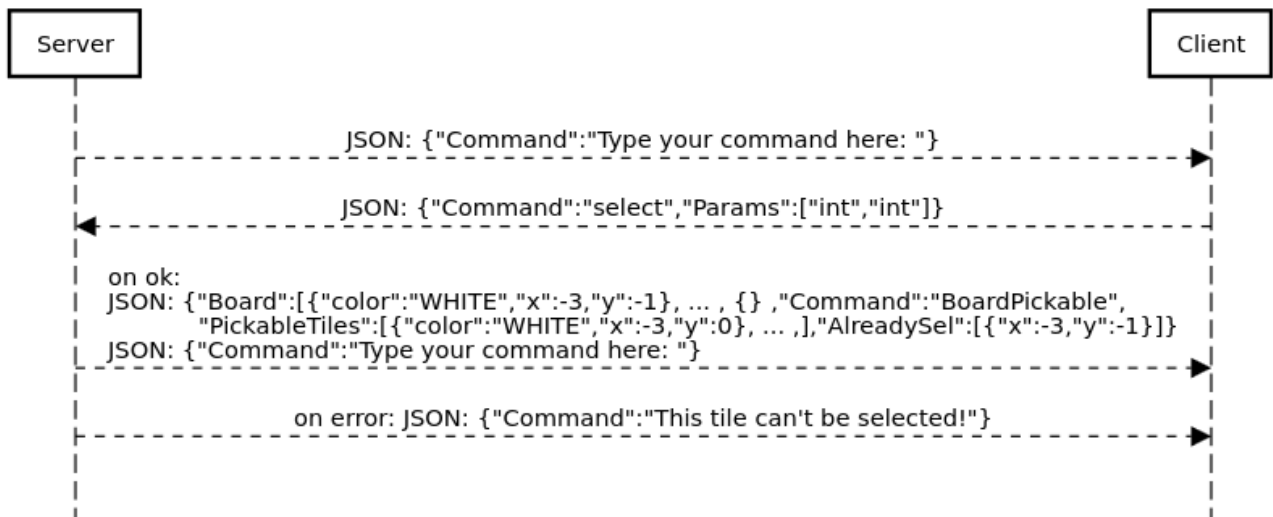
SetPlayers



Comando Select: comando per selezionare una tile dalla board. Da notare il fatto che al client vengono inviate informazioni sulle “pickable tiles”, ovvero le tiles che considerando quella selezione possono esser prese successivamente. Inoltre vengono inviate informazioni sulle tiles già

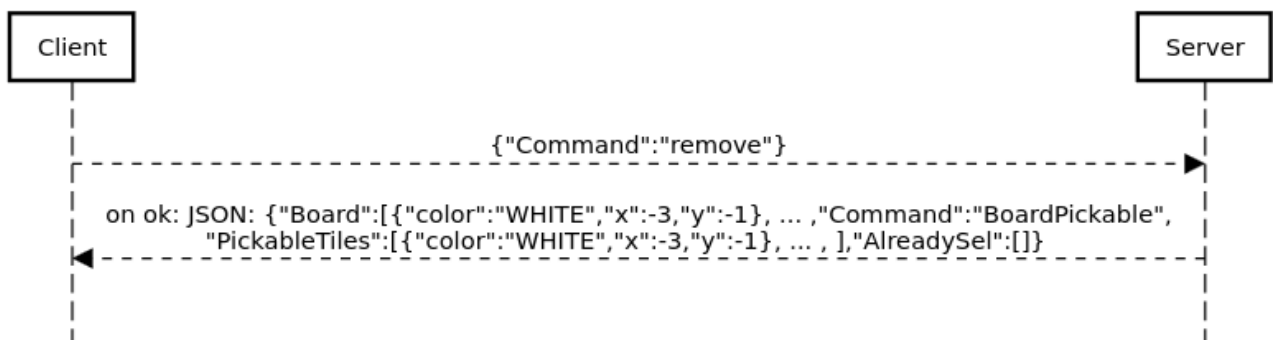
selezionate. Questo perché l'array di tiles selezionate viene aggiornato lato Server (mantenere una copia delle tiles selezionate lato server potrebbe portare vantaggi per la resilienza)

Select



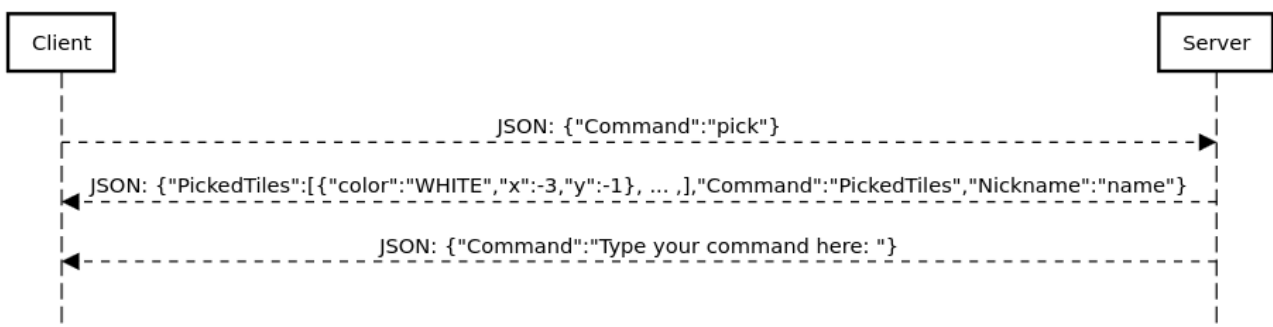
Comando Remove: deselecta le tiles selezionate in precedenza

Remove

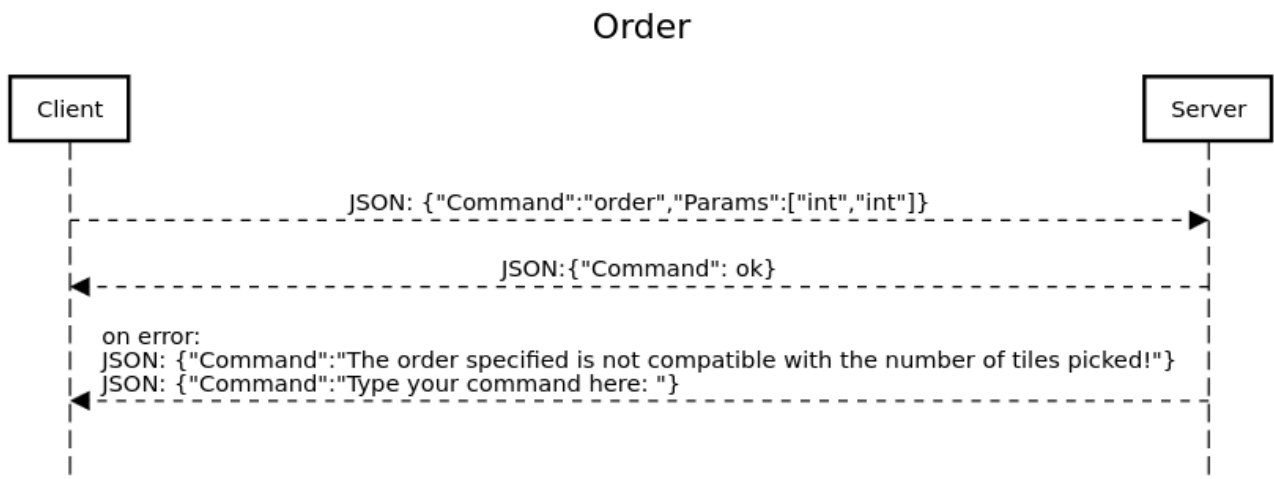


Comando Pick: comando di conferma della selezione, comporta modifiche nel Model (vengono tolte le tiles selezionate dalla Board)

Pick



Comando Order: comando usato per ordinare le tiles selezionate. Gestiamo anche il caso in cui il numero dei parametri (o il loro valore) non coincida con il numero di tiles selezionate.

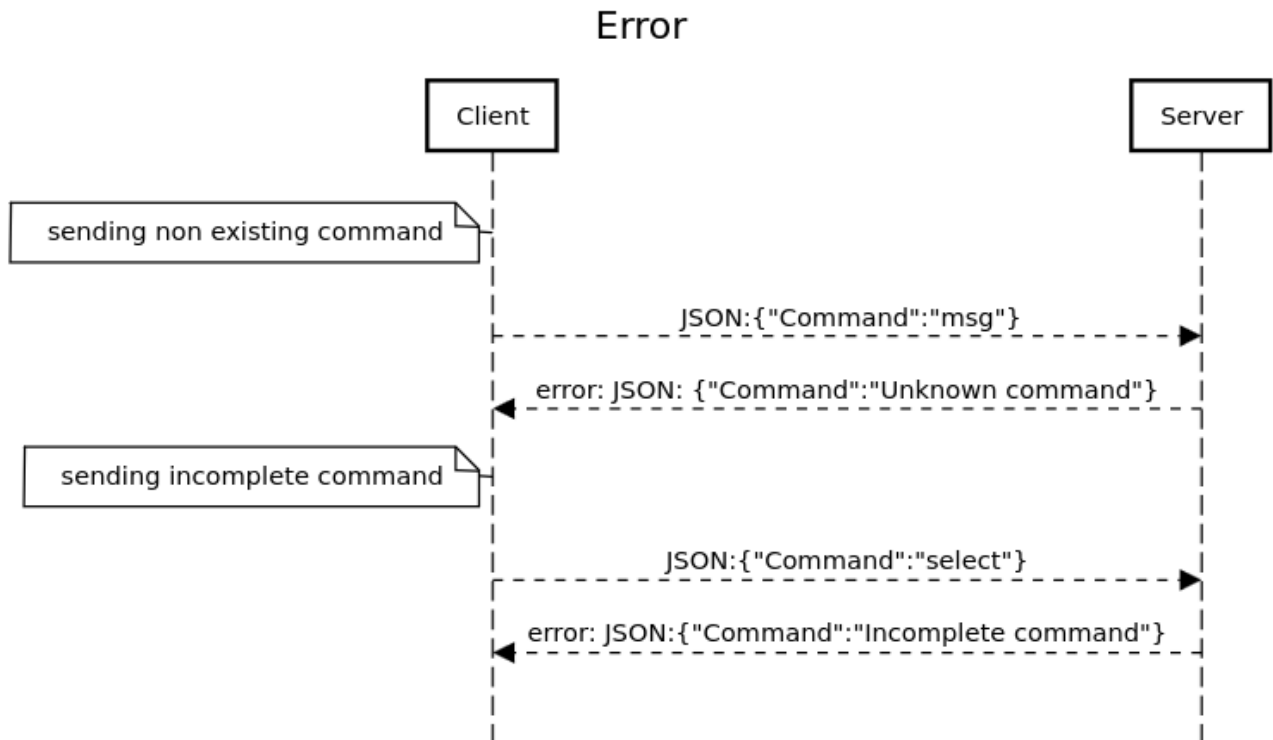


Comando Insert: comando usato per selezionare la colonna in cui inserire le tiles selezionate (che devono essere già ordinate). Avviene una modifica del model (aggiornamento della bookshelf del giocatore)



IMPORTANTE! ->

Tutte le fasi del gioco possono portare a questi tipi di errore, che gestiamo comunicando al client che il comando è sconosciuto o che è incompleto.



COMANDI FUORI TURNO

Nel caso in cui un player provi a effettuare un comando durante il turno di un altro giocatore si analizza il comando, possono accadere due eventi:

- Il comando era legittimo (es quit) e viene eseguito
- Il comando non era legittimo (es pick) non viene eseguito. Viene comunicato al player che non è il suo turno