**Documentation of the Protocol**

**Message Format**

The messages passed in this protocol are in the form of an array: [type, pid, p2id, head, msg].

* **type** - this field contains the type of message being sent over the connection. We have different codes to indicate the type of message being sent. They are
  + **200** - OK, no problems occured
  + **666** - ERR, there were errors that happened
  + **69** - QUIT, the client will quit the game
  + **55** - WAIT, the client is waiting for another player to play the game
  + **22** - PAIR, two available players will be paired for the game
  + **20** - UPDATE, there were changes that happened that needs to be updated
  + **5** - READY, the players are ready to start the game
  + **1** - ORPHAN, the player’s partner quitted and is now available to pair with another player
  + **88** - SET, used for setting a player’s username
* **pid** - this is the player’s id
* **p2id** - this is the player’s partner’s id
* **head** - extra message headers
* **msg** - the message itself, all other needed information depending on the type of message (ie, if it’s an UPDATE message, it will contain the keys etc.)

**Actions Taken by the client/server**

We are using multiprocessing in our server to scale for larger number of users. If we will just use threading, the clients will only share in the CPU memory of the main python code. However, since they are processes, they can get new CPU memory without compromising the speed.

Once a client successfully connects to the server, the server will send a message ([0, self.clientid]) containing the client's id, which will also serve as the player id. After receiving the player id, the player will send a SET message to the server which contains its chosen username. This will be used for identification purposes once pairing will be done.

After this, the client will now wait for the server to give him a partner. It will send a WAIT message to the server. Once the server receives this, it will attempt to find in the list of clients a player who is still available. If the server finds one, it will send a PAIR message which will now contain the partner’s id. Otherwise, if there is still no available message, the server will still send a PAIR message but it will contain the value -1 as the p2id, indicating that there is still no player available at the moment. Once the client receives the message, it will start the game if the p2id is not -1 or will wait first (for just a very short time) before sending another WAIT message to the server if the value of p2id is -1.

Now, if the players are paired, they will now send a READY message so that they can now play the game. For every movement of a player or any other events that are happening during the game, including death of any of the players or the players already won the game, they will send an UPDATE message to the server and the server will notify the other player of the updates that are happening. That way, the screens of each player will be synchronized with each other.

To distinguish what kind of UPDATE will happen, we will look at the msg attribute. If it contains the keys, it means just updating the movements of the players to synchronize the screens of the players. If it contains a string “G.O”, then the game is over and one of the players died. If it contains a string “win”, the players finished the game successfully.

At any time a player wishes to quit the game, it will now send a QUIT message to the server. When the server receives this, it will also notify the other player that his partner will be quitting the game. This way, the other player who did not quit will have its screen go back to the main menu and will now attempt to connect again to find another player. Also, this player will send an ORPHAN message to the server to notify it that its partner already left. The server would then remove this player from the paired list and will return it to those who are available to be paired. If this player wants to play again, it will just repeat the process to find another player.

Also, when either “G.O” or “win” is sent, both players will send an ORPHAN message to the server. That way, these two players will now be available to play the game again with other players or with their original partners.