* TCP Protocol
  + It assures that every message is received by the destination
  + Easier to check if a player disconnects in the middle of a game
* Message Format
  + All messages are terminated with ‘\n’
  + All parameters are separated with spaces
  + Optional parameters are shown with brackets [type\_of\_parameter]
  + Mandatory parameters are shown like <type\_of\_parameter>

|  |  |
| --- | --- |
| **Direction** | **Message format**  **Brief explanation** |
| Client -> Server | HELP\n  Shows available commands |
| Server -> Client | OK <success\_code>\n  Success message showing what the server did |
| Server -> Client | ERROR <error\_code>\n  Error message explaining the error and possible solutions |
| Client -> Server | REGISTER <username>\n  Register user with <username> |
| Client -> Server | LIST\n  Shows list of users and their status |
| Client -> Server | INVITE [username]\n  Invetes <username> for a match or rematch last player |
| Client -> Server | ACCEPT\n  Accepts the game |
| Client -> Server | REJECT\n  Rejects the game invite |
| Client -> Server | PLAY <x> <y>\n  Makes a move on the selected square |
| Client -> Server | FOLD\n  Quit the current game |
| Client -> Server | EXIT\n  Exits the application |
| Server -> Client | BOARD <turn> <[[<>,<>,<>],[<>,<>,<>],[<>,<>,<>]]>\n  Sends updates game to client |
| Server -> Client | GAME <username>  Game started against <username> |
| Server -> Client | DISPLAY <message>  Prints <message> for the user |

|  |
| --- |
| **Error Messages** |
| Username in use |
| User doesn’t exist |
| User is busy |
| Invalid play |
| Command not valid |
| Server offline |
| User disconnected |
| Connection Lost |
| Not turn to play |
| User already registered |
| No invites pending |

|  |
| --- |
| **Success Messages** |
| User Registerd |
| Accepted match |
| Rejected match |
| Your turn to play |
| Game completed win |
| Game completed lose |
| Game completed tie |