Contents

1 Documento di descrizione Sequence Diagrams

Gruppo IS-AM41: D'Alessio Edoardo, De Ciechi Samuele, Deidier Simone, Ermacora Iacopo.

1.1 Introduzione

Ciao!
Siamo il gruppo AM41 ed innanzitutto vi ringraziamo per il tempo speso sulla revisione del nostro progetto.
Nonostante abbiamo cercato di ideare dei sequence diagrams intuitivi, vi alleghiamo questo breve documento contenente dei commenti sulle idee progettuali e concettuali dei nostri diagrammi.

1.2 Fase di *Set-Up*

La fase di setu-up della partita procede molto in modo lineare, nel diagramma abbiamo presentato due casi (*tenete conto che non è nostra intenzione implementare il multi partita*):

- **fase in cui si connette il primo giocatore in assoluto**. In questa fase il giocatore manda un messaggio di permesso di connessione (*Hello*) ed una volta connesso comunica il suo nickname con il messaggio *Presentation*. Nel caso il gicoatore appena connesso è uno dei giocatori dell'ultima lobby crashata (*vogliamo implementare la persistenza lato server*), il server effettua il restore della lobby e quindi di tutti i parametri e comunica al giocatore che la lobby è stata resettata. Nel caso invece sia un giocatore nuovo, il server richederà i parametri per la creazione della lobby (*quindi numero di giocatori e se si vuole giocare con una o due common target card*) fintantoché non gliene verranno forniti di corretti, dopodiché la lobby verrà creata.
- **fase in cui si connette un giocatore che non è il primo**. Quando un *n-esimo* giocatore richiede di connettersi alla lobby, esso può ricevere due diversi messaggi: *Goodbye* nel caso la unica lobby è piena, oppure *Connected* nel caso vi sia posto. In caso di giocatore connesso, esso comunicherà al

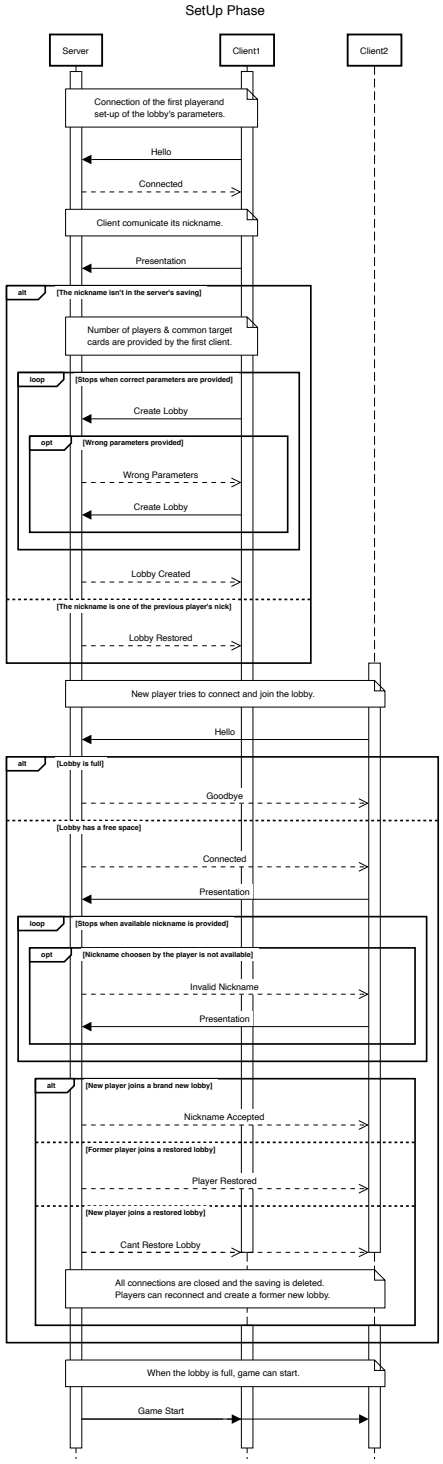

Figure 1: *Sequence Diagram* della fase di Set-Up.

server il suo *nickname*, il quale verrà accettato dal server solamente quando non sarà uguale a quello di uno dei giocatori già connessi. Una volta comunicato un *nick* valido, si presentano tre diverse situazioni:

- *il giocatore si connette ad una nuova lobby*, viene notificato l'ok per il *nickname*.
- *il giocatore si connette ad una lobby caricata da salvataggio ed è uno dei giocatori di tale lobby*, viene comunicato il reset del giocatore.
- *il giocatore si connette ad una lobby caricata da salvataggio, ma non è uno dei giocatori*, in questo caso un messaggio a tutti i player connessi viene inviato notificandoli che non è possibile resettare la lobby, tutte le connessioni vengono chiuse ed il salvataggio sel server viene eliminato. I giocatori potranno riconnettersi nuovamente al server per creare una nuova lobby.

Nel momento in cui la lobby è *full*, la partita comincia e si passa alla fase di *Starting Game*.

1.3 Fase di *Starting Game*

Come descritto nei commenti del *Sequence Diagram*, il server sceglie randomicamente le carte obiettivo comune (*o la carta, nel caso il primo giocatore abbia deciso di giocare con solo una carta*) e per ogni giocatore la carta obiettivo personale e comunica il tutto ai giocatori. Dopo aver instanziato la board con gli item, invia un messaggio di *Update View*.

1.4 Fase di *Running Game*

In questa fase, il giocatore di turno decide la mossa e la comunica al server, il quale la controlla (*nonstante venga fatto un check di correttezza già a lato client*) e, nel caso di errore, la richiede intantoché non ne arriva una valida. In quel caso, il server aggiorna il model edil puntaggio del giocatore di turno, il quale verrà notificato a tale giocatore (*molto probabilmente faremo si che tutti i giocatori verranno notificati cosicché ogni giocatore sa i punteggi anche degli avversari*) ed infine verrà inviato un messaggio di *Update View* con i cambiamenti della shelf e della board.

1.5 *Chat Handling*

Per quanto riguarda la chat vi sono due diverse situazioni: * **un client manda un messaggio *broadcast***, in questo caso il messaggio arriverà al server, il quale si occuperà di reindirizzarlo a **tutti** i client, anche al client che lo ha inviato. Infatti abbiamo deciso che anche al client che ha inviato il messaggio esso comparirà nella chat nel momento in cui viene ricevuto dalla rete (*e non quando il player lo scrive; ci siamo ispirati al funzionamento della chat di Minecraft*). * **un client manda un messaggio *peer-to-peer***, in questo caso il server effettuerà un check sul *nickname* del giocatore a cui è indirizzato il messaggio e, nel caso di *nick* sbagliato, verrà comunicato al client che ha inviato

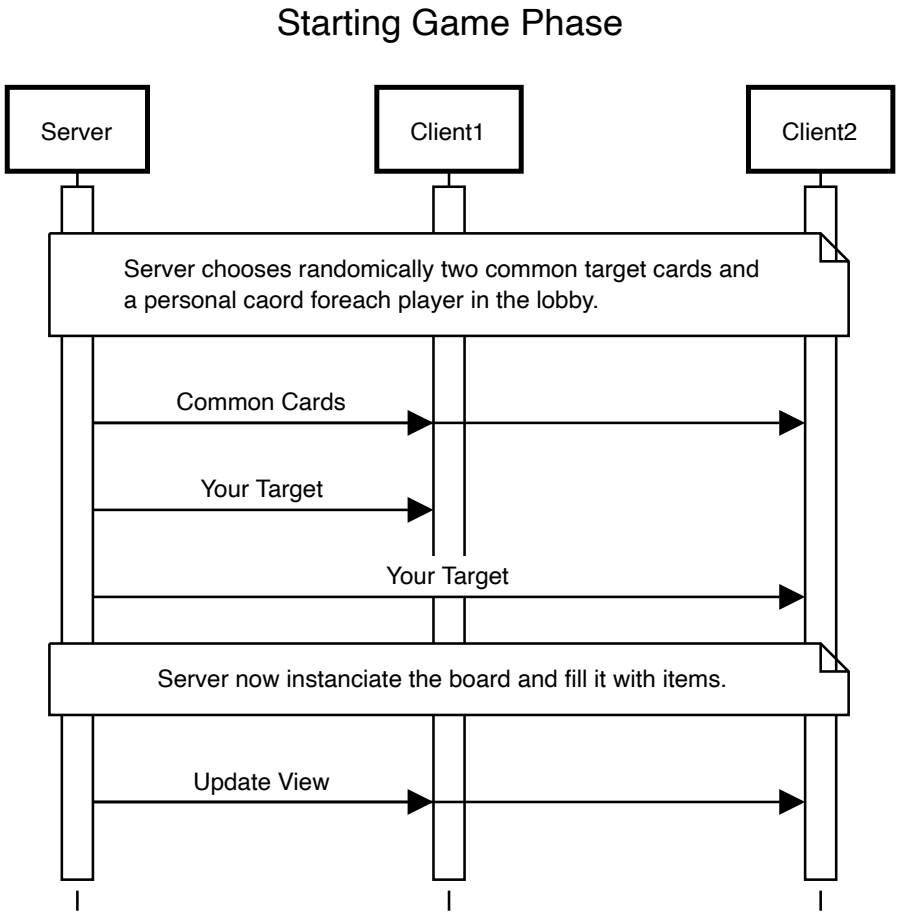

Figure 2: Sequence Diagram della fase di *Starting Game*.

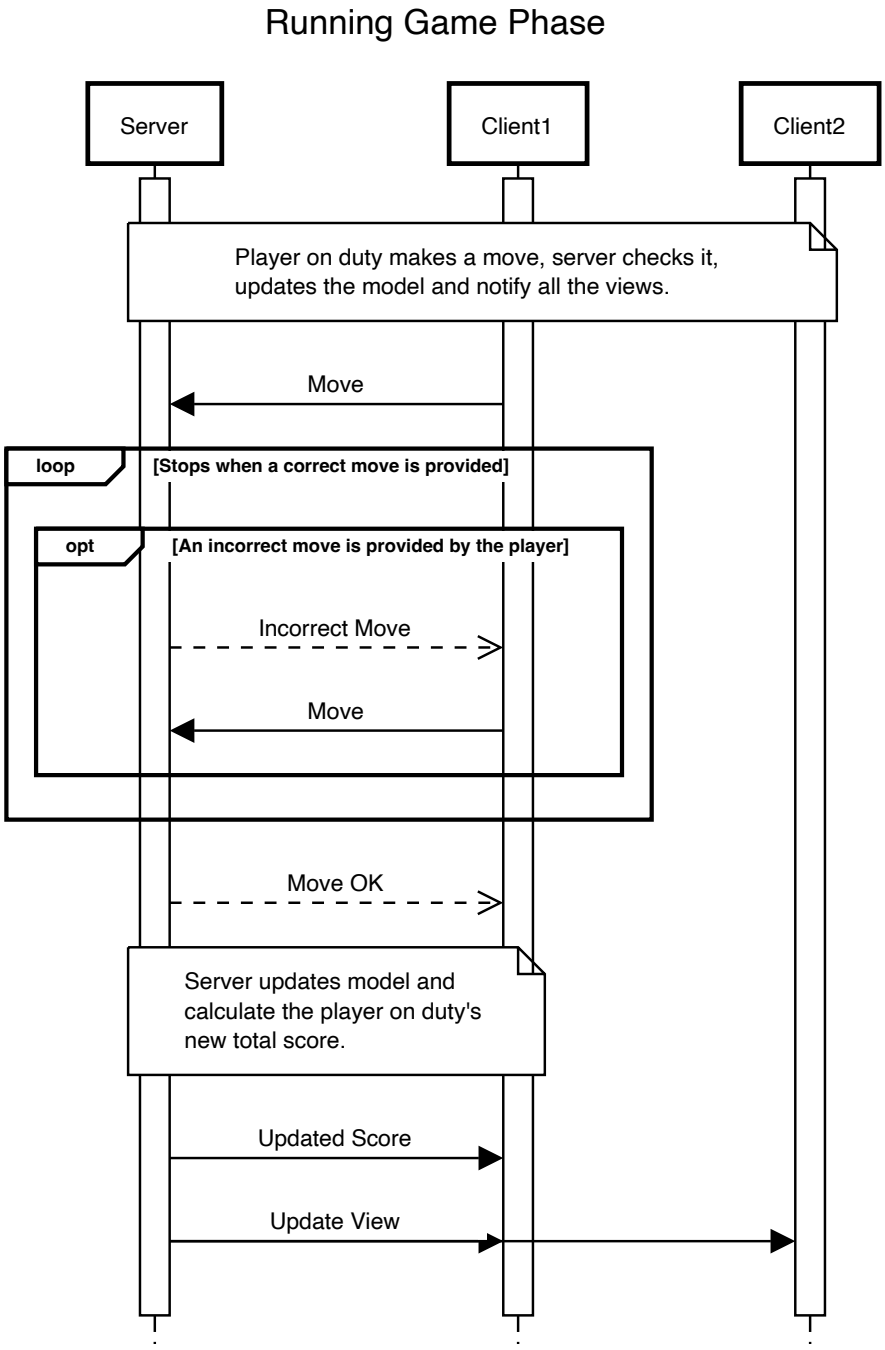

Figure 3: Sequence Diagram della fase di *Running Game*.

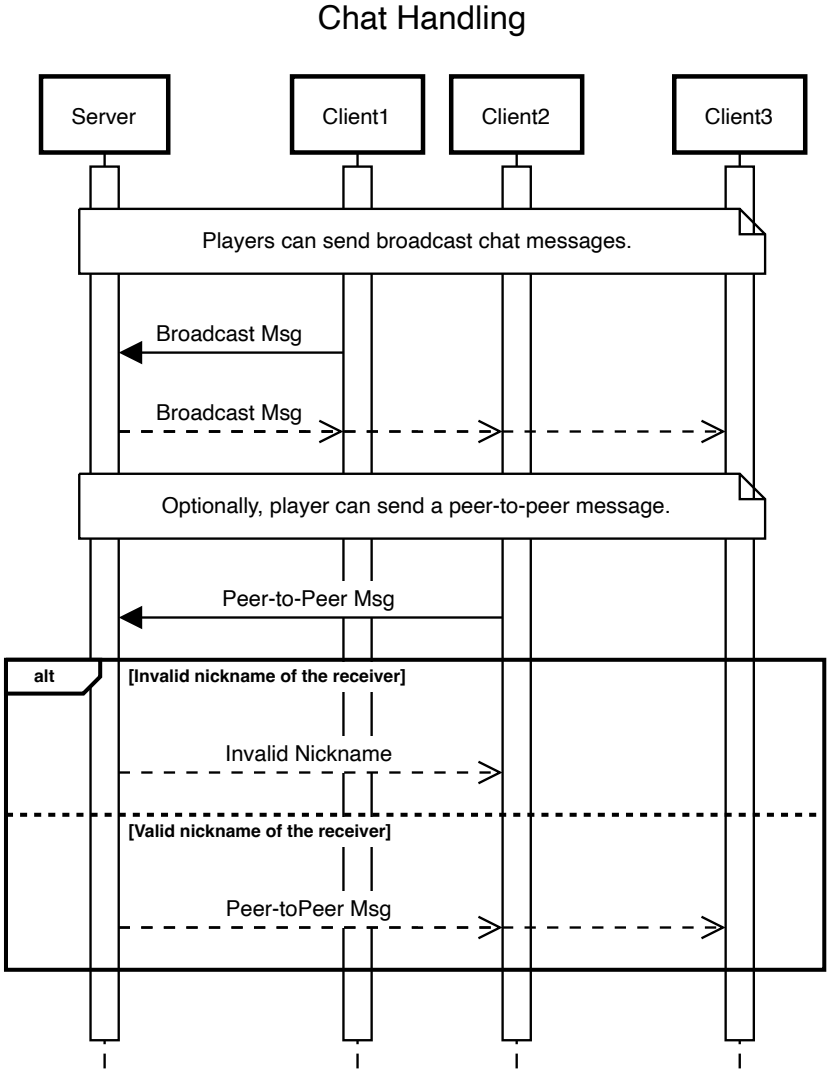

Figure 4: Sequence Diagram della gestione della chat.

il messaggio, altrimenti il messaggio verrà correttamente inoltrato al giocatore segnalato.

1.6 *Player Disconnection Handling*

Abbiamo deciso di implementare un sistema di *ping-pong* tra server e client, ovvero ogni secondo il server invia un messaggio di *Check* mentre i client mandano un messaggio di *Clear*. Questa cosa non deve succedere per forza in sequenza, è un meccanismo asincrono, l'importante è che tra ogni messaggio di *Check* inviato dal server vi intercorra un periodo di un seconodo, così come tra ogni messaggio di *Clear* dei client. Abbiamo quindi poi individuato due diversi scenari: * **caso in cui il client si disconnette**, il che può avvenire in due modalità diverse: * *il client non risponde*, ovvero dopo cinque messaggi di *Check* del server che non ricevono almeno un *Clear*, il server assume il giocatore disconnesso e la partita continua saltando il turno di tale giocatore disconnesso. * *il client si disconnette intenzionalmente*, in questo caso il client invia un messaggio di disconnesione (*Disconnect*), la connessione dal server viene chiusa e la partita continua saltando il turno del giocatore disconnesso.

Una volta disconnesso, un giocatore può ricollegarsi alla lobby riutilizzando il *nickname* scelto per la partita alla quale si è disconnesso, mentre nel caso di *nickname* errato esso non verrà riconneso.

- **caso in cui il server crasha**, cinque messaggi *Clear* dei client che non ricevono almeno un *Check* è indice di un crash del server, pertanto le connessioni lato client vengono chiuse e la partita termina. Una volta che il server viene riavviato, si ricomincia dalla fase di *Set-Up*, dove la partita interrotta dal crash del server può essere ricaricata grazie alla persistenza di tale (*ogni tot il server esegue il salvataggio in locale della partita*).

1.7 Ringraziamenti

Vi ringraziamo nuovamente per il tempo dedicatoci, non esitate a contattarci per ulteriori chiarimenti.
Edoardo, Samuele, Simone e Iacopo - Gruppo AM41

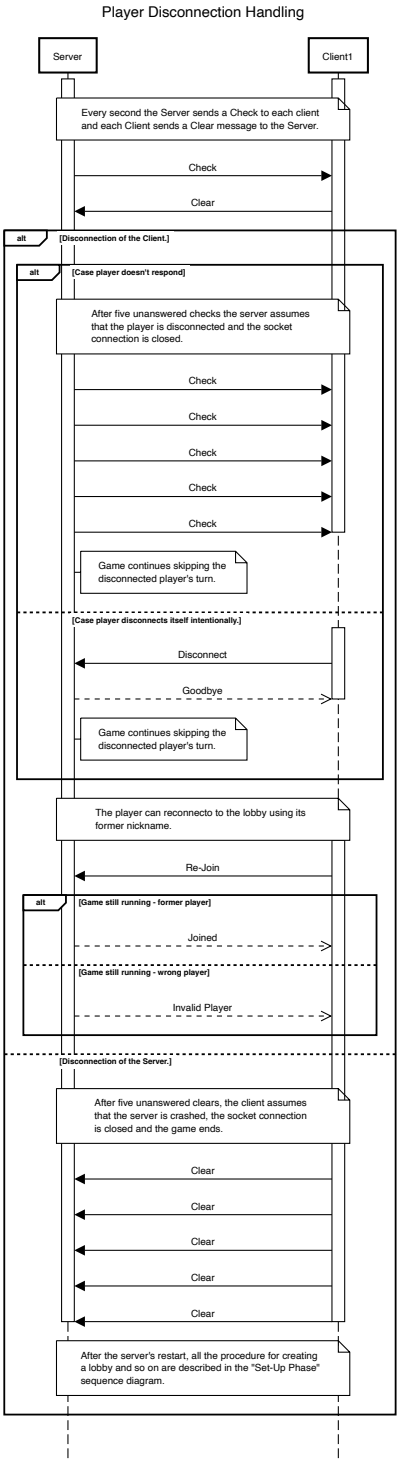

Figure 5: Sequence Diagram della gestione della disconnesione di un giocatore.