Communication protocol

Giovanni Demasi
Vittorio Antigio Dall'Oglio
Alessandro Curti

April 2021

Contents

1 Introduction

When the connection is initialized the server sends a "ping" message and the client replies with a "pong" message. This action is continuously repeated, the client has two seconds to sends a pong, if this doesn't happen the server detects the disconnection of the client and stops to ping him. This procedure is performed every time a new client connects to the server.

All the message exchanged between client and server are serialized objects (only the messages, the model isn't serialized) and everyone of them implements an interface and overrides one method called handleMessage.

These objects are divided in two formats: one that runs from client to server(client Message) and the other one with the opposite route(server Message).

The whole communication can be divided into eight phases (described below), each one beginning with a ping-pong message exchange.

Every phase has some common messages:

ErrorMessage (server)
enumeration error

This message is sent to the client whenever the controller detects an error. An enumeration of errors is used.

disconnection message (server)
String (nickname of the disconnected player)
reconnection message (server)
String (nickname of the reconnected player)

The disconnection and reconnection messages are sent to every player (excepted the one who disconnected/reconnected).

If the match is singleplayer a lorenzo update message is sent.

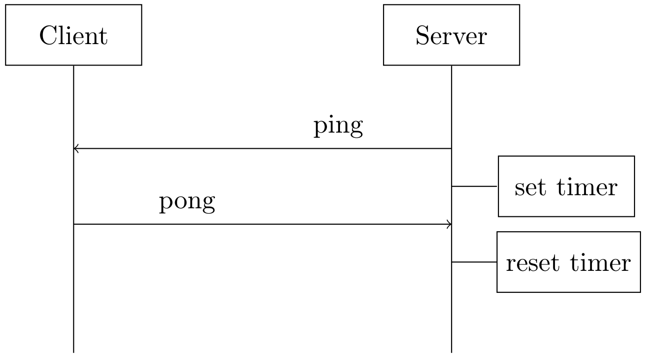
lorenzo update (server)
Integer (Lorenzo faith track position)
String [][] (dev card grid)
String (message about what Lorenzo has done)

2 Ping

Once the connection is established, the server immediately begins pinging the client on a regular basis to check if the connection is still alive. The client responds to the ping with a pong. If the client does not answer to the ping within a certain amount of time, the connection is dropped.

ping (server)

pong (client)

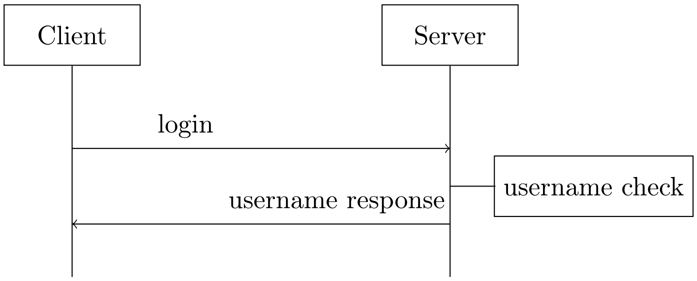


3 Login

This interaction concerns all the players and happens only once when a player connects to the game. It is used to let the player choose his nickname that must be unique between all game sessions and it must respect some rules like minimum or maximum length.

login request (client)
String "username"

username response (server)
boolean (true if the nickname can be used)
String "username" (in this way the client saves it)



4 Client Re-Connection

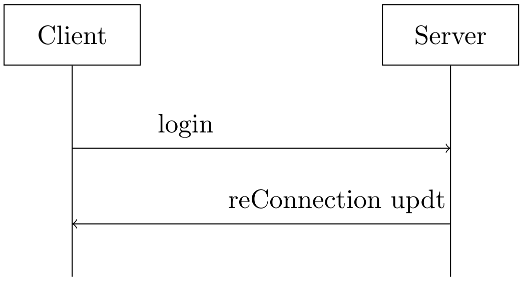
This interaction concerns only the reconnecting player and happens when a player reconnects to the game (which is not finished yet) after a past disconnection. The player must insert the equivalent of the old nickname.

login request (client)
String "username"

reconnection update (server)
map of String (player nickname) and arraylist of id (string) of development card (i.e. array[0] contains the id of the dev card in the first slot)
map of String (player nickname) and integer (faith track position of every player)
map of String (nickname) and arraylist of id (string) of the played leader cards of every player
list of id (string) of in-hand leader cards
map of String (nickname) and map of integer and resources (strongbox)
map of String (nickname) and map of integer and arraylist of resources (warehouse)
map of String (nickname) and integer (number of cards in-hand of every player)
map of String (nickname) and boolean (true if the player is connected)
map of String (nickname) and map of integer and boolean (true if the vatican report has been activated)
boolean (true if the player has selected the starting resources and leader cards)

dev card grid (server)
String[][] (the ids of the cards)

market (server)
MarbleColors[][]
MarbleColors freeMarble

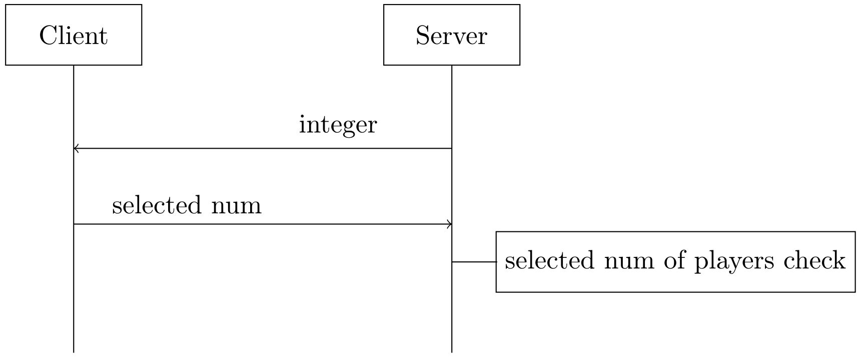


5 Number of players selection

This interaction concerns only the first player and follows the login phase. The first player chooses the game's players number. This number must be included between 1 and 4.

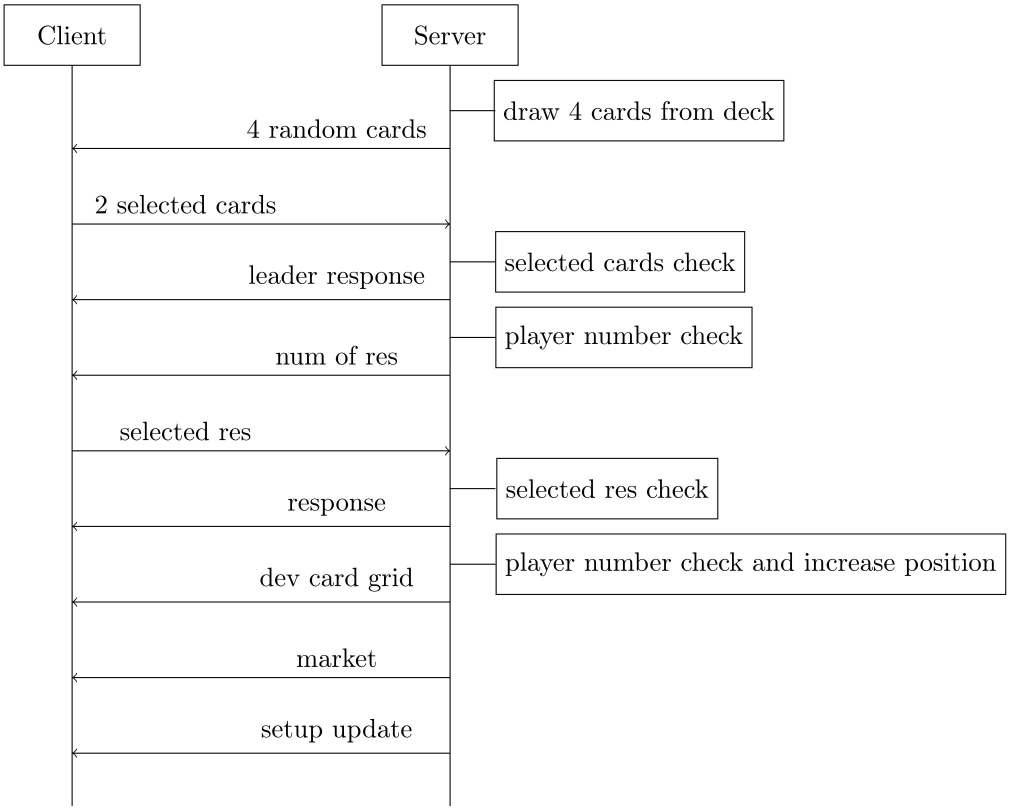
number of players selection request (server)
integer (4, the maximum number of players)

number of players selection reply (client)
integer



6 SetupGame

This interaction concerns all the players of the game and it happens only once and concerns the setup of the game. Every player, depending by his turn, chooses its initial leader cards and resources.
leader cards request (server)
list of the id (string) of four random Leader Cards
leader cards reply (client)
id (string) of two Leader Cards
resources request (server)
integer
resources reply (client)
map of integer and arraylist (warehouse configuration)
dev card grid (server)
String[][] (the ids of the cards)
market (server)
MarbleColors[][]
MarbleColors freeMarble
setup update (server)
map of String (nickname) and id (string) of the two selected leader cards
map of String (nickname) and map of integer and arraylist (warehouse configuration)
map of String (nickname) and integer (new faith track position)


Client
Server
draw 4 cards from deck
4 random cards
2 selected cards
selected cards check
leader response
player number check
num of res
selected res
selected res check
response
player number check and increase position
dev card grid
market
setup update

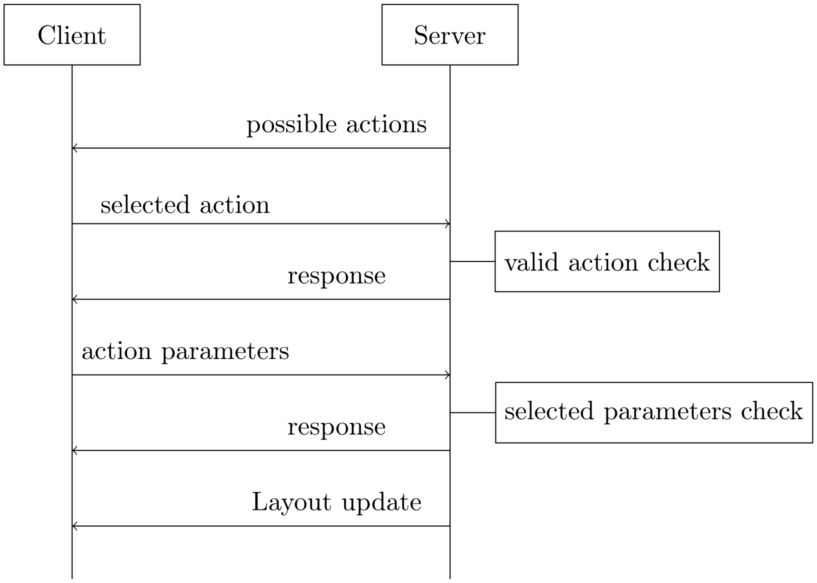
7 Actions

During this interaction, which may be different for different actions, every player chooses the desired action.
It concerns only one player at a time during his turn.

7.1 Actions (almost all)

Almost every action has a similar interaction. It is described below.

possible action (server)
enumeration list of (possible) actions
action reply (client)
action (enum)
action parameters (client)
username (of the player that do the action)
objects (depending by the type of the action*)
Layout update (server)
objects (depending by the type of the action*)



(*) Now we describe in details, for every action, the attributes of the parameters request/reply and layout update messages.

Below we will use the **payment resources** message which contains:
lists of resources used to pay (from warehouse depot, leader cards depot and strongbox).
We will also use the **faith track update message** which contains a list of String (nickname of who activated the vatican reports), integer (position of the vatican report), boolean Report (true if the vatican report has been activated), map of String (nickname) and integer (faith track position of every player).
Every layout update has a String which corresponds to the player who made the action.

7.1.1 Play Leader Card action

action parameters (client)
id (string) of the selected leader card
Layout update (server)
id (string) of the selected leader card
map of integer and arraylists of resources (warehouse)

7.1.2 Discard Leader Card action

action parameters (client)
id (string) of the selected leader card
Layout update (server)
id (string) of the selected leader card
faith track update

7.1.3 Buy Development Card action

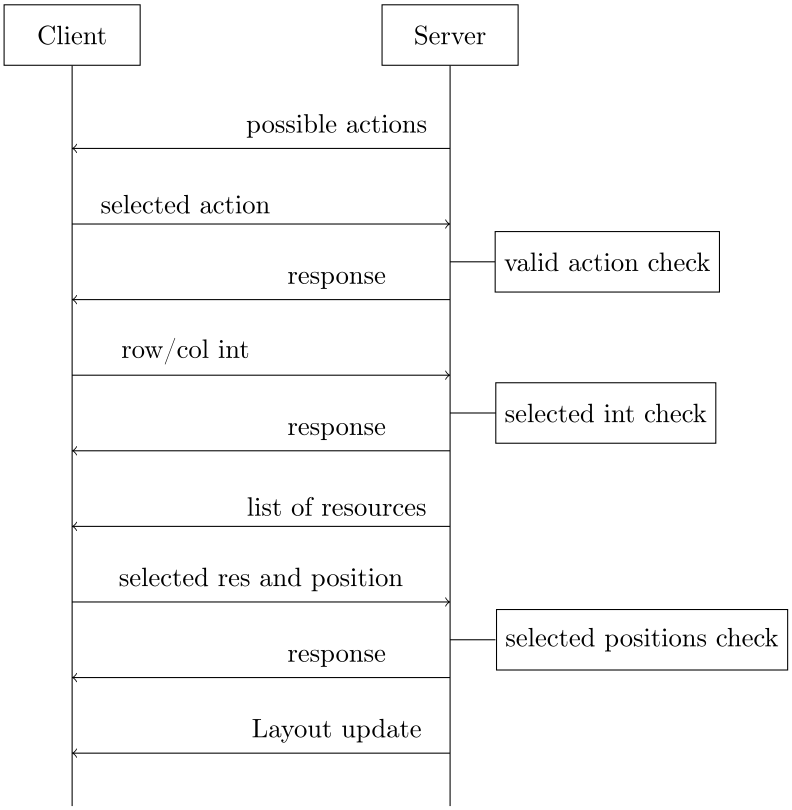
action parameters (client)
color (enum)
level (int)
int (dev card slot)
payment resources
Layout update (server)
id (string) of the selected card
id (string) of the card under the selected card, null if the selected card is the last of the deck
color (enumeration) and level (integer) of the new card
map of integer and arraylists of resources (warehouse)
map of resource and integer (strongbox)int (dev card slot)

7.1.4 Use Production action

action parameters (client)
payment resources
arraylists of resources (one for the board production resources gain and one for the leader card production resources gain)
arraylists of integer (one for the indexes of development card slot to use for production and one for the indexes of leader card slot to use for production)
Layout update (server)
map of integer and arraylists of resources (warehouse) map of resource and integer (strongbox)

7.2 Market action

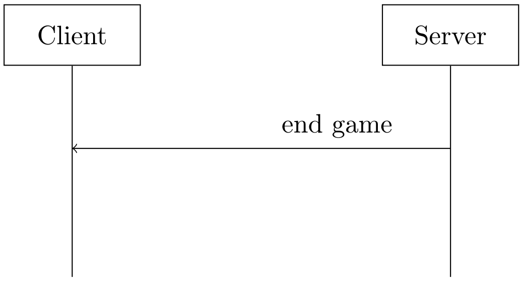
action request (server)
enumeration list of (possible) actions
row/column selection (client)
username (of the player that do the action)
integer (number of column/row)
boolean (true if the user selected a row, false if column)
list of resources for the effect of exchange leader cards
resources manage request (server)
list of resources (enumeration)
resources manage reply (client)
map of integer and arraylists of resources (warehouse)
Layout update (server)
array[][] of marbles
free marble
lists of resources (one for every storage, null if not modified)



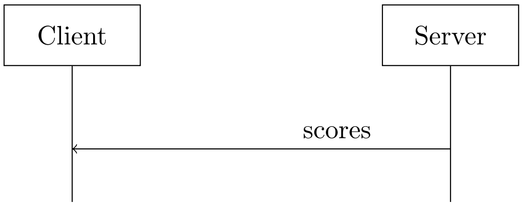
8 End game

This interaction concerns all the players of the game.
If a player performs a end game move, a last turn message is sent to every clients.
The server remains available to begin a new match and the clients can reconnect or exit the game.

end game (server)



final score (server)
map of String (nickname) and integer (score of the player)
boolean (true if Lorenzo won, false if he has lost or if the game is multiplayer)


Client
Server
scores