

LOGIN PHASE:

client : loginEvent : String payload ("nickname");

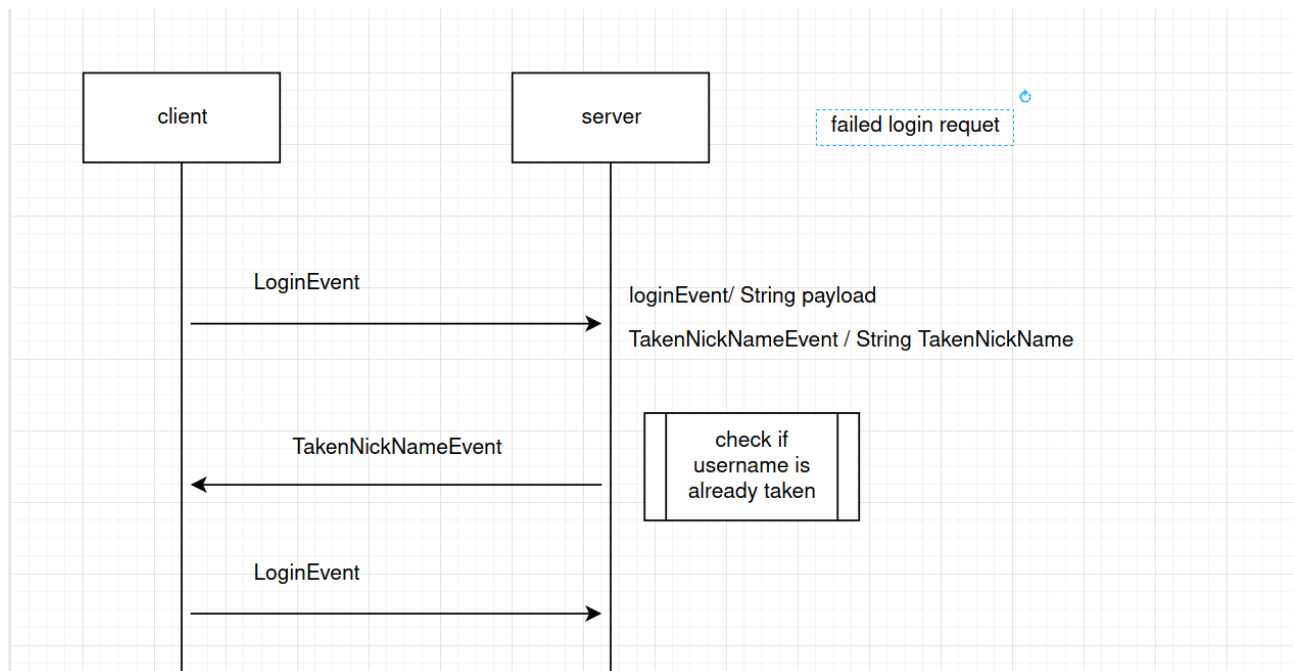
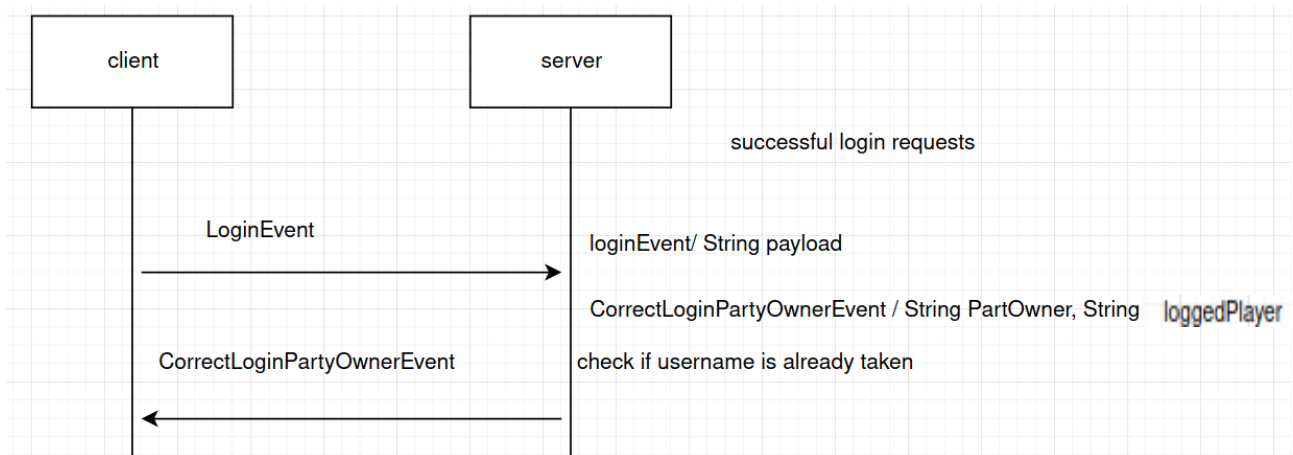
server : CorrectLoginPartyOwner : String partyOwner,String logged Player (if logged player is the first to log-in he is also the partyOwner).

Server :TakeNickNameEvent: String TakenNickNamePlayer.

When a client send LoginEvent to the server, the server responds in 3 ways : if this client is the first to request a log-in it become also the partyOwner.

If the client sends an existing NickName the server sends back an TakeNickNameEvent.

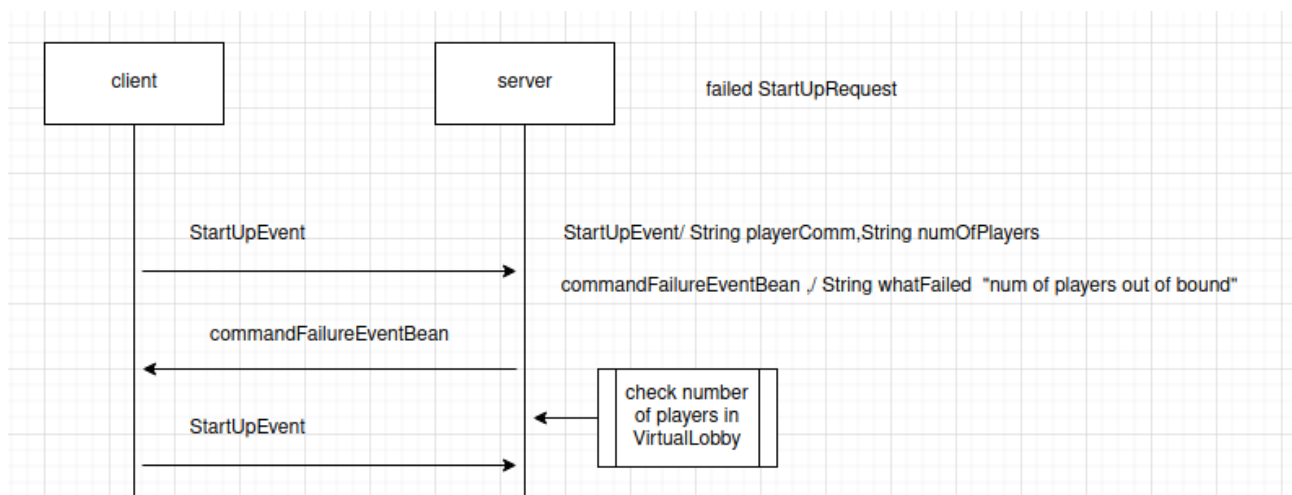
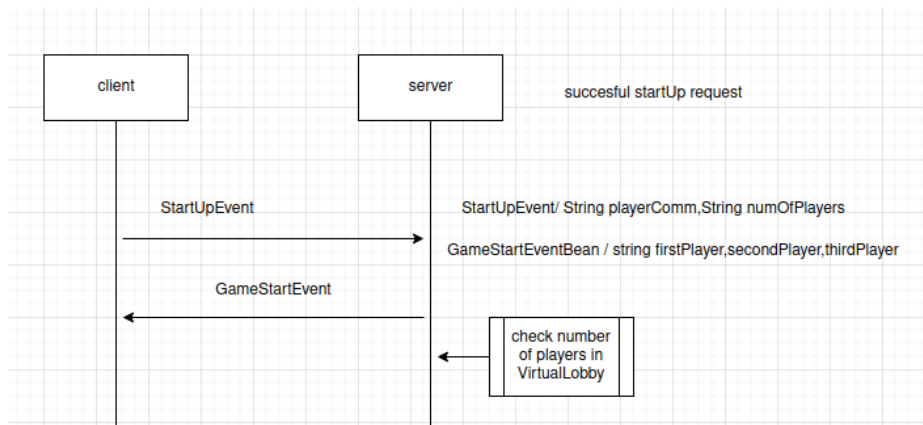
Else the server acknowledge the requests and send broadcast a CorrectLoginPartyOwnerEventBean notifying all the other player of who just logged in and who is the PartyOwner.



STARTING PHASE:

The first client that has logged in can make a StartUp request by creating a StartUpEvent client-side and including as payload the number of players to play with.

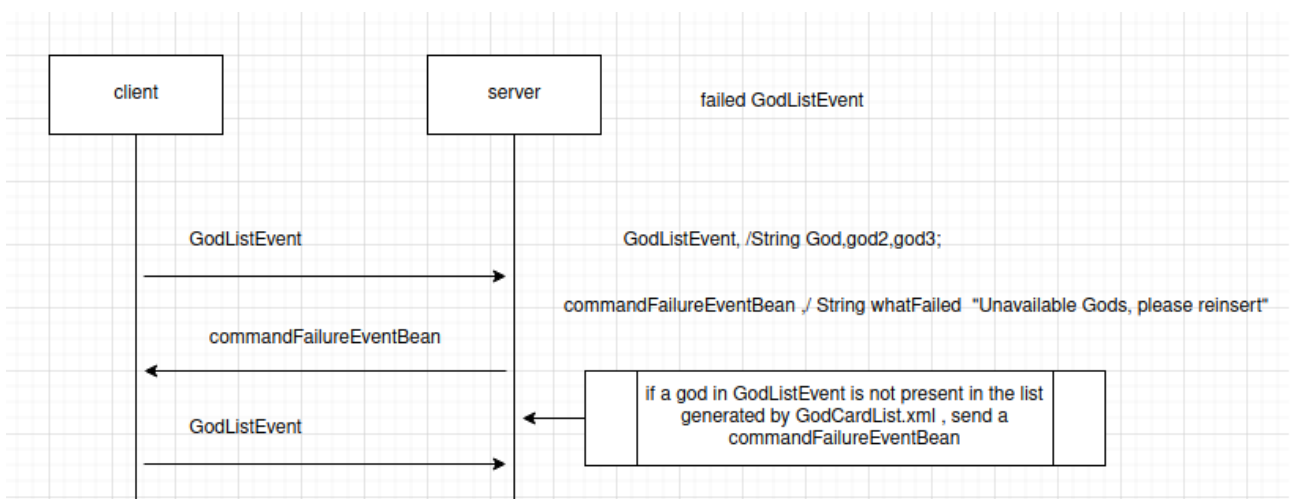
If number of player in the payload of the request is minor of the numbers of players currently in the lobby the server sends back a `CommandFailureEventBean` with payload "Num of player out of bound". Else the server sends `GameStartEventBean` with payload the names of who will play.



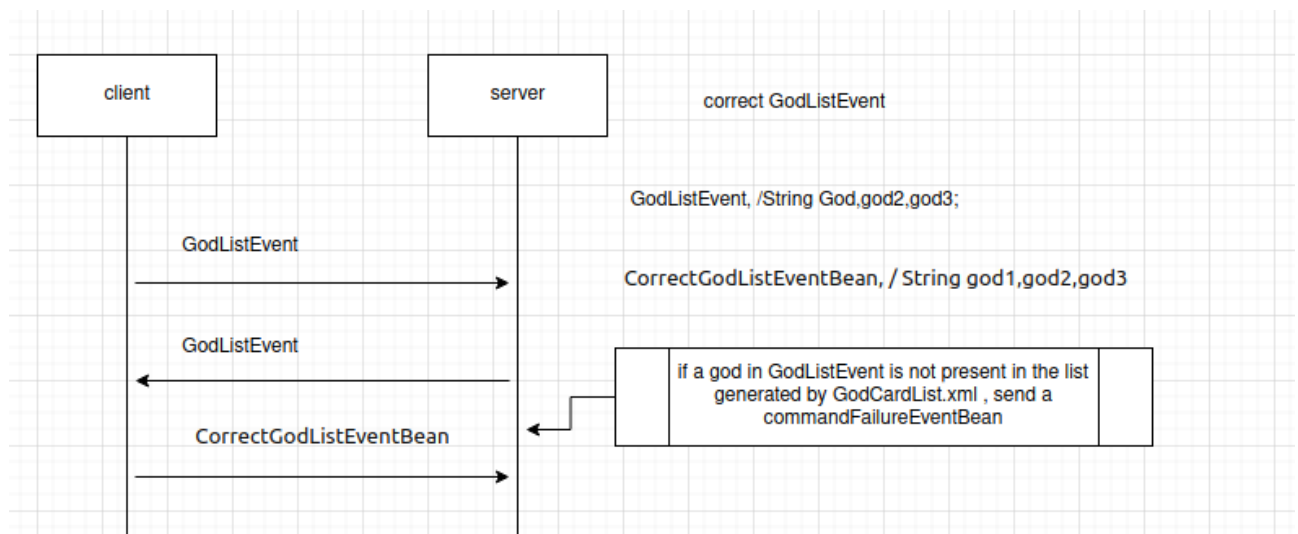
GOD LIST PHASE:

After the correct StartUp of the match, the PartyOwner sends a `GodListEvent` containing the name of 2/3 gods (depending on the previous choice mad during StartUp).

If `GodListEvent` contains unavailable gods or typos the server sends back a `CommandFailureEventBean` with payload "Unavailable gods, please reinsert".



If GodListEvent contains available gods the server sends back CorrectGodListEventBean with the list of selected gods to display on every client.

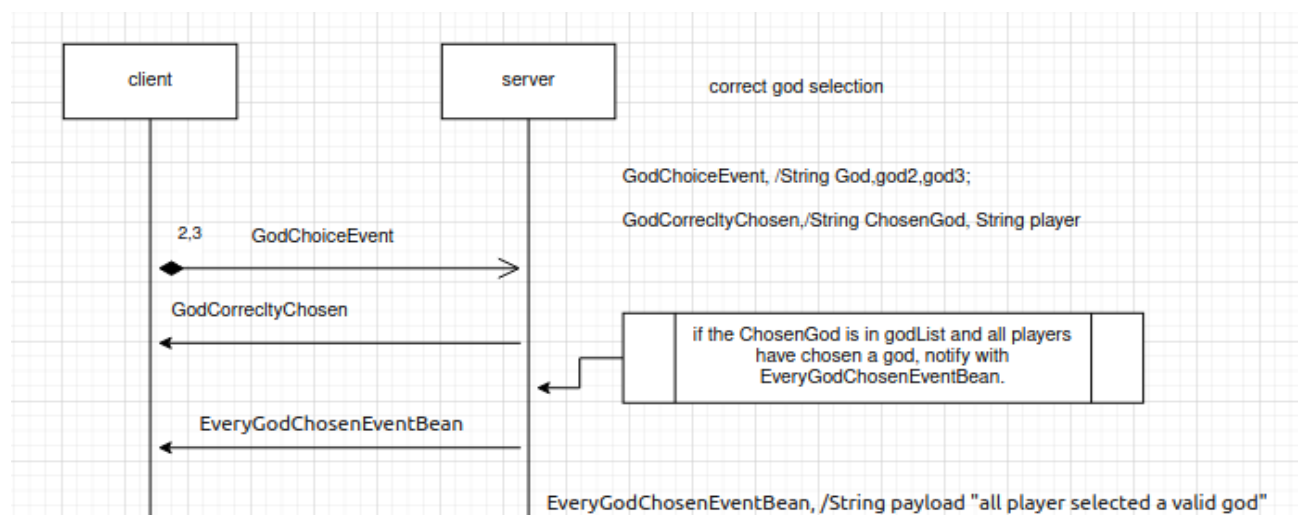


GOD CHOICE EVENT:

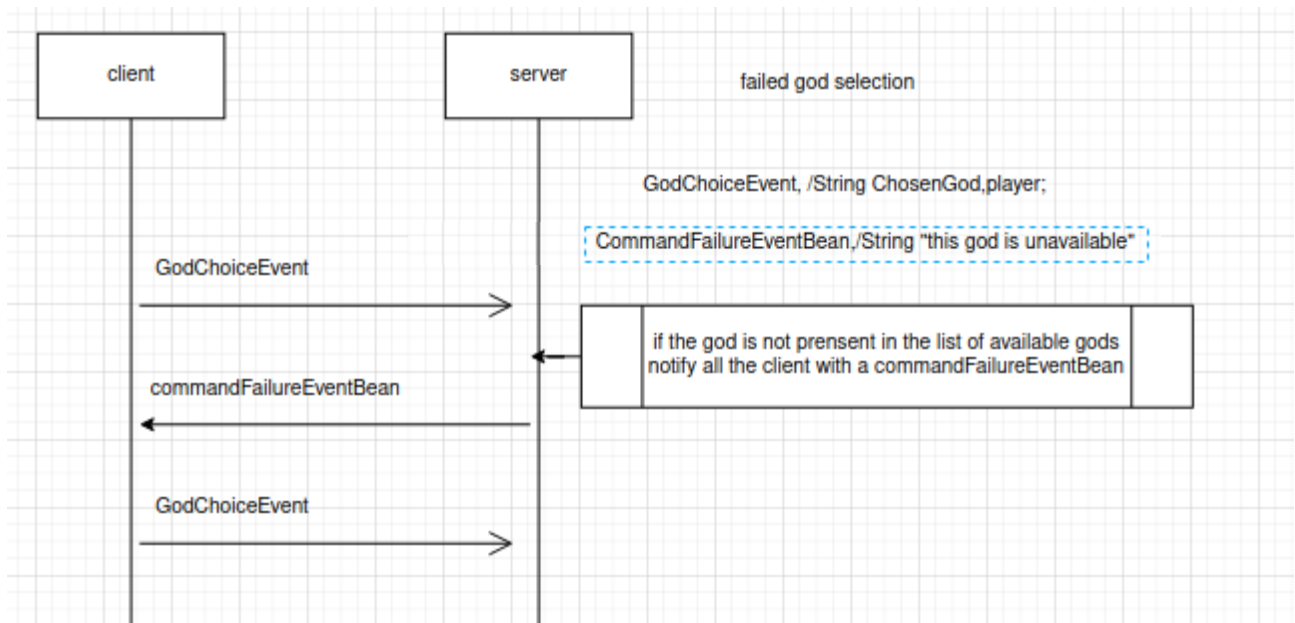
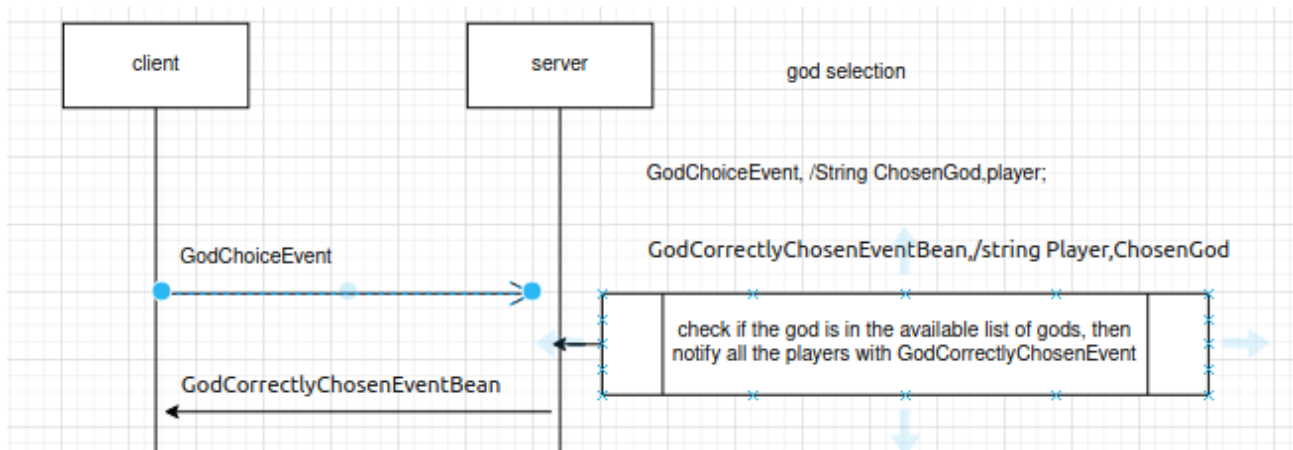
After an event of CorrectGodListEventBean the first player pick a god form the list by sending a GodChoiceEvent with payload the name of the chosen God.

If the god isn't in the list or contains typos the server sends back CommandFailureeventBean with payload "This god doesn't exists, reinsert another god form the list" and the client need to send another event.

After all client selected a god the server send an event EveryGodChosenEventBean notifying all players of the next game phase.

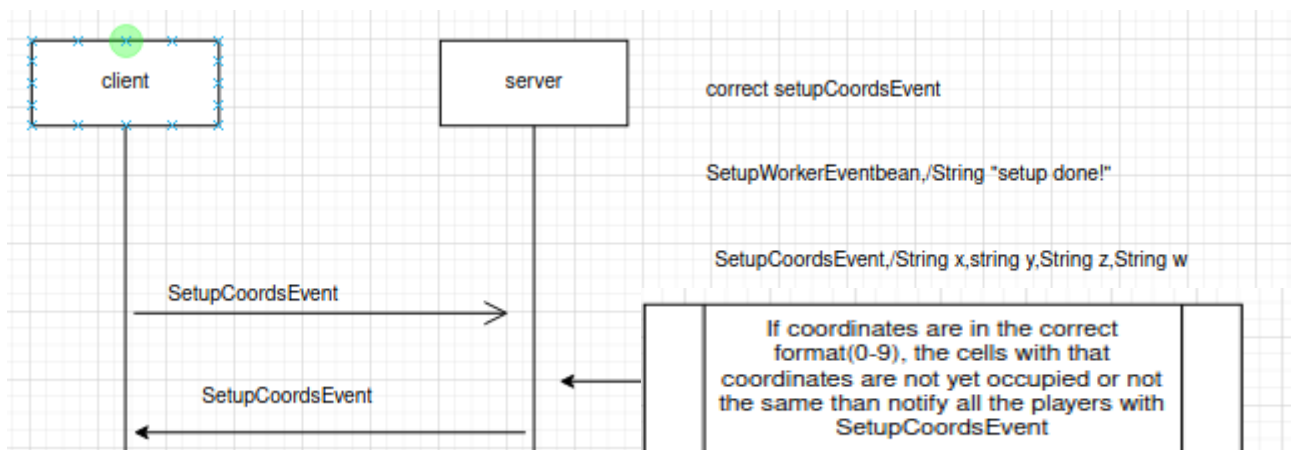


If the player select a valid god the server send back GodCorrectlyChosenEventBean with payload who did the choice and what god was taken.



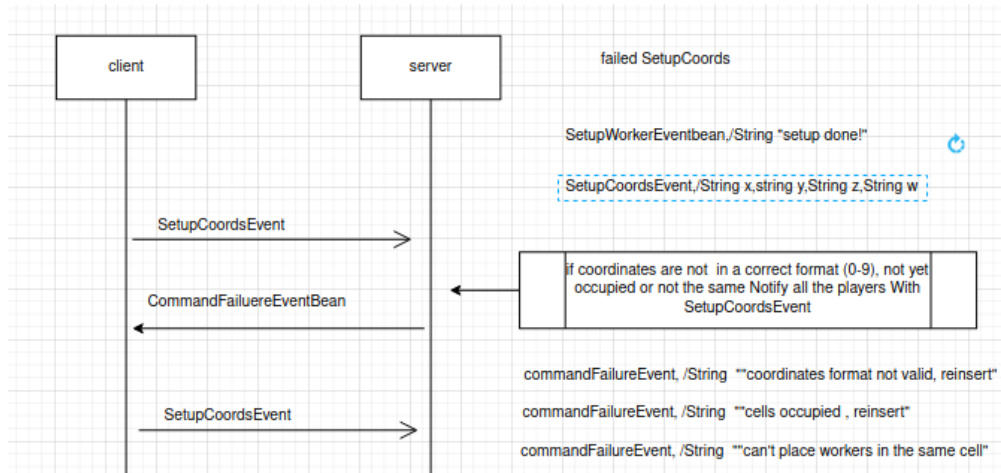
WORKER SETUP PHASE:

After the correct selection of all gods the first player have to set its two workers with a `SetupCoordsEvent` with payload string x, String y, String z,String w representing the position of its two workers.



If the coordinates are in wrong format (not 0-9 represented as Strings) or are occupied from other player's workers or are the same, the server notify all the clients with a CommandFailureEvent;

```
commandFailureEvent, /String ""coordinates format not valid, reinsert";
commandFailureEvent, /String ""cells occupied , reinsert";
commandFailureEvent, /String ""can't place workers in the same cell";
```



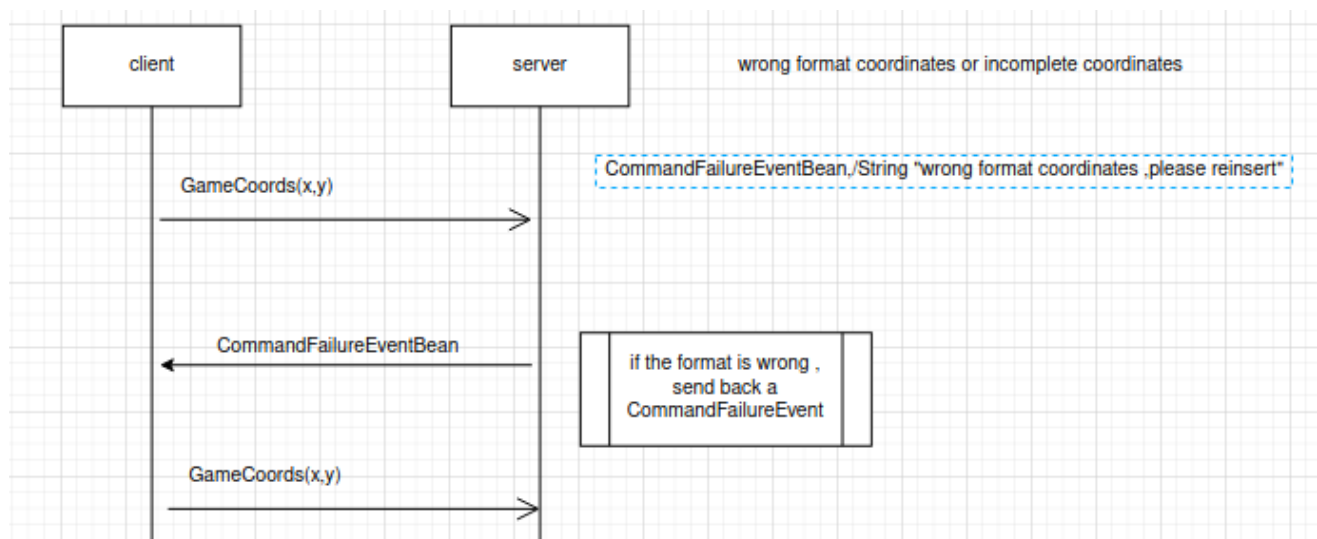
GAME COORDINATES PHASE:

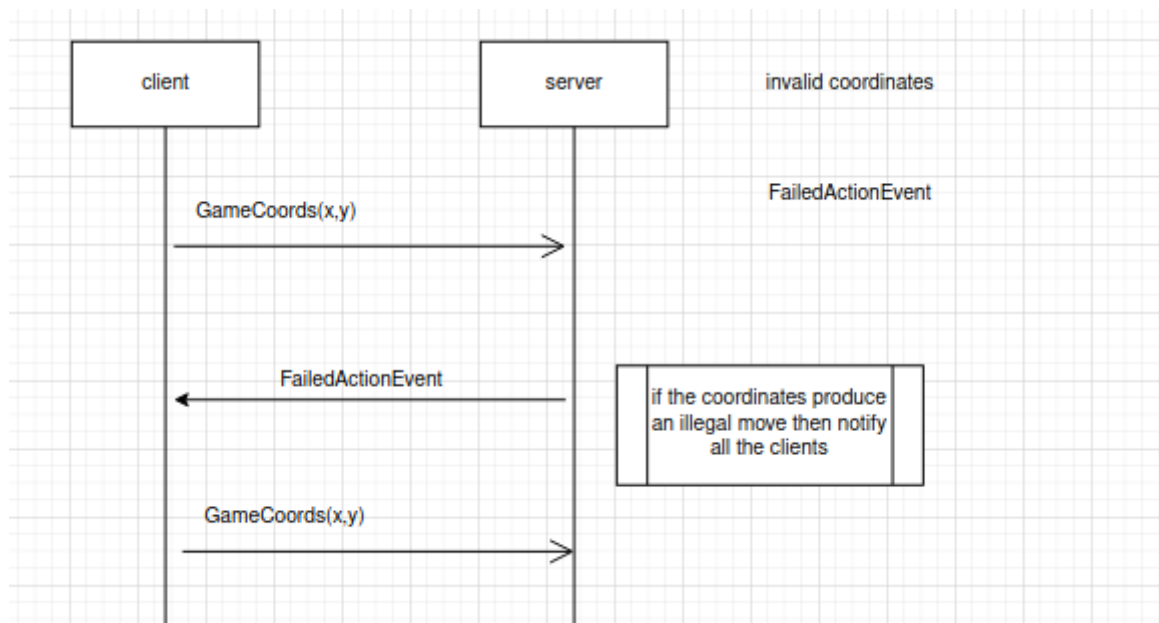
After the setup every player have to send to the server two coordinates representing a workers position, a worker movement or where a worker is going to build.

The Client send a GameCoordsEvent with payload two String , String x , String y representing Cartesian coordinates.

If the coordinates are in the wrong format (not 0-9) or incomplete(missing one coordinates of both two) the server sends back a FailedActionEvent.

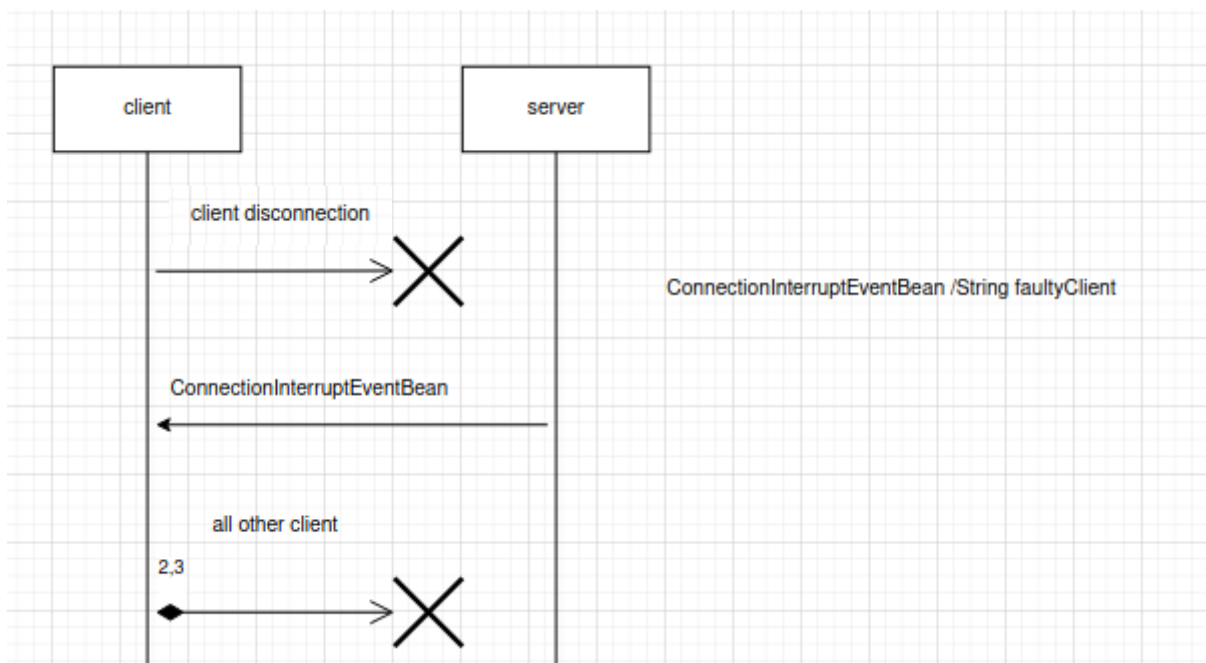
Else send back a waitingForActionEventBean which payload consists of all the available cells for the next power, noUpdate(), BooleanRequestEvent for Boolean requests , PlayerLostEventBean with the name of who lost, PlayerWonEventBean with the player that won and WorkerMovement for notifying the movement of a worker and Worker SelectionEventBean for the selection of a worker;





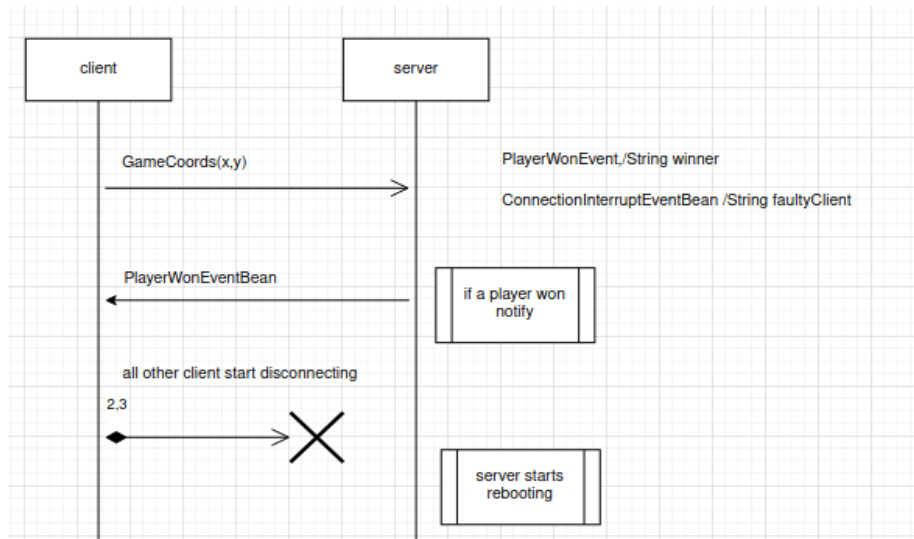
CLIENT DISCONNECTION:

If a client suddenly get disconnected from the server or decides to quit the game, the server notify all the other player in the lobby to quit and starts the process of rebooting.



PLAYER LOST,PLAYER WON:

If a player won the server notify all the clients of who is the winner and disconnect their socket one by one giving the ability to reboot and be ready to start a new game



If the match consists of only two player and a player loses instead of sending a `PlayerWonEvent` the server send a `PlayerLostEvent` that has the same exact behavior of `PlayerWonEvent`.

CONNECTION TO THE SERVER:

Clients sends requests to the server for connection , in the server accepts them(num of player is ≤ 3 or no exception thrown) than they are connected to the server, for joining in a room they have to do a Log-in request with A `LoginRequestEvent`.

