**Protocollo di comunicazione**

Tutti i messaggi scambiati tra client e server sono oggetti Java serializzati, ciascuno dei quali implementa un’interfaccia comune Message.

Quando la connessione viene inizializzata il server manda un messaggio Ping al client, il quale risponde con un messaggio Pong. Questa procedura viene ripetuta per l’intera durata della connessione: al venire meno del messaggio Pong da parte del client il server procede con la disconnessione del relativo partecipante.

Per la gestione degli errori causati dall’input del client il server è in grado di inviare messaggi appartenenti alla classe ErrorMessage, caratterizzati da un attributo di tipo enum ErrorKind, contente le tipologie di errori che possono essere commessi dal client.

public enum ErrorKind {

INVALID\_NICKNAME,

INVALID\_INPUT,

INCORRECT\_TURN,

ILLEGAL\_MOVE }

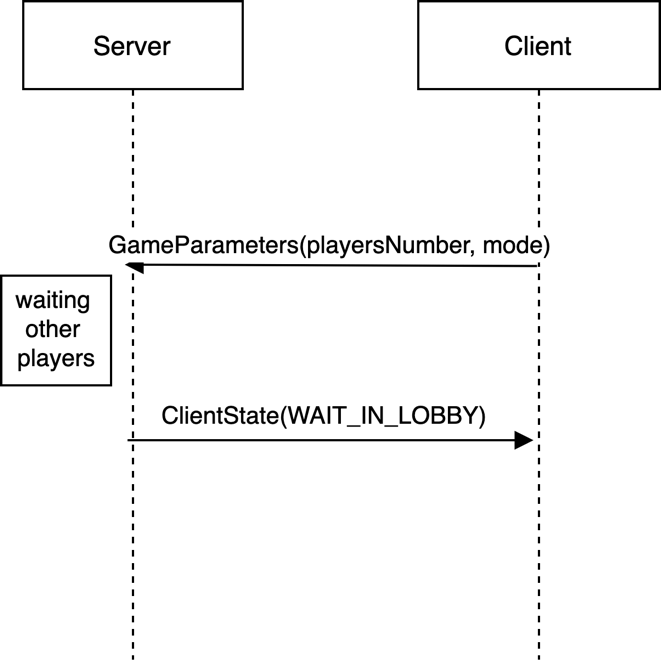
I messaggi possono essere divisi per fasi: ConnectionPhase, SetupPhase, PlanningPhase, ActionPhase e EndGamePhase.

**ConnectionPhase**

Immagine che contiene testo

Descrizione generata automaticamente

Stabilita la connessione client-server il client invia il proprio nickname al server: nel caso in cui il nome scelto fosse disponibile il server risponde confermando il nickname e facendo passare il client allo stato successivo, nel caso non lo fosse il server invia un messaggio di tipo errore, specificando la tipologia INVALID\_NICKNAME.



Il client invia al server il numero desiderato di giocatori e la modalità scelta.

Nel caso in cui l’input fosse valido il server aggiunge il giocatore ad una lobby in attesa che si connettano tanti giocatori quanto il numero impostato.

Lo stesso messaggio viene inviato ad un giocatore che si connette quando è esistente una partita che non ha ancora raggiunto il numero necessario di partecipanti.

**SetupPhase**

****

Questa iterazione di messaggi tra server e singolo client viene ripetuta per ogni partecipante alla partita, procedendo nell’ordine di registrazione della connessione al server.

Una volta che ogni partecipante ha selezionato la torre e il mago il server comunica le scelte a tutti i client e dà inizio alla partita.

**PlanningPhase**

****

Seguendo l’ordine di gioco ogni partecipante seleziona una carta assistente tra quelle disponibili: in caso di scelta valida il server si occupa di aggiornare lo stato del gioco, in caso contrario il server manda un messaggio di errore del tipo ILLEGAL\_MOVE.

**ActionPhase**

****

Seguendo l’ordine stabilito attraverso le priorità delle carte assistente inizia la fase di azione, caratterizzata da un turno per ogni partecipante. Ogni giocatore muove tre studenti da spostare dall’ingresso della propria plancia al tavolo o su un’isola. A seconda che lo spostamento avvenga su un’isola o sulla tavola e nel caso fosse una mossa legale il server risponde con uno dei due messaggi.

Il partecipante successivamente può procedere con lo spostamento di Madre Natura: nell’eventualità lo spostamento generasse anche l’unione di isole il server lo comunicherebbe con un messaggio specifico.

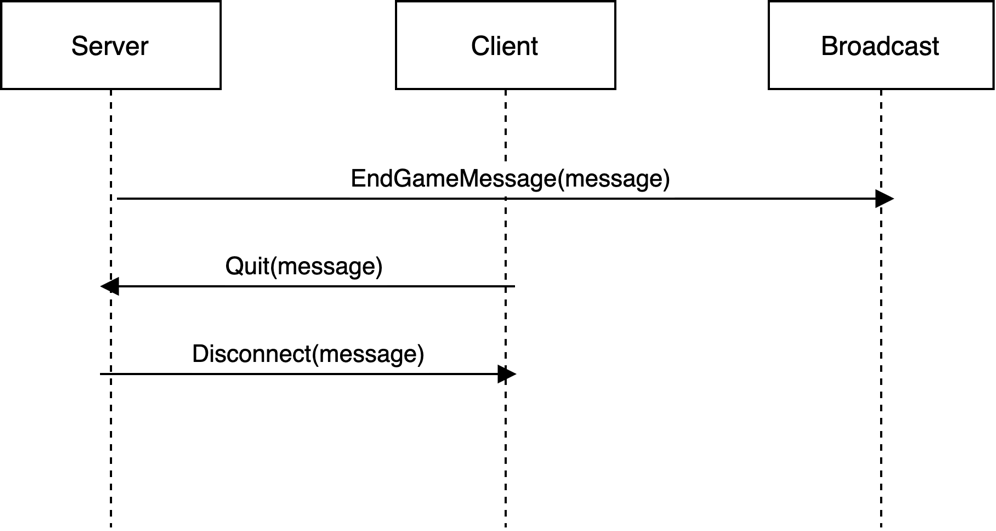
Il partecipante può decidere di giocare una carta personaggio durante questa fase e, in funzione della carta giocata, possono svilupparsi scambi di messaggi molto differenti.

Nel caso venisse soddisfatta una condizione per cui al termine del round corrente la partita giungerebbe al termine il server comunica la situazione a tutti i client.

Prima di terminare il turno il partecipante sceglie una nuvola da cui prelevare le pedine studente.

Una volta svolte le azioni previste dal gioco il partecipante comunica la sua decisione di chiudere il turno al server.

**EndGamePhase**



Al termine della partita il server comunica a tutti i client il nome del vincitore, se presente, e in caso di pareggio il server comunica in broadcast il risultato.

In seguito a ciò ogni partecipante può uscire volontariamente dalla partita oppure aspettare che il server lo disconnetta automaticamente.