Jacob Feldman
CSCI 379
Programming Project 2

Documentation

- Programming info
    - Python 2.7.13 using Python's built in networking and threading libraries.
    - Multithreaded for the ability to handle multiple client at once.
    - Four files:
        - Server.py which contains the basic server loop.
        - ServerModule.py which contains functions that the server will call on.
        - Game.py which contains most of the games math
        - Client.py which sends and receives strings from the server.
    - Server.py includes ServerModule.py and Game.py. Main server thread starts in Server.py
    - Client doesn't include anything except for general python libraries.
- Networking method
    - TCP socket that accepts incoming connections on Port 9999.
    - Accepts 1024 bytes.
- Client to Server formatting.
    - Client will prompt for an IP to connect to on start. Use LOCALHOST or 127.0.0.1 for a server on the same computer.
    - After connecting to server client will prompt for a unique username.
    - A menu will open up for the client.
- Client commands
    - Client accepts 3 commands in the syntax "command sentence"
    - Uses a single space to separate the first command. Using strings only.
    - Command "display" only displays a string of data, the client will then wait to accept another command.
    - Command "display_respond" Displays data then waits for the client to respond and relays response to server.
    - Command "display_enter" Just waits for the client to press enter, it then sends back empty data so the server knows the client is ready.
- Server commands
    - 5 commands
    - "1" beings the matchmaking services for that client.
    - "2" displays users online.

    - “3” displays users in game.
    - “4” displays user who are idle.
    - “5” disconnects the client.
- Game communication
    - The game uses only the client commands described above to communicate.
    - The client base loop looks like

```
while True:
    mSentence = cSocket.recv(1024)
    if not mSentence:
        Break
    parse(mSentence)
```

    - As long is the client follows this base it should work out fine with the server. Server sends to client a display or display_respond depending on whose turn it is.

Note: This program was tested on windows 10 x64 using python 2.7.13. Written with PyCharm and tested in windows powershell.