Explanation of the sequence diagrams

Mirabella-Modolo-Patano-Paris

1. Access to the game

In the first step, the client decides which kind of connection he wants to establish: socket or RMI. For the first one, he sends a connection request to the server, clarifying IP and port. For the second one, the server establish a connection with a bind with the RMI-Registry, then the client can ask for the connection to the registry by using the IP and port. For both the procedures, the connection is established after these operations and the client can communicate directly with the server. The client sends his nickname and, until the username is different, the client should provide his authentication: if the username is already taken, the client receives a notification and the loop continues. In order to verify the nickname, the server checks the presence of the name using the controller. After the login, if the client is the first player, the server asks for the number of players: the client, which has a little logic, checks if the number is legit according to the rules and then he sends it. When the number of the players is reached, the server sends a beginning message, so the game can start. However, when the number is not reached, the server sends a message of waiting room.

2. Add a ship component

The construction of the ship is completely inside the client, so the client decides the X and Y coordinates. After that he can decides to rotate the component. In the end, the client has the opportunity to place the component or undo all his actions.

3. Drawing a ship component

During the time dedicated to create the own ship dashboard, the client decides to draw a covered or uncovered ship component. For the first choice, the client sends the request and the server handles the message using the controller. The controller calls the model in order to return the component. Received the component, the client has the alternative to place or book the component, invoking personal methods, because the ships are created in the client. The other choice is leaving the component, sending a message to the server: the server is handled by the controller, which invokes methods of the model in order to modify and get the uncovered components list. On the other hand, if the client wants an uncovered component, he sends the request, handled by the controller using the model. The model gives the component and a new list of uncovered components. The client receives his component and all the players can see the new list of uncovered components.