Eryantis Protocol Documentation

Motelli, Moretti, Ianieri

Gruppo PSP16

1) Game Creation

The initial creation of the connection between client and server and initialization of the game.

1.1) startClient

connect a Client to a Server.

Responses: loginMsg, successful or denied and roomSizeRequest.

1.1.1) Successful

The login is successful.

1.1.2) Denied

The login is denied.

1.1.3) roomSizeRequest

message from Server to Client, ask for the number of players in this game.

1.2) roomPlayer

message to the Server to instantiate a game.

Arguments

- numPlayer: the number of players that the game must have to start.
- variant: the variant of the game between NORMAL or EXPERT.

1.3) username

set the username of the player and add it to the game.

Arguments

- playerName: the nickname of the player.

1.4) createRoom

message from Server to Controller to initialize the game.

Arguments

- numPlayer: the number of players that the game must have to start.
- variant: the variant of the game between NORMAL or EXPERT.

1.5) addPlayer

adding the player to the game.

Arguments

- playerName: the nickname of the player.

1.6) ready

message from Client to Server, set that the client is ready to start.

Possible respondes: startGame

1.6.1) startGame

message from Server to Clients that report the starting of the game.

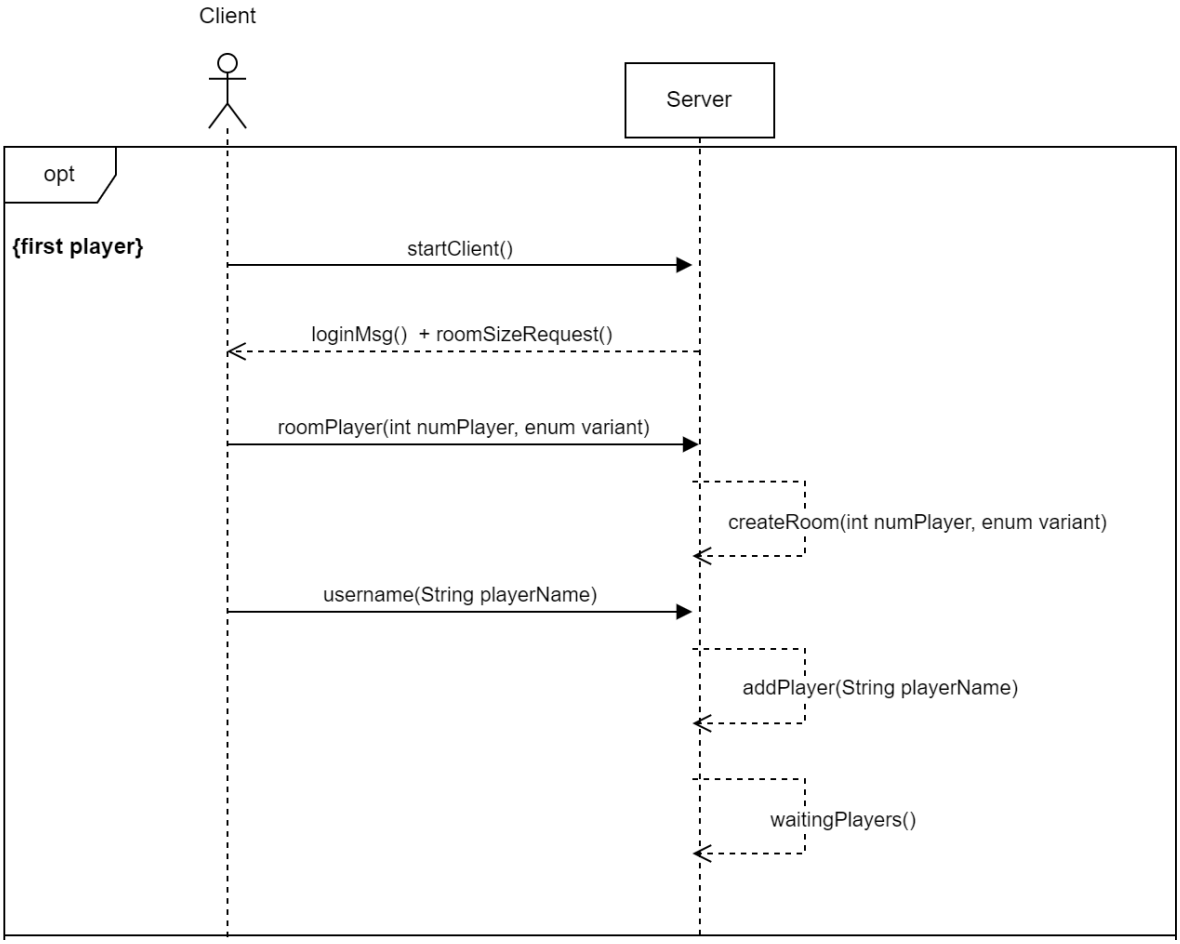
Arguments

- model: the initial model

1.7) creationGeneralGame

message from Server to the Controller, the game must be initialized.

Scenarios



The first player logs in and instantiates a new game.

2) Planning Phase

Refilling the clouds and get the assistant card from the current player.

2.1) planningPhaseStarted

Message from Server to Client, warns that the planning phase is begun.

2.2) useAssistantCard

Message from Client to Server that reports the assistant card selected by the current player.

Arguments

- ac: the assistant cart selected by the player.

2.3) newSituation

Message from Server to Client, shows the model situation after the phase

Arguments

- model: the model of the game.

2.4) refillClouds

Message from Server to Controller, shows that the clouds have to be refilled.

Arguments

- students: the students that must be placed on the clouds.

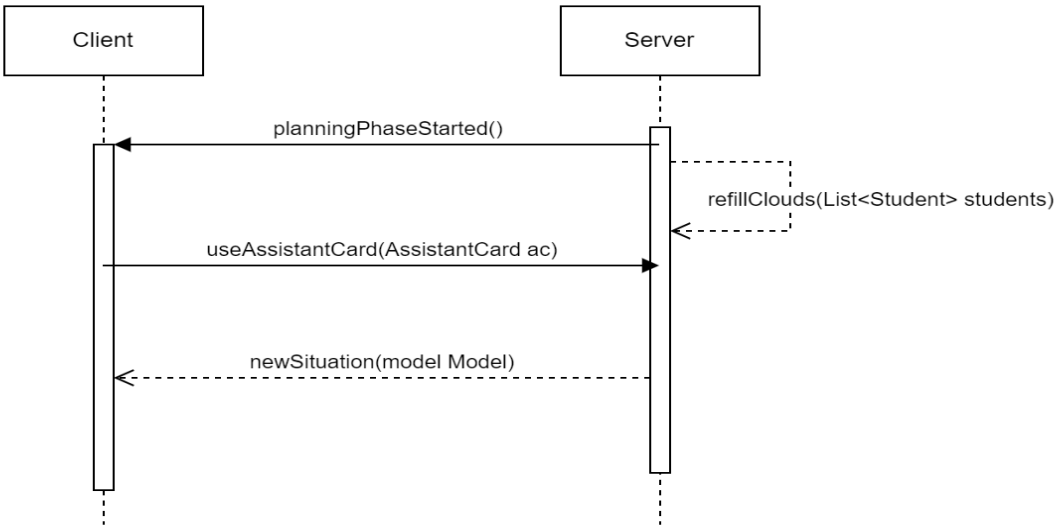
2.5) useAssistantCardToModel

Message from Server to Controller that reports the assistant card selected by the current player.

Arguments

- assistant: the assistant cart selected by the player.

Scenarios



A new planning phase is begun. The Server notifies the Clients and refills the clouds in the game. Then the Client, the current player, selects the assistant card and the Server spreads it to the model and changes the condition. The Server notifies the Clients of the new model's situation.

3) Action Phase

Based on how many players are playing the game the current player (the Client) must place students on islands or in the school's hall. The Server notifies the Client that it must place other students 3 or 4 times. Then the Client must move mother nature on an island from those remaining. At the end of the phase, the player takes all the students from a cloud with students. The Server notifies all the Clients of the new model's situation.

3.1) wherePlaceStudents

The Server notifies the client that must place students.

Arguments

- students: the students that can be placed.

Responses: placeStudentInHall or placeStudentOnIsland.

3.1.1) palceStudentInHall

The Client notifies the Server that a student must be placed in the hall.

Arguments

- student: the student that must be placed.

Responses: newSituation.

3.1.2) palceStudentOnIsland

The Client notifies the Server that a student must be placed on one island.

Arguments

- student: the student that must be placed.
- Island: the island selected where the student must be placed.

Responses: newSituation.

3.2) moveMotherNature

The Client notifies the Server on which island mother nature will be at the end of the player's turn.

Arguments

- island: the island selected where mother nature ends.

Responses: newSituation.

3.3) takeStudentsFromClouds

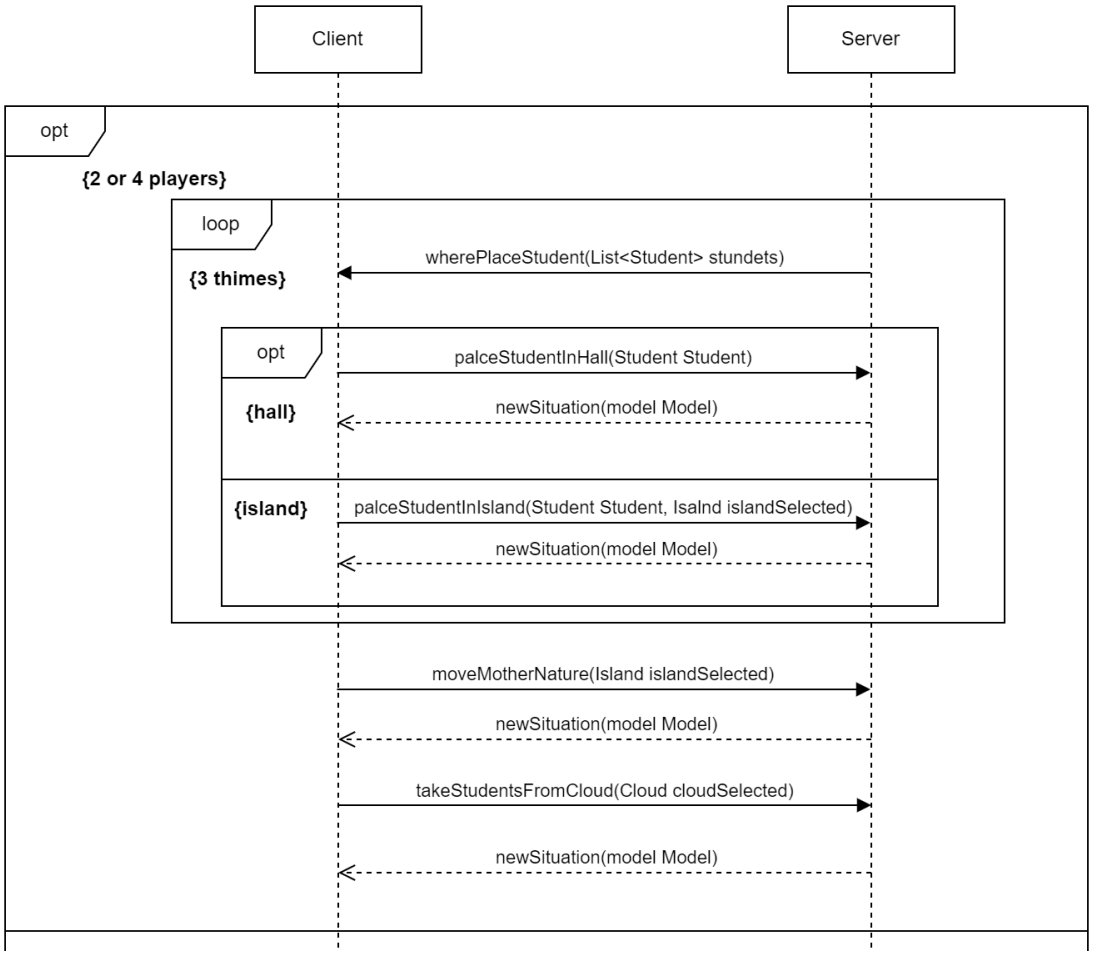
The Client notifies the Server of which cloud it will get the students.

Arguments

- cloud: the cloud selected.

Responses: newSituation.

Scenarios



Game with 2 or 4 players. The Server asks three times to the Client where to place the students. Then the Client selects to move mother nature and finally, it takes the students from one cloud.

4) Ending Game

The Server checks all the possible scenarios in which the game ends or it is the last turn of the game.

4.1) checkPlayerTowers

The server asks the model if the players have other towers.
Responses: numOfTowers.

4.1.1) noTtowers

A player has no other towers, the Server will communicate that the game will end and the winner(s) (more winners if the game is a team game).

4.1.1.1) endingGame

The message, from the Server to Clients, of the end of the game, the end of the connection and the winners.

4.1.2) Towers

Every player has at least one tower, new turn must begin (new planning phase)

4.2) checkIslands

The server asks the model the number of islands on the table.
Responses: numOfIslands.

4.2.1) threeIslands

Minimum number of the islands to end the game.

4.2.1.1) endingGame

4.2.2) moreThanThreeIslands

The game can continue from the current phase and turn.

4.3) checkAssistantDeck

Check number of assistants in the deck of each player.
Responses: numofAssistantCardInDeck

4.3.1) noAssistants

No assistant card in the assistant deck, last turn begins.

4.3.1.1) lastTurn

The current turn is the last turn, the next one will declare the winners by checking the influence

4.3.1) moreAssistants

Each player has at least one assistant card in their assistant deck, the turn will continue normally.

4.4) checkStudentsInBag

Check if the bag on the table has other students in it.
Responses: numOfStudentsInBag

4.4.1) noStudents

No students in the bag, the last turn begins.

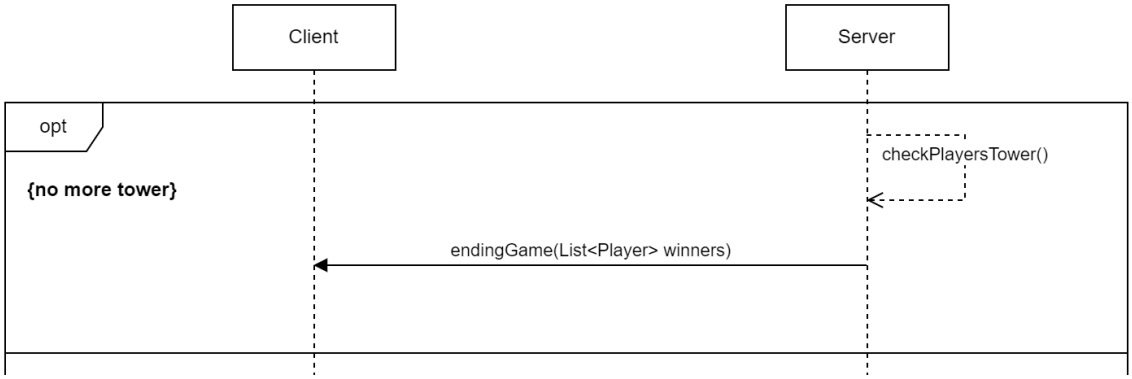
4.3.1.1) lastTurn

The current turn is the last turn, the next one will declare the winners by checking the influence

4.3.1) moreStudents

The bag has at least one student in it, the turn will continue normally.

Scenarios



The current player places its last tower and wins the game.