Sequence Diagrams

Gruppo PSP039

Contents

1 Introduction **2**

2 Connecting to a Game **2**
2.1 Join . . . 2
2.2 Create . . . 2
2.3 Reconnect . . . 2
2.3.1 Restore . . . 2

3 Player's Turn **5**
3.1 First Turn . . . 5
3.1.1 Choose Goal . . . 5
3.1.2 Place Starting . . . 6
3.2 Standard Turn . . . 7
3.2.1 Play . . . 7
3.2.2 Draw . . . 8
3.3 Player Turn Update . . . 12

4 End of Game **13**
4.1 Adjust Score . . . 13

1 Introduction

The project we now present will include the following Advanced Features: Persistence - Multiple Games - Connection resilience, in addition to the Standard Features of the Game.

2 Connecting to a Game

After a first ACK is sent in an attempt to establish a connection between Client and Server, the Player is asked which technology they would like to use throughout the Game, along with the preferred User Interface.

Since the Player is supposed to be automatically assigned to an existing game, the Server assumes that an existing Game already exists, therefore only allowing the player to pick its username, but not the Player count; only if no existing Game are present then a new Game is created and the Player can choose the number of Players to play with. For all these reasons connecting to a Game is a loop that only ends with a successful connection.

It is noted that the first thing a Player is asked to do is inserting a username in order to avoid creating or joining a Game with a username that isn't unique: in case of a duplicate the Client is asked if they want to reconnect to a previous Game.

2.1 Join

As previously stated, an exception that is accounted for is a Player choosing a username that is already present in the GameServer. Along with that, the Server also checks that a username is correctly typed only using alphabet letters and/or numbers. Otherwise, it keeps asking for a new username in a loop until the requirements are met.

2.2 Create

A Create Game attempt is launched only after the JoinStatus from JoinGame is set to Joining - meaning the username is an acceptable format, but no Games are present and the Player has manifested the desire to create a new game. Since Joining is a loop only exited when a successful join or create has occurred, in case of an unacceptable number of players typed, the Join Status will be "Connection Failed" therefore staying in the loop for a brand new attempt. Same goes if the Player decides they don't want to create a Game.

2.3 Reconnect

The reconnect option is only available after a duplicate username has been typed by the Client, where instead of simply joining the Player is asked if they want to reconnect to a previous Game. That's applicable only if there is at least one started Game in the Server, as in any other case the username will forcibly be unique.

It's important to note that when reconnecting, the old communication port will be asked both as confirmation that the rightful Player is joining and for an easy access to the Game from the Server. Afterwards, a new port will be assigned.

2.3.1 Restore

When the server fails and suddenly collapses, the Player will be notified immediately, so that each Player can attempt a Reconnection, which will only be successful once the Server is back up.

Join Status. The Join Status is essential to include other types of errors independent from the chosen username or the typed player count that could occur, during all phases of connecting to a Game; for example, there could be an unresponsive Server or an unknown cause of failed connection. After a successful attempt, the Join Status will become 'Joining', otherwise it will display 'Connection failed'.

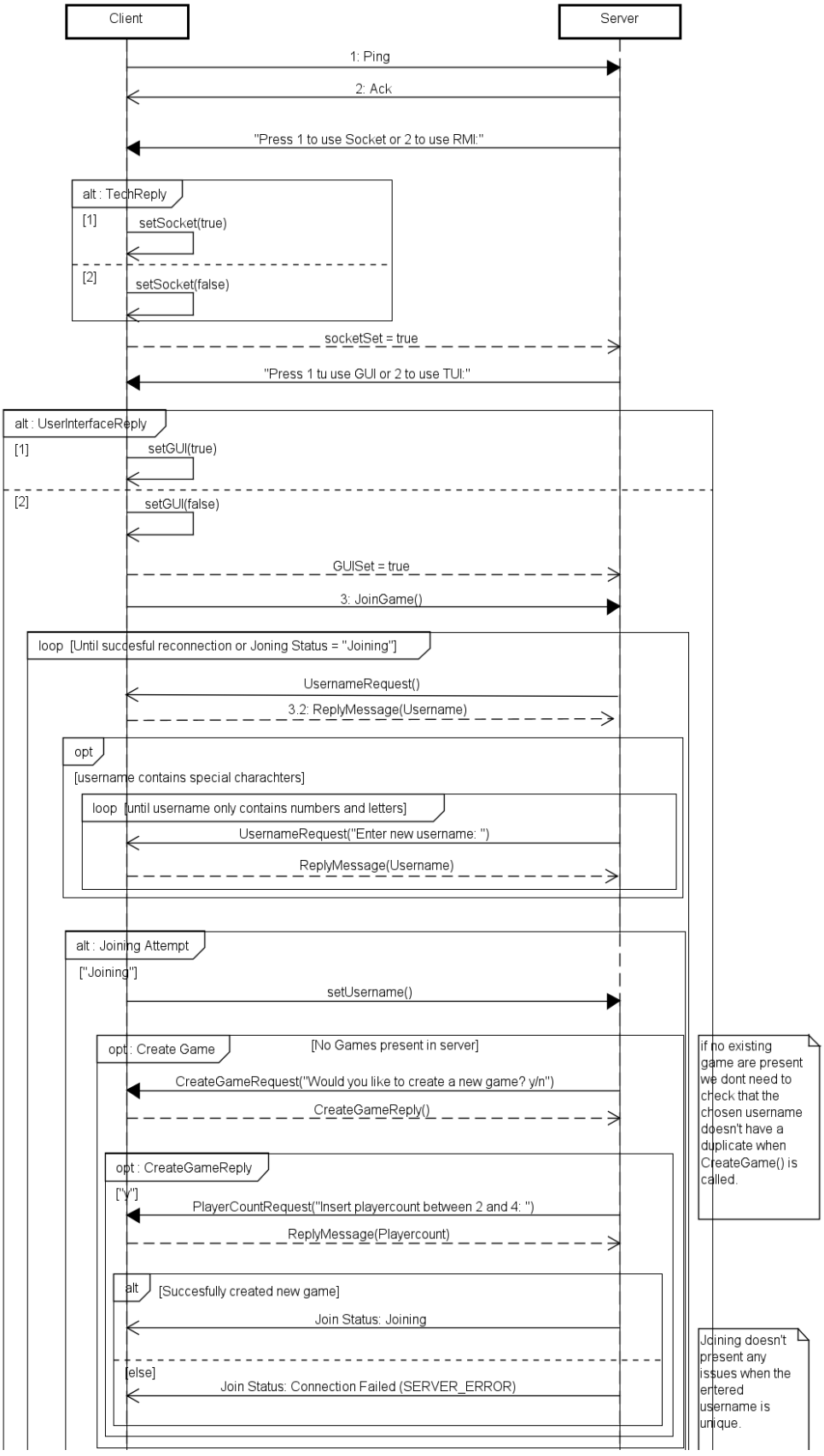

Figure 1: Create Game and Join Game.

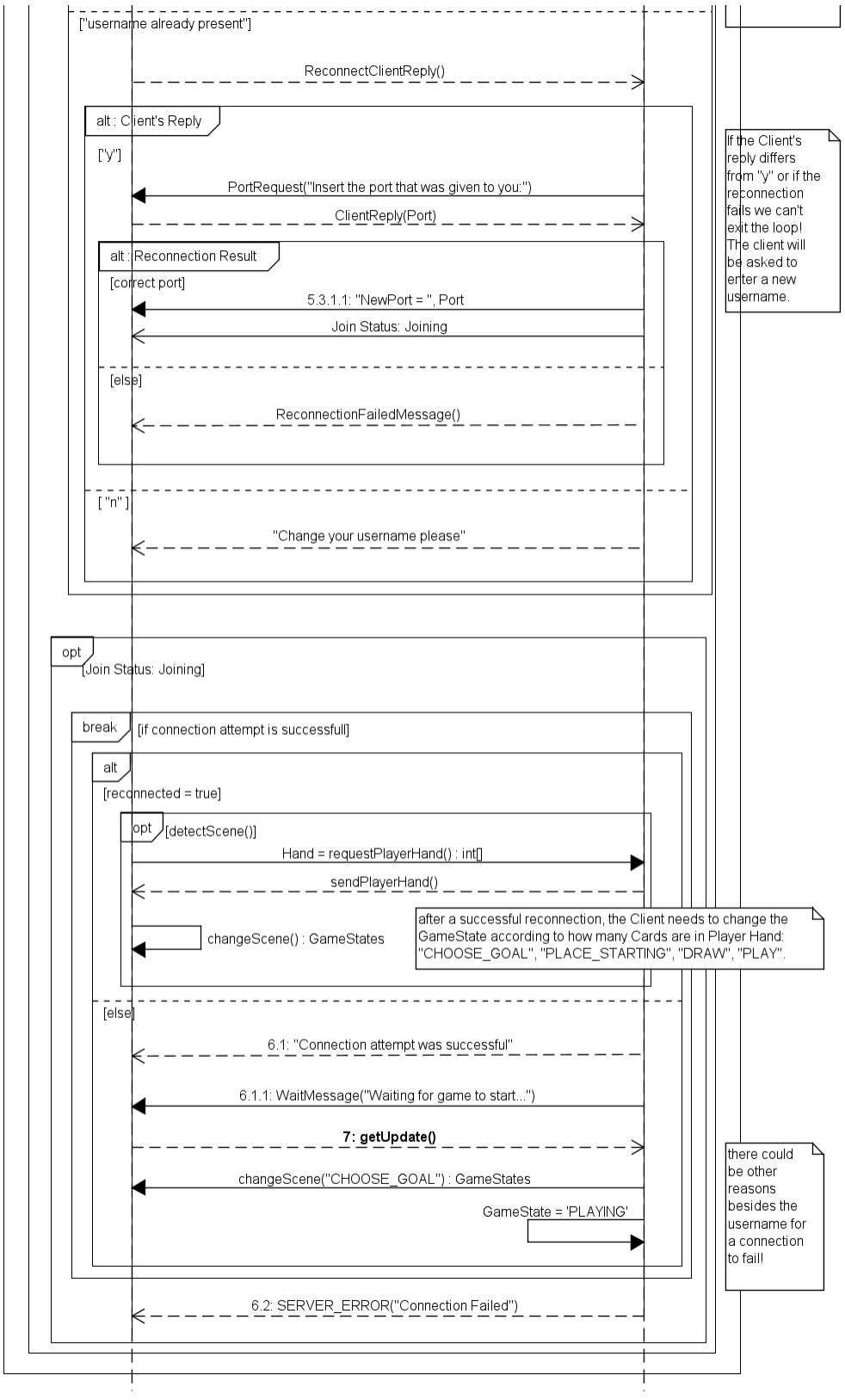

Figure 2: Reconnection and successful Join.

3 Player's Turn

At the start of the Game, and therefore within the First turn, all Cards, Common and Private, are randomly assigned to the Players. In accordance with the rules, the Common Cards will be the same for all Players while the Private Hand cards will differ. A Player's turn is a loop that ends with the end of the Game, as we consider not only the phases where the Player is active, but also the ones in which they are spectating another Player's Playboard. In parallel to the aforementioned options, it's taken into account the fact that a Client could disconnect, willingly or unwillingly, at any point, or similarly the Server could fail. In the first case a timer will allow the next Player to play without having to wait for the disconnected player to reconnect.

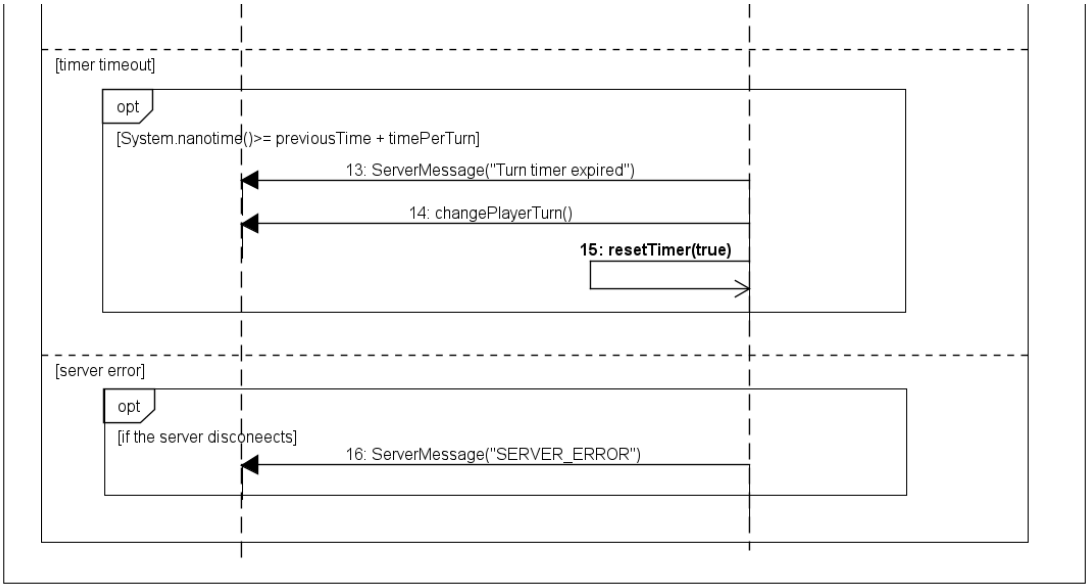

Figure 3: Disconnection and Server Error.

3.1 First Turn

The first turn has additional preliminary actions that are fundamental for the rest of the Game, that's the reason why a separate section is dedicated to it. The actions are choosing a personal Goal Card, from the 2 provided to the Player, and placing its assigned Goal Card, whether flipped or not flipped. Following these two operations the turn can proceed just like a standard turn.

3.1.1 Choose Goal

The Player can either choose the Goal Card in position 3 or the one in position 5 in his Hand of cards, and whichever one they choose it will be placed in position 3, while position 5 will remain empty for the rest of the Game.

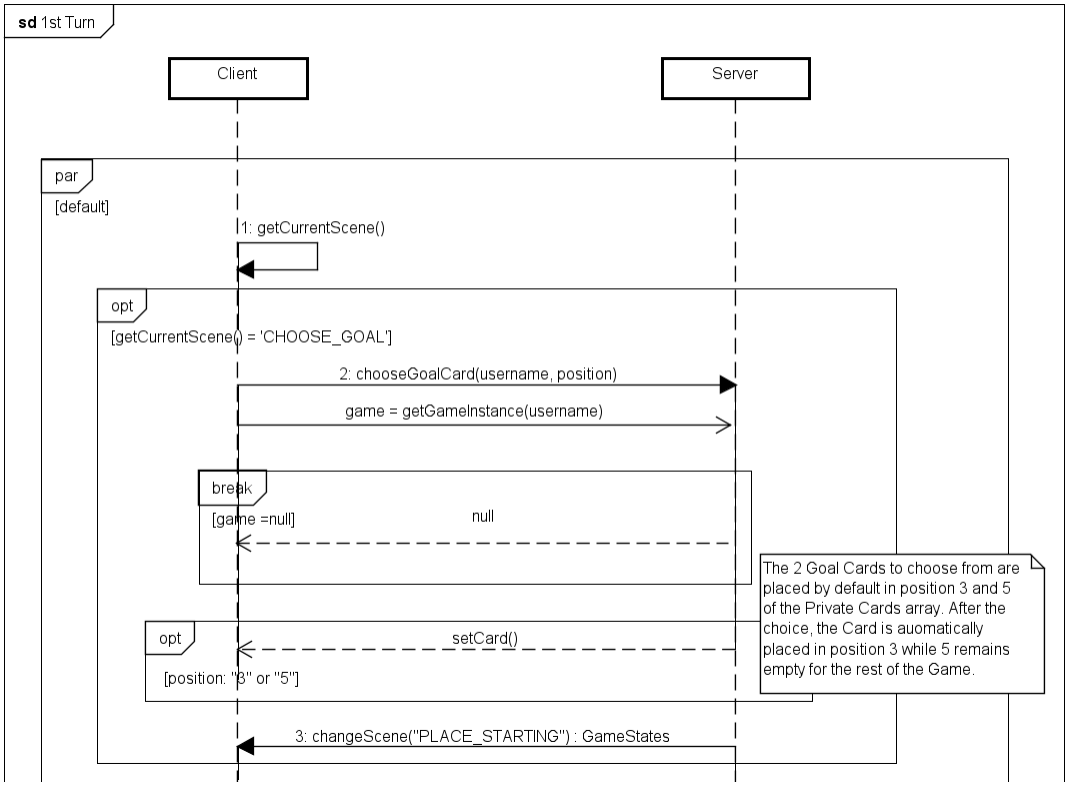

Figure 4: Choose Goal Card.

3.1.2 Place Starting

The assigned starting Card is to be placed first thing in the Game (only preceded by the Goal choice) because the rest of the Cards will be placed starting from its corners! The Card can be placed with the 'front' showing, therefore with the resources in the corners, or with the back showing, in which case the resources will be in the centre of the Card.

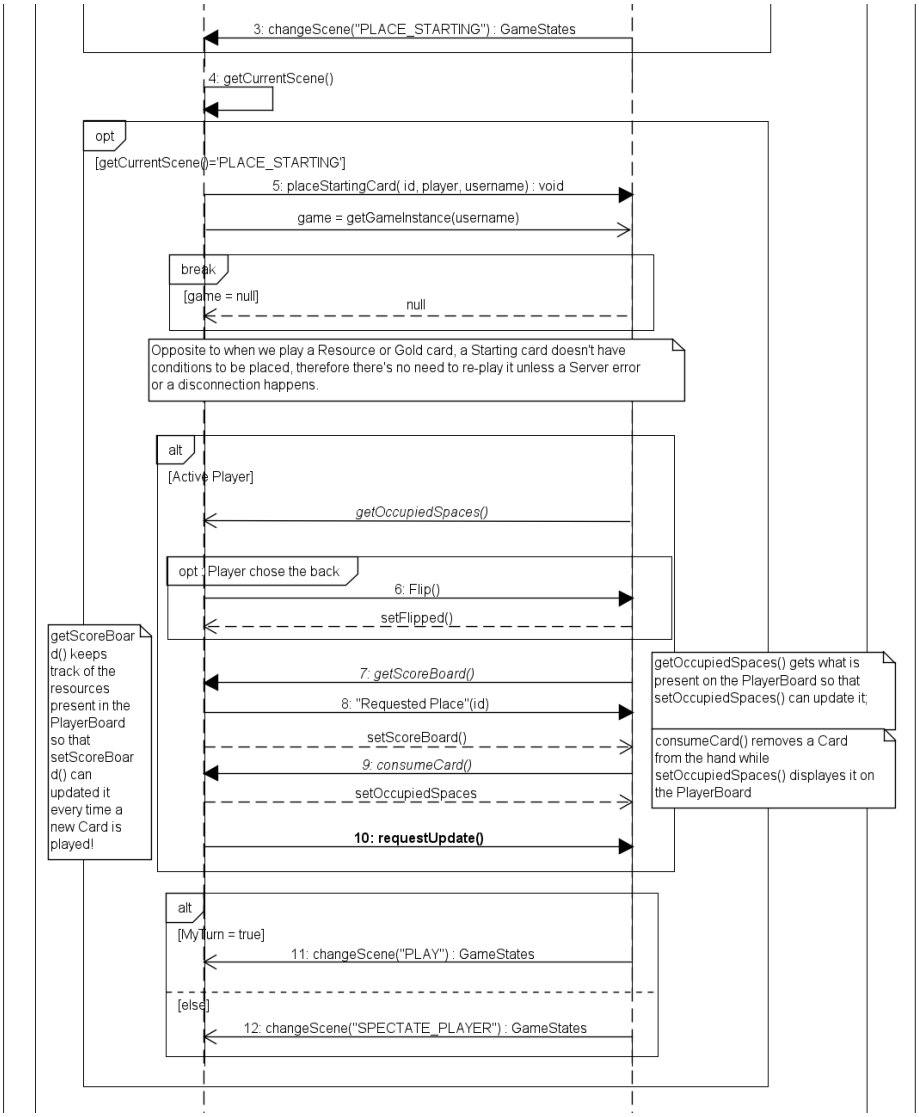

Figure 5: Place Starting Card.

3.2 Standard Turn

A standard active turn consists of a Play phase, and then a Draw phase, in this order.

3.2.1 Play

Different types of Cards can be placed, some of them require resources in order to be played, others give points to the Player according to the resources present on their board, some have both conditions simultaneously, some others simply just give points unrelated to the Board. That's why playing a Card also calls for a method that adjusts the score, meaning calculating new points, updating the resources count array and eventually, once 20 points are reached, triggering the end of the Game.

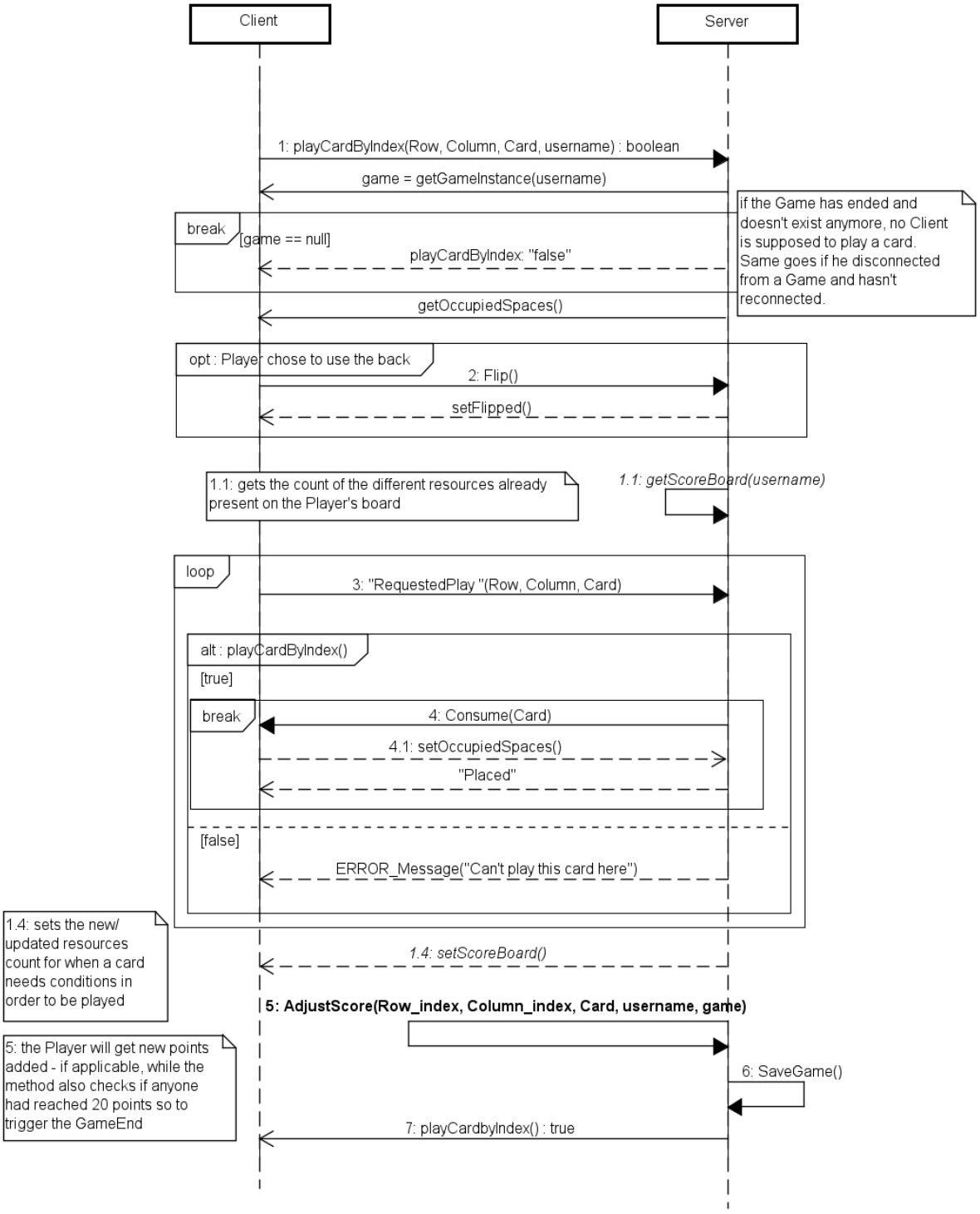

Figure 6: Play.

3.2.2 Draw

The Player can choose to draw from the Deck piles of Resource or Goal cards, or from 1 of the 4 visible Cards (2 Resources, 2 Goals) in accordance to the Game rules. The end of a drawing phase calls for the method that will change turns, or for a better understanding change who the active Player is, as drawing is the last action of a Player in a turn.

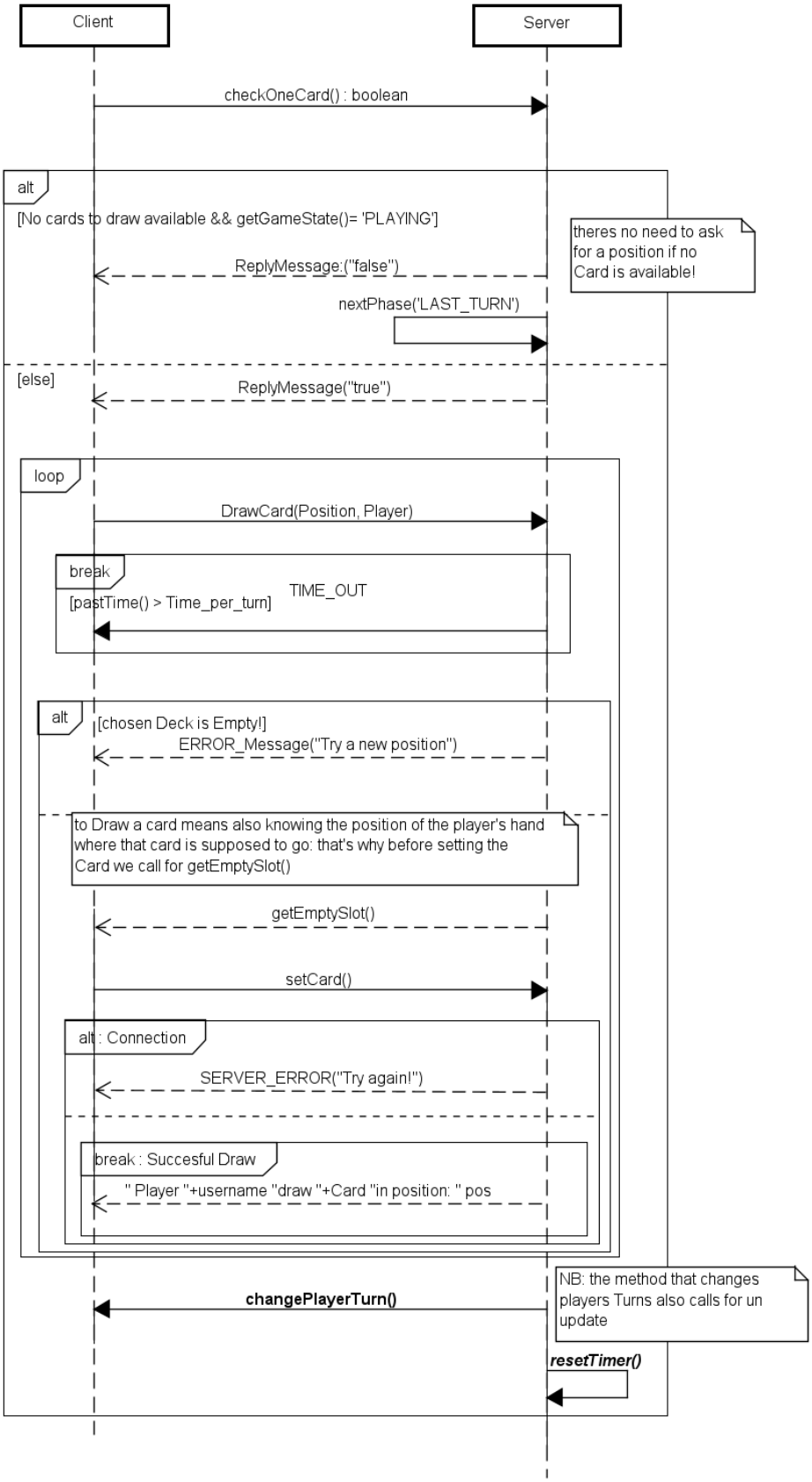

Figure 7: Draw.

The case of changePlayerTurn() Besides its normal function, *changePlayerTurn()* method also takes care of drawing randomly for a Player that has disconnected, so not to jeopardize the Game for all the other players. This occurrence is obviously subordinated to a disconnection taking place before a Player has the chance to draw, because otherwise no Empty Slot would be present in the Player Hand for the Playable Cards and there would be no need for a draw. Another task this method takes upon itself is drawing a new Card for the slot from which the previous player has drawn his last Card, if applicable. In the end, *changePlayerTurn()* also shuts the Game definitively after the last Player, when in 'LAST TURN' GameState, has concluded their turn.

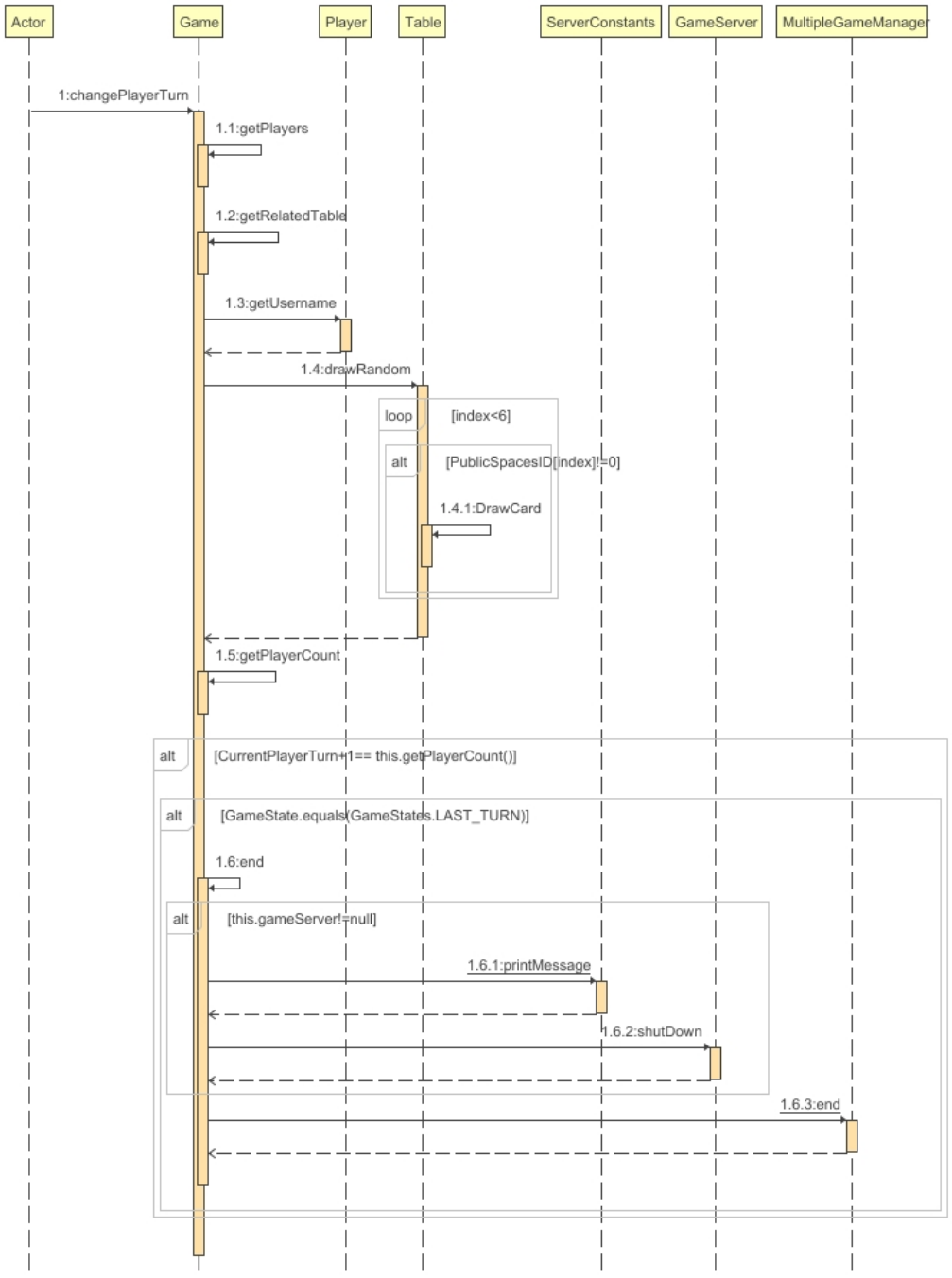

Figure 8: Automatic draw.

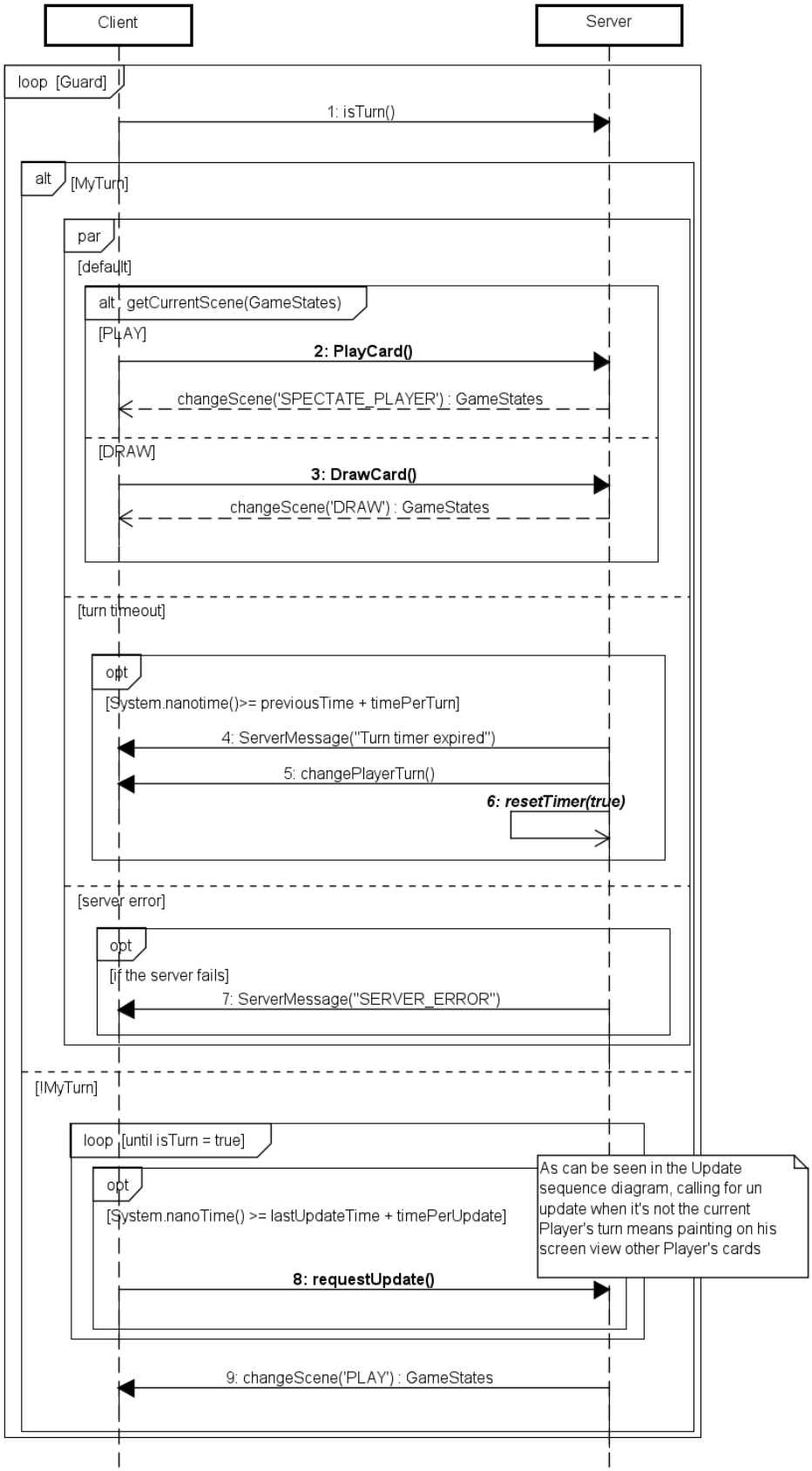

Figure 9: Turn Overview.

3.3 Player Turn Update

When un update request is called, a specific set of information is sent, but they differ depending on which player has the current turn. When spectating, a Player gets his screen updated with the PlayBoard of whoever is playing that turn.

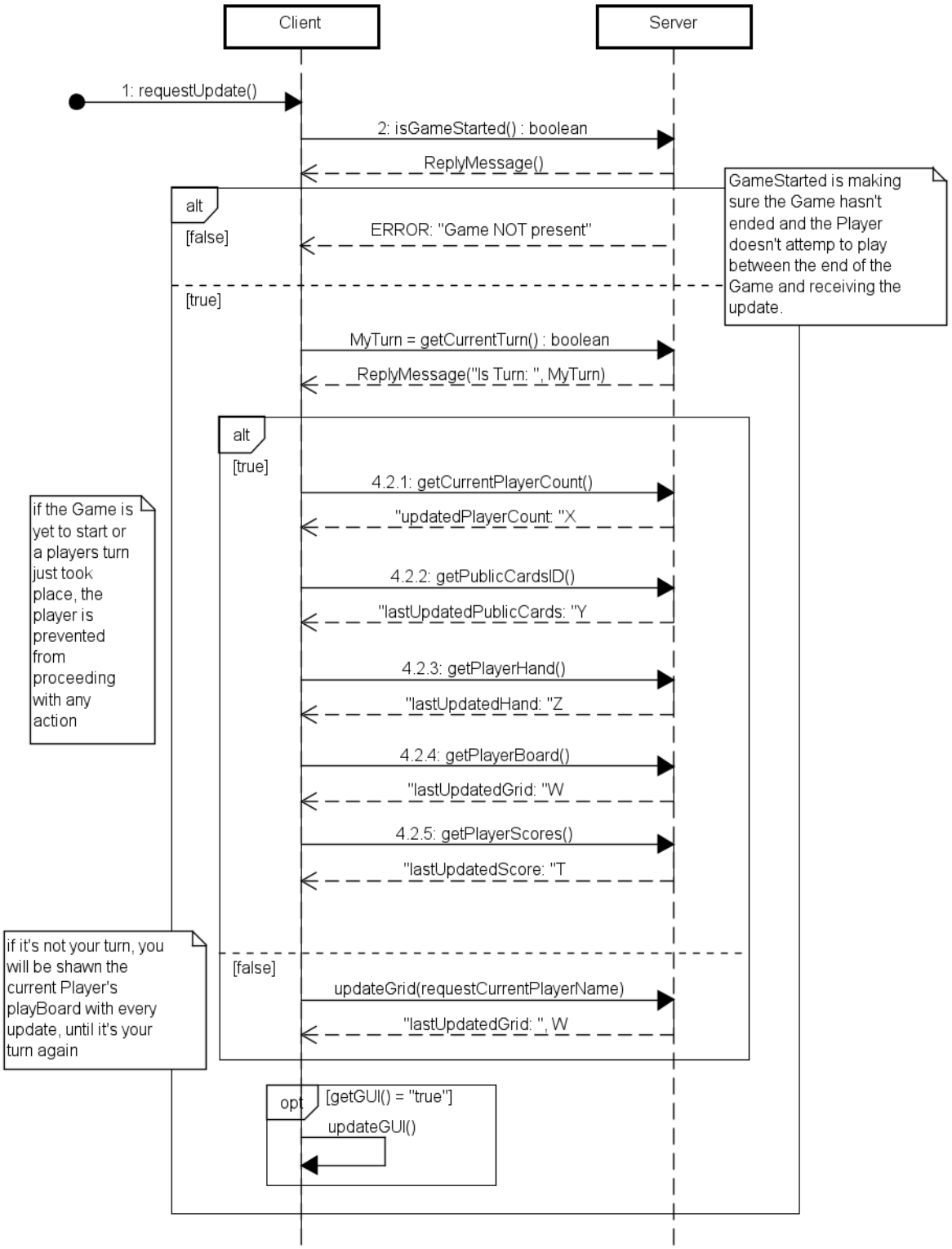

Figure 10: Turn Update Method.

4 End of Game

In addition to the Game ending being triggered when any player reaches 2o points, in accordance with the Game's rules, it is noted that the End of Game could be triggered by empty Decks. This aspect/option is taken care of in *changeplayerturn()*.

4.1 Adjust Score

AdjustScore() is essential to decide who the winner is: it makes sure the second to last turn, referred to in the diagrams as 'LAST TURN' where a Player actually reached 20 points is fully played to the last player in the player index, before proceeding with the actual last turn, where the turning goes back to being control by *changePlayerTurn*, so every player has played the same amount of Cards. Afterwards, the personal Goal points for each Player are added and the player will each know whether they have been decreed the winner or not!

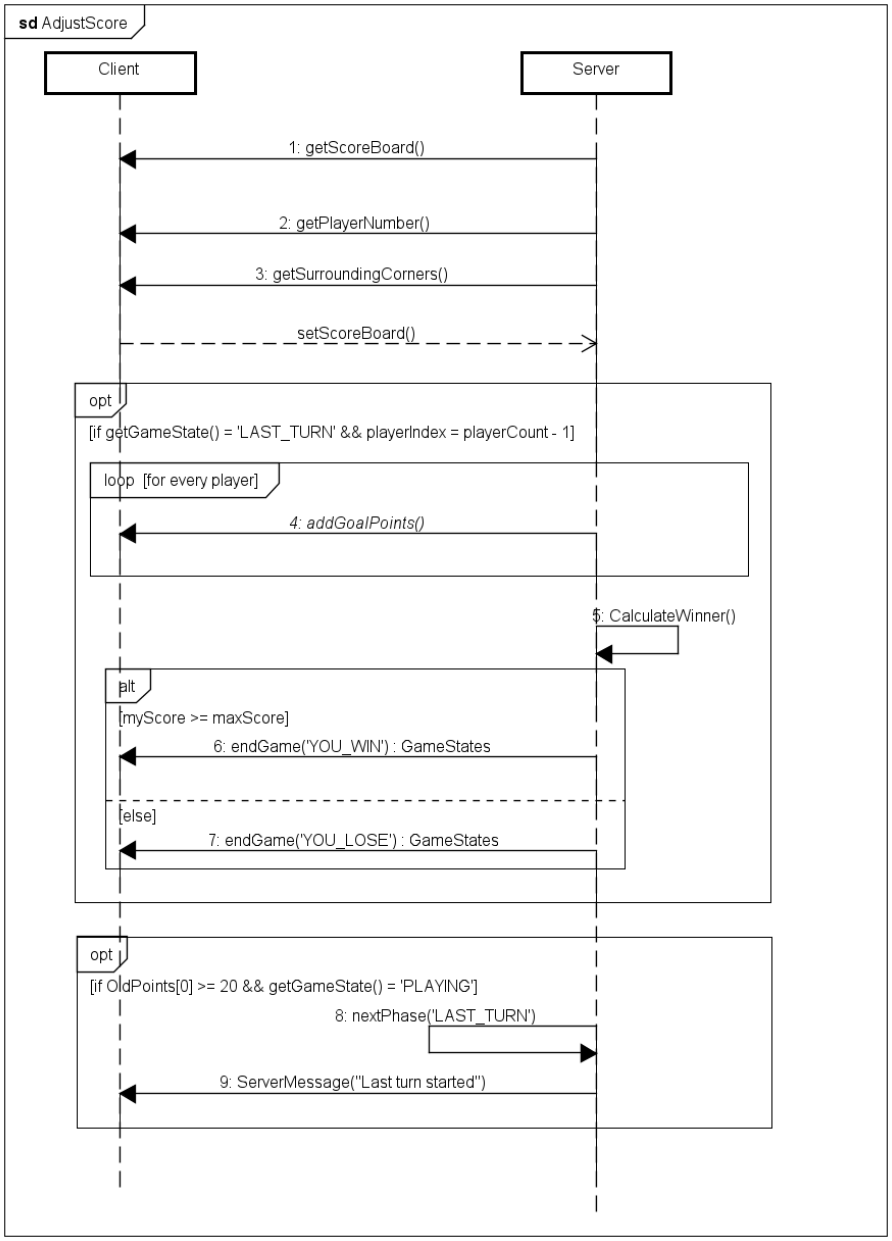

Figure 11: Calculate score.

The case of resetTimer: multiple time-outs *resetTimer()* is called at the end of an active turn no matter if the Player was able to conclude his actions before or after the timer; but in order to count how many consecutive time-outs occurred, *resetTimer* needs a boolean argument indicating whether the time-out comes from GameServer, in which case the Player wasn't actually able to complete the turn. After a number of consecutive timeouts equal to the number of Players, the Game is shut down and the last remaining Player automatically wins the Game.

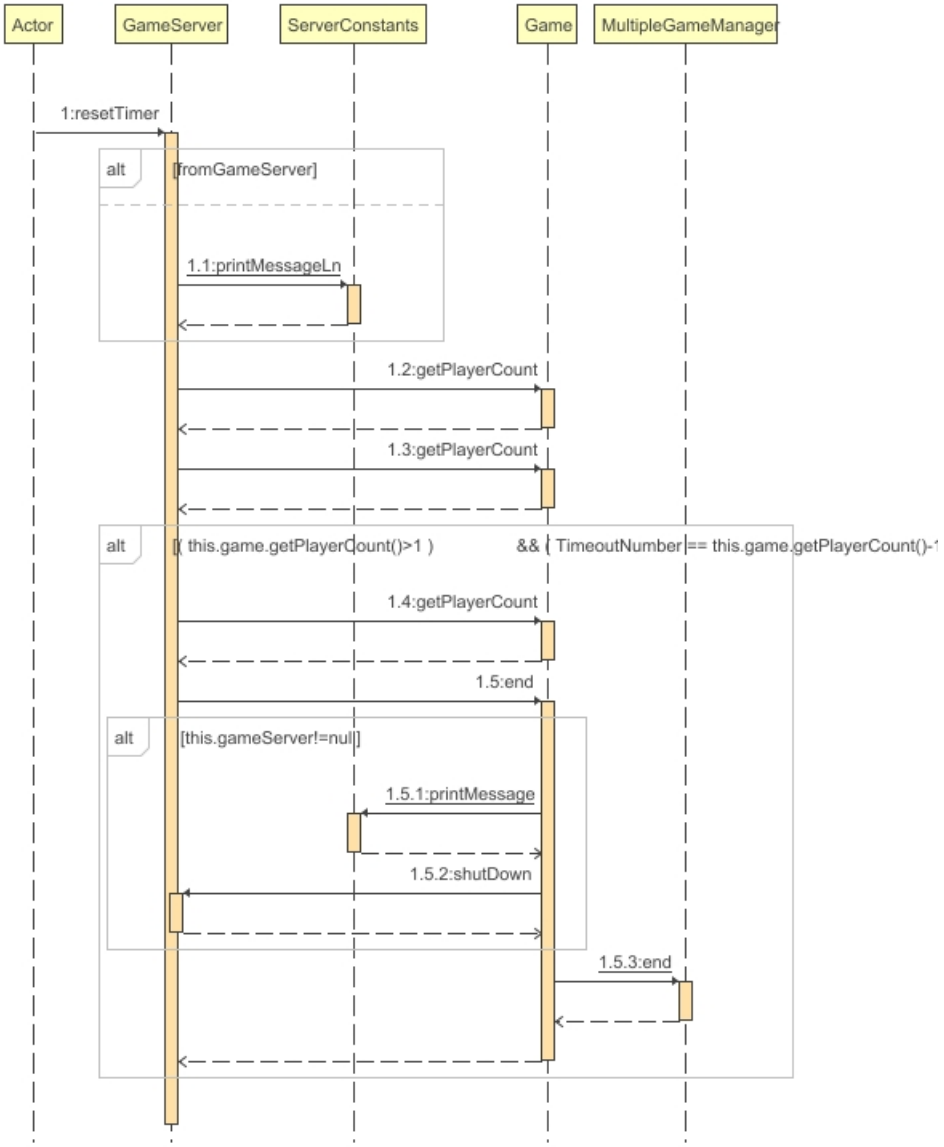

Figure 12: Multiple time-outs Exception.

References

Sequence Diagram plug-in for IntelliJ IDEA has been used to generate only the diagrams for *changeplayerturn()* and *adjustscore()*.