

INTRODUZIONE

La comunicazione all'interno dell'implementazione può essere svolta tramite RMI o socket, a seconda dell'utente. Il server è predisposto per accettare entrambi i tipi di connessione utilizzando due thread separati.

In entrambi i casi, vengono passati tra client e server dei Messages. Tra client e server sono chiamati InputMessages, mentre tra server e client sono chiamati ResponseMessages. I messaggi consistono in degli oggetti serializzabili contenenti un metodo execute e un attributo, di classe SerializableInput per i messaggi di input e SerializableGame per quelli di response. SerializableInput comprende i valori inseriti come input, mentre SerializableGame comprende una copia totale o parziale del model, a seconda del tipo di messaggio.

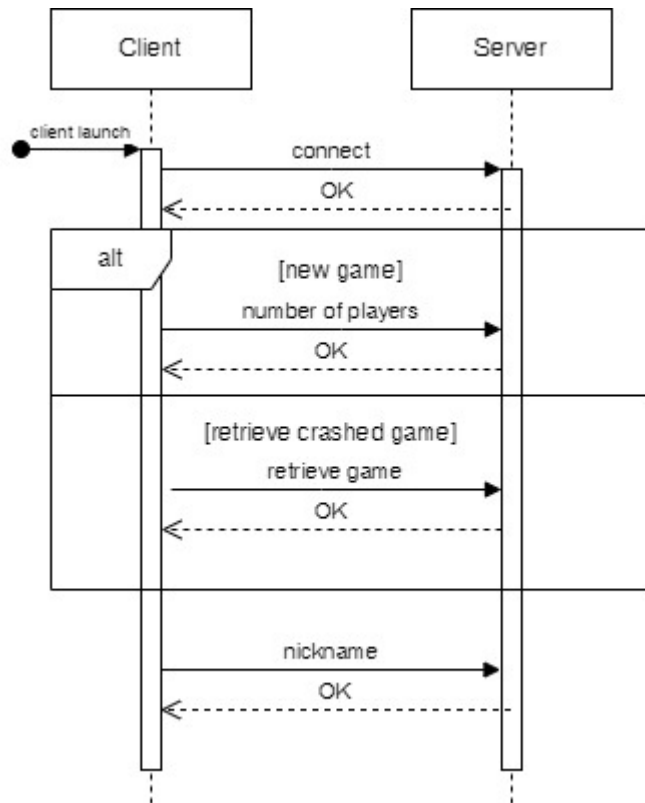
I Messages vengono passati ad un observer sull'entità che genera il messaggio, che procede a chiamare un update sull'entità che riceve il messaggio, dove infine verrà chiamato il metodo execute.

Con RMI viene attesa automaticamente la fine del metodo execute. Ciò non avviene con socket, perciò sono state aggiunte delle wait con la finalità di mimare il comportamento di RMI.

Le comunicazioni vengono gestite tramite un AlphanumericID di 100 cifre generato randomicamente dal software, composto da lettere minuscole, maiuscole e numeri dell'alfabeto latino. Non vengono effettuati controlli sull'univocità poiché i possibili ID sono nell'ordine di grandezza di 10^{179} , quindi nella pratica è impossibile che ne vengano generati due uguali.

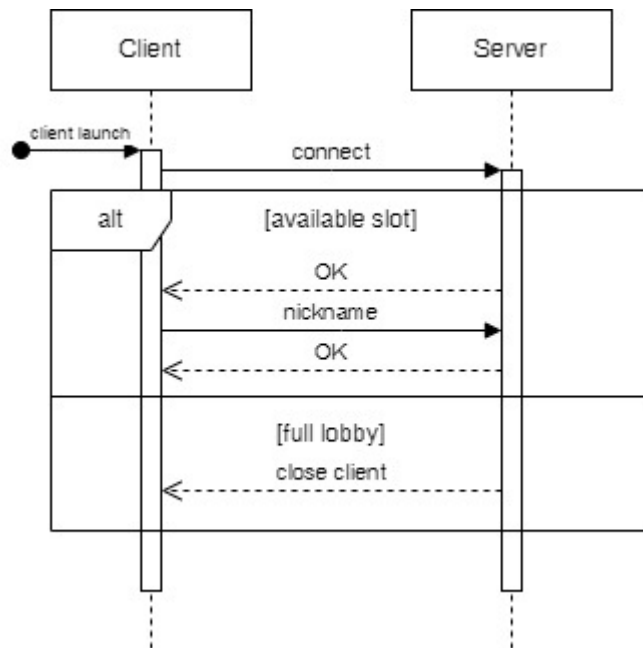
Di seguito si trovano i sequence diagram delle comunicazioni principali dell'implementazione:

INIZIALIZZAZIONE DEL PRIMO GIOCATORE



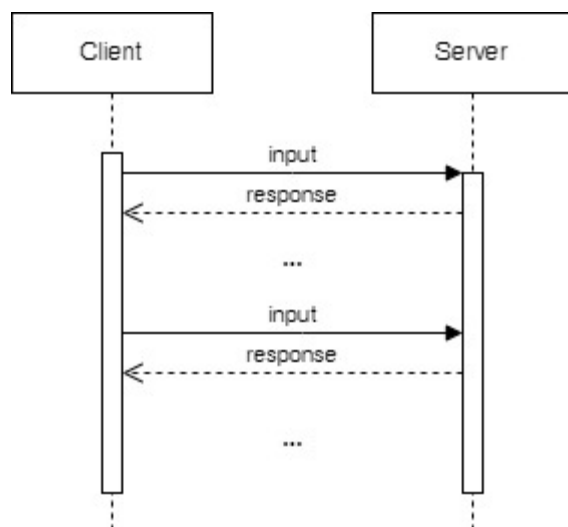
Per prima cosa, il client cerca di connettersi al server. Essendo il primo a connettersi, il server invia l'OK alla connessione. Il client effettua la scelta di creare un nuovo game oppure di recuperare un eventuale salvataggio trovato. Nel primo caso, gli viene chiesto il numero di giocatori e poi il nickname. Nel secondo caso viene recuperato il salvataggio e gli viene chiesto il nickname che aveva nel game precedente.

INIZIALIZZAZIONE DI UN GIOCATORE



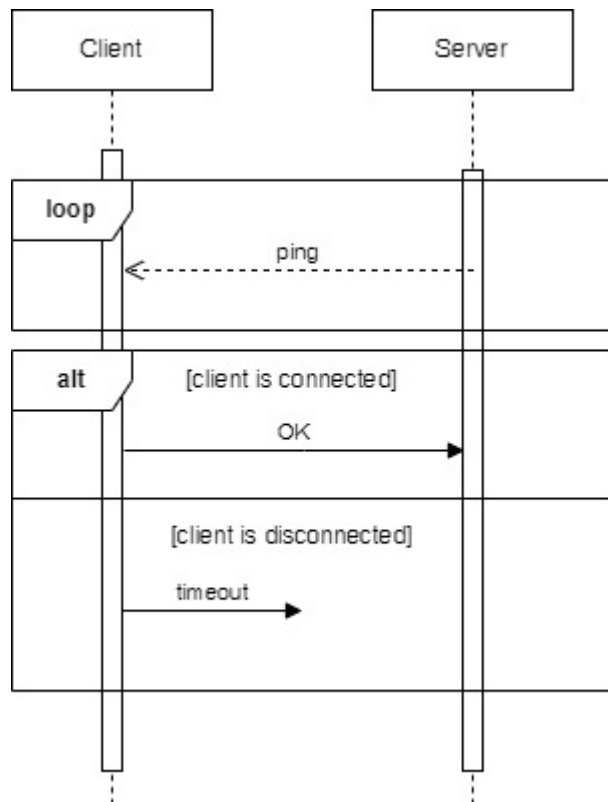
Per prima cosa, il client cerca di connettersi. Se la lobby è piena, il client viene notificato e chiuso. Altrimenti arriva l'OK del server con una lista dei nickname già utilizzati. Il client sceglie quindi un nickname non utilizzato, che viene inviato al server.

INPUT DEL CLIENT



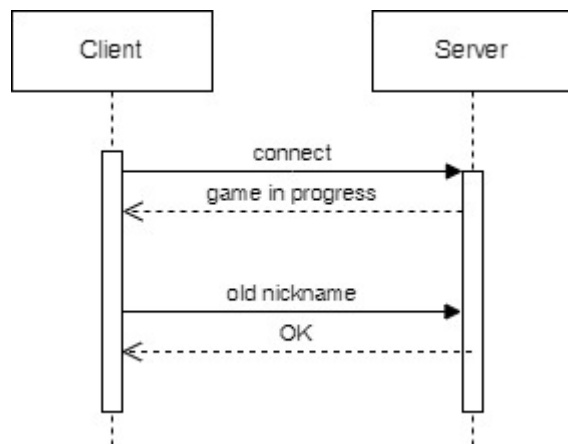
La trasmissione di messaggi si basa su un semplice invio di un messaggio di input da parte del client, che viene salvato in una coda. Al termine dell'esecuzione, viene inoltrata al client la risposta.

RILEVAMENTO DISCONNESSIONE DI UN CLIENT



Il server effettua un ping al client periodicamente. Se non c'è risposta, il client viene segnalato come inattivo e verranno saltati tutti i suoi turni.

RICONNESSIONE DI UN CLIENT



Quando viene lanciato un client a partita iniziata e sono presenti disconnessioni, il server notifica che la partita è in corso. Se il client inserisce il nickname di un giocatore disconnesso, viene riammesso alla partita.