PROTOCOLLO DI COMUNICAZIONE

Gruppo 42

Arturo, Lorenzo, Luca

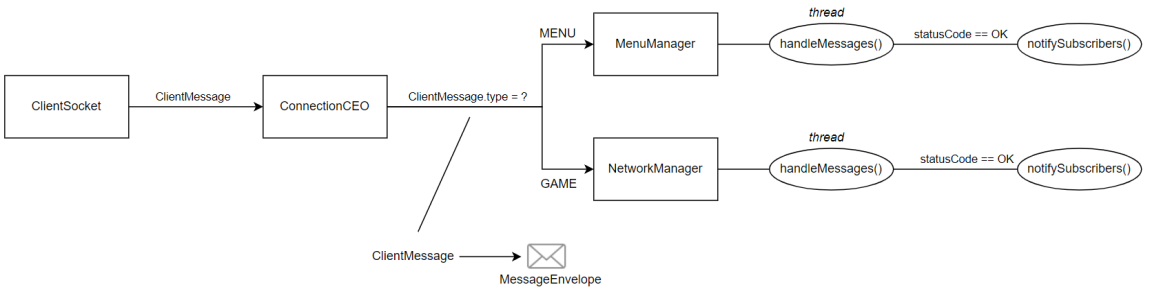
Classi

- **Server:** Classe singleton lato server, si occupa di inizializzare il gestore di connessioni (ConnectionCEO) quando arriva una nuova richiesta di connessione da un client.
- **ConnectionCEO**: Classe che gestisce le connessioni in arrivo lato server. Accetta le connessioni controllando l'univocità dell'username scelto, inoltra i messaggi in entrata al MenuManager o NetworkManager in base al tipo di messaggio. Alla creazione, si iscrive al MenuManager.
- **MenuManager:** Classe che gestisce (ed esegue) la coda di messaggi di tipo MENU. Gestisce anche l'invio di MenuContent ai suoi iscritti. Tale invio avviene ogni volta che viene gestito un messaggio con successo.
- **NetworkManager:** Classe che gestisce (ed esegue) la coda di messaggi di tipo GAME. Gestisce anche l'invio di LobbyContent/GameContent ai suoi iscritti. Tale invio avviene ogni volta che viene gestito un messaggio con successo; in questo modo siamo in grado di propagare i cambiamenti a tutti i componenti della lobby/game. Contiene un LobbyHandler ed un GameHandler, che utilizzerà in base al suo stato corrente (Lobby o Game).
- **LobbyHandler:** Classe che modella la Lobby, contiene metodi per l'aggiunta di giocatori e l'assegnazione del Wizard scelto.
- **GameHandler:** Classe che modella il Game, contiene il controller del gioco (Game), ed attributi utili alla gestione dello stato della partita.
- **ClientSocket:** Classe lato client per la gestione delle connessioni.

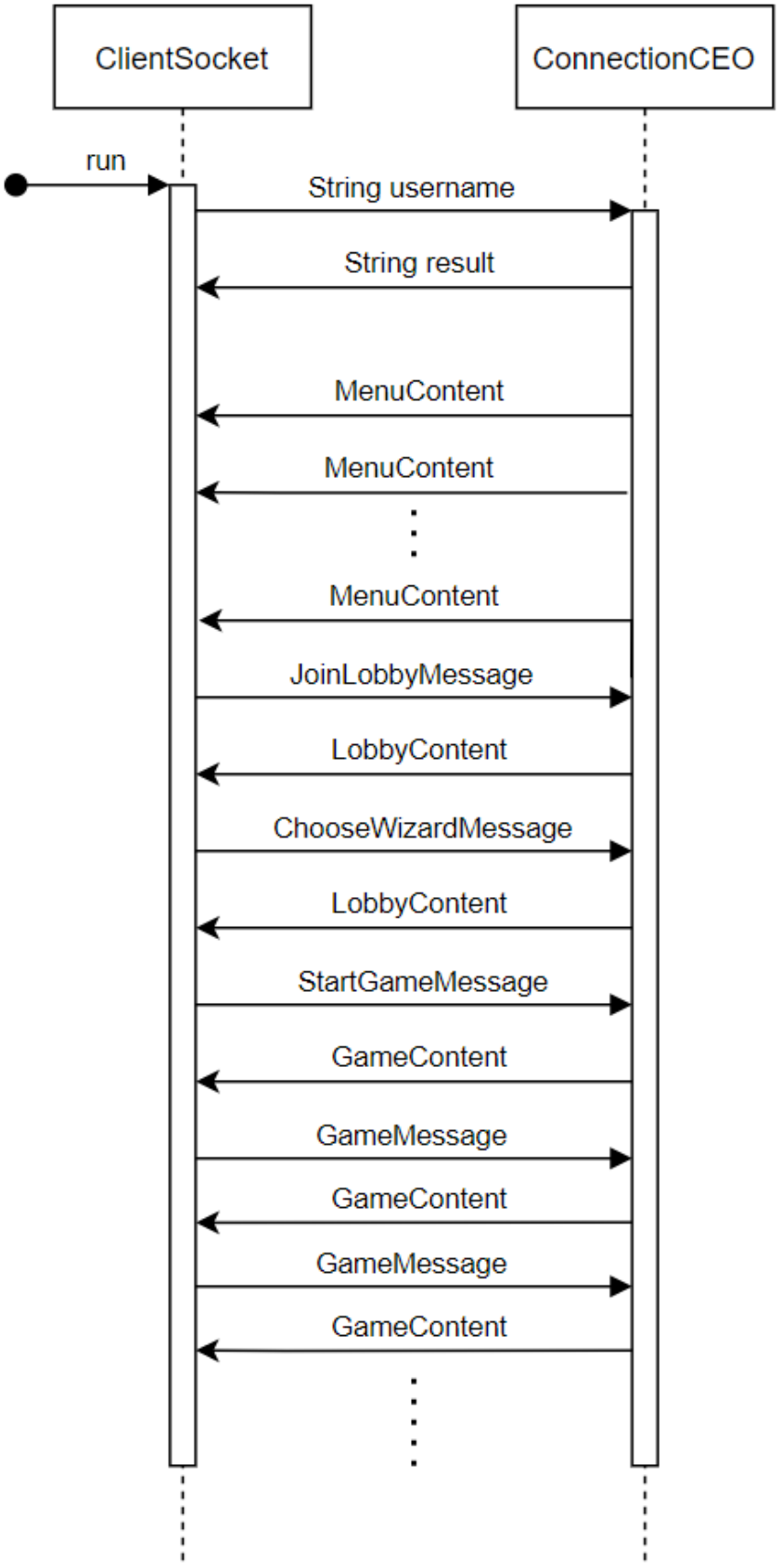
- **ClientMessage**: Classe astratta che modella i messaggi inviati dal client al server.
    - **CreateLobbyMessage:** Messaggio di creazione di una lobby; in caso di successo crea un NetworkManager (in stato Lobby) e vi iscrive il ConnectionCEO, disiscrivendolo dal MenuManager.
    - **JoinLobbyMessage:** Messaggio di join di una lobby; in caso di successo iscrive il ConnectionCEO al NetworkManager, disiscrivendolo dal MenuManager.
    - **ChooseWizardMessage:** Messaggio di selezione di un wizard. Controlla che il wizard sia disponibile e che quel giocatore non abbia già scelto un wizard.
    - **StartGameMessage:** Messaggio di inizio di una partita; in caso di successo crea un nuovo GameHandler.
    - **SelectedCharacterMessage:** Messaggio per indicare quale character selezionare.
    - **ActivateCharacterMessage:** Messaggio per l'attivazione di un character. Vengono indicati parametri per attivarlo. Questo messaggio è da mandare dopo aver selezionato il character (SelectedCharacterMessage)
    - **ChooseCloudMessage:** Messaggio per la scelta della nuvola a fine turno.
    - **MoveMotherNatureMessage:** Messaggio per il movimento di madre natura.
    - **PlayAssistantMessage:** Messaggio per la scelta di un assistente.
    - **NextStateMessage:** Messaggio per passare da uno stato del gioco al successivo. Questo deve essere inviato dopo tutte le azioni del gioco.
    - **MoveStudentMessage:** Messaggio di movimento di uno studente.
- **MessageEnvelope**: Contiene il giocatore che ha inviato il messaggio (LobbyPlayer sender), il messaggio inviato (ClientMessage message), il gestore della connessione del giocatore che ha inviato il messaggio (ConnectionCEO connectionCEO).
- **ViewContent**: Classe astratta che contiene le informazioni della view che il server deve mandare al client. La separazione tra le classi figlie è stata fatta per permettere di inviare solo le informazioni relative alla specifica fase di gioco.
    - **MenuContent:** Inviato dal server mentre il giocatore è nella selezione della lobby. Contiene l'elenco delle lobby.
    - **LobbyContent:** Inviato dal server mentre il giocatore è nella lobby. Contiene informazioni riguardo la lobby interessata.
    - **GameContent:** Inviato dal server mentre il giocatore è in gioco. Contiene le informazioni del modello dalla partita interessata.

Protocollo di comunicazione

1) Il client inizializza la comunicazione.
2) Il client invia il suo username. Se è corretto il server risponderà con la stringa “OK”, mentre se non è valido restituisce “ERROR: Username already taken” e ripete il punto 2.
3) Il server invia un MenuContent iniziale, altri MenuContent verranno inviati quando qualcuno creerà o entrerà in una lobby.
4) Il client invia uno dei messaggi per passare dal Menu ad una Lobby (ovvero JoinLobbyMessage e CreateLobbyMessage). Il server smette di inviare MenuContent e inizia ad inviare LobbyContent ogni volta che avviene un cambiamento nella lobby (es. un giocatore si unisce alla partita).
5) Il client invia dei messaggi relativi alla lobby (ovvero ChooseWizardMessage e StartGameMessage).
6) Appena il server riceve uno StartGameMessage valido, il server smette di inviare messaggi LobbyContent e inizia ad inviare GameContent.



Flusso dei messaggi lato Server



Sequence diagram esemplificativo: comunicazione client-server