Protocols

The rules governing client-server interaction

Every individual command sent by the client should be a String, by which I mean a sequence of ***one-byte*** ASCII characters followed by a network newline (“\r\n”). The server will respond with either a String or integer, depending on the nature of the transmission.

For example, usernames and gameIDs will be sent by the server as Strings while return codes will be sent as four-byte ints. While this is pretty much self-evident, in the document below String transmissions will be coloured green while ints will be coloured blue.

Since I’m not really sure what the convention is, I should make it clear that when I say the server send an int, I mean it send the 4 bytes that correspond to that int and that’s it. No network newline afterward. So if you’re expecting 4 ints and then a string, perform 4 four-byte reads to get the 4 ints and then start reading the string, keeping an eye out for the “\r\n.”

We note a couple constants:

- -3 – A return code of -3 means generally that something is wrong with the format of the command. For example, a login command that places a comma between username and password instead of a space, or a move command that omits the “->”
- -2 – The server returns -2 if a client attempts to issue a command without logging in a user. “Logging in” a user means that the server will associate the client with the logged in user, allowing it to access data and make moves on the client’s behalf. The only requests clients can make of the server without having logged in a user are to either login a user or create a new account. All other requests for data must be preceded by the logging in of a user.
- -1 - A response of -1 from the server always means that there was a server error and something went wrong server-side. If the connection hasn’t been terminated, the client can try again if they’d like, but should not assume that the last command they issued was processed

Now, we address each potential client-server interaction individually.

- Creating a new account for a user
    - Performing this action will automatically log out and previously logged in user. A client cannot access the data of two players at a time
    - Client – “create username password”
        - ***<u>Note:</u>*** the client should only accept usernames and passwords that do not contain either spaces or commas and are non-empty. Otherwise, corruption of data could occur. If the client does not check for this, the server will not allow the account to be created
    - Server – responds with one of the following:
        - -3 – the content of the command is invalidly formatted
        - -1 – server error; request could not be properly processed
        - 0 – account creation successful, client can now make requests on behalf of the new user
        - 1 – the given username is already in use
        - 2 – the username and/or password aren’t correctly formatted. For example, one is non-empty, or contains a comma.
- Logging in a user
    - Performing this action will automatically log out and previously logged in user. A client cannot access the data of two players at a time
    - Client – “login username password”

    - Server – responds with one of the following:
        - -3 – the content of the command is invalidly formatted
        - -1 – server error; request could not be properly processed
        - 0 – login successful, client can now make requests on behalf of the given user
        - 1 – the given username does not exist
        - 2 – the given password is invalid for the given username
        - 3 – the command is invalid in some way
    - If the login is successful, the server then does the following:
        - First, it sends an integer, which corresponds to the number of ongoing games that the logged in user is involved in
        - Then it sends that many batches of transmissions, where each batch corresponds to the following data points, each sent on a separate line:
            - GameID
            - Name of White player
            - Name of Black player
            - State - 0 if it's White's turn, 1 if it's Black's
            - Turn – the current turn number
            - White check – 1 if White is in check, 0 otherwise
            - White checkmate – 1 if White is in checkmate, 0 otherwise
            - Black check -1 if Black is in check, 0 otherwise
            - Black checkmate – 1 if Black is in checkmate, 0 otherwise
            - Promotion needed – 1 if the player whose turn it is needs to promote a pawn before their turn can end, 0 otherwise
                - Tells the client that, if it is their user's turn, they should prompt the user to promote their pawn
                - See below for how to promote a pawn
        - If the server encounters a critical error, such as a flaw in its local database, it will send -1, but only BETWEEN batches, never in the middle of sending one. If part of a batch of data has been received, you may assume the transmission will proceed as normal at least until the end of the batch
        - The client, of course, is free to use as little or as much of this information as it pleases
- Creating a new game
    - If the client has already logged a user in, and the user wants to create a new game
    - Client – "creategame gameID"
        - Note: gameIDs cannot include commas, and must contains at least one alphabetical character
    - Server – responds with one of the following
        - -3 – the content of the command is invalidly formatted
        - -2 – the client does not have a user logged in, so the request couldn't be handled
        - -1 – server error; request could not be properly processed
        - 0 – game was successfully created
            - Whichever user creates the game automatically plays White
        - 1 – the given game already exists
        - 2 – the given gameID is invalidly formatted
- Joining a game
    - If the client has already logged a user in, and the user wants to join a particular game
    - Client – "joingame gameID"
    - Server – responds with one of the following:
        - -3 – the content of the command is invalidly formatted

        - -2 – the client does not have a user logged in, so the request couldn’t be handled
        - -1 – server error; request could not be properly processed
        - 0 – the user has successfully joined the game
        - 1 – the given game does not exist
        - 2 – the given game already has two players
        - 3 – the user is already in the given game
- Loading a game
    - If the client has already logged a user in, and the user wants to load a game that he is playing and view the board
    - Client – “loadgame gameID”
    - Server – responds with one of the following
        - -3 – the content of the command is invalidly formatted
        - -2 – the client does not have a user logged in, so the request couldn’t be handled
        - -1 – server error; request could not be properly processed
        - 0 – success; the user is a player in the given game, and the board data will be sent shortly
        - 1 – the given game does not exist
        - 2 – the user is not in the given game
    - If the server responded with success, it then sends the contents of the requested game’s data file, one line at a time. For details on how these files are formatted and how game data is encoded, see the file named Data in this directory
- Sending a move
    - If the client has already logged a user in, and the user wants to make a move in a game that he is playing
    - Client – “move gameID src_row,src_col->dest_row,dest_col”
        - src_row,src_col is the square occupied by the piece that is moving
        - dest_row,dest_col is the square the piece is moving to
        - Where row 0 is white’s back row, and column 0 is the first column from the left (from white’s perspective)
    - Server – responds with one of the following:
        - -3 – the content of the command is invalidly formatted
        - -2 – the client does not have a user logged in, so the request couldn’t be handled
        - -1 – server error; request could not be properly processed
        - 0 – move was successfully made; game has been updated
        - 1 – the move is invalid
        - 2 – it is not the user’s turn to make a move
        - 3 – it is the user’s turn, but they need to promote a pawn rather than make a normal move
        - 4 – move was successfully made, and now the user needs to promote a pawn
- Promoting a pawn
    - If the client has logged a user in, and the user wants to promote a pawn which has reached the enemy’s back row
    - Client – “promote gameID charRep”
        - Where “charRep” is one of the following:
            - “r” – promotes the pawn to a rook
            - “n” – promotes the pawn to a knight
            - “b” – promotes the pawn to a bishop
            - “q” – promotes the pawn to a queen
    - Server – responds with one of the following:
        - -3 – the content of the command is invalidly formatted

        - -2 – the client has no user logged in; request could not be handled
        - -1 – server error, request could not be properly handled
        - 0 – promotion successful
        - 1 – charRep is invalid
        - 2 – it is not the user’s turn to make a move
- Logging out
    - If the user has logged out of the client, but the client doesn’t want to close the connection yet, they should send a logout request so the server can free up some memory associated with the particular user that was signed in
    - This does not close the connection the client has with the server
    - Client – “logout”
    - The server does not respond, but the client should know that it can no longer make requests on behalf of the previously logged in user