**Options**

This section refers to the Options script in PE O (and ties in with the options in Main.rb)

* HOST: This is the machine on which the server is hosted. 127.0.0.1 is the default and this refers to your own machine. If you want to go ahead and get it working across different computers, you will have to change this value to the corresponding machine which hosts it. I suggest getting your external IP address and setting it to this.
* PORT: This is the port with which you access the server. If you are going to use your external IP address, you will have to do port forwarding on your router. There are hundreds of guides on the internet for this so I won’t go through it now. The basics are that you will want to tell the router to route all commands that come in on port XXXX to the machine connected to the router which hosts the server.
  + For example, let’s say that my external IP address is 1.1.1.1 and the port I chose is 5000. 1.1.1.1 represents your router on the global internet so when someone tries to access 1.1.1.1:5000, nothing will happen as the router is not the machine hosting the server. To solve this, we forward the port 5000 to the local IP address of the computer hosting the machine (usually of the form 192.168.x.x)
  + Future versions of PE O will come with a guide which easily allows you to host your server online without using your computer.
* VERSION: this is the version of the game that is synced with the server. If the game does not have the same version of the server, the request is denied. This is to prevent older versions of the game from accessing a newer version of the server.
* ONLINEXPGAIN: set this to true if you want online battles to have EXP gain
* ONLINEMONEYGAIN: set this to true if you want online battles to have money gain

**Advanced**

**Client expected commands**

* Connect screen
  + <CON result=x>, x=
    - 0: The game is outdated and needs to be updated
    - 1: The server is full
    - 2: Connection successful
  + <DSC> : disconnects the client
  + <REG result=x>,x=
    - 0: Username already taken
    - 1: Email already used
    - 2: Registration successful
  + <LOG result=x>,x=
    - 0: Username does not exist
    - 1: Password incorrect
    - 2: Account banned
    - 3: IP banned
    - 4: Login successful
  + <TRA user=x result=y>, x= username of player selected,y=
    - 0: Player does not exist
    - 1: Player is banned
    - 2: Player is not online
    - 3: Player did not respond or decline trade request
    - 4: Trade accepted
  + <BAT user=x result=y> , same as trade logic
  + <PNG> : command sent by server to check if client is still connected
* Trade Screen
  + <TRA party=x>, x = base 64 encoded serialised $Trainer.party (with Marshal dump, encoding necessary as server only accepts plain text and not binary)
  + <TRA offer=x> x= base 64 encoded serialised $Trainer.party[index]
  + <TRA accepted> - other player accepted trade
  + <TRA declined> - other player declined trade
  + <TRA dead> - other player exited trade or disconnected
* Battle screen
  + <BAT choices=x> - x= base 64 encoded serialised @choices[0] (contains choices of player)
  + <BAT dead> - other player exited battle or disconnected
  + <BAT new=x> x=new enemy choice (simply the index of the new enemy)
  + <BAT seed=x> x=current time of the server, used to seed the PRNG of the client to make sure all random events are synced on both clients.

**Server expected commands**

* <CON version=x> - received when client is connecting x= version of game client
* <REG user=x pass=y email=z> - info received when client is registering, y is a hash of the password.
* <LOG user=x pass=y> - info received when client is logging in, y is a hash of the password
* <TRA user=x> - received when a player wants to start a trade, x = username of requested player
* <TRA start> - NOT CURRENTLY USED
* <TRA party=x> - received when player sends party,x= base 64 encoded serialised $Trainer.party
* <TRA offer=x> x = base 64 encoded serialised $Trainer.party[index]
* <TRA accepted> - sent when client accepts trade
* <TRA declined> - sent when client declines trade
* <TRA dead> - sent when client ends trade or disconnects
* <BAT user=x trainer=y> - sent when client wants to start a battle with user x, y = base 64 encoded serialised $Trainer
* <BAT choices=x> - sent when client chooses a battle choice, x = base 64 encoded serialised @choices[0]
* <BAT new=x> - sent when client has chosen new battler, x= index of new battler
* <BAT seed turn=x> - received when client requests new seed, x=@turncount, required to make sure clients receive same seed in the same turn