Eriantys Protocol Documentation

Legramandi, Leone, Locatelli
Group 60

June 29, 2022

1 Messages

1.1 Notification

This message is sent from the Server to the Client to print in Client View string messages.

Arguments

1. Notification: the string to print in Client View.

Possible responses

No response message.

1.2 Ping

This message is sent from the Server to the Client and vice versa to ensure that the connection is stable and there aren't any connection error.

Arguments

No Arguments.

Possible responses

Response with a Ping message after a time unit (i.e. 1000 ms).

1.3 ChooseYesOrNo

This message is sent from the Server to the Client in order to ask for a 'yes' or 'no' answer.

Arguments

1. Data: an integer containing '1' for 'yes' or '0' for 'no'.

1.4 ChooseInt

This message is sent from the Client to the Server, it stores a Client int choice (essentially an integer given in input by the user).

Arguments

1. Data: the integer to be sent to the Server.

Possible responses

No response.

1.5 ChooseString

This message is sent from the Client to the Server, it stores a Client String choice (essentially a String given in input by the user).

Arguments

1. Data: the string to be sent to the Server.

Possible responses

No response message.

1.6 ChooseCreateOrAddGame

This message is sent from the Server to the Client to allow the player to choose if he wants to create a new game (i.e. to be a lobby leader) or to be added to an existing game.

Arguments

No Arguments.

Possible responses

1. ChooseInt: message sent to the Server with '1' if the Client wants to be leader, else '0'.

1.7 ChooseExpert

This message is sent from the Server to the Client to allow the player to choose if he wants to play with advanced/expert or normal rules.

Arguments

No Arguments.

Possible responses

1. ChooseInt: message sent to the Server with '1' if the Client wants to play with advanced/expert rules, else '0'.

1.8 ChooseAssistantCard

This message is sent from the Server to the Client to allow the Client View to display that the Client must choose an assistant card to play the turn.

Arguments

No Arguments.

Possible responses

1. ChooseInt: message sent to the Server with the assistant card value chosen.

1.9 ChooseCloudTile

This message is sent from the Server to the Client to allow the Client View to display that the Client must choose a Cloud Tile to get new students for the entrance room.

Arguments

No Arguments.

Possible responses

1. ChooseInt: message sent to the Server with the cloud tile index chosen.

1.10 ChooseIsland

This message is sent from the Server to the Client to allow the Client View to display that the Client must choose an island to perform an action (e.g. move a student/mother nature on it)

Arguments

No Arguments.

Possible responses

1. ChooseInt: message sent to the Server with the index of the chosen island.

1.11 ChooseNickname

This message is sent from the Server to the Client to allow the Client View to display that the Client must choose a nickname for the game.

Arguments

No Arguments.

Possible responses

1. ChooseString: message sent to the Server with the chosen nickname.

1.12 ChooseLobbySize

This message is sent from the Server to the Client to allow the Client View to display that the Client must choose a lobby size (number of players in the game).

Arguments

No Arguments.

Possible responses

1. ChooseInt: message sent to the Server with the chosen lobby size between 2 or 3.

1.13 ChooseStudentColourToMove

This message is sent from the Server to the Client to allow the Client View to display that the Client must choose a student to move.

Arguments

No Arguments.

Possible responses

1. ChooseString: message sent to the Server with colour of the student to move.

1.14 ChooseWhereToMove

This message is sent from the Server to the Client to allow the Client View to display that the Client must choose if it wants to move a student to the dining room or to an island.

Arguments

No Arguments.

Possible responses

1. ChooseString: message sent to the Server with "Dining Room" or "Island"

1.15 WantToChooseCharacterCard

This message is sent from the Client to the Server to notify the TurnHandler that the Client has chosen a character card.

Arguments

1. characterCard: the index of the character card chosen.

Possible responses

1. Answer with the effect of that card, only if the player has enough coins to afford that card.

1.16 InfoForDecision

This message is sent from the Server to the Client to show through the Client View the most important info for the Client to take a decision (boards, islands...).

Arguments

No Arguments.

Possible responses

No response.

1.17 InfoMyDeck

This message is sent from the Server to the Client to show through the Client View the remaining assistant cards.

Arguments

No Arguments.

Possible responses

No response.

1.18 InfoCloudTiles

This message is sent from the Server to the Client to show through the Client View the remaining cloud tiles for the cloud tiles game phase.

Arguments

No Arguments.

Possible responses

No response.

1.19 UpdatePlayGround

This message is sent from the Server to the Client to allow the Client model to be updated.

Arguments

1. PlayGroundUpdated: this is an object that contains the updated playground with all of the info of the game.

Possible responses

No response.

1.20 Winner

This message is sent from the Server to the Client to allow the Client to know that there is a game winner and to know who that winner is.

Arguments

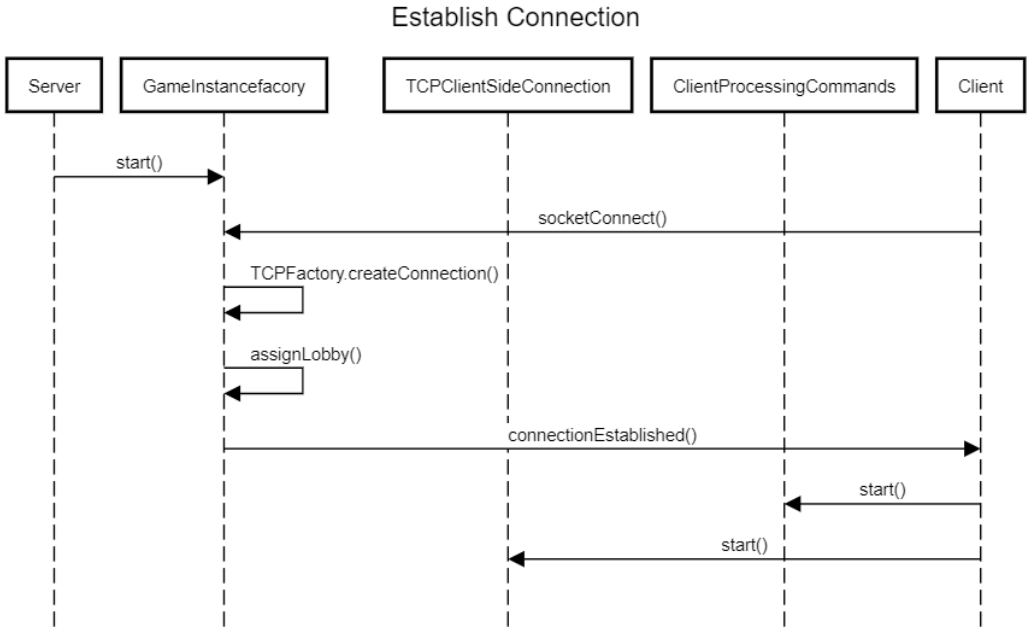
1. Winner: the nickname of the winner of the game.

Possible responses

No response.

2 Scenarios

2.1 Establishing Connection

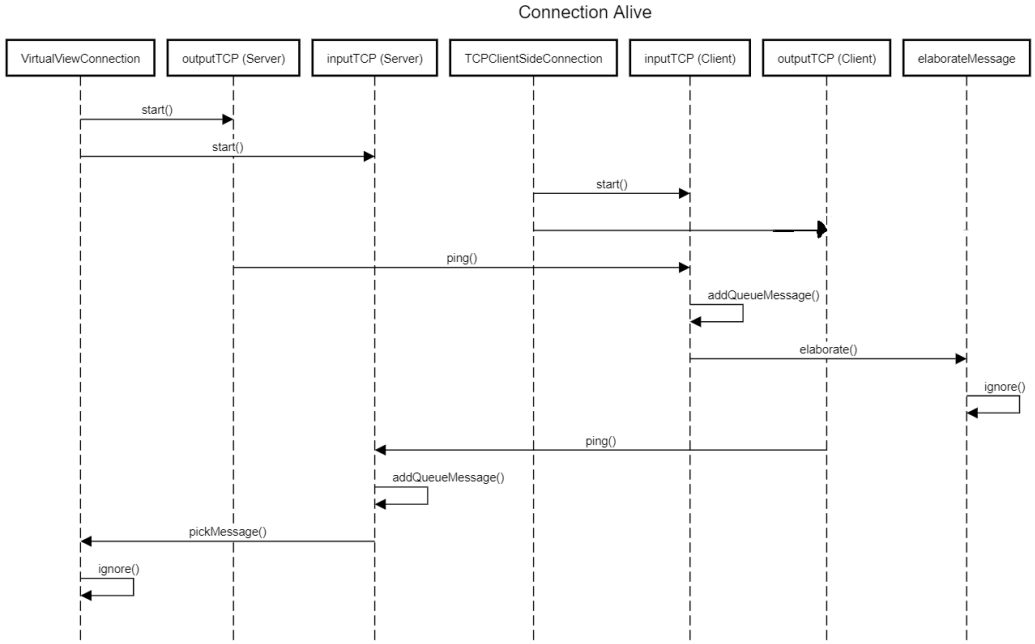


First connection of the Client to the Server. The Server starts a games factory thread which allows to listen for new socket connection, create a new TCP connection instance and responds to Client requests. The Client sends a socket TCP connection request to the Server, the Server creates a TCP connection (thanks to the factory associated) and assigns the connection

to a new lobby. Then, the Server notifies the Client that he accepts the connection. The Client starts threads for TCP Client connection, that allows to create a stable TCP connection with the Server, and for Client processing commands that allows to elaborate and implement the messages received from the Server.

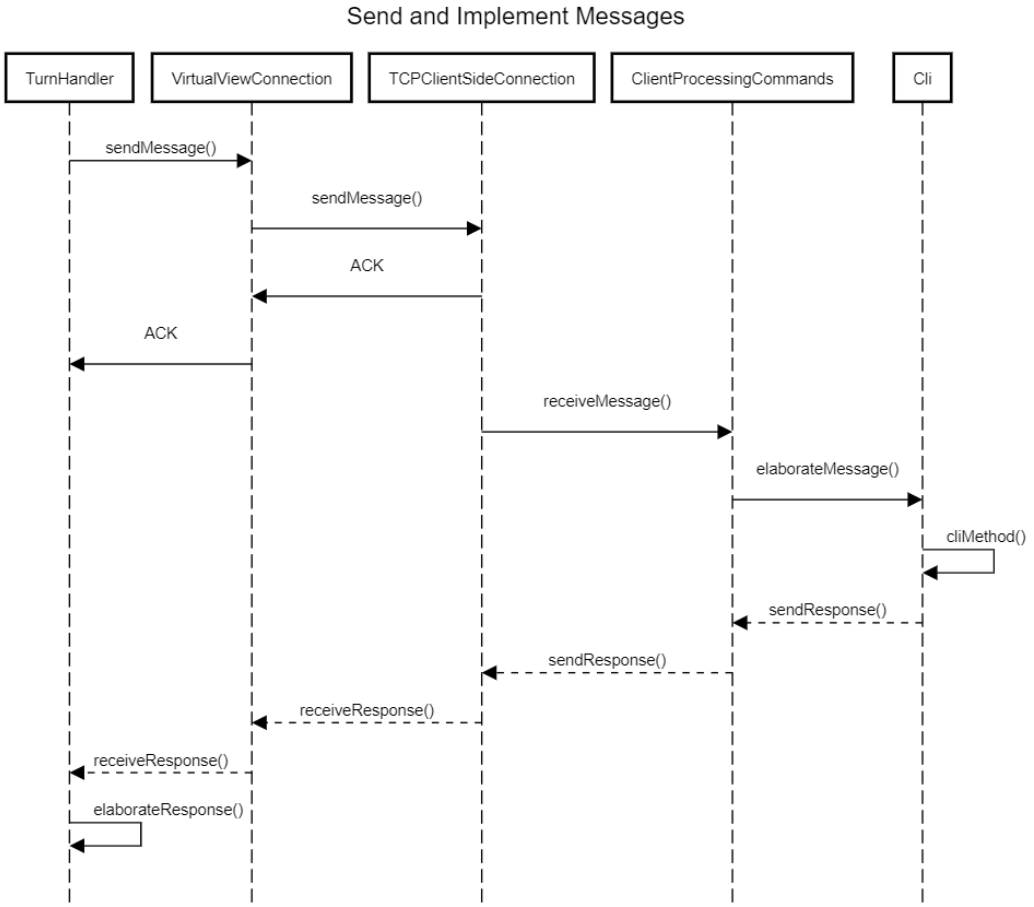
Note: int the scenario below there are more details about how we allow multiple connections: for every Client the Server creates an instance of VirtualViewConnection (this is the interface implemented by the object VirtualViewTCP) containing an object TCP for connection in outbound (OutcomingTCP) and inbound (IncomingTCP); the same logic is implemented in the Client with the class TCPClientSideConnection.

2.2 Detect Disconnection



This scenarios represent an action done in the background during all if the game. The virtual Client connection on the Server starts two threads out/in TCP for continuously send and receive ping messages. In Client side part there is the class TCP Client side connection that create the two threads out/in TCP for the Client. All ping messages are ignored. At a specific rate of few milliseconds the ping messages are sent from the Server and the Client.

2.3 Send and Implement Messages



This is the responding way to messages sent from the Server to the Client. In Server side there is the ""TurnHandler": the main thread that handles all of the actions during the game. Turnhandler uses the virtual connection specific of the Client to send a message. The TCP Client side connection handles this message and allows the Client processing commands to receive and elaborate the message content by calling the right method in View. Then the View elaborates the message respond and send it through all previous components to the Server.

Note: ACK messages are essentially the ping message explained in scenarios 2.

2.4 Server Architecture