Gruppo AM23: De Lucia, Battiston, Dell'Era

CLIENT SERVER COMMUNICATION PROTOCOL

Clients and server communicate through the exchange of java objects. We define a Notification interface that is implemented in various ways. We use the visitor pattern to evaluate notifications in both client side and server side. Each of the message implementations contains attributes and methods to be satisfied.
We allow multiple games to start, but only fill one game at a time.

Notification implementations

- LoginNotification
- ChooseGameModeNotification
- ChooseWizardAndTowerNotification
- ChooseAssistantNotification
- MoveStudentNotification
- MoveMNNotification
- ChooseCloudNotification
- UseCharacterNotification
- DisconnectionNotification
- WinNotification
- ...

Lobby

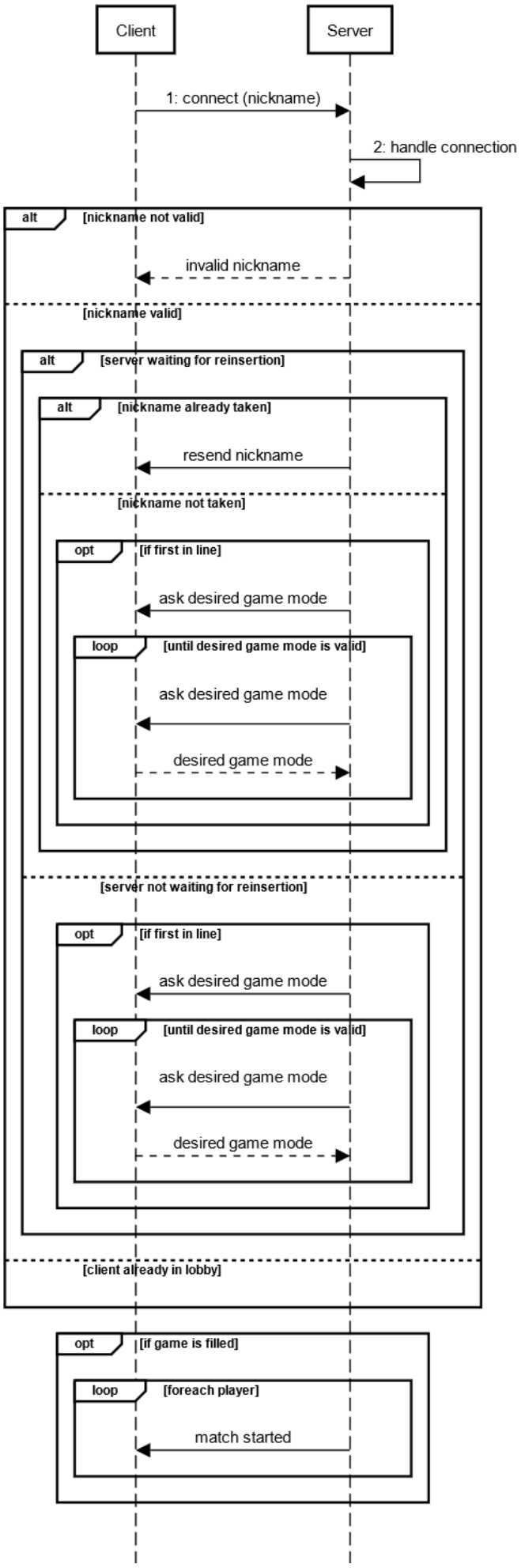
When a client connects, it immediately sends a nickname to the server. Based on this nickname, several things can happen:

- the client is already in the lobby, the message is ignored
- the client has chosen a nickname already used, he is asked to re-enter it
- the client is the first in the waiting queue: he is asked for the game modes and the number of players

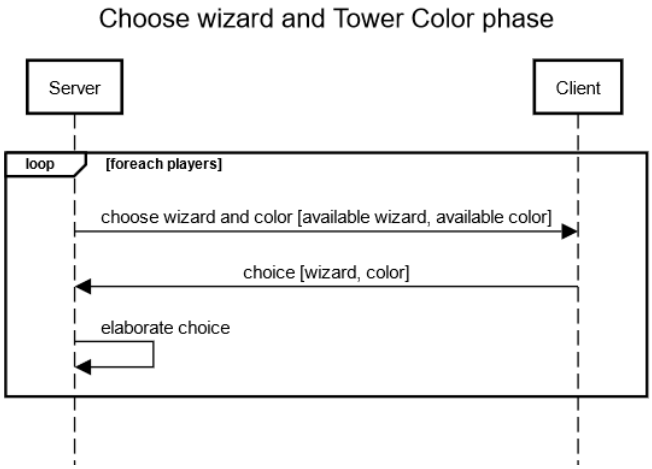
After a player logs in, if the queue of players meets the requirements chosen by the first player, the game can begin and the players are notified.

After the start of a game, if there are still clients in the queue, the first in line will be prompted for game mode.

Login phase


Client
Server
1: connect (nickname)
2: handle connection
alt
[nickname not valid]
invalid nickname
[nickname valid]
alt
[server waiting for reinsertion]
alt
[nickname already taken]
resend nickname
[nickname not taken]
opt
[if first in line]
ask desired game mode
loop
[until desired game mode is valid]
ask desired game mode
desired game mode
[server not waiting for reinsertion]
opt
[if first in line]
ask desired game mode
loop
[until desired game mode is valid]
ask desired game mode
desired game mode
[client already in lobby]
opt
[if game is filled]
loop
[foreach player]
match started

When a game is ready to be created all players are asked to choose a wizard and a color from a list of possible values.

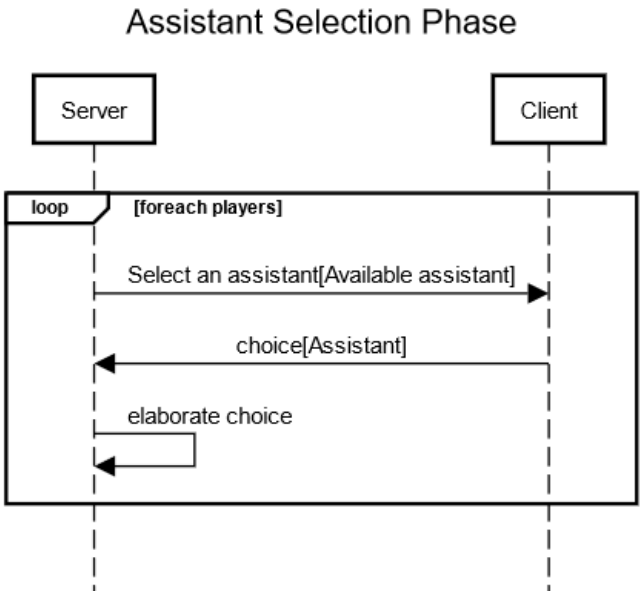


In Game

The first "first player" is drawn randomly.
Each game turn is made up of two subturns that iterate over all players: planning and action.
The possible choices of the various requests are pre-calculated on the server side and verified on the client side. With each request from the server a timeout is initiated, so the players have a finite time to execute the moves.

Planning phase

Following the game logic, the server pre-calculates a list of Assistants that the client can play; sends it to the client asking to select one.

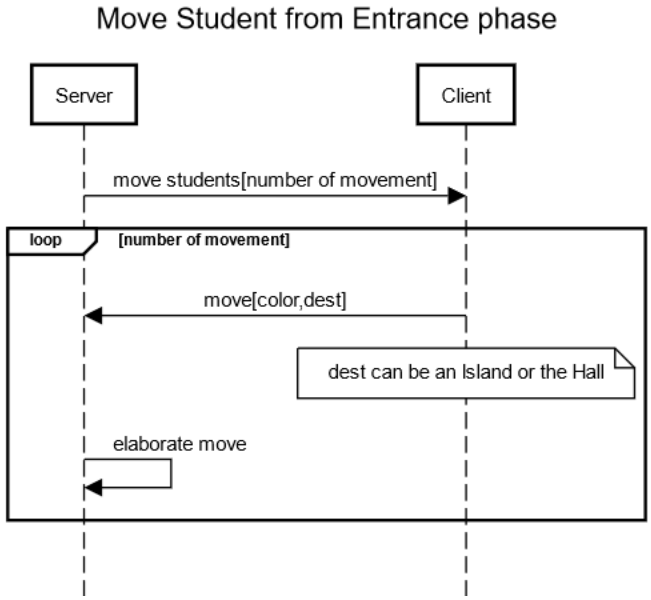


Action phase

The action phase consists of 3 actions that are performed in sequence
Following the order calculated by the server, each of the clients is asked to execute these phase.

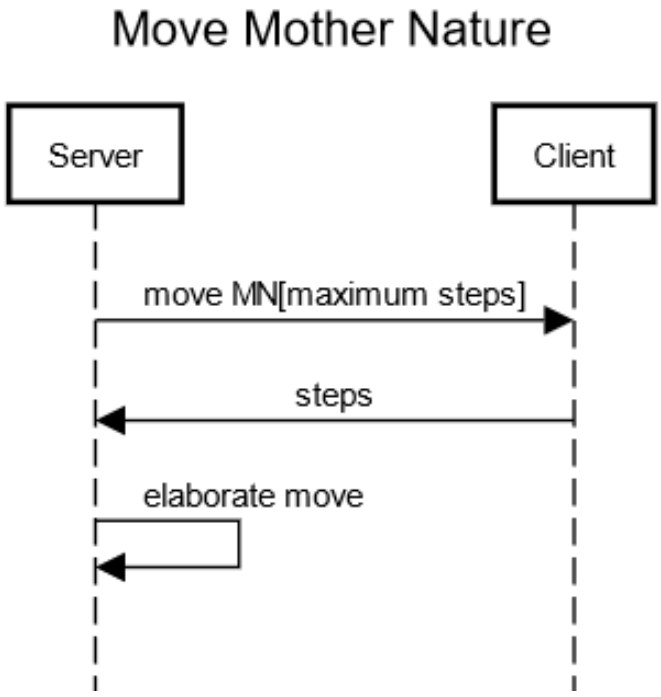
Moving Student phase

The client is asked to move a certain number of students to an Island or to the Hall.



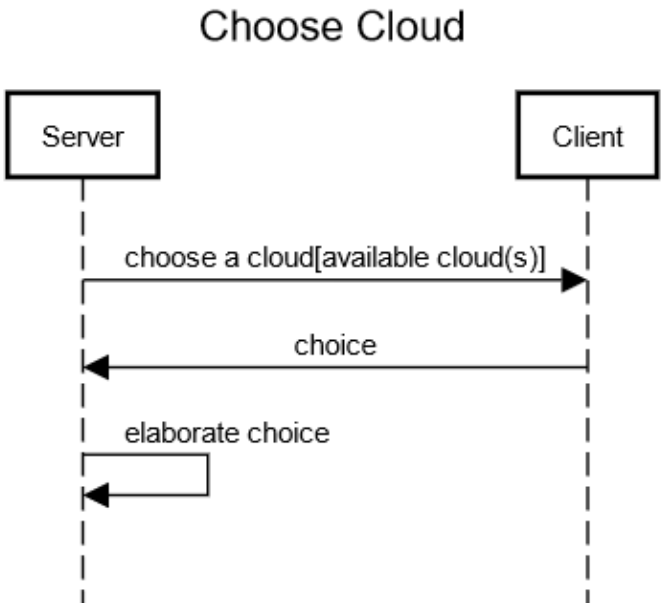
Moving MN phase

A MoveMotherNature message is sent to the client with an integer representing the maximum number of steps that mother nature can do.



Choose cloud phase

The client is sent a ChooseCloud message with all the available clouds. Then it will have to choose one



Character Use

At any point during the action turn, a client can decide to use a character. Each character has a set of requirements that must be indicated by the client in order for the character to perform a corresponding action.