Network protocol «Edgy» (group 14)

Client to Server

"command&arguments"

For all commands, the same technique is used. The String sent from client to server can be split into several substrings that were separated by ampersands. The first substring is the command and has always a length of five characters. It is followed by the individually required arguments (Ex: nuser&Baumgartner).
If helpful an alias has been defined which can be used instead of the five characters command. The Alias is mentioned in brackets.

For ensuring clarity, commands that can be accessed within the chat begin with a slash (Ex: /list).

"login&username&password"

The client can log in with the command "login username password”.
The method login(String username, Player player) is called. It checks if user is already online.
Message will be sent from Server to Client. Ex: “The player helloweekend successfully connected to the server”.
Saves connection in SQLite table.

login Rüdiger

"nuser&username&password" (newuser)

A new user is created with the command "nuser username". This can be done by clicking on the 'register' button on the login window. The username is then checked for its uniqueness and the password has to have at least four chars.

Example: nuser&Coco&Ideen++

"cuser&newUsername" (changeuser)

The command "cuser" is used for changing the username.
The function changeUsername(String newUsername, Player player). The new name is again checked for its uniqueness and the username changes. This action can only be performed if a user is already logged in.

Example: cuser&Platon

"lsort"(highscore)

The command "lsort" prints out a list on the console of all the players saved in the SQLite table in accordance to their score.

"pumsg&message"

The command "pumsg" sends a public message to all clients including the own by calling chat class and publicMessage (String msg).

Example: puMsg&Hello Platon

"nonmg&message"

The command "nonmg" sends a nonreciprocal message to all the Clients except the own by calling chat class and nonreciprocal (String msg).

Example: nonmg&Hello Platon

"prmsg&username&message"(privatemessage, private)

The command ‘prMsg’ sends a private message to a specific client.

Example: prmsg&Platon&Where is Sokrates?

"echo_&message"(echo)

Sends a message to server which is directly returned.

Example: echo_&Hello Echo

"help_" (help)

If the user is not sure what commands can be entered, "help" supplies a menu with all commands and their basic function. By entering "help", the function help() is called printing out the menu on the console of the client.

"maopt" (options)

Lists game specific commands.

"list_" (list)

If the user wanders who else is online, "list" calls the function list() that prints out a list of all clients that are currently logged in the server.

"quit_" (quit)

The command "quit" leads to the client being disconnected from the server. The function disconnect() that is called first logs out the player if this did not already happen. Next the function tries to close is, os and clientSocket informing the client of a successful disconnection. If that does not work, it leads to a hard disconnection informing the client that something went wrong.

"lgout"(logout)

With the command "logout", the client can log out. The function logout() that is called prints out that the player successfully logged out. Command can only be executed if player is not logged in.

"nwmat" (newmatch)

A new match can be created by entering "nwmat". The new match automatically gets an ID in form of an integer so other players can enter a particular match. The same can be done by clicking on the button 'Create new match'.

"joinm&machtchID" (joinmatch)

If the player wants to join an already existing match, a click on the button 'join' sends the command "join" plus the matchID of the particular match to the server.

Example: joinm&1

Output: The player joins match 1 as far as there are not already four players in the match.

"lstgm"(matchlist)

Before joining a match, the command "lstgm" lists all current games inclusive the player number of the respective match to the player's console. The new player can therefore directly see which matches can be joined and what number they have.

"prtpc"

The command "prtpc" can be entered to print all pieces in their true current configuration to the console. Especially after flipping or turning the piece, this command shows the alignment of all pieces at this moment.

"board"

The client requests the current board String with the command "board".

"flipp&pieceID"

A piece can be flipped upside- down with the command "flipp pieceID" where the pieceID is a particular integer between 1 and 21 describing the piece.

Example: flip&4

Output: The piece number 4 is flipped.

"turnl&pieceID" (turnleft)

"turnl pieceID" turns the piece counterclockwise by 90 degrees. The pieceID is between 1 and 21.

Example: turnl&6

Output: The piece number 6 is turned to the left.

"turnr&pieceID" (turnright)

The command "turnr pieceID" analogous to "turnl" turns the piece clockwise by 90 degrees. The pieceID is again between 1 and 21.

Example: turnl&7

Output: The piece number 7 is turned to the right.

"moved&pieceID&row&column" (move)

After the game has started and it is your turn to move a piece to the board, "moved pieceID row column" has to be entered. The pieceID is an integer between 1 and 21 and represents which piece has to be moved. The row and the column are integers between 0 and 19. The row and column position on the board are the coordinates where the center of the piece is moved to. If the piece is moved by drag and drop, the coordinates are calculated automatically.

Example: moved&3&5&7

Output: If the move gets accepted, the piece number 3 is moved to the board at position (5|7).

"skipp" (skip)

If the player realizes that no further move is possible, "skipp" can be entered. This leads to the skipping of this player in all further rounds until the match is finished and the score inclusive ranking is printed.

"highs"

The client can request the highscore by entering the command "highs".

"lobby"

The client can request an update of the lobby (number of players, lobbies) by entering the command "lobby".

"lomsg&matchID&message"

The message (argument two) is sent to all clients who have joined a particular match (argument one).

Example: lomsg&2&Hallo Lobby

"chpwd&newPassword" (changepassword)

If the client wants to change his or her password, "chpwd" followed by the new password resets the login to the new password. This message is also send when changing the username by clicking on the 'Change password' button and entering the new password.

Example: chpwd&Ferien Ahoi

"rankm"

By entering "rankm", the client requests the total rank over all games, the points as well as the username.

"force&username"

This command forces a specific user to surrender in a match. Who should be forced to surrender can be determined by entering the respective username (argument one). This command is considered to be a cheat and should only be used for testing or demonstration purposes.

Example: force&Platon

"kill_&username"

Another client can be disconnected from the server by entering "kill_" plus the respective username (argument one). This command is considered to be a cheat and should only be used for testing or demonstration purposes.

Example: kill&Platon

"pongg"

By receiving a "pongg" signal from the client, the server gets informed that the client is still online and the connection is active.

"pingg"

By sending "pingg" to the client, the server confirms to have received a signal from the client and thus knows that the connection to this client is not lost.

Server to Client

"board&boardarray"

When the board was updated, a new string containing the board information (argument one) is sent to all players in the match with the command "board". The board array is composed of chars symbolizing a yellow (y), red (r), blue (b), green (g) or empty (*) board quadrant.

Example: board&y******bb***g****...

"delpi&pieceID"

The command "delpi" leads to the removal of a particular piece (argument one) from the GUI after it was successfully moved to the board. The pieces are numbered from 1 to 21.

Example: delpi&12

"error&message"

The command "error" prints an error message (argument one) to the client's GUI.

Example: error&Dies ist eine Warnung

"highs&highscoreString"

The String "highs" is sent to update the high score list on the client's GUI. The high score list (argument one) is sent in form of a String containing userID, username, score and online status from each user. The different users and their information is separated by colons.

Example: highs&1002:Platon:99:1

"inf&message"

The command "info" plus the attached message (argument one) leads to the display of a message in the player's chat.

Example: print&Hello Platon

"login"

After receiving the login request from the client, the server checks the correctness of the login data. Depending on the output, the server sends the command "login" plus whether the login worked or not (argument one) and an according message (argument two) to the client. In case of a successful login, the username (argument three) and the score (argument 4) are sent as well.

"login OK message username score"

The status ok symbolizes a successful login.

Example: login&OK&You did it&Aristote&80

"login ONLINE message"

In this case, the login failed because this player is already online.

Example: login&ONLINE&You failed

"login WRONGPWD message"

The status wrong password suggests that the player entered an incorrect password.

Example: login&WRONGPWD&You failed

"login WRONGUSR message"

If the username is not present in the user database the status wrong user is sent.

Example: login&WRONGUSR&Wrong username

"lost_&message"

All players who did not win the match receive the command "lost_" plus an according message (argument one).

Example: lost_&You lost!! Rank: 2

"lslob&L#1 L#2 L#3"

"lslob" lists the number of players per lobby plus the lobby status. The lobby status symbolizes whether the match is not full yet (1), ongoing (2) or finished (3).

Example: lslob&4:3;2:1;

"nuser " (newusername)

If a new user was requested, the server sends the command "nuser" plus whether the user could be created or not (argument one). A human readable message is attached (argument two). There are two possible cases:

"nuser NOTUNIQUE message"

The user could not be created.

Example: nuser&NOTUNIQUE&The username Aristote already exists. Please create another name or login.

"nuser OK message username"

The user was successfully registered in the data base. The username is in this case sent as well (argument three).

Example: nuser&OK&You successfully created a new user account on Edgy&Aristote

"piece&"

The command "piece" is sent to confirm one of the following actions (argument one) the client wants to execute with one of the 21 pieces (argument two):

"piece flip pieceID"

The argument "flip" leads to the vertical flip of the particular piece.

Example: piece&flip&11

"piece turnleft pieceID"

"turnleft" leads to a 90-degree rotation to the left.

Example: piece&turnleft&21

"piece turnright pieceID"

"turnright" leads to a 90-degree rotation to the right.

Example: piece&turnright&16

"print&message"

The command "print" plus the attached message (argument one) prints a message to the client's console.

Example: print&Hello Platon

"rankm&rank username points status"

To update the total rank over all games, the current points and the current rank of the ongoing game are sent to the client after each move. The command "rankm" is followed by the ranks, usernames, points and game states of the four players (argument). The information of one player is separated by colons and the different players are separated by semicolons.

Example: rankm&12:Platon:60:skipped;4:Sokrates:3:wait;12:Kant:5:turn;6:Mill:12:skipped;

"start&matchID&color"

When a match is complete, all four clients who have joined this match receive the command "start" plus the matchID of this match (argument one) and their color for this game in form of an Integer between 1 and 4 (argument two).

Example: start&2&3

"win__&message"

After the match has finished, the server calculates the ranks and sends to the winner the command "win__" plus an appropriate message to congratulate this player (argument one).

Example: win__&You won the game

"cuser"

After the client has sent the request to change the current username to a new one, the server replies whether the change was successful or not (argument one). Additionally, a human- readable message is attached describing the state of the changed username (argument two). In case of a successful change, the new username is sent as well (argument three).

"cuser OK message newUsername"

The status "OK" tells the receiving client that the change could be completed.

Example: cuser&OK&You have been successfully be renamed to newUsername& newUsername

"cuser FAILUNIQUE message"

In this case, the change failed because the new username was not unique.

Example: cuser&FAILUNIQUE&Your preferred username is already taken. Please choose another one.

"chpwd"

The command "chpwd" is used to change the password of a user. The server informs the client whether this process was successful or not (argument one) and directly sends a message describing why or why not the password could or could not be changed (argument two). There are three possible cases:

"chpwd OK message"

The password could be changed successfully.

Example: chpwd&OK&You have successfully changed your password."

"chpwd FAILREQUIREMENT message"

The password could not be changed because the new password did not match the required length of four character.

Example: chpwd&FAILREQUIREMENT&The password should be at least 4 characters long.

"chpwd FAILSQL message"

The password could not be changed due to an error in the SQL database.

Example: chpwd&FAILSQL&The password couldn't be saved. Please try again!

"force"

The command "force" forces the player to surrender.

"pongg"

By sending "pongg" to the client, the server confirms to have received a signal from the client and thus knows that the connection to this client is not lost.

"pingg"

By receiving a "pingg" signal from the server, the client gets informed that the server is still online and the connection is active.

"kill_"

This command disconnects the client from the server.

"priva&username&message"

An incoming message with the command "priva" is a private message that was sent from another user (argument one) to this client. The message gets displayed in the chat window (argument two).

Example: priva&Aristote&Hurry up!

"moved&status&message"

The command "moved" informs the client whether the move was accepted or not (argument one). A message that describes why a move was or was not accepted is sent as well (argument two).

Example: moved&FAIL&Sorry, not your turn.

"score&highscore"

After moving a piece to the board, the current total score of the player is updated (argument one).

Example: score&40

"trolo"

This command leads to an annoying sound being displayed.

"start&color"

This command is sent after four players have joined the same match. According to the color, the individual matchGUIs are displayed.

Example: start&Red

"hint_&pieceID&row&column"

The command "hint_" suggests a piece (argument one) the player can move to a particular position (argument two, three). The piece and the position get highlighted on the GUI. If no move is possible anymore or the first move should touch an edge of the board, an according message is printed.

Example: hint_&5&7&19

"delpi&pieceID"

The command "delpi" leads to the piece (argument one) being deleted after an accepted move.

Example: delpi&10

"turn_"

The command "turn_" informs the player that it is his or her turn. A sound is played.