Explanation of the sequence diagrams

Mirabella-Modolo-Patano-Paris

1. Access to the game: "SequenceDiagram_AccessGame"

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, then he receives an ack as notification. 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 URL. For both the procedures, the connection is established after these operations and the client can communicate directly with the server. The server asks for a login request 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. 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. So the server communicate to the controller to create the game. After that, in any case (also if the client is not the first one), the server asks to the controller to add the client as a player to the game.

2. Add a ship component: "SequenceDiagram_AddShipComponent"

Until the position is not legal according to the rules of the game, the servers asks the client to decide the position of such component: the client sends the position and the server notifies it to the controller. If one of the two coordinates is wrong, the controller notifies it to the server and the server does the same to the client, bringing back to the loop. If the position is legit, the server communicates to the controller both the client and his component and, in the end, the controller notifies the success to the server and then the server does the same to the client.

3. Drawing a ship component: "Get_Ship_Component"

During the time dedicated to create the own ship dashboard, the client receives a request to draw a ship component. So the client asks for a ship component and the server sends the request for such client to the controller. The controller asks for an optional rotation of the component, but the client is in front of two different alternatives: he can leave the component, sending a notification to the server, which sends it as well to the controller. Otherwise, the client can take the component and he decides to put it on the ship or to reserve it. In the first case, he can ask, passing through the server, for an optional clockwise or counter clockwise rotation. In the second case, a reserve request goes from the client to the server, reaching the controller.