Commmunication

Edited by
Riccardo Ambrosini Barzaghi
Marco Bonalumi
Simone Cattaneo

AM26

Academic year 2019/2020

Communication Protocol

The communication in our application relies on serializable objects. These objects are divided in two formats: one that runs from client to server(client Event) and the other one with the opposite route(server Event).
Server Events can be separated in those that communicate changes in the Model to send to the View, those handling the user interaction and subsequently changing the Model, and lastly those that manage the connection.
Client Events can be categorized in three types: one transports the requests that are going to be elaborated by the server; one that manages the connection; one that manages the game.

Server Events: serverToClient

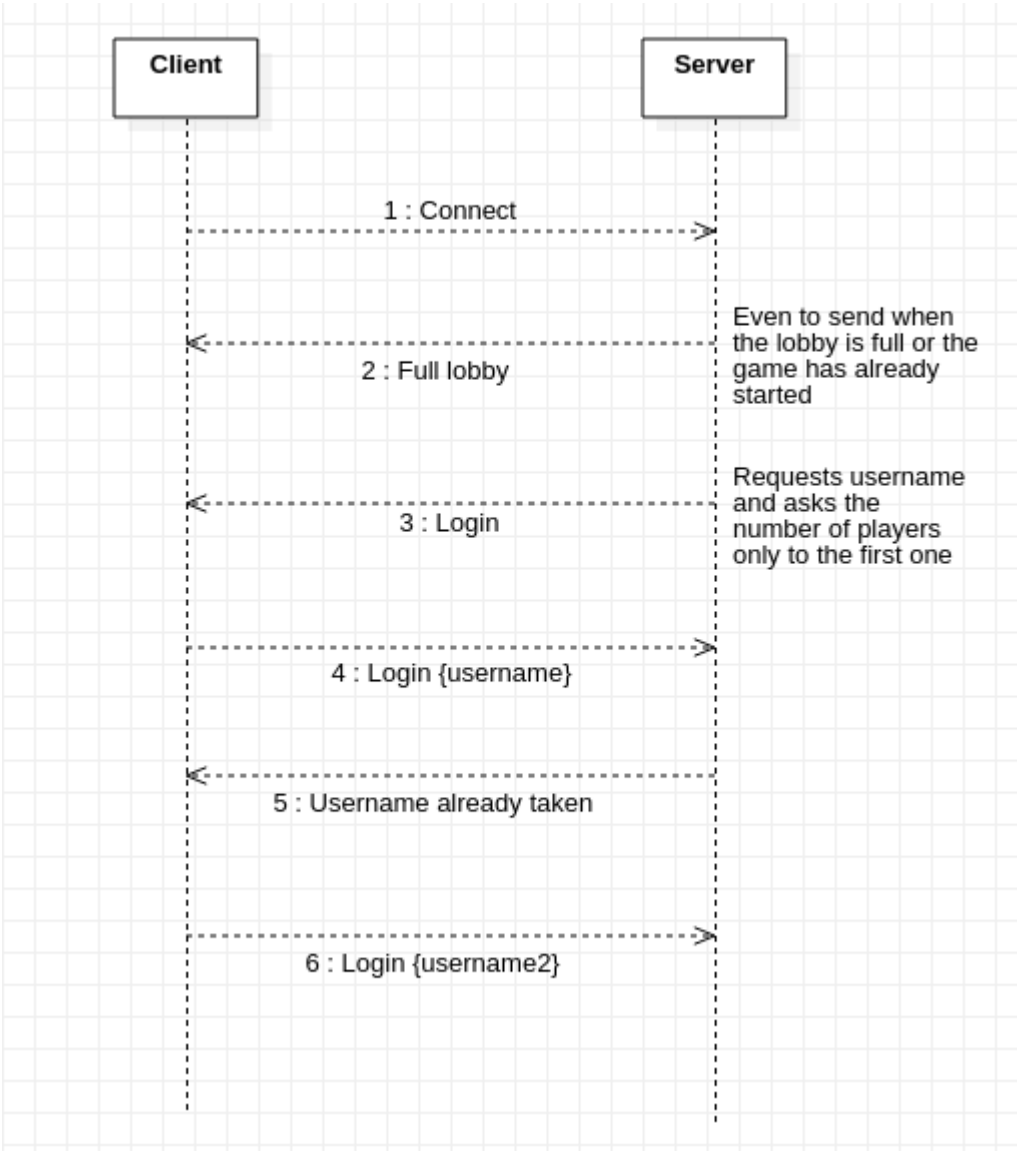
- WorkerSelectionEvent: sets the selected worker on Model
- BuildEvent: tile on which the selected worker built
- MoveEvent: tile on which the selected worker moved
- InvalidUsernameEvent: sent when a player chooses an unavailable username
- LobbyFullEvent: sent when a client tries to connect to a full lobby
- PlayerDisconnectionEvent: see Ping section below
- MatchBeginEvent: communicates the others' players selection (name, color, divinity)
- Ping: see Ping section below
- PossibleActionsEvent: list of possible actions sent to the client (see Simulation of a turn)
- LoginRequestEvent: manages the login of players
- StartTurnEvent: sent at the start of a player's turn
- TextMessageEvent: sent to the clients with error or communication messages
- WinnerEvent: sent to the clients asserting the victory of a player
- LoserEvent:sent to the clients asserting the loss of a player
- SelectedDvinitiesEvent: complete list of divinities (see Divinity selection)
- DivinitiesInGameEvent: list of divinities to pick from (see Divinity selection)

Client Events: clientToServer

- WorkerSelectionEvent: selects the worker that will perform an action
- MoveQuestionEvent: requests a move on a tile, giving its coordinates
- BuildQuestionEvent: requests a build on a tile, giving its coordinates
- BuildDomeQuestionEvent: requests to build a dome on a tile, giving its coordinates
- EndTurnQuestionEvent: requests to end turn (relevant for divinities with options)
- Pong: see Ping section below
- LoginEvent: handles the user login
- UnexpectedDisconnectionEvent: sent to controller when a connection is not alive (see Ping)
- DivinitiesInGameSelectionEvent: picks which divinities will be playable in the upcoming game
- PlayerDivinitySelectionEvent: pick a player's divinity (see Divinity selection)

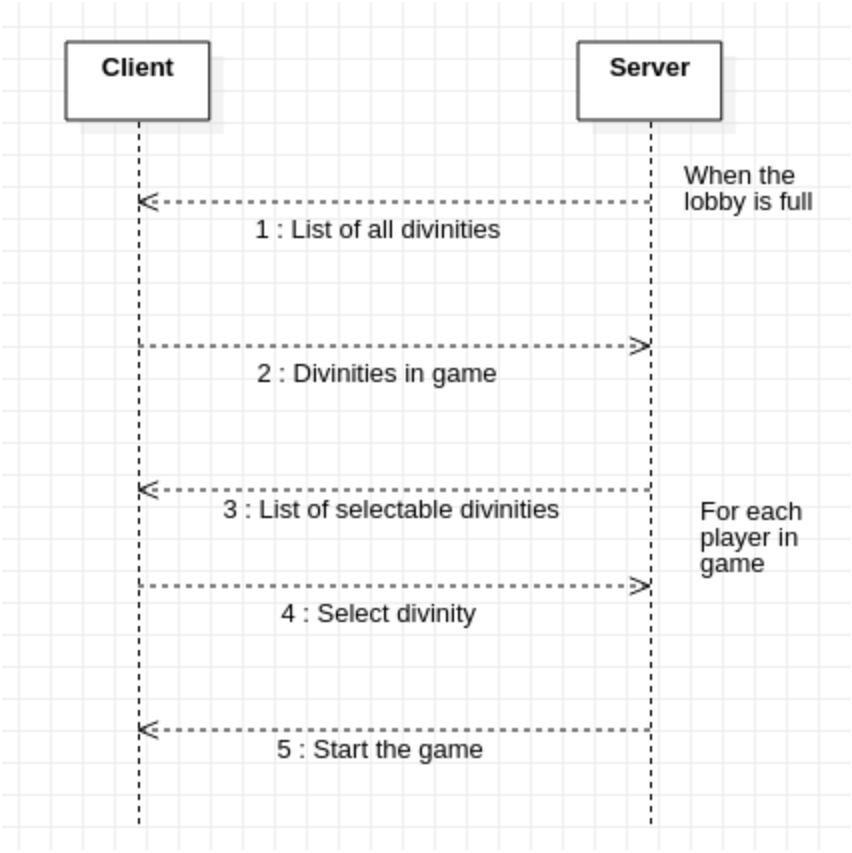
Login

When a client connects, if he is the first player, the server asks him how many player will participate in the game; if he does not respond within a certain amount of time, the server will drop his connection and will consider the next player connecting as the first one.
The generic player will send his username and if the lobby is not full nor the game has started, he will be added to the list of active players. If the username is already taken, the player will have to choose another one.



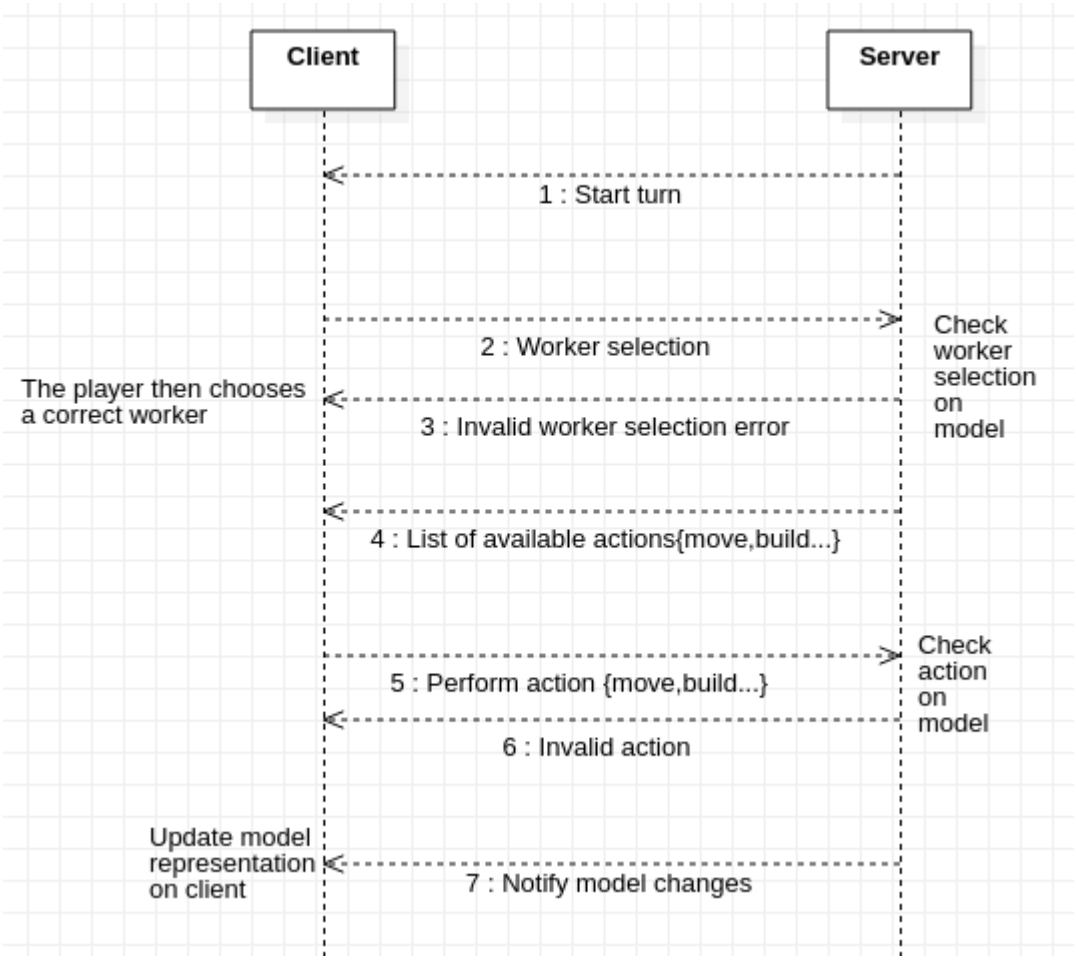
Divinity Selection

After all the players have connected (2 or 3), the last player, will have to choose which divinities will be selectable by the others. After this event, the server will ask each player, starting from the first, what divinity they will play, sending the client the list of possible divinities.
At the end of this phase, the game will begin but beforehand every player will be informed with the others' username, divinity and color; the first player will then start their turn.



Simulation of a turn

When a player has to start their turn, they will be notified by the server.
They will then proceed to select a worker; if the selection is not valid they will be notified and will have to choose again. Upon the worker selection, the server sends a list of possible actions (move,build, build dome, end turn). When the list arrives to the client, if this one contains more than one element, they will select an action to perform and send such event to the server; otherwise, if this list contains a single element, the player will be bounded to perform such action without the possibility to select differently.
As previously the server checks the action and then, if it is correct, it will send the changes in the Model to the client to eventually update its View.



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 not answer the ping within a certain quantity of time, the connection is dropped.

If the player disconnects before the match begins, i.e. when he is waiting for other players to join, another place is made available to take. If he disconnects during the game, according to specifications, the game ends for everyone (the clients connections are dropped).