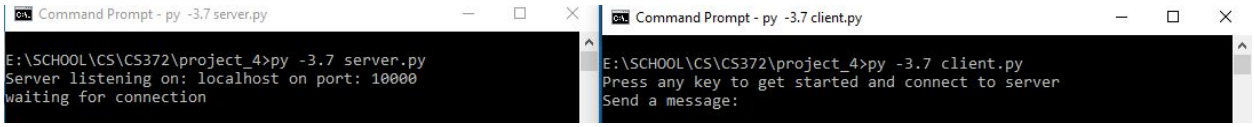
Giselle Northy
Project 4 - Client Server Chat
12/6/2020

Introduction

- In this project we implemented a simple client-server chat program using python sockets. For extra credit I turned mine into a simple ascii game of Rock Paper Scissors in which the Client plays against the Server.

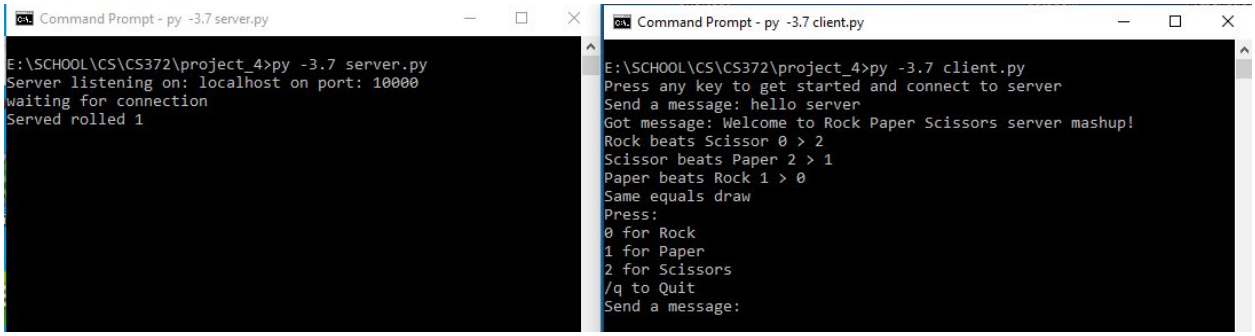
How to Run

- Written in **Python 3.7.2**
- Runs on 'localhost' and port 10000
- To run from Windows command prompt
    - Download both client.py and server.py into same folder
    - Open 2 command prompt windows and navigate to the folder
        - In 1st window start server by typing **py -3.7 server.py**
        - In 2nd window, start client by typing **py -3.7 client.py**

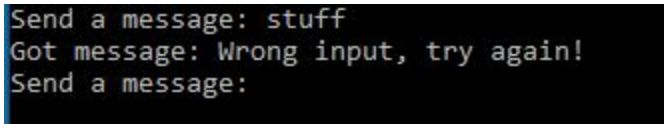


Program Details and Extra Credit - Rock Paper Scissors

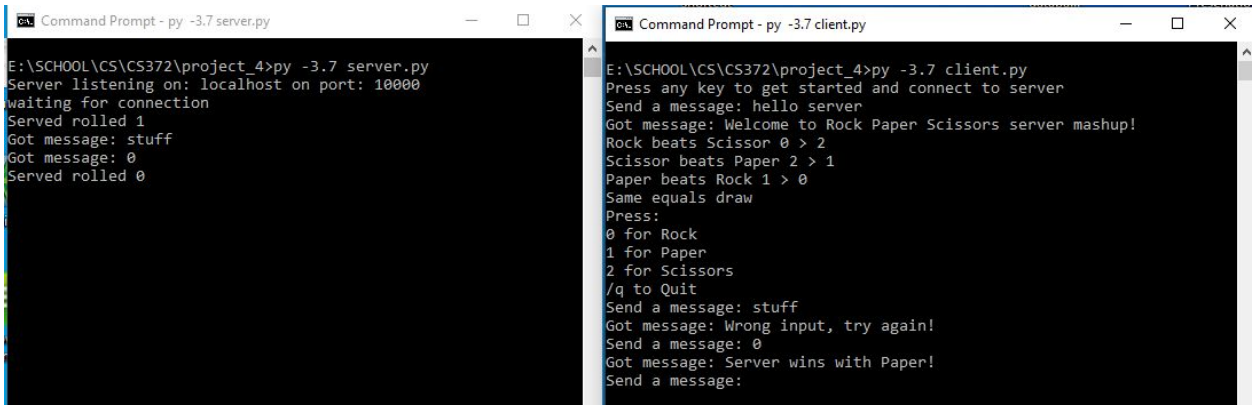
- Works by the Server randomly generating a number between 0, 1, 2 for each 'round'
    - 0 represents rock, 1 represents paper and 2 represents scissors
- On startup, the server generates a random number
    - Then for each round, it generates a new number
- A round is essentially when the client sends either 0, 1, or 2 to the server
    - The server compares the random number with the client message number
    - Then the server decides who won, client or server, and displays winner to client
        - rock beats scissor 0 > 2
        - scissor beats paper 2 > 1
        - paper beats rock 1 > 0
        - same number = tie, e.g. 0 = 0
- Communication works Synchronously with server waiting for input from Client
- Client input
    - On startup, any input will start the game and display the menu



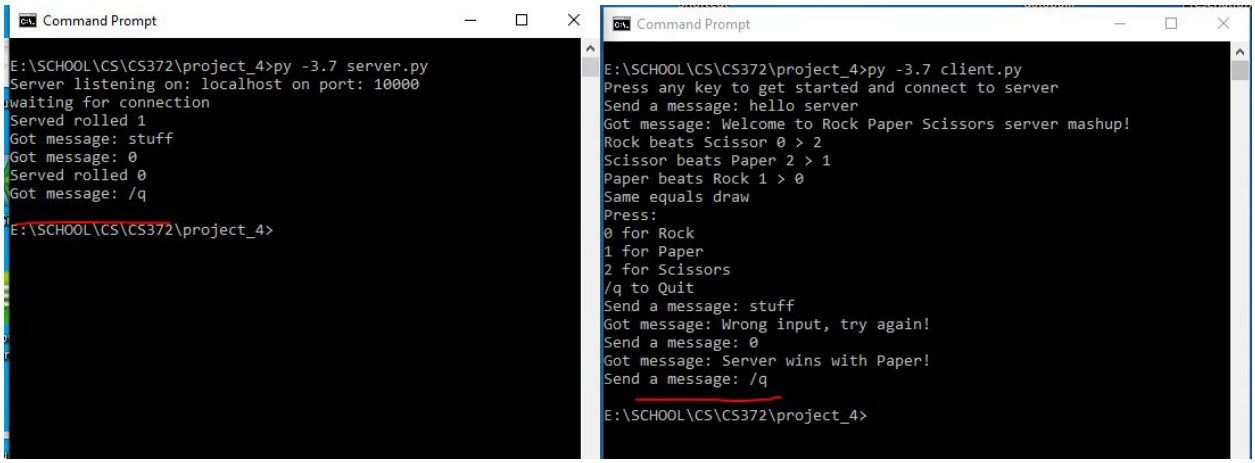
- If the client selects /q then the socket is closed on both server and client
- If the client presses anything other than the above, an ‘invalid input message’ is displayed to client



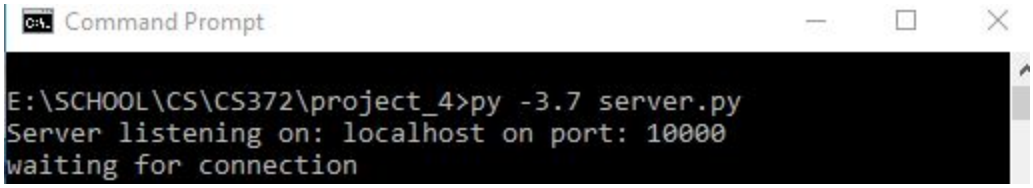
- If client selects 0, 1, or 2 then round happens and winner is displayed (client or server)



- This repeats until game is quit

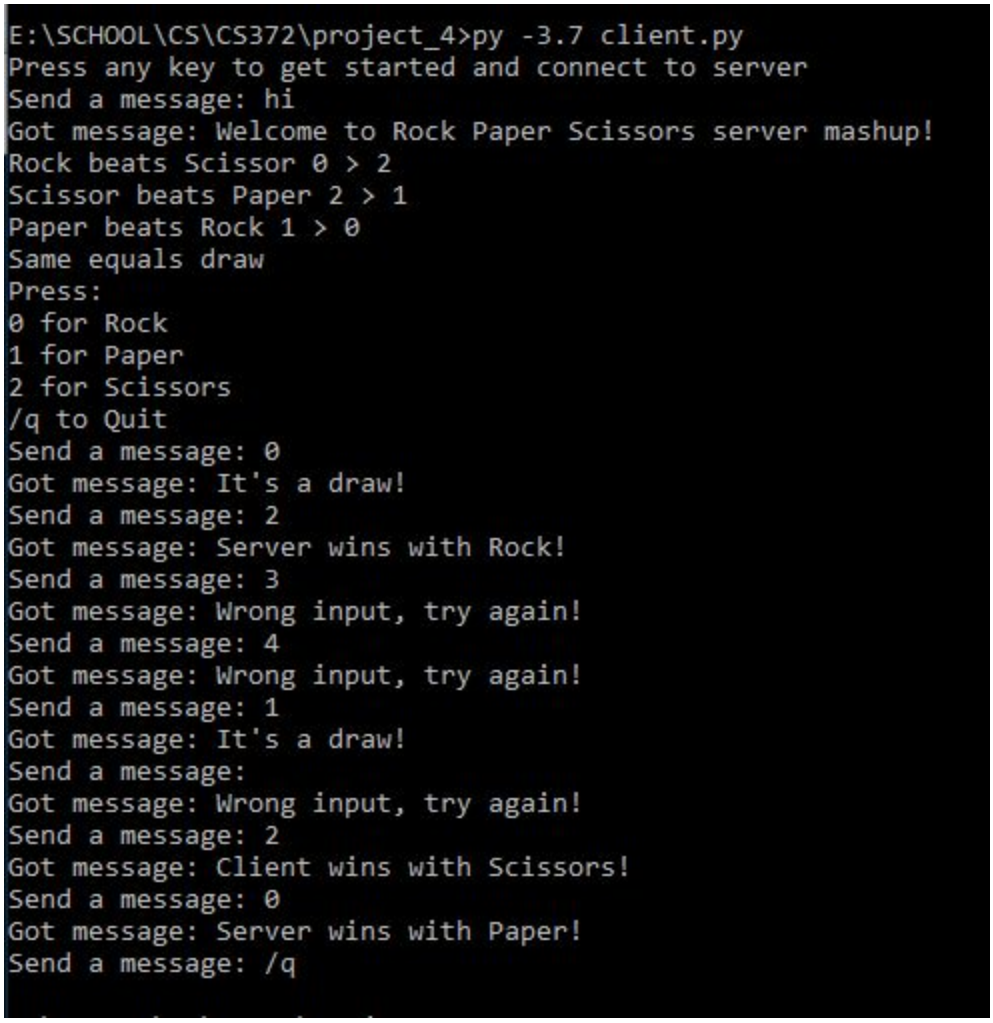


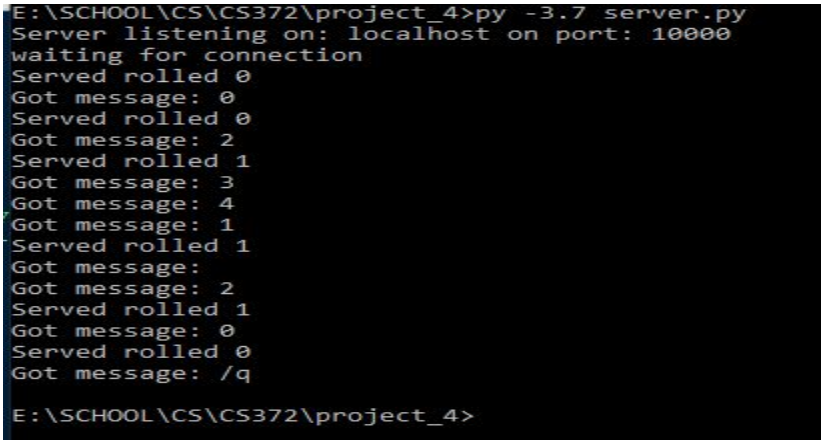
- Server
  - Does not need input but needs to be started before client.py
  - Displays listening port and host on startup



- Port is set to 10000 and host is ‘localhost’

- Displays current random number stored
- Displays message received from client





Extra Credit Code example screencaptures

```
# EXTRA CREDIT
# Method sets the game message, i.e. if it's 0, say Rock!
def getRPSstring(self, numberIn):
    stringOut = ''
    if (numberIn == 0):
        stringOut = ("Rock!")
    elif (numberIn == 1):
        stringOut = ("Paper!")
    elif (numberIn == 2):
        stringOut = ("Scissors!")
    return stringOut


# Method generates random number between 0 - 2
def randomgen(self):
    self.randomnumber = random.choice([0, 1, 2])
    self.numbertoprint = self.getRPSstring(self.randomnumber)

    print("Served rolled "  + str(self.randomnumber))

# Method decides win conditions
#   rock beats scissor 0 > 2
#   scissor beats paper 2 > 1
#   paper beats rock 1 > 0
#   rock ties rock 0 = 0
#   scissor ties scissor 2 = 2
#   paper ties paper 1 = 1
#   server random 0-2
#   0 = rock
#   1 = paper
#   2 = scissor
#   if not those then try again
#   quit /q close socket
def rockpaperscissor(self, message):

# draw condition
    retval = ''
    if (message == '/q'):
        self.sock.close()
        retval = 'quit'
    elif (str(self.randomnumber) == message):
        retval = "It's a draw!"
        self.randomgen()

# server wins
    elif (self.randomnumber == 0 and message == '2' or self.randomnumber == 2 and message == '1' or self.randomnumber == 1 and message == '0'):
        retval = ("Server wins with " + self.numbertoprint)
        self.randomgen()

# client wins
    elif (self.randomnumber == 2 and message == '0' or self.randomnumber == 1 and message == '2' or self.randomnumber == 0 and message == '1'):
        retval = ("Client wins with " + self.getRPSstring(int(message)))
        self.randomgen()

    else:
        retval = ("Wrong input, try again!")
    return retval
```

```
#Prints the welcome message and menu
welcomemsg = ("Welcome to Rock Paper Scissors server mashup!" +
              "\nRock beats Scissor 0 > 2\nScissor beats Paper 2 > 1" +
              "\nPaper beats Rock 1 > 0\nSame equals draw" +
              "\nPress:\n0 for Rock\n1 for Paper\n2 for Scissors\n/q to Quit")
print ( 'waiting for connection')
connection, client_address = chatConnection.sock.accept()
clientConnection = MySocket(connection)

# take first message to start connection, disregard for game
message = clientConnection.myreceive().decode()
clientConnection.randomgen()

# send out welcome message menu
clientConnection.mysend(welcomemsg)

# Conditinues 'game' until client sends /q
while True:
        message = clientConnection.myreceive().decode()
        print("Got message: " + message)
        result = clientConnection.rockpaperscissor(message)
        if (result == 'quit'): #close connection if /q
            break
        clientConnection.mysend(result)
```