Comunication protocol

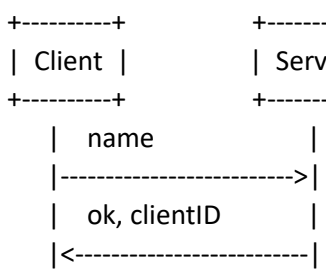
For the message exchange we used java serialization. The notification classes implements Serializable and NotificationInterface, that is composed by 3 methods:

- void accept(NotificationVisitor v) used by ClientUpdater (client) and ModelUpdater (server) for executing the methods of the serialized class using visitor pattern.
- void setClientID(int clientID) for setting the clientID.
- int getClientID() let the server know which client sent the notification and the client if the notification is supposed to call the class methods or if it is an update of the game state.

All the notifications from the server to the clients (except for setName, move, build, elimination, disconnection, win) request the reply only from the client with the same clientID as the one stored in the notification.

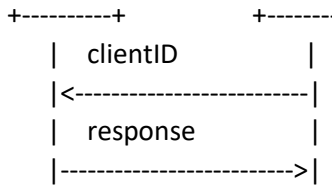
The notifications are:

SetNameNotification



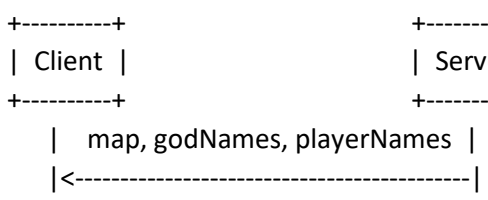
Client: send the name to the server
Server: if the name is valid, send the setNameNotification with ok = true and the ClientID of the client, otherwise ok = false and the client send another name

AskForReloadStateNotification



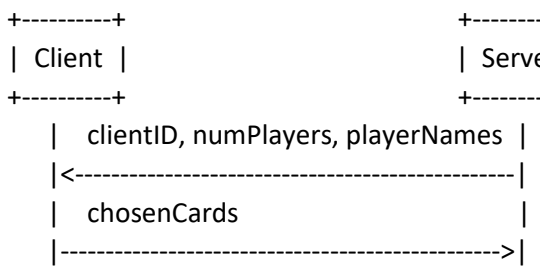
Server: send the AskForReloadStateNotification
Client: reply with the boolean response

LoadGameNotification



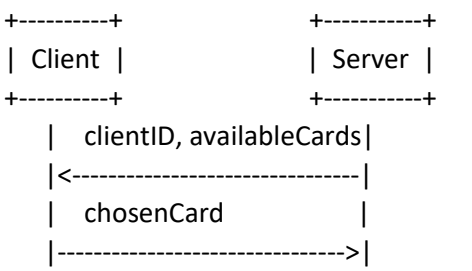
Server: send all the informations for loading the game

ChooseCardListNotification



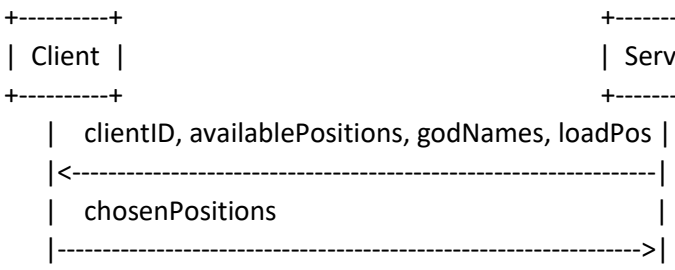
Server: send all the informations that the client need to know for choosing the cards
Client: reply with the list of chosen cards

ChooseCardNotification



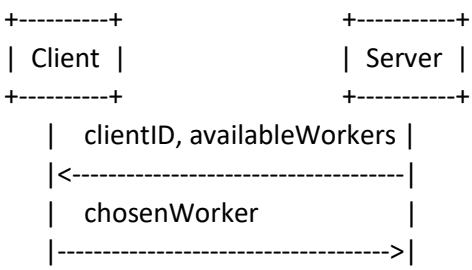
Server: send the list with all the available cards
Client: reply with the chosen card

FirstPositioningNotification



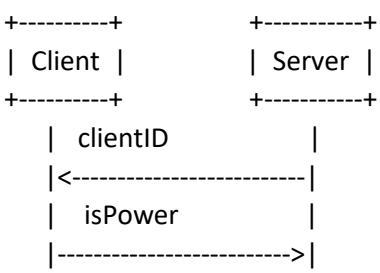
Server: send all the available positions, the names of the chosen gods and the boolean loadPos. If loadPos is true the client has to do the first positioning, otherwise it has to render the workers of clientID in availablePositions
Client: send the list of chosen positions

ChooseWorkerNotification



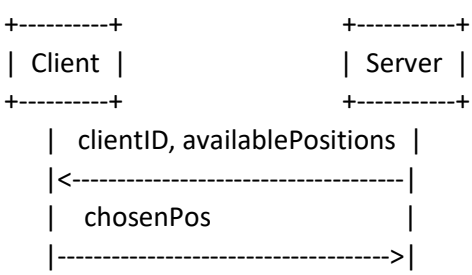
Server: send all the available workers
Client: send the position of the chosen worker

SetPowerNotification



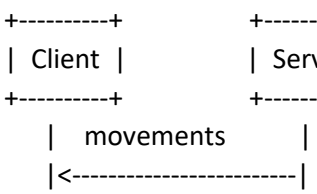
Client: set isPower to true if he wants to use the god's power

ChoosePosNotification



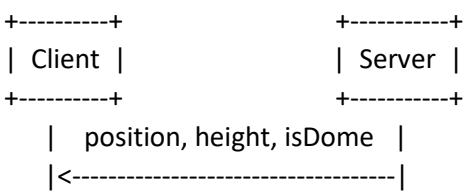
Server: send all the available positions
Client: reply with the chosen position

MoveNotification



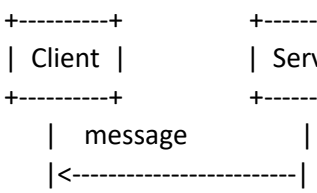
Server: send the movements to all the clients. The movement has the old and the new position of the worker

BuildNotification



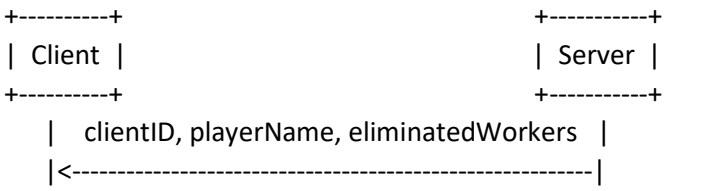
Server: send the position, the height of the block built, isDome = true if it is a dome

DisconnectionNotification



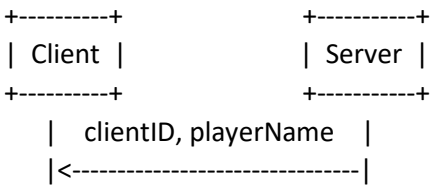
Server: send the message to show
Client: show the message and close the connection

EliminationNotification



Server: send the id of the eliminated client, his name and the positions of his workers
Client: show a message and deletes the workers from the table

WinNotification



Server: send the id and the name of the winner
Client: show a message and closes the connection