

Eryantis Protocol Documentation

Lorenzo Forgia, Luigia Falasca, Federico Angelo Luigi Ferreri

Gruppo 13

Messages

Connect

This message is sent from the client to the server to connect clients to server.

Arguments

- Int IP port: IP of the port.
- Int IP Address: IP of the game.

Possible Responses

- Accept: Server accepts connection.

LoginMsg

This message is sent from the client to the server to create a player session.

Arguments

- String Nickname: Nickname of the player.

Possible Responses

- AnsIfAllowedMsg==Valid: Nickname unique.
- AnsIfAllowedMsg==Invalid: Nickname already taken.

AnsIfAllowedMsg

This message is sent from the server to the client when Login message has been acknowledged.

Arguments

- Status: Valid or Invalid.

Possible Responses

This message has no responses.

AskIfGameCreatedMsg

This message is sent from the server to the client to know if the game has already created.

Arguments

This message has no arguments.

Possible Responses

- AnsIfGameCreatedMsg==Invalid: the game has already created.
- AnsIfGameCreatedMsg==Valid: the game hasn't already created.

AnsGameCreatedMsg

This message is sent from the server to the client when IfGameCreated message has been acknowledged.

Arguments

- Status: Valid or Invalid.

Possible Responses

This message has no responses.

NewGameMsg

This message is sent from the client to the server when the player wants to start a new game with another player.

Arguments

- Int numplayers: number of the players in the game.
- modEXP: boolean.

Possible Responses

- BooleanCheckMsg: the message has been received.

BooleanCheckMsg

This message is sent from the server to the client when NewGame message has been acknowledged.

Arguments

- Status: OK or KO.

Possible Responses

This message has no responses.

GameStartedMsg

This message is sent from the server to the clients to notify the start of the game.

Arguments

This message has no arguments.

Possible Responses

- AnsGameStartedMsg: nicknames of the players in the game.

AnsGameStartedMsg

This message is sent from the server to the client when GameStarted message has been acknowledged.

Arguments

- Nicknames of the players.

Possible Responses

This message has no responses.

FirstPlayerTurnMsg

This message is sent from the server to the clients to notify turn of the player.

Arguments

This message has no arguments.

Possible Responses

- AnsFirstTurnPlayerMsg: general board and assistant card deck because the player has to play a card.

AnsFirstPlayerTurnMsg

This message is sent from the server to the client when FirstPlayerTurn message has been acknowledged.

Arguments

- String: Player's nickname.
- DeckCardAssistant: deck of the assistant cards.
- General Board.
- ArrayList<CardAssistant>: assistant cards have been already played.

Possible Responses

This message has no responses.

TurnDecidedMsg

This message is sent from the client to the server to notify the chosen Assistantcard.

Arguments

- AssistantCard: chosen assistantcard of the player.

Possible Responses

- AnsTurnDecidedMsg: the players' order for the round.
- AnsTurnDecidedExcMsg: the card has been already played, a new TurnDecided message is expected.

AnsTurnDecidedMsg

This message is sent from the server to the client when TurnDecided message has been acknowledged and the chosen card has not been already played.

Arguments

- ArrayList<String>: nicknames of the players.

Possible Responses

This message has no responses.

AnsTurnDecidedExcMsg

This message is sent from the server to the client when TurnDecided message has been acknowledged and the chosen card has been already played.

Arguments

- String: Player's nickname.
- DeckCardAssistant: deck of the assistant cards.
- General Board.
- ArrayList<CardAssistant>: Assistant cards have been already played.

Possible Responses

This message has no responses.

AskCAMsg

This message is sent from the client to the server to notify that the player is ready to play.

Arguments

- Nicknames of the players.

Possible Responses

This message has no responses.

AnsAskCAMsg

This message is sent from the server to the client when AskCA message has been acknowledged

Arguments

- String: Player's nickname.
- General Board.
- ArrayList<CharacterCard>: Character cards.

Possible Responses

This message has no responses.

AnsEndGameMsg

This message is sent from the server to the client when it's checked the game is finished.

Arguments

- ArrayList<String>: Nicknames of the players who won.

Possible Responses

This message has no responses.

CCMsg

This message is sent from the client to the server when the player wants to play a character card.

Arguments

- Int: character card's number the player played

Possible Responses

- AnsCannotPlayCCMsg: the player can't play a character card.
- AnsCCNotInTableExcMsg: there isn't in the game the card chosen.
- AnsPlayerAlrUsedCCExcMsg: the player has already played a character card in the same turn.
- AnsNotEnoughCoinsExcMsg: the player has not enough coins to play the character card chosen.
- AnsCC1/3/5/7/9/10/11/12Msg: the player plays one of this character cards (1 or 3,5,7,9,10,11,12).
- AnsPlayAfterCCMsg: the player plays one of the other character cards (2 or 4,6,8).

AnsCannotPlayCCMsg

This message is sent from the server to the client when CC message has been acknowledged.

Arguments

- String: player's nickname.

Possible Responses

This message has no responses.

AnsCCNotInTableExcMsg

This message is sent from the server to the client when CC message has been acknowledged.

Arguments

This message has no arguments.

Possible Responses

This message has no responses.

AnsPlayerAlrUsedCCExcMsg

This message is sent from the server to the client when CC message has been acknowledged.

Arguments

This message has no arguments.

Possible Responses

This message has no responses.

AnsNotEnoughCoinsExcMsg

This message is sent from the server to the client when CC message has been acknowledged.

Arguments

This message has no arguments.

Possible Responses

This message has no responses.

AnsCC1/3/5/7/9/10/11/12Msg

This message is sent from the server to the client when CC message has been acknowledged.

Arguments

- General Board.
- CharacterCard: Character card played.

Possible Responses

This message has no responses.

AnsPlayAfterCCMsg

This message is sent from the server to the client when CC message has been acknowledged.

Arguments

- String: player's nickname.
- General Board.
- CharacterCard: Character card played.

Possible Responses

This message has no responses.

CC1,3,5,7,9,10,11,12Msg

This message is sent from the client to the server when the player wants to play the character card 1 or 3,5,7,9,10,11,12.

Arguments

- Depends by the card played.

Possible Responses

- AnsPlayAfterNotCCMsg: the player plays the character card successfully.
- AnsCC1,3,5,7,9,10,11,12Msg: the required data are invalid, so the player has to insert the right information.

CannotPlayCCMsg

This message is sent from the client to the server when the player plays a character card even if he couldn't play it.

Arguments

- This message has no arguments.

Possible Responses

- AnsPlayAfterNotCCMsg: the player can't play a character card.

AnsPlayAfterNotCCMsg

This message is sent from the server to the client when CC message has been acknowledged.

Arguments

- String: player's nickname.
- General Board.
- CharacterCard: Character card played.

Possible Responses

This message has no responses.

MoveStudent1/2/3/4Msg

This message is sent from the client to the server when the player choses the student and location where move it.

Arguments

- Color: student chosen.
- Int: location chosen.

Possible Responses

- AnsMoveStudent1/2/3/4Msg: valid choice.
- AnsColorExc1/2/3/4Msg: invalid color.
- AnsIslandExc1/2/3/4Msg: invalid number of the island chosen.

AnsColorExc1/2/3/4Msg

This message is sent from the server to the client when MoveStudent message has been acknowledged.

Arguments

This message has no arguments.

Possible Responses

This message has no responses.

AnsIslandExc1/2/3/4Msg

This message is sent from the server to the client when MoveStudent message has been acknowledged.

Arguments

This message has no arguments.

Possible Responses

This message has no responses.

AnsMoveStudent1/2/3/4Msg

This message is sent from the server to the client when MoveStudent message has been acknowledged.

Arguments

- String: Player's nickname.
- General Board.
- ArrayList<CharacterCard>: Character cards.

Possible Responses

This message has no responses.

NumStepMNMsg

This message is sent from the client to the server when the player decides the steps of Mother Nature.

Arguments

- Int: indicates the number of steps of Mother Nature.

Possible Responses

- AnsNumStepMNMsg: the number is allowed.
- AnsNumStepMNExcMsg: the number is not allowed.

AnsNumStepMNMsg

This message is sent from the server to the client when NumStepMN message has been acknowledged.

Arguments

- String: Player's nickname.
- ArrayList<CloudTiles>: cloud tiles.

Possible Responses

This message has no responses.

AnsNumStepMNExcMsg

This message is sent from the server to the client when NumStepMN message has been acknowledged.

Arguments

- String: Player's nickname.

Possible Responses

This message has no responses.

CloudMsg

This message is sent from the client to the server when the player picks the cloud.

Arguments

- Int: it represents the cloud chosen by the player.

Possible Responses

- AnsCloudMsg: valid choice.
- AnsNumCloudExcMsg: invalid choice.

AnsNumCloudExcMsg

This message is sent from the server to the client when Cloud message has been acknowledged.

Arguments

- String: Player's nickname.

Possible Responses

This message has no responses.

AnsCloudMsg

This message is sent from the server to the client when Cloud message has been acknowledged.

Arguments

This message has no arguments.

Possible Responses

This message has no responses.

CheckTurnEndedMsg

This message is sent from the client to the server to notify that the player finished his turn.

Arguments

This message has no arguments.

Possible Responses

- AnsNewTurnMsg: the game is not finished so it has to start a new turn.
- AnsEndGameMsg: the game is finished.

AnsNewTurnMsg

This message is sent from the server to the client when CheckTurnEnded message has been acknowledged.

Arguments

- String: Player's nickname.

Possible Responses

This message has no responses.

NewOrderMsg

This message is sent from the client to the server to notify that the player starts a new turn.

Arguments

This message has no arguments.

Possible Responses

- **AnsNewOrderMsg**: new turn is started so the player has to play an assistant card.

AnsNewOrderMsg

This message is sent from the server to the client when NewOrder message has been acknowledged.

Arguments

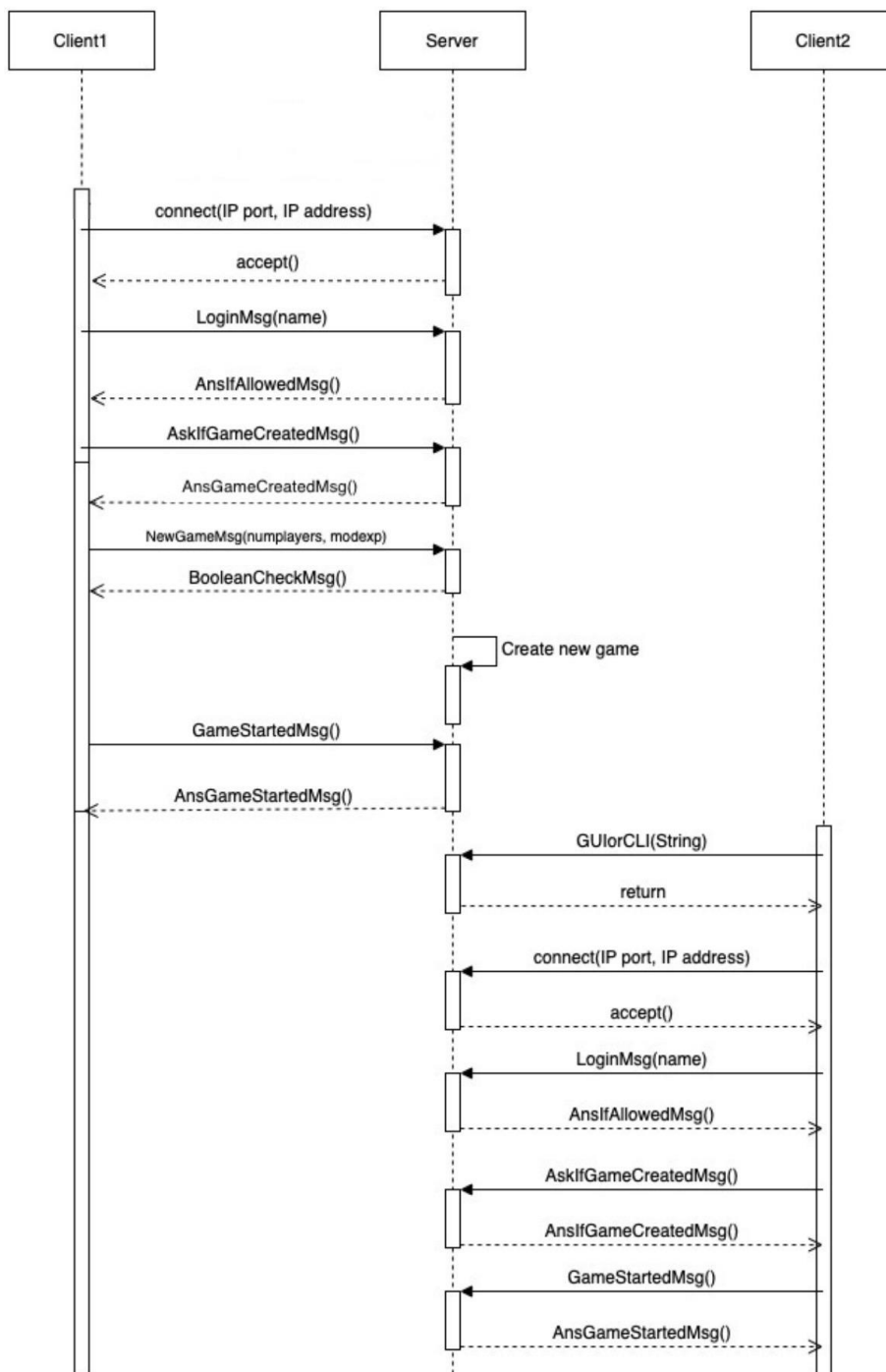
- **String**: Player's nickname.
- **DeckCardAssistant**: deck of the assistant cards.
- **General Board**.
- **ArrayList<CardAssistant>**: Cards assistant have been already played.

Possible Responses

This message has no responses.

Login

LOGIN



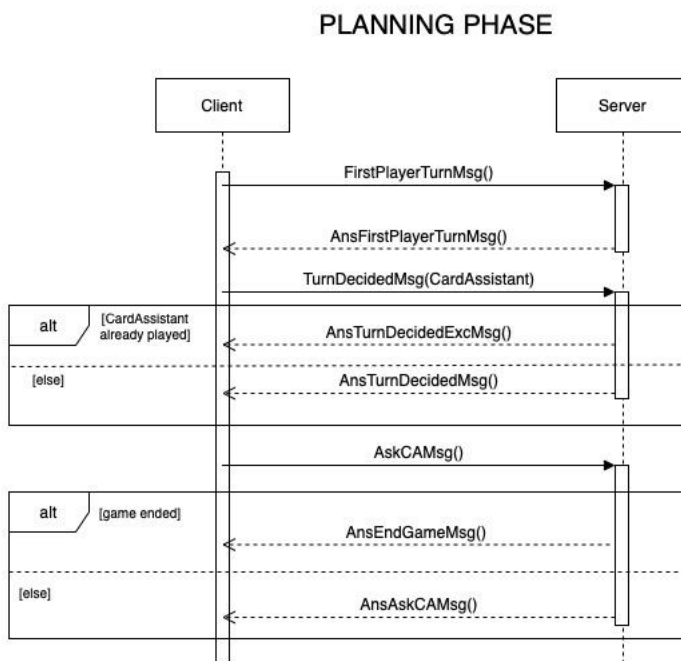
The client sends the IP address and port through the Connect message to create a socket.

Then the Login message is sent with the nickname inserted by the client. The server responds with the AnsIfAllowed message. If the server responds with the Status Valid of the AnsIfAllowed message, the connection is ready to accept game-related messages.

At this point the client sends a AskIfGameCreated message to know if the game has been already created or not. The response to the client1 will be an AnsGameCreated message with Status Valid

because the game hasn't been already created while in the case of client2 the Status of the message will be Invalid because the game has been already created. So the client1 sends NewGame message to server to create a new game and then sends the GameStarted message while the client2 sends directly a GameStarted message to notify that he is ready to play. After all players joined the game, the server, with AnsGameStarted message, notifies all clients the start of the game and it shows the name of the players in the game.

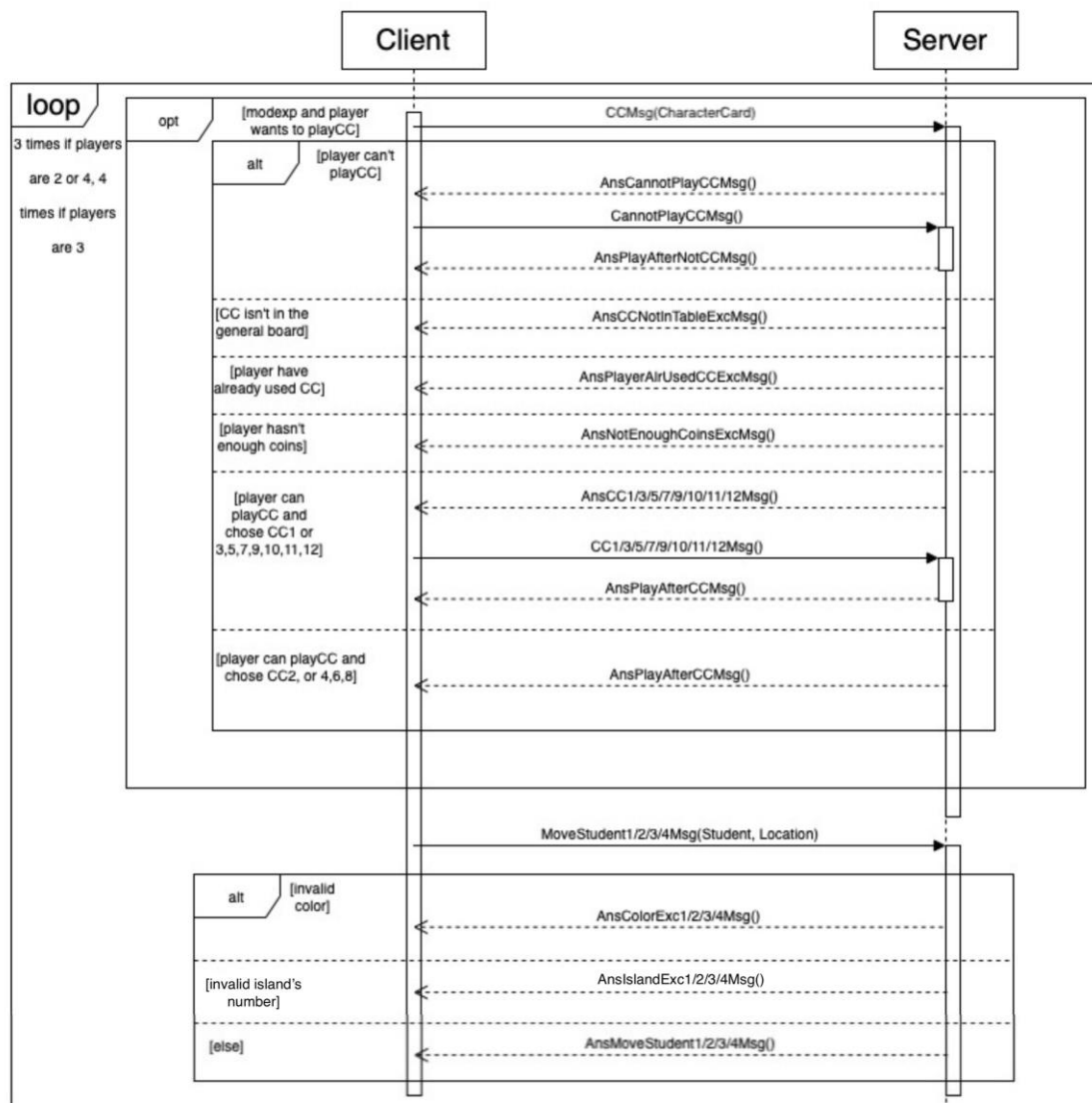
Planning Phase



When all the clients send the FirstPlayerTurn message to notify they are waiting to play, the server chooses the player's order randomly to play the card assistant and then it sends the AnsFirstPlayerTurn message to the client to notify it's his turn, so the client sends the chosen CardAssistant to server (TurnDecided message) which checks if the card has been already played (AnsTurnDedicedExc message) or not (AnsTurdDecided message) and sends a response. Then, after that all the players played an assistant card, the server checks the players' order of the round. The client notifies the server he is ready to play his turn (AskCA message) and when it's his turn the server checks if the game is ended (AnsEndGame message) or not (AnsAskCA message) and he can continue and plays his turn.

Action Phase 1

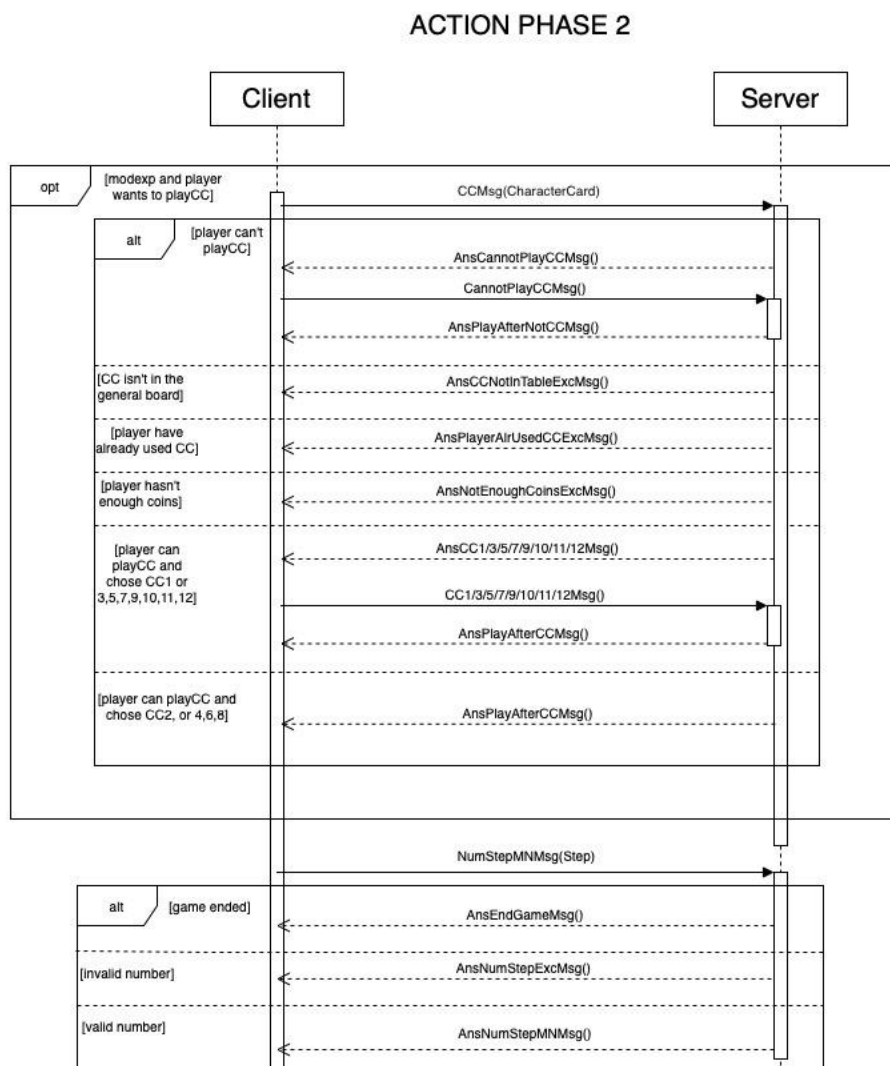
ACTION PHASE 1



The server checks if the game is in expert mode and, in that case, it asks if the player wants to play a character card: if the answer is yes, the player chooses the card and sends it to the server (CC message), otherwise the player chooses the student to move from the entrance (MoveStudent message). If the player plays a character card there could be different responses by the server: if the player can't play any cards, the server notifies that to the client (AnsCannotPlayCC message) then the client notifies the server he received the message and he can continue to play (CannotPlayCC and AnsPlayAfterNotCC messages); if the player chooses a card's number which it doesn't match with any cards he can play, the server notifies that and asks for another card (AnsCCNotInTableExc message); if the player has already played a character card in the same turn, the server notifies that because he can't play another card (AnsPlayerAlrUsedCCExc message), then the client notifies the server he received the message and he can continue to play (CannotPlayCC and AnsPlayAfterNotCC messages); if the player hasn't enough coins to play the card, the server notifies that and asks for another card (AnsNotEnoughCoinsExc message); if the

player can play the chosen card and he plays one of the character card 1,3,5,7,9,10,11,12 the server notifies what is required by the card then when it has been played successfully the server notifies it and the game can continue (AnsCC1—12, CC1—12, AnsPlayAfterCC messages); if the player can play the chosen card and he plays one of the character card 2,4,6,8 the server notifies it has been played successfully and the game can continue (AnsPlayAfterCC message) . At this point the client chooses a student and the location which could be an island tiles or the dining room and sends it to server (MoveStudent1—4 message). Then the server checks if the player can make that movement and if the choice is possible, it sends AnsMoveStudent1—4 message to the client, if it's not possible it sends AnsColorExc1—4 message if the student's color is invalid or AnsIslandExc1—4 if the island's number is invalid and so the server asks again the move to the client. This procedure is repeated three times when the players in the game are 2 or 4 and four times when they are 3. The messages 1—4 matches with the first, second, third and four student's moves.

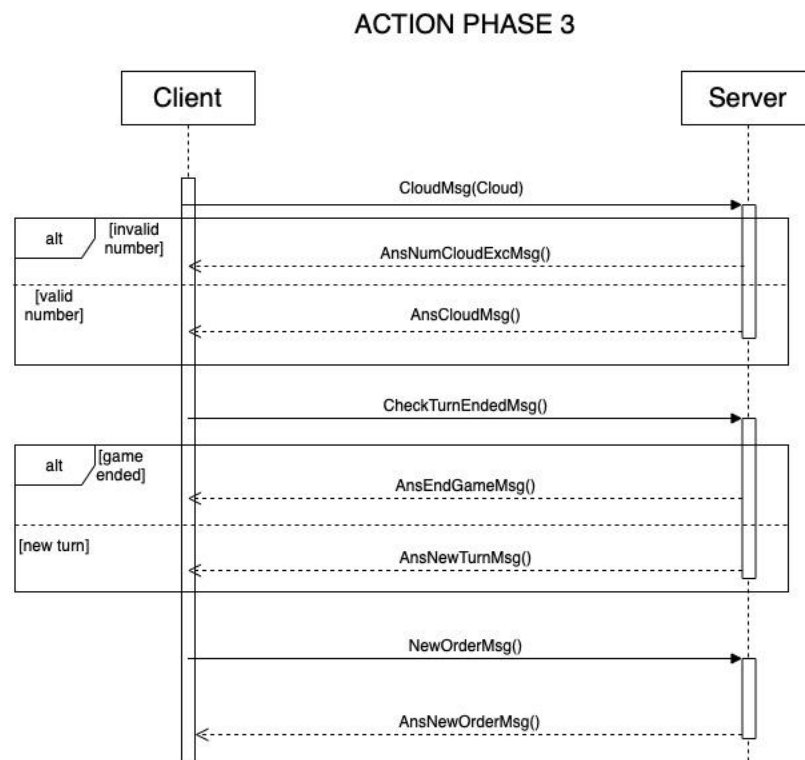
Action Phase 2



The server checks if the game is in expert mode and, in that case, it asks if the player wants to play a character card: if the answer is yes, the player chooses the card and sends it to the server (CC

message), otherwise the player chooses the steps that Mother Nature has to do (NumStepMN message). If the player plays a character card there could be different responses by the server: if the player can't play any cards, the server notifies that to the client (AnsCannotPlayCC message) then the client notifies the server he received the message and he can continue to play (CannotPlayCC and AnsPlayAfterNotCC messages); if the player chooses a card's number which it doesn't match with any cards he can play, the server notifies that and asks for another card (AnsCCNotInTableExc message); if the player has already played a character card in the same turn, the server notifies that because he can't play another card (AnsPlayerAlrUsedCCExc message), then the client notifies the server he received the message and he can continue to play (CannotPlayCC and AnsPlayAfterNotCC messages); if the player hasn't enough coins to play the card, the server notifies that and asks for another card (AnsNotEnoughCoinsExc message); if the player can play the chosen card and he plays one of the character card 1,3,5,7,9,10,11,12 the server notifies what is required by the card then when it has been played successfully the server notifies it and the game can continue (AnsCC1—12, CC1—12, AnsPlayAfterCC messages); if the player can play the chosen card and he plays one of the character card 2,4,6,8 the server notifies it has been played successfully and the game can continue (AnsPlayAfterCC message) . At this point the player chooses the steps that Mother Nature has to do (NumStepMN message), the server controls if the steps are allowed and if the game is finished: if they are allowed and the game isn't finished it sends the message AnsNumStepMN, if the steps are allowed and the game is finished it sends the message AnsEndGame, if the steps are not allowed and it asks again for the steps (AnsNumStepExc message). At this stage the game can end because all the towers are placed or because there are three or less than three groups of Island. So, the server will check if the game is in this condition.

Action Phase 3



The client picks the cloud and sends it through the message Cloud, the server checks if the cloud's number is allowed or not: if the number is invalid the server notifies it to the client and asks again the cloud's number (AnsNumCloudExc message), if the number is valid the server sends AnsCloud message to the client to notify it. Then the client communicates he terminated his turn to the server, and he waits until all the players terminate their turn (CheckTurnEnded message). So, when all the players played, the server checks if the game is finished: if the game is finished it sends the message AnsEndGame, otherwise it sends the message AnsNewTurn to notify the client that a new turn has to start, the client notifies the server he is ready to play the new turn (NewOrder message) so the server asks to the client to play the assistant card for the new turn (AnsNewOrder message).

End Game



The server sends the message AnsEndGame to the client to notify the game is finished and shows who won.