

Eryantis Protocol Documentation

Yanfeng Li, Leonardo Marazzi, Alessio Mangiarotti

Gruppo 5

We decided to implement a thin client network.

Messages

1.1 Acknowledgment

This message is 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.

1.2 Login

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

Arguments

- This Username: the name of the user

Possible Responses

- Login: success (1. The username exists but it's not used, 2. Create new username)
- Login: failure

1.3 Login Success

This message is sent from the server to the client when the login is successfully received.

Arguments

- This message has no arguments

Possible Responses

- This message has no responses.

1.4 Login Failure

This message is sent from the server to the client when a player session could not be created because the username is already in use

Arguments

- This message has no arguments

Possible Responses

- This message has no responses.

1.5 Create game

This message is sent from the client to the server after login in the game to create a new game.

Arguments

- This Username: the name of the game creator, number of player and mode

Possible Responses

- Acknowledgment

1.6 Show games

This message is sent from the client to the server to get all the joinable games.

Arguments

- This message has no arguments

Possible Responses

- Joinable games

1.7 Joinable game

This message is sent from the server to the client with all joinable games.

Arguments

- Array of all possible joinable Game object

Possible Responses

- Join game

1.8 Join game

This message is sent from the client to the server to join a game.

Arguments

- This Username: the name of the user and game to join

Possible Responses

- join: success
- failure: failure

1.9 Join failure

This message is sent from the server to the client when a player can't enter a game.

Arguments

- `print("join wasn't successful")`

Possible Responses

- this message has no responses

1.10 Join success

This message is sent from the server to the client when a player can enter a game.

Arguments

- `string("join is successful"), return Game`

Possible Responses

- this message has no responses

1.11 Start game

This message is sent from the client to the server when the game creator wants to start the game.

Arguments

- `game`

Possible Responses

- `failure`
- `success`

1.12 Start game failure

This message is sent from the server to the client when you can't start the game.

Arguments

- `print("not enough player yet")`

Possible Responses

- this message has no arguments

1.13 Start game success

This message is sent from the server to the client when a player can enter a game.

Arguments

- `Return Game`

Possible Responses

- `Acknowledgment`

1.14 Game update

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

Arguments

- **Game**

Possible Responses

- **Acknowledgment**

1.15 Play Assistant card

This message is sent from the client to the server to play an assistant card.

Arguments

- **Assistant card,**
- **Player**

Possible Responses

- **Game update**
- **Failure**

1.16 Play Assistant failure

This message is sent from the server when a player can not play a specific assistant.

Arguments

- **print("error")**
- **error could be assistant already used**

Possible Responses

- **Play assistant card**

1.17 Move student to dining room

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

Arguments

- **Student**
- **Player**

Possible Responses

- **Game update**

1.18 Move student to island

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

Arguments

- Student
- Island
- Player

Possible Responses

- Game update

1.19 Use character

This message is sent from the client to the server to use a character.

Arguments

- Character card
- Player

Possible Responses

- Game update

1.20 Move mother nature

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

Arguments

- Island

Possible Responses

- Game update
- Failure

1.21 Failure Move mother nature

This message is sent from the server to the client when mother nature can't be moved to that island

Arguments

- Message ("error")

Possible Responses

- Move mother nature

1.22 Select cloud

This message is sent from the client to the server to select the cloud

Arguments

- cloud

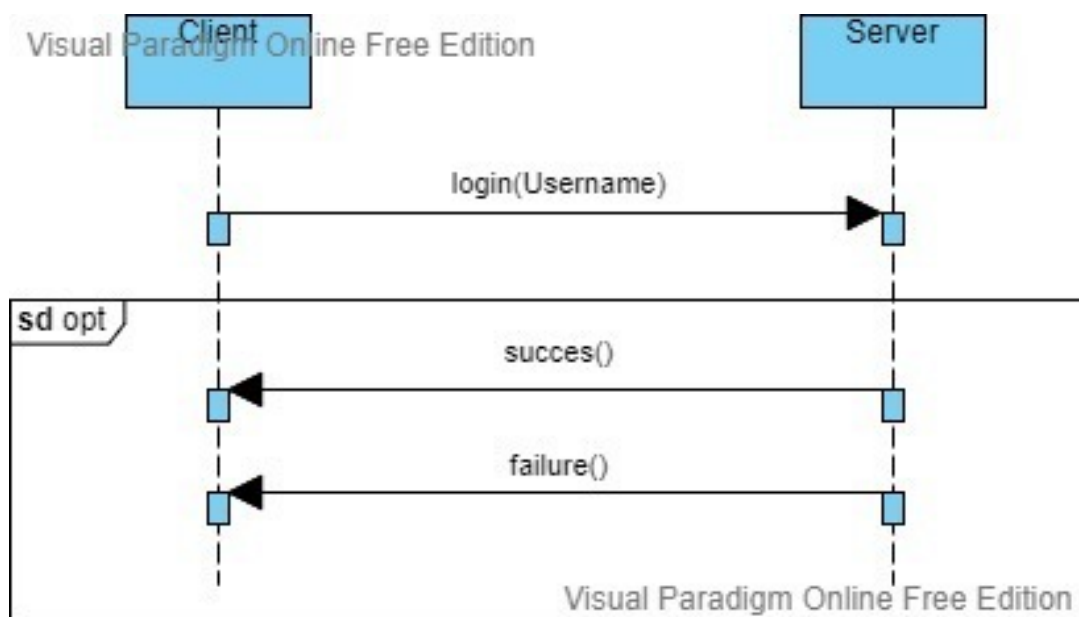
Possible Responses

- game update

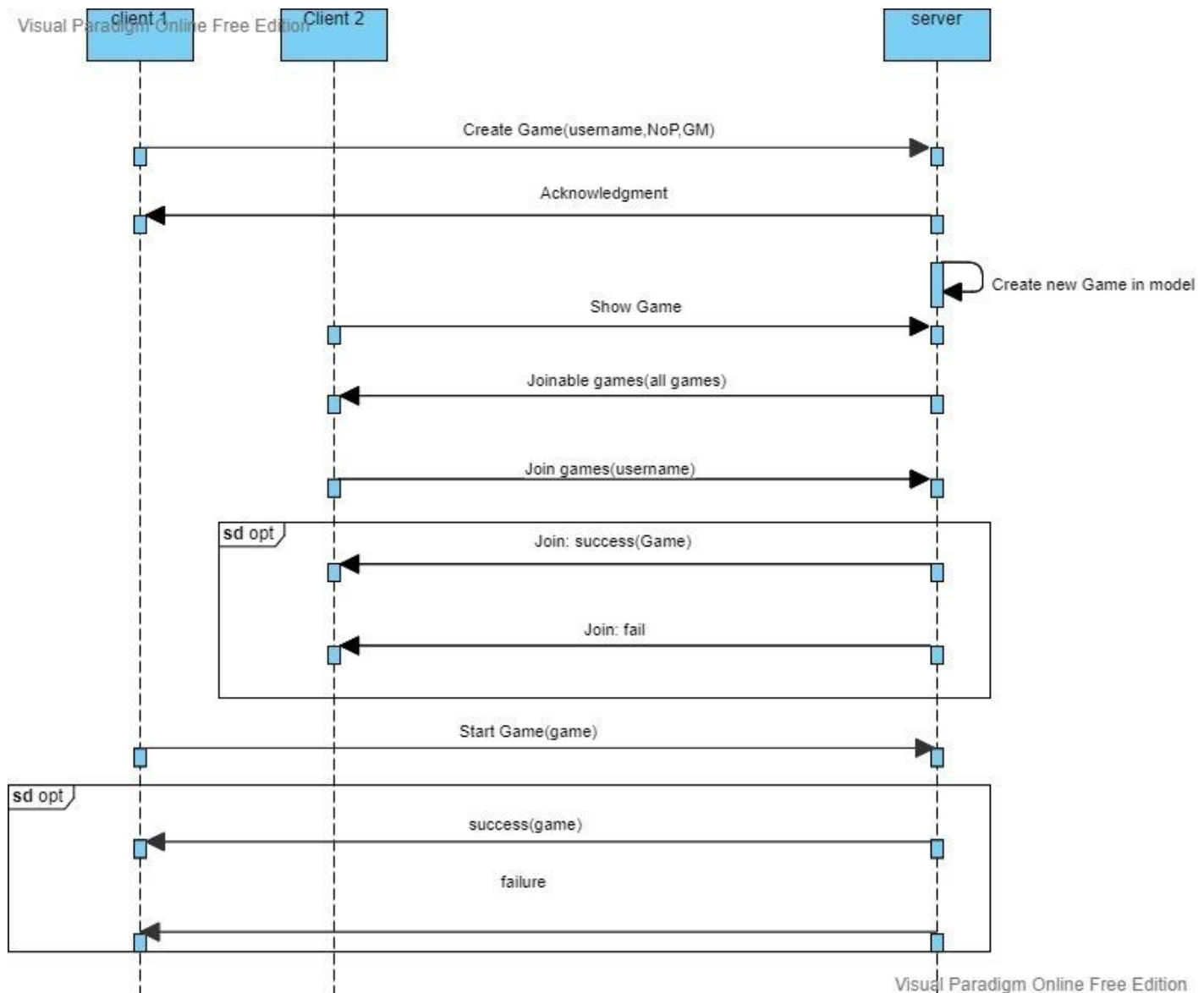
Scenarios

2.1 Login

A user have to log in using a Username, if the username is already used by another user the server sends a failure message and asks for a new Username

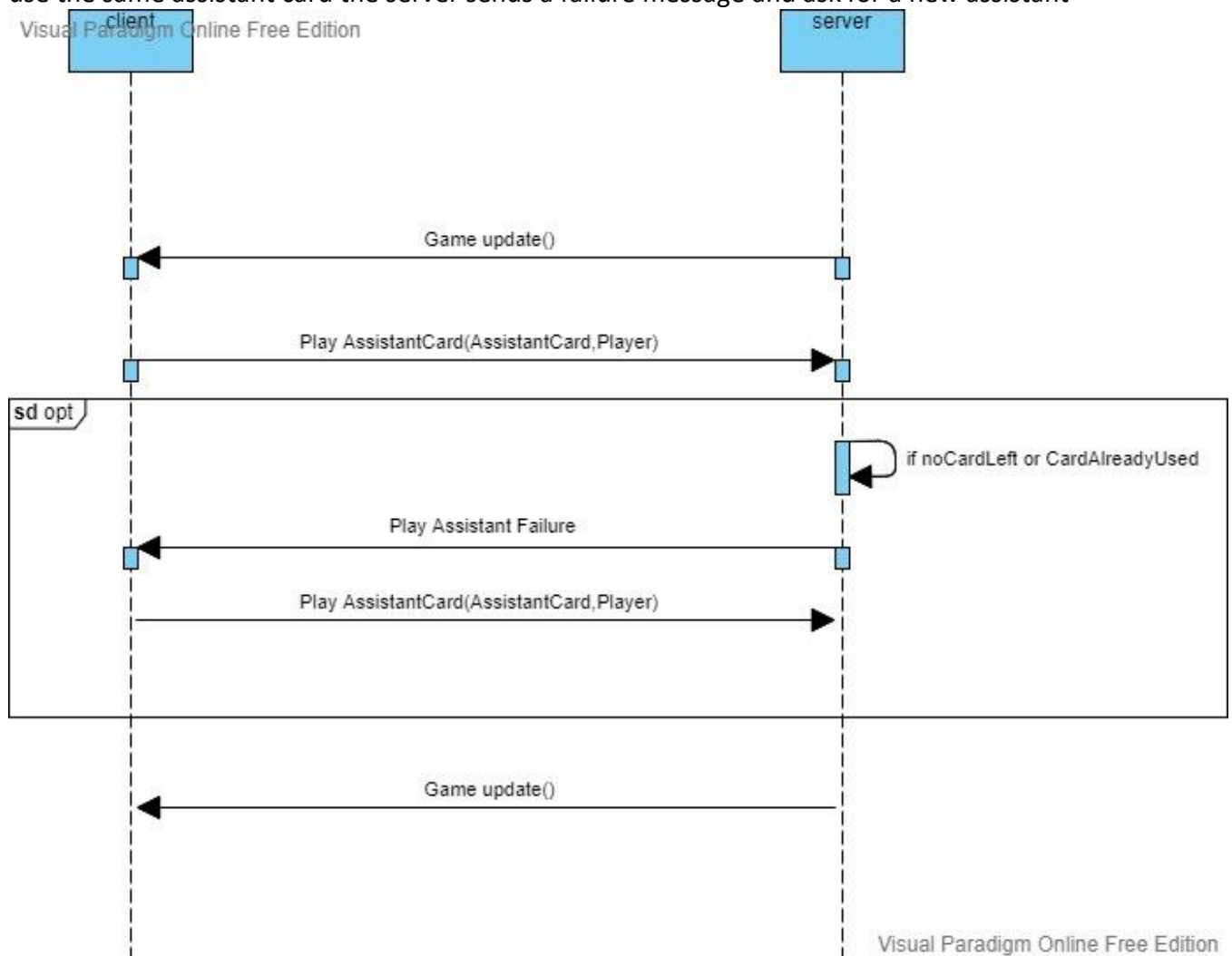


2.2 Create or join game



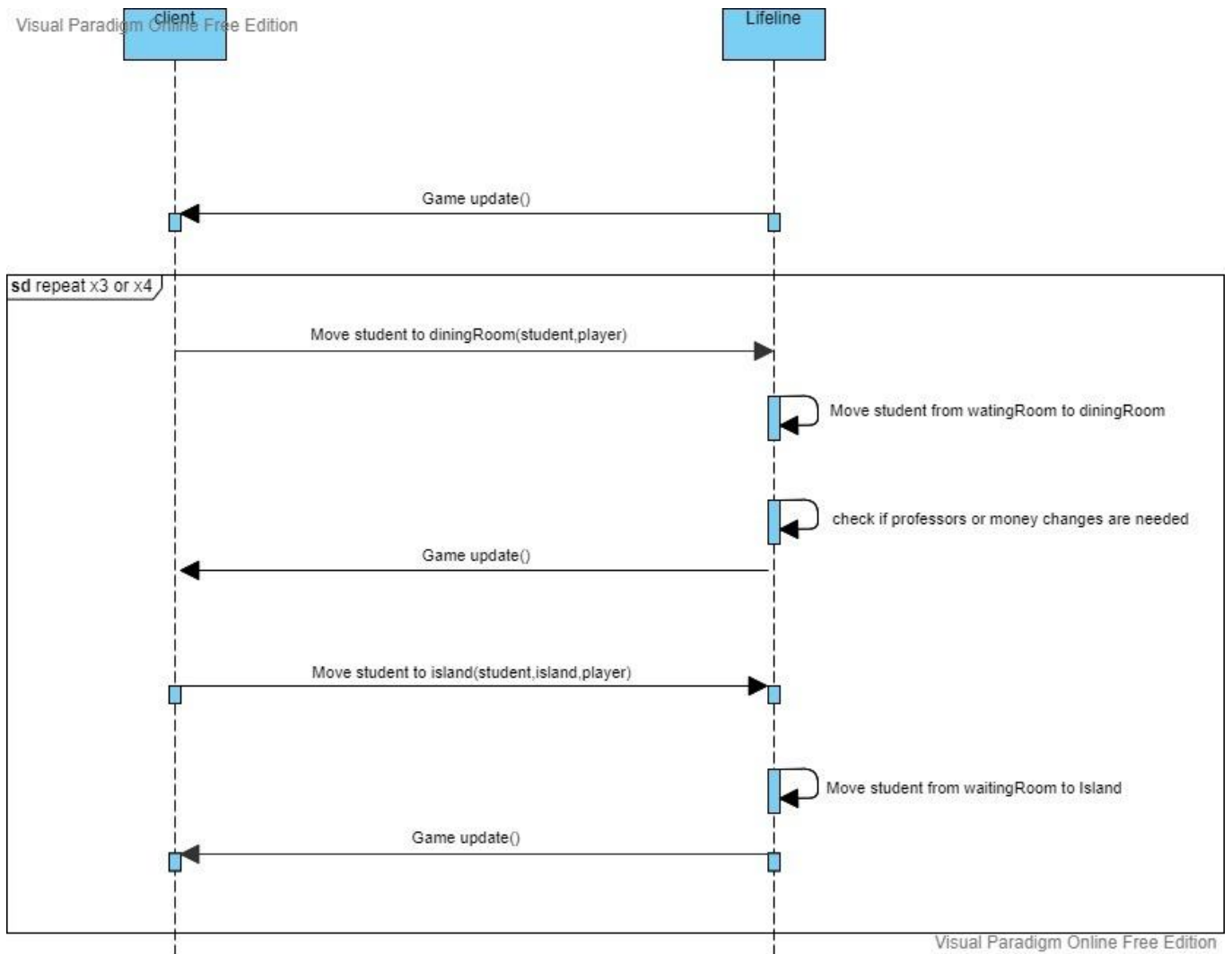
2.3 Planning Phase

Once the game is started and updated the first part of the turn is to play an assistant, if two player use the same assistant card the server sends a failure message and ask for a new assistant



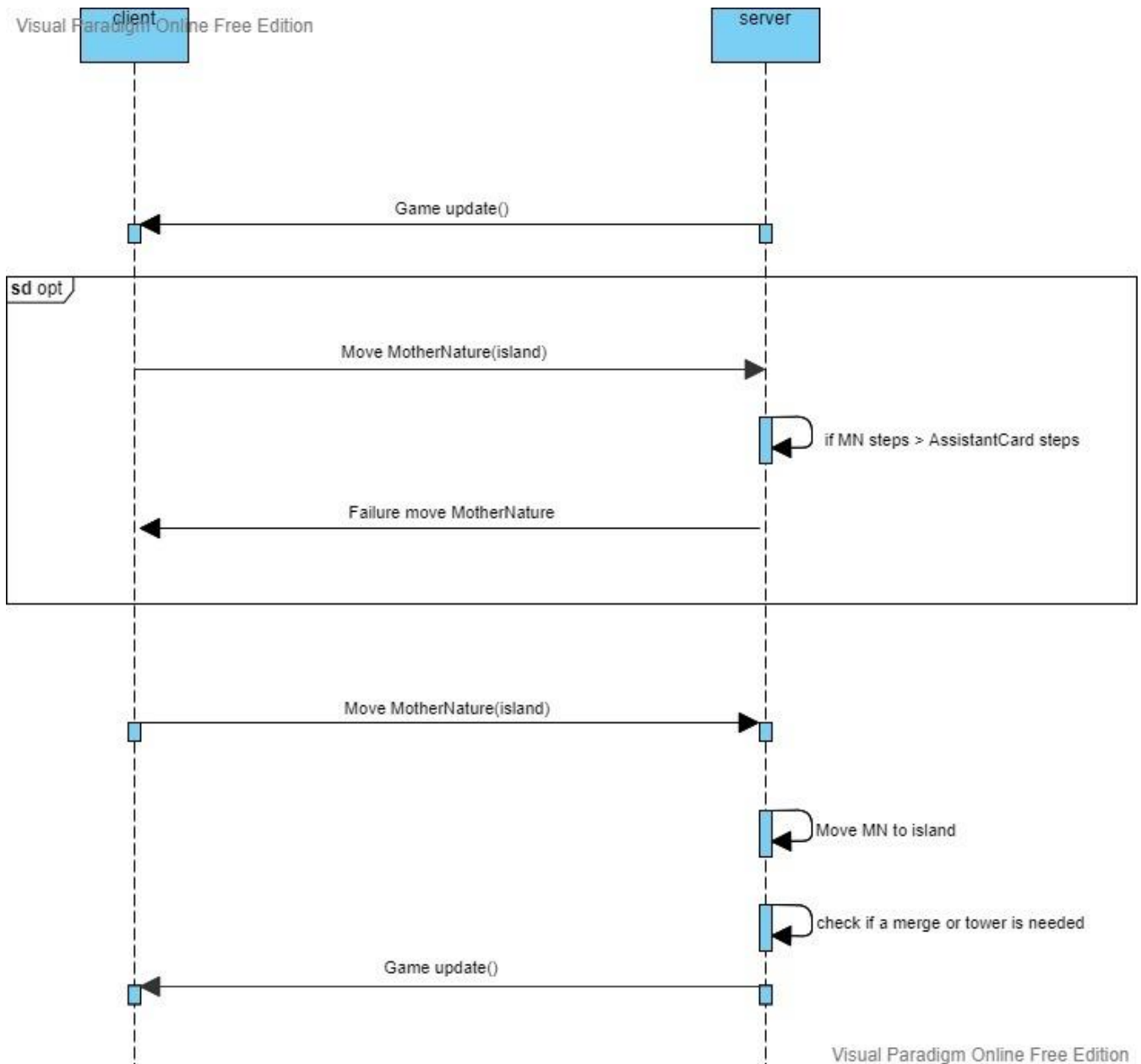
2.4 Action Phase_1

In the first part of the action phase the each player can move three student to the dining room or the an island



2.5 Action Phase_2

In the second part of the action phase the player moves mother nature, if for any reason the island selected is not usable (ex. Too many steps) the server sends a failure message and asks for a new island



2.6 Action Phase_3

In the last part of the Action phase the player selects the cloud he wants to draw the students

