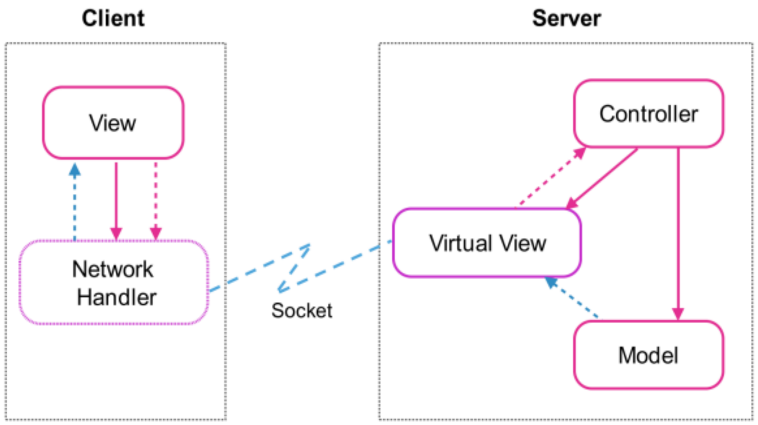
Eryantis Protocol Documentation

Christian Giosia, Davide Galbiati, Giuseppe Gentile

Group 62

MVC + Virtual View



Messages

LoginRequest

Message used by the client to request a login to the server.

Arguments

- Nickname: nickname of the player's client

Responses

- LoginReply: always sent

LoginReply

Message used to confirm or discard a login request of a client.

Arguments

- NicknameAccepted: (boolean) true if the nickname is valid
- connectionSuccessful: (boolean) if the connection is established

AddStudentFromCloudToWaitingMessage

Message used to send the cloud chosen by the player to the server and move the students on the cloud to waiting room.

Arguments

- Nickname: nickname of the player’s client who must move the students
- cloudIndex: the index of the clouds in the game
- TextMessage: sent to two players: to one saying that his turn is over, to the other that is starting

PlayerNicknameMessage

Message used to register a player in the game by sending some parameters that will identify him.

Arguments

- Nickname: nickname of the player’s client
- numPlayers: number of players in the game
- colorTower: color of the tower chosen
- gameMode: modality of the game

Possible Responses

- InvalidTowerMessage: sent if the color chosen is already taken
- InvalidNicknameMessage: sent if nickname is already taken
- TowerMessage: sent if login and tower are correct
- AssignPlayerDeckResponseMessage: sent if login and tower are correct
- DisplayHallMessage: sent if login and tower are correct
- DisplayEntranceMessage: sent if login and tower are correct
- CloudsMessage: sent if login and tower are correct
- DisplayIslandMessage: sent if login and tower are correct

InvalidTowerMessage

Message used by the server to the client to notify of the wrong tower choice.

Arguments

- Nickname: nickname of the player’s client who requested wrong tower

InvalidNicknameMessage

Message used by the server to the client to notify the invalid nickname chosen.

Arguments

- Nickname: nickname of the player’s client

TowerMessage

Message used by the server to the client to show the number of towers.

Arguments

- Nickname: nickname of the player’s client
- colorTower: the color of the tower chosen
- towerNumber: the number of towers assigned

AssignPlayerDeckResponseMessage

Message used by the server to the client to show the deck of the player.

Arguments

- Nickname: nickname of the player’s client

- assistantCardsModel: the deck of the player

DisplayHallMessage

Message used by the server to the client to show the hall of the player.

Arguments

- Nickname: nickname of the player’s client
- hall: the hall of the player

DisplayEntranceMessage

Message used by the server to the client to show the entrance of the player.

Arguments

- Nickname: nickname of the player’s client
- entrance: the entrance of the player

DisplayCloudsMessage

Message used by the server to the clients to show the clouds of the board game.

Arguments

- Nickname: nickname of the player’s client
- clouds: the clouds of the board of the game

PlayAssistantCardMessage

Message used by the clients to the server when a card is played.

Arguments

- Nickname: nickname of the player’s client
- assistantCard: the card played

Possible Responses

- DisplayCemeteryMessage: message to show to the client the cemetery of cards

DisplayCemeteryMessage

Message used by the server to the client when a card is played, to show the cemetery.

Arguments

- Nickname: nickname of the player’s client
- cemetery: the list of cards played

StudentToIslandMessage

Message used by the clients to the server when students are moved to island from the entrance.

Arguments

- Nickname: nickname of the player’s client
- students: the students to be moved from entrance to an island
- indexIsland: the index of the island of the game

Possible Responses

- DisplayIslandMessage: always sent

DisplayIslandMessage

Message used by the server to the client when player move to island some students.

Arguments

- Nickname: nickname of the player’s client
- islandModel: the island to move the students on
- islandIndex: the index of the island of the game

StudentToHallMessage

Message used by the clients to the server when students are moved to hall from the entrance.

Arguments

- Nickname: nickname of the player’s client
- students: the students to be moved from entrance to hall

Possible Responses

- DisplayHallMessage: always sent
- InvalidNumberStudentsMovedMessage: sent if the player didn’t move in total three students

InvalidNumberStudentsMovedMessage

Error message used by the server to the client when the player didn’t move in total three students.

Arguments

- Nickname: nickname of the player’s client

DisplayIslandMessage

Message used by the server to the client when player move to island some students.

Arguments

- Nickname: nickname of the player’s client
- islandModel: the island to move the students on
- islandIndex: the index of the island modified of the game

MoveMotherNatureMessage

Message used by the clients to the server when mother nature is moved.

Arguments

- Nickname: nickname of the player’s client
- movement: the movement of mother nature

Possible Responses

- InvalidMovementMessage: sent if the movement is greater than the movement of the card played
- DisplayIslandMessage: sent if the influence is changed, so the tower is set on the island
- TowerMessage: sent if the influence is changed, so the tower of the player is changed
- TextMessage: sent if island can be joined. Text saying “Joining islands”

AddStudentFromCloudToWaitingMessage

Error message used by the server to the client when the player didn’t move in total three students.

Arguments

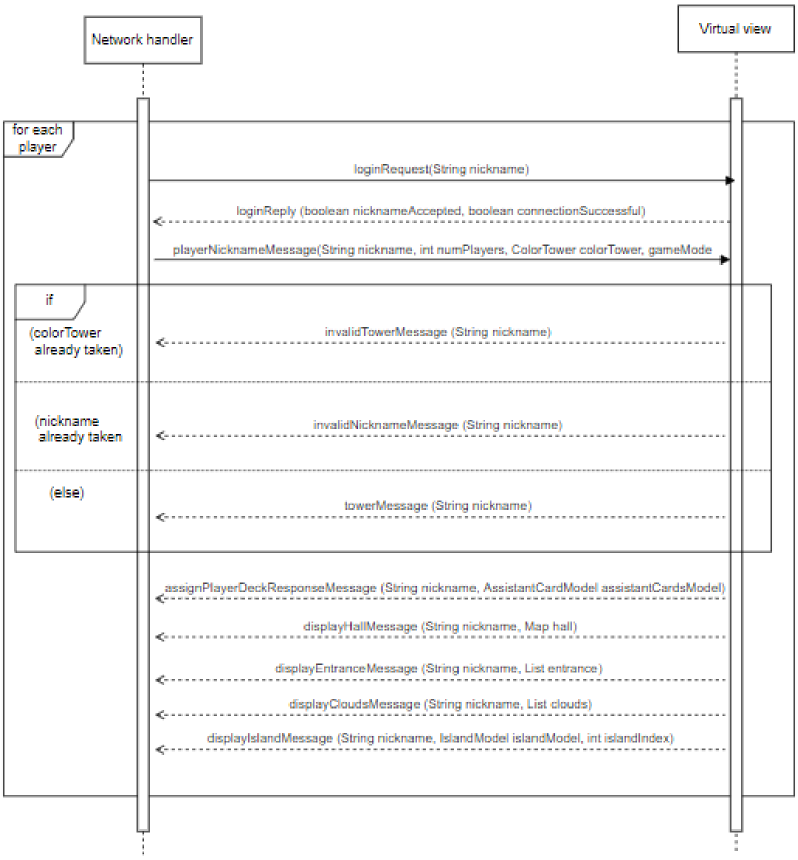
- Nickname: nickname of the player’s client

Scenarios

The following sequence diagrams explain the main game's phases. They are divided into login phase, planning phase, action phase. Each diagram shows in detail every messages' exchange between client and server.

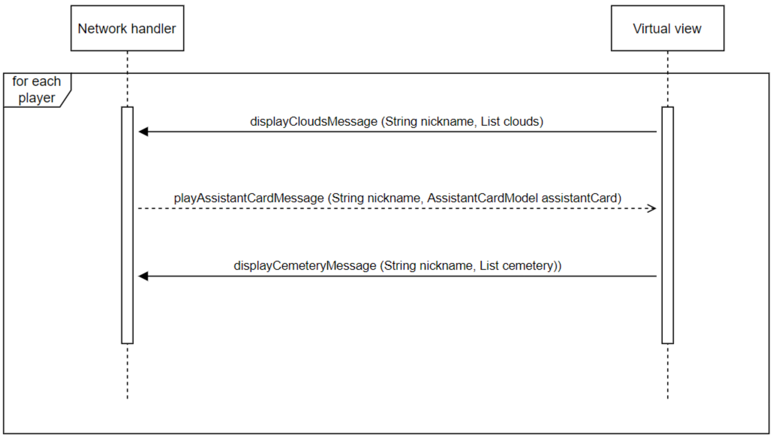
Login Phase

This is the first phase of the game in which players request a new game. When the client gets the server's response, each player chooses nickname, number of players, tower colour and game mode (simplified or expert). If even just one was already selected, the server sends an error message. When all the parameters are set, the server assigns a deck to each player and displays the game's board which includes Islands, Clouds and the player's board.


Network handler
Virtual view
for each player
loginRequest(String nickname)
loginReply (boolean nicknameAccepted, boolean connectionSuccessful)
playerNicknameMessage(String nickname, int numPlayers, ColorTower colorTower, gameMode
if
(colorTower already taken)
invalidTowerMessage (String nickname)
(nickname already taken
invalidNicknameMessage (String nickname)
(else)
towerMessage (String nickname)
assignPlayerDeckResponseMessage (String nickname, AssistantCardModel assistantCardsModel)
displayHallMessage (String nickname, Map hall)
displayEntranceMessage (String nickname, List entrance)
displayCloudsMessage (String nickname, List clouds)
displayIslandMessage (String nickname, IslandModel islandModel, int islandIndex)

Planning Phase

During this phase every player has to choose an assistant card the decides the order in which each player makes a move and by how many islands mother nature can be moved. The server sends a message to display the possible actions that a player can make regarding the clouds, the client sends a message (for each player) to the server with the assistant card chosen and then the server sends back a message to show to the player the "cemetery" that is where the cards are discarded after being used.

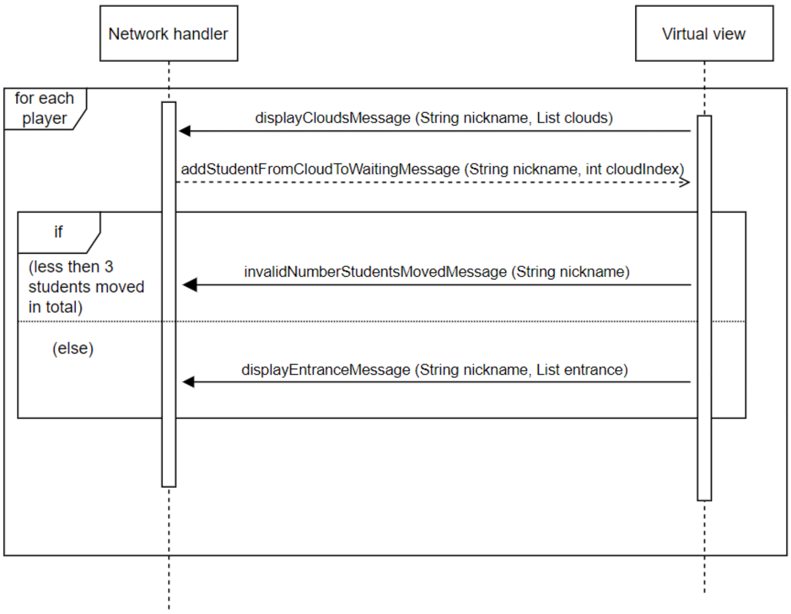

Network handler
Virtual view
for each player
displayCloudsMessage (String nickname, List clouds)
playAssistantCardMessage (String nickname, AssistantCardModel assistantCard)
displayCemeteryMessage (String nickname, List cemetery))

Action Phase

During the action phase each player moves mother nature and the students, trying to build towers on the islands. This phase is divided into 3 phases.

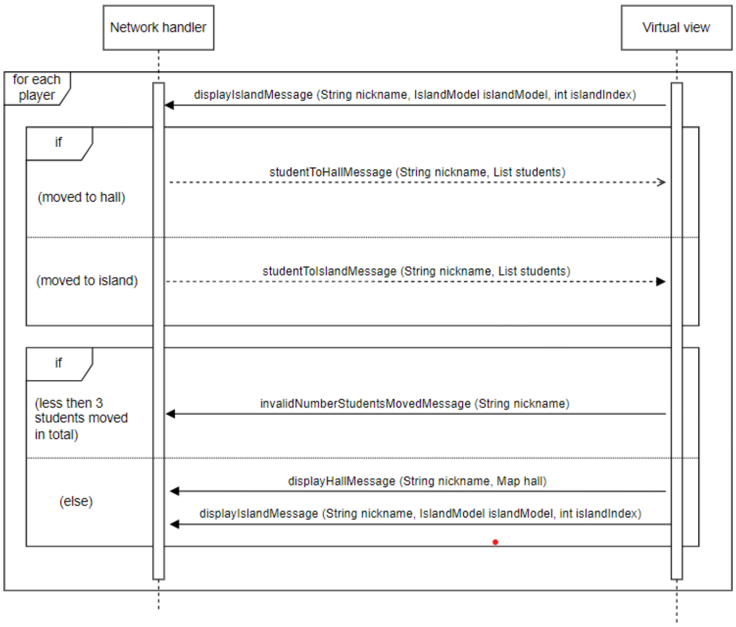
- ### Phase 1

  The server sends a massage to display the possible action regarding the clouds. Each player, during this phase must move three students from a cloud to the "waiting room" (the entrance of the player's board). If a player moves less than three the servers sends an error message to the client (invalidNumberStudentsMessage) or else it sends a message to display the possible actions regarding the waiting room.


Network handler
Virtual view
for each player
displayCloudsMessage (String nickname, List clouds)
addStudentFromCloudToWaitingMessage (String nickname, int cloudIndex)
if
(less then 3 students moved in total)
invalidNumberStudentsMovedMessage (String nickname)
(else)
displayEntranceMessage (String nickname, List entrance)

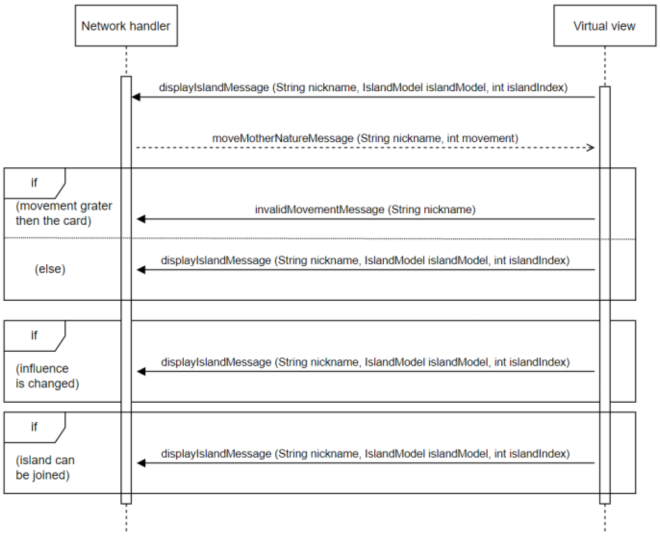
- ### Phase 2

  The server sends a message to show the possible action regarding the islands, then each player can choose to move the students from the waiting room to the hall (studentTo HallMessage) or to an island(studentToIslandMessage). The server sends an error message if less than three students are moved in total or sends messages to display the possible action regarding the hall and the islands.


Network handler
Virtual view
for each player
displayIslandMessage (String nickname, IslandModel islandModel, int islandIndex)
if
(moved to hall)
studentToHallMessage (String nickname, List students)
(moved to island)
studentToIslandMessage (String nickname, List students)
if
(less then 3 students moved in total)
invalidNumberStudentsMovedMessage (String nickname)
(else)
displayHallMessage (String nickname, Map hall)
displayIslandMessage (String nickname, IslandModel islandModel, int islandIndex)

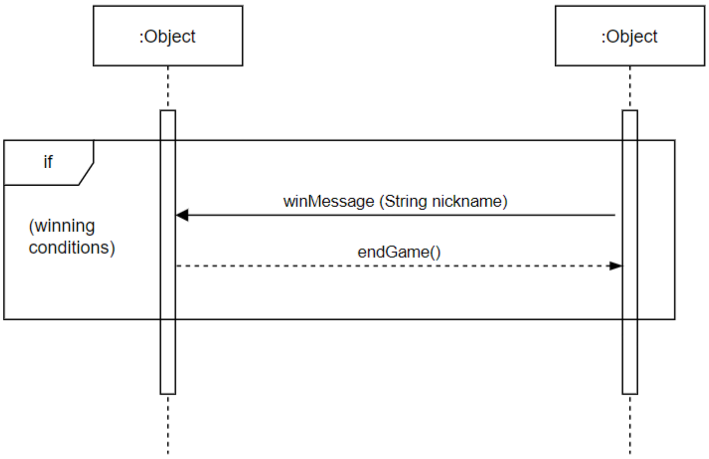
- ## Phase 3

  This is the last part of the action phase. The client sends a message to move mother nature, the server sends an error message if the movement is not valid. Every time the influence changes or two islands can be joined, the server sends a message to display the possible actions regarding the islands.


Network handler
Virtual view
displayIslandMessage (String nickname, IslandModel islandModel, int islandIndex)
moveMotherNatureMessage (String nickname, int movement)
if
(movement grater then the card)
invalidMovementMessage (String nickname)
(else)
displayIslandMessage (String nickname, IslandModel islandModel, int islandIndex)
if
(influence is changed)
displayIslandMessage (String nickname, IslandModel islandModel, int islandIndex)
if
(island can be joined)
displayIslandMessage (String nickname, IslandModel islandModel, int islandIndex)

Game’s end

When the winning conditions are matched, the server sends a message to each player showing the nickname of the winner, then the client sends the message of end game.


:Object
:Object
if
(winning conditions)
winMessage (String nickname)
endGame()