

AY 2021/2022



POLITECNICO DI MILANO

# Eryantis Protocol Documentation

Group 41:

Gabriele Giannotto   Maria Pia Marini   Christian Lisi

**Version 1.1**

# Indice

<b>1</b>	<b>Client Messages</b>	<b>1</b>
1.1	NewGameMessage . . . . .	1
1.2	JoinGameMessage . . . . .	1
1.3	PingACKMessage . . . . .	1
1.4	NicknameMessage . . . . .	2
1.5	AssistantCardMessage . . . . .	2
1.6	StudentToIslandMessage . . . . .	2
1.7	StudentToDiningRoomMessage . . . . .	3
1.8	UseCharacterCardMessage . . . . .	3
1.9	AmountOfMotherNatureMovementMessage . . . . .	4
1.10	GetStudentFromCloud . . . . .	4
1.11	EndTurnMessage . . . . .	4
1.12	ColorNoInfluenceCharacterCardMessage . . . . .	5
1.13	Exchange2StudentsMessage . . . . .	5
1.14	NoEntryCardMessage . . . . .	6
1.15	OneStudentToAnIslandCardMessage . . . . .	6
1.16	Remove3StudentsFromDiningRoomCardMessage . . . . .	6
1.17	Replace3StudentsInEntranceCardMessage . . . . .	7
1.18	ResolveIslandCardMessage . . . . .	7
1.19	Take1StudentToDiningRoomCardMessage . . . . .	8
1.20	TakeControlOfProfessorCardMessage . . . . .	8
<b>2</b>	<b>Server Messages</b>	<b>9</b>
2.1	ACKMessage . . . . .	9
2.2	NACKMessage . . . . .	9
2.3	PingMessage . . . . .	9
2.4	GameCreatedMessage . . . . .	10
2.5	ClientJoinedMessage . . . . .	10
2.6	StartGameMessage . . . . .	10
2.7	StartTurnMessage . . . . .	11
2.8	StartPlanningMessage . . . . .	11
2.9	WinMessage . . . . .	11
2.10	updateActivePlayerMessage . . . . .	12
2.11	updateGameBoardMessage . . . . .	12
2.12	updatePlayerBoardMessage . . . . .	12
2.13	updateEnemyPlayerBoardMessage . . . . .	13
2.14	ExitGameMessage . . . . .	13
2.15	SendViewMessage . . . . .	13
<b>3</b>	<b>Scenarios</b>	<b>14</b>
3.1	Create new game . . . . .	14
3.2	Create new player . . . . .	15

3.3	Start a game . . . . .	16
3.4	Planning scenario . . . . .	17
3.5	Add students to island . . . . .	18
3.6	Add students to diningRoom . . . . .	19
3.7	Player use a character card . . . . .	20
3.8	Player specifies the movement of mother nature scenario . . . . .	21
3.9	Game ends and one player win scenario . . . . .	22

# 1 Client Messages

## 1.1 NewGameMessage

NewGameMessage is a Message sent by the client to the server, requests the creation of a new game to the server.

### Arguments

- gameMode: the selected game difficulty.
- NumberOfPlayer: the number of players in the game.

### Possible responses

- GameCreatedMessage: the game has been created.
- NACK: error creating a game.

## 1.2 JoinGameMessage

JoinGameMessage is a Message sent by the client to the server, requesting to join a specific game session.

### Arguments

- game: the selected game.

### Possible responses

- ClientJoinedMessage : the client has joined the game session.
- NACK: impossible to join the game.

## 1.3 PingACKMessage

PingACKMessage is a message sent from the client to the server notifying that the connection is still up.

### Arguments

This message has no arguments.

### **Possible responses**

This message has no responses.

## **1.4 NicknameMessage**

NicknameMessage is a Message sent by the client to the server, communicating the nickname of the new player.

### **Arguments**

- Nickname: the nickname of the player.

### **Possible responses**

- ACK: action done.
- NACK: action not done.

## **1.5 AssistantCardMessage**

AssistantCardMessage is a Message sent by the client to the server when a player has selected an AssistantCard to play.

### **Arguments**

- AssistantCard: the AssistantCard played by the player.

### **Possible responses**

- ACK: action done.

## **1.6 StudentToIslandMessage**

StudentToIslandMessage is a Message sent by the client to the server when a player has decided to move some students to an island.

### **Arguments**

- students: the students selected by the player that has to be moved.
- island : the island selected by the player where the students has to be moved.

**Possible responses**

- updateGameBoardMessage: move has been performed and the game board has been updated.
- ACK: action done.

**1.7 StudentToDiningRoomMessage**

StudentToDiningRoomMessage is a Message sent by the client to the server when a player has decided to move some students to the dining room.

**Arguments**

- students: the students selected by the player that has to be moved.

**Possible responses**

- updatePlayerBoardMessage: move has been performed and the player board has been updated.
- ACK: action done.

**1.8 UseCharacterCardMessage**

UseCharacterCardMessage is a Message sent by the client to the server when a player has decided to play a CharacterCard.

**Arguments**

- CharacterCard: the character card selected by the player that has to be played.

**Possible responses**

- updateGameBoardMessage && updatePlayerBoardMessage: move has been performed and the game and player board has been updated.
- ACK: effect done.

## 1.9 AmountOfMotherNatureMovementMessage

AmountOfMotherNatureMovementMessage is a Message sent by the client to the server communicating the amount of mother nature's movement selected by the player.

### Arguments

- AmountOfMotherNatureMovement: the amount of mother nature's movement selected by the player.

### Possible responses

- updateGameBoardMessage: move has been performed and the game board has been updated.
- updatePlayerBoardMessage: move has been performed and the player board has been updated.
- updateEnemyPlayerBoard: move has been performed and the enemy player board has been updated.
- ACK: action done.

## 1.10 GetStudentFromCloud

GetStudentFromCloud is a message sent from the client to the server communicating the cloud from where the player wants to take the students.

### Arguments

- Cloud: the cloud selected by the player.

### Possible responses

- updateGameBoardMessage && updatePlayerBoardMessage: move has been performed and the game and player board has been updated.
- ACK: action done.

## 1.11 EndTurnMessage

EndTurnMessage is a message sent from the client to the server notifying the end of his turn.

**Arguments**

This message has no arguments.

**Possible responses**

This message has no responses.

## 1.12 ColorNoInfluenceCharacterCardMessage

ColorNoInfluenceCharacterCardMessage is a message sent from the client to the server when a player has decided to play this CharacterCard.

**Arguments**

- ColorNoInfluenceCharacterCard
- PawnType color choosed by the player

**Possible responses**

- ACK: action done.

## 1.13 Exchange2StudentsMessage

Exchange2StudentsMessage is a message sent from the client to the server when a player has decided to play this CharacterCard.

**Arguments**

- Exchange2StudentsCharacterCard
- 

**Possible responses**

- updateGameBoardMessage: move has been performed and the game board has been updated.
- updatePlayerBoardMessage: move has been performed and the player board has been updated.
- updateEnemyPlayerBoard: move has been performed and the enemy player board has been updated.
- ACK: action done.



### 1.14 NoEntryCardMessage

NoEntryCardMessage is a message sent from the client to the server when a player has decided to play this CharacterCard.

#### Arguments

- NoEntryCard

#### Possible responses

- updateGameBoardMessage: move has been performed and the game board has been updated.
- ACK: action done.

### 1.15 OneStudentToAnIslandCardMessage

OneStudentToAnIslandCardMessage is a message sent from the client to the server when a player has decided to play this CharacterCard.

#### Arguments

- OneStudentToAnIslandCard

#### Possible responses

- updateGameBoardMessage: move has been performed and the game board has been updated.
- ACK: action done.

### 1.16 Remove3StudentsFromDiningRoomCardMessage

Remove3StudentsFromDiningRoomCardMessage is a message sent from the client to the server when a player has decided to play this CharacterCard.

#### Arguments

- Remove3StudentsFromDiningRoomCard

#### **Possible responses**

- `updatePlayerBoardMessage`: move has been performed and the player board has been updated.
- `updateEnemyPlayerBoard`: move has been performed and the enemy player board has been updated.
- `ACK`: action done.

### **1.17 Replace3StudentsInEntranceCardMessage**

`Replace3StudentsInEntranceCardMessage` is a message sent from the client to the server when a player has decided to play this `CharacterCard`.

#### **Arguments**

- `Replace3StudentsInEntranceCard`

#### **Possible responses**

- `updatePlayerBoardMessage`: move has been performed and the player board has been updated.
- `updateEnemyPlayerBoard`: move has been performed and the enemy player board has been updated.
- `ACK`: action done.

### **1.18 ResolveIslandCardMessage**

`ResolveIslandCardMessage` is a message sent from the client to the server when a player has decided to play this `CharacterCard`.

#### **Arguments**

- `ResolveIslandCard`

#### **Possible responses**

- `updatePlayerBoardMessage`: move has been performed and the player board has been updated.
- `updateEnemyPlayerBoard`: move has been performed and the enemy player board has been updated.

- `updateGameBoardMessage`: move has been performed and the game board has been updated.
- `ACK`: action done.

### 1.19 `Take1StudentToDiningRoomCardMessage`

`Take1StudentToDiningRoomCardMessage` is a message sent from the client to the server when a player has decided to play this `CharacterCard`.

#### Arguments

- `Take1StudentToDiningRoomCard`

#### Possible responses

- `updatePlayerBoardMessage`: move has been performed and the player board has been updated.
- `updateEnemyPlayerBoard`: move has been performed and the enemy player board has been updated.
- `ACK`: action done.

### 1.20 `TakeControlOfProfessorCardMessage`

`TakeControlOfProfessorCardMessage` is a message sent from the client to the server when a player has decided to play this `CharacterCard`.

#### Arguments

- `TakeControlOfProfessorCard`

#### Possible responses

- `updatePlayerBoardMessage`: move has been performed and the player board has been updated.
- `updateEnemyPlayerBoard`: move has been performed and the enemy player board has been updated.
- `ACK`: action done.

## 2 Server Messages

### 2.1 ACKMessage

ACKMessage is a message sent from the server to the client when a generic message has been acknowledged.

#### Arguments

This message has no arguments.

#### Possible responses

This message has no responses.

### 2.2 NACKMessage

NACKMessage is a message sent from the server to the client notifying that the message has been reject or to indicate some kind of error.

#### Arguments

This message has no arguments.

#### Possible responses

This message has no responses.

### 2.3 PingMessage

PingMessage is a message sent from the server to the client checking if the connection is stil up.

#### Arguments

This message has no arguments.

#### Possible responses

- PingACKMessage: client positive respnose.

## 2.4 GameCreatedMessage

GameCreatedMessage is a message sent by server notifying the client that a new game session has been created.

### Arguments

This message has no arguments.

### Possible responses

This message has no responses.

## 2.5 ClientJoinedMessage

ClientJoinedMessage is a message sent by server notifying the client that has joined the game session.

### Arguments

This message has no arguments.

### Possible responses

This message has no responses.

## 2.6 StartGameMessage

StartGameMessage is a message sent by server when all the configuration and initialization actions are successfully done and the game can start

### Arguments

- player : the nickname of the player who has the start the first game phase.

### Possible responses

This message has no responses.

## 2.7 StartTurnMessage

StartTurnMessage is a message sent by server for notify a player that his turn is started.

### Arguments

- Nickname: the nickname of the player.

### Possible responses

This message has no responses.

## 2.8 StartPlanningMessage

StartPlanningMessage is a message sent by server for notify a player that his planning phase is started.

### Arguments

- Nickname: the nickname of the player.

### Possible responses

This message has no responses.

## 2.9 WinMessage

WinMessage is a message sent by server for notify which player has won the game.

### Arguments

- Nickname: the nickname of the player.

### Possible responses

This message has no responses.

## 2.10 updateActivePlayerMessage

updateActivePlayerMessage is a message sent by server notifying the new player that can perform action.

### Arguments

- Nickname: the nickname of the player.

### Possible responses

This message has no responses.

## 2.11 updateGameBoardMessage

updateGameBoardMessage is a message sent by server notifying that the gameBoard has changed and need to be updated.

### Arguments

- gameBoard: the state of the game board.

### Possible responses

This message has no responses.

## 2.12 updatePlayerBoardMessage

updatePlayerBoardMessage is a message sent by server notifying that the player board has changed and need to be updated.

### Arguments

- playerBoard: the state of the player board.

### Possible responses

This message has no responses.

### 2.13 updateEnemyPlayerBoardMessage

updateEnemyPlayerBoardMessage is a message sent by server notifying that the enemy player board has changed and need to be updated.

#### Arguments

- enemyPlayerBoard: the state of the enemy player board.

#### Possible responses

This message has no responses.

### 2.14 ExitGameMessage

ExitGameMessage is a message sent by server notifying that a player has closed the game.

#### Arguments

This message has no arguments.

#### Possible responses

This message has no responses.

### 2.15 SendViewMessage

SendViewMessage is a message sent by server notifying to a client his new view.

#### Arguments

- view : the view to set for the client

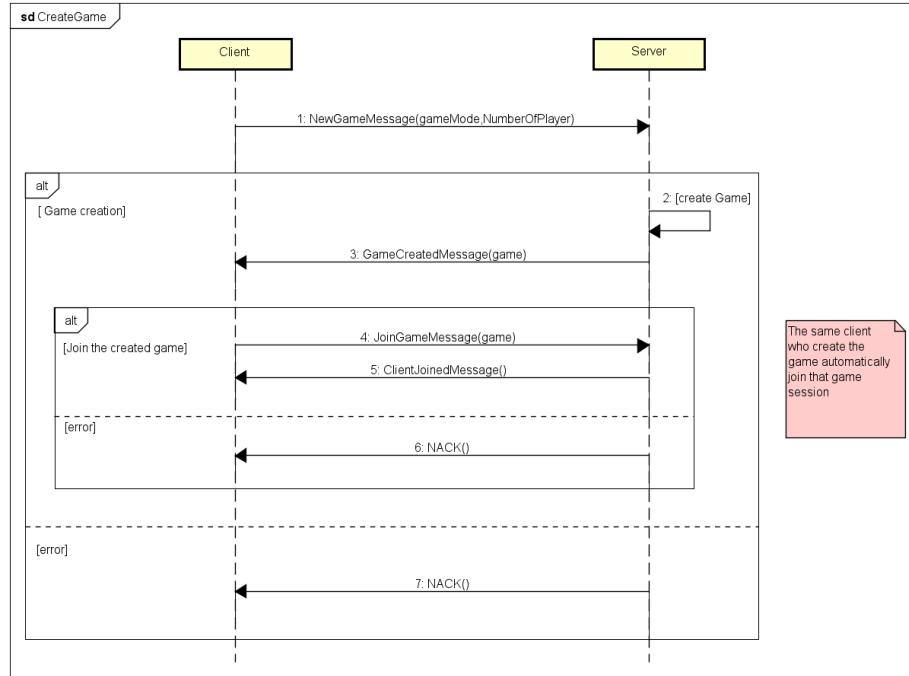
#### Possible responses

This message has no responses.



### 3 Scenarios

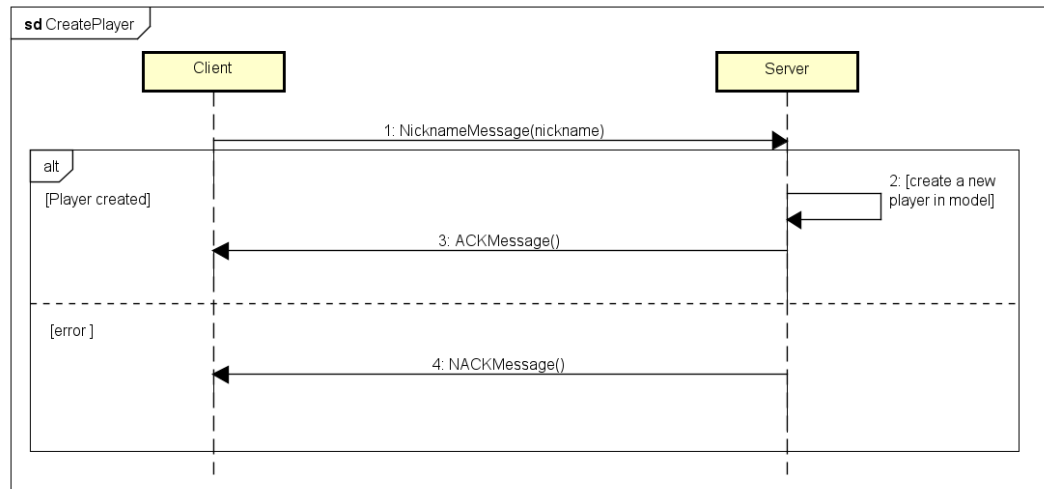
#### 3.1 Create new game



A client sends a `NewGameMessage` and request the creation of a new game session with specified `gameMode` and number of player, the server creates the game and responds with a `GameCreatedMessage`. Then the client that has created the game automatically request to join that game session.

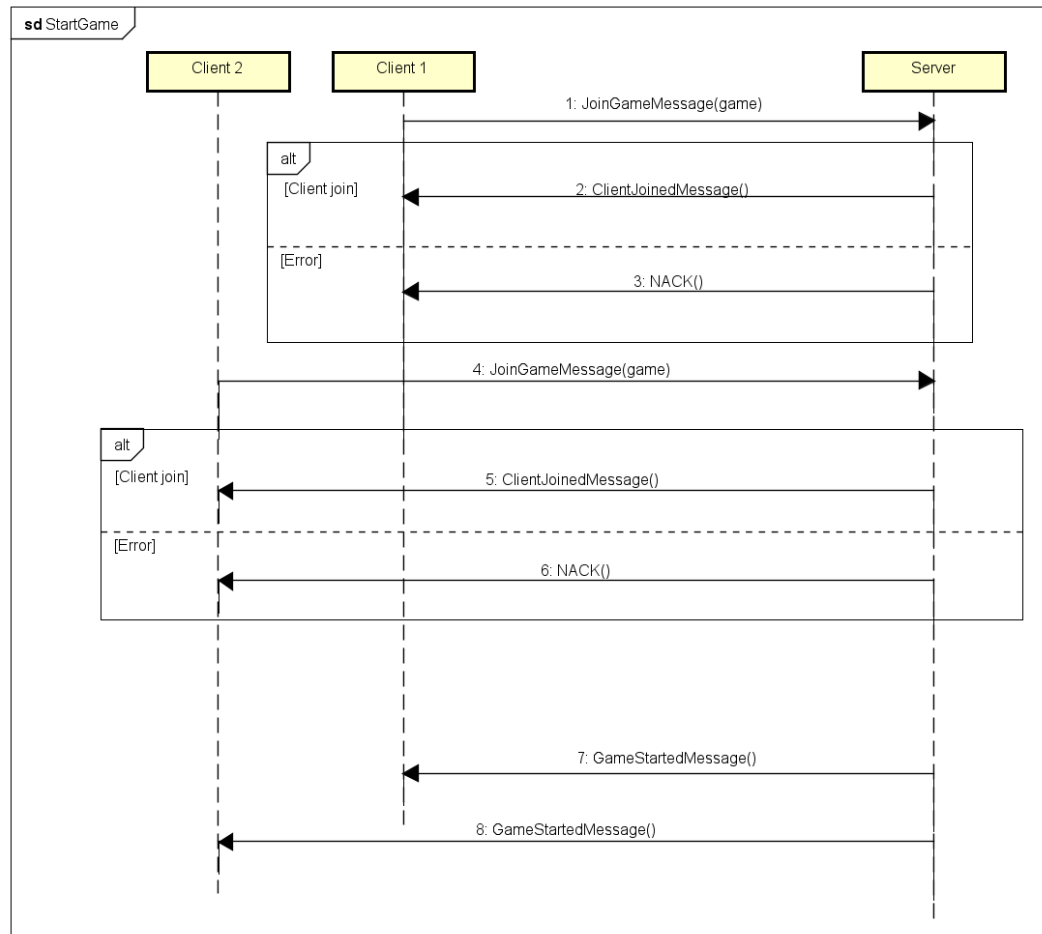
In case of any kind of error, the server notifies the client with a `NACKMessage`.

### 3.2 Create new player



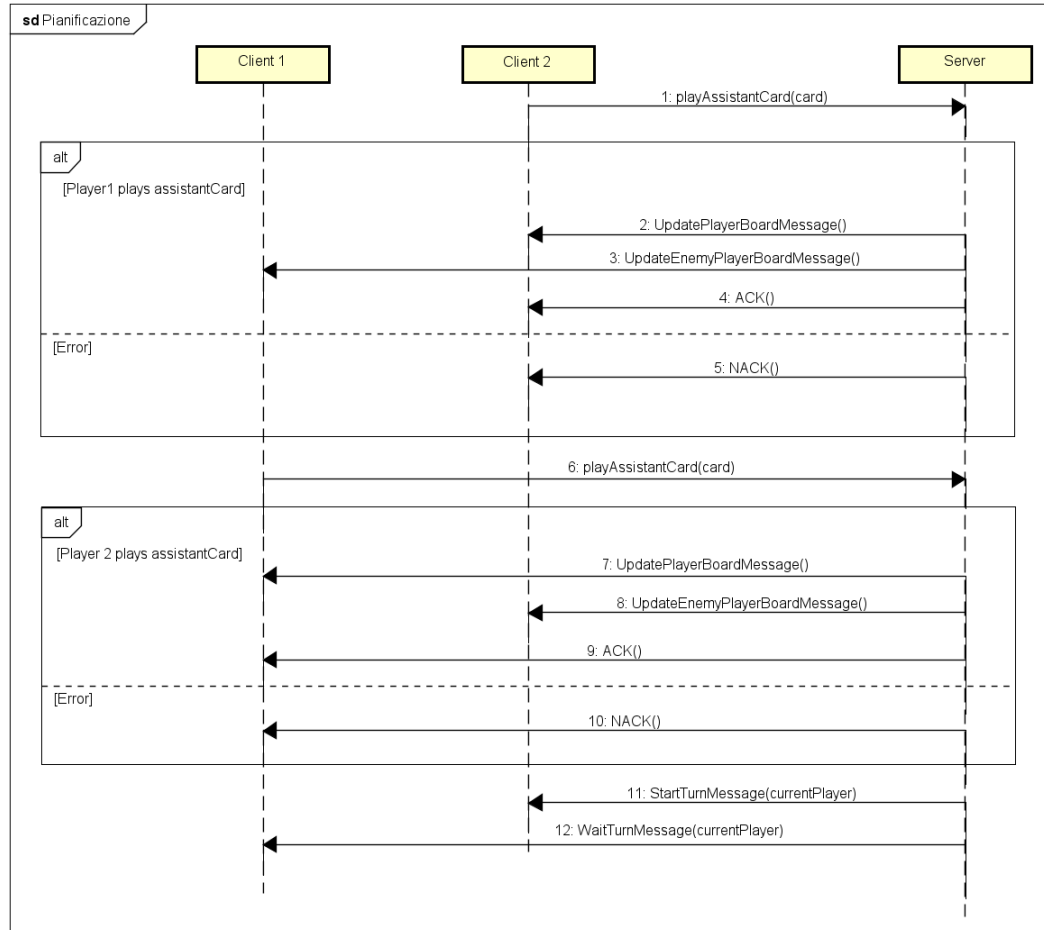
A client sends a NickNameMessage and request the creation of a new player with specified nickname, the server creates the player and responds with a PlayerCreationSuccess. In case of any kind of error, the server notifies the client with a PlayerCreationError.

### 3.3 Start a game



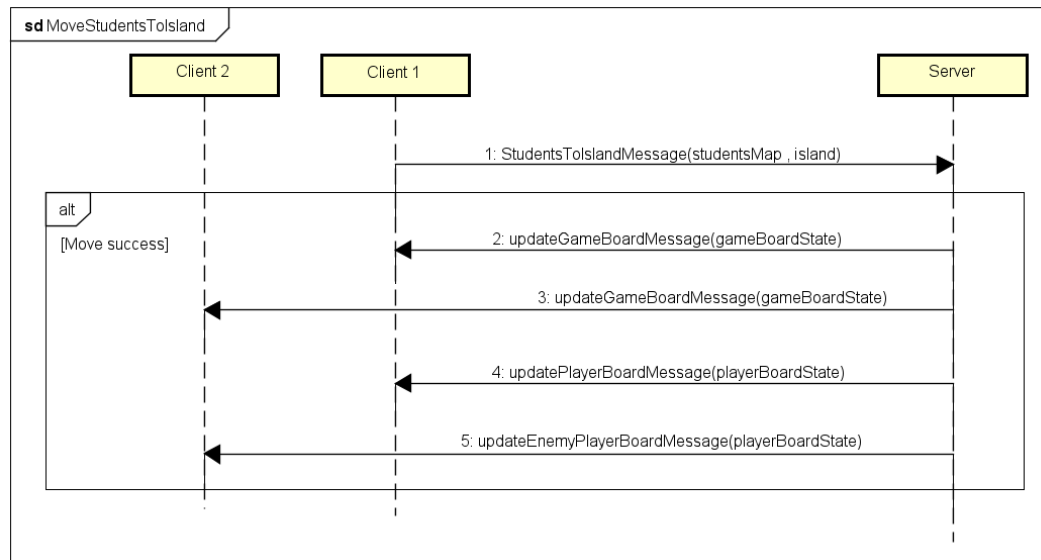
The server waits for clients to send JoinGameMessage. Each time a JoinGameMessage is received, the server try to let the client join the game session, if any kind of error occur the server notifies with a NACKMessage. When the game's lobby is full, the server responds with a GameStartedMessage to each client.

### 3.4 Planning scenario



First, the `updateGameBoardMessage` message is sent to all the users notifying that the state of the cloud tile has changed. Then each player sends a `playAssistantCard` message with the choosed card , the server responds with an `ACK` or a `MoveError` in case any error occurs(eg. assistant card already played by an other player). In the end, a `updatePlayerTurn` message is sent from the server to all the clients to inform about the new playing order.

### 3.5 Add students to island

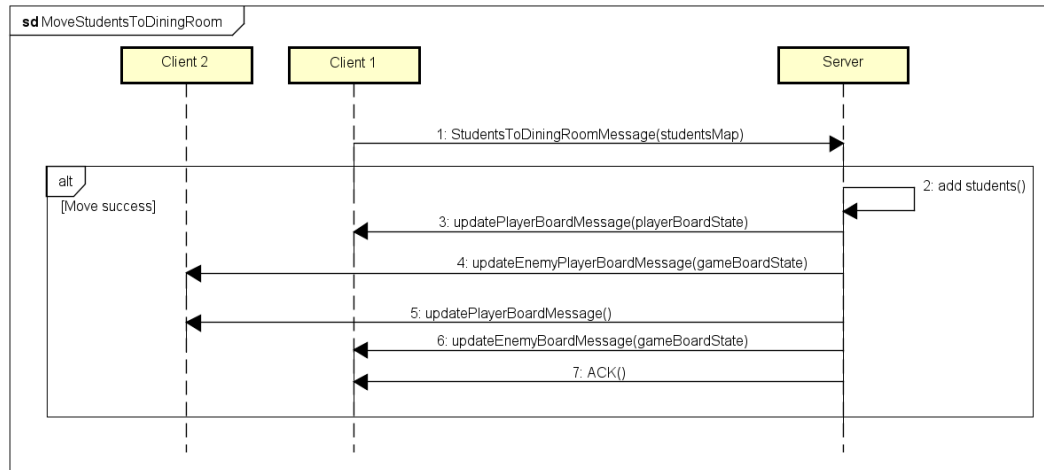


A client requests the transfer of some students to an island of his choice with the `StudentToIslandMessage` and as arguments the map of the students and the island choosed by the player. The server perform the move and update :

- the current state of client's `playerBoard`.
- the current state of the `gameBoard` for each client connected to the game.

In case of any kind of error, the server notifies the client with a `MoveErrorMessage`.

### 3.6 Add students to diningRoom



A client requests the transfer of some students to his dining room of with the `StudentToDiningRoomMessage` and as arguments the map of the students. The server perform the move and update :

- the current state of client's `playerBoard`.
- the current state of the `gameBoard` for each client connected to the game.

In case of any kind of error, the server notifies the client with a `MoveErrorMessage`.

### 3.7 Player use a character card

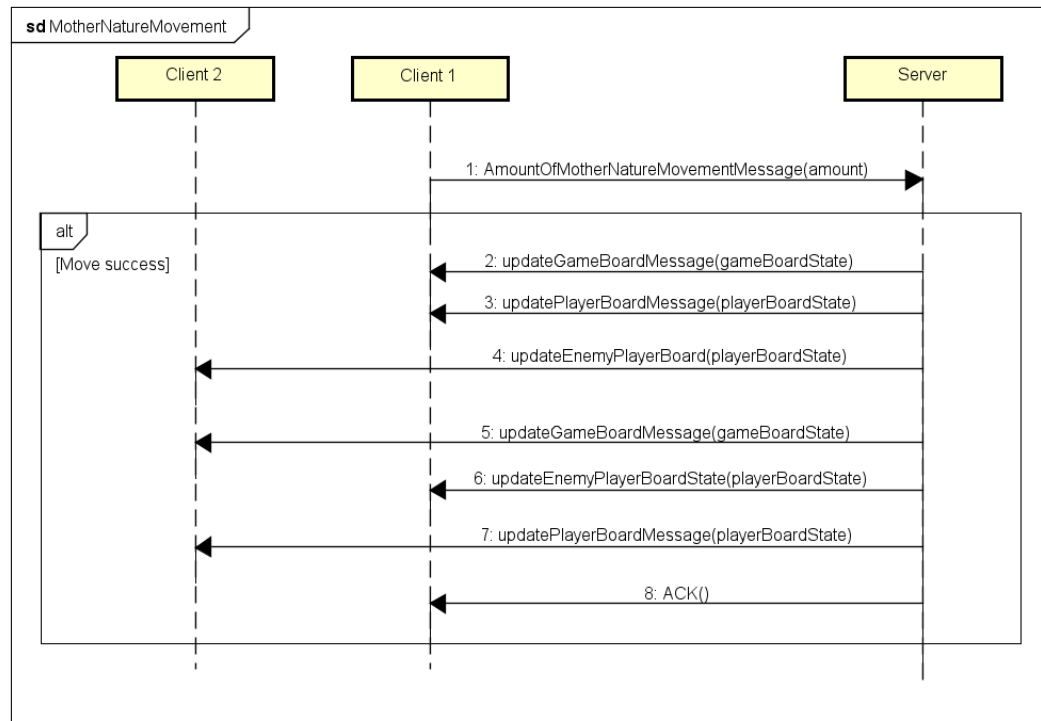


A client requests the activation of a character card's effect with the `UseCharacterCardMessage` and as a argument the card chosen by the player. The server perform the move and update :

- the current state of client's `playerBoard`.
- the current state of the `gameBoard` for each client connected to the game.

In case of any kind of error, the server notifies the client with a `MoveErrorMessage`.

### 3.8 Player specifies the movement of mother nature scenario



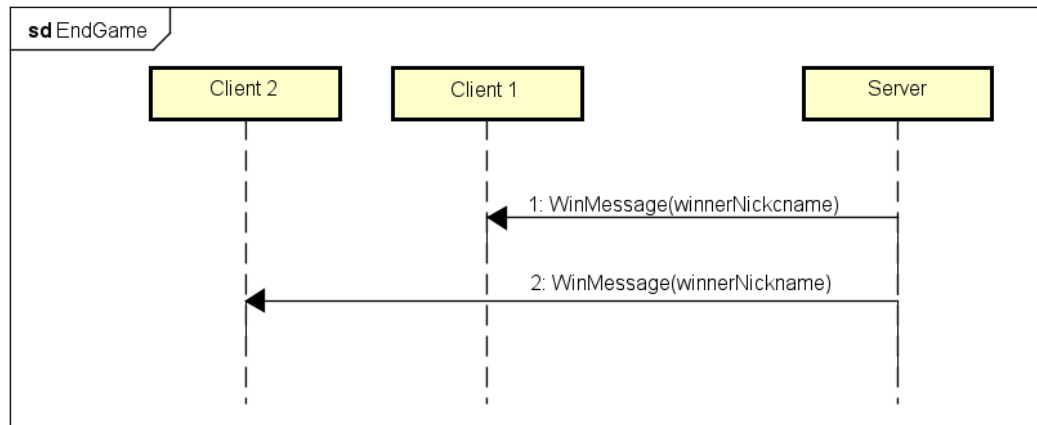
A client sends to the server the `AmountOfMotherNatureMovementMessage` with the amount specified by the player. The server perform the move and update :

- the current state of client's `playerBoard` for each client connected to the game.
- the current state of the `gameBoard` for each client connected to the game.

In case of any kind of error, the server notifies the client with a `MoveErrorMessage`.



### 3.9 Game ends and one player win scenario



The server, if an end game condition is satisfied, sends to all the client the WinMessage with the nickname of the winner.