Chess Protocol

Overview:

This chess protocol is an application-level protocol for engaging in various chess games. Players may access a chess server in order to compete against each other.

All commands and responses are comprised of ASCII encoded printable characters, as well as the carriage return '\r' and newline '\n'.

Commands are always a single word, case insensitive, followed by a \r\n pair. All commands in this document will be indicated by uppercase letters, such as REGISTER, or QUIT. If the command requires parameters, they follow the command as keyword/value pairs, each on their own line, followed by a \r\n pair. Keyword/value pairs are always delimited by a ':' (colon) character, and therefore no ':' character is allowed in keyword/value pairs except as the delimiter. Multiple parameters may be present in the value field, separated by the ',' (comma) character. All whitespace is ignored in all lines, so the line:

```
Email Address:  john@doe.com\r\n
```

is equivalent to the line:

```
EMAILADDRESS:JOHN@DOE.COM\r\n
```

A command or response is complete and processed when the receiver reads a a double newline, \r\n\r\n. At this point, the command or response can be processed along with all of its parameters.

All commands or messages in either direction are acknowledged by a return message from the opposite connection. A response takes one of the two following forms:

```
OK <optional message>/r/n
<optional response keyword/value pairs>/r/n
...
/r/n
```

or

```
ERROR: <error message>/r/n
```

Registered users may engage in as many simultaneous games as they please. Player rankings are stored and updated on the server according to the Elo ranking system. Move validation and first move are determined by the players after a game has been initialized.

States:

A connection to the server can be in one of 2 states: the Authentication state and the Transaction state. All new connections to the server begin in the Authentication state. In order to move from the Authentication state to the Transaction state, a successful log in procedure must occur. A user is authenticated by providing a valid username and password to the server. New users may also register a request for access by providing a unique username and email address. After a successful authentication, the server moves into the Transaction state, where the user may request a new game, resume or pause current games, abort a current game, or request information about registered players and rankings.

Authentication State:

The Authentication state allows only three commands: REGISTER, LOGIN, and QUIT.
REGISTER:
The REGISTER command requires the following parameters:

Username: <desired username>
Password: <desired password>
Email: <desired email address>

If either the desired username or email address are taken by another registered user, the response will be an ERROR: with an optional message indicating the specific failure. If the password contains less than 4 characters, the response will be an ERROR:. If the registration is successful, the server responds with an OK, followed by an echo of the registered username and email address. An email is then sent to the registered email address containing a welcome message, as well as the registered username and password. All new users begin with a Elo player ranking of 1200.

LOGIN:
The LOGIN command requires the following parameters:

Username: <registered username>
Password: <registered password>

The username and password are transferred unencrypted in clear text over the network. For individuals who have concerns over the security of this transaction, there is an option to chill out and get over it. If the username and password are accepted, the server responds with an OK line, followed by a list of games, if any, that the user is currently engaged in. The games are listed in the following manner:

OK <number of games in progress>
Game: <game type>, <game id>, <opponent username>, <current player>, <time remaining>
Game: <game type>, <game id>, <opponent username>, <current player>, <time remaining>
...

Every individual match in the system is issued a unique game id number. This number is used to differentiate between multiple concurrent games, or to look up the move history of a particular game. At this point, the user has moved from the Authentication state to the Transaction state.

If the username does not exist, is already logged in, or the password is invalid, the server responds with an ERROR line and remains in the Authentication state.

Transaction state:

The Transaction state accepts the following commands: QUIT, REQUEST, LIST, ACCEPT, REJECT, DISPLAY, MESSAGE, ADJOURN, RESUME, ABORT, MOVE, RESIGN, DRAW, TAKEBACK, INFO, RANKINGS, and TIME.

QUIT:
The QUIT command can be issued at any time. QUIT will log the user out of the system gracefully, notifying any opponents that the user has left their games. The game timers of all current games will continue. A game is forfeit when a move timer expires whether a player is logged into the server or not. A QUIT command is responded by an OK, and the server closes the connection.

REQUEST:
The REQUEST command allows a user to announce their availability to play a new game. The request command can have the following optional parameters:

```
Game-Type: <Game/k, n/k, daily, any>
Display-Username: <true, false>
Game-ID: <id>
```

If not specified, Game-Type defaults to <any>, and Display-Username defaults to <false>. Once the command is processed, the request will be available to other registered players to view with the LIST command. If the Game-ID keyword is not present, the server responds with an OK, followed by a uniquely generated Game-ID number, as follows:

```
OK
Game-ID: <id>
```

The Game-ID keyword in a request indicates that you are trying to join another player's requested game. If the Game-ID keyword is present, a message is sent to the player associated with that Game-ID requesting that the game begin. The other player can then respond with the ACCEPT or REJECT commands. The message sent to the requesting player is as follows:

```
REQUEST
Game-ID: <id>
Username: <username> (if set to display)
Ranking: <rank>
```

If the opponent has declined to show their username, a unique id number is displayed in place of the <username>. This is necessary in order to differentiate between multiple anonymous requests.

LIST:
The LIST command allows a user to see a list of all the requests for new games. The LIST command responds with an OK line, followed by the available games in this arrangement:

```
OK
New-Game: <game id>, <game type>, <ranking>, (<opponent username>)
New-Game: <game id>, <game type>, <ranking>, (<opponent username>)
...
```

The opponent username will only be shown if the request was made with the Display-Username: true option in the REQUEST command. Games requested by the user issuing the LIST command will always display the username, but cannot be joined.

ACCEPT:
The ACCEPT command indicates to a player that you have accepted their request to join one of your requested games. The ACCEPT command can have the following optional parameters:

```
ACCEPT
```

```
Game-ID: <id>
Username: <username>
```

The Game-ID parameter is necessary only if the player has multiple game requests pending acceptance. As well, the Username parameter is necessary only if the player has multiple game requests pending from different users for the same game. If there are multiple users waiting for acceptance for the same game, only one may be accepted. The remaining players are notified as if a REJECT command was issued. The game is then removed from the available requested games list.

Once the ACCEPT command is processed, and the OK line is returned from the server, the opponent is sent a message indicating that the new game has been initialized. The game timer does not begin until the first move is made. The message sent to the opponent is as follows:

```
ACCEPT
Game-ID: <id>
Username: <username> (if specified)
```

REJECT:

The REJECT command indicates to a player that you have rejected their request to join one of your requested games. The REJECT command can have the following optional parameters:

```
REJECT
Game-ID: <id>
Username: <username>
```

Like in ACCEPT, the Game-ID parameter is necessary only if the player has multiple game requests pending acceptance. As well, the Username parameter is necessary only if the player has multiple game requests pending from different users. The rejected opponent will receive the following message:

```
REJECT
Game-ID: <id>
```

DISPLAY:

The DISPLAY command is used to get a representation of the current board state. Captured pieces are not displayed. By default, DISPLAY simply provides the positions of each piece as keyword/value pairs, as follows:

```
OK
K1: A5
Q1: A4
B1: C4
N1: C5
...
P2: F4
K2: G4
...
```

If the DISPLAY command is issued with the parameter:

```
Print-Board: true
```

then a human-readable text representation of the current board state is displayed. If multiple games are being played, the Game-ID keyword must be present as well.

MESSAGE:

The MESSAGE command is used to send a simple text message to the client of an opponent, or other logged-in player. The MESSAGE command takes the following arguments:

```
MESSAGE
Username: <username>
Game-ID: <id>
Message-Lines: <lines>
```

The Message-Lines keyword indicates the total number of \r\n pairs present in the message. This allows the client to know how many lines of text to read. If a game is in play, and the username of the opponent is unknown, the Game-ID keyword can be used to send the message to the opponent of that specific game.

The client receives the message in the following format:

```
MESSAGE
Username: <username>
Game-ID: <id> (if username is hidden)
Message-Lines: <lines>
(\r\n)
<Message line 1>
<Message line 2>
<Message line 3>
```

There is no need for an extra \r\n pair after the last message line, as the total is indicated in the Message-Lines header.

ADJOURN:

The ADJOURN command can be used to indefinitely postpone a game. The only parameter is the Game-ID to adjourn. The ADJOURN message is then sent to the client of the opponent, who can either ACCEPT or REJECT the proposed postponement. The opponent can also ignore the ADJOURN request, which has the same effect as a REJECT.

RESUME:

The RESUME command is used to resume a game that was paused by ADJOURN. The only parameter is the Game-ID to resume. RESUME will issue an error if the opponent in the given Game-ID is not currently logged into the server. The RESUME command is sent to the client of the opponent, who can either ACCEPT or REJECT the request. A REJECT response has no effect, whereas an ACCEPT will pick up the game exactly where it was left off, including the timer state.

ABORT:

The ABORT command is used to end the current game with no adjustment to the ratings. The ABORT request works the same as the ADJOURN request, except that an ACCEPT ends the game entirely.

RESIGN:

The RESIGN command is used to forfeit the current game. RESIGN is useful if you no longer wish to play the current game because you are losing. The game is then flagged as lost for the player who issued the RESIGN, and player rankings are adjusted accordingly.

DRAW:

The DRAW command is used to offer a draw to an opponent. It functions exactly the same as the ABORT command, except an ACCEPT will end the game marked as a draw.

TAKEBACK:

The TAKEBACK command is used to offer an opponent the opportunity to undo their last move. This is the only method to correct invalid moves. The opponent must ACCEPT the TAKEBACK command in order for it to apply. The timer is then reset to the time left when the original move was issued.

TIME:

The TIME command returns the current clock times of all games in progress, as well as the current player. If it is the player's turn, that player's username is displayed. If it's an opponent's turn, their username or unique id is displayed as follows:

```
OK
Game: <id>, <username>, <time remaining>
Game: <id>, <username>, <time remaining>
Game: <id>, <username>, <time remaining>
...
```

INFO:

The INFO command is used to display the profile information of the currently logged in user. This includes username, games played, Elo ranking, and other statistics.

RANKINGS:

The RANKINGS command is used to display the rankings of all the users registered with the server. Rankings are listed in the following format:

```
OK
<username>: <ranking>
<username>: <ranking>
<username>: <ranking>
<username>: <ranking>
<username>: <ranking>
...
```

(Note. Hiding usernames is somewhat pointless if the rankings are available for cross-reference in the manner. Perhaps having a profile setting to always replace your username with an id number is more appropriate?)

MOVE:

The MOVE command is used to issue moves in a game. The MOVE command takes the following parameters:

```
MOVE
Game-ID: <id>   (if playing more than one game)
Apply: <move>
```

All move values in the Apply block are written in English Algebraic notation. An ERROR response is issued if the move does not parse correctly, or is ambiguous. If the move is accepted, it is applied to the current board state, and the MOVE message is passed on to the opponent's client.