Eryantis Protocol Documentation

Christian Giosia, Davide Galbiati, Giuseppe Gentile

Gruppo 62

# 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 send the cloud chosen by the player and move the students on the cloud to waiting room.

### 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 studends

### 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

(Add sequence diagram picture here)

Description of what happens in this scenario.