

Eryantis Protocol Documentation

Finazzi Alessandro, Gallazzini Andrea, Lancini Andrea

Gruppo 48

Messages

ServerMessage

This message is sent from the server to the client when the client must perform an action.

Arguments

- Request: the information required to the client.

Possible Responses

- LoginMessage: if the request value is LOGIN.
- PlayerNumber: if the request value is PLAYERNUMBER.
- Difficult: if the request value is DIFFICULT.
- PlayAssistantCard: if the request value is PLAY_ASSISTANT_CARD.
- MoveStudentDining: if the request value is MOVE_STUDENT and the player wants to move a student into the dining room.
- MoveStudentIsland: if the request value is MOVE_STUDENT and the player wants to move a student on an island.
- MoveMotherNature: if the request value is MOVE_MOTHER_NATURE.
- ChooseCloud: if the request value is CHOOSE_CLOUD.

LoginMessage

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

Arguments

- Username: the username chosen by the player.

Possible Responses

- ServerMessage: if the username is already in use and the player must choose another username.

PlayerNumber

This message is sent from the client to the server to choose the number of players of the game.

Arguments

- Username: the username of the player that sent the message.
- Player: the selected number of players

Possible Responses

This message has no responses.

Difficult

This message is sent from the client to the server to choose the game mode (Standard or expert).

Arguments

- Username: the username of the player that sent the message.
- GameMode: the game mode chosen by the player.

Possible Responses

This message has no responses.

StartGameMessage

This message is sent from the server to the client when the game starts.

Arguments

- jsonModel: a String that represent board state converted in JSON.

Possible Responses

This message has no responses.

PlayAssistantCard

This message is sent from the client to the server when the player must play his assistant card.

Arguments

- Username: the username of the player that sent the message.
- Card: the card chosen by the player.

Possible Responses

- ServerMessage: if the player chooses an already played card.

MoveStudentDining

This message is sent from the client to the server to move a student into the dining room.

Arguments

- Username: the username of the player that sent the message.
- Color: the color of the student to move into the dining room.

Possible Responses

- ServerMessage: if the move is wrong.

MoveStudentIsland

This message is sent from the client to the server to move a student on an island.

Arguments

- Username: the username of the player that sent the message.
- Color: the color of the student to move on the island.
- Island: the island to put the student.

Possible Responses

- ServerMessage: if the move is wrong.

MoveMotherNature

This message is sent from the client to the server to move mother nature.

Arguments

- Username: the username of the player that sent the message.
- Position: the number of positions to move mother nature.

Possible Responses

This message has no responses.

ChooseCloud

This message is sent from the client to the server to take students from a cloud.

Arguments

- Username: the username of the player that sent the message.
- Position: the number of positions to move mother nature.

Possible Responses

- ServerMessage: if the player chooses a cloud that is already taken.

GameUpdated

This message is sent from the server to the client every time the state of the board changes.

Arguments

- jsonModel: a String that represent board state converted in JSON.

Possible Responses

This message has no responses.

PlayCharacterCard

This message is sent from the client to the server to play the character cards soldier, cook, centaur or wizard.

Arguments

- Username: the username of the player that sent the message.
- CharacterType: the character card that the player wants to play.

Possible Responses

This message has no responses.

PlayCharacterCard

This message is sent from the client to the server to play the character card bishop.

Arguments

- Username: the username of the player that sent the message.
- CharacterType: the character card that the player wants to play.
- Island: the island chosen by the player.

Possible Responses

This message has no responses.

PlayCharacterCard

This message is sent from the client to the server to play the character cards courtier or collector.

Arguments

- Username: the username of the player that sent the message.
- CharacterType: the character card that the player wants to play.
- Color: the students' color chosen by the player

Possible Responses

This message has no responses.

PlayCharacterCard

This message is sent from the client to the server to play the character card minstrel.

Arguments

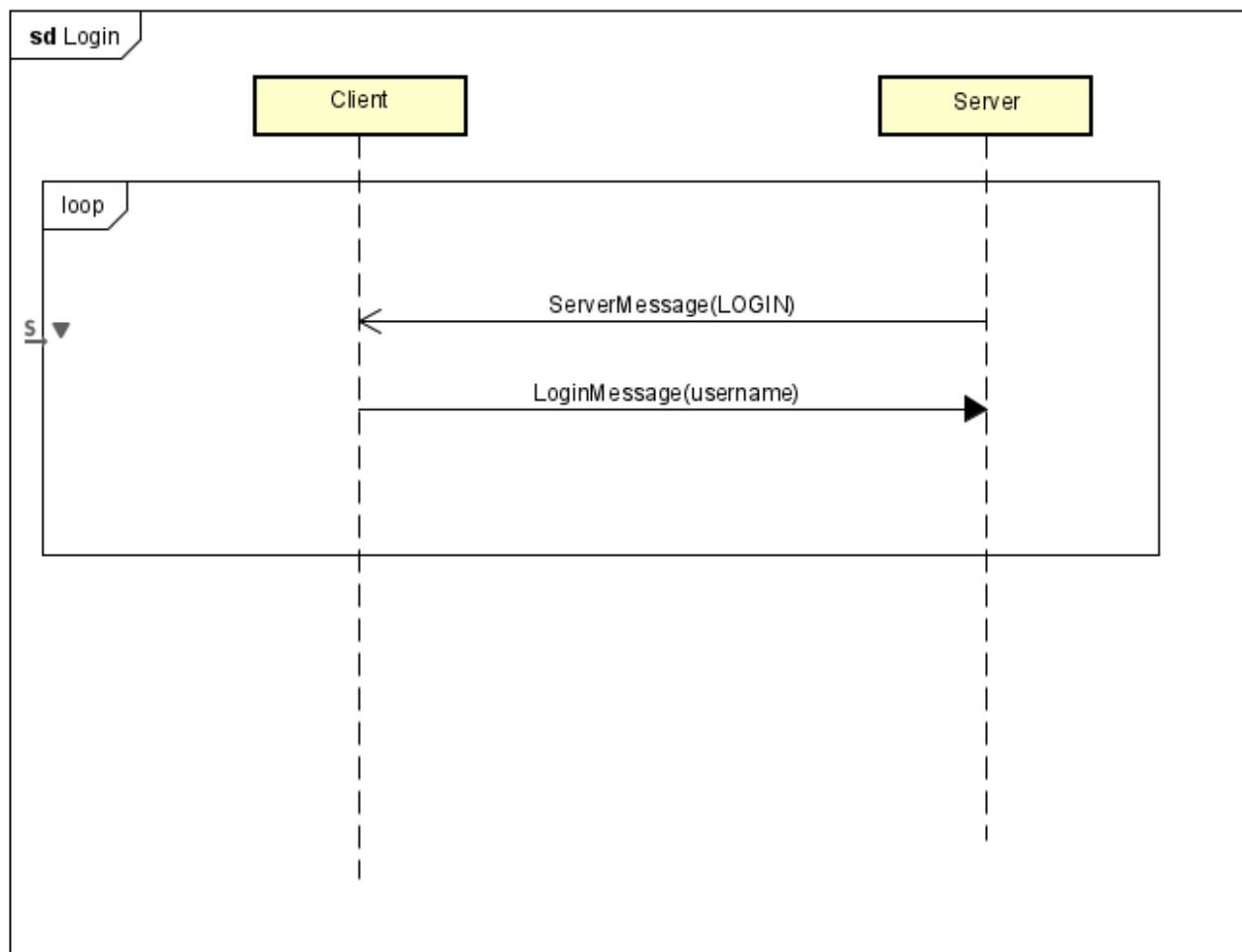
- Username: the username of the player that sent the message.
- CharacterType: the character card that the player wants to play.
- Entrance: the array containing the students to take from the entrance and to put into the dining room.
- DiningRoom: : the array containing the students to take from the dining room and to put into the entrance.

Possible Responses

This message has no responses.

Scenarios

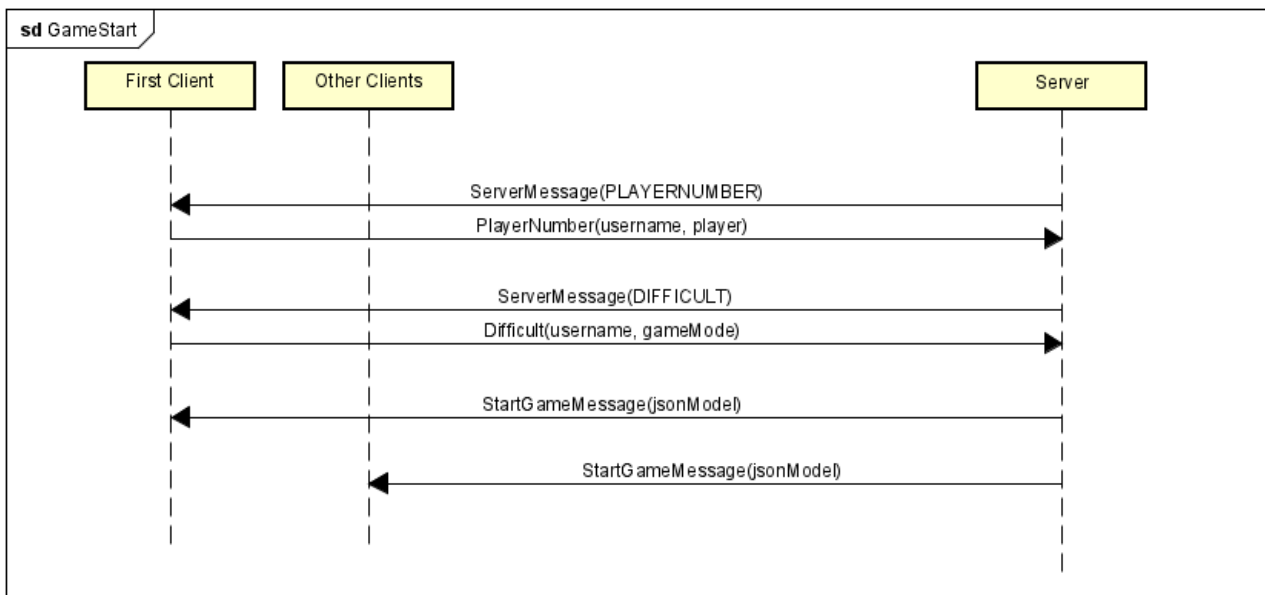
Login



Once the connection is set up the server sends a `ServerMessage` to the client with the value `LOGIN` to ask the player to choose his username. The client responds with a `LoginMessage` that contains the chosen username.

If the username is already in use the server will ask the username again until the player choose a valid username.

Game Start

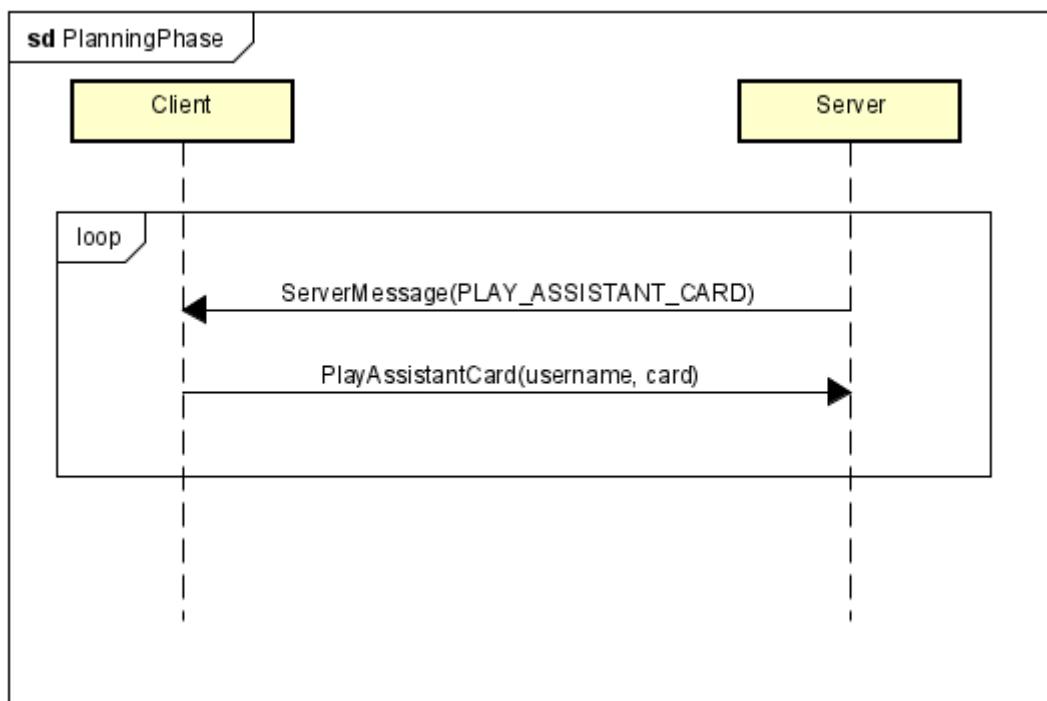


When after the first player has chosen his username, the server sends him a `ServerMessage` with the value `PLAYERNUMBER` asking him to choose the number of players of the game. The client responds with a `PlayerNumber` message containing the choice of the player.

After that, the server sends another `ServerMessage` with the value `DIFFICULT` to ask the player the game mode he wants to play, and the client will respond with a `Difficult` message containing the game mode he has chosen.

When all the players connect to the server, the game starts, and the server sends a `ServerMessage` to each client containing the board state.

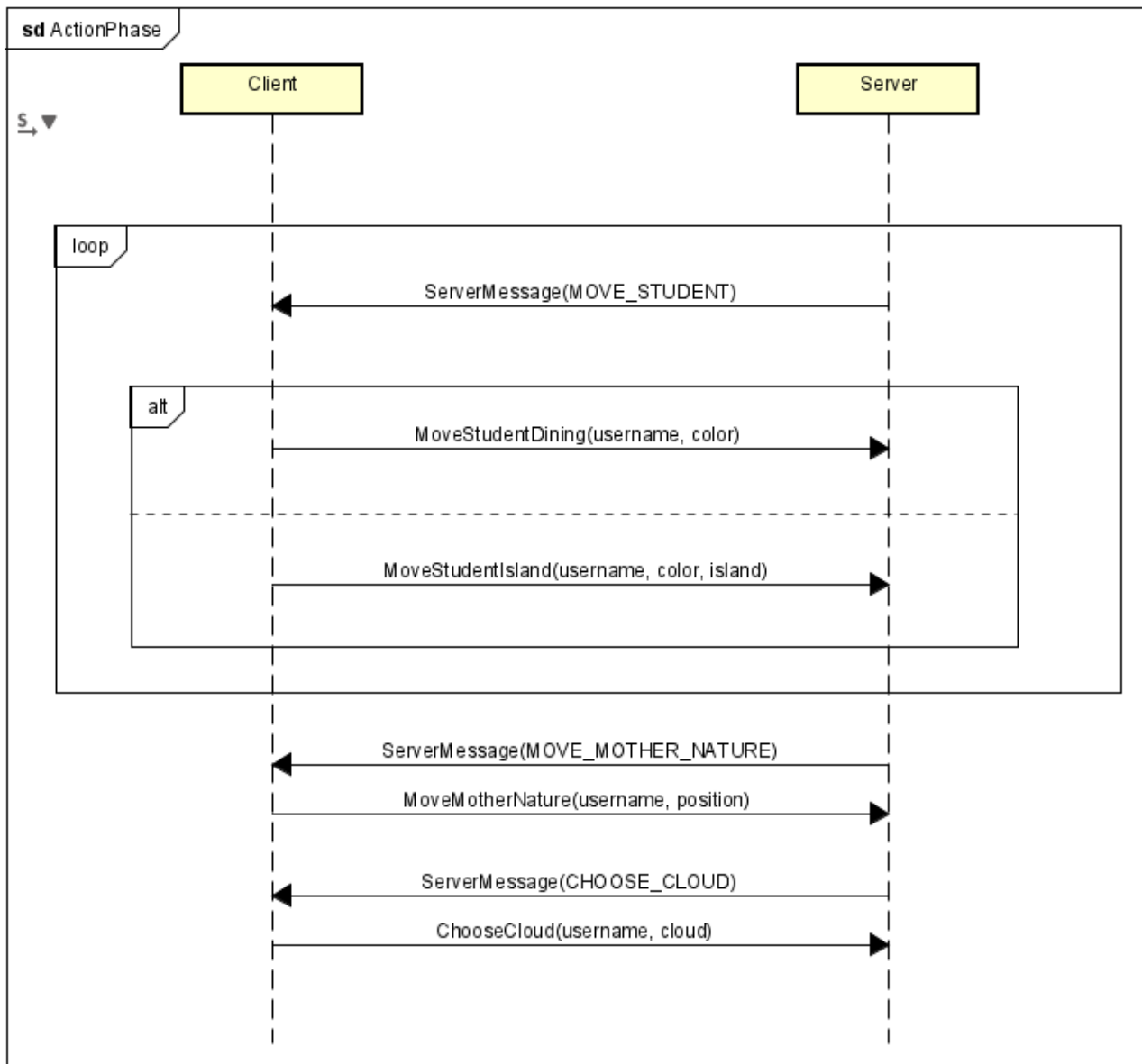
Planning phase



After the planning phase of a turn starts, the server notifies the client that must play a card with a `ServerMessage` with the value `PLAY_ASSISTANT_CARD`. The client responds with a `SelectCard`

message to choose which card to play. If the player wants to play a card that has been already played the server will ask to play an assistant card again until the player choose a playable one.

Action phase

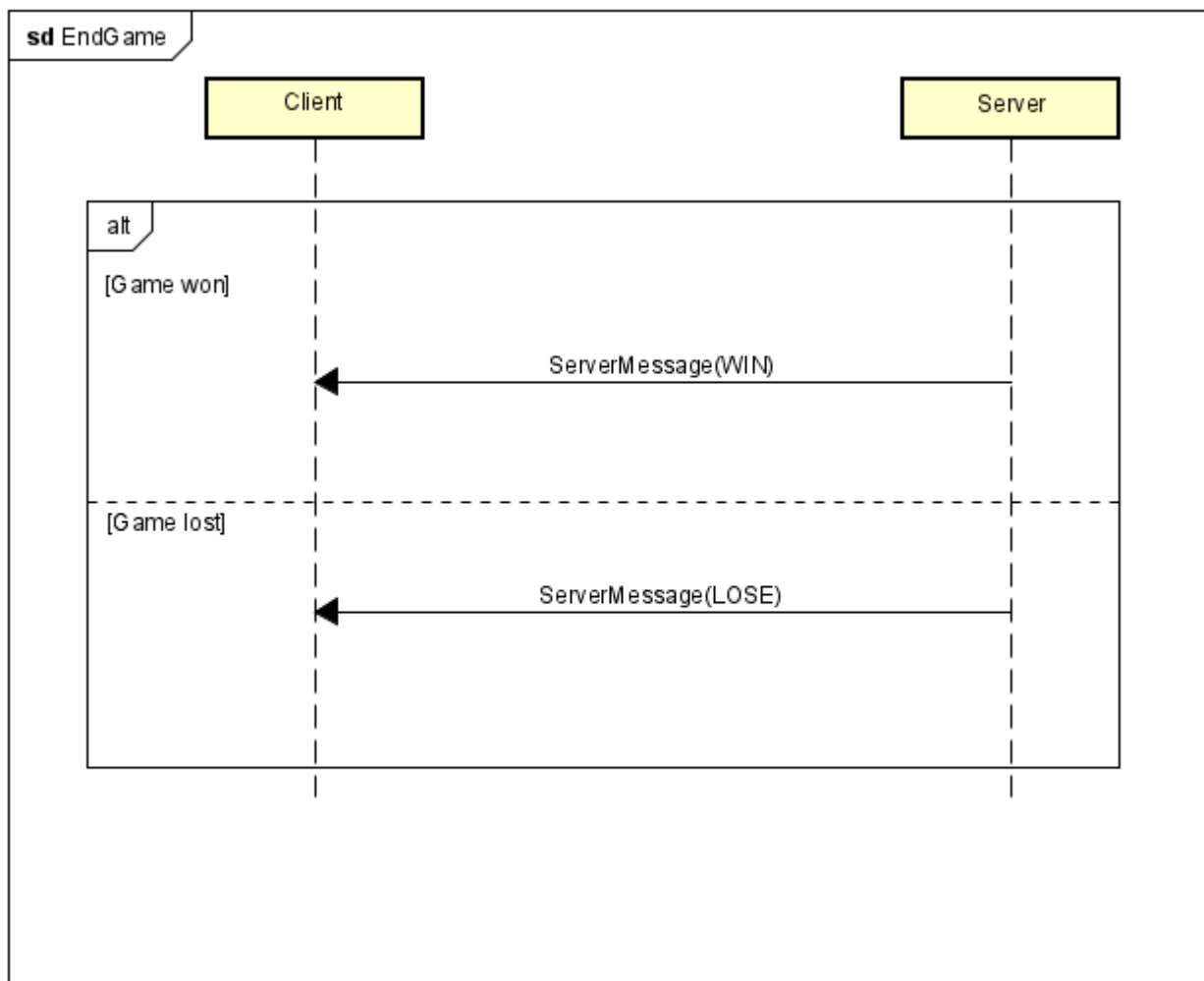


When the turn of the action phase of a player starts, the server notifies him asking him to move a student. The server will send a `ServerMessage` with the value `MOVE_STUDENT` and the client will answer with a `MoveStudentDining` message if the player wants to move a student into the dining room or with a `MoveStudentIsland` if he wants to move the student on an island. If the player chose a student, he does not have the server will ask again the movement of a student. The server will ask the player to move a student 3 or 4 times, depending on the number of players of the game.

Once the player has moved all his students, the server will send a `ServerMessage` with the value `MOVE_MOTHER_NATURE` to ask the player to move mother nature and the client will respond with a `MoveMotherNature` message. If the player chooses a wrong island, the server will send a `ServerMessage` again until the player chooses a correct island.

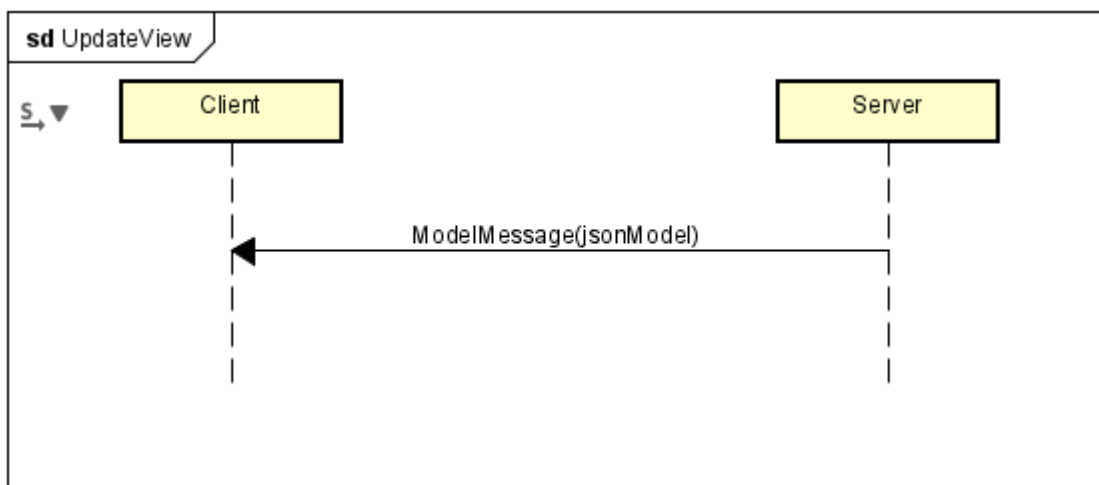
After that, the server will ask which cloud the player chooses with a `ServerMessage` with the value `CHOOSE_CLOUD` until the player chooses a correct one.

End of the game



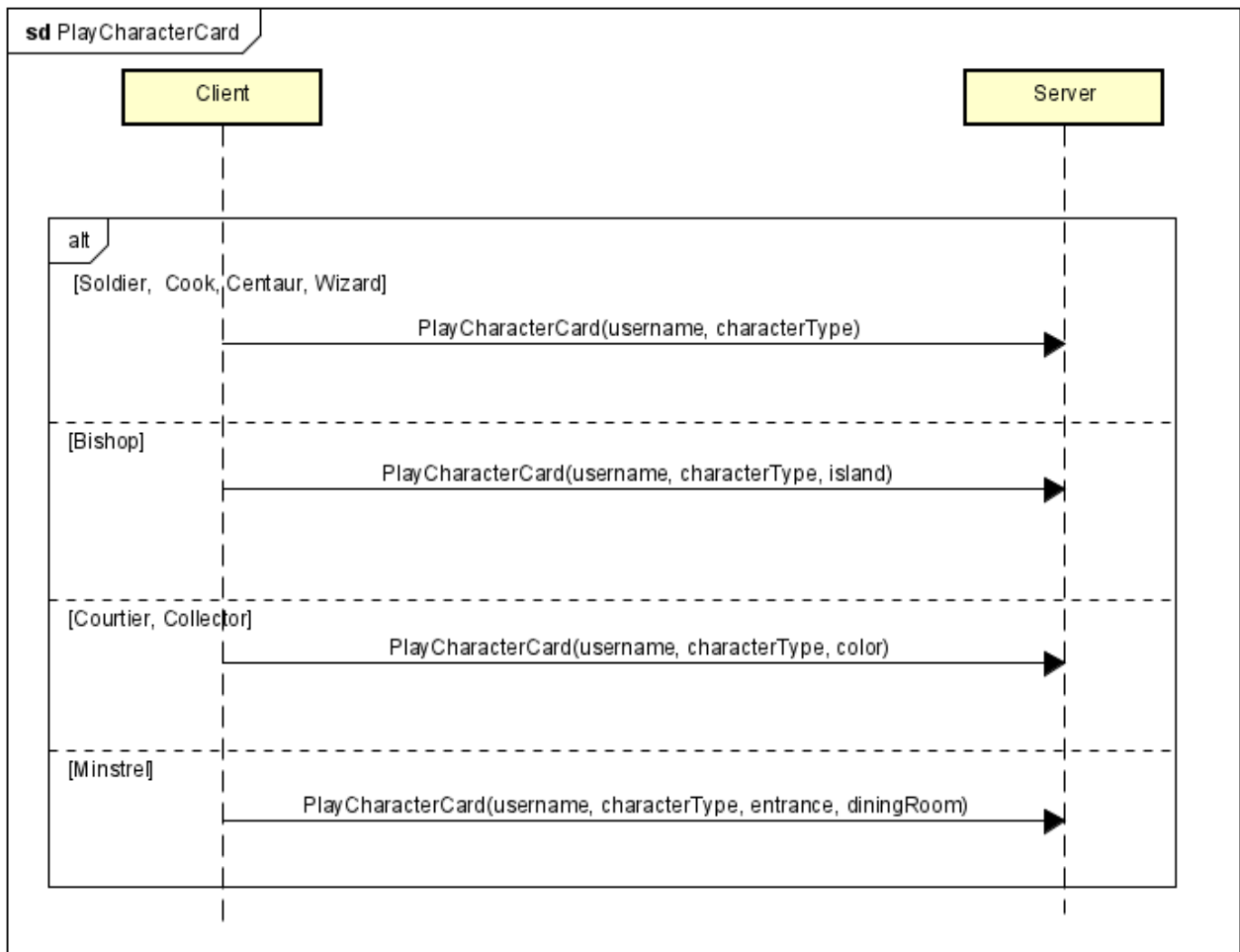
When the game ends the server sends a `ServerMessage` to every client with the value `WIN` if the client is the winner or `LOSE` if the client is not the winner.

Update view



Every time the Board change the server sends a message to every client to notify them the change with a `ModelMessage`.

Play character card



Every time a player wants to play a character card the client sends to the server a PlayCharacterMessage with the card he wants to play and the variables the server need.