Eryantis Protocol Documentation

Matteo Rigat, Nicolò Avarino, Giuseppe Schiavoni

Group GC18

1.   Messages

1.1 Connection accepted

This message is sent from the server to the client after establishing a connection, so the player knows the connection has been established.

1.2 Login Request

This message is sent from the server to the client to request the player his nickname

Possible responses:

- Login Reply (1.3)

1.3 Login Reply

This message is sent from the client to the server to create a player session. The session is tied to the TCP connection.

Arguments:

- Username: the name of the user

Possible responses:

- NumberOfPlayersRequest(1.4) : when the login is successful

- Error : when the login fails (due to a repeated nickname)

- Wait: when another player has already started a match and is choosing the game variables

- ModelState: when the other players has already chosen all the variables

1.4 Number of Players Request

This message is sent from the server to the client when a new Game instance is created, so only the first player of every match is asked to provide this information

Possible responses:

- NumberOfPlayersReply (1.5)

1.5 Number of Players Reply

This message is sent from the client to the server to specify the desired number of players

Arguments:

- NumberOfPlayers

Possible responses:

- ExpertModeRequest (1.6) when the number is correct (2 or 3)

- Error: when the number is wrong

1.6 Expert Mode Request

This message is sent from the server to the client. Same considerations of the Number of Players message about the first player

Possible responses:

- ExpertModeReply (1.7)

1.7 Expert Mode Reply

This message is sent from the client to the server.

Arguments:

- A Boolean indicating the choice made (y/n)

Possible responses:

- Waiting for another player(s): when the match is correctly configured and the server is waiting for the chosen number of players to connect

- ModelState: if all the other players did connect while the first one was choosing game parameters

- Error: when user types an incorrect symbol

1.8   GameStarted

This message is sent from the server to the client when two or three players have been matched and a game starts.

Arguments:

- OtherPlayerName: the username of the other player ("opponent")

- ModelState: the current (initial) state of the game board (islands) and the player's school boards

1.9   GameUpdated

This message is sent from the server to the client when the state of the gameboard (islands and players's school boards) has been updated.

Arguments:

- CurrentPlayer: the ID of the player who currently has control.

- GamePhase: the current phase of the game (PlayAssistantCard, MoveStudents, MoveMotherNature, ChooseCloud, GameEnded)

- ModelState: the current state of the board.

1.10   PerformGameMove

This message is sent from the client to the server when a move is performed by the player.

Arguments:

- TypeOfMove: The identifier of the kind of move which the player wants to perform

- Variable(s) for the specific kind of move (e.g. card number, island number, student colour, cloud number, MotherNature movements)

Possible responses:

- GameUpdated: the message has been received and the board has been updated

- MoveError: the message has been received but the move was impossible

1.11   MoveError

This message is sent from the server to the client after an incorrect PerformGameMove message is sent, either because the players who asked the move is not in control or because the specific move does not match the current GamePhase
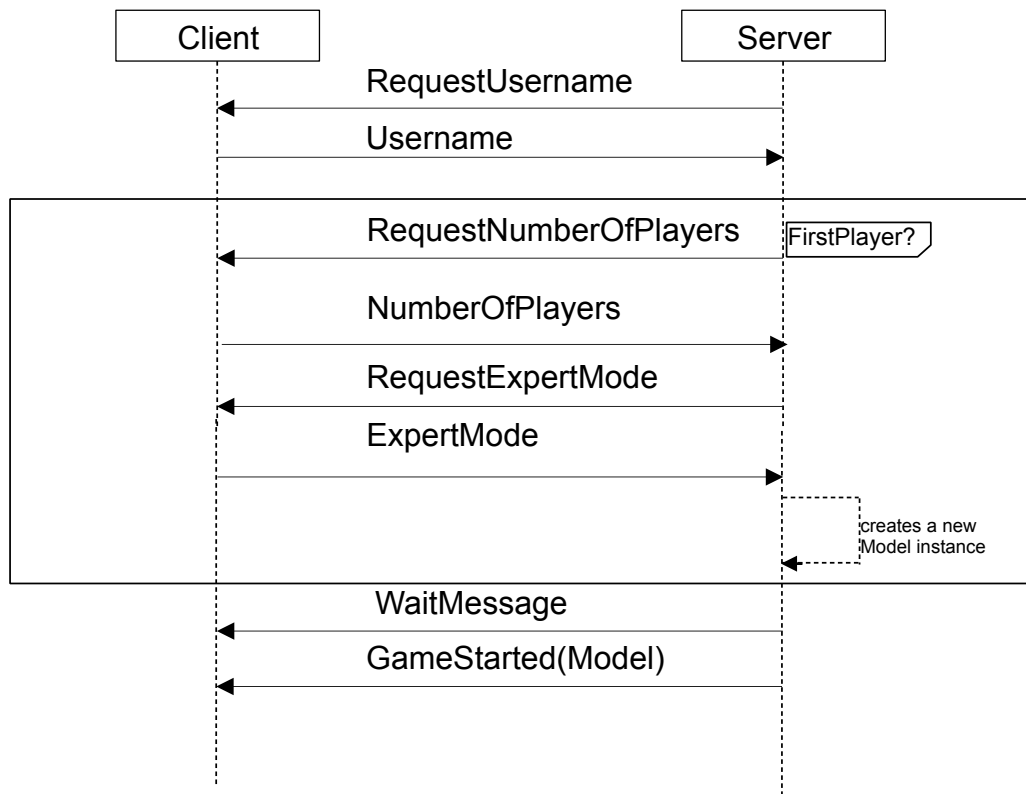
1.12   GameEnded

This message is sent from the server to the client after the end-of-game condition is reached. This message also terminates the current game session and shows each player wether he's the winner or not.
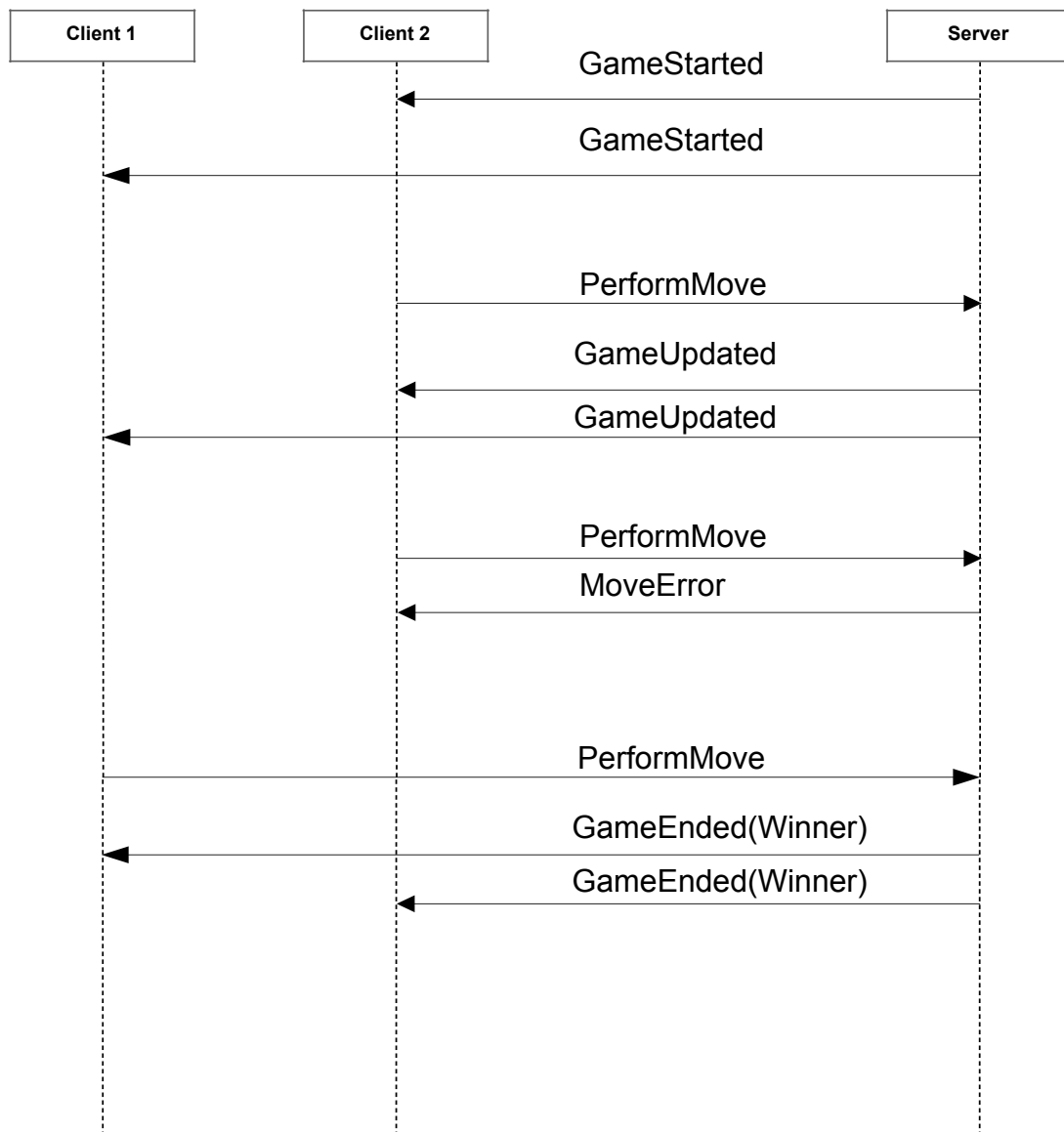
Arguments

• Winner: The player ID of the winner of the match.

2   Scenarios

2.1   Starting the game

2.2 Playing (and ending) a match

```
┌──────────┐        ┌──────────┐                          ┌──────────┐
│ Client 1 │        │ Client 2 │                          │  Server  │
└────┬─────┘        └────┬─────┘                          └────┬─────┘
     ┆                   ┆              GameStarted             ┆
     ┆                   ┆ ◄───────────────────────────────────┤
     ┆                   ┆              GameStarted             ┆
     ┆ ◄─────────────────────────────────────────────────────┤
     ┆                   ┆                                      ┆
     ┆                   ┆              PerformMove             ┆
     ┆                   ├─────────────────────────────────────►
     ┆                   ┆              GameUpdated             ┆
     ┆                   ◄─────────────────────────────────────┤
     ┆                   ┆              GameUpdated             ┆
     ◄────────────────────────────────────────────────────────┤
     ┆                   ┆                                      ┆
     ┆                   ┆              PerformMove             ┆
     ┆                   ├─────────────────────────────────────►
     ┆                   ┆              MoveError               ┆
     ┆                   ◄─────────────────────────────────────┤
     ┆                   ┆                                      ┆
     ┆                   ┆              PerformMove             ┆
     ├─────────────────────────────────────────────────────────►
     ┆                   ┆          GameEnded(Winner)           ┆
     ◄────────────────────────────────────────────────────────┤
     ┆                   ┆          GameEnded(Winner)           ┆
     ┆                   ◄─────────────────────────────────────┤
     ┆                   ┆                                      ┆
```

2.2 Sending a (correct) move (details of server classes)