Sequence Diagrams

Gruppo 52

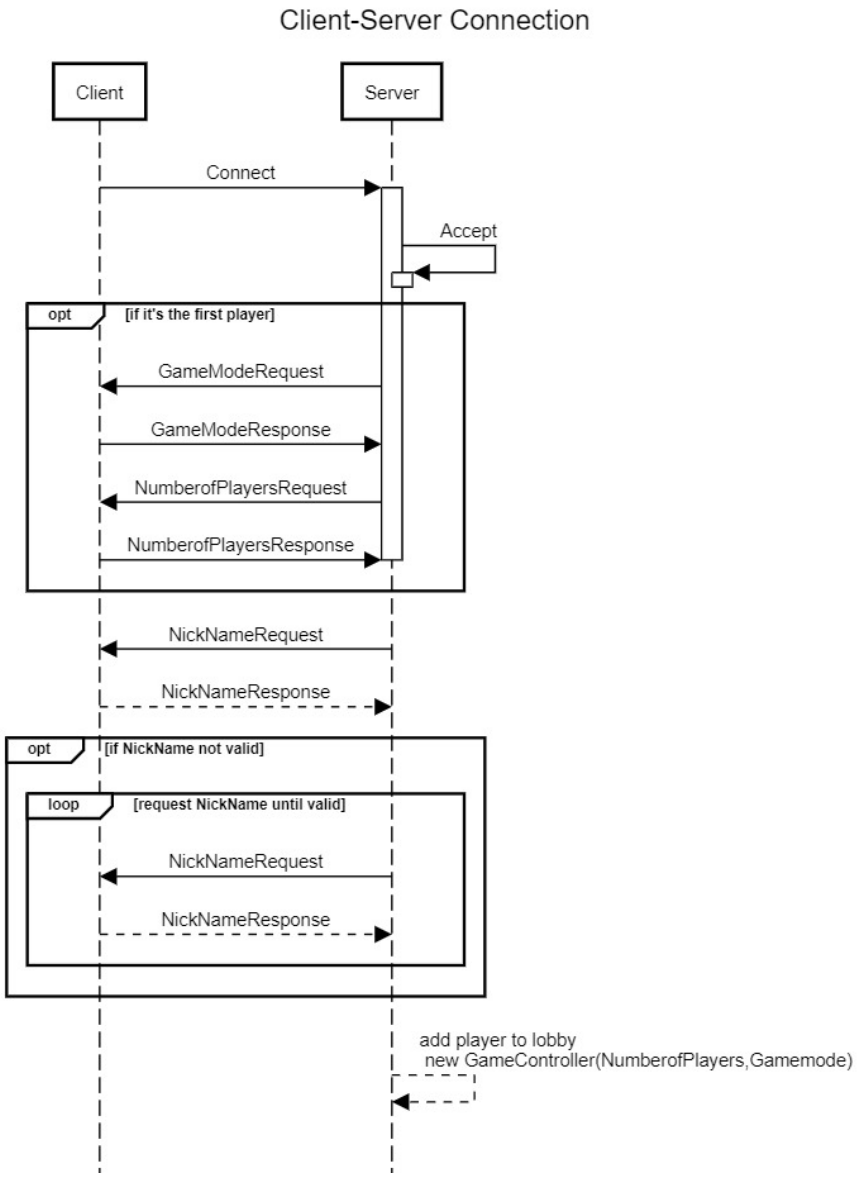
Filippo Fini
Matteo Fiorentino
Luca Longinotti

Contents

1 Player access to the game

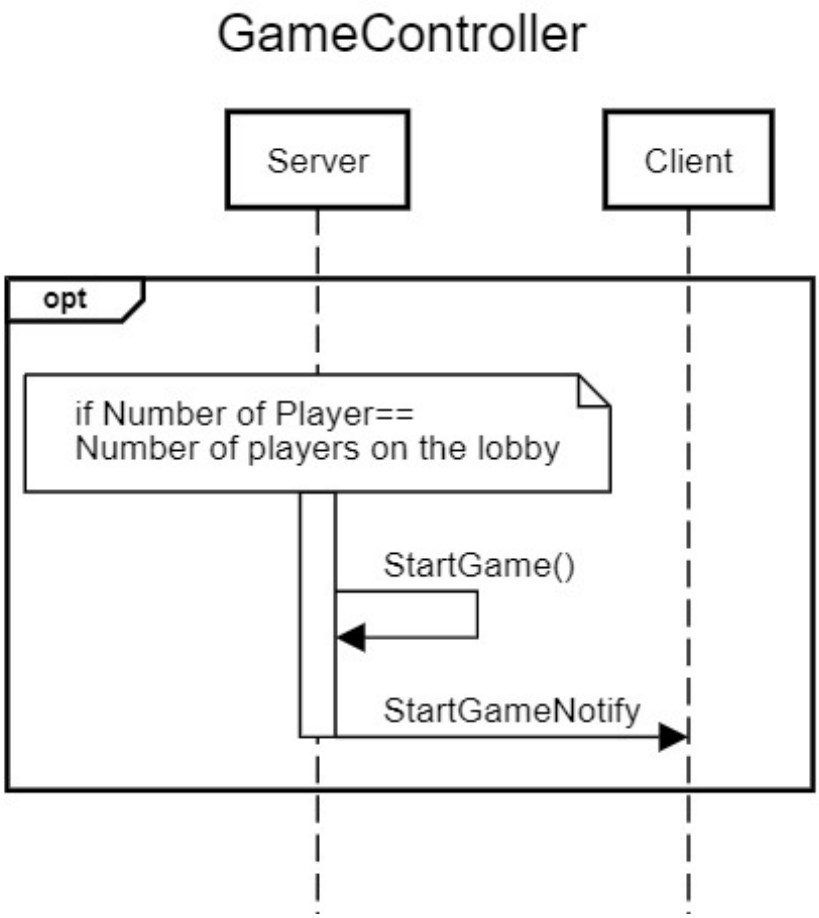
1.1 Player connection

After the server has been started with its own port number, a client can start a connection by providing the server's port number and IP address.
Once the client is connected, if that is the first client, the server asks for the game mode (standard or expert) using the GameModeRequest message and for the number of players using the NumberofPlayersRequest message.
The server waits for the responses from the client and then asks for the nickname with NickNameRequest, it checks if the nickname is valid, if not, it asks again for a new nickname until it's valid.After that, the server adds the player to the lobby, and creates a new GameController.



1.2 GameController

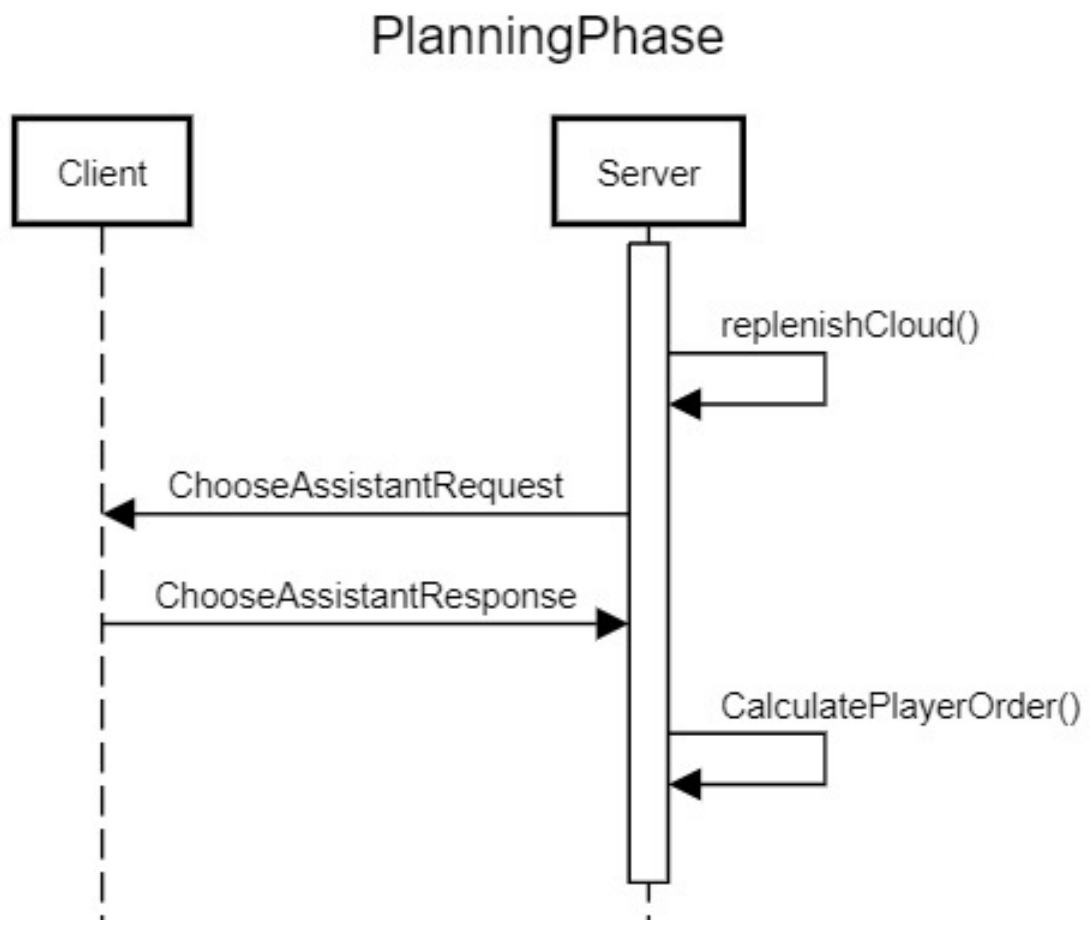
After each player has entered the lobby the server checks if the number of players is equal to the numbers of player specified by the first player: if it is, the server starts the game and sends to each client the StartGameNotify; if not the server waits for the next player.
The server creates a TurnController that will manage the turn flow.



2 Turn

2.1 Planning phase

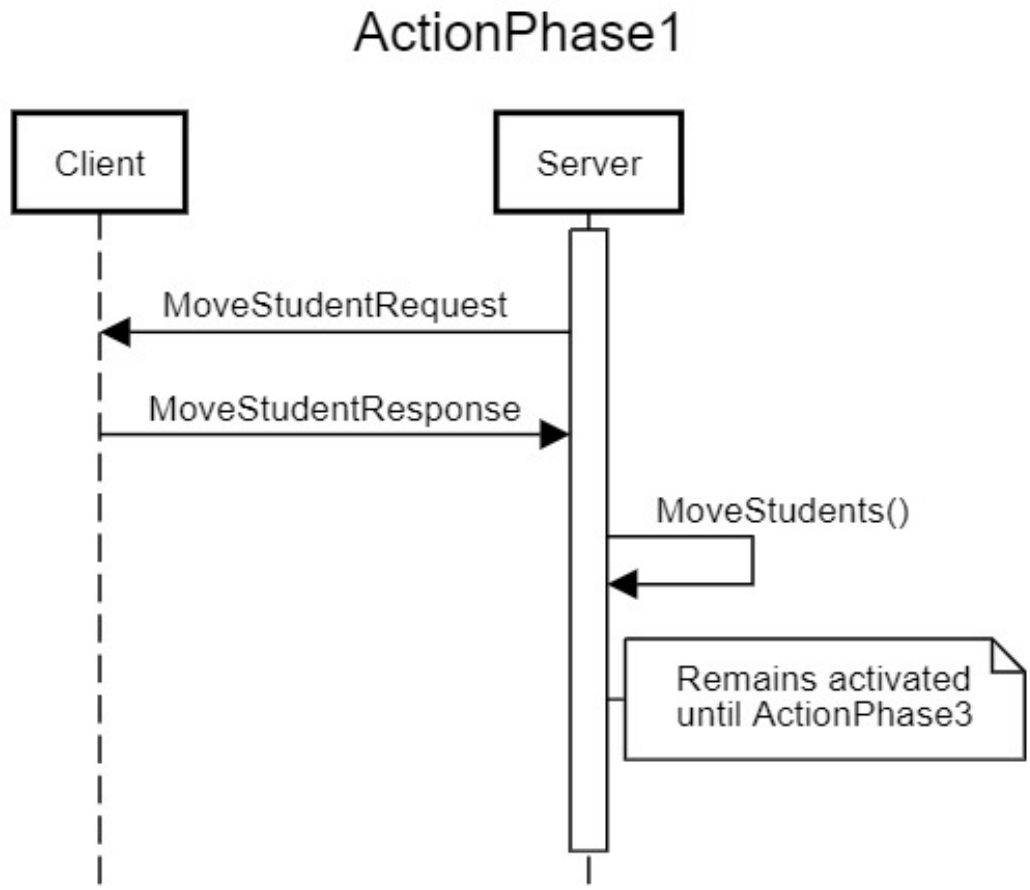
At the start of each turn, in the planning phase, the server calls the replenishCloud() method and then request to each player to choose an assistant using a synchronous message: ChooseAssistantRequest.
At the end of the planning phase, the server calls the calculatePlayerOrder() method to enqueue the players.



2.2 Action phase

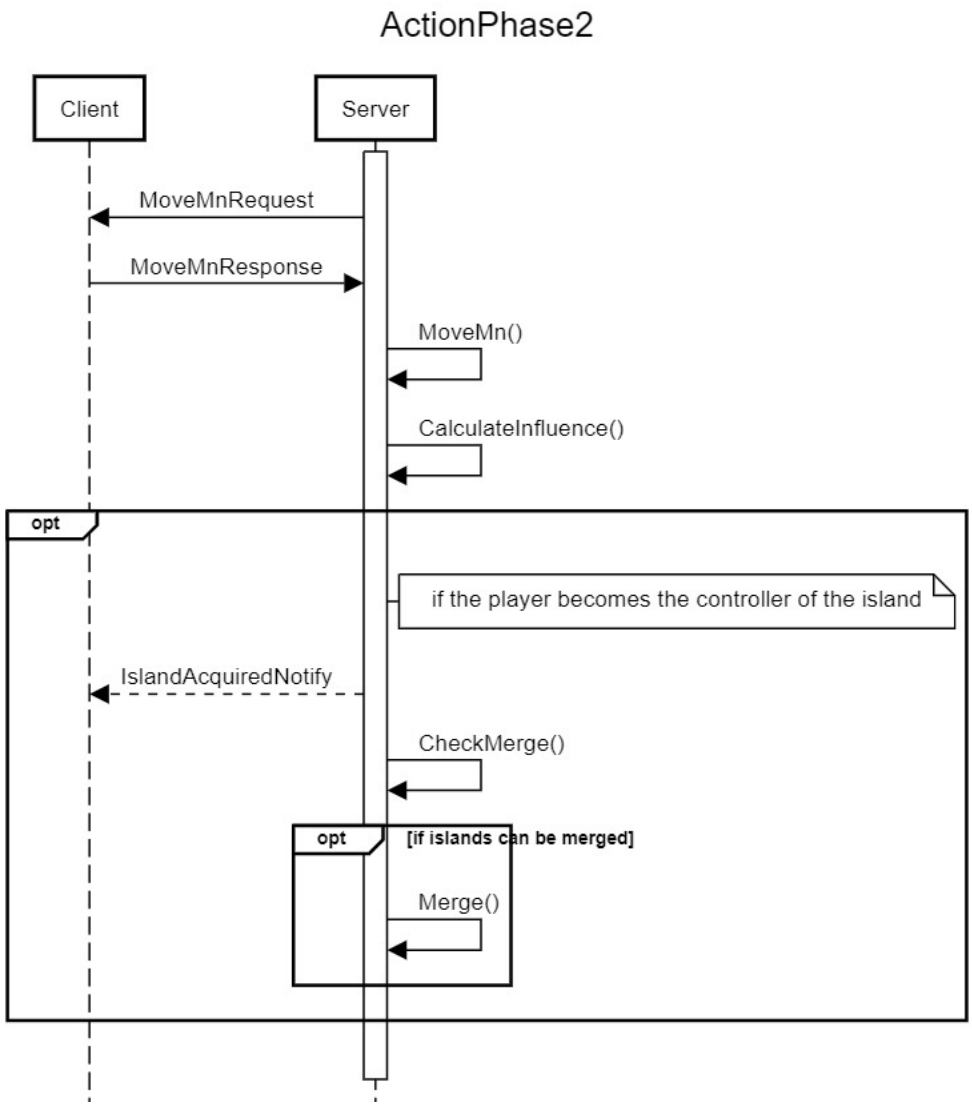
2.2.1 Action phase 1

In the action phase 1, the client is required by the server to choose three of his students and to say where to move them by a MoveStudentRequest message; after the client has sent his decision, the server uses the moveStudent(List<Student>) method to move the chosen students.



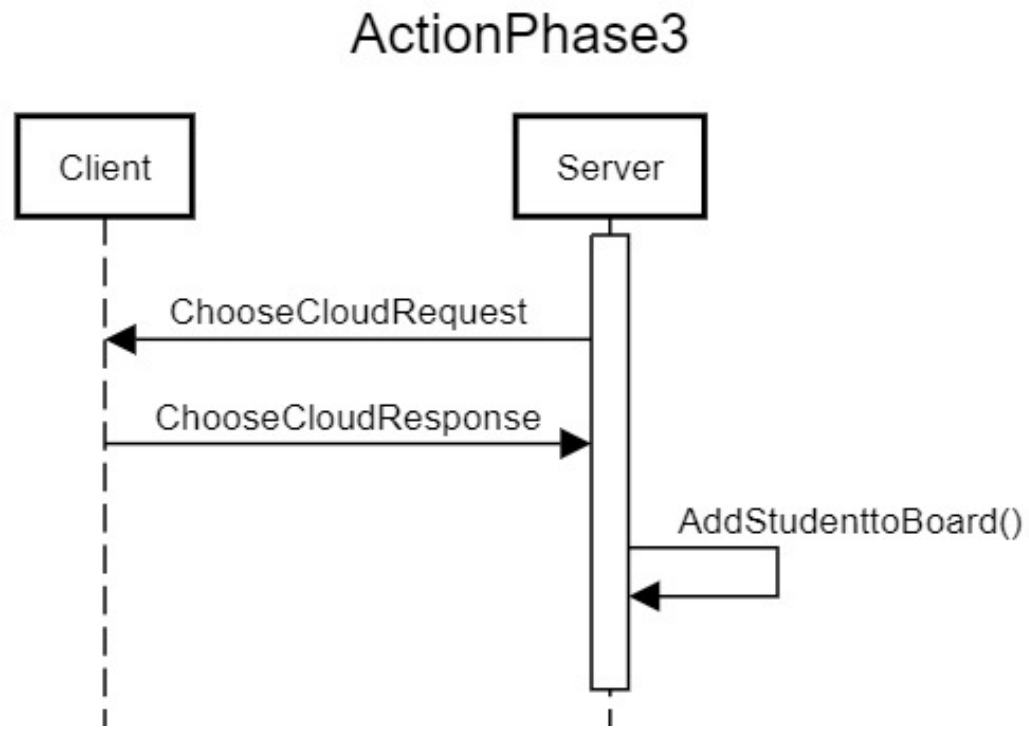
2.2.2 Action phase 2

In the action phase 2, the server sends a MoveMnRequest message and, after the client's response, utilizes the moveMn(int moves) method and then the calculateInfluence() method.
If the player has become the controller of the island where mother nature has been moved to, the server sends to the player an IslandAcquiredNotify and then checks if the closed islands can be merged into the island where mother nature is.
If it's possible the server merges the islands.



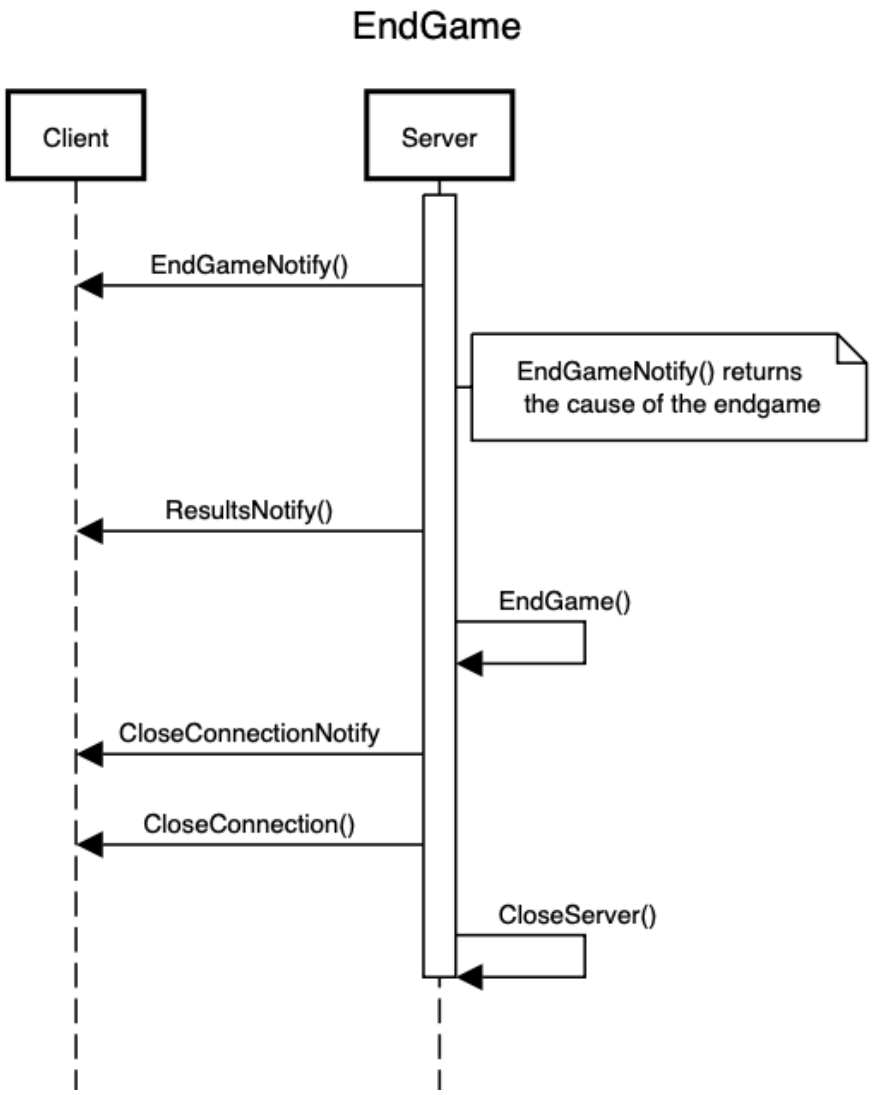
2.2.3 Action phase 3

At the end of the action phase, each player must choose a cloud, using a ChooseCloud message, with some students on and then, the server adds those students to the entrance of the player's board.



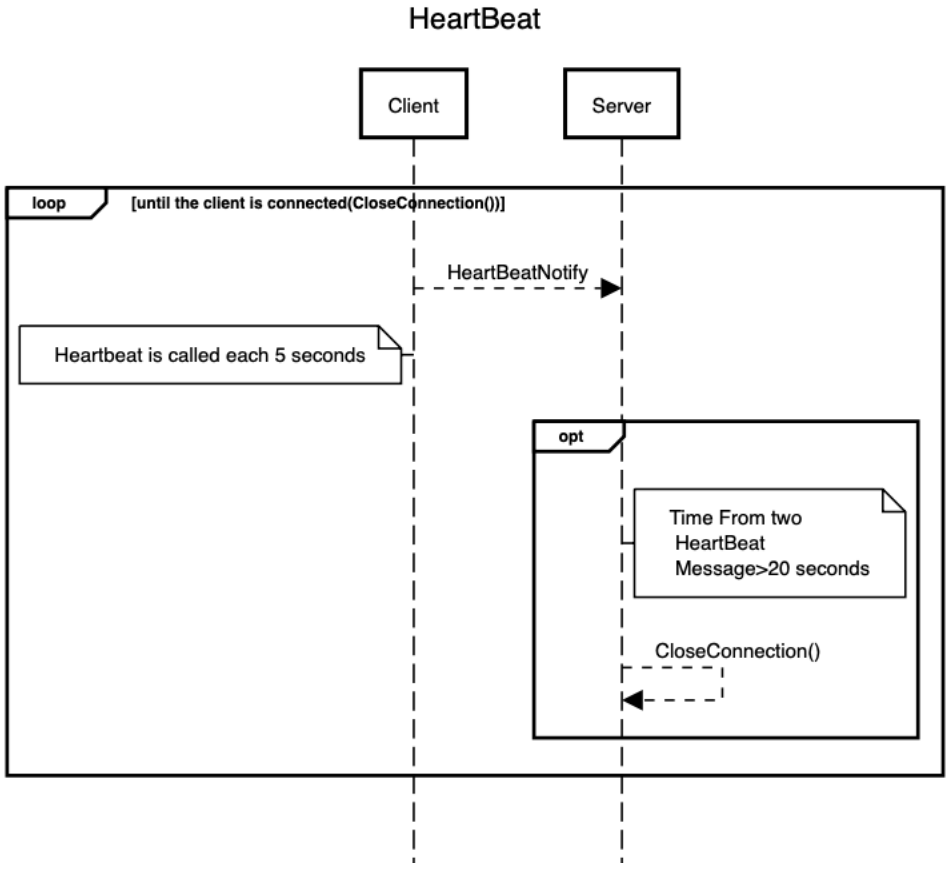
3 Endgame

If one of the conditions to end the game is meted, the server sends to each client an EndGameNotify (with the cause of the endgame) and a ResultNotify (with the results of the game).
Then the server calls the endGame() method, sends a CloseConnectionNotify, closes the connection with the clients and lastly closes the server (terminates).



4 Heartbeat

Since the connection is established, the player must send an HeartBeatNotify every 5 seconds to signalize to the server that the connection is still opened.
If the time between two HeartBeatNotify is bigger than 20 seconds (duration of the timeout timer), the server closes the connection with the client (it keeps the player's data if the player tries to reconnect).



5 General structure