

Explanation of the sequence diagrams

Mirabella-Modolo-Patano-Paris

1. Access_Game_Diagram

The client decides using the UI (TUI or GUI) which kind of connection to establish, calling the client interface and giving IP, port and a boolean for the connection type. For socket, it is opened a socket with a thread for receiving the different messages, for RMI the client is bounded with the server via RMI registry. The client interface calls the “asking nickname” method of the UI interface and then it sends the login request to the server: for socket connection, the messages are serialized and sent to the server, which calls a handling method in the controller for message dispatching with visitor pattern. In case of RMI, it is called a “send message” method which dispatches the different messages with RMI Client Sender visitor class, calling different methods for each message on the server. The server methods, according to the state of the game, call directly the controller methods, differently from socket that uses a dispatching method in the controller. After the login request, the server sends the login response, telling if the connection is well established and if the name is unique: in case of a nickname repetition the client has to send another nickname. When the client is the first player, the server asks for the number of players of the current game: with socket connection serializing the message and sending it, instead with RMI dispatching the message with a RMI Server to Client visitor, which calls different methods for each message on the client. The client methods call directly the client interface methods in order to interact with the player, differently from socket that uses a dispatching method in the client interface. The client checks autonomously the right range of players between 2 and 4 before sending it to the server. If the number of players is not reached, the clients receive a waiting partners message, which put them in the waiting room, otherwise they receive the begin message, starting the game. The messages are always sent according to the kind of connection in the previously explained way.

2. Draw_Component_Diagram

When the player wants to draw a component, they send a draw component request, sending it with the same paradigm explained for the first sequence diagram according to the connection kind. The server sends back a draw component response message, handled differently for socket and RMI as previously explained. When the message is received, the client can interact with the component when the client interface calls for the handling component method on the UI. In case of a booked component, the player receives the component calling the getter on the client interface.

3. Add_Component_Diagram

The player contains a copy of their ship on their personal client Interface, in this way the client can place the component calling directly the methods on the client interface. The ship is sent to the server only when the ship construction state has ended and the fixing state can begin. This method avoids to send for each component too many messages to the server in order to give the coordinates. Received the component, the player can decide to put it into a booking list contained in the client interface or to return it to the server, putting the component in a return component message, sent according to the socket or RMI connection previously explained.