Communication

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 (e.g. login); one that manages the connection; one that manages the game.

Server Events: serverToClient

- BuildEvent: tile on which the selected worker built
- DivinitiesInGameEvent: list of divinities to pick from (see Divinity selection)
- DivinitiesSetupEvent: List of divinities chosen by the last player (see Divinity selection)
- DivinityInitializationEvent: Informs each client what divinity other players chose
- EndTurnEvent: Sent to inform the client of the end of the turn
- GameSetupEvent: Bears all the information regarding players(name, divinity,color)
- GameStartEvent: Tells the client that the game started
- InLobbyEvent: Confirms the correct joining of a lobby
- InvalidBuildEvent: Sent if the build cannot be performed
- InvalidMoveEvent: Sent if the move cannot be performed
- InvalidWorkerPlacementEvent: Sent if the the workers cannot be placed correctly
- InvalidWorkerSelectionEvent: Sent if there is not a player's worker at the coordinates
- LobbyFullEvent: sent when a client tries to connect to a full lobby
- LoserEvent:sent to the clients asserting the loss of a player
- MoveEvent: tile on which the selected worker moved
- Ping: checks if a client is online
- PlayerDisconnectionEvent: sent when a client disconnects
- PossibleActionsEvent: Map containing the possible actions and the list of coordinates
- ServerDisconnectionEvent: called on client side when the server drops
- StartTurnEvent: notifies a client that their turn started.
- TextMessageEvent: sent to the clients with error or communication messages
- UsernameAlreadyUsed: sent if the client chose an already taken name. Send a list of names
- WelcomeEvent: sent when a user connects. This event tells them to input a username

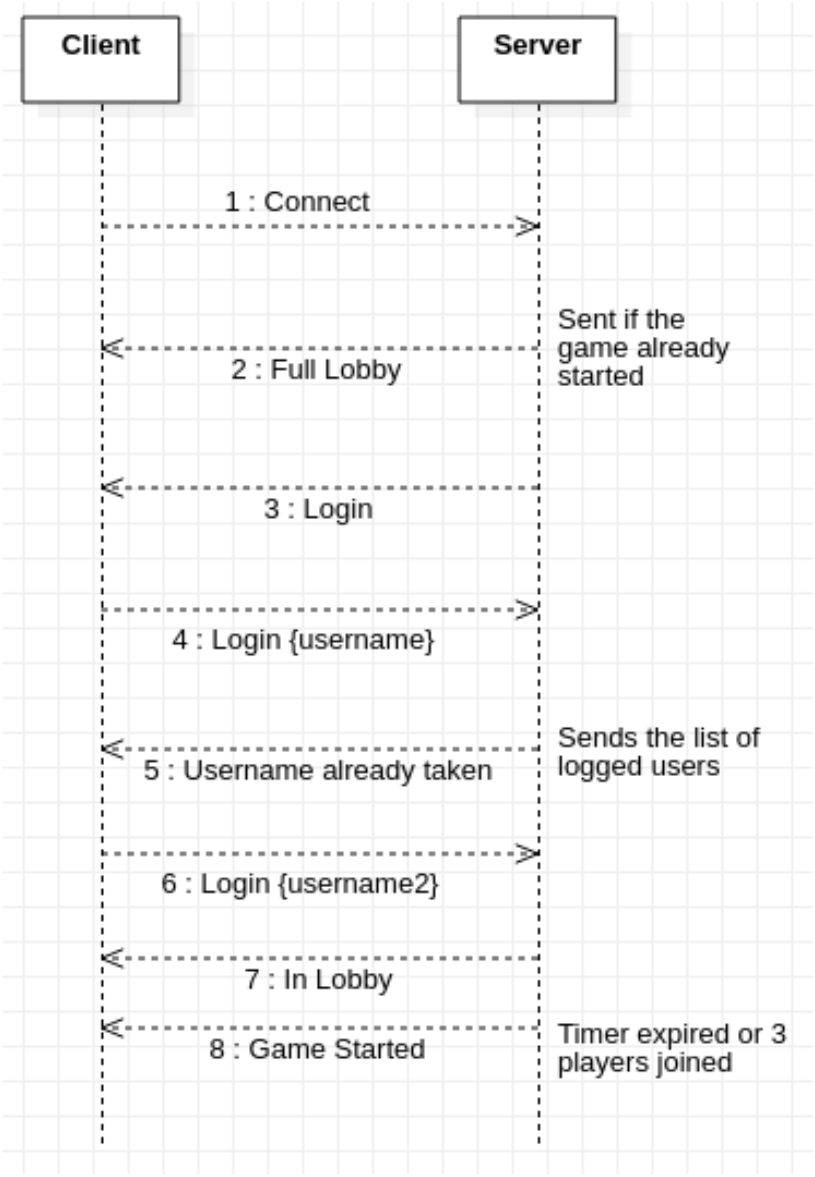
- WinnerEvent: sent to the clients asserting the victory of a player
- WorkerInitializationEvent: Asks the client to place their workers
- WorkerPlacementEvent: Informs the client where another player placed the workers
- WorkerSelectionEvent: tells the client to select a worker

Client Events: clientToServer

- ActionQuestionEvent: performs an action with the selected worker on a tile
- DisconnectionEvent: Sent when the client wants to disconnect
- DivinitiesInGameSelectionEvent: picks which divinities will be playable in the upcoming game
- DivinitySelectionEvent: Informs the server which divinity the client chose
- LoginEvent: handles the user login
- Pong: answers the ping to assert the presence of the client
- WorkerSelectionQuestionEvent: selects the worker that will perform an action
- WorkersPlacementSelectionEvent: tells the server where the client want to place the workers

Login

When two players connect a timer start; if a third player does not arrive within the expiration of the timer, the game starts with two players.
A generic player will send a username and if the lobby is not full nor the game has started, he will be added to the list of in game players. If the username is already taken, the player will have to choose another one.

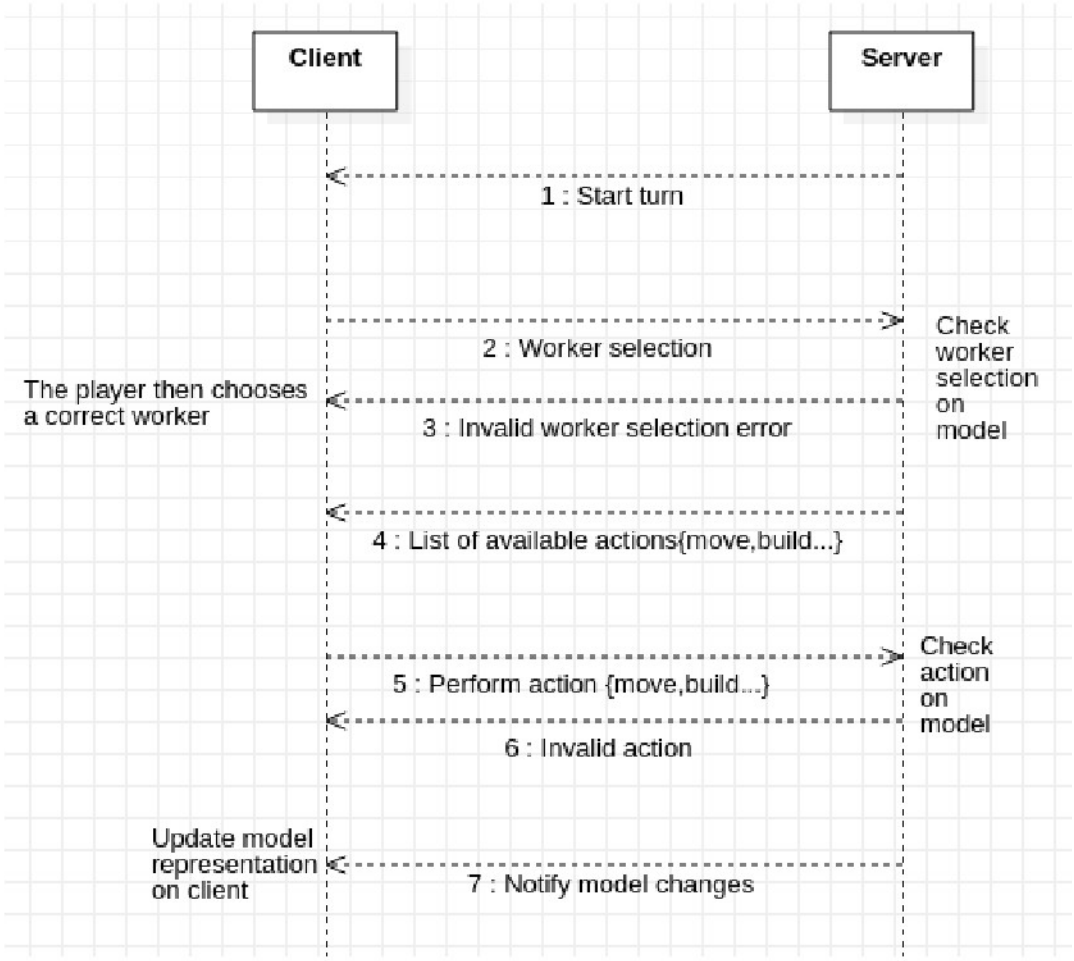


Game setup

After all the players have connected (2 or 3), the setup will begin.
The server sends to the last player a list of all the divinities with their description and the list of players. The last player will have to choose which divinities will be selectable by the others and the starting player.
Once the server receives the information, it will ask the starting player which divinity they want to choose. After every player chose their divinity, the server sends to the clients a packet, containing the list of users (in order of their turn) alongside their divinity. Then every player chooses where to place the workers, after receiving the information about where the others placed theirs. If one of these operation is not successful, they will have to select again.
Eventually the game starts with the first player selecting a worker.

Simulation of a turn

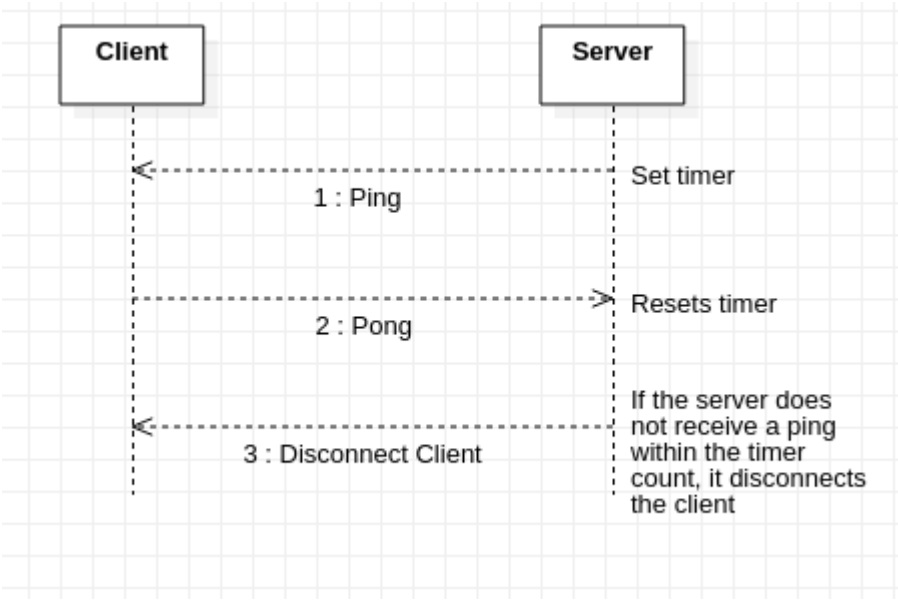
When a player has to start their turn, they will be notified by the server.
They will then proceed to select a worker; if the selection is not valid, they will be notified and will have to choose again. Upon the worker selection, the server sends a list of possible actions (move,build, build dome, end turn) and the relative coordinates where they are performable.
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.
The server checks the action and then, if it is correct, it will sent to every client an event to update their board's representation and it will send to the client another event containing new actions and coordinates. Once the client exhausts the possible actions, an event is sent to inform about the end of the turn and the successive player starts their turn.



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 amount 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).



End of the game

If a player performs a losing move, a losing event is sent to the clients and the losing client can no longer perform actions.
When a there are two players and one performs a losing move, or one player is in a winning condition, the server sends a winning event informing the players of the winner’s name and ending the current match.
The server remains available to begin a new match and the clients can reconnect or exit the game.