

Eryantis Protocol Documentation

Demiraj Arbi, Carlini Edoardo, De Paola Francesco
Group AM13

April 29, 2022

1 Messages

1.1 Serialization

For the messages we use Java Serialization and Deserialization by creating a class for each message that implements Serializable

1.2 Client Messages

1.2.1 LoginMessage

This message is sent from the client to the server after establishing a connection to create a player session

Arguments

1. Nickname

Possible responses

- LobbyMessage: it contains the games already created that the player can join
- ErrorMessage: it contains the reason of the error, for example: Nickname not unique

1.2.2 EndTurnMessage

This message is sent from the client to the server to notify the end of the turn

Arguments

1. Nickname

Possible responses

- SuccessMessage
- ErrorMessage: it contains the reason of the error

1.2.3 NewGameMessage

This message is sent from the client to the server when the user wants to create a new game

Arguments

1. NumPlayers: the maximum number of player of the game created

Possible responses

- SuccessMessage: the game has been created
- ErrorMessage: it contains the reason of the error, for example invalid number of players

1.2.4 JoinGameMessage

This message is sent from the client to the server when the user wants to join a game

Arguments

1. gameId: the id of the game that the player wants to join

Possible responses

- SuccessMessage: the player has joined
- ErrorMessage: it contains the reason of the error, for example invalid game id

1.2.5 ReloadGameMessage

The player wants to reload a game that recently went down (Persistence)

Arguments

The message has no arguments

Possible responses

- SuccessMessage: the game has been reloaded
- ErrorMessage: it contains the reason of the error

1.2.6 MoveStudentMessage

This message is sent from the client to the server when the user wants to move a student

Arguments

1. From: the object where is the student
2. Color: the color of the student to move
3. To: the object where the player wants to move the student

Possible responses

- SuccessMessage: the student has been moved
- ErrorMessage: it contains the reason of the error

1.2.7 MoveMotherNatureMessage

This message is sent from the client to the server when the player wants to move mother nature

Arguments

1. Steps: the number of steps the player wants to move mother nature

Possible responses

- SuccessMessage: mother nature has been moved
- ErrorMessage: it contains the reason of the error, for example: invalid number of steps

1.2.8 ChooseCloudMessage

The player choose the cloud he wants to get the students from.

Arguments

1. CloudId: the id of the cloud the player selected

Possible responses

- SuccessMessage
- ErrorMessage: it contains the reason of the error

1.2.9 PlayCardMessage

This message is sent from the client to the server when the user wants to play an assistant card

Arguments

1. AssistantCardId: the id of the assistant card played

Possible responses

- SuccessMessage: the card has been correctly played
- ErrorMessage: it contains the reason of the error, for example: card already played in this turn

1.2.10 ActivateCharacterMessage

This message is sent from the client to the server when the user wants to buy a character

Arguments

1. EffectId: the id of the card effect the player selected

Possible responses

- AskIslandMessage: card effect correctly activated, asks the player to choose an island for the effect
- AskStudentMessage: card effect correctly activated, asks the player to choose a student for the effect
- SuccessMessage: the card has been correctly activated
- ErrorMessage: it contains the reason of the error, for example: not enough coins

1.2.11 DisconnectMessage

This message is sent from the client to the server when the user disconnects

Arguments

1. Nickname

Possible responses

- SuccessMessage

1.2.12 ChooseTowerColorMessage

This message is sent from the client to the server

Arguments

1. TowerColor: the chosen tower color

Possible responses

- MoveStudentMessage

1.2.13 IslandEffectMessage

This message is sent from the client to the server as a response of the Ask-IslandMessage

Arguments

1. chosenIsland: the island where he wants to move or that he selected

Possible responses

- SuccessMessage: card effect correctly activated
- ErrorMessage: it contains the reason of the error

1.2.14 StudentEffectMessage

This message is sent from the client to the server as a response of the AskStudentMessage

Arguments

1. chosenStudent: the student where he wants to move or that he selected

Possible responses

- SuccessMessage: card effect correctly activated
- ErrorMessage: it contains the reason of the error

1.3 Server Messages

1.3.1 ErrorMessage

This message is sent from the server to the client to notify an error to the user

Arguments

Reason: the reason of the error

1.3.2 SuccessMessage

This message is sent from the server to the client to notify that the command has been executed successfully

1.3.3 GameInfoMessage

This message is sent from the server to the client and contains some current info of the game

Arguments

1. CurrentPlayer: the nickname of the player currently playing
2. CurrentActivePlayers: the list of the nicknames of the players currently active
3. ActivatedCharacters: the list of the characters of the players currently active

1.3.4 DisconnectionMessage

This message is sent from the server to the client to notify a disconnection of a player

Arguments

1. Nickname: the nickname of the player disconnected

1.3.5 GenericMessage

This message is sent from the server to the client to notify generic informations to the user

Arguments

1. Message: the message you want to notify to the user

1.3.6 StartTurnMessage

This message is sent from the server to the client to notify the start of the turn

Arguments

1. Phase: action or preparation

1.3.7 LobbyMessage

This message contains the lobby with the games that the user can join

Arguments

1. GameId: the id of the game
2. NumPlayers: the number of players currently in the game

Possible responses

- PingMessage

1.3.8 AskStudentMessage

This message is sent from the server to the client to ask the client to move a selected student

Arguments

1. Students: list of the students the player can move
2. NumStudents: number of students he can move

Possible responses

- MoveStudentMessage

1.3.9 AskIslandMessage

This message is sent from the server to the client to ask the client to select an island

Arguments

1. Students: list of the students the player can move

Possible responses

- MoveStudentMessage

1.3.10 AskTowerColorMessage

This message is sent from the server to the client to ask the client to move a selected student

Arguments

1. AvailableColors: contains the list of the color the player can choose

Possible responses

- ChooseTowerColorMessage

1.3.11 GameEnd

This message is sent when the game ends

Arguments

1. Winner: the nickname of the winner player

Possible responses

- This message has no responses

1.4 PingMessage

This message is sent from the client to the server or from the server to the client to verify the network is still working, used to keep the connection alive

Arguments

This message has no arguments

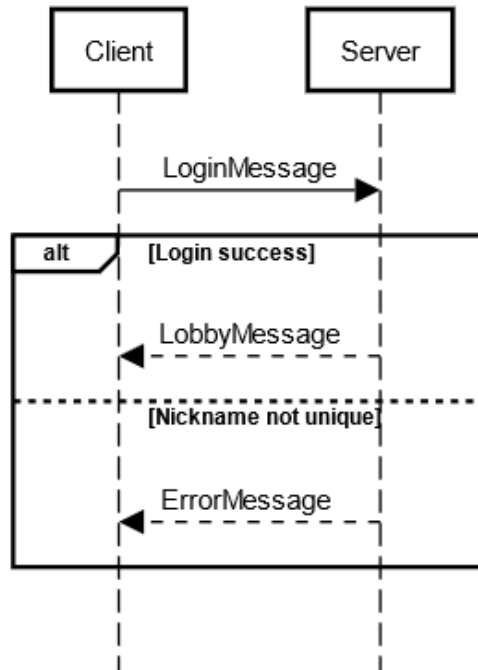
Possible responses

- PingMessage: response to confirm the network is still working

2 Scenarios

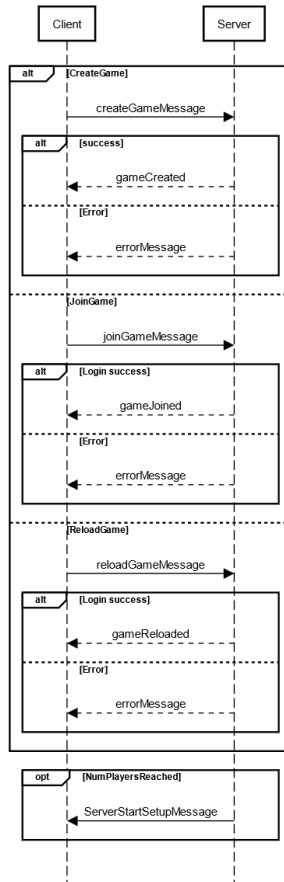
2.1 User connection and Login

2.1.1 Login



The connection starts with the Login, if it goes well the Server sends to the client the `LobbyMessage` so the user can see the game he can join.

2.1.2 User menu

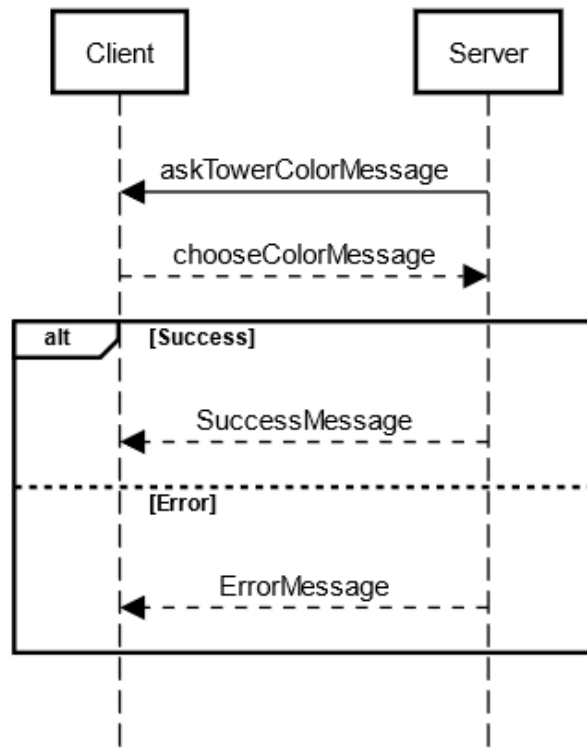


The user has 3 options:

1. Create a new game: the user can create a new game and has to insert the number of player he wants (2 or 3), the clients sends a Create-NewGameMessage which contains the number of player selected .
2. Join a game: the user selects a game he wants to join and send a JoinGameMessage which contains the id of the game he wants to join.
3. Reload a game: the user can reload a game that recently crashed (persistence).

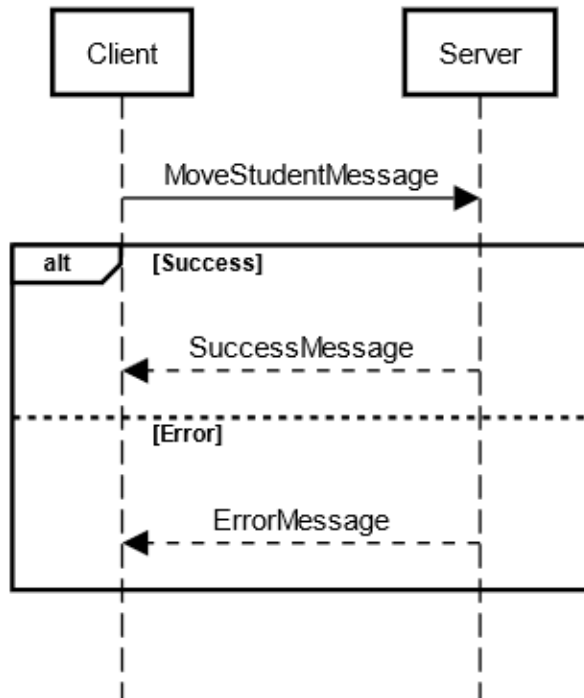
When a game reaches the number of player it starts, the server sends a message to the clients to notify the start of the game.

2.2 Setup Phase



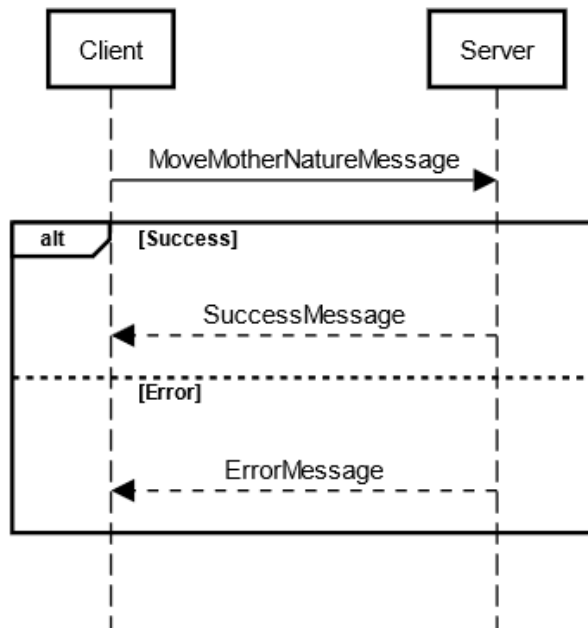
After the game starts, during the setup phase the server asks to all the clients to choose a tower color by sending a message with the available colors

2.3 Move Student



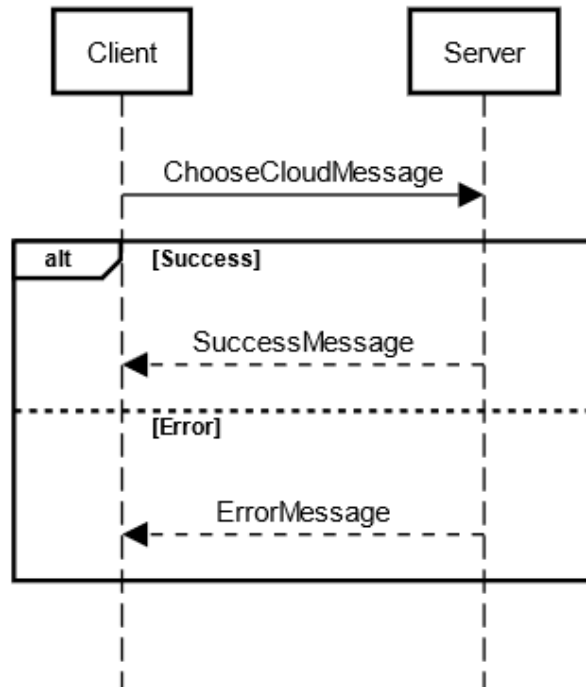
When a player wants to move a student sends a message containing From (the object where is currently the student), Color (the student color), To (Where he wants to move the student). If the student is correctly moved the server sends a SuccessMessage, if there is an error the server sends an error message containing the reason

2.4 Move mother nature



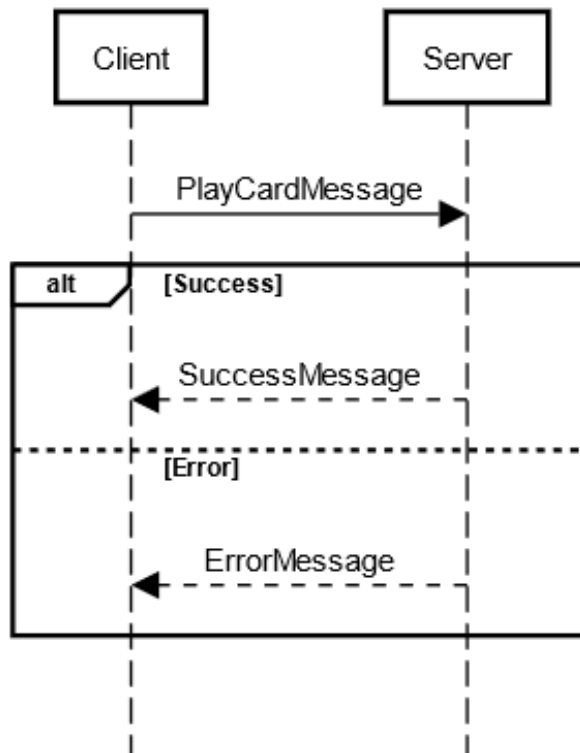
When a player wants to move mother nature sends a message containing the number of steps he wants to move mother nature. If mother nature is correctly moved the server sends a SuccessMessage, if there is an error the server sends an error message containing the reason of the error.

2.5 Choose cloud



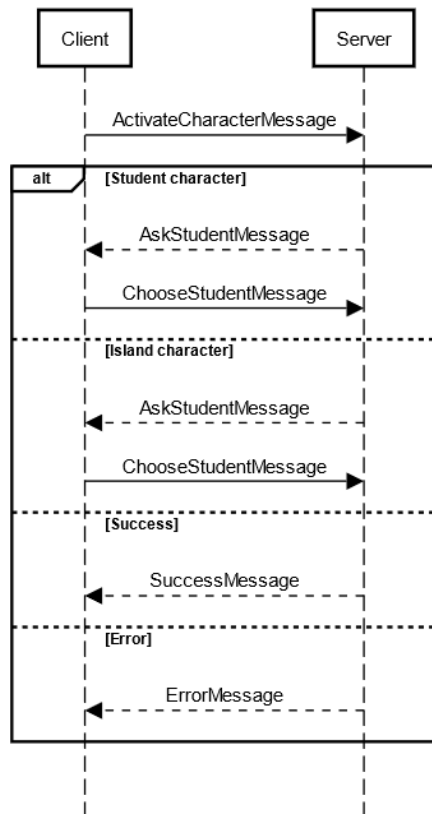
At the start of the action phase a player has to choose a cloud to remove the students from, the clients sends a message containing the id of the cloud he wants to choose.

2.6 Play assistant card



When the user has to play an assistant card he sends a message containing the id of the assistant card he wants to play

2.7 Activate character



When the player decides to activate a character card a message containing the id is sent, the server can reply in four ways:

1. `SuccessMessage`: the character has been correctly activated
2. `AskStudentMessage`: the selected character needs to select the students/ the student to move
3. `AskIslandMessage`: the selected character needs to select an island
4. `ErrorMessage`: for example the player has not enough coins for the selected character

So the player has to reply by sending a `ChooseStudentMessage` or `ChooseIslandMessage`(for the first effect the server will ask for an island and also for a student) containing the selected Island or Student after the server asked.