

# Sequence Diagram

## Communication Protocol

List of all messages between client and server, divided in messages sent by the server and those sent by the client. We decided to try to implement in the communication protocol three advanced functionalities: chat, multigame and resilience to disconnections.

### Server messages:

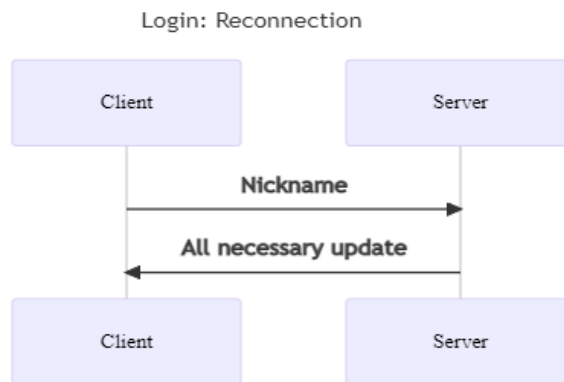
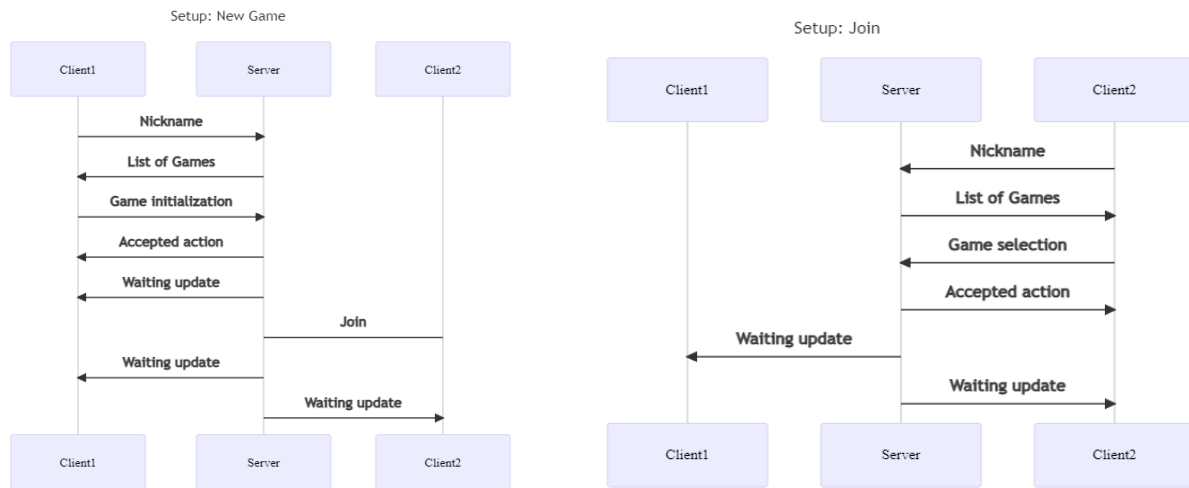
List of Games	<code>[{id: int, numberPlayers: int, numberCommonGoalCard: int}]</code>
Accepted action	<code>{accepted: boolean}</code>
Livingroom update	<code>{livingroom: [[Item]]}</code>
Bookshelf update	<code>{nickname: String, bookshelf: [[Item]]}</code>
Waiting update	<code>{missing: int}</code>
Scores update	<code>[{nickname: String, score: int}]</code>
EndingToken update	<code>{endingToken: String}</code>
CommonGoalCard update	<code>{id: int, top: int}</code>
PersonalGoalCard update	<code>{id: int}</code>
Chat update	<code>{nickname: String, text: String}</code>
Start turn update	<code>{nickname: String}</code>
EndGame update	<code>{winner: String}</code>
Tile selection response	<code>{items: [Item]}</code>

### Client messages:

Nickname	<code>{nickname: String}</code>
Game selection	<code>{nickname: String, id: int}</code>
Game initialization	<code>{nickname: String, numberPlayers: int, numberCommonGoalCard: int}</code>
Livingroom selection	<code>{nickname: String, positions: [Position]}</code>
Bookshelf insertion	<code>{nickname: String, column: int, permutation: [int]}</code>
Chat messages	<code>{nickname: String, text: String}</code>

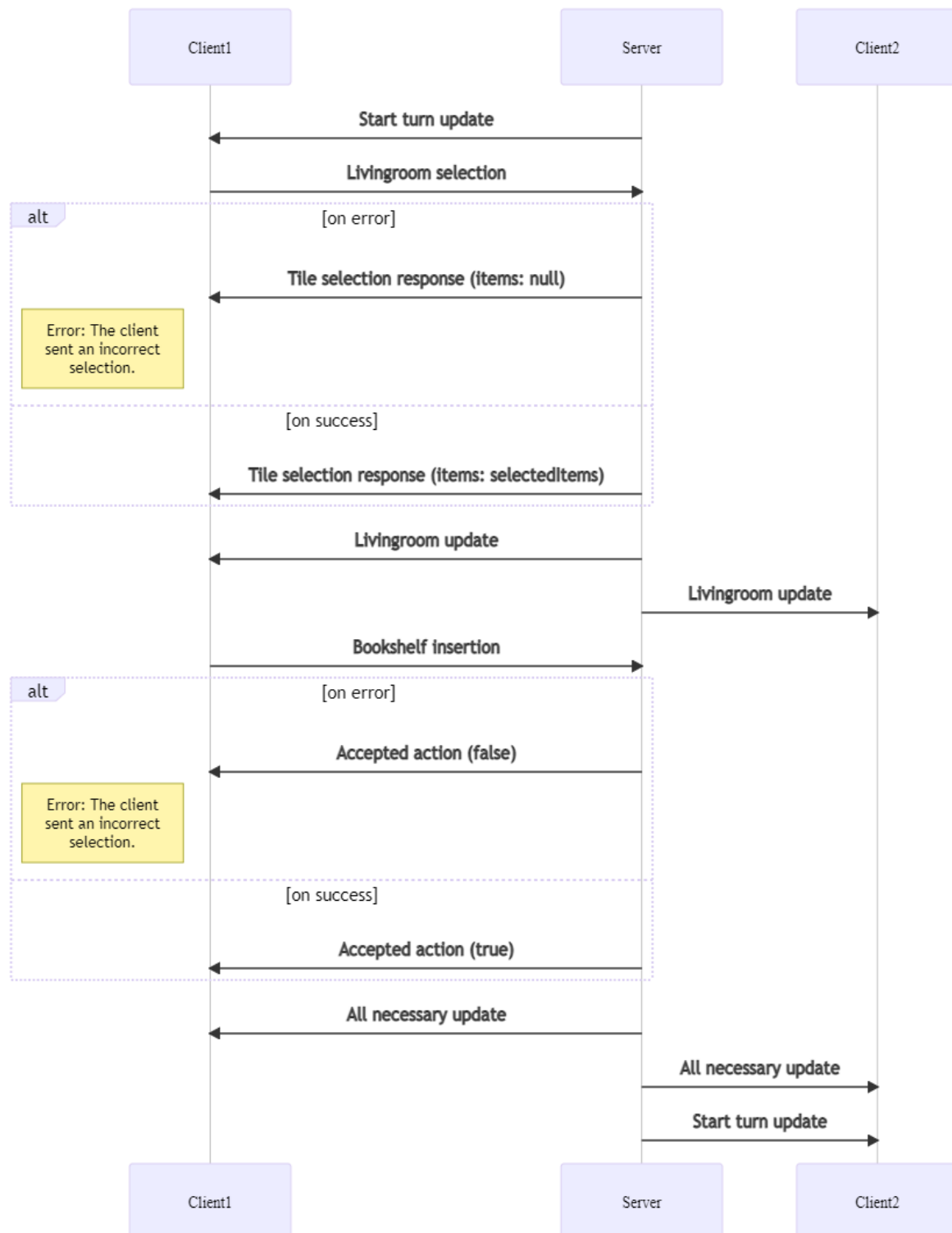
## Login

The client first inserts their Nickname, then they can choose to create a new game or to join an existing game which has not started yet. Instead, if the client's nickname is already registered and the client is inactive, a reconnection is triggered.



## Turn

First of all the server notifies the Start Turn, then the client selects the tiles in the Livingroom. On success the client needs to select the column and the order of the selected tiles to put in the Bookshelf. On success the server starts the next turn.



## Endgame

The last turn is played and then the server notifies all connected clients that the game has just finished. There is no need to send the scores as the clients already have that information.

