For Peer-Review 2: SOCKET part

Barabino Francesco, Bertolini Gianluca, Borrelli Elisa – Group AM53

Network Socket - Sequence diagrams and description of the exchange of messages

Description for Client connection:

The Server supports the management of a single game.

The Game class is created and the Server interfaces with the different Clients, each associated with a Player (identified by id).

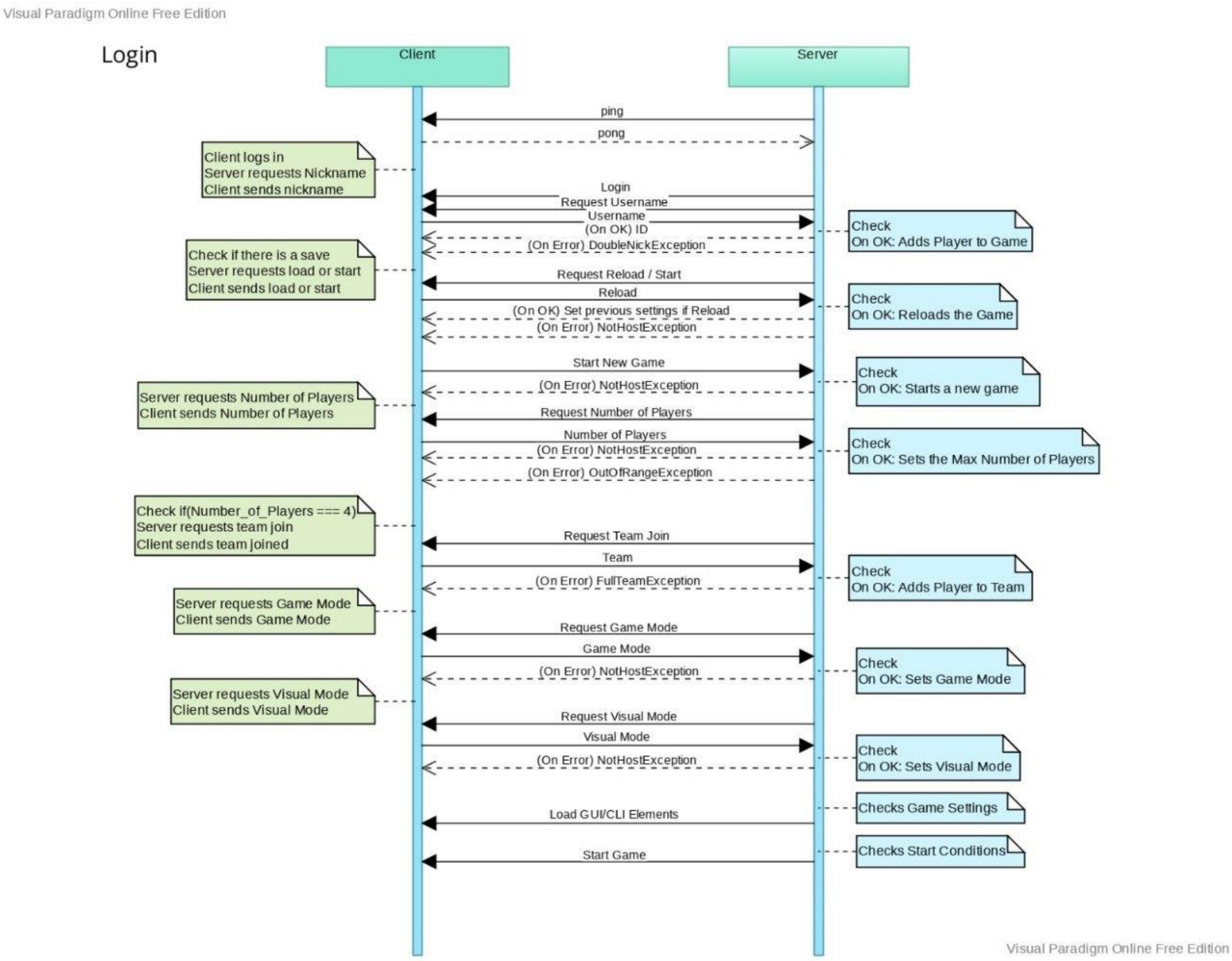
For the Server Client Handler, we use libraries and getInputStream/getOutputStream. We serialize the exchange of classes using Gson library.

During the connection, and as represented in the following diagrams, the Server makes requests, waits for Client and then sends a reply with: or a positive response (on OK) in addition to the requested information, or an error (on Error).

In general, for each message it is better to conclude with \n (also for matter of order).

Every interaction should start with an exchange of pings and pongs.

We have decided to represent the 3 salient sequence diagrams for our structure: the "Login", the management of the "Round" and in particular the final phase of the "Endgame".


Login
Client
Server
ping
pong
Client logs in
Server requests Nickname
Client sends nickname
Login
Request Username
Username
(On OK) ID
(On Error) DoubleNickException
Check
On OK: Adds Player to Game
Check if there is a save
Server requests load or start
Client sends load or start
Request Reload / Start
Reload
(On OK) Set previous settings if Reload
(On Error) NotHostException
Check
On OK: Reloads the Game
Start New Game
(On Error) NotHostException
Check
On OK: Starts a new game
Server requests Number of Players
Client sends Number of Players
Request Number of Players
Number of Players
(On Error) NotHostException
(On Error) OutOfRangeException
Check
On OK: Sets the Max Number of Players
Check if(Number_of_Players === 4)
Server requests team join
Client sends team joined
Request Team Join
Team
(On Error) FullTeamException
Check
On OK: Adds Player to Team
Server requests Game Mode
Client sends Game Mode
Request Game Mode
Game Mode
(On Error) NotHostException
Check
On OK: Sets Game Mode
Server requests Visual Mode
Client sends Visual Mode
Request Visual Mode
Visual Mode
(On Error) NotHostException
Check
On OK: Sets Visual Mode
Load GUI/CLI Elements
Checks Game Settings
Start Game
Checks Start Conditions

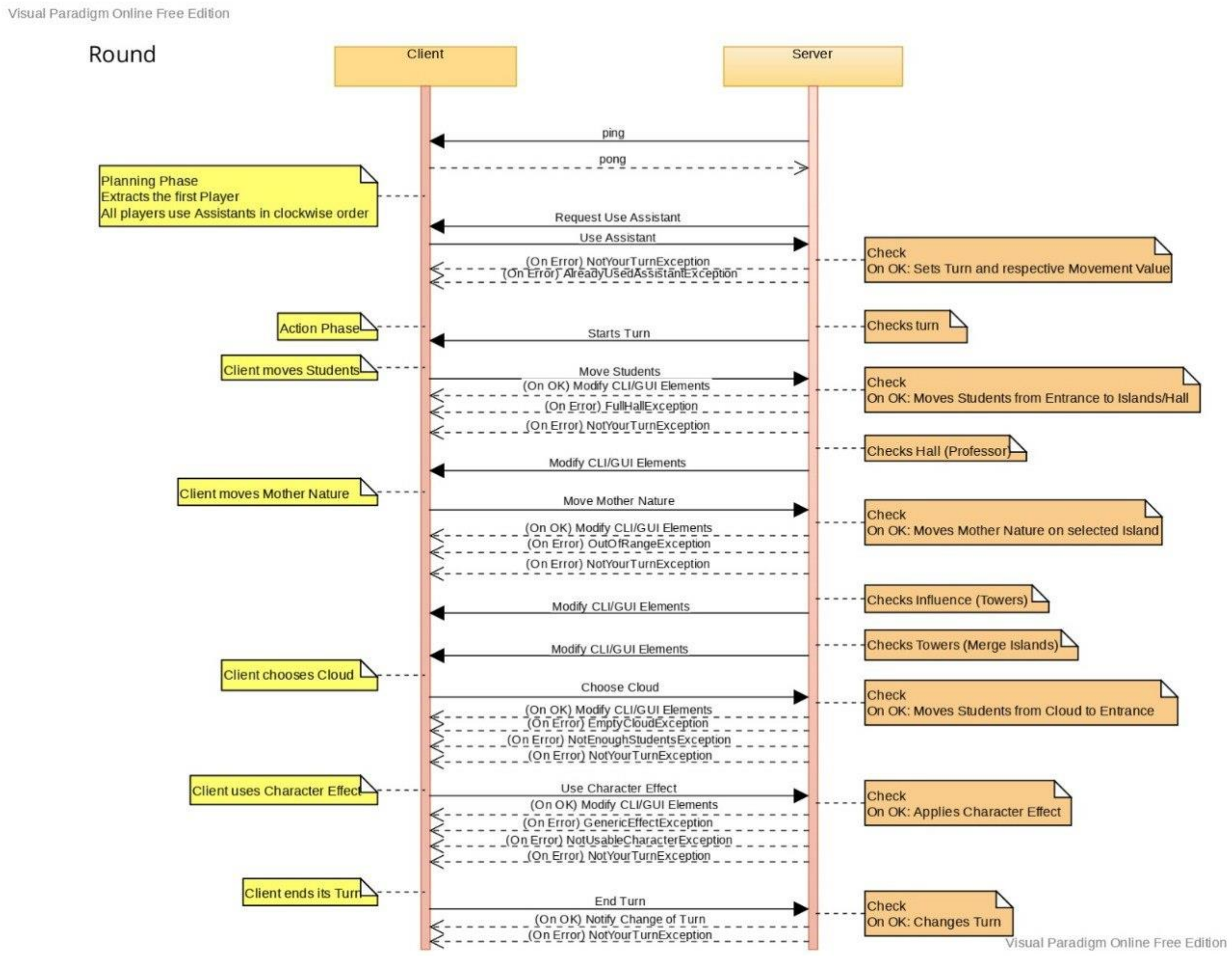
Login:

More precisely we start the exchange between Client-Server by the nickname request (Server side), to identify and add the Player.

Subsequently, it is immediately managed the case of saved/new Game, then: the Player is asked whether to reload from file or not.

Set/choose the number of players, if it is the value 4, the Game is organized for teams, otherwise the Game between single Players.

Finally, the Player is asked for the Game mode, basic or advanced rules, the CLI view or with GUI elements and the Game can start.


Round
Client
Server
ping
pong
Planning Phase
Extracts the first Player
All players use Assistants in clockwise order
Request Use Assistant
Use Assistant
(On Error) NotYourTurnException
(On Error) AlreadyUsedAssistantException
Check
On OK: Sets Turn and respective Movement Value
Action Phase
Starts Turn
Checks turn
Client moves Students
Move Students
(On OK) Modify CLI/GUI Elements
(On Error) FullHallException
(On Error) NotYourTurnException
Check
On OK: Moves Students from Entrance to Islands/Hall
Modify CLI/GUI Elements
Checks Hall (Professor)
Client moves Mother Nature
Move Mother Nature
(On OK) Modify CLI/GUI Elements
(On Error) OutOfRangeException
(On Error) NotYourTurnException
Check
On OK: Moves Mother Nature on selected Island
Modify CLI/GUI Elements
Checks Influence (Towers)
Modify CLI/GUI Elements
Checks Towers (Merge Islands)
Client chooses Cloud
Choose Cloud
(On OK) Modify CLI/GUI Elements
(On Error) EmptyCloudException
(On Error) NotEnoughStudentsException
(On Error) NotYourTurnException
Check
On OK: Moves Students from Cloud to Entrance
Client uses Character Effect
Use Character Effect
(On OK) Modify CLI/GUI Elements
(On Error) GenericEffectException
(On Error) NotUsableCharacterException
(On Error) NotYourTurnException
Check
On OK: Applies Character Effect
Client ends its Turn
End Turn
(On OK) Notify Change of Turn
(On Error) NotYourTurnException
Check
On OK: Changes Turn

<u>Round</u>:

Along the lines of the Game phases, the Round opens with the planning phase, where the first Player is drawn and continues clockwise, asking which Assistant to use.

Immediately we move on to the action phase, with the management of the turns and, lately, continuing with students and mother nature movement. For the latter, an exception is deferred if you leave the range of accepted locations, on the islands (OutOfRangeException).

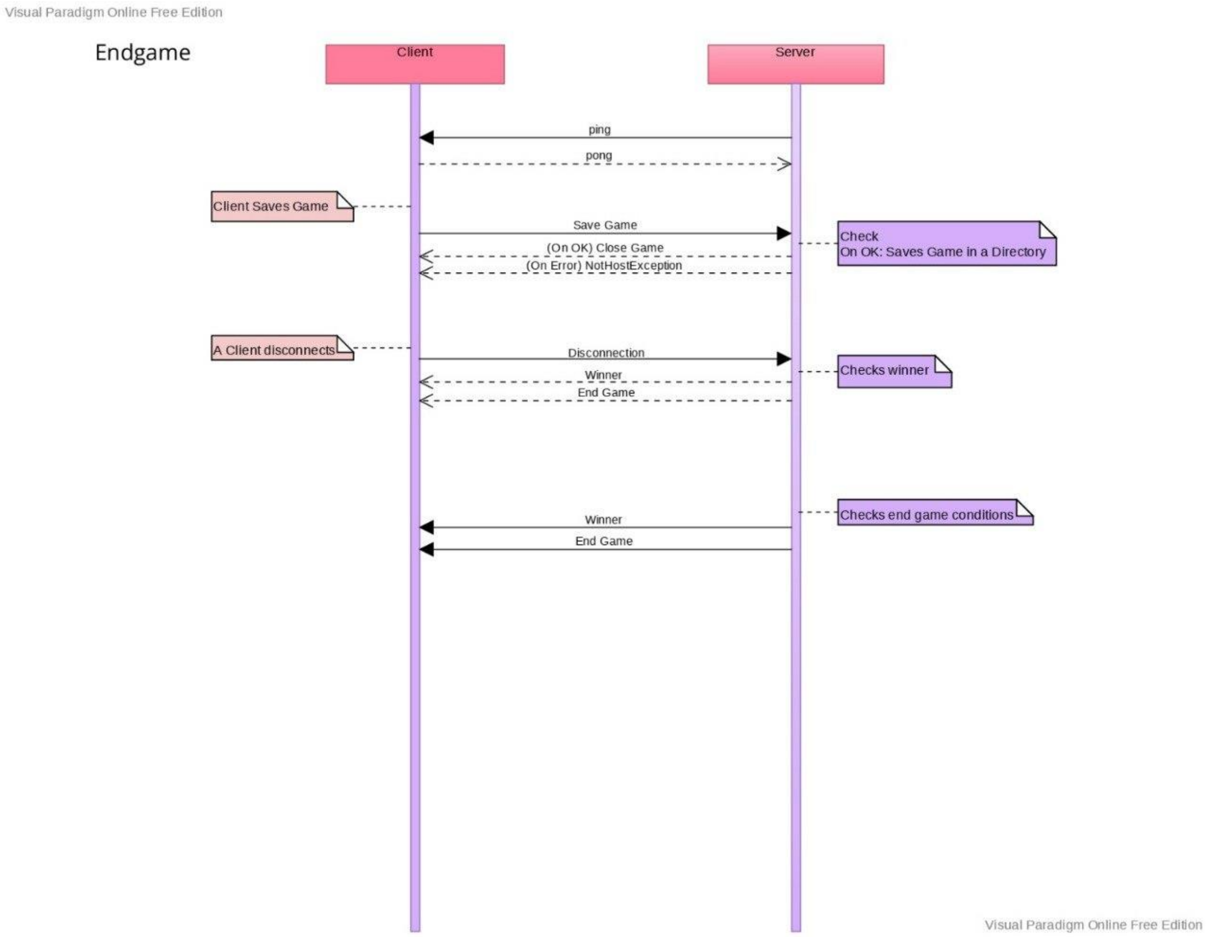
The Server, for GUI side, manages the calculation of influence and the fusion of islands.

In every part of your turn, you can use an effect of a Character, among those available (throw exceptions otherwise).

Before moving on to the next Player, you have to conclude your turn, with a specific command.

The Player can choose a Cloud as one of its last possible actions.

During the Round, in almost every ClientReply-ServerRequest step, by the Server, you can receive exceptions in case of attempted interactions of the players, out of theirselves turns.


Visual Paradigm Online Free Edition
Endgame
Client
Server
ping
pong
Client Saves Game
Save Game
Check
On OK: Saves Game in a Directory
(On OK) Close Game
(On Error) NotHostException
A Client disconnects
Disconnection
Checks winner
Winner
End Game
Checks end game conditions
Winner
End Game
Visual Paradigm Online Free Edition

Endgame:

Very briefly we also represented an exchange of messages for the closure of the Game, with a previous request for saving by lead Player (the first logged in).

Otherwise, the winner is checked when a Client is disconnected.

With “Winner” (ServerReply side) the victory conditions are checked.