POLITECNICO
MILANO 1863

Protocol Desing

Codex Naturalis' Java Application

Ingegneria del software 1 - Prova Finale
Academic year 2023 - 2024

April 2024

Authors:
Lorenzo Galatea
Luca Lamperti
Alessandro Lodetti
Diego Quattrone

Chapter 1

Introduction

In this document we illustrate the specific of the Protocol Design for our Codex Naturalis' Java Application. Before proceeding, it is important to specify that we have chosen a Model-View-Controller (MVC) architecture, which is implemented through Server-Client communication. We will now proceed to illustrate three scenarios related to the use of our application. In particular, the messages exchanged between the Server and the Client are shown, assuming that the client implements everything related to the View. Once the communication is made to the Server, it will perform all the necessary operations to execute the desired operation, properly distributing roles and operations between the Model and the Controller. These internal operations are represented in the sequence diagrams as self-calls from the server.

Chapter 2

Sequence Diagrams

The first scenario (2.1) is the one in which a player wants to access a game. In particular, we have also specified the options related to the additional functionality of "resilience to disconnections".

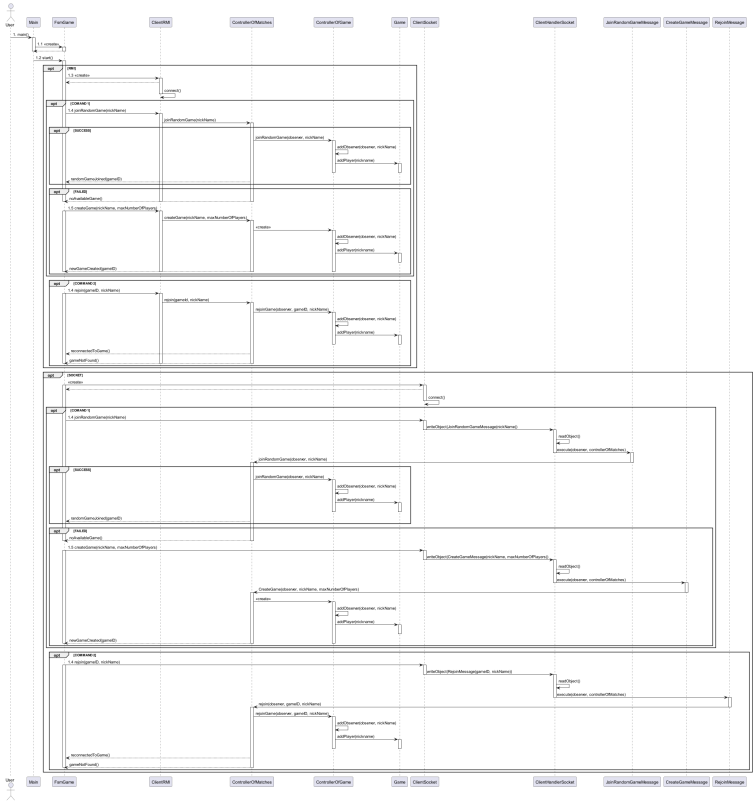

Figure 2.1:

After verifying the server's existence, the client attempts to join a game by entering a NickName. The server handles the exceptions related to the ID and attempting to join games that are already full, allowing access to a game. In case there aren't any games available, it asks for the number of players to create a new one. Additionally, an option for implementing persistence and resilience from disconnections has been developed, allowing a player to rejoin ongoing games using ID and NickName.

The second scenario (2.2) refers to the action of playing a card from the own hand.

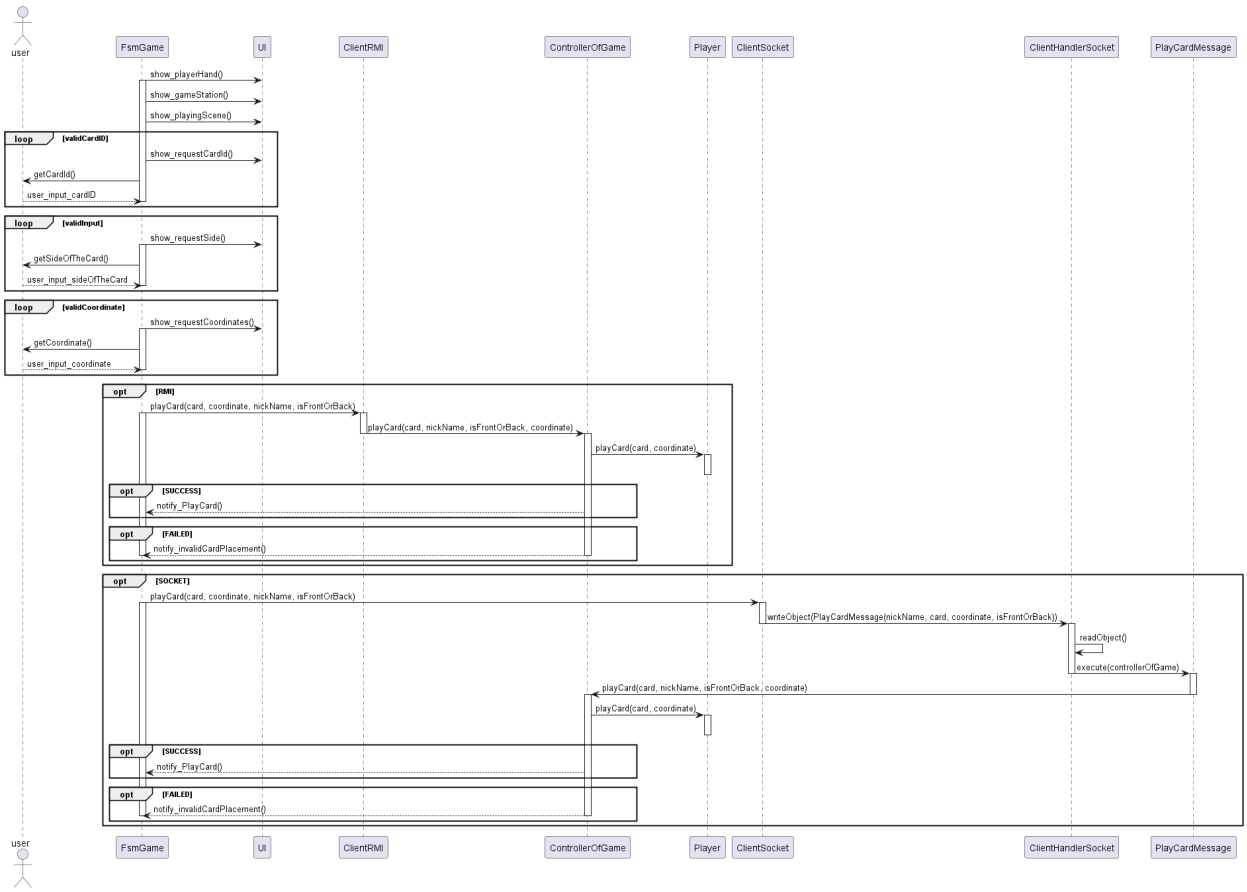

Figure 2.2:

The player selects the card the he wants to play and the coordinate to place it, then the client communicates it to the server. The server checks the validity of the card placement, thanks to the Model and Controller methods, and it returns a boolean. If the boolean is true, the Server updates GameStation and returns a message of success; if the boolean is false, a message is sent to the client and the player needs to choose a new card or placement.

The third scenario (2.3 represent the action of drawing a card from the table.

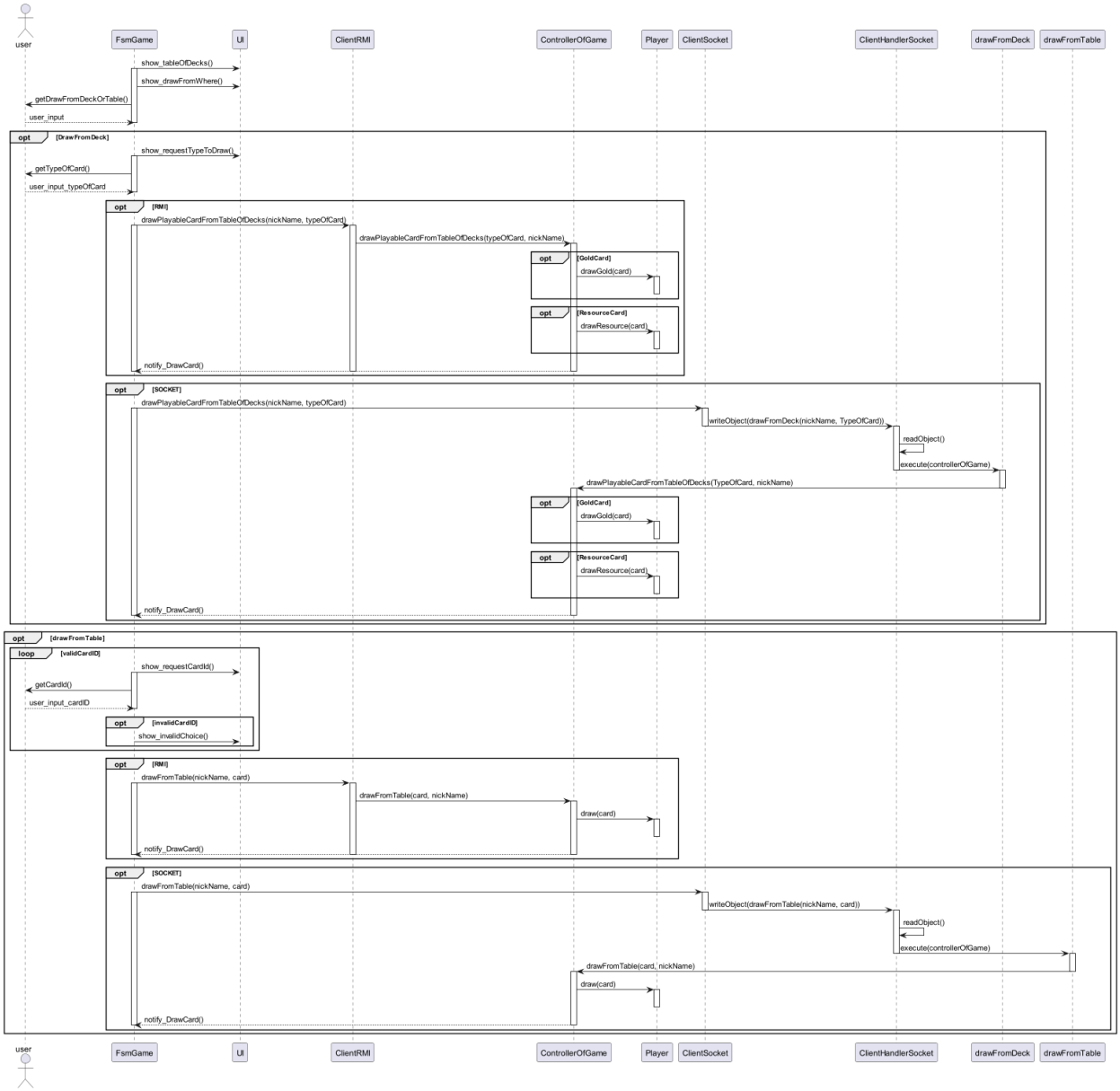

Figure 2.3:

The action of drawing a card can involve a card on the table or a deck. In the first case, the drawn card needs to be replaced with a card from a deck. In the second case, the player simply draws a card that is added to its hand.